

Full UDS Data Dictionary

QPP Analytics and Reporting

Exported on 08/20/2020

Table of Contents

1	Overview	20
2	Concepts	21
2.1	Normalized and Denormalized Data.....	21
2.2	Data Load / Snapshots.....	21
2.3	Data Quality.....	22
2.4	No Fake / Test Data in Production	22
2.5	Best Practices for Querying the UDS.....	22
3	Data Model Diagram	23
4	Submissions	24
4.1	Data Sources	24
4.1.1	Data Source documentation	24
4.1.2	Data ingestion processes.....	25
4.2	Tables	26
4.2.1	submission	26
4.2.1.1	Description	26
4.2.1.2	Columns.....	26
4.2.2	measurement_set.....	28
4.2.2.1	Description	28
4.2.2.2	Columns.....	28
4.2.3	measurement	30
4.2.3.1	Description	30
4.2.3.2	Columns.....	30
4.2.3.3	Measurement Type Field Mappings	33
4.2.4	multi_performance_rate_stratum.....	36
4.2.4.1	Description	36
4.2.4.2	Columns.....	36
4.2.5	acr_measurement_count	37
4.2.5.1	Description	37
4.2.5.2	Columns.....	37
4.2.6	submission_audit_log	38
4.2.6.1	Description	38

4.2.6.2	Columns.....	39
4.2.7	submission_request_log	39
4.2.7.1	Description	39
4.2.7.2	Columns.....	40
4.2.8	submission_request_json_flattened	41
4.2.8.1	Description	41
4.2.8.2	Columns.....	42
5	Submission Scoring	45
5.1	Data Sources	45
5.1.1	Data Source documentation	45
5.1.2	Data ingestion processes.....	45
5.2	Submission Scoring Discrepancies	45
5.3	Tables	46
5.3.1	submission_score	46
5.3.1.1	Description	46
5.3.1.2	Columns.....	46
5.3.2	score_part	51
5.3.2.1	Description	51
5.3.2.2	Columns.....	51
5.3.3	category_score.....	52
5.3.3.1	Description	52
5.3.3.2	Columns.....	52
5.3.4	measurement_set_score_part.....	53
5.3.4.1	Description	53
5.3.4.2	Columns.....	53
5.3.5	measurement_score	58
5.3.5.1	Description	58
5.3.5.2	Columns.....	58
5.3.6	measurement_score_part.....	64
5.3.6.1	Description	64
5.3.6.2	Columns.....	64
5.3.7	measures_picked	65
5.3.7.1	Description	65
5.3.7.2	Columns.....	65

6	Final Scoring.....	66
6.1	Data Sources	66
6.1.1	Data Source documentation	66
6.1.2	Data ingestion processes.....	67
6.2	Tables	67
6.2.1	final_score.....	67
6.2.1.1	Description	67
6.2.1.2	Columns.....	67
6.2.2	candidate_score.....	68
6.2.2.1	Description	68
6.2.2.2	Columns.....	68
6.2.3	targeted_review	72
6.2.3.1	Description	72
6.2.3.2	Columns.....	72
6.2.4	hvbp_provider_score/hvbp_provider_preview	74
6.2.4.1	Description	74
6.2.4.2	Columns.....	74
6.2.5	hvbp_score_percentiles/hvbp_preview_percentiles	76
6.2.5.1	Description	76
6.2.5.2	Columns.....	76
6.2.6	payment_adjustment	77
6.2.6.1	Description	77
6.2.6.2	Columns.....	77
6.2.7	apm_entity_aci_rollup_score	78
6.2.7.1	Description	78
6.2.7.2	Columns.....	78
6.2.8	apm_participant_aci_rollup_score	79
6.2.8.1	Description	79
6.2.8.2	Columns.....	79
6.2.9	neutral_payment_adjustment	80
6.2.9.1	Description	80
6.2.9.2	Columns.....	81
6.2.10	neutral_payment_adjustment_load.....	82
6.2.10.1	Description	82
6.2.10.2	Columns.....	82

6.2.11	payment_adjustment_delta_eligibility	82
6.2.11.1	Description	82
6.2.11.2	Columns.....	83
6.2.12	payment_adjustment_delta_eligibility_load.....	83
6.2.12.1	Description	83
6.2.12.2	Columns.....	83
6.2.13	final_scoring_rules.....	84
6.2.13.1	Description	84
6.2.13.2	Columns.....	84
7	Final Scoring, Score Profiler - Deprecated.....	85
7.1	Data Sources	85
7.1.1	score_profile	85
7.1.1.1	Description	85
7.1.1.2	Columns.....	85
7.1.2	final_score_score_profile	86
7.1.2.1	Description	86
7.1.2.2	Columns.....	86
8	Final Scoring, QIS	88
8.1	Data Sources	88
8.1.1	Data generation processes	88
8.1.2	quality_improvement_score.....	88
8.1.2.1	Description	88
8.1.2.2	Columns.....	88
9	Final Scoring, Payment Adjustment.....	91
9.1	Data Sources	91
9.1.1	Data ingestion processes.....	91
9.1.2	final_score_payment_adjustment_map	91
9.1.2.1	Description	91
9.1.2.2	Columns.....	91
9.1.3	Tables to be deprecated	92
10	Eligibility.....	93
10.1	Data Sources	93
10.1.1	Data Source documentation	93
10.1.2	Data ingestion processes.....	93

10.1.2.1	Associating an APM entity to its TIN-NPI clinicians	93
10.2	Tables	94
10.2.1	apm_entity	94
10.2.1.1	Description	94
10.2.1.2	Columns.....	94
10.2.2	apm_entity_organization.....	95
10.2.2.1	Description	95
10.2.2.2	Columns.....	95
10.2.3	provider_apm_entity.....	96
10.2.3.1	Description	96
10.2.3.2	Columns.....	96
10.2.4	organization	98
10.2.4.1	Description	98
10.2.4.2	Columns.....	98
10.2.5	provider	99
10.2.5.1	Description	99
10.2.5.2	Columns.....	99
10.2.6	provider_org.....	101
10.2.6.1	Description	101
10.2.6.2	Columns.....	101
10.2.7	group_special_scenario	101
10.2.7.1	Description	101
10.2.7.2	Columns.....	102
10.2.8	individual_special_scenario.....	104
10.2.8.1	Description	104
10.2.8.2	Columns.....	104
10.2.9	virtual_group.....	108
10.2.9.1	Description	108
10.2.9.2	Columns.....	108
10.2.10	virtual_group_organization	110
10.2.10.1	Description	110
10.2.10.2	Columns.....	110
10.2.11	state_region	111
10.2.11.1	Description	111
10.2.11.2	Columns.....	111

10.2.12	tin_npi_ccn_mapping.....	112
10.2.12.1	Description	112
10.2.12.2	Columns.....	112
10.3	Definition of run parameter	112
11	Eligibility, LVT Opt In.....	114
11.1	Data Sources	114
11.1.1	Data Source documentation	114
11.1.2	Data ingestion processes.....	114
11.2	Tables	114
11.2.1	group_lvt_opt_in_status.....	114
11.2.1.1	Description	114
11.2.1.2	Columns.....	114
11.2.2	individual_lvt_opt_in_status	115
11.2.2.1	Description	115
11.2.2.2	Columns.....	115
12	Eligibility, SSP.....	117
12.1	Data Sources	117
12.2	Tables	117
12.2.1	provider_apm_entity_group.....	117
12.2.1.1	Description	117
12.2.1.2	Columns.....	117
13	Measures.....	118
13.1	Data Sources	118
13.1.1	Data Source documentation	118
13.1.2	Data ingestion process	118
13.2	Tables	118
13.2.1	measure.....	118
13.2.1.1	Description	118
13.2.1.2	Columns.....	119
13.2.2	ia_measure.....	120
13.2.2.1	Description	120
13.2.2.2	Columns.....	120
13.2.3	aci_measure	121

13.2.3.1	Description	121
13.2.3.2	Columns.....	121
13.2.4	pi_measure.....	121
13.2.4.1	Description	121
13.2.4.2	Columns.....	121
13.2.5	quality_measure	122
13.2.5.1	Description	122
13.2.5.2	Columns.....	122
13.2.6	cost_measure.....	123
13.2.6.1	Description	123
13.2.6.2	Columns.....	123
13.2.7	benchmark	124
13.2.7.1	Description	124
13.2.7.2	Columns.....	124
13.2.8	aci_measure_measure_sets.....	126
13.2.8.1	Description	126
13.2.8.2	Columns.....	126
13.2.9	aci_measure_substitute	126
13.2.9.1	Description	126
13.2.9.2	Columns.....	126
13.2.10	code_object.....	126
13.2.10.1	Description	126
13.2.10.2	Columns.....	126
13.2.11	cost_measure_submission_methods.....	127
13.2.11.1	Description	127
13.2.11.2	Columns.....	127
13.2.12	eligibility_option	127
13.2.12.1	Description	127
13.2.12.2	Columns.....	127
13.2.13	eligibility_option_additional_procedure_codes	128
13.2.13.1	Description	128
13.2.13.2	Columns.....	128
13.2.14	eligibility_option_procedure_codes.....	128
13.2.14.1	Description	128
13.2.14.2	Columns.....	129

13.2.15	eligibility_option_submission_method	129
13.2.15.1	Description	129
13.2.15.2	Columns.....	129
13.2.16	measure_category	129
13.2.16.1	Description	129
13.2.16.2	Columns.....	129
13.2.17	measure_set.....	130
13.2.17.1	Description	130
13.2.17.2	Columns.....	130
13.2.18	measure_specification	130
13.2.18.1	Description	130
13.2.18.2	Columns.....	130
13.2.19	measure_subcategory	131
13.2.19.1	Description	131
13.2.19.2	Columns.....	131
13.2.20	measure_type	131
13.2.20.1	Description	131
13.2.20.2	Columns.....	131
13.2.21	measurement_objective.....	132
13.2.21.1	Description	132
13.2.21.2	Columns.....	132
13.2.22	metric_type	132
13.2.22.1	Description	132
13.2.22.2	Columns.....	132
13.2.23	performance_option.....	133
13.2.23.1	Description	133
13.2.23.2	Columns.....	133
13.2.24	performance_option_quality_codes	133
13.2.24.1	Description	133
13.2.24.2	Columns.....	133
13.2.25	performance_option_submission_methods.....	133
13.2.25.1	Description	133
13.2.25.2	Columns.....	134
13.2.26	performance_stratum	134
13.2.26.1	Description	134

13.2.26.2	Columns.....	134
13.2.27	pi_measure_measure_sets	135
13.2.27.1	Description	135
13.2.27.2	Columns.....	135
13.2.28	pi_measure_substitute.....	135
13.2.28.1	Description	135
13.2.28.2	Columns.....	135
13.2.29	quality_measure_eligibility_options.....	135
13.2.29.1	Description	135
13.2.29.2	Columns.....	135
13.2.30	quality_measure_measure_sets.....	136
13.2.30.1	Description	136
13.2.30.2	Columns.....	136
13.2.31	quality_measure_performance_option	136
13.2.31.1	Description	136
13.2.31.2	Columns.....	136
13.2.32	quality_measure_stratum.....	136
13.2.32.1	Description	136
13.2.32.2	Columns.....	136
13.2.33	quality_measure_submission_methods.....	137
13.2.33.1	Description	137
13.2.33.2	Columns.....	137
13.2.34	submission_method	137
13.2.34.1	Description	137
13.2.34.2	Columns.....	137
14	Web Interface - Deprecated.....	138
14.1	Sources	138
14.2	Tables	138
14.2.1	wi_activity	138
14.2.2	wi_beneficiary.....	138
14.2.2.1	Description	138
14.2.2.2	Columns.....	139
14.2.3	wi_clinic.....	139
14.2.4	wi_file_process	140
14.2.5	wi_file_process_message.....	140

14.2.6	wi_measure_metadata.....	140
14.2.7	wi_measure.....	140
14.2.8	wi_organization_address.....	141
14.2.9	wi_organization_contact.....	141
14.2.10	wi_organization.....	141
14.2.10.1	Description.....	141
14.2.10.2	Columns.....	141
14.2.11	wi_registration.....	142
14.2.12	wi_submission_data.....	142
14.2.13	wi_user.....	142
14.2.14	wi_user_organization.....	143
14.2.15	wi_user_role.....	143
15	ETL / Metadata	144
15.1	Data Sources	144
15.2	Tables	144
15.2.1	data_load.....	144
15.2.1.1	Description.....	144
15.2.1.2	Columns.....	144
15.2.2	api_response.....	145
15.2.2.1	Description.....	145
15.2.2.2	Columns.....	146
15.2.3	xref_data_load_data_source.....	147
15.2.3.1	Description.....	147
15.2.3.2	Columns.....	147
15.2.4	data_source.....	147
15.2.4.1	Description.....	147
15.2.4.2	Columns.....	147
15.2.5	fs_run_metrics_diff.....	148
15.2.5.1	Description.....	148
15.2.5.2	Columns.....	148
15.2.6	fs_run_agg_metrics.....	149
15.2.6.1	Description.....	149
15.2.6.2	Columns.....	149
15.2.7	fs_run_descriptions.....	149
15.2.7.1	Description.....	149

15.2.7.2	Columns.....	149
15.2.8	tin_npi_svc_audit_log - Deprecated.....	150
15.2.8.1	Description	150
15.2.8.2	Columns.....	150

- [Overview](#)(see page 20)
- [Concepts](#)(see page 21)
 - [Normalized and Denormalized Data](#)(see page 21)
 - [Data Load / Snapshots](#)(see page 21)
 - [Data Quality](#)(see page 22)
 - [No Fake / Test Data in Production](#)(see page 22)
 - [Best Practices for Querying the UDS](#)(see page 22)
- [Data Model Diagram](#)(see page 23)
- [Submissions](#)(see page 24)
 - [Data Sources](#)(see page 24)
 - [Data Source documentation](#)(see page 24)
 - [Data ingestion processes](#)(see page 25)
 - [Tables](#)(see page 26)
 - [submission](#)(see page 26)
 - [Description](#)(see page 26)
 - [Columns](#)(see page 26)
 - [measurement_set](#)(see page 28)
 - [Description](#)(see page 28)
 - [Columns](#)(see page 28)
 - [measurement](#)(see page 30)
 - [Description](#)(see page 30)
 - [Columns](#)(see page 30)
 - [Measurement Type Field Mappings](#)(see page 33)
 - [multi_performance_rate_stratum](#)(see page 36)
 - [Description](#)(see page 36)
 - [Columns](#)(see page 36)
 - [acr_measurement_count](#)(see page 37)
 - [Description](#)(see page 37)
 - [Columns](#)(see page 37)
 - [submission_audit_log](#)(see page 38)
 - [Description](#)(see page 38)
 - [Columns](#)(see page 39)
 - [submission_request_log](#)(see page 39)
 - [Description](#)(see page 39)
 - [Columns](#)(see page 40)
 - [submission_request_json_flattened](#)(see page 41)
 - [Description](#)(see page 41)
 - [Columns](#)(see page 42)
- [Submission Scoring](#)(see page 45)
 - [Data Sources](#)(see page 45)
 - [Data Source documentation](#)(see page 45)
 - [Data ingestion processes](#)(see page 45)
 - [Submission Scoring Discrepancies](#)(see page 45)
 - [Tables](#)(see page 46)
 - [submission_score](#)(see page 46)
 - [Description](#)(see page 46)
 - [Columns](#)(see page 46)
 - [score_part](#)(see page 51)
 - [Description](#)(see page 51)
 - [Columns](#)(see page 51)
 - [category_score](#)(see page 52)
 - [Description](#)(see page 52)

- Columns(see page 52)
- measurement_set_score_part(see page 53)
 - Description(see page 53)
 - Columns(see page 53)
- measurement_score(see page 58)
 - Description(see page 58)
 - Columns(see page 58)
- measurement_score_part(see page 64)
 - Description(see page 64)
 - Columns(see page 64)
- measures_picked(see page 65)
 - Description(see page 65)
 - Columns(see page 65)
- Final Scoring(see page 66)
 - Data Sources(see page 66)
 - Data Source documentation(see page 66)
 - Data ingestion processes(see page 67)
 - Tables(see page 67)
 - final_score(see page 67)
 - Description(see page 67)
 - Columns(see page 67)
 - candidate_score(see page 68)
 - Description(see page 68)
 - Columns(see page 68)
 - targeted_review(see page 72)
 - Description(see page 72)
 - Columns(see page 72)
 - hvbp_provider_score/hvbp_provider_preview(see page 74)
 - Description(see page 74)
 - Columns(see page 74)
 - hvbp_score_percentiles/hvbp_preview_percentiles(see page 76)
 - Description(see page 76)
 - Columns(see page 76)
 - payment_adjustment(see page 77)
 - Description(see page 77)
 - Columns(see page 77)
 - apm_entity_aci_rollup_score(see page 78)
 - Description(see page 78)
 - Columns(see page 78)
 - apm_participant_aci_rollup_score(see page 79)
 - Description(see page 79)
 - Columns(see page 79)
 - neutral_payment_adjustment(see page 80)
 - Description(see page 80)
 - Columns(see page 81)
 - neutral_payment_adjustment_load(see page 82)
 - Description(see page 82)
 - Columns(see page 82)
 - payment_adjustment_delta_eligibility(see page 82)
 - Description(see page 82)
 - Columns(see page 83)
 - payment_adjustment_delta_eligibility_load(see page 83)
 - Description(see page 83)

- Columns(see page 83)
- final_scoring_rules(see page 84)
 - Description(see page 84)
 - Columns(see page 84)
- Final Scoring, Score Profiler - Deprecated(see page 85)
 - Data Sources(see page 85)
 - score_profile(see page 85)
 - Description(see page 85)
 - Columns(see page 85)
 - final_score_score_profile(see page 86)
 - Description(see page 86)
 - Columns(see page 86)
- Final Scoring, QIS(see page 88)
 - Data Sources(see page 88)
 - Data generation processes(see page 88)
 - quality_improvement_score(see page 88)
 - Description(see page 88)
 - Columns(see page 88)
- Final Scoring, Payment Adjustment(see page 91)
 - Data Sources(see page 91)
 - Data ingestion processes(see page 91)
 - final_score_payment_adjustment_map(see page 91)
 - Description(see page 91)
 - Columns(see page 91)
 - Tables to be deprecated(see page 92)
- Eligibility(see page 93)
 - Data Sources(see page 93)
 - Data Source documentation(see page 93)
 - Data ingestion processes(see page 93)
 - Associating an APM entity to its TIN-NPI clinicians(see page 93)
 - Tables(see page 94)
 - apm_entity(see page 94)
 - Description(see page 94)
 - Columns(see page 94)
 - apm_entity_organization(see page 95)
 - Description(see page 95)
 - Columns(see page 95)
 - provider_apm_entity(see page 96)
 - Description(see page 96)
 - Columns(see page 96)
 - organization(see page 98)
 - Description(see page 98)
 - Columns(see page 98)
 - provider(see page 99)
 - Description(see page 99)
 - Columns(see page 99)
 - provider_org(see page 101)
 - Description(see page 101)
 - Columns(see page 101)
 - group_special_scenario(see page 101)
 - Description(see page 101)
 - Columns(see page 102)
 - individual_special_scenario(see page 104)

- Description(see page 104)
 - Columns(see page 104)
- virtual_group(see page 108)
 - Description(see page 108)
 - Columns(see page 108)
- virtual_group_organization(see page 110)
 - Description(see page 110)
 - Columns(see page 110)
- state_region(see page 111)
 - Description(see page 111)
 - Columns(see page 111)
- tin_npi_ccn_mapping(see page 112)
 - Description(see page 112)
 - Columns(see page 112)
- Definition of run parameter(see page 112)
- Eligibility, LVT Opt In(see page 114)
 - Data Sources(see page 114)
 - Data Source documentation(see page 114)
 - Data ingestion processes(see page 114)
 - Tables(see page 114)
 - group_lvt_opt_in_status(see page 114)
 - Description(see page 114)
 - Columns(see page 114)
 - individual_lvt_opt_in_status(see page 115)
 - Description(see page 115)
 - Columns(see page 115)
- Eligibility, SSP(see page 117)
 - Data Sources(see page 117)
 - Tables(see page 117)
 - provider_apm_entity_group(see page 117)
 - Description(see page 117)
 - Columns(see page 117)
- Measures(see page 118)
 - Data Sources(see page 118)
 - Data Source documentation(see page 118)
 - Data ingestion process(see page 118)
 - Tables(see page 118)
 - measure(see page 118)
 - Description(see page 118)
 - Columns(see page 119)
 - ia_measure(see page 120)
 - Description(see page 120)
 - Columns(see page 120)
 - aci_measure(see page 121)
 - Description(see page 121)
 - Columns(see page 121)
 - pi_measure(see page 121)
 - Description(see page 121)
 - Columns(see page 121)
 - quality_measure(see page 122)
 - Description(see page 122)
 - Columns(see page 122)
 - cost_measure(see page 123)

- Description(see page 123)
 - Columns(see page 123)
- benchmark(see page 124)
 - Description(see page 124)
 - Columns(see page 124)
- aci_measure_measure_sets(see page 126)
 - Description(see page 126)
 - Columns(see page 126)
- aci_measure_substitute(see page 126)
 - Description(see page 126)
 - Columns(see page 126)
- code_object(see page 126)
 - Description(see page 126)
 - Columns(see page 126)
- cost_measure_submission_methods(see page 127)
 - Description(see page 127)
 - Columns(see page 127)
- eligibility_option(see page 127)
 - Description(see page 127)
 - Columns(see page 127)
- eligibility_option_additional_procedure_codes(see page 128)
 - Description(see page 128)
 - Columns(see page 128)
- eligibility_option_procedure_codes(see page 128)
 - Description(see page 128)
 - Columns(see page 129)
- eligibility_option_submission_method(see page 129)
 - Description(see page 129)
 - Columns(see page 129)
- measure_category(see page 129)
 - Description(see page 129)
 - Columns(see page 129)
- measure_set(see page 130)
 - Description(see page 130)
 - Columns(see page 130)
- measure_specification(see page 130)
 - Description(see page 130)
 - Columns(see page 130)
- measure_subcategory(see page 131)
 - Description(see page 131)
 - Columns(see page 131)
- measure_type(see page 131)
 - Description(see page 131)
 - Columns(see page 131)
- measurement_objective(see page 132)
 - Description(see page 132)
 - Columns(see page 132)
- metric_type(see page 132)
 - Description(see page 132)
 - Columns(see page 132)
- performance_option(see page 133)
 - Description(see page 133)
 - Columns(see page 133)

- performance_option_quality_codes(see page 133)
 - Description(see page 133)
 - Columns(see page 133)
- performance_option_submission_methods(see page 133)
 - Description(see page 133)
 - Columns(see page 134)
- performance_stratum(see page 134)
 - Description(see page 134)
 - Columns(see page 134)
- pi_measure_measure_sets(see page 135)
 - Description(see page 135)
 - Columns(see page 135)
- pi_measure_substitute(see page 135)
 - Description(see page 135)
 - Columns(see page 135)
- quality_measure_eligibility_options(see page 135)
 - Description(see page 135)
 - Columns(see page 135)
- quality_measure_measure_sets(see page 136)
 - Description(see page 136)
 - Columns(see page 136)
- quality_measure_performance_option(see page 136)
 - Description(see page 136)
 - Columns(see page 136)
- quality_measure_stratum(see page 136)
 - Description(see page 136)
 - Columns(see page 136)
- quality_measure_submission_methods(see page 137)
 - Description(see page 137)
 - Columns(see page 137)
- submission_method(see page 137)
 - Description(see page 137)
 - Columns(see page 137)
- Web Interface - Deprecated(see page 138)
 - Sources(see page 138)
 - Tables(see page 138)
 - wi_activity(see page 138)
 - wi_beneficiary(see page 138)
 - Description(see page 138)
 - Columns(see page 139)
 - wi_clinic(see page 139)
 - wi_file_process(see page 140)
 - wi_file_process_message(see page 140)
 - wi_measure_metadata(see page 140)
 - wi_measure(see page 140)
 - wi_organization_address(see page 141)
 - wi_organization_contact(see page 141)
 - wi_organization(see page 141)
 - Description(see page 141)
 - Columns(see page 141)
 - wi_registration(see page 142)
 - wi_submission_data(see page 142)
 - wi_user(see page 142)

- [wi_user_organization](#)(see page 143)
- [wi_user_role](#)(see page 143)
- ETL / Metadata(see page 144)
 - [Data Sources](#)(see page 144)
 - [Tables](#)(see page 144)
 - [data_load](#)(see page 144)
 - [Description](#)(see page 144)
 - [Columns](#)(see page 144)
 - [api_response](#)(see page 145)
 - [Description](#)(see page 145)
 - [Columns](#)(see page 146)
 - [xref_data_load_data_source](#)(see page 147)
 - [Description](#)(see page 147)
 - [Columns](#)(see page 147)
 - [data_source](#)(see page 147)
 - [Description](#)(see page 147)
 - [Columns](#)(see page 147)
 - [fs_run_metrics_diff](#)(see page 148)
 - [Description](#)(see page 148)
 - [Columns](#)(see page 148)
 - [fs_run_agg_metrics](#)(see page 149)
 - [Description](#)(see page 149)
 - [Columns](#)(see page 149)
 - [fs_run_descriptions](#)(see page 149)
 - [Description](#)(see page 149)
 - [Columns](#)(see page 149)
 - [tin_npi_svc_audit_log - Deprecated](#)(see page 150)
 - [Description](#)(see page 150)
 - [Columns](#)(see page 150)

1 Overview

The Universal Data Set is a curated data warehouse of all QPP data for analytics and reporting. This page is the full detailed documentation of the UDS data. Wherever possible, we try to illuminate business rules as we learn them. Data tables (and their documentation) are separated into domains that define related data sets and sources.

2 Concepts

2.1 Normalized and Denormalized Data

The UDS Postgres data warehouse provides two versions of QPP data. The first is a set of normalized, relational tables. Tables are grouped into six data domains, broadly representing one or more datasets for a component of the QPP program: Measures, Eligibility, Submissions, Submission Scoring, Final Scoring, and Web Interface. (A seventh domain, ETL, contains metadata about how we load the data domains.) Each table represents a distinct concept within its domain, or is a join table between two concepts. Almost every table has a unique, auto-incrementing, integer primary key identifier for every row in the table called `db_id`. Tables can be joined together with foreign keys that reference these primary identifiers with the format `db_<other table name>_id`. So for example, the `final_score` table has a foreign key `db_candidate_score_id` which joins to the `candidate_score` table's primary key `db_id` column. Sometimes it is also possible to join between tables using natural keys (keys that are present within the source dataset). The normalized tables reside in the `uds` schema; see the "Data Model" section below for a visual diagram of the domains, their tables, and how various tables are associated to each other. A second schema, `active`, contains Postgres views with the same names and schema / structure as the `uds` tables, but which represent a subset of the data (see Data Load / Snapshots below).

This page is primarily concerned with documenting these relational uds tables, their data sources, relationships, and business rules. The other version of QPP data that is provided to users is the denormalized data marts. Data marts are consolidated, denormalized or 'flattened' sets of QPP data that are used to power a dashboard, generate a report, or allow for easier querying of the data. Use of the data marts is highly recommended in cases where users don't want or need to concern themselves with determining which version of the data is the most up-to-date and validated, or don't want to join across uds tables themselves. Data marts are documented here: [UDS Data Mart Data Dictionary](#)¹.

2.2 Data Load / Snapshots

Almost all datasets that are loaded in a batch process (as opposed to a streaming process) are subject to our "data load", or snapshot, concept, and tied to a record in the data_load table. What this means is that whenever we reload a dataset or data domain, we never update existing rows for a given record. Instead, we always create a new record even if the record already exists in the database. For each dataset (identified by a combination of data_load.uds_domain and data_load.year), only one load of the dataset is marked 'active' at any time. "Active" data indicates that the data set has been validated, is the latest version of the data, and is appropriate for use in analytics and reporting. The benefits of this data load concept are:

- the ability to see how a dataset changes over time
- the ability to fully load a dataset before exposing it to users (no partial / halfway data changes)
- the ability to run data quality checks / validations before allowing the user access to the data

This document is organized with each table listed under its associated data_load.uds_domain value. The current domains are:

- Eligibility
- Measures
- Submissions
- Submission Scoring
- Final Scoring
- Final Scoring, Score Profiler
- Final Scoring, QIS

¹ <https://confluence.cms.gov/display/QPPAR/UDS+Data+Mart+Data+Dictionary>

- Final Scoring, Payment Adjustment
- Eligibility, LVT Opt In
- Eligibility, SSP

2.3 Data Quality

A&R performs a series of data quality checks nightly, and whenever data is loaded or reloaded. For more information, see [Data Quality Validation](#)².

2.4 No Fake / Test Data in Production

For the A&R team, the QPP data is our product. Other QPP components use our data warehouse to drive their processes; reports generated from our data are used to determine payment adjustments; and analytics from our data are used make policy decisions. As such, the A&R team mandates that no known fake or test data will be allowed into the UDS or its data marts in production. Source teams may generate fake data in production in order to test the functionality of their systems, but it must be easily and consistently identifiable, exclusive of real data, and filterable. A&R will filter out all known fake or test data when ingesting a dataset into our production data warehouse.

This Confluence document: [QPP Fake/Test Data in Production](#)³ represents an agreement between all QPP components as to the definition of fake data in production. (Some teams, ourselves included, also use these criteria to generate our fake data in lower tiers like develop, to ensure that lower tiers do not accidentally contain real PII data.) All QPP teams and the Data Governance stewards have a responsibility to ensure that the team upholds these definitions, or that new definitions are agreed upon across the QPP product teams.

2.5 Best Practices for Querying the UDS

Guidance on querying the UDS is located here: [Querying the UDS for Fun and Glory](#)⁴.

Known data invariants are documented here: [QPPAR Invariants](#)⁵

² <https://confluence.cms.gov/display/QPPAR/Data+Quality+Validation>

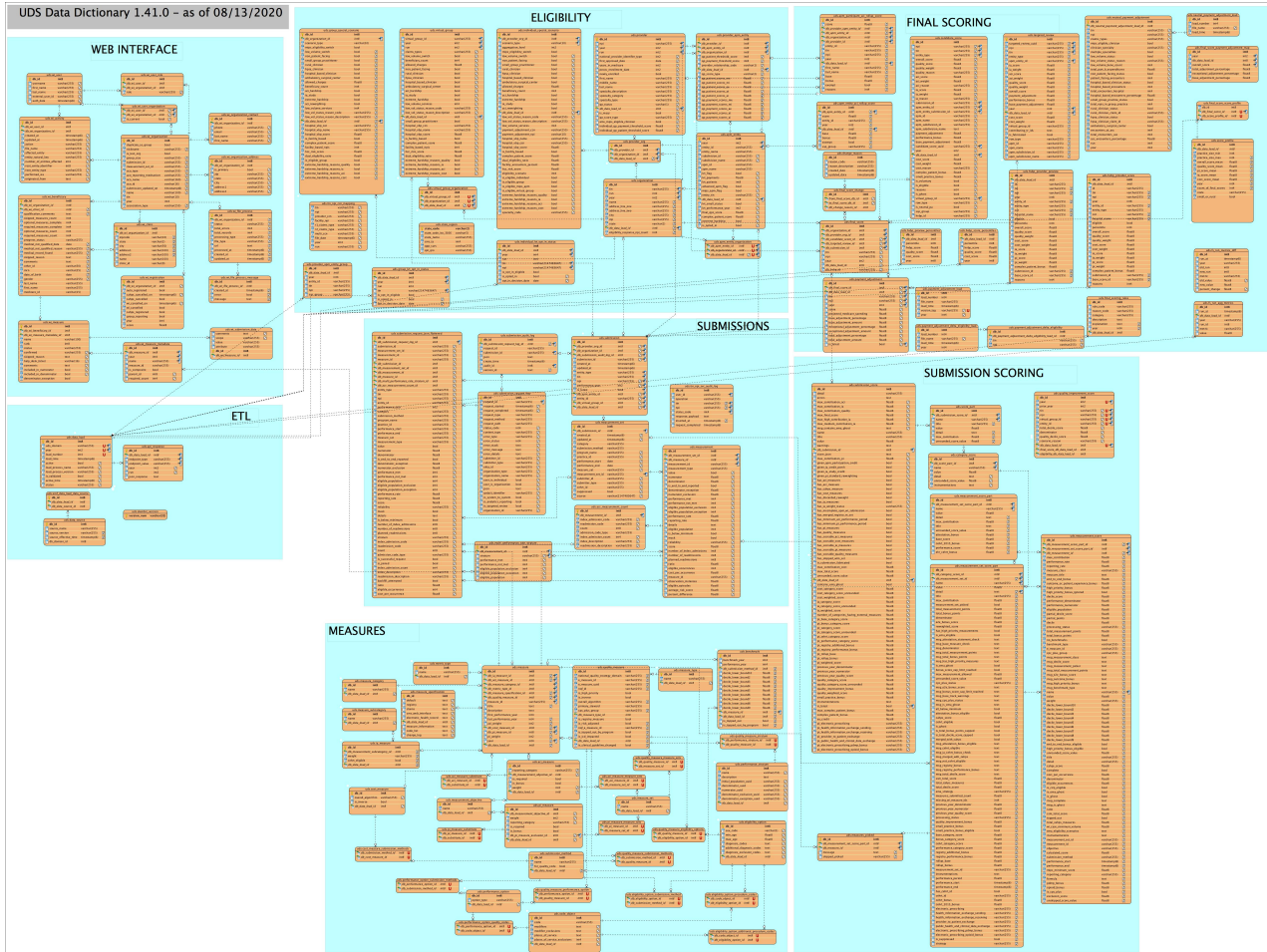
³ <https://confluence.cms.gov/pages/viewpage.action?pageId=108372205>

⁴ <https://confluence.cms.gov/display/QPPAR/Querying+the+UDS+for+Fun+and+Glory>

⁵ <https://confluence.cms.gov/display/QPPAR/QPPAR+Invariants>

3 Data Model Diagram

Following is an entity-relationship diagram of the UDS database schema.



4 Submissions

4.1 Data Sources

The Submissions domain comprises the measurements that clinicians have recorded during the performance year for activities related to improved patient quality of care, and how they transmitted this data to QPP. This data is recorded in a nested hierarchy:

- A *Submission* represents the set of data that applies to a single entity. For any given entity, there will be only one single submission per performance year. There are four types of entities: a individual (representing data for a single clinician; both TIN and NPI are specified); a group (representing data for an organization or collection of clinicians; TIN is specified and NPI will be empty); an APM entity (representing data for an Advanced Payment Model, which is a collection of many clinicians across multiple organizations; TIN and NPI will be empty and entity_id will be populated with entity_type = 'apm'); or a Virtual Group (starting in 2018; representing data for a collection of many clinicians across multiple organization who are not in an APM; TIN and NPI will be empty and entity_id will be populated with entity_type = 'virtual_group'). A submission represents, "who does this data apply to".
 - Within a Submission there are one or more *Measurement Sets*. These are collections of measurements that are grouped together according to: their category (currently four types: ia, aci, quality, cost), submission method (currently XXX types), and submitter details. This can be thought of as, "how was this data collected and transmitted".
 - Within a Measurement Set there are one or more *Measurements*. Measurements are the actual data recorded for specific measures. The data depends on the type of Measure being recorded. There are 9 types of measurements (see Measurement Type Field Mappings for more details).

No matter which pathway / submission method, all submissions data goes through the Submissions API. A REST request to create or modify submission, measurement set, or measurement data is made to the Submissions API; if it is valid, the data is transmitted successfully and stored in the Submission database (MySQL). So, the source of truth for Submissions data is **the Submissions database and API**.

The information of the requests to the API (request logs) are also stored in the UDS Submissions domain. We store both successful and failed requests, all metadata around who was making the request, and what data they sent as part of the submission. Logs are a lossy data source, as they can fail to be transmitted across multiple connections between the API and UDS. Because request logs can fail for many validation reasons, all Submissions data that are contained within the request logs cannot be expected to fully conform to our expected schemas.

4.1.1 Data Source documentation

This is the documentation of the Submissions API: <https://cmsgov.github.io/qpp-submissions-docs/>. Of particular interest for this domain are the Submissions, Measurement Sets, and Measurements links.

This is the documented schema of the Submissions database: <https://github.com/CMSgov/qpp-submissions-api/blob/master/tools/schema.sql>

Audit documentation ...?

Log documentation ...? [Data Models⁶](#)

⁶ https://confluence.cms.gov/display/QPPAR/Data+Models#DataModels-AV1.1:av1_1_subm_log

4.1.2 Data ingestion processes

We ingest Submissions data into the UDS via two different methods.

1. *Audits during the Submission Window.*
 - a. With every data create/change request that is made to the Submissions API and processed successfully, the Submissions API creates an audit log. This log is stored in Amazon s3 and contains the entire Submission JSON object, including the changes from the successful request. We take this JSON object and process it into the following tables: submission, measurement_set, measurement, multi_performance_rate_stratum, acr_measurement_count. The metadata about the audit log is stored in submission_audit_log.
 - b. Because of the nature of the constantly-updating Submissions data during the submission window (generally Jan-March after the performance year), the Submissions data during this time is considering a streaming dataset, not a data-loaded dataset. That means that the latest audit for any given submission is the latest validated data. During the submissions window, the Submissions domain for the performance year is not tied to a data_load row. Each row of `uds.submission` will have a non-null db_submission_audit_log_id and a null db_data_load_id.
 - c. Documentation of this pipeline, called UDS_submission_audit: TBD
2. *Submissions API Data Load*
 - a. At the end of the submissions window, we must do a final load of all submissions data into the UDS for the performance year. There are a couple of reasons for this. First: concurrent (near-simultaneous) requests to the Submissions API can result in the latest audit not containing the full and final state of the submission as it is stored in the submissions database. By fetching every submission from the Submission API in a single data load, we ensure that we have the complete, latest data for each submission. Second: sometimes an audit is not created by the Submissions API, leaving us with no record from the submissions window of the submission. We can retrieve the full list of submission ids from the Submissions database directly. *Warning:* Care needs to be taken here to ensure that we don't ingest fake submissions into the UDS.
 - b. During this method of obtaining Submissions data, we tie the submission to a data_load. At this point, the Submissions domain is therefore snapshotted and validated with data quality checks before being made available to end users. Each row of `uds.submission` will have a null db_submission_audit_log_id and a non-null db_data_load_id.
 - c. Documentation of this pipeline, called UDS_submission_api_score (legacy name) is here: [Submission Pipeline - via Data Load / Submissions API](#)⁷

The documentation of the Logs pipeline, called UDS_Logs, is here: [Logs Pipeline](#)⁸. Request logs are transferred into the UDS via the following chain of technologies:

- Submissions API log file → splunk forwarder → A&R syslog → A&R Kafka queue → Nifi / Python processing → UDS storage.

⁷ <https://confluence.cms.gov/pages/viewpage.action?pagelId=274541657>

⁸ <https://confluence.cms.gov/display/QPPAR/Logs+Pipeline>

4.2 Tables

4.2.1 submission

4.2.1.1 Description

This table, along with its other associated Submission domain tables, represents a complete snapshot of a submission after a request is successfully processed by Submission API to create or modify the submission in some way. The structure of this table and associated measurement_set, measurement, multi_performance_rate_stratum, and acr_measurement_count tables closely follows the JSON structure of the Submission API, which is documented [here](#)⁹. Following is an excerpt from that documentation.

"The Submissions resource represents one year of performance data for a given individual or group [sic; or apm or virtual group]. Submissions contain MeasurementSets which can be accessed both via Submissions methods [meaning API endpoints] and MeasurementSets methods [endpoints]. Submissions resources are 'shared' in the sense that they contain Measurement Sets that may be created by multiple users."

The UDS documentation includes a mapping of this JSON structure to the UDS tables. See [here](#)¹⁰.

Keys: The natural key for this table includes submission_id, performance_year, and updated_at. The surrogate key is db_id. The is_latest flag indicates the record representing the latest state of the submission that was successfully processed by the API. At the end of the submission window, records with is_latest equal to true are used to retrieve the complete category score JSON from the Submission API. Therefore, only these records have reliable associated category score information.

Each submission will be linked to one of the following: a provider_org, an organization, an apm_entity, or a virtual_group depending on whether the submission is for an individual, group, APM entity, or virtual group. If a submission is for an APM entity or virtual group (entity_id is not null or empty, entity_type = 'apm' or 'virtual_group'), that entity or virtual group must exist in the Eligibility domain. Submissions for groups or individuals that cannot be linked to any records in the provider_org or organization tables will still be saved. These are "voluntary" submitters who do not appear in the eligibility data (though this is not the full definition of voluntary). The Submission API is required by law to accept all Submissions, regardless of whether or not the submission's tin, np, apm_entity_id, or virtual_group_id are real and under a CMS program. In theory, any provider could submit data, and any third party system could submit fake data.

4.2.1.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_provider_org_id
 - type: int8
 - description: Foreign key to the provider_org table - provider_org.db_id.
- db_organization_id
 - type: int8
 - description: Foreign key to the organization table - organization.db_id.
- db_submission_audit_log_id

⁹ <https://cmsgov.github.io/qpp-submissions-docs/submissions>

¹⁰ https://docs.google.com/spreadsheets/d/1D9IX3NKZroi4Feqkr0ag_T6mSRQHFvicg6kBHBya-ts/edit#gid=0

- type: int8
- description: Foreign key to the submission_audit_log table - submission_audit_log.db_id. Non-null when the submission is processed via a streaming process. Null when the submission is processed via a data load process.
- submission_id
 - type: varchar(255)
 - description: The unique UUID of the submission. Created by the Submissions API when creating the submission. Although there is no not null constraint here, this field can never be legitimately null.
- created_at
 - type: timestamp not null
 - description: The creation time of the submission in RFC_3339 format.
- updated_at
 - type: timestamp not null
 - description: The modification time of the submission in RFC_3339 format.
- entity_type
 - type: varchar(255) not null
 - description: The type of entity associated with the submission. Acceptable values are "individual", "group", "apm", and "virtual_group" (for 2018 and later).
- tin
 - type: varchar(255) not null
 - description: The 9-digit identifier of the provider associated with the submission (Taxpayer Identification Number). Although this field has a not null constraint, it is allowed to be empty string when the entity_type is "apm" or "virtual_group".
 - PII field
- npi
 - type: varchar(255) default ""
 - description: The 10-digit identifier of the provider associated with the submission (National Provider Identifier).
- performance_year
 - type: int2
 - description: The year in which performance data for the submission was collected.
- is_latest
 - type: bool not null
 - description: Whether the particular row corresponds to the last/most recent version of the submission. This field is crucial in helping us to flip back and forth between submissions that are loaded via audits (streaming process) and those loaded via the API (batch process).
- db_apm_entity_id
 - type: int8 foreign key
 - description: Foreign key to the apm_entity table - apm_entity.db_id.
- entity_id
 - type: varchar(255)
 - description: Either the APM entity id or the Virtual Group id, depending on the submission's entity_type.
- db_virtual_group_id
 - type: int8 foreign key
 - description: Foreign key to the virtual_group table - virtual_group.db_id.
- db_data_load_id
 - type: int8 foreign key
 - description: Foreign key to the data_load table - data_load.db_id. Non-null when the submission is processed via a batch data load process. Null when the submission is processed via a streaming (audits) process.

4.2.2 measurement_set

4.2.2.1 Description

From the API documentation (see [here](#)¹¹):

"The MeasurementSets resource represents performance data for a specified category. Each Submission can have multiple MeasurementSets. Each MeasurementSet in a given Submission must be uniquely identified by category and submission method [sic; submitter_id and submitter_type must also be used to uniquely identify measurementSets]. MeasurementSets contain Measurements, which can be accessed both via MeasurementSets methods [endpoints] and Measurements methods [endpoints]."

The structure of this table and associated measurement, multi_performance_rate_stratum, and acr_measurement_count tables closely follows the JSON structure of the Submission API.

Keys: The surrogate key is db_id. The natural key includes measurement_set_id, which is generated by the source system and is unique within the scope of a submission, and updated_at. Within a submission, the combination of submission_method, category, submitter_id, and submitter_type are unique.

4.2.2.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_submission_id
 - type: int8 not null
 - description: Foreign key to the submission table - submission.db_id.
- measurement_set_id
 - type: varchar(255) not null
 - description: The UUID of the measurement set. Created by the Submissions API when the measurement set is first created.
- created_at
 - type: timestamp not null
 - description: The creation time of the submission in RFC_3339 format.
- updated_at
 - type: timestamp not null
 - description: The modification time of the submission in RFC_3339 format.
- category
 - type: varchar(255) not null
 - description: QPP scoring category to which the measures in this measurement set belong: Improvement Activities, Quality, Advancing Care Information / Promoting Interoperability, and Cost. Acceptable values for 2017 are: ia, quality, aci, cost. Acceptable values for 2018 and later are: ia, quality, pi, cost.
 - The ACI category was renamed to PI for performance year 2018.
- submission_method
 - type: varchar(255) not null
 - description: The method by which data is submitted. Acceptable values are cmsWebInterface, electronicHealthRecord, claims, registry, certifiedSurveyVendor, and administrativeClaims.

¹¹ <https://cms.gov.github.io/qpp-submissions-docs/measurement-sets>

- **program_name**
 - type: varchar(255)
 - description: The quality payment program under which the submission belongs. Acceptable values are "mips" or "cpcPlus". If not provided, program_name will be recorded as "mips".
- **practice_id**
 - type: varchar(255)
 - description: The id of the practice associated with the measurement set. Required if program_name is set to "cpcPlus". Must be omitted if program_name is not set to "cpcPlus".
- **performance_start**
 - type: date not null
 - description: A date in RFC 3339 format with only the date part (for instance, "2013-01-15"). The first date when the measurement data is applicable.
- **performance_end**
 - type: date not null
 - description: A date in RFC 3339 format with only the date part (for instance, "2013-01-15"). The last date when the measurement data is applicable.
- **measure_set**
 - type: varchar(255)
 - description: The specialty set which the measurement set is attesting. Acceptable values are "allergyImmunology", "anesthesiology", "cardiology", "dermatology", "diagnosticRadiology", "electrophysiologyCardiacSpecialist", "emergencyMedicine", "gastroenterology", "generalOncology", "generalPracticeFamilyMedicine", "generalSurgery", "hospitalists", "internalMedicine", "interventionalRadiology", "mentalBehavioralHealth", "neurology", "obstetricsGynecology", "ophthalmology", "orthopedicSurgery", "otolaryngology", "pathology", "pediatrics", "physicalMedicine", "plasticSurgery", "preventativeMedicine", "radiationOncology", "rheumatology", "thoracicSurgery", "urology" or "vascularSurgery".
 - This gets complicated. TBD More deets here.
- **submitter_id**
 - type: varchar(255)
 - description: An identifier for the user / system that created the measurement set.
- **submitter_type**
 - type: varchar(255)
 - description: The type of user / system that created the measurement set.
- **cehrt_id**
 - type: varchar(255)
 - description: Added for 2019 data, this is a unique identifier that is generated by ONC (?) and identifies a specific bundle of software or EHR (electronic health registry). The CEHRT is a 15 character alpha number. It will be included only on PI category measurement sets.
 - From [CEHRT ID Options](#)¹²: Generally, the pattern is:
 - XX14EXXXXXXXXXXX for 2014 Edition
 - XX15EXXXXXXXXXXX for 2015 Edition
 - XX15HXXXXXXXXXX for 2014/2015 hybrid
 - We only want to accept 2015 Edition CEHRT IDs.
- **suppressed**
 - type: boolean
 - description: A flag to indicate if a measurement set should be suppressed for scoring due to invalid data.
- **source**
 - type: varchar(255)
 - description: Indicates source, e.g. QRDA3

¹² <https://confluence.cms.gov/display/QPPARCH/CEHRT+ID+Options>

4.2.3 measurement

4.2.3.1 Description

From the API documentation (see [here](#)¹³):

"The Measurements resource represents performance data for a specific measure within a MeasurementSet. There are five types of Measurements: Boolean, Proportion, Non-Proportion, Single-Performance Rate, and Multi-Performance Rate. Each MeasurementSet can have multiple Measurements. No two Measurements in a given MeasurementSet can have the same measureId."

All types of measurements have been mapped to this single table. Each measurement type includes different fields. Therefore, the fields in this table will be populated differently depending on the type of measure the record represents. The measurement_type field indicates the type of measurement. Its valid values are as follows (each is linked to useful documentation from the API):

- [boolean](#)¹⁴
- [proportion](#)¹⁵
- [nonProportion](#)¹⁶
- [singlePerformanceRate](#)¹⁷
- registrySinglePerformanceRate
- [multiPerformanceRate](#)¹⁸
- registryMultiPerformanceRate
- [cahps](#)¹⁹
- [costScore](#)²⁰ (All-Cause Readmission (ACR))

For example, the "score" field is populated only for measurements of type "costSore" or "cahps". Refer to the **Measurement Type Field Mappings** below for details. For the multi-performance rate measurements, the strata values are contained in the associated multi_performance_rate_stratum table. For cost scores, counts for each admission code combination are contained in the acr_measurement_count table.

Keys: The surrogate key is db_id. The natural key includes measurement_id (generated by the source system) and updated_at.

4.2.3.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_measurement_set_id
 - type: int8 not null
 - description: Foreign key to the measurement_set table - measurement_set.db_id.
- db_measure_id

¹³ <https://cmsgov.github.io/qpp-submissions-docs/measurements>

¹⁴ <https://cmsgov.github.io/qpp-submissions-docs/measurements#boolean-measurements>

¹⁵ <https://cmsgov.github.io/qpp-submissions-docs/measurements#proportion-measurements>

¹⁶ <https://cmsgov.github.io/qpp-submissions-docs/measurements#non-proportion-measurements>

¹⁷ <https://cmsgov.github.io/qpp-submissions-docs/measurements#single-performance-rate-measurements>

¹⁸ <https://cmsgov.github.io/qpp-submissions-docs/measurements#multi-performance-rate-measurements>

¹⁹ <https://cmsgov.github.io/qpp-submissions-docs/measurements#cahps>

²⁰ <https://cmsgov.github.io/qpp-submissions-docs/measurements#all-cause-readmission-measurements>

- type: int8 not null
 - description: Foreign key to the measure table - measure.db_id.
- measurement_id
 - type: varchar(255) not null
 - description: The UUID of the measurement.
- measurement_type
 - type: varchar(255) not null
 - description: Represents the type of measurement. bmv = Boolean, pmv = Proportion, npmv = nonProportion, cmv = CAHPS, sprmv = Single-Performance Rate, mprmv = Multi-Performance Rate, mprmvS = Multi-Performance Rate Stratum, csmv = costScore.
- value
 - type: bool
 - description: True if attesting to the associated measure (null if measurement_value_type != "bmv").
- numerator
 - type: float8
 - description: The number of patients or episodes of care for which the measure criteria are satisfied. Must be greater than or equal to zero and less than or equal to the denominator.
- denominator
 - type: float8
 - description: The total number of patients or episodes of care as described by the measure. Must be greater than or equal to zero. Can only be 0 if the numerator is 0 as well.
- is_end_to_end_reported
 - type: bool
 - description: True if the measure was reported via certified EHR technology without any manual interference.
- denominator_exception
 - type: float8
 - description: The exceptions from the denominator field as described in the QCDR measure specification.
- numerator_exclusion
 - type: float8
 - description: The exclusions from the numerator field as described in the QCDR measures specification.
- performance_met
 - type: int4
 - description: The number of patients for which the measure criteria are satisfied. Must be greater than or equal to zero and less than or equal to the eligible_population.
- performance_not_met
 - type: int4
 - description: The number of patients for which the measure criteria are not satisfied.
- eligible_population
 - type: int4
 - description: The total number of eligible patients as described by the measure. Must be greater than or equal to zero.
- eligible_population_exclusion
 - type: int4
 - description: The number of patients who are excluded from the measure.
- eligible_population_exception
 - type: int4
 - description: The number of patients for which the measure criteria are not satisfied but who are excluded from the measure.
- performance_rate
 - type: float8

- description: Performance rates are stored and represented as percentages with values from 0.00 to 100.00. The performance rate for a multiple performance rate measurement is calculated using the overallAlgorithm of the corresponding measure. NOTE: Performance rate calculations exclude any "overall" strata, except where the overallAlgorithm is overallStratumOnly. Additionally, for the simpleAverage and overallStratumOnly algorithms, each stratum's performance rate is calculated as $\text{performance_met} / (\text{performance_met} + \text{performance_not_met} - \text{exclusions})$. Currently, a measure's overallAlgorithm may be one of the following: simpleAverage: $(\text{sum strata's performance rates}) / (\text{number of strata}) * 100$, weightedAverage: $(\text{sum strata's performance_met}) / (\text{sum strata's performance_met and performance_not_met}) * 100$, sumNumerators: sum strata's performance_met, or overallStratumOnly: performance rate of the "overall" stratum.
- reporting_rate
 - type: float8
 - description: The reporting rate, ranging from zero to one-hundred and representing a percentage, is equal to $(\text{performanceMet} + \text{eligiblePopulationExclusion} + \text{eligiblePopulationException} + \text{performanceNotMet}) / \text{eligiblePopulation} * 100$.
- score
 - type: float8
 - description: The score of the measurement.
 - For CAHPS measurements 1 - 12: score for survey question. For CAHPS measurement 321: average of scores for CAHPS measurements 1 - 12 <https://confluence.cms.gov/display/QPPRACTICE/CAHPS+Scoring>
 - when MSPB measure: MSPB spending for the practice based, includes risk adjustment (<https://confluence.cms.gov/pages/viewpage.action?spaceKey=QPPCP&title=Output+Data+Elements>)
 - when TPCC measure: Average cost per beneficiary for the practice, includes risk adjustment and specialty adjustment. (<https://confluence.cms.gov/pages/viewpage.action?spaceKey=QPPCP&title=Output+Data+Elements>)
 - when other cost measure: The dollar value associated with the measure output to be scored against the benchmark (EG Delivery & Patient Reports²¹)
- reliability
 - type: varchar(255)
 - description: <https://confluence.cms.gov/display/QPPRACTICE/CAHPS+Scoring>
- mask
 - type: bool
 - description: <https://confluence.cms.gov/display/QPPRACTICE/CAHPS+Scoring>
- details
 - type: text
 - description: Details attribute for 'costScore' measurements.
- is_below_minimum
 - type: bool
 - description: isBelowMinimum attribute for CAHPS measurements.
- number_of_index_admissions
 - type: int8
 - description: The number of eligible admissions to an acute care hospital assessed in the measure for the outcome (readmitted or not within 30 days).
- number_of_readmissions
 - type: int8
 - description: The number of admissions to an acute care hospital within 30 days of discharge from an acute care hospital. A readmission may in turn serve as an index admission.
- planned_readmissions
 - type: int8

²¹ <https://confluence.cms.gov/pages/viewpage.action?pagelD=233542007>

- description: The number of intentional readmissions within 30 days of discharge from an acute care hospital that is a scheduled part of the patient’s plan of care. Planned readmissions are not counted as outcomes in this measure.
- measure_id
 - type: varchar(255)
 - description: The identifier of the measure to which this data applies.
- ratio
 - type: float8
 - description: <https://confluence.cms.gov/pages/viewpage.action?spaceKey=QPPCP&title=Output+Data+Elements>
 - when MSPB measure: Average ratio of per episode spend and the risk-adjusted expected cost for the practice. A lower ratio indicates better performance.
 - when TPCC measure: Ratio of the practice's total per capita cost vs the national average total per capital cost, specialty adjusted. A lower ratio indicates better performance.
- eligible_occurrences
 - type: int4
 - description: <https://confluence.cms.gov/pages/viewpage.action?spaceKey=QPPCP&title=Output+Data+Elements>
 - when MSPB measure: Count of eligible episodes within the practice.
 - when TPCC measure: Count of eligible beneficiaries within practice.
- cost_per_occurrence
 - type: float8
 - description: <https://confluence.cms.gov/pages/viewpage.action?spaceKey=QPPCP&title=Output+Data+Elements>
 - when MSPB measure: Average spend per episode for the practice. Does not account for patient risk profiles.
 - when TPCC measure: Average cost per beneficiary for the practice. Does not account for patient risk profile factor or practitioner specialty.
- observation_instances
 - type: float8
 - description: The number of denominator eligible instances that are used as input in the calculation to derive the numerator (i.e. average, ratio).
- eligible_episodes
 - type: float8
 - description: For cost measures. The count of episodes associated with the score output. Added in the 2019 performance year.
- average_risk_score
 - type: float8
 - description: For cost measures. The average risk score of patients to determine the TIN or TIN | NPIs Risk Bracket. Added in the 2019 performance year.
- percent_difference
 - type: float8
 - description: For cost measures. Percent difference between TIN or TIN | NPIs Average Risk-Adjusted Episode Cost and National Average Risk-Adjusted Episode Cost. Added in the 2019 performance year.

4.2.3.3 Measurement Type Field Mappings

Different types of measurements have different fields in the `uds.measurement` table that apply to them.

Field Name	boolean	proportion	nonProportion	singlePerformanceRate	multiPerformanceRate	registryMultiPerformanceRate	registrySinglePerformanceRate	caps	costScore
value	X								
numerator		X	X						
denominator		X	X						
is_end_to_end_reported			X	X	X	X			
denominator_exception			X						
numerator_exclusion			X						
performance_met				X	in_multi_performance_rate_stratum	in_multi_performance_rate_stratum	X		
performance_not_met				X	in_multi_performance_rate_stratum	in_multi_performance_rate_stratum	X		
eligible_population				X	in_multi_performance_rate_stratum	in_multi_performance_rate_stratum	X		
eligible_population_exclusion				X	in_multi_performance_rate_stratum	in_multi_performance_rate_stratum	X		

Field Name	boolean	proportion	nonProportion	singlePerformanceRate	multiPerformanceRate	registryMultiPerformanceRate	registrySinglePerformanceRate	caps	costScore
eligible_population_exception				X	in_multi_performance_rate_stratum	in_multi_performance_rate_stratum	X		
performance_rate				X	X	X	X		
reporting_rate			X	X	X	X	X		
score								X	X
reliability								X	
mask								X	
details									X
is_below_minimum								X	
number_of_index_admissions									X
number_of_readmissions									X
planned_readmissions									X
ratio									X
eligible_occurrences									X
cost_per_occurrence									X

Field Name	boolean	proportion	nonProportion	singlePerformanceRate	multiPerformanceRate	registryMultiPerformanceRate	registrySinglePerformanceRate	caps	costScore
observation_instances			X						
eligible_episodes									X
average_risk_score									X
percent_difference									X

4.2.4 multi_performance_rate_stratum

4.2.4.1 Description

From the API documentation (see [here](#)²²):

"Multi-Performance Rate Measurements are applicable to Quality measures. There are two types of Multi-Performance Rate Measurements: registry and normal. The difference between the two is that for Registry Multi-Performance Rate Measurements, the performanceRate field is both writable and required. Multi-Performance Rate Measurements contain multiple strata and the stratum field is required for each."

"A Multi-Performance Rate Stratum represents the performance data for a specified subset of the population, as described by the stratum field."

Multi-performance rate measurements will have one or more associated multi_performance_rate_stratum records.

Keys: The surrogate key is db_id. The candidate key includes db_measurement_id and stratum.

4.2.4.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_measurement_id
 - type: int8 not null
 - description: Foreign key to the measurement table - measurement.db_id.
- stratum
 - type: varchar(255)
 - description: The stratum associated with the performance rate measurement.
- performance_met
 - type: int4

²² <https://cmsgov.github.io/qpp-submissions-docs/measurements#multi-performance-rate-measurements>

- description: The number of patients for which the measure criteria are satisfied. Must be greater than or equal to zero and less than or equal to the eligible_population.
- performance_not_met
 - type: int4
 - description: The number of patients for which the measure criteria are not satisfied.
- eligible_population
 - type: int4
 - description: The total number of eligible patients as described by the measure. Must be greater than or equal to zero.
- eligible_population_exclusion
 - type: int4
 - description: The number of patients who are excluded from the measure.
- eligible_population_exception
 - type: int4
 - description: The number of patients for which the measure criteria are not satisfied but who are excluded from the measure.

4.2.5 acr_measurement_count

4.2.5.1 Description

From the API documentation (see [here](#)²³):

"An All Cause Readmission Measurement represents beneficiaries aged 65 and older that were hospitalized at a short-stay acute-care hospital and were readmitted to an acute-care hospital for any reason within 90 days of being discharged from the original hospital. Third party integrators cannot submit All Cause Readmission measurements."

This table represents (re)admission counts by index code, readmission code, and index-readmission code pair. The type of count is indicated by the admission_code_type filed:

- indexAdmissionCountByDiagnosis: One record per unique diagnosis code.
- readmissionCountByDiagnosis: One record per unique diagnosis code.
- indexReadmissionDiagnosisPairCounts: One record per unique pair of index diagnosis code and readmission diagnosis code.

Keys: The surrogate key is db_id. The candidate key includes all other fields.

4.2.5.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_measurement_id
 - type: int8 not null
 - description: Foreign key to the measurement table - measurement.db_id.
- index_admission_code
 - type: varchar(255)
 - description: The code associated to the reason for admittance of a patient to an acute care hospital.

²³ <https://cms.gov.github.io/qpp-submissions-docs/measurements#all-cause-readmission-measurements>

- readmission_code
 - type: varchar(255)
 - description: Code associated to procedure or reasoning for patient readmittance.
- count
 - type: int8
 - description: The number of occurrences for an index admission/readmission combination within a measurement.
- admission_code_type
 - type: varchar(255) not null
 - description: Determines whether: indexReadmissionDiagnosisPairCounts, indexAdmissionCountByDiagnosis, or readmissionCountByDiagnosis
- index_admission_count
 - type: int4
 - description:
- index_description
 - type: varchar(255)
 - description:
- readmission_description
 - type: varchar(255)
 - description:

4.2.6 submission_audit_log

4.2.6.1 Description

Represents audit logs that are produced by the source system, the Submission API. Each time the Submission API successfully processes a request, it writes a submission JSON object to its audit log. This "submission object" represents a snapshot of the entire state of the submission object after the API request has been processed. The "json" field of this table captures a JSON representation of the audit log that is created by the source system - the Submission API. This audit log JSON includes 1) the submission JSON object; 2) the JSON representation of eligibility information for the provider, organization, or APM entity that made the submission; 3) certain attributes related to the request that resulted in the audit; 4) the user JSON object of the user making the request/audit; and 5) the JSON output of the scoring engine, which represents category scores for the submission. The data from the JSON submission object is used to populate associated tables in the Submission domain, including submission, measurement_set, measurement, multi_performance_rate_stratum, and acr_measurement_count. The JSON score object is used to populate associated tables in the Scoring domain, including submission_score, score_part, category_score, measurement_set_score_part, measurement_score_part, measurement_score, and measures_picked. (No longer the case starting in the 2019 performance year.)

NOTE: The scoring engine JSON included in this table will be incomplete because the Submission API uses credentials which do not have access to all the data. (No longer the case starting in the 2018 performance year.) The complete scoring engine JSON is included in the score_json field of the submission_score table.

Keys: The natural key for these API requests is request_id and create_time. Each audit also has a key to the submission, submission_id. The source system uses audit_id as its internal surrogate key. A&R uses db_id as its surrogate key and primary key.

Note regarding the active.submission_audit_log view: Audits will only populate in the active view, based on the is_latest=true submission row, for the current performance year, before the submission window closes. Once the submission window closes, the Submission domain switches from populating via the streaming audits to being populated via the batch data load from the submissions API. This data load creates `uds.submission` rows with a

non-null `db_data_load_id`, and a null `db_submission_audit_log_id`, because the full submission is no longer tied to a single point-in-time audit. So the `active.submission` table for those years won't join to `submission_audit_log`.

4.2.6.2 Columns

- `db_id`
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- `db_submission_request_log_id`
 - type: int8 foreign key
 - description: Foreign key to the `submission_request_log` table - `submission_request_log.db_id`. The natural keys that result in this join are: `request_id`.
 - notes: Although every audit must correspond to an API request and a single `request_id`, this foreign key allows nulls. Because API request logs are not a robust data source, we sometimes receive an audit record, but never receive the corresponding request log record. In this situation, the audit is stored in this table, but there will never be a record in `submission_request_log` to link to.
- `audit_id`
 - type: int8 not null
 - description: The id of the audit. This is created by the source system. For performance year 2017, this corresponds to the `audit_logs.id` column in the Submissions database. For later performance years, this is generated by our audit processing pipeline.
- `request_id`
 - type: varchar(255) not null
 - description: Unique identifier for the API request. The format is a UUID string, automatically generated by the Submissions API.
- `submission_id`
 - type: varchar(255) not null
 - description: The unique identifier of the submission. The format is a UUID string, automatically generated by the Submissions API.
- `json`
 - type: text not null
 - description: The full audit data in JSON format. This data is parsed into the `submission` and related tables.
 - PII field (due to TINs nested within the JSON)
- `create_time`
 - type: timestamp not null
 - description: The timestamp of when the audit was created. Created by UDS upon insert to `submission_audit_log` table.
- `version_id`
 - type: text
 - description: Unique audit identifier. Created by Nifi?

4.2.7 submission_request_log

4.2.7.1 Description

This table represents logs that are created by the Submission API when processing an HTTP POST/PUT/PATCH/DELETE request. The table includes details about the request that are not part of the submission itself, such as information about the submitter and the status of the request. During the submission window, this information is used to understand the volume of requests to the API and how the API is performing, as well as the nature of any

errors that might be occurring. Any API request that results in an error (status_code >= 400) will not result in an submission_audit_log record. The API emits multiple logs per HTTP request, designated by a unique request_id identifier. A record in this table merges those logs to capture the start and the end time of the request.

Keys: The surrogate key is db_id. The natural key is request_id.

4.2.7.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- request_id
 - type: varchar(255) not null
 - description: Unique identifier for the request.
- request_started
 - type: timestamp
 - description: The timestamp for the msg = "Started" request.
- request_completed
 - type: timestamp
 - description: The timestamp for the msg = "Completed" request.
- request_type
 - type: varchar(255) not null
 - description: Indicates the type of resource being created or updated. These types correspond to the Submission API resource types. Extracted from the Splunk Log path.
- request_method
 - type: varchar(255) not null
 - description: The HTTP verb.
- request_path
 - type: varchar(255) not null
 - description: The full HTTP request path.
- status_code
 - type: int4
 - description: The HTTP status code.
- content_type
 - type: varchar(255)
 - description: The type of content sent to the API (e.g., "application/json", "application/xml", etc.)
- error_type
 - type: varchar(255)
 - description: Any valid error type thrown by the API.
- error_status
 - type: int4
 - description: The HTTP status code.
- error_stack
 - type: text
 - description: The API stack trace.
- error_message
 - type: text
 - description: The API error message.
- error_details
 - type: text
 - description: JSON array of details.
- submitter_id

- type: varchar(255)
 - description: Uniquely identifies the entity making the request.
- submitter_type
 - type: varchar(255)
 - description: The entity type of the submitter.
- okta_id
 - type: varchar(255)
 - description: Identity management ID specifying the user making the request.
- organization_type
 - type: varchar(255)
 - description: The submitter organization type - e.g. registry.
- organization_name
 - type: varchar(255)
 - description: The submitter organization name - e.g. Falcon Registry.
- user_is_individual
 - type: bool
 - description: Submitted by an individual - true/false.
- user_is_organization
 - type: bool
 - description: Submitted by an individual associated with an organization, or an organization directly integrated via API - true/false.
- qrdaiii_identifier
 - type: varchar(255)
 - description: The UUID from the QRDA3 → JSON converter.
- json
 - type: text
 - description: The log in json format.
- is_system_to_system
 - type: bool
 - description: Indicator if the request is coming from a QPP component system.
- is_analytics_reporting
 - type: bool
 - description: Indicator if the user making the request is QPP A&R (hey, that's us!)
- is_targeted_review
 - type: bool
 - description: Indicator if the user making the request is permitted to create / modify targeted reviews.
- organization_id
 - type: varchar(255)
 - description:

4.2.8 submission_request_json_flattened

4.2.8.1 Description

This table provides a "flattened" representation of the request JSON body that is parsed out of the "json" field of a submission_request_log record. The JSON object, which is hierarchically structured (nested), has been "flattened" into a denormalized relational structure. That means there will be multiple records in this table per submission_id to capture values of the nested structures included in measurement_set, measurement, multi_performance_rate_stratum, and acr_measurement_count.

This data comes from request logs, any of which could fail due to data validation reasons; this data should not be trusted for most use cases.

Keys: The surrogate key is db_id.

4.2.8.2 Columns

Note: For most column descriptions, refer to the Submissions domain tables: submission, measurement_set, etc.

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_submission_request_log_id
 - type: int8 not null
 - description: Foreign key to the submission_request_log table - submission_request_log.db_id
- submission_id
 - type: varchar(255)
- measurement_set_id
 - type: varchar(255)
- measurement_id
 - type: varchar(255)
- measure_id
 - type: varchar(255)
- db_submission_id
 - type: int8
 - description: If the request is successful and we are able to find the submission_id, this links to the submission table row that contains the audit data for the request.
- db_measurement_set_id
 - type: int8
 - description: If the request is successful and we are able to find the measurement_set_id, this links to the measurement_set table row that contains the audit data for the request.
- db_measurement_id
 - type: int8
 - description: If the request is successful and we are able to find the measurement_id, this links to the measurement table row that contains the audit data for the request.
- db_measure_id
 - type: int8
 - description: If the request is successful and we are able to find the measure_id, this links to the active measure table row that contains the measure metadata.
- db_multi_performance_rate_stratum_id
 - type: int8
 - description: If the request is successful and we are able to find the measurement_id and it's a multi performance rate measure, this links to the multi_performance_rate_stratum table row that contains the audit data for the request.
- db_acr_measurement_count_id
 - type: int8
 - description: If the request is successful and we are able to find the measurement_id and it's an acr measure, this links to the acr_measurement_count table row that contains the audit data for the request.
- entity_type
 - type: varchar(255)
- tin

- type: varchar(255)
- npi
 - type: varchar(255)
- entity_id
 - type: varchar(255)
- performance_year
 - type: int2
- category
 - type: varchar(255)
- submission_method
 - type: varchar(255)
- program_name
 - type: varchar(255)
- practice_id
 - type: varchar(255)
- performance_start
 - type: varchar(255)
- performance_end
 - type: varchar(255)
- measure_set
 - type: varchar(255)
- measurement_type
 - type: varchar(255)
- value
 - type: bool
- numerator
 - type: float8
- denominator
 - type: float8
- is_end_to_end_reported
 - type: bool
- denominator_exception
 - type: float8
- numerator_exclusion
 - type: float8
- performance_met
 - type: int4
- performance_not_met
 - type: int4
- eligible_population
 - type: int4
- eligible_population_exclusion
 - type: int4
- eligible_population_exception
 - type: int4
- performance_rate
 - type: float8
- reporting_rate
 - type: float8
- score
 - type: float8
- reliability
 - type: varchar(255)

- mask
 - type: bool
- details
 - type: text
- is_below_minimum
 - type: bool
- number_of_index_admissions
 - type: int8
- number_of_readmissions
 - type: int8
- planned_readmissions
 - type: int8
- stratum
 - type: varchar(255)
- index_admission_code
 - type: varchar(255)
- readmission_code
 - type: varchar(255)
- count
 - type: int8
- admission_code_type
 - type: varchar(255)
- is_joined
 - type: bool
 - description: A boolean flag indicating that this record has been joined to either the audit of this request or the previously-successful audit of the same data.
- index_admission_count
 - type: int4
- index_description
 - type: varchar(255)
- readmission_description
 - type: varchar(255)
- backfill_attempted
 - type: bool
- is_successful_request
 - type: bool
 - description: True if submission_request_log.status_code < 400 (you're welcome Paul)
- ratio
 - type: float8
- eligible_occurrences
 - type: int4
- cost_per_occurrence
 - type: float8

5 Submission Scoring

5.1 Data Sources

The data in these tables represents the output of the scoring engine for a given submission that is received through the Submission API. This output is represented in a JSON format. The data from that JSON structure is loaded into the submission_score table and the associated tables: score_part, category_score, measurement_set_score_part, measurement_score_part, measurement_score, and measures_picked. Each of these tables represents a nested layer of the JSON.

NOTE: The latest version of each submission record should correspond to a submission_score record, and generally, the records in tables that represent parts of a submission (e.g. measurement) should have corresponding records in tables that represents parts of the score (e.g. measurement_score). However, in many cases, this is not true for reasons that are described in the "Submissions-Scoring Discrepancies" section below.

5.1.1 Data Source documentation

Documentation of JSON representation of that data USED TO BE [here](#)²⁴. Another option for documentation (which is not complete / comprehensive either) is here: <https://github.com/CMSgov/qpp-scoring-engines/blob/develop/swagger.yml>.

5.1.2 Data ingestion processes

The Submission Scoring domain is populated from scoring engine JSONs generated during Final Scoring processes. The pipeline that ingests this data is called UDS_Scoring and documented here: [Final Scoring pipeline](#)²⁵.

5.2 Submission Scoring Discrepancies

The majority of submissions should receive category scores from the scoring engine. However, there are situations in which a submission (or the measures it contains) will not be scored. Therefore, there will be cases in which, for example, a record in the submission table does not have a corresponding record in the submission_score table; a record in the measurement table does not have a corresponding record in the measurement_score table. This section describes the causes of these discrepancies.

The reasons for these differences will vary over time and are dependent on how the QPP system's IT infrastructure and rules evolve. The latest information is included in individual investigations into the data. Following is a list of those investigations.

[QPPAR-2877](#)²⁶ - UDS Research - rows in the MEASUREMENTS table that do not have matching row in MEASUREMENT_SCORES CLOSED

Following is a list over general issues that lead to such discrepancies.

2017 cost measurements: The 2017 cost measurement scores are not loaded into the UDS. Therefore, there will be no corresponding measurement_score records for cost measurement records.

²⁴ <https://cmsgov.github.io/qpp-submissions-docs/scoring>

²⁵ <https://confluence.cms.gov/display/QPPAR/Final+Scoring+pipeline>

²⁶ <https://jira.cms.gov/browse/QPPAR-2877?src=confmacro>

No eligibility data: If the Eligibility API does not provide eligibility data for the provider, the submission will not be scored.

TBD more here

5.3 Tables

5.3.1 submission_score

5.3.1.1 Description

The submission_score table represents the top-level of the score JSON structure. From the API documentation (see [here](#)²⁷):

"The Score Resource represents the top level scoring data for a submission object. Each Score will have multiple Score Parts and may have multiple Score Warnings or Score Errors. The Score Detail field lists the current version of the Scoring Engine used to score the corresponding submission."

Keys: The surrogate key is db_id.

5.3.1.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_submission_id
 - type: int8 not null
 - description: Foreign key to the submission table - submission.db_id.
- name
 - type: varchar(255) not null
 - description: Name of score object. Always "total".
- title
 - type: varchar(255) not null
 - description: Title of score object. Always "Total Score".
- detail
 - type: varchar(255) not null
 - description: Semantic version of scoring engine.
- value
 - type: float8 not null
 - description: Total score of submission.
- warnings
 - type: text not null
 - description: Array of warnings (scoring issues that do not halt the scoring process).
- errors
 - type: text not null
 - description: Array of errors (scoring issues that halt the scoring process). When the scoring engine runs alone, it can populate this errors object due to not being able to find Eligibility data for the

²⁷ <https://cmsgov.github.io/qpp-submissions-docs/scoring#score-resource>

submission's entity. But when Final Scoring runs, it will supply a dummy Eligibility object, and so no errors are expected from that process.

- max_final_score
 - type: float8
 - description: Maximum possible final score. (Deprecated?)
- max_medium_contribution_ia
 - type: float8
 - description: Medium score contribution of IA measurements.
- max_high_contribution_ia
 - type: float8
 - description: High score contribution for IA measurements.
- max_contribution_ia
 - type: float8
 - description: Maximum score the IA category can contribute to the final score.
- max_contribution_aci
 - type: float8
 - description: Maximum score the ACI category can contribute to the final score.
- max_contribution_pi
 - type: float8
 - description: Maximum score the PI category can contribute to the final score.
- max_contribution_quality
 - type: float8
 - description: Maximum score the Quality category can contribute to the final score.
- max_contribution_cost
 - type: float8
 - description: Maximum score the Cost category can contribute to the final score.
- msg_contains_ema_ghost
 - type: text
 - description: EMA Ghost Submission messaging.
- score_json
 - type: text
 - description: The entire score engine JSON output.
- given_apm_participation_credit
 - type: bool
 - description:
- given_ia_credit_pcmh
 - type: bool
 - description:
- given_ia_study_credit
 - type: bool
 - description:
- given_pi_standard_reweighting
 - type: bool
 - description:
- has_aci_measures
 - type: bool
 - description:
- has_acr_measure
 - type: bool
 - description:
- has_cahps_measure
 - type: bool
 - description:

- has_cost_measures
 - type: bool
 - description:
- has_discarded_reweight
 - type: bool
 - description:
- has_ia_measures
 - type: bool
 - description:
- has_ia_weight_status
 - type: varchar(255)
 - description: Possible values are 'special' and 'normal'. If clinician's submission is eligible to receive 2X point then, this column will be set to 'special' otherwise, the column will be set to 'normal'. The special status that are allowed to receive this special pointing are small practitioners.
- has_incomplete_apm_wi_submission
 - type: bool
 - description:
- has_merged_registry_m_set
 - type: bool
 - description:
- has_minimum_aci_performance_period
 - type: bool
 - description:
- has_minimum_pi_performance_period
 - type: bool
 - description:
- has_pi_measures
 - type: bool
 - description:
- has_quality_measures
 - type: bool
 - description:
- has_scorable_aci_measures
 - type: bool
 - description:
- has_scorable_cost_measures
 - type: bool
 - description:
- has_scorable_ia_measures
 - type: bool
 - description:
- has_scorable_pi_measures
 - type: bool
 - description:
- has_scorable_quality_measures
 - type: bool
 - description:
- has_skipped_wim_set
 - type: bool
 - description:
- is_submission_fabricated
 - type: bool
 - description:

- max_total_score
 - type: float8
 - description:
- unrounded_score_value
 - type: float8
 - description:
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id. Non-null when the submission score is processed via a batch data load process.
- contains_ema_ghost
 - type: bool
 - description:
- cost_category_score
 - type: float8
 - description:
- cost_category_score_unrounded
 - type: float8
 - description:
- cost_weighted_score
 - type: float8
 - description:
- ia_category_score
 - type: float8
 - description:
- ia_category_score_unrounded
 - type: float8
 - description:
- ia_weighted_score
 - type: float8
 - description:
- number_of_categories_having_external_measures
 - type: float8
 - description:
- pi_base_category_score
 - type: float8
 - description:
- pi_bonus_category_score
 - type: float8
 - description:
- pi_category_score
 - type: float8
 - description:
- pi_category_score_unrounded
 - type: float8
 - description:
- pi_cehrt_category_score
 - type: float8
 - description:
- pi_performance_category_score
 - type: float8
 - description:
- pi_registry_additional_bonus

- type: float8
 - description:
- pi_registry_performance_bonus
 - type: float8
 - description:
- pi_rollup_base
 - type: float8
 - description:
- pi_rollup_bonus
 - type: float8
 - description:
- pi_weighted_score
 - type: float8
 - description:
- previous_year_denominator
 - type: float8
 - description:
- previous_year_numerator
 - type: float8
 - description:
- previous_year_quality_score
 - type: float8
 - description:
- quality_category_score
 - type: float8
 - description:
- quality_category_score_unrounded
 - type: float8
 - description:
- quality_improvement_bonus
 - type: float8
 - description:
- quality_weighted_score
 - type: float8
 - description:
- small_practice_bonus
 - type: float8
 - description:
- instrumentations
 - type: text
 - description:
- is_latest
 - type: bool
 - description: Just as we specify the latest row of the submission, we specify the latest submission score associated to that submission (because we may load Submission Scoring multiple times for the same latest Submission data load).
- max_complex_patient_bonus
 - type: float8
- complex_patient_bonus
 - type: float8
- eu_credit
 - type: float8
- pi_electronic_prescribing

- type: varchar(255)
- pi_health_information_exchange_sending
 - type: varchar(255)
- pi_health_information_exchange_receiving
 - type: varchar(255)
- pi_provider_to_patient_exchange
 - type: varchar(255)
- pi_public_health_and_clinical_data_exchange
 - type: varchar(255)
- pi_electronic_prescribing_pdmp_bonus
 - type: varchar(255)
- pi_electronic_prescribing_opioid_bonus
 - type: varchar(255)

5.3.2 score_part

5.3.2.1 Description

From the API documentation (see [here](#)²⁸):

"The Score Part resource represents the categorical scoring data for a submission object. Categorical scoring refers to one of the three QPP Performance Categories, which are IA, ACI or Quality."

The value field indicates the **weighted** score for the performance category.

Keys: The surrogate key is db_id.

5.3.2.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_submission_score_id
 - type: int8 not null
 - description: Foreign key to the score table - score.db_id.
- name
 - type: varchar(255) not null
 - description: Performance category identifier, e.g. "ia", "aci", "pi", "quality", "cost".
- title
 - type: varchar(255) not null
 - description: Title of the score part object, e.g. "ACI component of final score".
- value
 - type: float8 not null
 - description: WEIGHTED score at the category level (not for the individual measure). category_score has the UNWEIGHTED value.
- detail
 - type: text not null
 - description: Performance category score weight.
- max_contribution

²⁸ <https://cms.gov.github.io/qpp-submissions-docs/scoring#score-part-resource>

- type: float8
- description: Maximum score the Score Part can contribute to the final score.
- unrounded_score_value
 - type: float8
 - description: The unrounded value of the score part.

5.3.3 category_score

5.3.3.1 Description

From the API documentation (see [here](#)²⁹):

"Each of the three individual category scoring engines has a Category Score resource. It represents the scoring output contained in a submission."

The value field indicates the **unweighted** score for the performance category.

Keys: The surrogate key is db_id.

5.3.3.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_score_part_id
 - type: int8 not null
 - description: Foreign key to the score_part table - score_part.db_id.
- name
 - type: varchar(255) not null
 - description: Performance category identifier, e.g. "ia", "aci", "pi", "cost", "quality".
- value
 - type: int4 not null
 - description: Performance category score. This is the UNWEIGHTED score (score_part is the WEIGHTED score)
- detail
 - type: text
 - description: Highest scoring measurement set by submission method. Also shows whether the measurement set was picked.
- unrounded_score_value
 - type: float8
 - description: The performance category score with no rounding.
- instrumentations
 - type: text
 - description: An array of business codes / logic?

²⁹ <https://cmsgov.github.io/qpp-submissions-docs/scoring#category-score-resource>

5.3.4 measurement_set_score_part

5.3.4.1 Description

From the API documentation (see [here](#)³⁰):

"The Scored Measurement Set represents the scoring output for each category measurement set in a submission."

Records in this table represent the score for the measurement set itself. Therefore, ideally, each measurement_set_score_part record will map to exactly one measurement_set record. Refer to the [API JSON to UDS Mapping](#)³¹ for details regarding which fields of the scoring JSON object are mapped to this table.

This table has two "child" tables:

- measurement_score: Represents individual measurement score values for all performance categories.
- measurement_score_part: The ACI measurement scores are grouped under aci_base, aci_bonus, aci_performance, and cerht_bonus. This table represents that grouping. Therefore, measurement_score is also a child of measurement_score_part.

This table structure permits querying for all measurement score values from a single table - measurement_score - while also allowing ACI measurement score values to be grouped.

Keys: The surrogate key is db_id.

5.3.4.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_category_score_id
 - type: int8 not null
 - description: Foreign key to the category_score table - category_score.db_id.
- db_measurement_set_id
 - type: int8
 - description: Foreign key to the measurement_set table - measurement_set.db_id. This is not always populated, if we can't link to the original measurement set due to the input scoring data.
- name
 - type: varchar(255) not null
 - description: Performance category identifier, e.g. "ia", "aci"/"pi", "quality", or "cost"
- value
 - type: float8 not null
 - description: The score achieved by the measurement set. This will be the same as the category's score value, if this is the highest scoring picked measurement set.
- detail
 - type: text
 - description: Submission method. In instances where measurement_set.submission_method is not available, then this field can be used to fill in the blanks.
- title

³⁰ <https://cmsgov.github.io/qpp-submissions-docs/scoring#measurement-set-score-part-resource>

³¹ <https://confluence.cms.gov/display/QPPAR/API+JSON+to+UDS+Mapping>

- type: varchar(255)
- description: Measurement set title.
- max_contribution
 - type: float8
 - description: Maximum contribution the measurement set could contribute to the category score.
- measurement_set_picked
 - type: bool
 - description: Flag if the measurement set contributed to the final score. Y1/Y2 true/false indicating whether measurement was picked for category score
- total_measurement_points
 - type: float8
 - description: Sum of individual measurement base scores, whether picked or skipped.
- total_bonus_points
 - type: float8
 - description: Sum of individual measurement bonus scores, whether the measurement was picked or skipped.
- denominator
 - type: float8
 - description: Denominator value used when calculating the measurement set score.
- e2e_bonus_score
 - type: float8
 - description: Total end-to-end bonus score for the measurement set.
- reweighted_score
 - type: float8
 - description: Score as if this measurement set was chosen as its overall category score and sent to final scoring and reweighted.
- has_high_priority_measurements
 - type: bool
 - description: Flag if measurement set contains high priority measurements.
- is_ema_eligible
 - type: bool
 - description: Flag if measurement set satisfies an EMA scoring scenario.
- is_ema_ghost
 - type: bool
 - description: Flag if measurement is a ghost.
- bonus_score_cap_limit_reached
 - type: bool
 - description:
- max_measurement_allowed
 - type: float8
 - description:
- unrounded_score_value
 - type: float8
 - description: The unrounded value of the measurement set score part.
- cpc_plus_status
 - type: varchar(255)
 - description: Flag if measurement set score part belongs to a CPC+ eligible measurement set.
- msg_attestation_statement_check
 - type: text
 - description: A string message for front end systems.
- msg_base_measure_check
 - type: text
 - description: A string message for front end systems.

- msg_denominator
 - type: text
 - description: A string message for front end systems. Details how denominator was determined.
- msg_total_measurement_points
 - type: text
 - description: A string message for front end systems. Description of this metadata field.
- msg_total_bonus_points
 - type: text
 - description: A string message for front end systems. Description of this metadata field.
- msg_has_high_priority_measures
 - type: text
 - description: A string message for front end systems. Describes Scoring Engine detection of high priority measurements.
- msg_e2e_bonus_score
 - type: text
 - description: A string message for front end systems. A message that states Bonus from isEndToEndReported, e2eBonusScore is defined as Total end-to-end bonus score for the measurement set
- msg_bonus_score_cap_limit_reached
 - type: text
 - description: A string message for front end systems.
- msg_base_track_warnings
 - type: text
 - description: A string message for front end systems.
- msg_cpc_plus_status
 - type: text
 - description: A string message for front end systems.
- msg_is_ema_ghost
 - type: text
 - description: A string message for front end systems. EMA Ghost Submission messaging
- all_below_minimum
 - type: bool
 - description:
- attestation_bonus_eligible
 - type: bool
 - description: Y1/Y2 attestation bonus eligibility
- cahps_score
 - type: float8
 - description:
- cehrt_eligible
 - type: bool
 - description:
- is_ghost
 - type: bool
 - description:
- is_total_bonus_points_capped
 - type: bool
 - description:
- is_total_decile_score_capped
 - type: bool
 - description:
- merged_with_cahps
 - type: bool

- description:
- msg_attestation_bonus_eligible
 - type: text
 - description: A string message for front end systems.
- msg_cehrt_eligible
 - type: text
 - description: A string message for front end systems.
- msg_ia_cehrt_bonus_check
 - type: text
 - description: A string message for front end systems.
- msg_merged_with_cahps
 - type: text
 - description: A string message for front end systems.
- msg_not_cehrt_eligible
 - type: text
 - description: A string message for front end systems.
- msg_registry_bonus
 - type: text
 - description: A string message for front end systems.
- msg_registry_performance_bonus
 - type: text
 - description: A string message for front end systems.
- msg_total_decile_score
 - type: text
 - description: A string message for front end systems.
- ssm_total_score
 - type: float8
 - description:
- total_cahps_measures
 - type: float8
 - description:
- total_decile_score
 - type: float8
 - description:
- ema_strategy
 - type: varchar(255)
- measures_submitted_count
 - type: float8
- missing_wi_measure_ids
 - type: varchar(255)[]
- previous_year_denominator
 - type: float8
- previous_year_numerator
 - type: float8
- previous_year_quality_score
 - type: float8
- processing_status
 - type: varchar(255)
- quality_improvement_bonus
 - type: float8
- small_practice_bonus
 - type: float8
- small_practice_bonus_eligible

- type: bool
- base_category_score
 - type: float8
 - description: Y1/Y2 base category score
- bonus_category_score
 - type: float8
 - description: Y1/Y2 bonus category score
- cehrt_category_score
 - type: float8
 - description: Y1/Y2 cehrt category score
- performance_category_score
 - type: float8
 - description: Y1/Y2 performance category score
- registry_additional_bonus
 - type: float8
 - description: Y1/Y2 additional registry bonus score
- registry_performance_bonus
 - type: float8
 - description: Y1/Y2 performance registry bonus score
- rollup_base
 - type: float8
 - description: Y1/Y2 rollup base score used for final scoring
- rollup_bonus
 - type: float8
 - description: Y1/Y2 rollup bonus score used for final scoring
- measurement_set_id
 - type: varchar(255)
 - description: The unique identifier of the measurement_set being scored. This could also be a non-identifier string in the case of aggregation or merges.
- instrumentations
 - type: text
- performance_period
 - type: float8
 - description: The total number of days calculated from the performance start and end dates
- performance_start
 - type: date
 - description: The start date of the performance period in the following format yyyy/mm/dd
- performance_end
 - type: date
 - description: The end date of the performance period in the following format yyyy/mm/dd
- has_cehrt_id
 - type: bool
 - description: Boolean indicating the measurement set has a cehrt identification number
- cehrt_id
 - type: varchar(255)
 - description: The cehrt identification number
- cehrt_bonus
 - type: float8
- cehrt_2015_bonus
 - type: float8
- electronic_prescribing
 - type: varchar(255)
 - description: Score contributed by Electronic Prescribing Objective

- health_information_exchange_sending
 - type: varchar(255)
 - description: Score contributed by Health Information Exchange Sending Objective
- health_information_exchange_receiving
 - type: varchar(255)
 - description: Score contributed by Health Information Exchange Receiving Objective
- provider_to_patient_exchange
 - type: varchar(255)
 - description: Score contributed by Provider To Patient Exchange Objective
- public_health_and_clinical_data_exchange
 - type: varchar(255)
 - description: Score contributed by Public Health And Clinical Data Exchange Objective
- electronic_prescribing_pdmp_bonus
 - type: varchar(255)
- electronic_prescribing_opioid_bonus
 - type: varchar(255)
- is_suppressed
 - type: boolean
 - description: Indicates that the measurement set was ignored for scoring
- strategy
 - type: varchar(255)

5.3.5 measurement_score

5.3.5.1 Description

From the API documentation (see [here](#)³²):

"The Measurement Score resource represents the scoring result for each measurement within a submission's measurement sets."

Represents the score for a given measurement. Measurement score values for all performance categories should be in this table.

Keys: The surrogate key is db_id.

5.3.5.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_measurement_score_part_id
 - type: int8
 - description: Foreign key to the measurement_score_part table - measurement_score_part.db_id. Only populated for ACI / PI measurements.
- db_measurement_set_score_part_id
 - type: int8 not null
 - description: Foreign key to the measurement_set_score_part table - measurement_set_score_part.db_id.

³² <https://cms.gov.github.io/qpp-submissions-docs/scoring#measurement-score-resource>

- db_measurement_id
 - type: int8
 - description: Foreign key to the measurement table - measurement.db_id. Sometimes this is not populated if the measurement was "moved" by the scoring engine and we can't link back to the original measurement in the submissions data.
- name
 - type: varchar(255)
 - description: Measurement score identifier. Usually the measure_id.
- value
 - type: float8
 - description: Measurement score value.
- weight
 - type: varchar(255)
 - description: This applies only applies to IA measures. Improvement activities have been assigned to one of two categories: Medium-Weighted or High-Weighted
- max_contribution
 - type: float8
 - description: Maximum point contribution possible for the measurement.
- performance_rate
 - type: float8
 - description: Performance rate for the measurement.
- reporting_rate
 - type: float8
 - description: Reporting rate for the measurement.
- measure_class
 - type: varchar(255)
 - description: The class of this measurement, used to determine how it is scored. This is based on the availability of benchmark and measure data, as well as attributes of the measure's score, as well. Class I, Class II, or Class III.
- measure_title
 - type: text
 - description: Title of the measurement.
- end_to_end_bonus
 - type: float8
 - description: How many end-to-end bonus points this measure can contribute.
- end_to_end_bonus_eligible
 - type: bool
 - description: Boolean indicator of whether the measurement is eligible to receive end-to-end bonus points.
- outcome_or_patient_experience_bonus
 - type: float8
 - description: How many outcome or patient experience bonus points this measure can contribute.
- high_priority_bonus
 - type: float8
 - description: How many high priority bonus points this measure can contribute.
- high_priority_bonus_eligible
 - type: bool
 - description: Boolean indicator of whether a measure is eligible to receive high priority bonus points.
- high_priority_bonus_ignored
 - type: bool
 - description: Whether or not the measurement's high priority bonus eligibility was ignored.
- decile_score
 - type: float8

- description: The decile according to the benchmark if the measurement is a Class I measurement, or the base score if the measurement is a Class II. Does not include bonus points, and is the raw score.
- performance_denominator
 - type: float8
 - description: The performance denominator used in calculating the performance and reporting rates.
- performance_numerator
 - type: float8
 - description: The performance numerator used in calculating the performance and reporting rates.
- eligible_population
 - type: float8
 - description: Eligible population passed in from the submission.
- partial_decile_score
 - type: float8
 - description: For Class I measurements, the decimal part of the decile score.
- partial_points
 - type: float8
 - description: For Class I measurements, the difference between the performance rate and the lower bound of the decile range. Thought of as "how far into" the decile the measurement fits.
- decile
 - type: float8
 - description: The decile a Class I measurement fits in to according to its benchmark data.
- decile_lower_bound2
 - type: float8
 - description: The lower bound from the range of Decile 2 level of national performance within the baseline period.
- decile_lower_bound3
 - type: float8
 - description: The lower bound from the range of Decile 3 level of national performance within the baseline period.
- decile_lower_bound4
 - type: float8
 - description: The lower bound from the range of Decile 4 level of national performance within the baseline period.
- decile_lower_bound5
 - type: float8
 - description: The lower bound from the range of Decile 5 level of national performance within the baseline period.
- decile_lower_bound6
 - type: float8
 - description: The lower bound from the range of Decile 6 level of national performance within the baseline period.
- decile_lower_bound7
 - type: float8
 - description: The lower bound from the range of Decile 7 level of national performance within the baseline period.
- decile_lower_bound8
 - type: float8
 - description: The lower bound from the range of Decile 8 level of national performance within the baseline period.
- decile_lower_bound9
 - type: float8
 - description: The lower bound from the range of Decile 9 level of national performance within the baseline period.

- `decile_lower_bound10`
 - type: float8
 - description: The lower bound from the range of Decile 10 level of national performance within the baseline period.
- `processing_status`
 - type: varchar(255)
 - description: Whether or not the measurement was picked to calculate the base score. Skipped measurements may still contribute to the score by bonuses. Can be: `SKIPPED`, `PICKED`, `INVALID`.
- `total_measurement_points`
 - type: float8
 - description: Total points the measurement can contribute including base and bonus.
- `total_bonus_points`
 - type: float8
 - description: Total bonus points the measurement can contribute.
- `no_benchmarks`
 - type: bool
 - description: If the measurement lacks benchmark data.
- `benchmark_type`
 - type: varchar(255)
 - description: Submission method with which the measurement benchmark data corresponds.
- `e_measure_id`
 - type: varchar(255)
 - description: The measurement's eMeasureId.
- `cpc_plus_group`
 - type: varchar(255)
 - description: CPC+ group identifier for the measurement if it belongs to a CPC+ eligible measurement set.
- `unrounded_score_value`
 - type: float8
 - description: The unrounded value of the measurement score.
- `msg_measurement_class`
 - type: text
 - description: Details of how the measurement class was determined.
- `msg_decile_score`
 - type: text
 - description: Details how the decile score was applied according to the classification of the measurement.
- `msg_measurement_picker`
 - type: text
 - description: The place in which the measurement was picked for scoring.
- `msg_total_measurement_points`
 - type: text
 - description: How points are included in the total score for the measurement based on its processing status.
- `msg_e2e_bonus_score`
 - type: text
 - description: A message that states Bonus from `isEndToEndReported`, `e2eBonusScore` is defined as Total end-to-end bonus score for the measurement set
- `msg_outcome_bonus`
 - type: text
 - description: This message defines how many outcome or patient experience bonus points this measurement can contribute. EX) `outcomeOrPatientExperienceBonus = 0`, Message = Default

- outcome bonus; outcomeOrPatientExperienceBonus = 2, Message = Allotting 2 points to outcome bonus for measure type 'outcome'
- msg_high_priority_bonus
 - type: text
 - description: Condition 1: highPriorityBonusEligible: True && highPriorityBonus: 0 && highPriorityBonusIgnored: True, Message = Measure having highest decile score (preference given to outcome, intermediate outcome, and patient engagement experience) is not eligible for high priority bonus ; Condition 2: highPriorityBonusEligible: True && highPriorityBonus: 0, Message = Default high priority bonus where highPriorityBonus: How many high priority bonus points this measurement can contribute and highPriorityBonusIgnored: First high priority bonus measure ignored
- msg_benchmark_type
 - type: text
 - description: Submission method with which the measurement benchmark data corresponds
- title
 - type: varchar(255)
 - description:
- detail
 - type: varchar(255)
 - description:
- cahps_score
 - type: float8
 - description:
- complete
 - type: bool
 - description:
- cost_per_occurrence
 - type: float8
 - description:
- denominator
 - type: float8
 - description:
- eligible_occurrences
 - type: float8
 - description:
- is_c2q_eligible
 - type: bool
 - description: Whether or not the measurement is C2Q eligible
- is_ema_ghost
 - type: bool
 - description: Whether or not the EMA measurement scored 0/0.
- is_ghost
 - type: bool
 - description: Whether or not the measurement scored 0/0.
- msg_complete
 - type: text
 - description:
- msg_is_ghost
 - type: text
 - description:
- ratio
 - type: float8
 - description:
- ssm_total_score

- type: float8
 - description:
- topped_out
 - type: bool
 - description: Whether or not the measurement is at max points.
- total_cahps_measures
 - type: float8
 - description:
- wi_case_minimum_volume
 - type: float8
 - description: Eligible Population - (Eligible Population Exception + Eligible Population Exclusion)
- ema_eligibility_scenarios
 - type: text
 - description: The EMA eligibility scenarios that this measure qualifies for, in key-value pairs
- instrumentations
 - type: text
 - description:
- measurement_set_id
 - type: varchar(255)
 - description: Unique identifier of the measurement_set being scored. May be a non-identifier in the case of merge or aggregation.
- measurement_id
 - type: varchar(255)
 - description: Unique identifier of the measurement being scored. May(?) be empty or blank if due to auto-credit.
- objective
 - type: varchar(255)
- calculated_score
 - type: float8
- submission_method
 - type: varchar(255)
- performance_start
 - type: date
- performance_end
 - type: date
- mips_minimum_score
 - type: float8
- reporting_category
 - type: varchar(255)
- formula
 - type: text
- pdmp_bonus
 - type: float8
- opioid_bonus
 - type: float8
- is_cpc_plus
 - type: bool
- exclusion_score
 - type: float8
- unskipped_score_value
 - type: float8

5.3.6 measurement_score_part

5.3.6.1 Description

The ACI measurement scores are grouped under aci_base, aci_bonus, aci_performance, and cerht_bonus. This table represents that grouping. Therefore, measurement_score is also a child of measurement_score_part.

Keys: The surrogate key is db_id.

5.3.6.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_measurement_set_score_part_id
 - type: int8 not null
 - description: Foreign key to the measurement_set_score_part table - measurement_set_score_part.db_id.
- name
 - type: varchar(255) not null
 - description: Score part identifier. 'aci_base' / 'aci_performance' / 'aci_bonus' / 'cerht_bonus', if the measurementSet is ACI (year 1). 'pi_base' / 'pi_performance' / 'pi_bonus' / 'cerht_bonus', if the measurementSet is PI (year 2+).
- title
 - type: text
 - description: Title of the measurement score part.
- value
 - type: float8 not null
 - description: Total score the score part contributes.
- detail
 - type: text
 - description: Optional description to further explain measurement score part.
- max_contribution
 - type: float8
 - description: Maximum point contribution possible for the measurement score part.
- unrounded_score_value
 - type: float8
 - description: The unrounded value of the measurement score part.
- attestation_bonus
 - type: float8
 - description:
- base_score
 - type: float8
 - description:
- cerht_2015_bonus
 - type: float8
 - description:
- performance_score
 - type: float8

- description:
- ehr_cehrt_bonus
 - type: float8
 - description:

5.3.7 measures_picked

5.3.7.1 Description

Represents the quality measures that were picked or skipped for scoring in the quality performance category.

Keys: The surrogate key is db_id.

5.3.7.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_measurement_set_score_part_id
 - type: int8 not null
 - description: Foreign key to the measurement_set_score_part table - measurement_set_score_part.db_id.
- db_measure_id
 - type: int8 not null
 - description: Foreign key to the measure table - measure.db_id.
- message
 - type: text
 - description: If the measure was skipped, the message indicates why.
- skipped_picked
 - type: varchar(255) DEFAULT 'picked'
 - description: Indicates whether the measure was "picked" or "skipped" for scoring.

6 Final Scoring

6.1 Data Sources

The Final Scoring domain contains several data sets from different sources that all relate to "what score did someone get for the performance year, and what payment adjustment will they receive next year corresponding to that score". The sources of data include:

- Final Scoring - the name of the processes by which final scores and metadata are calculated. Final Scoring generates several different data sets that we pull into the UDS. All Final Scoring data is stored in Hive / parquet files.
 - final_score - these are our "candidate scores" as well as the final_score table. It contains every generated score for an individual clinician, organization, virtual group, and MIPS APM entity, and includes which score is the final score for all individual clinicians (both eligible and ineligible).
 - rollup - PI rollup metadata for APM Entities
 - rollup_participation - PI rollup metadata for APM Participants
 - final_score_se_results - The scoring engine JSONs which we use to populate the Submission Scoring domain
 - hvbp_provider_score - Hospital Value-Based Purchasing. Eligible providers may receive a Final Score that is based on their Hospital VBP score.
 - hvbp_provider_preview - same as hvbp_provider_score, but estimates for the next PY
 - hvbp_score_percentiles - Percentiles associated with hvbp scores.
 - hvbp_preview_percentiles - same as hvbp_score_percentiles, but estimates for the next PY
 - final_scoring_rules - Rules / Codes for a particular run
- Payment Adjustments - Calculated every performance year by OACT (Office of the Actuary), accounting for score distributions, projected claims spending, and budget neutrality mandates.
 - Payment adjustment data can be provided in two different ways. For more details see [Payment Adjustment Map \("Matrix"\)](#)³³.
- Targeted Reviews - scores explicitly set by a user after review of a clinician's eligibility and submissions. Overrides any score from "regular" Final Scoring processes. Because PI rollup is recalculated as a result of Targeted Review scores, our source for all targeted review data is also the Final Scoring Hive / parquet files of the following type:
 - targeted_review - All targeted review scores, whether user-created (ServiceNow, not "fabricated") or generated by final scoring processes ("fabricated")

6.1.1 Data Source documentation

[Data Schemas/Models \(HIVE/Parquet File Structures\)](#)³⁴ - This page contains the documented schemas of the Hive Final Scoring data sources.

[Final Scoring - How To Generate And Publish](#)³⁵ - This page documents JSON metadata that Final Scoring uses to generate and publish runs of Final Scoring.

[Final Score Dissection](#)³⁶ - writeup by FS team on various scoring scenarios.

33 <https://confluence.cms.gov/pages/viewpage.action?pagelId=274548341>

34 <https://confluence.cms.gov/pages/viewpage.action?pagelId=133290401>

35 <https://confluence.cms.gov/display/QPPRACTICE/Final+Scoring+-+How+To+Generate+And+Publish>

36 <https://confluence.cms.gov/display/QPPRACTICE/Final+Score+Dissection>

6.1.2 Data ingestion processes

Documentation of the Final Scoring pipeline UDS_Scoring is here: [Final Scoring pipeline](#)³⁷. This pipeline ingests all Final Scoring data sets (including targeted review), Submission Scoring, and score-to-payment adjustment maps.

6.2 Tables

6.2.1 final_score

6.2.1.1 Description

Each record in this table represents the final score for a MIPS-eligible TIN-NPI pair. This includes individual participants in MIPS APMs. This table is primarily used to produce a final score file which is sent to OACT, who provides payment adjustments. OACT is responsible for ensuring that the payment adjustments keep the program budget-neutral. Final scores are provided in two ways: 1) through the final score calculation process, which runs after all submission data is received and validated; and 2) through targeted reviews. The former is captured in the candidate_score table. The latter is captured in the targeted_review table. In most cases, TIN-NPIs will receive a final score during the final score calculation process. However, there are cases when TIN-NPIs that should have received a final score but they have not. In those cases, no corresponding candidate_score record will exist, but a targeted_review record will exist.

Load Process: The data in this table, candidate_score, and targeted_review is loaded periodically from the Submission API database. Each time these data are loaded, new records are inserted; they are never updated. This load is represented by the year and load_number in the data_load table (uds_domain='Final Scoring'). Therefore, queries against this table should always join to candidate_score/targeted_review and data_load in order to restrict the query to a given performance year and load number.

Keys: This table is a join table consisting of foreign keys to other tables. Its surrogate key is db_id.

6.2.1.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_submission_id
 - type: int8
 - description: Foreign key to the submission table - submission.db_id. This isn't always populated, but if the candidate_score record includes a submission_id (or apm_entity_submission_id for 2017 data), this foreign key uses that field to link to the active submission row.
- db_organization_id
 - type: int8 not null
 - description: Foreign key to the organization table - organization.db_id.
- db_provider_org_id
 - type: int8 not null
 - description: Foreign key to the provider_org table - provider_org.db_id.
- db_candidate_score_id

³⁷ <https://confluence.cms.gov/display/QPPAR/Final+Scoring+pipeline>

- type: int8
 - description: Foreign key to the candidate_score table - candidate_score.db_id.
- db_targeted_review_id
 - type: int8
 - description: Foreign key to the targeted_review table - targeted_review.db_id.
- tin
 - type: varchar(255)
 - description: The 9-digit identifier of the participant (Taxpayer Identification Number).
- npi
 - type: varchar(255)
 - description: The 10-digit identifier of the participant (National Provider Identifier).
- year
 - type: int2
 - description: The performance year
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

6.2.2 candidate_score

6.2.2.1 Description

Each record in this table represents intermediate scores that are calculated during the final scoring process. The final score assigned to a participant will be linked to this table through final_score.db_candidate_score_id. Refer to the data load process described in the final_score table description above.

The field "quality_reason" identifies the reporting type, with values of: (individual, group, apm, noData, neutral, extremeHardship, and system) - the last two are rarely used. The value of "neutral" indicates that the candidate got a neutral score due to a special circumstance. NOTE: These need to be updated to reflect 2018 data.

Keys: The surrogate key in UDS is db_id. The candidate_score_uuid field maps to the surrogate key "id" in the source system.

6.2.2.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_data_load_id
 - type: int8 not null
 - description: Foreign key to data_load.db_id
- candidate_score_uuid
 - type: varchar(255) not null.
 - description: UUID for the candidate score. Generated by the source system.
- npi
 - type: varchar(255)
 - description: The 10-digit identifier of the provider associated with the score (National Provider Identifier). Populated if the row represents an individual clinician.
- tin
 - type: varchar(255)

- description: The 9-digit identifier of the provider associated with the score (Taxpayer Identification Number). Populated if the row represents an individual clinician or a group.
- entity_type
 - type: varchar(255)
 - description: This is the type of entity by which the score was generated. This is not necessarily the same thing as the row_type (especially for 2018 and beyond). Values include: "individual", "group", "virtual_group", "apm". Other values include: "neutral_apm" (will be adjusted to "apm" in a future release), "neutral_ineligible", and "neutral_ghost".
 - Right now, entity_type is only mapped correctly for final scores for 2018. Some work to be done here.
- overall_score
 - type: float8 not null
 - description: Overall score - out of 100, 2 fractional digits.
- quality_score
 - type: float8 not null
 - description: Unweighted category score. Out of 100, 8 fractional digits.
- quality_weight
 - type: float8 not null
 - description: The weight of the quality category score that should be applied to the unweighted quality score to generate the total overall_score. (what does this mean? weighting scenario lookup directly by APM ID or indirectly by APM Entity ID → APM ID). Percentage expressed as decimal (65% → 0.65).
- quality_reason
 - type: varchar(255) not null
 - description: Reason for quality score. The field identifies the reporting type. In 2017 it had values of: (individual, group, apm, noData, neutral, extremeHardship, and system) - the last two are rarely used. The value of "neutral" indicates that the candidate got a neutral score due to a special circumstance. In 2018 (more details here)
- aci_score
 - type: float8 not null
 - description: Unweighted PI/ACI category score. (is this true...? Numerator = Group Score * count of NPIs / 100 * count of NPIs.) 2 decimal when coming from scoring engine, 3 when from aci rollup. Out of 100, 8 fractional digits.
- aci_weight
 - type: float8 not null
 - description: The weight of the pi/aci category score that should be applied to the unweighted pi/aci score to generate the total overall_score. (what does this mean? From the weighting scenario for the APM subdivision or because of special status.) Percentage expressed as decimal (65% → 0.65).
- aci_reason
 - type: varchar(255) not null
 - description: Reason for ACI / PI score.
- ia_score
 - type: float8 not null
 - description: Unweighted category score. Out of 100, 8 fractional digits.
- ia_weight
 - type: float8 not null
 - description: The weight of the ia category score that should be applied to the unweighted ia score to generate the total overall_score. (what does this mean? From the weighting scenario for the APM subdivision or because of special status.) Percentage expressed as decimal (65% → 0.65).
- ia_reason
 - type: varchar(255) not null
 - description: Reason for IA score.
- submission_id
 - type: varchar(255)

- description: Optional (only present if the score comes from a submission). Submission UUID of the selected candidate score.
- apm_entity_id
 - type: varchar(255)
 - description: Optional (only present if NPI or TIN is an APM participant or the row is an APM entity row). ID of the APM entity.
- apm_entity_submission_id
 - type: varchar(255)
 - description: 2017 field only. Optional (only present if NPI is an APM participant). Submission UUID of the APM entity submission (used only for reporting Quality).
- apm_id
 - type: varchar(255)
 - description: Optional (only present if NPI or TIN is an APM participant). ID of the APM model.
- apm_name
 - type: varchar(255)
 - description: Optional (only present if NPI or TIN is an APM participant). Name of the APM model.
- apm_subdivision_id
 - type: varchar(255)
 - description: Optional (only present if NPI or TIN is an APM participant). ID of the APM subdivision.
- apm_subdivision_name
 - type: text
 - description: Optional (only present if NPI or TIN is an APM participant). Name of the APM subdivision.
- payment_adjustment
 - type: float8
 - description: The total payment adjustment achieved by the candidate overall_score, the sum of the base payment adjustment and the exceptional performance bonus. Payment adjustments are determined by OACT based on the distribution of eligible final scores for the performance year and taking into account budget neutrality. This field is equivalent to `uds.payment_adjustment.total_adjustment_percentage`. This version of the data is stored as a "percentage", so if the value in the payment_adjustment table is 0.0187, the value displayed here is 1.87.
- performance_bonus
 - type: float8
 - description: If the candidate overall_score exceeds a certain threshold, an exceptional performance bonus is awarded on top of the base payment adjustment. This field plus the `base_payment_adjustment` equals the total payment_adjustment field. This is equivalent to `uds.payment_adjustment.exceptional_adjustment_percentage`. This version of the data is stored as a "percentage", so if the value in the payment_adjustment table is 0.0187, the value displayed here is 1.87.
- year
 - type: int2 not null
 - description: The year in which performance data for the scoring was collected.
`payment_adjustment.year=candidate_score.year`
- base_payment_adjustment
 - type: float8
 - description: The base payment adjustment achieved by the candidate overall_score. This field plus the `performance_bonus` equals the total payment_adjustment field. This is equivalent to `uds.payment_adjustment.base_adjustment_percentage`. This version of the data is stored as a "percentage", so if the value in the payment_adjustment table is 0.0187, the value displayed here is 1.87.
- cost_score
 - type: float8
 - description: Unweighted category score. Out of 100, 8 fractional digits.

- **cost_weight**
 - type: float8
 - description: The weight of the cost category score that should be applied to the unweighted cost score to generate the total overall_score. Percentage expressed as decimal (65% → 0.65).
- **cost_reason**
 - type: varchar(255)
 - description: Reason for cost score.
- **complex_patient_bonus**
 - type: float8
 - description: A bonus on top of the four categories that was added in 2018.
- **small_practice_bonus**
 - type: float8
 - description: A bonus on top of the four categories that was added in 2018.
- **is_voluntary**
 - type: bool
 - description: Boolean flag added in 2018. Ideally, this would indicate if a participant is voluntary; but it does NOT. Beware until we get a concrete definition of how the source team set this flag.
- **is_eligible**
 - type: bool
 - description: Boolean flag added in 2018 indicating if the individual clinician is eligible for a final score and payment adjustment.
- **reasons**
 - type: varchar(255)[] (string array)
 - description: An array of reason codes, added in 2018, indicating various situations that applied to end up with this score record.
- **is_ghost**
 - type: bool
 - description: Boolean flag added in 2018. Indicates if the individual clinician is receiving this score due to their presence in the "ghost" run of Eligibility (aka run 5; the run that is conducted over the last four months of claims for the performance year).
- **virtual_group_id**
 - type: varchar(255)
 - description: Optional (only present if NPI or TIN is a virtual group participant or the row is a virtual group row). ID of the virtual group.
- **row_type**
 - type: varchar(255)
 - description: Added in 2018. Indicates what type of entity (individual clinician; group organization; APM entity; virtual group) this row represents. Possible values are: "individual", "group", "apm", "virtual_group".
- **is_neutral_pa**
 - type: bool
 - description: Boolean flag added in 2018 in which the final scoring code indicates whether this participant should receive a neutral payment adjustment, regardless of score received.
- **ssp_group**
 - type: varchar(255)
 - description: Indicates which APM Entity SSP Group (A, B, or null) that the clinician or APM entity's score utilized for PI rollup calculation. Only applies for the 2019 performance year.
- **hvbp_id**
 - type: varchar(255)
 - description: Foreign Key to hvbp_provider_score.id

6.2.3 targeted_review

6.2.3.1 Description

Each record in this table represents a final score that has been changed or created through the targeted review process. Refer to the "Load Process" discussion on the final_score section above.

Keys: The surrogate key in UDS is db_id. The targeted_review_uuid field maps to the surrogate key "id" in the source system.

6.2.3.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- targeted_review_uuid
 - type: varchar(255) not null
 - description: UUID for the targeted review. Created by the source system.
- npi
 - type: varchar(255)
 - description: The 10-digit identifier of the provider associated with the targeted review (National Provider Identifier), if applicable.
- tin
 - type: varchar(255) not null
 - description: The 9-digit identifier of the organization associated with the targeted review (Taxpayer Identification Number).
- entity_type
 - type: varchar(255) not null
 - description: Type of entity of the targeted_review: "individual", "group", "apm", or "virtual_group". This represents the pathway through which a clinician achieves their score, and may not be the same as row_type.
- apm_entity_id
 - type: varchar(255)
 - description: ID of the APM entity. Present if the targeted review is for the entire APM entity, or if the NPI or TIN is an APM participant.
- ia_score
 - type: float8 not null
 - description: Unweighted IA category score. Out of 100, 8 fractional digits (? what does this mean?).
- ia_weight
 - type: float8 not null
 - description: From the weighting scenario for the APM subdivision or because of special status. Percentage expressed as decimal (65% → 0.65).
- aci_score
 - type: float8 not null
 - description: Unweighted ACI / PI category score. Out of 100.
 - some notes (what are these?) Numerator = Group Score * count of NPIs / 100 * count of NPIs. 2 decimal when coming from scoring engine, 3 when from aci rollup. Out of 100, 8 fractional digits.
- aci_weight
 - type: float8 not null

- description: From the weighting scenario for the APM subdivision or because of special status. Percentage expressed as decimal (65% → 0.65).
- quality_score
 - type: float8 not null
 - description: Unweighted quality category score. Out of 100, 8 fractional digits. (?)
- quality_weight
 - type: float8 not null
 - description: weighting scenario lookup directly by APM ID or indirectly by APM Entity ID → APM ID. Percentage expressed as decimal (65% → 0.65).
- overall_score
 - type: float8 not null
 - description: Overall score calculation - out of 100, 2 fractional digits. This is to be the entity's new final score, if higher than their candidate score(s).
- payment_adjustment
 - type: float8
 - description: The total payment adjustment achieved by the candidate overall_score, the sum of the base payment adjustment and the exceptional performance bonus. Payment adjustments are determined by OACT based on the distribution of eligible final scores for the performance year and taking into account budget neutrality. This field is equivalent to uds.payment_adjustment.total_adjustment_percentage. This version of the data is stored as a "percentage", so if the value in the payment_adjustment table is 0.0187, the value displayed here is 1.87.
- performance_bonus
 - type: float8
 - description: If the candidate overall_score exceeds a certain threshold, an exceptional performance bonus is awarded on top of the base payment adjustment. This field plus the base_payment_adjustment equals the total payment_adjustment field. This is equivalent to uds.payment_adjustment.exceptional_adjustment_percentage. This version of the data is stored as a "percentage", so if the value in the payment_adjustment table is 0.0187, the value displayed here is 1.87.
- base_payment_adjustment
 - type: float8
 - description: The base payment adjustment achieved by the candidate overall_score. This field plus the performance_bonus equals the total payment_adjustment field. This is equivalent to uds.payment_adjustment.base_adjustment_percentage. This version of the data is stored as a "percentage", so if the value in the payment_adjustment table is 0.0187, the value displayed here is 1.87.
- year
 - type: int2 not null
 - description: The performance year
- db_data_load_id
 - type: int8
 - description: Foreign key to data_load.db_id
- cost_score
 - type: float8
 - description: Unweighted Cost category score. Out of 100.
- cost_weight
 - type: float8
 - description: The weight applied to the cost category.
- virtual_group_id
 - type: varchar(255)
 - description: ID of the virtual group. Present if the targeted review is for the virtual group.
- contributing_tr_ids

- type: text
- description: New for 2018. If this targeted review record is generated via multiple other targeted reviews (as in the case for an APM entity), this contains the array of targeted_review_uuid string identifiers that added to the overall targeted review score calculation.
- is_fabricated
 - type: bool
 - description: New for 2018. A boolean flag indicating if the targeted review was generated via the Final Scoring process (true) or not (data comes from ServiceNow).
- row_type
 - type: varchar(255)
 - description: The type of row for this targeted review: "individual", "group", "apm", "virtual_group".
- apm_id
 - type: varchar(255)
 - description: Optional (only present if targeted review is due to APM participation). ID of the APM model.
- apm_name
 - type: varchar(255)
 - description: Optional (only present if targeted review is due to APM participation). Name of the APM model.
- apm_subdivision_id
 - type: varchar(255)
 - description: Optional (only present if targeted review is due to APM participation). ID of the APM subdivision.
- apm_subdivision_name
 - type: text
 - description: Optional (only present if targeted review is due to APM participation). Name of the APM subdivision.

6.2.4 hvbp_provider_score/hvbp_provider_preview

6.2.4.1 Description

Eligible providers may receive a Final Score that is based on their Hospital VBP score. Preview table is sometimes generated with the same schema that estimate next years scores.

Keys: The surrogate key in UDS is db_id. The id field is mapped to the hvbp_id in the source final_scoring table.

6.2.4.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id
- id
 - type: varchar(255) not null
 - description: The UUID for this Provider HVBP Score VO object. Created by the source system.
- tin
 - type: varchar(255)

- description: The 9-digit identifier of the organization associated with the hvbp score (Taxpayer Identification Number).
- npi
 - type: varchar(255)
 - description: The 10-digit identifier of the provider associated with the hvbp score (National Provider Identifier), if applicable.
- entity_id
 - type: varchar(255)
 - description: The id of the individual, group or VG
- entity_type
 - type: varchar(255)
 - description: Enum of individual, group, or virtualGroup
- ccn
 - type: varchar(255)
 - description: Cms Certification Number
- hospital_name
 - type: varchar(255)
 - description: hospital name
- eligible
 - type: bool
 - description: Whether or not the provider is eligible to receive a score—should mirror their Final Scoring eligibility profile
- percentile
 - type: int
 - description: Signed integer of the percentile this Hospital VBP score falls into. Defaults to -1 for ineligible providers
- overall_score
 - type: double
 - description: Calculated based on component scores and weights
- quality_score
 - type: double
 - description: score that represents the quality of care.
- quality_weight
 - type: double
 - description: Ranges 0-1, defaults to 0.45
- cost_score
 - type: double
 - description: score that reflects cost of care.
- cost_weight
 - type: double
 - description: Ranges 0-1, defaults to 0.45
- pi_score
 - type: double
 - description: Will always be 0.0 for preview, or if submission_id is null
- pi_weight
 - type: double
 - description: Will always be 0.0 for preview, or if submission_id is null
- ia_score
 - type: double
 - description: Will always be 0.0 for preview, or if submission_id is null
- ia_weight
 - type: double
 - description: Will always be 0.0 for preview, or if submission_id is null

- complex_patient_bonus
 - type: double
 - description: Will always be 0.0 for preview, or if submission_id is null
- submission_id
 - type: varchar(255)
 - description: The submission ID that is responsible for the IA and PI values on this score. Always null during preview.
- base_score_id:
 - type: varchar(255)
 - description: The base score ID that is responsible for PI and IA weights in the absence of a submission score. Always null during preview.
- reasons
 - type: text
 - description: A string array, consisting of business rules applied (not optional)

6.2.5 hvbp_score_percentiles/hvbp_preview_percentiles

6.2.5.1 Description

Reference table for hvbp scores. Preview table is sometimes generated to estimate next years percentiles.

Keys: The surrogate key in UDS is db_id.

6.2.5.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id
- percentile
 - type: int
 - description: The percentile associated with these scores, 0-99.
- hvbp_score
 - type: double
 - description: The HVBP score associated with this percentile
- quality_score
 - type: double
 - The quality score associated with this percentile
- cost_score
 - type: double
 - description: The cost score associated with this percentile
- year
 - type: int
 - The performance year of the run that produced these percentiles

6.2.6 payment_adjustment

6.2.6.1 Description

Each record in this table represents a payment adjustment for a TIN-NPI pair as assigned by OACT. The data in this table is loaded (appended, not overwritten) whenever a payment adjustment file is received from OACT. In 2017, this data was loaded as a "diff", not a full data load.

Keys: The surrogate key is db_id. The candidate key includes tin, npi, year, and db_data_load_id.

6.2.6.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_final_score_id
 - type: int8
 - description: Foreign key to the final_score table - final_score.db_id.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id
- tin
 - type: varchar(255)
 - description: The 9-digit identifier of the organization (Taxpayer Identification Number).
- npi
 - type: varchar(255)
 - description: The 10-digit identifier of the provider (National Provider Identifier).
- year
 - type: int4
 - description: The performance year from which the score and payment adjustment was calculated.
- score
 - type: float8
 - description: The total score achieved by the clinician in the performance year.
- projected_medicare_spending
 - type: float8
 - description: An estimate of the dollar amount of claims that will be submitted by the clinician in the following year to which the payment adjustment will be applied.
- base_adjustment_percentage
 - type: float8
 - description: The base payment adjustment achieved by the score. This field plus the exceptional_adjustment_percentage equals the total_adjustment_percentage field. This is equivalent to $uds.candidate_score.base_payment_adjustment$. This version of the data is stored as a decimal, so if the value in the candidate_score table is 1.87%, the value displayed here is 0.0187.
- base_adjustment_amount
 - type: float8
 - description: $projected_medicare_spending * base_adjustment_percentage$
- exceptional_adjustment_percentage
 - type: float8

- description: If the score exceeds a certain threshold, an exceptional performance bonus is awarded on top of the base payment adjustment. This field plus the base_adjustment_percentage equals the total_adjustment_percentage field. This is equivalent to uds.candidate_score.performance_bonus. This version of the data is stored as a decimal, so if the value in the candidate_score table is 1.87%, the value displayed here is 0.0187.
- exceptional_adjustment_amount
 - type: float8
 - description: ?? projected_medicare_spending * exceptional_adjustment_percentage
- total_adjustment_percentage
 - type: float8
 - description: The total payment adjustment achieved by the score, the sum of the base adjustment percentage and the exceptional performance bonus adjustment percentage. Payment adjustments are determined by OACT based on the distribution of eligible final scores for the performance year and taking into account budget neutrality. This field is equivalent to uds.candidate_score.payment_adjustment. This version of the data is stored as a decimal, so if the value in the candidate_score table is 1.87%, the value displayed here is 0.0187.
- total_adjustment_amount
 - type: float8
 - description: ?? projected_medicare_spending * total_adjustment_percentage
- is_latest
 - type: bool
 - description: Payment adjustments in 2017 were supplied as a "diff", not a full data set, so a single data_load could not be considered as the latest version of all the payment adjustment data for every clinician. This flag indicates whether this record is the latest version of the payment adjustment for the specified clinician.

6.2.7 apm_entity_aci_rollup_score

6.2.7.1 Description

Represents the ACI/PI rollup score that is calculated for an APM entity during the final scoring process. This score is based on the ACI/PI data submitted by the organizations and providers associated with the APM entity.

Keys: The surrogate key is db_id. The natural key is year, entity_id, db_data_load_id.

6.2.7.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_apm_entity_id
 - type: int8 not null
 - description: Foreign key to the apm_entity table - apm_entity.db_id.
- score
 - type: float8
 - description: This is the rolled-up unweighted ACI/PI score for a given APM entity
- entity_id
 - type: varchar(255)
 - description: The APM entity identifier.
- year

- type: int2
 - description: The performance year.
- db_data_load_id
 - type: int8
 - description: Foreign key to data_load.db_id
- base
 - type: float8
 - description: The base score contribution (only applicable for Y2+)
- bonus
 - type: float8
 - description: The bonus contribution (only applicable for Y2+)
- exempt
 - type: bool
 - description: default value is `false`, will tell if the apm entity is ACI/PI exempt
- ssp_group
 - type: varchar(255)
 - description: Indicates which APM Entity SSP Group (A, B, or null) that the clinician or APM entity's score utilized for PI rollup calculation. Only applies for the 2019 performance year.

6.2.8 apm_participant_aci_rollup_score

6.2.8.1 Description

Represents the ACI/PI rollup score that is calculated for a group or an individual APM participant during the final scoring process.

Keys: The surrogate key is db_id. The natural key is year, entity_id, tin, npi, db_data_load_id.

6.2.8.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_provider_apm_entity_id
 - type: int8
 - description: Foreign key to the provider_apm_entity table - provider_apm_entity.db_id.
- db_apm_entity_id
 - type: int8 not null
 - description: Foreign key to the apm_entity table - apm_entity.db_id.
- db_organization_id
 - type: int8 not null
 - description: Foreign key to the organization table - organization.db_id.
- db_provider_id
 - type: int8
 - description: Foreign key to the provider table - provider.db_id.
- score
 - type: float8
 - description: This refers to the unweighted ACI/PI score for a given TIN or TIN/NPI who is participating in a given APM entityId.
- entity_id

- type: varchar(255)
 - description: The APM entity identifier.
- tin
 - type: varchar(255)
 - description: The organization TIN
 - PII field
- npi
 - type: varchar(255)
 - description: The provider NPI
- year
 - type: int2
 - description: The performance year.
- db_data_load_id
 - type: int8
 - description: Foreign key to data_load.db_id
- first_name
 - type: varchar(255)
 - description: The first name of the provider
- last_name
 - type: varchar(255)
 - description: The last name of the provider
- base
 - type: float8
 - description: The PI score consists of a base score component, this field represent that. Added in 2018.
- bonus
 - type: float8
 - description: The bonus contribution (only applicable for Y2+)
- exempt
 - type: bool
 - description: Boolean flag added in 2018, tells if the provider / group's score is contributing towards the APM entity's rollup score.
- reasons
 - type: varchar(255)[] (string array)
 - description: A string array, consisting of business rules applied, added in 2018

6.2.9 neutral_payment_adjustment

6.2.9.1 Description

Each record in this table represents a TIN-NPI combination that should receive a neutral payment adjustment. A&R uses this information to generate the payment adjustment file that it sends to the MACs and FISS. This data was only supplied in this format for 2017. In 2018 and beyond, we have this information available in the regular Final Scoring domain tables.

Load Process: Cost+ generated this file once per user, following [this format](#)³⁸, and stored the data in FMS. A&R then loaded the data into the UDS. This pipeline is now deprecated.

Keys: The surrogate key is db_id. The natural key includes tin, npi, run, year.

³⁸ <https://confluence.cms.gov/display/QPPELIG/MIPS+Eligibility+and+special+status+file+layout>

6.2.9.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_neutral_payment_adjustment_load_id
 - type: int8 not null
 - description: Foreign key to the neutral_payment_adjustment_load table - neutral_payment_adjustment_load.db_id
- tin
 - type: varchar(255)
 - description: The taxpayer identification number of the clinician.
- npi
 - type: varchar(255)
 - description: The national provider identifier of the clinician.
- claims_types
 - type: varchar(255)
- mips_eligible_clinician
 - type: bool
- clinician_specialty
 - type: varchar(255)
- multiple_specialties
 - type: bool
- low_volume_status
 - type: bool
- low_volume_status_reason
 - type: varchar(255)
- low_volume_bene_count
 - type: int4
- total_part_b_expenditures
 - type: float8
- non_patient_facing_status
 - type: bool
- patient_facing_encounters
 - type: int4
- hospital_based_clinician_status
 - type: bool
- hospital_based_encounters
 - type: int4
- total_encounters_hospital
 - type: int4
- hospital_based_encounters_percentage
 - type: float8
- small_group_practice_status
 - type: bool
- total_npis_in_group_practice
 - type: int4
- rural_clinician_status
 - type: bool
- rural_clinician_claim_id
 - type: varchar(255)

- hpsa_clinician_status
 - type: bool
- hpsa_clinician_claim_id
 - type: varchar(255)
- ambulatory_surgical_center
 - type: bool
- encounters_at_asc
 - type: int4
- total_encounters_asc
 - type: int4
- asc_encounters_percentage
 - type: float8
- run
 - type: int4
 - description: The run number indicating the determination period from claims.
- year
 - type: int4
 - description: The performance year

6.2.10 neutral_payment_adjustment_load

6.2.10.1 Description

6.2.10.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- load_number
 - type: int4 not null
 - description: An incrementing integer showing how many times the data has been loaded.
- file_name
 - type: varchar(255)
 - description: The name of the neutral payment adjustment file
- load_time
 - type: timestamp not null
 - description: The date and time the neutral payment adjustment data was loaded to UDS

6.2.11 payment_adjustment_delta_eligibility

6.2.11.1 Description

This data is the 2017 TIN/NPI Delta Eligibility File from Eligibility team which identifies all TIN/NPIs present in Eligibility. It was used to ensure Clinicians who are not eligible for QPP, do not receive a payment adjustment. This data was only needed during the 2017 performance year. After that, we have everything we need in the regular Eligibility domain tables.

Keys: The surrogate key is db_id. The natural keys are tin, and np. There is a foreign key to payment_adjustment_delta_eligibility_load via db_payment_adjustment_delta_eligibility_load_id

6.2.11.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_payment_adjustment_delta_eligibility_load_id
 - type: int8 not null
 - description: Foreign key to the payment_adjustment_delta_eligibility_load table - payment_adjustment_delta_eligibility_load.db_id
- tin
 - type: varchar(255)
 - description: The taxpayer identification number of the clinician.
- np
 - type: varchar(255)
 - description: The national provider identifier of the clinician.

6.2.12 payment_adjustment_delta_eligibility_load

6.2.12.1 Description

This table contains one record for each time data is loaded into the table payment_adjustment_delta_eligibility. It includes metadata about the load, such as the sequential load_number, the file_name, the year the data pertains to, and the time of the load.

Keys: The surrogate key is db_id. The natural key is effectively file_name, year, and load_number.

PII Fields: none

NOTE: This table and other "load" tables will be consolidated in future UDS releases.

6.2.12.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- load_number
 - type: int4 not null
 - description: An incrementing integer showing how many times the data has been loaded.
- file_name
 - type: varchar(255)
 - description: The name of the file containing the payment adjustment delta eligibility data.
- load_time
 - type: timestamp not null
 - description: The time and date at which the file was processed.
- year
 - type: int4 not null

- description: The performance year.

6.2.13 final_scoring_rules

6.2.13.1 Description

Table contains the Final Scoring business rule identifiers and descriptions for each candidate score and rollup score record.

Keys: Surrogate key = db_id

PII Fields: none

6.2.13.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- rule_code
 - type: varchar(255)
 - description: The short alphanumeric code to identify the business rule that was evaluated.
- reason_code
 - type: varchar(255)
 - description: The semantic name for the business rule that was evaluated, and how it is accessed in the codebase.
- category
 - type: varchar(255)
 - description: The category for this rule, in camelCase
- description
 - type: varchar(255)
 - description: A short description for this rule, and what it means
- explanation
 - type: Text
 - description: An optional longer explanation for this rule, including examples and/or circumstances where it would apply.
- year
 - type: int4
 - description: performance year
- db_data_load_id
 - type: int8
 - description: Foreign key to data_load.db_id

7 Final Scoring, Score Profiler - Deprecated

7.1 Data Sources

This data was generated for the 2018 performance year. Due to lack of use of the data, it is no longer being created / updated for other performance years.

7.1.1 score_profile

7.1.1.1 Description

Segments the population of clinicians who received final scores into common profiles. This is used so that a clinician can compare her final score to the average score received by clinicians similar to herself. The profiles and average score data are in support of the Peer Comparison Feedback feature.

This document describes how the population of clinicians is segmented into distinct, non-overlapping profiles for the 2018 performance year: [Analytics Portal Score Profile POC](#)³⁹

In 2018, no APM participants or Virtual Group participants were included in score profile segmentation and score calculations.

Keys: The surrogate key is db_id. The natural key is the combination of a distinct profile: (practice_size_min, practice_size_max, state, small_or_rural).

Grains: The `uds` schema grain is: (db_data_load_id, practice_size_min, practice_size_max, state, small_or_rural). The `active` schema grain is: (year, practice_size_min, practice_size_max, state, small_or_rural).

7.1.1.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id
- practice_size_min
 - type: int4 not null
 - description: The minimum value of the practice size range for the profile. In 2018, profiles have the following options for practice size ranges: 1-1 (Solo), 2-15 (Small), 16-99 (Medium), 100-null (Large).
- practice_size_max
 - type: int4 not null
 - description: The maximum value of the practice size range for the profile. In 2018, profiles have the following options for practice size ranges: 1-1 (Solo), 2-15 (Small), 16-99 (Medium), 100-null (Large).
- state
 - type: varchar(255) not null
 - description: The organization's state under which the clinician practices.
- small_or_rural

⁹ <https://confluence.cms.gov/display/QPPAR/Analytics+Portal+Score+Profile+POC#AnalyticsPortalScoreProfilePOC-ProfileswithThreeNewFields:State,SmallorRural,PracticeSizeSize>

- type: boolean not null
- description: Whether the organization (if scored under a group submission) or provider (if scored under an individual submission) has either rural_clinician=true or small_group_practitioner=true. (If a group submission, we look at the group_special_scenario flags to determine this; if an individual submission, we look at individual_special_scenario.)
- overall_score_mean
 - type: float8
 - description: The average overall score achieved by all clinicians within this profile.
- quality_score_mean
 - type: float8
 - description: The average unweighted quality category score achieved by all clinicians within this profile.
- pi_score_mean
 - type: float8
 - description: The average unweighted pi category score achieved by all clinicians within this profile.
- ia_score_mean
 - type: float8
 - description: The average unweighted ia category score achieved by all clinicians within this profile.
- cost_score_mean
 - type: float8
 - description: The average unweighted cost category score achieved by all clinicians within this profile.
- year
 - type: int4
 - description: The performance year corresponding to the profile and average score data.
- count_of_final_scores
 - type: int4
 - description: The number of distinct clinicians (TIN-NPI) who received an eligible final score and correspond to this profile.
 - Notes: The QPPAR PCF API endpoint will only return Peer Comparison Feedback results for the clinician if their profile's count_of_final_scores is greater than 20. This ensures that only statistically significant feedback is shown to a clinician.

7.1.2 final_score_score_profile

7.1.2.1 Description

A mapping table between the final_score table and the score_profile table, such that a single final_score is assigned to its respective score_profile.

Keys: The surrogate key is db_id. There is one foreign key to final_score via db_final_score_id and one foreign key to score_profile via db_score_profile_id.

7.1.2.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_final_score_id
 - type: int8 not null
 - description: Foreign key to the final_score table - final_score.db_id

- db_score_profile_id
 - type: int8 not null
 - description: Foreign key to the score_profile table - score_profile.db_id

8 Final Scoring, QIS

8.1 Data Sources

The information here is generated by an A&R process using the current performance year's Eligibility and the previous performance year's Submission Scoring data.

For more information: [Quality Improvement Score Tech Design](#)⁴⁰

8.1.1 Data generation processes

The sources of this data are the Eligibility of the current performance year and Final Scoring of the previous performance year. The pipeline, `quality_improvement_score`, which calculates the data is documented here: [Quality Improvement Score Pipeline](#)⁴¹

8.1.2 `quality_improvement_score`

8.1.2.1 Description

For all entities (TIN-NPI; TIN; APM entity; virtual group) that are eligible in a given performance year, this table lists their `quality_decile_score` from the previous performance year. See [Quality Improvement Score Tech Design](#)⁴² for more details.

Keys: The surrogate key is `db_id`. The natural key is `year`, `npi`, `tin`, `virtual_group_id`, `entity_id`.

8.1.2.2 Columns

- `db_id`
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- `year`
 - type: int2 not null
 - description: The Current Performance Year
- `prior_year`
 - type: int2 not null
 - description: The Prior Performance Year
- `tin`
 - type: varchar(255)
 - description: Populated if Group (TIN) is Eligible or TIN-NPI is Eligible in Current Performance Year and if this row represents the data for that TIN or TIN-NPI.
- `npi`
 - type: varchar(255)
 - description: Populated if individual (TIN/NPI) is Eligible in Current Performance Year and if this row represents the data for that TIN-NPI.

⁴⁰ <https://confluence.cms.gov/display/QPPAR/Quality+Improvement+Score+Tech+Design>

⁴¹ <https://confluence.cms.gov/display/QPPAR/Quality+Improvement+Score+Pipeline>

⁴² <https://confluence.cms.gov/display/QPPAR/Quality+Improvement+Score+Tech+Design>

- virtual_group_id
 - type: varchar(255)
 - description: Populated if Virtual Group is Eligible in Current Performance Year and if this row represents the data for that virtual group.
- entity_id
 - type: varchar(255)
 - description: Populated if APM Entity (APM_Entity_ID) is Eligible in Current Performance Year and if this row represents the data for that APM entity.
- total_decile_score
 - type: float8
 - description: The Aggregate Quality Decile Score for the Measurement Set picked for final score, or the quality_decile_score if due to targeted review/averaged participant score. Null if Data Insufficiency or No Final Score.
- denominator
 - type: float8
 - description: Denominator value for the Measurement Set picked for final score, or 100 if the quality_decile_score came from a targeted review/averaged participant score. Used when calculating the quality_decile_score. Null if Data Insufficiency or No Final Score.
- quality_decile_score
 - type: float8
 - description: The prior year’s quality decile score. 0-100 Based Quality Score. Null if Data Insufficiency or No Final Score.
- scenario_reason
 - type: varchar(255)
 - description: String with one of the following values:
 - “Individual Score Match” - The score values derive from the 2017 final score of the TIN-NPI (multiple entity_types possible)
 - “NPI Score Match via Alternate TIN” - The score values derive from a 2017 final score for the NPI, achieved under a different TIN. Multiple entity_types possible. Highest value for “quality_decile_score” chosen.
 - “Group Score Match” - The score values derive from the 2017 final score of the TIN (entity_type=’group’)
 - “Group Participant Averaged Scores” - The score values derive from all 2018 TIN-NPI participants associated with the group who received a 2017 final score (whether from a direct TIN-NPI match or from an NPI-under-a-different-TIN match); all participant quality_decile_scores averaged together for the group.
 - “APM Score Match” - The score values derive from the 2017 final score of the APM (entity_type=’apm’)
 - “APM Participant Averaged Scores” - The score values derive from all 2018 TIN-NPI participants associated with the APM who received a 2017 final score (only from a direct TIN-NPI match); all participant quality_decile_scores averaged together for the group.
 - “Virtual Group Score Match” - not possible for 2018
 - “Virtual Group Participant Averaged Scores” - The score values derive from all 2018 TIN-NPI participants associated with the group who received a 2017 final score (whether from a direct TIN-NPI match or from an NPI-under-a-different-TIN match); all participant quality_decile_scores averaged together for the group.
 - “Data Insufficiency” - A final score was found for the entity, but the total_decile_score did not exist (no quality data) or was 0.
 - “No Final Score” - No final score was found for the entity, or any of its participants.
- db_data_load_id
 - type: int8 not null
 - description: Foreign key to the data_load table - data_load.db_id
- final_score_db_data_load_id

- type: int8 not null
- description: Foreign key to the data_load table - data_load.db_id. This is the prior year's active Final Scoring data load.
- eligibility_db_data_load_id
 - type: int8 not null
 - description: Foreign key to the data_load table - data_load.db_id. This is the current performance year's active Eligibility data load.

9 Final Scoring, Payment Adjustment

9.1 Data Sources

The mapping of all possible score values to the associated payment adjustment data comes from OACT. In the 2018 performance year, A&R was provided an Excel file with the formulas from which to calculate the payment adjustment data for all scores. A&R generated a CSV file using those formulas, and loaded it to this table.

9.1.1 Data ingestion processes

The mini-pipeline to load the CSV to the UDS is documented here: [Final Scoring pipeline](#)⁴³

Background on why payment adjustments are provided in two different formats is here: [Payment Adjustment Map \("Matrix"\)](#)⁴⁴

9.1.2 final_score_payment_adjustment_map

9.1.2.1 Description

A mapping table that lists every possible overall_score (final score) for the performance year and that score's associated payment adjustment values. Payment adjustments are calculated by the Office of the Actuary based on the distribution of final scores for a given performance year. The payment adjustments for performance year N are calculated during program year N+1 and will be applied to the clinician's payments for year N+2.

Grains: The grain of the `uds` table is (db_data_load_id, overall_score). The grain of the `active` table is (year, overall_score).

9.1.2.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- year
 - type: int2 not null
 - description: The performance year of the payment adjustment data.
- db_data_load_id
 - type: int8 not null
 - description: Foreign key to the data_load table - data_load.db_id
- overall_score
 - type: float8 not null
 - description: The overall score. In 2017 and 2018, these are values between 0.00 and 100.00 with a precision of 0.01.
- total_adjustment_percentage
 - type: float8 not null

⁴³ https://confluence.cms.gov/display/QPPAR/Final+Scoring+pipeline#FinalScoringpipeline-Score_PaymentAdjustment_Map

⁴⁴ <https://confluence.cms.gov/pages/viewpage.action?pageId=274548341>

- description: The total adjustment percentage achieved by any clinician who received this record's overall_score within the specified performance year. This equals base_adjustment_percentage + exceptional_adjustment_percentage.
- exceptional_adjustment_percentage
 - type: float8 not null
 - description: The exceptional adjustment percentage achieved by any clinician who received this record's overall_score within the specified performance year. Only overall_scores of 70 or higher have exceptional adjustments.
- base_adjustment_percentage
 - type: float8 not null
 - description: The base adjustment percentage achieved by any clinician who received this record's overall_score within the specified performance year.

9.1.3 Tables to be deprecated

final_score_change - This table was meant to show various reasons why a final score changed between one load and the next. The data for this was never implemented. This table will be removed in a future version of the UDS.

change_reason - This table was meant to show various reasons why a final score could change between one load and the next. The data for this was never implemented. This table will be removed in a future version of the UDS.

10 Eligibility

10.1 Data Sources

The Eligibility domain is populated by two overlapping data sources.

The first, usually also called Eligibility, is processed from PECOS and claims.

The second source contains all APM membership data, and comes from the AMS team.

10.1.1 Data Source documentation

Eligibility API documentation: <http://dev.qpp.cms.gov/api/eligibility/docs/>

<https://qpp.cms.gov/api/eligibility/docs/>⁴⁵

AMS documentation:

Guidance for run 5 "ghost" clinicians in 2018: [Reporting on Ghost TIN-NPI with eligible final scores](#)⁴⁶

10.1.2 Data ingestion processes

The pipeline that loads all Eligibility and AMS data, UDS_Eligibility, is documented here: [Eligibility pipeline](#)⁴⁷

10.1.2.1 Associating an APM entity to its TIN-NPI clinicians

This data processing caveat addresses how we handle populating the `provider_apm_entity` table, which joins an APM entity to a TIN-NPI clinician. APM data from AMS like in the `apm_entity` table does NOT have a run concept. The `apm_entity` table lists only a year. So we have an APM entity, which doesn't have a run concept, and we have to associate it to a TIN-NPI, which does have a run concept. What we do here is create the association between the APM entity and the clinician for the highest non-ghost run available in the data. What does that mean?

- In 2017, run 3 was the highest run for that performance year.
- In 2018, run 4 was the highest non-ghost (run=5) run for that performance year.
- In 2019, currently run 3 is the highest available run for this performance year. We will get run=4 in the future.
- In 2020, currently run 1 is the highest available run for this performance year. We will get additional runs in the future.

So when we are processing an APM entity, and we need to join it to its clinician in `provider_apm_entity`, we don't join it to every single run for the TIN-NPI. We only join it for the highest run. (Of course in the case of no claims APM participants, the run that we join to may be run=0 for any of these).

⁴⁵ https://qpp.cms.gov/api/eligibility/docs/#/V3/get_api_eligibility_npi__npi_

⁴⁶ <https://confluence.cms.gov/display/QPPAR/Reporting+on+Ghost+TIN-NPI+with+eligible+final+scores>

⁴⁷ <https://confluence.cms.gov/display/QPPAR/Eligibility+pipeline>

10.2 Tables

10.2.1 apm_entity

10.2.1.1 Description

Represents an APM entity for a given performance year. (Run concept does NOT apply.)

Keys: The surrogate key is db_id. The natural key is year, entity_id.

10.2.1.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- year
 - type: int2 not null
 - description: The performance year.
- entity_id
 - type: varchar(255) not null
 - description: Unique combination of characters, numbers or symbols that are assigned by CMS Model operations to support establishing the identity of an organization participating in the APM Program.
- entity_name
 - type: varchar(255) not null
 - description: Official name of the APM entity.
- entity_tin
 - type: varchar(255)
 - description: The 9-digit identifier of the entity (Taxpayer Identification Number).
 - This might be deprecated in 2019.
- subdivision_id
 - type: varchar(255)
 - description: Identifier for a specific grouping within the APM Program for QPP purposes.
- subdivision_name
 - type: varchar(255)
 - description: Formal label by which the QPP subdivision is identified.
- apm_id
 - type: varchar(255) not null
 - description: Unique combinations of characters, numbers or symbols that are assigned by CMS Model operations to support establishing the identity of an APM Program.
 - 08 == SSP (Shared Savings Program)
 - etc....
- apm_name
 - type: varchar(255) not null
 - description: Formal label by which the APM Program is identified.
- lvt_flag
 - type: bool
 - description: Indicates if the APM Entity is below the low volume threshold.
- lvt_payments

- type: float8
 - description: Total aggregated expenditures of the APM Entity.
- lvt_patients
 - type: int8
 - description: Total number of unique patients of the APM Entity.
- advanced_apm_flag
 - type: bool
 - description: APMs that require MIPS comparable measures, use of CEHRT, and more than nominal downside risk. Eligible clinicians that participate in Advanced APMs are able to become QPs and receive a 5% bonus and be exempt from MIPS.
- mips_apm_flag
 - type: bool
 - description: APMs that already assess participants on cost and quality under the terms of the APM. This flag is used to designate "MIPS APMs" which are the ones who get scored during the QPP program year.
- db_data_load_id
 - type: int8
 - description: Foreign key to data_load.db_id
- lvt_small_status
 - type: bool
 - description: Indicates if the APM Entity has Small Practice status
- lvt_performance_year
 - type: int2
 - description: The relevant year for the APM dataset. Basically the same thing as year.
- final_qpc_score
 - type: float8
 - description: ??
- complex_patient_score
 - type: float8
 - description:
- extreme_hardship
 - type: bool
 - description: True if Extreme and Uncontrollable Circumstance Exemption credit has been received.
- is_opted_in
 - type: bool
 - description: Indicates that the APM Entity has chosen to participate in the QPP program, even though they are under the low volume threshold (lvt).

10.2.2 apm_entity_organization

10.2.2.1 Description

This table is a join table consisting of foreign keys to other tables.

Keys:

10.2.2.2 Columns

- db_apm_entity_id
 - type: int8 not null
 - description: Foreign key to the apm_entity table - apm_entity.db_id.

- db_organization_id
 - type: int8 not null
 - description: Foreign key to the organization table - organization.db_id.
- db_data_load_id
 - type: int8
 - description: Foreign key to data_load.db_id

10.2.3 provider_apm_entity

10.2.3.1 Description

Association of a TIN-NPI clinician to an APM entity for a given performance year. Any TIN-NPI that are linked together under an APM will also be linked in the provider_org table. Links are only created for the highest available non-ghost run for the clinician.

Keys:

10.2.3.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_apm_entity_id
 - type: int8 not null
 - description: An internal surrogate key for entity_id, year. Foreign key to the apm_entity table - apm_entity.db_id.
- db_provider_id
 - type: int8 not null
 - description: An internal surrogate key for npi, year, and run. Foreign key to the provider table - provider.db_id
- db_organization_id
 - type: int8 not null
 - description: An internal surrogate key for tin, year, and run. Foreign key to the organization table - organization.db_id.
- qp_patient_threshold_score
 - type: int8
 - description: Threshold score, as a truncated percentage, based on number of patients seen.
- qp_payment_threshold_score
 - type: int8
 - description: Threshold score, as a truncated percentage, based on expenditures.
- qp_score_type
 - type: varchar(2)
 - description: Now Deprecated, contains what is stored in the equivalent API response field
- qp_payment_scores_me
 - type: float8
 - description: Parsed out score value from JSON response
 - ME = Medicare Entity (includes CJR, etc.)
- qp_payment_scores_mi
 - type: float8
 - description: Parsed out score value from JSON response

- MI = Medicare Individual
- qp_payment_scores_ae
 - type: float8
 - description: Parsed out score value from JSON response
 - AE = All-Payer Entity
- qp_payment_scores_at
 - type: float8
 - description: Parsed out score value from JSON response
 - AT = All-Payer TIN
- qp_payment_scores_ai
 - type: float8
 - description: Parsed out score value from JSON response
 - AI = All-Payer NPI
- qp_patient_scores_me
 - type: float8
 - description: Parsed out score value from JSON response
 - ME = Medicare Entity (includes CJR, etc.)
- qp_patient_scores_mi
 - type: float8
 - description: Parsed out score value from JSON response
 - MI = Medicare Individual
- qp_patient_scores_ae
 - type: float8
 - description: Parsed out score value from JSON response
 - AE = All-Payer Entity
- qp_patient_scores_at
 - type: float8
 - description: Parsed out score value from JSON response
 - AT = All-Payer TIN
- qp_patient_scores_ai
 - type: float8
 - description: Parsed out score value from JSON response
 - AI = All-Payer NPI
- provider_relationship_code
 - type: char(1)
 - description: Code that identifies the relationship of the provider with the APM Entity. For NextGen ACO's an indication of a preferred provider.
 - P - Provider is designated as preferred. This means they are a full APM participant and are eligible to be scored under that APM.
 - S - Provider is designated as secondary. This means they are not scored under the APM, though they may separately receive a MIPS group or individual score.
 - F - NextGen Preferred Provider. I (Jenni) have no idea what this means in terms of scoring.
- db_data_load_id
 - type: int8
 - description: Foreign key to data_load.db_id

10.2.4 organization

10.2.4.1 Description

This table represents an organization consisting of one or more providers that are collected under a Taxpayer Identification Number. A given organization can have one group submission per performance year. The size of the organization depends on how many providers are associated to it.

Keys: The surrogate key is db_id. The natural key is year, run, tin.

10.2.4.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- tin
 - type: varchar(255) not null
 - description: The 9-digit identifier of the organization (Taxpayer Identification Number). Sometimes this may be a SSN or EIN.
 - PII field.
- run
 - type: int2 not null
 - description: Eligibility is reviewed from two different time periods of claims for the performance year (Run 1 or 2). Runs 3 and 4 reconciles the data between the two runs at the end of the performance year. Run 0 indicates there is no special scenario information for the organization, and that it is associated to one or more APMs. Run 5 is a final eligibility determination period covering the final four months of the performance year. See "Definition of run parameter" below for more details.
- year
 - type: int2 not null
 - description: The performance year.
- name
 - type: varchar(255)
 - description: The name of the organization.
- address_line_one
 - type: varchar(255)
 - description: The address of the organization (line 1).
- address_line_two
 - type: varchar(255)
 - description: The address of the organization (line 2).
- city
 - type: varchar(255)
 - description: The city of the organization's address.
- zip
 - type: varchar(255)
 - description: The zip code of the organization's address.
- state
 - type: char(2)
 - description: The two-letter abbreviation of the U.S. state of the organization's address.
- db_data_load_id

- type: int8
- description: Foreign key to data_load.db_id
- eligibility_response_npi_count
 - type: int4
 - description: The number of NPIs associated to the TIN according to the Eligibility API. Warning: This may not account for NPIs that are associated to the TIN through an APM where no special scenario data exists for the pair.

10.2.5 provider

10.2.5.1 Description

This table represents a care provider (clinician). A record is uniquely determined by the combination of National Provider Identifier, year, and run for a single db_data_load_id.

Keys: The surrogate key is db_id. The natural key is year, run, npi.

10.2.5.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- npi
 - type: varchar(255) not null
 - description: The 10-digit identifier of the provider (National Provider Identifier).
- year
 - type: int2 not null
 - description: The performance year.
- run
 - type: int2 not null
 - description: Eligibility is reviewed from two different time periods of claims for the performance year (Run 1 or 2). Runs 3 and 4 reconciles the data between the two runs at the end of the performance year. Run 0 indicates there is no special scenario information for the organization, and that it is associated to one or more APMs. Run 5 is a final eligibility determination period covering the final four months of the performance year. See "Definition of run parameter" below for more details.
- national_provider_identifier_type
 - type: int2
 - description: This is a numeric code describing the type of provider that is represented by the National Provider Identifier (NPI). Possible values are "individual" or "organization". When someone talks about "Type 2 providers" they are referring to this field.
- first_approved_date
 - type: date
 - description: The start date of the Medicare Enrollment. This is the first date for which the clinician submitted a valid Medicare claim. Warning: This can change between performance years. No, it doesn't make sense. But this is due to upstream data sets (PECOS).
- years_in_medicare
 - type: int2
 - description: The number of years between the first_approved_date and the performance year.
- pecos_enrollment_date
 - type: int2

- description: Date of enrollment into Provider Enrollment, Chain and Ownership System (PECOS).
- newly_enrolled
 - type: bool
 - description: Boolean value indicating whether a clinician (NPI) enrolled in Medicare on or after the start of program year (i.e., January 1st, 2017 for program year 1, 2017). If this flag is true, it means the provider isn't eligible for final scoring and payment adjustment. Even if a clinician is run 0 (no claims APM participant, thus no special scenario data), we try to populate this field for as many providers as possible.
- first_name
 - type: varchar(255)
 - description: The first name of the provider.
- middle_name
 - type: varchar(255)
 - description: The middle name of the provider.
- last_name
 - type: varchar(255)
 - description: The last name of the provider.
- specialty_description
 - type: varchar(255)
 - description: Type of specialty, e.g. Psychiatry. The specialty code that matches this description determines if the clinician is "MIPS eligible" (aka individual_special_scenario.mips_eligibility_switch). **Warning: The specialty_code on the fields on the individual_special_scenario table can disagree with this description, because specialty is truly at the TIN-NPI level, but only one specialty description is provided per NPI. This needs to be fixed in a future effort in coordination with the Eligibility team.**
- specialty_category
 - type: varchar(255)
 - description: Category of provider, e.g. Physicians. See notes for specialty_description.
- specialty_type
 - type: varchar(255)
 - description: Type of provider, e.g. Doctor of Medicine. See notes for specialty_description.
- qp_status
 - type: char(1)
 - description: The QP Status Code dictates if a Score should be used or not used. This is an AMS / APM field.
 - QP=Y: Qualifying APM Participant, QP status, Does not report, exempt from QPP program.
 - QP=Q: Treat as QP=Y (Does not Report).
 - QP=N: Not exempt, should receive payment adjustment based on submission. Reports, if no submission receive score of 0.
 - QP=P: Treat as QP=N, Partial QP (Reports, if no submission, receive score of 0).
 - QP=R: If reported (at any level group or individual) then score (follow QP=N), If does not report, exempt (follow QP=Y) // if reported include, if no submission exclude
- db_data_load_id
 - type: int8
 - description: Foreign key to data_load.db_id
- is_maqi
 - type: bool
 - description: Boolean indicator, True if clinician has been granted a MAQI Demonstration Waiver. This means they are not eligible for a final score or payment adjustment.
- qp_score_type
 - type: varchar(255) Indicator that score is Entity or Individual. An AMS / APM field.
 - Starting in 2019:
 - ME = Medicare Entity (includes CJR, etc.)

- MI = Medicare Individual
- AE = All-Payer Entity
- AT = All-Payer TIN
- AI = All-Payer NPI
- ams_mips_eligible_clinician
 - type: bool
 - description: MIPS Eligible Clinician (EC) Indicator. This is a flag from AMS / APM data indicating that APM considers the provider eligible for the MIPS program. It does NOT trump other data.
- individual_qp_payment_threshold_score
 - type: float8
 - description:
- individual_qp_patient_threshold_score
 - type: float8
 - description:

10.2.6 provider_org

10.2.6.1 Description

This is a join table that links together an organization (TIN) with one of its providers (NPI). Any TIN-NPI clinician association, whether via claims or APMs, is present here.

Keys: The surrogate key is db_id.

10.2.6.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_provider_id
 - type: int8 not null
 - description: Foreign key to the provider table - provider.db_id.
- db_organization_id
 - type: int8 not null
 - description: Foreign key to the organization table - organization.db_id.
- db_data_load_id
 - type: int8
 - description: Foreign key to data_load.db_id

10.2.7 group_special_scenario

10.2.7.1 Description

The special scenario eligibility data that applies to an organization.

Keys: The surrogate key is db_id.

10.2.7.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_organization_id
 - type: int8 not null
 - description: Foreign key to the organization table - organization.db_id.
- scenario_type
 - type: char(10)
 - description: Whether switches apply to group or individual. Permissible values: "group"
- mips_eligibility_switch
 - type: bool
 - description: Identifies if one or more clinicians (NPI) in the Group has a 'Specialty_code' eligible to participate in MIPS.
- low_volume_switch
 - type: bool
 - description: Boolean value indicating whether a provider or group meets the criteria for below low volume threshold. 2017 Performance Year: Low volume threshold = < 30,000\$ of Part B Medicare Allowance billing charges during performance period OR < 100 Part B enrolled Medicare beneficiaries. 2018 Performance Year: Low volume threshold = < 90,000\$ of Part B Medicare Allowance billing charges during performance period OR < 200 Part B enrolled Medicare beneficiaries.
- non_patient_facing
 - type: bool
 - description: Boolean indicator of whether patient-facing encounters met the encounter threshold.
- small_group_practitioner
 - type: bool
 - description: Boolean indicator of whether there were 15 or fewer clinicians billed under the group. TRUE = 15 or fewer clinicians bill under the group. FALSE = More than 15 clinicians bill under the group.
- rural_clinician
 - type: bool
 - description: Boolean indicator of whether the clinician has at least one claim with a zip code designated as a rural area. TRUE = Clinician has at least one claim with a zip code designated as rural. FALSE = Clinician does not have at least one claim with a zip code designated as rural.
- hpsa_clinician
 - type: bool
 - description: ACI hardship flag
- hospital_based_clinician
 - type: bool
 - description: Boolean indicator of whether a clinician met the threshold of hospital-based expenditures.
- ambulatory_surgical_center
 - type: bool
 - description: Ambulatory Surgical Centers (ASCs) provide surgery without an overnight hospital stay. When this field appears as part of a group scenario, it indicates that all providers within this group work within designated ASCs.
- allowed_charges
 - type: float8
 - description: The volume of allowed charges
- beneficiary_count

- type: int4
 - description: Number of patients used to determine low volume threshold status.
- aci_hardship
 - type: bool
 - description: Boolean indicator of whether there is an approved ACI Hardship Exemption application.
- ia_study
 - type: bool
 - description: Boolean indicator if Improvement Activities Study data is present
- extreme_hardship
 - type: bool
 - description:
- aci_reweighting
 - type: bool
 - description: Reweighting of ACI score
- low_vol_status_reason_code
 - type: varchar(255)
 - description: If lowVolumeSwitch is true, this will contain the reason for the status. Possible values- "BENE" Clinician's number of unique beneficiaries fell below the low volume threshold. "CHRG" Clinician's Part B expenditures fell below the low volume threshold. "SRVC" Clinician's low volume services fell below the low volume threshold. "BOTH" Both the clinician's number of unique beneficiaries and Part B expenditures fell below the low volume threshold. "BENE/SRVC" Both the clinician's number of unique beneficiaries and low volume services fell below the low volume threshold. "CHRG/SRVC" Both the clinician's Part B expenditures and low volume services fell below the low volume threshold. "ALL" Clinician's number of unique beneficiaries, Part B expenditures, and low volume services fell below the low volume threshold
- low_vol_status_reason_description
 - type: varchar(255)
 - description: Description of low_vol_status_reason_code
- low_volume_services
 - type: int4
 - description: Number of claim lines counted toward low volume service threshold
- db_data_load_id
 - type: int8
 - description: Foreign key to data_load.db_id
- hospital_vbp_ccn
 - type: varchar(255)
 - description: CMS Certification Number assigned by the Hospital Value-Based Purchasing Program
- hospital_vbp_name
 - type: varchar(255)
 - description: Facility name within the Hospital Value-Based Purchasing Program
- hospital_vbp_score
 - type: float8
 - description: Hospital Value-Based Purchasing Program score
- is_facility_based
 - type: bool
 - description: Boolean indicator, True if the threshold of facility based encounters is met
- complex_patient_score
 - type: float8
 - description:
 - description: Composite Complex-Patient Score calculated by a combination of the dual eligibility ratio and the Hierarchical Conditions Category (HCC) risk score; max 5.00, min 0.00.
- facility_based_npis
 - type: int4

- description: [Number of?] npis based in facility
- `hcc_risk_score`
 - type: float8
 - description:
 - description: Average Hierarchical Conditions Category (HCC) risk score of beneficiaries. HCC risk score is a component of complex patient score; max 5.00, min 0.00.
- `dual_eligibility_ratio`
 - type: float8
 - description:
 - description: Ratio of unique patients who have dual eligible status seen by the provider among all unique patients seen during run 2; max 1.00, min 0.01.
- `is_eligible_group`
 - type: bool
 - description: Boolean indicator of group eligibility for the TIN
- `extreme_hardship_reasons_quality`
 - type: bool
 - description: True if extreme hardship exemption due to quality
- `extreme_hardship_reasons_ia`
 - type: bool
 - description: True if extreme hardship exemption due to improvement activities
- `extreme_hardship_reasons_aci`
 - type: bool
 - description: True if extreme hardship exemption due to aci/pi
- `extreme_hardship_reasons_cost`
 - type: bool
 - description: True if extreme hardship exemption due to cost

10.2.8 individual_special_scenario

10.2.8.1 Description

The special scenario eligibility data that applies to a single TIN-NPI clinician. Note that not all clinicians who are eligible for the QPP program will have a record here, because there are APM participants without claims, and this data is derived from claims.

Keys: The surrogate key is `db_id`. The grain is `db_provider_org_id` (a single TIN-NPI-run-year-data load).

10.2.8.2 Columns

- `db_id`
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- `db_provider_org_id`
 - type: int8 not null
 - description: Foreign key to the provider_org table - provider_org.db_id.
- `scenario_type`
 - type: char(10) not null
 - description: Whether switches apply to group or individual. Permissible values: "individual". This field is a hold-over from when we used to store group-level and individual-level special scenario data in the same table.

- aggregation_level
 - type: int2 not null
 - description: 1 = "TIN level", 2 = "NPI/TIN level", 3 = "Other level". Note: 1 and 2 correspond to scenario_type.
- mips_eligibility_switch
 - type: bool not null
 - description: Identifies if the clinician (TIN-NPI) has a 'Specialty_code' eligible to participate in MIPS. Note that while provider.specialty_... fields are only present in the UDS at the NPI level, specialty truly applies at the TIN-NPI level, meaning that a single NPI could have two different specialties under two different TINs. Since this switch is at the TIN-NPI level, it is correctly accounting for this multiple specialty-per-NPI scenario.
- low_volume_switch
 - type: bool
 - description: Boolean value indicating whether a provider or group meets the criteria for below low volume threshold.
 - 2017 Performance Year: Low volume threshold = < 30,000\$ of Part B Medicare Allowance billing charges during performance period OR < 100 Part B enrolled Medicare beneficiaries.
 - 2018 Performance Year: Low volume threshold = < 90,000\$ of Part B Medicare Allowance billing charges during performance period OR < 200 Part B enrolled Medicare beneficiaries.
 - 2019 Performance Year: TBD
- non_patient_facing
 - type: bool
 - description: Boolean indicator of whether patient-facing encounters met the encounter threshold.
- small_group_practitioner
 - type: bool
 - description: Boolean indicator of whether there were 15 or fewer clinicians billed under the group. TRUE = 15 or fewer clinicians bill under the group. FALSE = More than 15 clinicians bill under the group.
- rural_clinician
 - type: bool
 - description: Boolean indicator of whether the clinician has at least one claim with a zip code designated as a rural area. TRUE = Clinician has at least one claim with a zip code designated as rural. FALSE = Clinician does not have at least one claim with a zip code designated as rural.
- hpsa_clinician
 - type: bool
 - description:
- hospital_based_clinician
 - type: bool
 - description: Boolean indicator of whether a clinician met the threshold of hospital-based expenditures.
- ambulatory_surgical_center
 - type: bool
 - description: Ambulatory Surgical Centers (ASCs) provide surgery without an overnight hospital stay. When this field appears as part of a group scenario, it indicates that all providers within this group work within designated ASCs.
- allowed_charges
 - type: float8
 - description: The volume of allowed charges
- beneficiary_count
 - type: int4
 - description: Number of patients used to determine low volume threshold status.
- aci_hardship
 - type: bool

- description: Boolean indicator of whether there is an approved ACI Hardship Exemption application.
- extreme_hardship
 - type: bool
 - description: Boolean indicator of whether Extreme and Uncontrollable Circumstance Exception credit has been received.
- ia_study
 - type: bool
 - description: Boolean indicator if Improvement Activities Study data is present
- aci_reweighting
 - type: bool
 - description: Reweighting of ACI score
- payment_adjustment_ccn
 - type: varchar(255)
 - description: CMS Certification Number (based on CAH II claims), used for payment adjustment purposes
- payment_adjustment_npi
 - type: varchar(10)
 - description: NPI for the associated facility (based on CAH II claims), used for payment adjustment purposes
- hospital_vbp_name
 - type: varchar(255)
 - description: Facility name within the Hospital Value-Based Purchasing Program
- hospital_vbp_ccn
 - type: varchar(255)
 - description: CMS Certification Number assigned by the Hospital Value-Based Purchasing Program
- hospital_vbp_score
 - type: float8
 - description: Hospital Value-Based Purchasing Program score
- low_vol_status_reason_code
 - type: varchar(255)
 - description: If lowVolumeSwitch is true, this will contain the reason for the status. Possible values- "BENE" Clinician's number of unique beneficiaries fell below the low volume threshold. "CHRG" Clinician's Part B expenditures fell below the low volume threshold. "SRVC" Clinician's low volume services fell below the low volume threshold. "BOTH" Both the clinician's number of unique beneficiaries and Part B expenditures fell below the low volume threshold. "BENE/SRVC" Both the clinician's number of unique beneficiaries and low volume services fell below the low volume threshold. "CHRG/SRVC" Both the clinician's Part B expenditures and low volume services fell below the low volume threshold. "ALL" Clinician's number of unique beneficiaries, Part B expenditures, and low volume services fell below the low volume threshold
- low_vol_status_reason_description
 - type: varchar(255)
 - description: Description of low_vol_status_reason_code
- low_volume_services
 - type: int4
 - description: Number of claim lines counted toward low volume service threshold
- db_data_load_id
 - type: int8
 - description: Foreign key to data_load.db_id
- is_facility_based
 - type: bool
 - description: Boolean indicator, True if the threshold of facility based encounters is met
- complex_patient_score
 - type: float8

- description:
- description: Composite Complex-Patient Score calculated by a combination of the dual eligibility ratio and the Hierarchical Conditions Category (HCC) risk score; max 5.00, min 0.00.
- dual_eligibility_ratio
 - type: float8
 - description: Ratio of unique patients who have dual eligible status seen by the provider among all unique patients seen during run 2; max 1.00, min 0.01.
- facility_encounters_percent
 - type: float8
 - description: Quotient obtained by dividing the count of facility-based encounters by total number of encounters and rounding to the hundreths place.
- hcc_risk_score
 - type: float8
 - description: Average Hierarchical Conditions Category (HCC) risk score of beneficiaries. HCC risk score is a component of complex patient score; max 5.00, min 0.00.
- eligibility_scenario
 - type: int4
 - description: Numeric code for the MIPS reporting scenario from the eligibility rules grid
- is_eligible_individual
 - type: bool
 - description: Boolean indicator of individual eligibility for the TIN-NPI
- is_eligible_group
 - type: bool
 - description: Boolean indicator of group eligibility for the TIN-NPI
- is_eligible_mips_apm
 - type: bool
 - description: Boolean indicator of MIPS APM eligibility for the TIN-NPI
- is_eligible_virtual_group
 - type: bool
 - description: Boolean indicator of virtual group eligibility for the TIN-NPI
- extreme_hardship_reasons_quality
 - type: bool
 - description: True if extreme hardship exemption due to quality
- extreme_hardship_reasons_ia
 - type: bool
 - description: True if extreme hardship exemption due to improvement activities
- extreme_hardship_reasons_aci
 - type: bool
 - description: True if extreme hardship exemption due to aci/pi
- extreme_hardship_reasons_cost
 - type: bool
 - description: True if extreme hardship exemption due to cost
- specialty_code
 - type: string
 - description: Unique identifier indicating the specialty of the clinician. This code determines the clinician's mips_eligibility_switch value. Warning: The specialty fields on the provider table can disagree with this code, because specialty is truly at the TIN-NPI level, but only one specialty description is provided per NPI.

10.2.9 virtual_group

10.2.9.1 Description

Represents a virtual group and its associated special scenario eligibility data.

Keys: The surrogate key is db_id. The natural key is year, run, virtual_group_id

10.2.9.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- virtual_group_id
 - type: varchar(255) not null
 - description: The identifier for the Virtual Group.
- year
 - type: int2 not null
 - description: The performance year.
- run
 - type: int2 not null
 - description: Eligibility is reviewed from two different time periods of claims for the performance year (Run 1 or 2). Runs 3 and 4 reconciles the data between the two runs at the end of the performance year. Run 0 indicates there is no special scenario information for the organization, and that it is associated to one or more APMs. Run 5 is a final eligibility determination period covering the final four months of the performance year. See "Definition of run parameter" below for more details.
- claims_type
 - type: varchar(255)
 - description: Type of claims made by this Virtual Group.
- low_volume_switch
 - type: bool
 - description: Boolean indicator whether the virtual group fell below the low volume threshold
- beneficiary_count
 - type: int4
 - description: Number of patients used to determine low volume threshold status.
- allowed_charges
 - type: float8
 - description: Billing amount used to determine low volume threshold status
- non_patient_facing
 - type: bool
 - description: Boolean indicator of whether patient-facing encounters met the encounter threshold.
- rural_clinician
 - type: bool
 - description: Boolean indicator of whether the clinician has at least one claim with a zip code designated as a rural area. TRUE = Clinician has at least one claim with a zip code designated as rural. FALSE = Clinician does not have at least one claim with a zip code designated as rural.
- hpsa_clinician
 - type: bool
 - description:

- hospital_based_clinician
 - type: bool
 - description: Clinician met the threshold of hospital based expenditures
- ambulatory_surgical_center
 - type: bool
 - description: Ambulatory Surgical Centers (ASCs) provide surgery without an overnight hospital stay. When this field appears as part of a groupScenario it indicates that all providers within this group work within designated ASCs.
- aci_hardship
 - type: bool
 - description: Boolean indicator if there is an approved ACI Hardship Exemption application
- ia_study
 - type: bool
 - description: Boolean indicator if Improvement Activities Study data is present
- extreme_hardship
 - type: bool
 - description: Boolean indicator if Extreme and Uncontrollable Circumstance Exemption credit has been received
- low_volume_services
 - type: int4
 - description: Number of claim lines counted toward low volume service threshold
- low_vol_status_reason_code
 - type: varchar(255)
 - description: If lowVolumeSwitch is true, this will contain the reason for the status. Possible values- "BENE" Clinician's number of unique beneficiaries fell below the low volume threshold. "CHRG" Clinician's Part B expenditures fell below the low volume threshold. "SRVC" Clinician's low volume services fell below the low volume threshold. "BOTH" Both the clinician's number of unique beneficiaries and Part B expenditures fell below the low volume threshold. "BENE/SRVC" Both the clinician's number of unique beneficiaries and low volume services fell below the low volume threshold "CHRG/SRVC" Both the clinician's Part B expenditures and low volume services fell below the low volume threshold "ALL" Clinician's number of unique beneficiaries, Part B expenditures, and low volume services fell below the low volume threshold
- low_vol_status_reason_description
 - type: varchar(255)
 - description: Description of low_vol_status_reason_code
- db_data_load_id
 - type: int8
 - description: Foreign key to data_load.db_id
- small_group_practitioner
 - type: bool
 - description: Boolean indicator, true if 15 or fewer clinicians bill under the group
- hospital_vbp_ccn
 - type: varchar(255)
 - description: CMS Certification Number assigned by the Hospital Value-Based Purchasing Program
- hospital_vbp_name
 - type: varchar(255)
 - description: Facility name within the Hospital Value-Based Purchasing Program
- hospital_vbp_score
 - type: float8
 - description: Hospital Value-Based Purchasing Program score
- is_facility_based
 - type: bool
 - description: Boolean indicator, True if the threshold of facility based encounters is met

- `complex_patient_score`
 - type: float8
 - description: Composite Complex-Patient Score calculated by a combination of the dual eligibility ratio and the Hierarchical Conditions Category (HCC) risk score; max 5.00, min 0.00.
- `facility_based_npis`
 - type: int4
 - description: [Number of?] npis based in facility
- `hcc_risk_score`
 - type: float8
 - description: Average Hierarchical Conditions Category (HCC) risk score of beneficiaries. HCC risk score is a component of complex patient score; max 5.00, min 0.00.
- `dual_eligibility_ratio`
 - type: float8
 - description: Ratio of unique patients who have dual eligible status seen by the provider among all unique patients seen during run 2; max 1.00, min 0.01.
- `extreme_hardship_reasons_quality`
 - type: bool
 - description: True if extreme hardship exemption due to quality
- `extreme_hardship_reasons_ia`
 - type: bool
 - description: True if extreme hardship exemption due to improvement activities
- `extreme_hardship_reasons_aci`
 - type: bool
 - description: True if extreme hardship exemption due to aci/pi
- `extreme_hardship_reasons_cost`
 - type: bool
 - description: True if extreme hardship exemption due to cost

10.2.10 virtual_group_organization

10.2.10.1 Description

This table is a join table consisting of foreign keys to other tables. When an organization is linked to a virtual group, all providers under that organization are also linked under the virtual group. This is different to APM membership, where some providers under an organization may be part of the APM, and others may not be.

Keys: The surrogate key is `db_id`.

10.2.10.2 Columns

- `db_id`
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- `db_virtual_group_id`
 - type: int8 not null
 - description: Foreign key to `virtual_group.db_id`
- `db_organization_id`
 - type: int8 not null
 - description: Foreign key to `organization.db_id`
- `db_data_load_id`

- type: int8
- description: Foreign key to data_load.db_id

10.2.11 state_region

10.2.11.1 Description

A metadata table detailing each state, and which CMS TA / SURS / QIO regions it corresponds to. Unlike most tables, this table does not correspond to a data load (no snapshots) nor does the data change between performance years.

Keys: The surrogate key is state_code.

10.2.11.2 Columns

- state_code
 - type: varchar(2) not null
 - description: The two-letter code corresponding to the state, i.e., "CA" for California.
- state_code_iso_3166
 - type: varchar(5) not null
 - description: The two-letter code corresponding to the state in ISO 3166 format, i.e. "US-CA" for California.
- state_name
 - type: varchar(255) not null
 - description: The name of the state, i.e., "California".
- cms_ta
 - type: int4 not null
 - description: A numeric value representing the CMS Technical Assistance region to which the state belongs. These are Regional Offices where field staff work closely with beneficiaries, health care providers, state governments, CMS contractors, community groups and others to provide education and address questions. Currently available CMS Regions are **Region 1 through Region 10**.
- qpp_surs
 - type: varchar(255)
 - description: A string indicating the QPP SURS region to which the state belongs. The Technical Assistance offered by QPP-SURS is free-of-charge to small practices of 15 or fewer ECs, with a focus on those located in medically underserved areas, health professional shortage areas, and rural locations. This data will move shortly to its own table, associated to individual clinicians.
- qin_qio
 - type: varchar(255)
 - description: A string indicating the QIN QIO to which the state belongs. QIN-QIOs are designed to bring together Medicare beneficiaries, physicians, and communities in data-driven initiatives that improve patient safety, make communities healthier, provide better coordinated post-hospital care, and enhance clinical quality. QIN-QIOs offer technical assistance, as well as tailored education, best practices, and tools and resources. QIN-QIOs help provide skills for transforming practices, including employing lean methodologies, assisting Medicare physicians with the transition to the QPP, and developing innovative approaches to quality improvement.

10.2.12 tin_npi_ccn_mapping

10.2.12.1 Description

A temporary mapping table used for 2017 performance year data.

10.2.12.2 Columns

- tin
 - type: varchar(255)
 - description: The 9-digit identifier of the organization (Taxpayer Identification Number). Sometimes this may be a SSN or EIN.
- npi
 - type: varchar(255)
 - description: The 10-digit identifier of the provider (National Provider Identifier).
- provider_ccn
 - type: varchar(255)
- facility_npi
 - type: varchar(255)
- r1_claims_type
 - type: varchar(255)
 - description: Run 1 type: CARR = Physician claims only. CAH2 = CAH II claims only. BOTH = Physician & CAH II claims.
- r2_claims_type
 - type: varchar(255)
 - description: Run 2 type: CARR = Physician claims only. CAH2 = CAH II claims only. BOTH = Physician & CAH II claims.
- multi_ccn
 - type: varchar(255)
 - description: Has multiple CCNs associated with the TIN/ NPI?
- file_date
 - type: timestamp
 - description: Date the file was created
- year
 - type: int4
 - description: The performance year
- run
 - type: varchar(255)

10.3 Definition of run parameter

The 'run' parameter applies to three different Eligibility entities: *organizations*, *providers*, and *virtual groups*. It represents how eligibility data of that entity was calculated, generally according to a certain period of time from claims data (see table below). The 'run' parameter is provided as part of the source data of these entities.

One complication is due to APMs (Advanced Payment Models). APM entities do not have a 'run' parameter, but they have participants with certain other eligibility information, so the APM data is 'constant' for a given year. (Note: APM data are indeed generated via multiple 'determination periods' but this information is not exposed to the UDS.) Thus an APM entity can be associated to an *organization* via a TIN and a *provider* via an NPI. In the UDS, we try to combine overlapping information wherever possible. So if a TIN and NPI appears in run=4 with individual special

scenario data, and that TIN-NPI is listed as participating in an APM, the organization row with run=4 and the provider row with run=4 are associated to the apm_entity row in the *provider_apm_entity* join table.

However, there are cases where APMs have participants that are not present at all in the claims dataset. To maintain a record of these participants, they are stored in the *organization* and *provider* tables with run=0. Thus run=0 is present only within the UDS, and indicates that the organization or provider is only available with APM-provided eligibility information.

One cannot assume, however, that run=0 means "all the APM participants." First as stated above, if the TIN-NPI participant has eligibility flags due to claims in (for example) run=4, that data will be associated to the APM (in other words, a run=0 row is only generated when no other match on an existing run is possible). Further, there may be *some* information present about the participant from claims, but not everything. For example, a provider (NPI) may have submitted claims and be eligible in the Reconciled run=4, but it did not submit claims under a certain organization (TIN). In that case, there would be no individual special scenario flags available for the TIN-NPI, as those are specific to a TIN-NPI pairing. Yet, the TIN-NPI may still be paired together in the *provider_org* join table, as the provider may participate under an APM through the TIN.

The two sources of Eligibility information are moving toward a unified dataset of both from-claims and from-APM clinicians. When this occurs, run=0 will be deprecated--but the lack of eligibility flags in certain cases can be expected to continue, if the organization or the provider-org lacks claims.

run	Short Description	Time Period for Calculations	Used in Final Scoring?
0	No claims available for the entity; APM information only		Yes
1	Run 1 from Claims	September 1 (201X - 2) through August 31 (201X - 1) for year 201X	Not directly
2	Run 2 from Claims	September 1 (201X - 1) through August 31 201X for year 201X	Not directly
3	Reconciliation of runs 1 and 2	Reconciliation is performed to adjust to provide clinician favorable status. If ineligible in Run 1 or 2, then Ineligible overall.	No
4	Reconciled (3) w/ Complex Patient Data	The same as Run 3 Reconciled with the addition of Complex Patient Data. This is the main dataset that is used for final scoring.	Yes
5	Ghost Run ("Diff File")	September 1 201X through December 31 201X for year 201X. This includes 'new' organizations and providers who had no claims in the prior runs, but submitted claims during the last four months of the year.	Yes, for neutral adjustments

11 Eligibility, LVT Opt In

11.1 Data Sources

The Eligibility, LVT Opt In domain is populated by querying the eligibility database directly.

11.1.1 Data Source documentation

Eligibility API documentation: <http://dev.qpp.cms.gov/api/eligibility/docs/>

11.1.2 Data ingestion processes

The eligibility database is queried for individual and group level LVT opt-in status information, and the results are inserted directly into UDS.

11.2 Tables

11.2.1 group_lvt_opt_in_status

11.2.1.1 Description

In 2019, individual clinicians and group organizations who are below the low volume threshold (`low_volume_switch=true`) and thus normally ineligible to participate in the QPP program may be designated as eligible to participate if they opt in (`is_opt_in_eligible=true`). This choice to opt in (`is_opted_in` and `opt_in_decision_date`) to the QPP program can be performed by the clinicians at any time during the submission window. If we want to accurately reflect the choice made by the clinicians in tools like the CLT, we must refresh this data faster than we normally refresh Eligibility data (2-4 weeks). Thus, the LVT Opt In information must be separated out from the regular Eligibility domain processing, into its own pipeline and subdomain that can run faster and therefore more often (daily or even every 6 hours).

Keys: The surrogate key is `db_id`. The natural key is (TIN-run-year-data load).

11.2.1.2 Columns

- `db_id`
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- `db_data_load_id`
 - type: int8
 - description: Foreign key to `data_load.db_id`
- `year`
 - type: int2 not null
 - description: The performance year.
- `run`
 - type: int2 not null

- description: Eligibility is reviewed from two different time periods of claims for the performance year (Run 1 or 2). Runs 3 and 4 reconciles the data between the two runs at the end of the performance year. Run 0 indicates there is no special scenario information for the organization, and that it is associated to one or more APMs. Run 5 is a final eligibility determination period covering the final four months of the performance year. See "Definition of run parameter" below for more details.
- tin
 - type: varchar(255)
 - description: The 9-digit identifier of the provider associated with the score (Taxpayer Identification Number). Populated if the row represents an individual clinician or a group.
- is_opt_in_eligible
 - type: bool
 - description: True if below low-volume threshold (LVT) but eligible to opt-in anyway.
- is_opted_in
 - type: bool
 - description: True if below low-volume threshold (LVT) but decided to opt-in.
- opt_in_decision_date
 - type: date
 - description: The date that an opt-in decision was made. Null if no decision has been made.

11.2.2 individual_lvt_opt_in_status

11.2.2.1 Description

In 2019, individual clinicians and group organizations who are below the low volume threshold (`low_volume_switch=true`) and thus normally ineligible to participate in the QPP program may be designated as eligible to participate if they opt in (`is_opt_in_eligible=true`). This choice to opt in (`is_opted_in` and `opt_in_decision_date`) to the QPP program can be performed by the clinicians at any time during the submission window. If we want to accurately reflect the choice made by the clinicians in tools like the CLT, we must refresh this data faster than we normally refresh Eligibility data (2-4 weeks). Thus, the LVT Opt In information must be separated out from the regular Eligibility domain processing, into its own pipeline and subdomain that can run faster and therefore more often (daily or even every 6 hours).

Keys: The surrogate key is `db_id`. The natural key is (TIN-NPI-run-year-data load).

11.2.2.2 Columns

- `db_id`
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- `db_data_load_id`
 - type: int8
 - description: Foreign key to `data_load.db_id`
- `year`
 - type: int2 not null
 - description: The performance year.
- `run`
 - type: int2 not null
 - description: Eligibility is reviewed from two different time periods of claims for the performance year (Run 1 or 2). Runs 3 and 4 reconciles the data between the two runs at the end of the performance year. Run 0 indicates there is no special scenario information for the organization, and that it is

associated to one or more APMs. Run 5 is a final eligibility determination period covering the final four months of the performance year. See "Definition of run parameter" below for more details.

- tin
 - type: varchar(255)
 - description: The 9-digit identifier of the provider associated with the score (Taxpayer Identification Number). Populated if the row represents an individual clinician or a group.
- npi
 - type: varchar(255)
 - description: The 10-digit identifier of the participant (National Provider Identifier).
- is_opt_in_eligible
 - type: bool
 - description: True if below low-volume threshold (LVT) but eligible to opt-in anyway.
- is_opted_in
 - type: bool
 - description: True if below low-volume threshold (LVT) but decided to opt-in.
- opt_in_decision_date
 - type: date
 - description: The date that an opt-in decision was made. Null if no decision has been made.

12 Eligibility, SSP

12.1 Data Sources

A File on FMS

12.2 Tables

12.2.1 provider_apm_entity_group

12.2.1.1 Description

The Final Scoring data for SSP group affiliation

12.2.1.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_data_load_id
 - type: int8
 - description: Foreign key to data_load.db_id
- year
 - type: int2 not null
 - description: The performance year.
- tin
 - type: varchar(255)
 - description: The 9-digit identifier of the provider associated with the score (Taxpayer Identification Number). Populated if the row represents an individual clinician or a group.
- npi
 - type: varchar(255)
 - description: The 10-digit identifier of the participant (National Provider Identifier).
- entity_id
 - type: varchar(255)
 - description: Either the APM entity id or the Virtual Group id, depending on the submission's entity_type.
- ssp_group
 - type: varchar(255)
 - description:

13 Measures

13.1 Data Sources

The Measures domain contains all metadata about the various measures that clinicians can report on during a performance year. A measure will belong to one and only one category, which determines its metadata and data structure, how it gets reported, and so on. *Note:* try to be distinct when referring to *measures* (the metadata about a measure) and *measurements* (the data reported by a clinician for a particular measure).

This domain also contains Benchmarks data. These are distributions (deciles) about how the population of clinicians performed on a particular measure historically, or during the performance year in question.

Starting in the performance year 2018, the category ACI was renamed to PI. In this domain, A&R created duplicate tables (pi_measure compared to aci_measure and so on) as a response to this renaming. Thus only 2017 measures will appear in aci_measure, while 2018 measures and beyond appear in pi_measure, etc.

13.1.1 Data Source documentation

Both the data and schema of measures data is listed out by performance year here: <https://github.com/CMSgov/qpp-measures-data/tree/master/measures>

The schema of benchmark data is listed out by performance year here: <https://github.com/CMSgov/qpp-measures-data/tree/master/benchmarks>

The schema of benchmarks via the Submissions API is documented here: <https://cmsgov.github.io/qpp-submissions-docs/benchmarks>

13.1.2 Data ingestion process

The Measures and Benchmarks for a given performance year are loaded via a single process (pipeline UDS_Measures). The measures-data.json file for the performance year is retrieved from Github, and then separated out into the various normalized tables listed below. Next, the benchmarks for the performance year are queried from the Submissions API, associated to their particular measure, and stored in the benchmark table. After all validations for successful data parsing occur, the new Measures data load is made active for all users.

The pipeline is documented here: [Measures pipeline](#)⁴⁸

13.2 Tables

13.2.1 measure

13.2.1.1 Description

All measures for each performance year.

⁴⁸ <https://confluence.cms.gov/display/QPPAR/Measures+pipeline>

A given measure row will have one and only one of the following foreign keys populated: db_quality_measure_id, db_ia_measure_id, db_aci_measure_id, db_pi_measure_id, or db_cost_measure_id. This corresponds to the fact that any measure corresponds to only one category.

13.2.1.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_quality_measure_id
 - type: int8
 - description: Foreign key to the quality_measure table - quality_measure.db_id.
- db_ia_measure_id
 - type: int8
 - description: Foreign key to the ia_measure table - ia_measure.db_id.
- db_aci_measure_id
 - type: int8
 - description: Foreign key to the aci_measure table - aci_measure.db_id.
- db_cost_measure_id
 - type: int8
 - description: Foreign key to the cost_measure table - cost_measure.db_id.
- db_measure_category_id
 - type: int8 not null
 - description: Foreign key to the measure_category table - measure_category.db_id. All measures must belong to one category, so this can never be null.
- db_metric_type_id
 - type: int8 not null
 - description: Foreign key to the metric_type table - metric_type.db_id. All measures must belong to one metric type, so this can never be null.
- db_measure_specification_id
 - type: int8
 - description: Foreign key to the measure_specification table - measurement_specification.db_id. This will never be populated for IA measures. It may or may not be populated for other measure categories.
- measure_id
 - type: varchar(255) not null
 - description: A unique identifier for the measure. The identifier is not unique between performance years. For quality measures, the measureId is the same as the quality number. For a Promoting Interoperability (PI, formerly ACI) measure, the measureId is the measure identifier for the ACI / PI measure, and for an improvement activity (IA) measure, the measureId is the measure identifier for the IA measure.
- title
 - type: text
 - description: The name of the measure.
- description
 - type: text not null
 - description: A description of the measure for more detail with key words/phrases.
- first_performance_year
 - type: int4 not null
 - description: Year in which the measure was introduced. Defaults to 2017.
- last_performance_year

- type: int4
 - description: Year in which the measure was deprecated.
- aci_weight
 - type: int2
 - description: From the weighting scenario for the APM subdivision or because of special status. Percentage expressed as decimal (65% → 0.65).
- db_pi_measure_id
 - type: int8
 - description: Foreign key to the pi_measure table - pi_measure.db_id.
- pi_weight
 - type: int2
 - description:
- year
 - type: int4
 - description: The performance year for this measure's metadata.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.2 ia_measure

13.2.2.1 Description

All Improvement Activities measures for each performance year.

13.2.2.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_measurement_subcategory_id
 - type: int8
 - description: Foreign key to the measurement_subcategory table - measurement_subcategory.db_id.
- weight
 - type: varchar(255)
 - description: Determines the points granted for attesting to the measure. Options include: [null, medium, high].
- cerht_eligible
 - type: bool
 - description: If true, attesting to the measure will qualify the provider for an ACI / PI CEHRT bonus. Deprecated for 2019 and beyond.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.3 aci_measure

13.2.3.1 Description

All Advancing Care Information measures for the 2017 performance year.

13.2.3.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_measurement_objective_id
 - type: int8
 - description: Foreign key to the measurement_objective table - measurement_objective.db_id.
- weight
 - type: int2
 - description: Determines the performance score points granted for attesting to the measure.
- reporting_category
 - type: varchar(255)
 - description: The reporting category of the ACI measure. Options: [preAttestation, base, registryBonus, performanceBonus, cehrtBonus]
- is_required
 - type: bool not null
 - description: If true, attesting to the measure is required in order to receive a non-zero ACI score.
- is_bonus
 - type: bool not null
 - description: If true, attesting to the measure will qualify the provider for ACI bonus points.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.4 pi_measure

13.2.4.1 Description

All Promoting Interoperability measures for each performance year, starting from 2018.

13.2.4.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_measurement_objective_id
 - type: int8
 - description: Foreign key to the measurement_objective table - measurement_objective.db_id.
- weight

- type: int2
- description: Determines the performance score points granted for attesting to the measure.
- reporting_category
 - type: varchar(255)
 - description: The reporting category of the PI measure. Options: [required, bonus, exclusion, null]
- is_required
 - type: bool not null
 - description: If true, attesting to the measure is required in order to receive a non-zero PI score.
- is_bonus
 - type: bool not null
 - description: If true, attesting to the measure will qualify the provider for PI bonus points.
- db_pi_measure_exclusion_id
 - type: int8
 - description: Foreign key to table pi_measure_exclusion
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.5 quality_measure

13.2.5.1 Description

All Quality measures for each performance year.

Measures with metricType multiPerformanceRate must also have the properties overallAlgorithm and strata; other metricTypes do not

13.2.5.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_measure_type_id
 - type: int8 not null
 - description: Foreign key to the measure_type table - measure_type.db_id.
- national_quality_strategy_domain
 - type: varchar(255)
 - description: The domain in the National Quality Strategy (NQS) covered by the quality measure: Personal and Caregiver-Centered Experience Outcomes (PECCEO), Patient Safety (PS), Communication and Care Coordination (CCC), Community, Population, and Public Health (CPH), Efficiency and Cost Reduction Use of Healthcare Resources (ECR), Effective Clinical Care (ECC)
- e_measure_id
 - type: varchar(255)
 - description: Identifier for Electronic Clinical Quality Measures (ECQM).
- e_measure_uuid
 - type: varchar(255)
 - description: UUID for Electronic Clinical Quality Measures (ECQM).
- nqf_e_measure_id
 - type: varchar(255)
 - description: Identifier for measure specified in the Health Quality Measure Format (HQMF).

- nqf_id
 - type: varchar(255)
 - description: Identifier for the National Quality Forum (NQF) measure.
- is_high_priority
 - type: bool not null
 - description: If true, can be used in the place of an outcome measure to satisfy quality category requirements.
- is_inverse
 - type: bool not null
 - description: If true, a lower performance rate correlates with better performance.
- overall_algorithm
 - type: varchar(255)
 - description: Formula to determine the overall performance rate, given multiple strata of performance rates. Only applicable to multiPerformanceRate measures. Options: [simpleAverage, weightedAverage, sumNumerators, overallStratumOnly]
- primary_steward
 - type: varchar(255) not null
 - description: Organization who submits and maintains the measure.
- is_registry_measure
 - type: bool not null
 - description: If true, this measure was authored by a QCDR (Qualified Clinical Data Registry).
- is_risk_adjusted
 - type: bool
 - description: Flag to determine if the quality measure is adjusted for risk.
- cpc_plus_group
 - type: varchar(255)
 - description: CPC+ group to which the measure belongs.
- is_topped_out_by_program
 - type: bool
 - description:
- is_icd_impacted
 - type: bool
 - description:
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id
- is_clinical_guideline_changed
 - type: bool
 - description: Added for 2019 measures. True for measures that have been deemed as potentially no longer aligning with best practices or could lead to patient harm, false otherwise.

13.2.6 cost_measure

13.2.6.1 Description

All Cost measures for each performance year.

13.2.6.2 Columns

- db_id

- type: int8 not null primary key
- description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- is_inverse
 - type: bool not null
 - description: Flag that specifies if true, a measurement value of 100% implies bad performance while 0% implies good performance. If true, a lower performance rate correlates with better performance.
- overall_algorithm
 - type: varchar(255)
 - description: Specifies the type of algorithm used; accepted values: weightedAverage | simpleAverage | first | last. Formula to determine the overall performance rate, given multiple strata of performance rates. Only applicable to multiPerformanceRate measures.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.7 benchmark

13.2.7.1 Description

All historical and performance benchmarks corresponding to all measures for each performance year.

There are two types of benchmarks. During the submissions window for a given performance year, only historical benchmarks will be available. They are based on data from N-2 performance years prior. So if the performance_year is 2018, the benchmark_year will be 2016 for all historical benchmarks.

After the submissions window closes and final scoring is run, benchmarks for this performance year's data are calculated and available for comparison. So if the performance year is 2018, the performance benchmarks will also have a benchmark_year of 2018.

13.2.7.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_measure_id
 - type: int8
 - description: Foreign key to the measure table - measure.db_id.
- db_submission_method_id
 - type: int8
 - description: Foreign key to the submission_method table - submission_method.db_id.
- benchmark_year
 - type: int4 not null
 - description: The benchmark_year corresponds to the year of performance data that was used to generate this benchmark. In other words, submissions for performance_year X will be compared against benchmark_year Y's results.
- performance_year
 - type: int4 not null
 - description: The performance_year corresponds to the time period in which the performance data that was submitted for scoring originated.

- `decile_lower_bound2`
 - type: float8
 - description: The lower bound from the range of Decile 2 level of national performance within the baseline period.
- `decile_lower_bound3`
 - type: float8
 - description: The lower bound from the range of Decile 3 level of national performance within the baseline period.
- `decile_lower_bound4`
 - type: float8
 - description: The lower bound from the range of Decile 4 level of national performance within the baseline period.
- `decile_lower_bound5`
 - type: float8
 - description: The lower bound from the range of Decile 5 level of national performance within the baseline period.
- `decile_lower_bound6`
 - type: float8
 - description: The lower bound from the range of Decile 6 level of national performance within the baseline period.
- `decile_lower_bound7`
 - type: float8
 - description: The lower bound from the range of Decile 7 level of national performance within the baseline period.
- `decile_lower_bound8`
 - type: float8
 - description: The lower bound from the range of Decile 8 level of national performance within the baseline period.
- `decile_lower_bound9`
 - type: float8
 - description: The lower bound from the range of Decile 9 level of national performance within the baseline period.
- `decile_lower_bound10`
 - type: float8
 - description: The lower bound from the range of Decile 10 level of national performance within the baseline period.
- `db_data_load_id`
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id
- `is_topped_out`
 - type: bool
 - description: Boolean value that describes whether a benchmark's latter deciles repeat a value of 100 or, in the case of inverse measures, a value of 0.
- `is_topped_out_by_program`
 - type: bool
 - description: Indicates whether or not a Topped Out Measure will receive special scoring. If “true” the measure cannot earn more than 7 points.

13.2.8 aci_measure_measure_sets

13.2.8.1 Description

A join table that links an ACI measure to all measure sets to which the measure is associated. See measure_set table for more details.

13.2.8.2 Columns

- db_aci_measure_id
 - type: int8 not null
 - description: Foreign key to the aci_measure table - aci_measure.db_id
- db_measure_set_id
 - type: int8 not null
 - description: Foreign key to the measure_set table - measure_set.db_id

13.2.9 aci_measure_substitute

13.2.9.1 Description

A join table that links an ACI measure to other ACI measures that clinicians can use instead of the current measure.

13.2.9.2 Columns

- db_aci_measure_id
 - type: int8 not null
 - description: Foreign key to the aci_measure table - aci_measure.db_id.
- db_substitute_id
 - type: int8 not null
 - description: Foreign key to the aci_measure table - aci_measure.db_id.

13.2.10 code_object

13.2.10.1 Description

Codes that apply to the measure. More details needed...

13.2.10.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- code
 - type: varchar(255) not null

- description: The Current Procedural Terminology code (CPT code) used by insurers to help determine the amount of reimbursement that a practitioner will receive for services provided.
- modifiers
 - type: text
 - description: List of modifier codes that are relevant to a claim, regarding additional information about a patient encounter.
- modifier_exclusions
 - type: text
 - description: List of modifier codes that are excluded because they are opposites of the measure and thus can't be submitted in the same measurementSet such as: GT (via interactive audio and video telecommunications systems) and GQ (via an asynchronous telecommunications system)
- places_of_service
 - type: text
 - description: List of Place of Service Codes which are two-digit codes placed on health care professional claims to indicate the setting in which a service was provided.
- places_of_service_exclusions
 - type: int4
 - description: List of Place of Service Codes that are excluded.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.11 cost_measure_submission_methods

13.2.11.1 Description

A join table that links a cost measure with all the submission methods that can report on the measure.

13.2.11.2 Columns

- db_submission_method_id
 - type: int8 not null
 - description: Foreign key to the submission_method table - submission_method.db_id.
- db_cost_measure_id
 - type: int8 not null
 - description: Foreign key to the cost_measure table - cost_measure.db_id.

13.2.12 eligibility_option

13.2.12.1 Description

Eligibility options mirror denominator options in QCDR measure specifications. Each option comprises a set of codes used to identify eligible instances of the associated measure. Applies to quality measures.

13.2.12.2 Columns

- db_id
 - type: int8 not null primary key

- description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- sex_code
 - type: char(1)
 - description: Character code for the sex of the patient (M/F).
- min_age
 - type: float8
 - description: Minimum value of age for eligibility.
- max_age
 - type: float8
 - description: Maximum value of age for eligibility.
- diagnosis_codes
 - type: text
 - description: List of diagnosis codes and procedures of the patient.
- additional_diagnosis_codes
 - type: text
 - description: Additional list of diagnosis codes and procedures of the patient for eligibility.
- diagnosis_exclusion_codes
 - type: text
 - description: List of excluded diagnosis codes and procedures of the patient from eligibility.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.13 eligibility_option_additional_procedure_codes

13.2.13.1 Description

A list of HCPCS or CPT codes, at least one of which must be present to meet the eligibility option. If present, this field imposes a requirement in addition to the one from the procedureCodes field.

13.2.13.2 Columns

- db_code_object_id
 - type: int8 not null
 - description: Foreign key to the code_object table - code_object.db_id.
- db_eligibility_option_id
 - type: int8 not null
 - description: Foreign key to the eligibility_option table - eligibility_option.db_id.

13.2.14 eligibility_option_procedure_codes

13.2.14.1 Description

A list of HCPCS or CPT codes, at least one of which must be present to meet the eligibility option.

13.2.14.2 Columns

- db_code_object_id
 - type: int8 not null
 - description: Foreign key to the code_object table - code_object.db_id.
- db_eligibility_option_id
 - type: int8 not null
 - description: Foreign key to the eligibility_option table - eligibility_option.db_id.

13.2.15 eligibility_option_submission_method

13.2.15.1 Description

A join table between eligibility options and submission methods.

13.2.15.2 Columns

- db_eligibility_option_id
 - type: int8 not null
 - description: Foreign key to the eligibility_option table - eligibility_option.db_id.
- db_submission_method_id
 - type: int8 not null
 - description: Foreign key to the submission_method table - submission_method.db_id.

13.2.16 measure_category

13.2.16.1 Description

Table of category names. All measure rows must link to one and only one row in this table.

13.2.16.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- name
 - type: varchar(255) not null
 - description: QPP scoring category to which the measure belongs: Improvement Activities, Quality, Advancing Care Information / Promoting Interoperability, and Cost. Acceptable values are: ia, quality, aci (2017 only), pi (2018+ only), cost.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.17 measure_set

13.2.17.1 Description

PI measures can belong to the transition measure set. Quality measures can belong to multiple measure sets that represent different specialties.

Measure sets are complicated. Ask Paul for more details.

13.2.17.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- name
 - type: varchar(255) not null
 - description: Name of the Measure Set
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.18 measure_specification

13.2.18.1 Description

13.2.18.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- default
 - type: text
 - description: URL link for Measure Specification PDF to download by Submission Method.
- registry
 - type: text
 - description: URL link for Measure Specification PDF to download by Registry Submission Method.
- claims
 - type: text
 - description: URL link for Measure Specification PDF to download by Claims Submission Method.
- cms_web_interface
 - type: text
 - description: URL link for Measure Specification PDF to download by CMS Web Interface Submission Method.
- electronic_health_record
 - type: text

- description: URL link for Measure Specification PDF to download by EHR (Electronic Health Record) Submission Method.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.19 measure_subcategory

13.2.19.1 Description

Applies to IA measures.

13.2.19.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- name
 - type: varchar(255) not null
 - description: IA category which the measure incentivizes.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.20 measure_type

13.2.20.1 Description

Applies to quality measures.

13.2.20.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- name
 - type: varchar(255) not null
 - description: Quality category which the measure incentivizes.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.21 measurement_objective

13.2.21.1 Description

Applies to ACI / PI measures.

13.2.21.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- name
 - type: varchar(255) not null
 - description: ACI / PI category which the measure incentivizes.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.22 metric_type

13.2.22.1 Description

This metadata describes the type of measure, and thus indicates what data must be reported for the measure.

13.2.22.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- name
 - type: varchar(255) not null
 - description: Type of measurement that the measure requires in order to attest. Available options are: boolean, proportion, singlePerformanceRate, multiPerformanceRate, registrySinglePerformanceRate, registryMultiPerformanceRate, nonProportion, cahps, costScore.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.23 performance_option

13.2.23.1 Description

Performance options mirror numerator options in QCDR measure specifications. Each option comprises a set of codes used to identify instances of performance met, performance not met, performance exclusion or performance exception.

13.2.23.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- option_type
 - type: varchar(255) not null
 - description: Type of performance: performanceNotMet, performanceMet, eligiblePopulationExclusion, eligiblePopulationException
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.24 performance_option_quality_codes

13.2.24.1 Description

A join table between performance options and code objects.

13.2.24.2 Columns

- db_performance_option_id
 - type: int8 not null
 - description: Foreign key to the performance_option table - performance_option.db_id.
- db_code_object_id
 - type: int8 not null
 - description: Foreign key to the code_object table - code_object.db_id.

13.2.25 performance_option_submission_methods

13.2.25.1 Description

A join table between performance options and submission methods.

13.2.25.2 Columns

- db_performance_option_id
 - type: int8 not null
 - description: Foreign key to the performance_option table - performance_option.db_id.
- db_submission_method_id
 - type: int8 not null
 - description: Foreign key to the submission_method table - submission_method.db_id.

13.2.26 performance_stratum

13.2.26.1 Description

Population segments for which the measure requires reporting data. Only applicable to multiPerformanceRate measures. Applies to quality measures.

13.2.26.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- name
 - type: varchar(255)
 - description: The name of the performance stratum that separates patients into classes.
- description
 - type: text not null
 - description: A description of the performance stratum, explaining how patients are grouped.
- initial_population_uuid
 - type: varchar(255)
 - description: UUID for the initial population.
- denominator_uuid
 - type: varchar(255)
 - description: UUID for the denominator.
- numerator_uuid
 - type: varchar(255)
 - description: UUID for the numerator.
- denominator_exclusion_uuid
 - type: varchar(255)
 - description: UUID for the denominator exclusion.
- denominator_exception_uuid
 - type: varchar(255)
 - description: UUID for the denominator exception.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

13.2.27 pi_measure_measure_sets

13.2.27.1 Description

A join table that links a PI measure to all measure sets to which the measure is associated. See measure_set table for more details.

13.2.27.2 Columns

- db_pi_measure_id
 - type: int8 not null
 - description: Foreign key to the pi_measure table - pi_measure.db_id
- db_measure_set_id
 - type: int8 not null
 - description: Foreign key to the measure_set table - measure_set.db_id

13.2.28 pi_measure_substitute

13.2.28.1 Description

A join table that links a PI measure to other PI measures that clinicians can use instead of the current measure.

13.2.28.2 Columns

- db_pi_measure_id
 - type: int8 not null
 - description: Foreign key to the pi_measure table - pi_measure.db_id.
- db_substitute_id
 - type: int8 not null
 - description: Foreign key to the pi_measure table - pi_measure.db_id.

13.2.29 quality_measure_eligibility_options

13.2.29.1 Description

A join table that links a quality measure to its eligibility options.

13.2.29.2 Columns

- db_quality_measure_id
 - type: int8 not null
 - description: Foreign key to the quality_measure table - quality_measure.db_id.
- db_eligibility_option_id
 - type: int8 not null
 - description: Foreign key to the eligibility_option table - eligibility_option.db_id.

13.2.30 quality_measure_measure_sets

13.2.30.1 Description

A join table that links a quality measure to its measure sets.

13.2.30.2 Columns

- db_quality_measure_id
 - type: int8 not null
 - description: Foreign key to the quality_measure table - quality_measure.db_id.
- db_measure_set_id
 - type: int8 not null
 - description: Foreign key to the measure_set table - measure_set.db_id.

13.2.31 quality_measure_performance_option

13.2.31.1 Description

A join table that links a quality measure with all the performance options for the measure.

13.2.31.2 Columns

- db_performance_option_id
 - type: int8 not null
 - description: Foreign key to the performance_option table - performance_option.db_id.
- db_quality_measure_id
 - type: int8 not null
 - description: Foreign key to the quality_measure table - quality_measure.db_id.

13.2.32 quality_measure_stratum

13.2.32.1 Description

A join table that links a quality measure with all the performance strata for the measure.

13.2.32.2 Columns

- db_performance_stratum_id
 - type: int8 not null
 - description: Foreign key to the performance_stratum table - performance_stratum.db_id.
- db_quality_measure_id
 - type: int8 not null
 - description: Foreign key to the quality_measure table - quality_measure.db_id.

13.2.33 quality_measure_submission_methods

13.2.33.1 Description

A join table that links a quality measure with all the submission methods that can report on the measure.

13.2.33.2 Columns

- db_submission_method_id
 - type: int8 not null
 - description: Foreign key to the submission_method table - submission_method.db_id.
- db_quality_measure_id
 - type: int8 not null
 - description: Foreign key to the quality_measure table - quality_measure.db_id.

13.2.34 submission_method

13.2.34.1 Description

Possible methods for submitting performance data for the measure. Applies to quality and cost measures.

13.2.34.2 Columns

- db_id
 - type: int8 not null primary key
 - description: y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- name
 - type: varchar(255) not null
 - description: The method by which data is submitted for this benchmark. Acceptable values are cmsWebInterface, electronicHealthRecord, claims, registry, certifiedSurveyVendor, and administrativeClaims.
- for_quality_code
 - type: bool not null
 - description: Flag that specifies if a submission method is used for quality or not. The only submission method that can be used for quality as well as not used, is claims, all other are quality only.
- db_data_load_id
 - type: int8
 - description: Foreign key to the data_load table - data_load.db_id

14 Web Interface - Deprecated

14.1 Sources

The Web Interface (WI) domain includes data that is pulled from the Web Interface for Beneficiary Sampling. This is our source for clinics, organizations, registrations, beneficiaries, and beneficiary level measurement data that is not available from other QPP component sources. Beneficiary level measurement data is aggregated by the WI before sending it to the Submission API. In Year 1, this data was pulled into the UDS once after the submission window closed, and then again as corrections were made.

This domain was ONLY loaded for the 2017 performance year.

Web Interface (possibly?): <http://cmswi-api-dev.semanticbits.com:3000/>

14.2 Tables

14.2.1 wi_activity

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- db_wi_user_id int8 y Foreign key to the wi_user table - wi_user.db_id.
- db_wi_organization_id int8 y Foreign key to the wi_organizations table - wi_organizations.db_id.
- created_at timestamp y The creation time of the activity in RFC_3339 format.
- updated_at timestamp y The latest updated time of user activity in RFC_3339 format.
- action varchar y Recorded action of the user for the associated time of activity: REGISTRATION_VIEW_WI, SAMPLING_UPLOAD, REGISTRATION_VIEW_CAHPS, WI_REGISTRATION_CANCELANATION_EMAIL_FAILURE, ENTITY_CREATION, CAHPS_REGISTRATION_EMAIL_SUCESSS, AMPLING_DOWNLOAD, SAMPLING_DOWNLOAD_NO_DATA, ORG_INFO_VIEW_CAHPS, ENTITY_UPDATE, ORG_INFO_VIEW_WI, WI_REGISTRATION_EMAIL_SUCCESS, WI_REGISTRATION_CANCELANATION_EMAIL_SUCCESS, ENTITY_DELETION, CAHPS_REGISTRATION_CANCELANATION_EMAIL_SUCCESS, LOGIN
- file_name varchar y Name of file uploaded or downloaded.
- affected_entity varchar y Internal entity (table) that was modified. Values: beneficiary_provider, measures, providers, organization_contacts, file_process, submission_data, registrations, beneficiaries, organizations, organization_addresses, clinics
- entity_natural_key varchar y The natural key for individual entities that include an attribute which makes them unique; a business identifier for an entity such as NPI, Clinic ID, or a Medicare ID.
- number_of_entities_affected int4 y Count of number of entities affected.
- root_entity_identifier int8 y Unique primary key (surrogate key) for the entity.
- root_entity_type varchar y Location of activity used to roll up data for the sake of the UI: organizations, organization_addresses, providers, clinics, organization_contacts, registrations, beneficiaries
- performed_via varchar y Where the user activity was performed; accepted values: UI, API
- originated_from text y A place on the UI from which an activity originated. REST API calls made by UI will have to provide this information via an HTTP header, where applicable.

14.2.2 wi_beneficiary

14.2.2.1 Description

Web Interface Beneficiaries

Keys: The surrogate key is db_id.

PII Fields: mrn, last_name, first_name, medicare_id, other_id, date_of_birth

14.2.2.2 Columns

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- db_wi_organization_id int8 y Foreign key to the wi_organizations table - wi_organizations.db_id.
- db_wi_clinic_id int8 y y Foreign key to the wi_clinics table - wi_clinics.db_id.
- qualifications_comments text Comments entered at the Qualifications level.
- skipped_measures_count int4 Cached values calculated whenever submission data changes for a given beneficiary - how many measures the beneficiary has been skipped in.
- optional_measures_complete int4 Cached values calculated whenever submission data changes for a given beneficiary - how many optional measures are complete for the beneficiary.
- required_measures_complete int4 Cached values calculated whenever submission data changes for a given beneficiary - how many required measures are complete for the beneficiary.
- optional_measures_count int4 Cached values calculated whenever submission data changes for a given beneficiary - how many optional measures the beneficiary has reported.
- required_measures_count int4 Cached values calculated whenever submission data changes for a given beneficiary - how many required measures the beneficiary has reported.
- progress_status varchar(255) Overall status of the beneficiary - once a beneficiary is complete on all measures, their status will be 'complete'. Other acceptable values are 'incomplete' or 'skipped'.
- medical_not_qualified_date date y If a beneficiary is skipped for 'medical not qualified', the date of that action is contained in this field.
- medical_not_qualified_reason varchar(255) y If a beneficiary is skipped for 'medical not qualified', the reason is contained in this field.
- medical_record_found varchar(255) y Was a medical record found? Acceptable values are 'yes', 'no', or 'not qualified'.
- skipped_reason text y Comments on why a beneficiary was skipped on all measures.
- comments text y General comments.
- other_id varchar(50) y Alternate id for the beneficiary to the Medicare ID.
- mrn varchar(25) y Medical Record Number (MRN) of the beneficiary.
- date_of_birth date y Date of birth of the beneficiary.
- gender varchar(255) y Gender of the beneficiary.
- last_name varchar(255) Last name of the beneficiary.
- first_name varchar(255) First name of the beneficiary.
- medicare_id varchar(15) Medicare ID of the beneficiary.

14.2.3 wi_clinic

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- db_organization_id int8 y Foreign key to the wi_organizations table - wi_organizations.db_id.
- zipcode varchar(10) y The zip code of the clinic's address.
- state varchar(2) y The state of the clinic's address.
- city varchar(50) y The city of the clinic's address.
- address1 varchar(100) y The street name, or first line, of the clinic's address.
- address2 varchar(100) y The second line of the clinic's address.
- name varchar(100) The name of the clinic.
- clinic_id varchar(9) The id of the clinic.

14.2.4 wi_file_process

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- db_wi_organization_id int8 y y Foreign key to the wi_organizations table - wi_organizations.db_id.
- status varchar(255) y Status of file processed: COMPLETED_OK, INCOMPLETE_ERRORS, COMPLETED_ERRORS, SUBMITTED
- total_errors int4 y Count of number of errors found in processed file.
- total_records int4 y Count of total number of records in processed file.
- processing_type varchar(255) y The type of processing done for the file: upload, download
- file_type varchar(255) y The type of file that was processed: PATIENT, DATA-CONFIRMATION, CLINIC, PATIENT-RANKING, MEASURE-RATE, PROVIDER
- url text y The URL associated to the file being processed.
- processed_at timestamp y The processing time of the file in RFC_3339 format.
- created_at timestamp y The creation time of the file processing in RFC_3339 format.
- updated_at timestamp y The updated timestamp of the file processing in RFC_3339 format.

14.2.5 wi_file_process_message

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- db_wi_file_process_id int8 y Foreign key to the wi_file_process table - wi_file_process.db_id.
- created_at timestamp y The creation time of the file processing message in RFC_3339 format.
- error bool y Flag that records is the message from the file processing is an error or not.
- message text y Message displayed to user when error field is true.

14.2.6 wi_measure_metadata

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- db_measure_id int8 y Foreign key to the wi_measures table - wi_measures.db_id.
- year int4 The performance year.
- name varchar(255) The name of the measure metadata.
- measure_id varchar(255) The id of the measure, across different applications.
- is_composite bool Specific to DM (Diabetes Mellitus). Flag to know how to evaluate the measure. A composite is a roll-up of the child measures.
- parent_id int8 y Specific to DM (Diabetes Mellitus). Flag to know how to evaluate the measure. Parent ID of the child measure.
- required_count int4 y The same number in all fields - may change YOY. Currently the value is 248.

14.2.7 wi_measure

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- db_wi_beneficiary_id int8 y Foreign key to the wi_beneficiaries table - wi_beneficiaries.db_id.
- db_wi_measure_metadata_id int8 y Foreign key to the wi_measure_metadata table - wi_measure_metadata.db_id.
- name varchar(100) The name of the measure.
- rank int8 The rank of the beneficiary in the measure.
- status varchar(255) The general status of the measure. Acceptable values are 'complete', 'incomplete', or 'skipped'.

- confirmed varchar(255) y Whether a beneficiary is eligible for the measure. If so, the value in this field will be 'confirmed'. If not confirmed, the value will be 'skipped', and there will be comments in the 'skipped_reason' field.
- skipped_reason text y Comments on the reason a beneficiary is skipped for the measure.
- help_desk_ticket varchar(16) y Input field that ties back to a help desk request for a beneficiary to be skipped on the measure.
- comments text y Free-form text that can be entered on the confirmation of a measure.
- included_in_numerator int4 y Cached values calculated in the background.
- included_in_denominator bool y Cached values calculated in the background.
- denominator_exception bool y Cached values calculated in the background.

14.2.8 wi_organization_address

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- db_wi_organization_id int8 y Foreign key to the wi_organizations table - wi_organizations.db_id.
- is_primary bool y Flag for whether address is the primary address of the organization.
- zip varchar(255) y The zip code of the organization's address.
- state varchar(255) y The state of the organization's address.
- city varchar(255) y The city of the organization's address.
- address1 varchar(255) y The street name, or first line, of the organization's address.
- address2 varchar(255) y The second line of the organization's address.

14.2.9 wi_organization_contact

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- db_wi_organization_id int8 y Foreign key to the wi_organizations table - wi_organizations.db_id.
- phone varchar(255) y Phone number of the organization's contact.
- email varchar(255) y Email address of the organization's contact.
- last_name varchar(255) y Last name of organization's contact.
- first_name varchar(255) y First name of organization's contact.

14.2.10 wi_organization

14.2.10.1 Description

Web Interface Organizations

Keys: The surrogate key is db_id.

PII Fields: tin

14.2.10.2 Columns

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- duplicate_as_group bool y Specific flag for ACOs - does their submission need to be duplicated as a group submission? Yes/No
- nickname varchar(255) y Nickname for an organization.
- is_test_org bool y Internal flag for whether the organization is a test organization.
- group_size varchar(255) y The size range of the group.
- submission_id varchar(255) y The id of the submission.

- measurement_set_id varchar(255) y The UUID of the measurement set.
- aco_type varchar(255) y The type of ACO (e.g. Shared Savings Program, CPC+).
- aco_reporting_mechanism varchar(255) y *not used*
- aco_name varchar(160) y The name of the ACO.
- aco_id varchar(16) y The id of the ACO.
- submission_updated_on timestamp y The timestamp of the latest update of the submission.
- name varchar(100) y The name of the organization.
- tin varchar(10) The 9-digit identifier of the provider associated with the submission (Taxpayer Identification Number).
- year int4 The performance year.
- association_type varchar(255) y The organization's association type: null, CEC, OCM, CPC+, NEXT_GEN, SSP

14.2.11 wi_registration

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- db_wi_organization_id int8 y Foreign key to the wi_organization table - wi_organization.db_id.
- auto_registered bool Flag for whether an organization was auto-registered with information from previous reporting year. If an organization is registered for the previous reporting year, they will be auto-registered for the current reporting year.
- cahps_cancelled_on timestamp y The timestamp of when an organization cancelled CAHPS registration via the web interface.
- cahps_cancelled bool Flag for whether the organization cancelled CAHPS registration via the web interface.
- wi_cancelled_on timestamp y The timestamp of when an organization cancelled web interface group reporting.
- wi_cancelled bool Flag for whether an organization cancelled web interface group reporting.
- cahps_registered bool Flag for whether an organization is CAHPS registered.
- group_reporting bool Flag for whether or not registration is through a group. ACOs will have group_reporting = FALSE.
- year int4 The performance year.
- score float4 The score associated to the organization and performance year.

14.2.12 wi_submission_data

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- db_wi_measure_id int8 y Foreign key to the wi_measure table - wi_measure.db_id.
- comments text y General comments.
- scope varchar(255) y Construct used to group submission data when there can be multiples for a given measure. Currently used only for Care 1.
- value varchar(255) Values stored in the 'choices' in the measures JSON.
- attribute varchar(255) The key to the measures JSON.

14.2.13 wi_user

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- username varchar(255) y The username of the user.
- first_name varchar(255) y The first name of the user.
- last_name varchar(255) y The last name of the user.
- external_user_id varchar(255) y The user id to be used externally.
- auth_date timestamp y The date/time stamp of authentication in RFC_3339 format.

14.2.14 wi_user_organization

- db_wi_user_id int8 y Y Foreign key to the wi_user table - wi_user.db_id.
- db_wi_organization_id int8 y Y Foreign key to the wi_organizations table - wi_organizations.db_id.
- is_current bool y Flag for whether the organization is the user's current organization.

14.2.15 wi_user_role

- db_id int8 y BIGSERIAL (autoincrementing integer)-type identifier for the record.
- db_wi_user_id int8 y Foreign key to the wi_user table - wi_user.db_id.
- db_wi_organization_id int8 y Foreign key to the wi_organizations table - wi_organizations.db_id.
- role varchar(255) y The role granted to a user for access: STAFF, SECURITY-OFFICIAL.

15 ETL / Metadata

15.1 Data Sources

15.2 Tables

15.2.1 data_load

15.2.1.1 Description

In the UDS, we "snapshot" all data; this means we fully reload a domain whenever we update its data. The records in this table represent a single load of all related data elements for a year in a "Domain" of data (such as "Eligibility", or a specific table, such as "payment_adjustment"). For example, when querying the table "uds.provider", there will be multiple full copies of this data for a given NPI-run-year, which allows us to backtrack to a point in time. If you want to get only the latest validated data, then you would query the "data_load" table, to find the row where uds_domain = 'Eligibility' and active=true. Alternatively, the schema `active` contains a replica of all UDS tables, but limited only to the active (latest, validated) data for each table.

Keys: The surrogate key is db_id. The natural keys are: uds_domain, year, active

15.2.1.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- uds_domain
 - type: varchar(255) not null
 - description: The domain of UDS that the data load pertains to. Possible values: Eligibility, Measures, Submissions, Submission Scoring, Final Scoring. There are also some "subdomains" like "Final Scoring, Payment Adjustment", "Final Scoring, QIS", "Final Scoring, Score Profiler", and "Eligibility, LVT Opt In". These are tables that generally relate to the parent domain (i.e. Final Scoring) but are loaded at different times and with different processes than the main domain.
- year
 - type: int2 not null
 - description: The performance year for the domain's data.
- load_number
 - type: int4 not null
 - description: An incrementing number showing how many times the [uds_domain, year] combination has been loaded.
- load_time
 - type: timestamp not null default now()
 - description: The timestamp when the data load began.
- active
 - type: bool
 - description: Boolean indicator if the data load is the most recent for the specified [uds_domain, year] combination.

- load_process_name
 - type: varchar(255)
 - description: The name of the pipeline, script, or process that performed the data load.
- load_process_version
 - type: varchar(255)
 - description: The version of the process that performed the data load; usually the UDS version.
- is_validated
 - type: boolean
 - description: This flag indicates that a data_load at some point was the active, validated version of the data. It may no longer be the currently active version (which is determined by the active flag), but it was at some point, indicated by the active_time timestamp. This can help distinguish between "what-if" data loads that were never active or data_loads that had data quality validation issues, and old data loads that were active in the past.
- active_time
 - type: timestamp:
 - description: The timestamp when the data load became the active version of that [uds_domain, year] combination's data.
- status
 - type: varchar(255)
 - description: The status of the data load / data mart refresh, used to prevent simultaneous refreshes from occurring.

15.2.2 api_response

15.2.2.1 Description

Our UDS data warehouse model is a set of normalized tables, but most of our data sources format their data in the form of nested JSON objects. This table stores the entire JSON dataset as it was received from its source (an API response, a JSON column, etc). The table includes several different types of JSON datasets (the column json_response):

- Eligibility API responses
 - Provider
 - The source is the Eligibility API `/secure/npi/<npi>` endpoint; the data is the full provider response. The `endpoint_type` will be `'npi'` and the NPI value will be stored in `endpoint_value`. The `year` and `run` parameters will be populated on the table row.
 - Organization
 - The source is the Eligibility API `/secure/tin` endpoint (where the specific TIN is in the header); the data is the full organization response. The `endpoint_type` will be `'tin'` and the TIN value will be stored in `endpoint_value`. The `year` and `run` parameters will be populated on the table row.
 - Virtual Group
 - The source is the Eligibility API `/secure/virtual-group` endpoint (where the specific Virtual Group ID is in the header); the data is the full virtual group response. The `endpoint_type` will be `'virtual_group'` and the virtual_group_id value will be stored in `endpoint_value`. The `year` and `run` parameters will be populated on the table row.
 - APM Entity
 - The source is the Eligibility API `/secure/apms/<entity_id>` endpoint; the data is the full APM entity response. The `endpoint_type` will be `'apm'` and the APM entity_id value will be stored in `endpoint_value`. The `year` parameter will be populated on the table row (but not `run`).

- Submission objects
 - The source is the Submissions API `/submissions/<submission_id>` endpoint; the data is the full submission object. The `endpoint_type` will be `submission` and the submission_id value will be stored in `endpoint_value`. The `year` parameter will be populated.
 - This data is loaded only at the end of the Submissions window for the performance year, when all submission data has been processed.
- Scoring objects
 - There are two sources.
 - The first source is the Submissions API `/submissions/<submission_id>/score` endpoint, aka the preliminary scoring engine; the data is the scoring engine output for the submission. This data is loaded at the end of the Submissions window for the performance year at the same time that the Submissions data is loaded, above.
 - The second is the Final Scoring Hive table `submissionscore/se_results`; the data is the full final scoring output for the submission. This data is loaded with each Final Scoring run.
 - The `endpoint_type` will be `score` and the submission_id value will be stored in `endpoint_value`. The `year` parameter will be populated.

Keys: The surrogate key is db_id. The natural keys are: db_data_load_id, endpoint_type, endpoint_value

15.2.2.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_data_load_id
 - type: int8 not null
 - description: Foreign key to the data_load table - data_load.db_id
- endpoint_type
 - type: varchar(255) not null
 - description: The type of API response. One of: 'tin', 'npi', 'virtual_group', 'apm', 'submission', 'score'
- endpoint_value
 - type: varchar(255) not null
 - description: The main identifier of the API response. For example: The NPI, TIN, virtual group id, APM entity ID, or submission ID.
 - PII field; contains the TIN when the response is for an Organization via the Eligibility API.
- year
 - type: int2
 - description: The performance year for the data, if applicable.
- run
 - type: int2
 - description: The run of the data, if applicable.
- json_response
 - type: text not null
 - description: The full JSON response from the Eligibility or Submission API
 - PII field; contains TINs for all Eligibility and most Submission objects

15.2.3 xref_data_load_data_source

15.2.3.1 Description

This table represents the relationship between a data_load and the data_source. A single data load may have many data sources. For example a data load of 2018 Eligibility has at least two sources: the Eligibility database (from PECOS claims) and the APM data (from AMS).

Keys: The surrogate key is db_id. The natural keys are: db_data_load_id and db_data_source_id

PII Fields: None

15.2.3.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- db_data_load_id
 - type: int8 not null
 - description: Foreign key linking data loads to data sources - data_load.db_id.
- db_data_source_id
 - type: int8 not null
 - description: Foreign key linking data loads to data sources - data_source.db_id.

15.2.4 data_source

15.2.4.1 Description

This table represents a single source of data. It is designed to store metadata around the version of the source data and the timestamp at which the data became available / active. Ideally the information in this table can be derived from the source team's metadata.json file in the FMS Data Repository, or some other automatic method.

As of 10/31/2019, the only data sources that get saved here are for a Final Scoring data load.

Keys: The surrogate key is db_id. The natural keys are name and version.

15.2.4.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- source_name
 - type: varchar(255) not null
 - description: The name of the source dataset, i.e. Eligibility, Measures, etc.
- source_version
 - type: varchar(255)
 - description: The version of the source dataset.
- source_effective_time

- type: timestamp
- description: The time when the source version became active.

15.2.5 fs_run_metrics_diff

15.2.5.1 Description

This table contains metrics and their percent difference between two final scoring runs.

Keys: The surrogate key is db_id.

15.2.5.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- run_at
 - type: timestamp not null
 - description: The datetime the diff was ran.
- year
 - type: varchar(255)
 - description: Either the year of both runs, or a string with both comparing different years.
- old_run
 - type: int8
 - description: The previous final scoring run_id.
- new_run
 - type: int8
 - description: The current final scoring run_id.
- metric
 - type: varchar(255)
 - description: The description of the metric calculated.
- old_value
 - type: float8
 - description: Value of the metric from old_run.
- new_value
 - type: float8
 - description: Value of the metric from new_run.
- percent_change
 - type: float8
 - description: Percent difference of old_value and new_value.

15.2.6 fs_run_agg_metrics

15.2.6.1 Description

This table contains aggregate metrics for FS run. The table needs to store metrics for different runs

Keys: The surrogate key is db_id .

15.2.6.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- run_at
 - type: timestamp not null
 - description: The datetime the diff was ran.
- db_data_load_id
 - type: int8 foreign key
 - description: Foreign key to the data_load table - data_load.db_id. Non-null when the submission is processed via a batch data load process. Null when the submission is processed via a streaming (audits) process.
- year
 - type: varchar(255)
 - description: Either the year of both runs, or a string with both comparing different years.
- run_id
 - type: int8
 - description: The current final scoring run_id.
- metric
 - type: varchar(255)
 - description: The description of the metric calculated.
- value
 - type: float8
 - description: Value of the metric from run_id.

15.2.7 fs_run_descriptions

15.2.7.1 Description

This table contains the description for fs runs. Used by the Final Scoring Diff Metrics pipeline and UDS_Scoring.

Keys: The surrogate key is db_id .

15.2.7.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.

- run_id
 - type: int8
 - description: The final scoring run_id
- year
 - type: int8
 - description: The year of the final scoring run
- description
 - type: varchar(255)
 - description: The description of the run

15.2.8 tin_npi_svc_audit_log - Deprecated

15.2.8.1 Description

This table will contain audit logs of requests made to the QPPAR API.

PII fields: tin

15.2.8.2 Columns

- db_id
 - type: int8 not null primary key
 - description: Unique BIGSERIAL (autoincrementing integer)-type identifier for the record. Created by UDS upon insert to table.
- request_completed
 - type: timestamp not null
 - description: The time and date of the API request.
- user_id
 - type: varchar(255) not null
 - description: An identifier of the user making the API request.
- operation
 - type: varchar(255) not null
- tin
 - type: varchar(255)
 - description: The taxpayer identification number in the request.
 - PII field.
- npi
 - type: varchar(255)
 - description: The national provider identifier in the request.
- status_code
 - type: int2 not null
 - description: The HTTP state code returned in the response. A successful request is indicated by codes less than 400.
- response_payload
 - type: text not null
 - description: The data that was sent back in the response.