CAliPHR Calculation Engine Design

Contents

[1. Overview 2](#_Toc439085034)

[2. HQMF 2](#_Toc439085035)

[3. Patient Data Copy to H2 2](#_Toc439085036)

[4. Views for similar types of data 3](#_Toc439085037)

[5. H2 Views for Data Criteria 3](#_Toc439085038)

[Example 5‑A: Simple Data Criterion compared to Measurement Period 3](#_Toc439085039)

[Example 5‑B: Data Criterion compared to another Data Criterion within a range 4](#_Toc439085040)

[Example 5‑C: Data Criterion with a FIRST subset 5](#_Toc439085041)

[Example 5‑D: Data Criterion with a MOST RECENT subset 7](#_Toc439085042)

[Example 5‑E: Data Criteria grouping 8](#_Toc439085043)

[Example 5‑F: Data Criteria groups with a COUNT/SUM subset 9](#_Toc439085044)

[Example 5‑G: Data Criterion with Negation 10](#_Toc439085045)

[Example 5‑H: Generated SQL for all populations of a measure 11](#_Toc439085046)

[Example 5‑I: Numerator with Specific Occurrences carried over from Denominator 14](#_Toc439085047)

[Example 5‑J: Generated SQL for preconditions that contain NOT in populations 15](#_Toc439085048)

[6. Time Relationships and Data Criteria 16](#_Toc439085049)

[6.1. Note how time differences were implemented in H2 17](#_Toc439085050)

[7. Specific Occurrences 17](#_Toc439085051)

[8. Patient-Based vs Episode-of-Care measures 18](#_Toc439085052)

[9. Continuous Variable Measures 18](#_Toc439085053)

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 1.0 | 12/28/2015 | Daniel Rosenbaum | Initial Design |
| 1.01 | 8/31/2016 | Daniel Rosenbaum | Updated for open source project |

# Overview

This document describes the design of the Measure Calculation Engine component of the CQM Aligned Population Health Reporting tool (CAliPHR) in detail.

# HQMF

The content and structure of a clinical quality measure (CQM) is specified in an HL7 Health Quality Measures Format (HQMF) file, which is a standard for representing health quality measures as electronic documents. A quality measure is a quantitative tool that provides an indication of an individual or organization’s performance in relation to a specified process or outcome via the measurement of an action, process or outcome of clinical care. Quality measures are often derived from clinical guidelines and are designed to determine whether the appropriate care has been provided given a set of clinical criteria and an evidence base. A quality measure expressed in HQMF format is referred to as an "eMeasure".

CAliPHR leverages the HQMF files contained within the Meaningful Use Stage 2 Quality Bundle, currently at version 2.6.0 in Project Cypress. The hqmf-parser (and associated hqmf-model) module of the open-source PopHealth/Project Cypress code was utilized as a reference during implementation, especially the XPath expressions used for HQMF parsing logic. The CAliPHR Java classes for HQMF data closely resemble the corresponding PopHealth Ruby classes, and are very similar when serialized to JSON format as well (useful for verification and comparison). The code can be found at [this link](https://github.com/projectcypress/health-data-standards/tree/master/lib/hqmf-parser).

The HQMF file content is first parsed to intermediate Java classes, which are in turn used to generate the series of SQL statements required to query data and perform calculations for the measure. This includes classes for the data criteria, preconditions, population criteria, temporal relationships, attributes, and other items that are specified in the HQMF file.

# Patient Data Copy to H2

Upon initiation of the calculation process, all the provider’s clinical patient data is copied from PostgreSQL and loaded into an embedded H2 in-memory database. The H2 tables mostly mirror the corresponding PostgreSQL table structures, but are somewhat de-normalized, simplified and optimized for measure calculation purposes. Only the minimum fields necessary for measure calculation are retained, and indexes are added for performance.

# Views for similar types of data

Similar to PopHealth’s concept of Patient API Function, data of similar types are UNIONed together so they would be treated logically the same. This is implemented in CAliPHR as SQL views. These groupings are specified [here](https://github.com/hquery/hqmf-parser/blob/master/README.md). Specifically:

|  |
| --- |
| * allProblems => conditions, socialHistories, procedures * encounters => encounters * allProcedures => procedures, immunizations, medications * procedureResults => results, vitalSigns, procedures * allergies => allergies * allMedications => medications, immunizations * laboratoryTests => results, vitalSigns * careGoals => careGoals * procedures => procedures * allDevices => conditions, procedures, careGoals, medicalEquipment |

The data\_criteria.json file from Project Cypress was utilized to identify the appropriate view (patient API function) to use for specific data element types.

# H2 Views for Data Criteria

To manage complexity of SQL, a new H2 view is generated for each data criterion defined for the measure. Note that some data criteria have source data criteria, in which case the SQL generated for the view references the source view. So you end up with a nesting of views that build on each other.

The following are example views for a data criterion and population preconditions. Note that a number of the examples have temporal relationships. For more information concerning temporal comparisons, see section **6** below.

|  |
| --- |
| Example ‑ Simple Data Criterion compared to Measurement Period Human-readable representation:  "Physical Exam, Performed: Visual Exam of Foot" during "Measurement Period" |
| **JSON representation:**  "PhysicalExamPerformedVisualExamOfFoot\_precondition\_27": {  "title": "Visual Exam of Foot",  "description": "Physical Exam, Performed: Visual Exam of Foot",  "codeListId": "2.16.840.1.113883.3.464.1003.103.12.1013",  "type": "physical\_exams",  "patientApiFunction": "procedureResults",  "definition": "physical\_exam",  "status": "performed",  "negation": false,  "sourceDataCriteria": "PhysicalExamPerformedVisualExamOfFoot",  "variable": false,  "isSourceDataDriteria": false,  "codeListXpath": "./\*/cda:code",  "temporalReferences": [  {  "type": "DURING",  "reference": "MeasurePeriod"  }  ]  } |
| **Generated SQL:**  CREATE VIEW caliphr.CMS123v3\_PhysicalExamPerformedVisualExamOfFoot\_precondition\_27 AS  SELECT t29.\*  FROM caliphr.CMS123v3\_PhysicalExamPerformedVisualExamOfFoot t29  WHERE  t29.negation\_code\_id IS NULL  AND  /\* DURING \*/  t29.effective\_time\_start >= MILLIS\_EPOCH('2013-01-01')  AND t29.effective\_time\_end <= MILLIS\_EPOCH('2013-12-31');  Source Views:  CREATE VIEW caliphr.CMS123v3\_PhysicalExamPerformedVisualExamOfFoot AS  SELECT t28.\*  FROM caliphr.procedureResults t28  WHERE  t28.code\_id IN  (  SELECT code\_id  FROM common.VALUE\_SET\_CODE  WHERE  value\_set\_id = 200  )  AND t28.status\_code\_name IN  (  'performed','completed'  ); |
| Example ‑ Data Criterion compared to another Data Criterion within a range Human-readable representation:  "Diagnosis, Active: Alcohol and Drug Dependence" <= 60 day(s) starts before start of "Occurrence A of Diagnosis, Active: Alcohol and Drug Dependence" |
| **JSON representation:**  "EncounterPerformedAlcoholAndDrugDependenceTreatment\_precondition\_43": {  "title": "Alcohol and Drug Dependence Treatment",  "description": "Encounter, Performed: Alcohol and Drug Dependence Treatment",  "codeListId": "2.16.840.1.113883.3.464.1003.106.12.1005",  "type": "encounters",  "patientApiFunction": "encounters",  "definition": "encounter",  "negation": false,  "sourceDataCriteria": "EncounterPerformedAlcoholAndDrugDependenceTreatment",  "variable": false,  "isSourceDataDriteria": false,  "codeListXpath": "./\*/cda:code",  "temporalReferences": [  {  "type": "SAS",  "reference": "OccurrenceAAlcoholAndDrugDependenceTreatment2\_precondition\_48",  "range": {  "type": "PQ",  "low": {  "type": "PQ",  "inclusive": false,  "derived": false  },  "high": {  "type": "PQ",  "unit": "d",  "value": "30",  "inclusive": true,  "derived": false  }  }  }  ]  } |
| **Generated SQL:**  CREATE VIEW caliphr.CMS137v3\_EncounterPerformedAlcoholAndDrugDependenceTreatment\_precondition\_43 AS  SELECT t36.\*, o2.ido2  FROM caliphr.CMS137v3\_EncounterPerformedAlcoholAndDrugDependenceTreatment t36  JOIN caliphr.CMS137v3\_OccurrenceAAlcoholAndDrugDependenceTreatment2\_precondition\_48 o2 ON t36.patient\_id = o2.patient\_id  WHERE  t36.negation\_code\_id IS NULL  AND  /\* SAS \*/  t36.effective\_time\_start > o2.effective\_time\_start  AND DAYS\_DIFFERENCE(o2.effective\_time\_start, t36.effective\_time\_start) <= 30;  Source Views:  CREATE VIEW caliphr.CMS137v3\_EncounterPerformedAlcoholAndDrugDependenceTreatment AS  SELECT t35.\*  FROM caliphr.encounters t35  WHERE  t35.code\_id IN  (  SELECT code\_id  FROM common.VALUE\_SET\_CODE  WHERE  value\_set\_id = 218  );  CREATE VIEW caliphr.CMS137v3\_OccurrenceAAlcoholAndDrugDependenceTreatment2 AS  SELECT t28.\*, t28.id as ido2  FROM caliphr.encounters t28  WHERE  t28.code\_id IN  (  SELECT code\_id  FROM common.VALUE\_SET\_CODE  WHERE  value\_set\_id = 218  );  CREATE VIEW caliphr.CMS137v3\_OccurrenceAAlcoholAndDrugDependenceTreatment2\_precondition\_48 AS  SELECT t37.\*  FROM caliphr.CMS137v3\_OccurrenceAAlcoholAndDrugDependenceTreatment2 t37  WHERE  t37.negation\_code\_id IS NULL; |

|  |
| --- |
| Example ‑ Data Criterion with a FIRST subset **Human-readable representation:**  FIRST: "Occurrence A of Procedure, Performed: Prostate Cancer Treatment" during "Measurement Period" |
| JSON representation:  "OccurrenceAProstateCancerTreatment4\_precondition\_14": {  "title": "Prostate Cancer Treatment",  "description": "Procedure, Performed: Prostate Cancer Treatment",  "codeListId": "2.16.840.1.113883.3.526.3.398",  "type": "procedures",  "patientApiFunction": "allProcedures",  "definition": "procedure",  "status": "performed",  "negation": false,  "specificOccurrence": "A",  "specificOccurrenceConst": "PROCEDURE\_PERFORMED\_PROSTATE\_CANCER\_TREATMENT",  "sourceDataCriteria": "OccurrenceAProstateCancerTreatment4",  "variable": false,  "isSourceDataDriteria": false,  "codeListXpath": "./\*/cda:code",  "subsetOperators": [  {  "type": "FIRST"  }  ],  "temporalReferences": [  {  "type": "DURING",  "reference": "MeasurePeriod"  }  ]  } |
| Generated SQL:  CREATE VIEW caliphr.CMS129v4\_OccurrenceAProstateCancerTreatment4\_precondition\_14 AS  SELECT \*  FROM  (  SELECT t8.\*  FROM caliphr.CMS129v4\_OccurrenceAProstateCancerTreatment4 t8  WHERE  t8.negation\_code\_id IS NULL  AND  /\* DURING \*/  t8.effective\_time\_start >= MILLIS\_EPOCH('2013-01-01')  AND t8.effective\_time\_end <= MILLIS\_EPOCH('2013-12-31')  ) AS t8\_MAIN INNER  JOIN  (  SELECT MIN(t8.effective\_time\_start) AS ETS, t8.id as record\_id  FROM caliphr.CMS129v4\_OccurrenceAProstateCancerTreatment4 t8  WHERE  t8.negation\_code\_id IS NULL  AND  /\* DURING \*/  t8.effective\_time\_start >= MILLIS\_EPOCH('2013-01-01')  AND t8.effective\_time\_end <= MILLIS\_EPOCH('2013-12-31') GROUP BY t8.id  ) AS t8\_FR ON t8\_MAIN.id = t8\_FR.record\_id  AND t8\_MAIN.effective\_time\_start = t8\_FR.ETS;  Source Views:  CREATE VIEW caliphr.CMS129v4\_OccurrenceAProstateCancerTreatment4 AS  SELECT t7.\*, t7.id as ido2  FROM caliphr.allProcedures t7  WHERE  t7.code\_id IN  (  SELECT code\_id  FROM common.VALUE\_SET\_CODE  WHERE  value\_set\_id = 580  )  AND t7.status\_code\_name IN  (  'performed','completed'  ); |
| Example ‑ Data Criterion with a MOST RECENT subset **Human-readable representation:**  MOST RECENT:"Occurrence A of Physical Exam, Finding: BMI LOINC Value" <= 6 month(s) starts before or during "Occurrence A of Encounter, Performed: BMI Encounter Code Set" |
| **JSON representation:**  "OccurrenceABmiLoincValue1\_precondition\_30": {  "id": "OccurrenceABmiLoincValue1\_precondition\_30",  "title": "BMI LOINC Value",  "description": "Physical Exam: BMI LOINC Value",  "codeListId": "2.16.840.1.113883.3.600.1.681",  "type": "physical\_exams",  "patientApiFunction": "procedureResults",  "definition": "physical\_exam",  "negation": false,  "specificOccurrence": "A",  "specificOccurrenceConst": "PHYSICAL\_EXAM\_BMI\_LOINC\_VALUE",  "sourceDataCriteria": "OccurrenceABmiLoincValue1",  "variable": false,  "isSourceDataDriteria": false,  "codeListXpath": "./\*/cda:code",  "subsetOperators": [  {  "type": "QDM\_LAST"  }  ],  "temporalReferences": [  {  "type": "SBE",  "reference": "OccurrenceABmiEncounterCodeSet2",  "range": {  "type": "PQ",  "low": {  "type": "PQ",  "inclusive": false,  "derived": false  },  "high": {  "type": "PQ",  "unit": "mo",  "value": "6",  "inclusive": true,  "derived": false  }  }  }  ]  } |
| **Generated SQL:**  CREATE VIEW caliphr.CMS69v3\_OccurrenceABmiLoincValue1\_precondition\_30 AS  SELECT \*  FROM  (  SELECT t25.\*, o1.ido1  FROM caliphr.CMS69v3\_OccurrenceABmiLoincValue1 t25  JOIN caliphr.CMS69v3\_OccurrenceABmiEncounterCodeSet2 o1 ON t25.patient\_id = o1.patient\_id  WHERE  t25.negation\_code\_id IS NULL  AND  /\* SBE \*/  t25.effective\_time\_start < o1.effective\_time\_end  AND MONTHS\_DIFFERENCE(t25.effective\_time\_start, o1.effective\_time\_end) <= 6  ) AS t25\_MAIN INNER  JOIN  (  SELECT MAX(t25.effective\_time\_start) AS ETS, t25.id as record\_id  FROM caliphr.CMS69v3\_OccurrenceABmiLoincValue1 t25  JOIN caliphr.CMS69v3\_OccurrenceABmiEncounterCodeSet2 o1 ON t25.patient\_id = o1.patient\_id  WHERE  t25.negation\_code\_id IS NULL  AND  /\* SBE \*/  t25.effective\_time\_start < o1.effective\_time\_end  AND MONTHS\_DIFFERENCE(t25.effective\_time\_start, o1.effective\_time\_end) <= 6 GROUP BY t25.id  ) AS t25\_MR ON t25\_MAIN.id = t25\_MR.record\_id  AND t25\_MAIN.effective\_time\_start = t25\_MR.ETS;  Source Views:  CREATE VIEW caliphr.CMS69v3\_OccurrenceABmiLoincValue1 AS  SELECT t22.\*, t22.id as ido3  FROM caliphr.procedureResults t22  WHERE  t22.code\_id IN  (  SELECT code\_id  FROM common.VALUE\_SET\_CODE  WHERE  value\_set\_id = 607  );  CREATE VIEW caliphr.CMS69v3\_OccurrenceABmiEncounterCodeSet2 AS  SELECT t1.\*, t1.id as ido1  FROM caliphr.encounters t1  WHERE  t1.code\_id IN  (  SELECT code\_id  FROM common.VALUE\_SET\_CODE  WHERE  value\_set\_id = 604  ); |

|  |
| --- |
| Example ‑ Data Criteria grouping **Human-readable representation:**   * OR: "Encounter, Performed: Office Visit" * OR: "Encounter, Performed: Emergency Department Visit" * OR: "Encounter, Performed: Detoxification Visit" * OR: "Encounter, Performed: Hospital Observation Care - Initial" * OR: "Encounter, Performed: Hospital Inpatient Visit - Initial" * OR: "Encounter, Performed: Discharge Services - Hospital Inpatient Same Day Discharge" * OR: "Encounter, Performed: Discharge Services - Hospital Inpatient" * OR: "Encounter, Performed: Face-to-Face Interaction" |
| **JSON representation:**  "GROUP\_SDU\_CHILDREN\_77": {  "title": "GROUP\_SDU\_CHILDREN\_77",  "type": "derived",  "patientApiFunction": "",  "definition": "derived",  "negation": false,  "sourceDataCriteria": "GROUP\_SDU\_CHILDREN\_77",  "childrenCriteria": [  "EncounterPerformedOfficeVisit\_precondition\_9",  "EncounterPerformedEmergencyDepartmentVisit\_precondition\_11",  "EncounterPerformedDetoxificationVisit\_precondition\_13",  "EncounterPerformedHospitalObservationCareInitial\_precondition\_15",  "EncounterPerformedHospitalInpatientVisitInitial\_precondition\_17",  "EncounterPerformedDischargeServicesHospitalInpatientSameDayDischarge\_precondition\_19",  "EncounterPerformedDischargeServicesHospitalInpatient\_precondition\_21",  "EncounterPerformedFaceToFaceInteraction\_precondition\_23"  ],  "derivationOperator": "UNION",  "variable": false,  "isSourceDataDriteria": false,  "codeListXpath": "./\*/cda:code"  } |
| **Generated SQL:**  CREATE VIEW caliphr.CMS137v3\_GROUP\_SDU\_CHILDREN\_77 AS  select t.patient\_id, t.effective\_time\_start, t.effective\_time\_end, t.id  from caliphr.CMS137v3\_EncounterPerformedOfficeVisit\_precondition\_9 t  UNION  select t.patient\_id, t.effective\_time\_start, t.effective\_time\_end, t.id  from caliphr.CMS137v3\_EncounterPerformedEmergencyDepartmentVisit\_precondition\_11 t  UNION  select t.patient\_id, t.effective\_time\_start, t.effective\_time\_end, t.id  from caliphr.CMS137v3\_EncounterPerformedDetoxificationVisit\_precondition\_13 t  UNION  select t.patient\_id, t.effective\_time\_start, t.effective\_time\_end, t.id  from caliphr.CMS137v3\_EncounterPerformedHospitalObservationCareInitial\_precondition\_15 t  UNION  select t.patient\_id, t.effective\_time\_start, t.effective\_time\_end, t.id  from caliphr.CMS137v3\_EncounterPerformedHospitalInpatientVisitInitial\_precondition\_17 t  UNION  select t.patient\_id, t.effective\_time\_start, t.effective\_time\_end, t.id  from caliphr.CMS137v3\_EncounterPerformedDischargeServicesHospitalInpatientSameDayDischarge\_precondition\_19 t  UNION  select t.patient\_id, t.effective\_time\_start, t.effective\_time\_end, t.id  from caliphr.CMS137v3\_EncounterPerformedDischargeServicesHospitalInpatient\_precondition\_21 t  UNION  select t.patient\_id, t.effective\_time\_start, t.effective\_time\_end, t.id  from caliphr.CMS137v3\_EncounterPerformedFaceToFaceInteraction\_precondition\_23 t;  (Source Views Omitted) |
| Example ‑ Data Criteria groups with a COUNT/SUM subset **Human-readable representation:**  Count >= 2 of:   * OR: "Encounter, Performed: Alcohol and Drug Dependence Treatment" * OR: "Encounter, Performed: Psych Visit - Psychotherapy" |
| JSON representation:  "GROUP\_COUNT\_79": {  "title": "GROUP\_COUNT\_79",  "type": "derived",  "patientApiFunction": "",  "definition": "derived",  "negation": false,  "sourceDataCriteria": "GROUP\_COUNT\_79",  "childrenCriteria": [  "EncounterPerformedAlcoholAndDrugDependenceTreatment\_precondition\_43",  "EncounterPerformedPsychVisitPsychotherapy\_precondition\_45"  ],  "derivationOperator": "UNION",  "variable": false,  "isSourceDataDriteria": false,  "codeListXpath": "./\*/cda:code",  "subsetOperators": [  {  "type": "SUM",  "value": {  "type": "IVL\_PQ",  "low": {  "type": "PQ",  "value": "2",  "inclusive": true,  "derived": false  }  }  }  ]  } |
| **Generated SQL:**  CREATE VIEW caliphr.CMS137v3\_GROUP\_COUNT\_79\_DERIVED AS  select t.patient\_id, t.effective\_time\_start, t.effective\_time\_end, t.id, t.ido2  from caliphr.CMS137v3\_EncounterPerformedAlcoholAndDrugDependenceTreatment\_precondition\_43 t  UNION  select t.patient\_id, t.effective\_time\_start, t.effective\_time\_end, t.id, t.ido2  from caliphr.CMS137v3\_EncounterPerformedPsychVisitPsychotherapy\_precondition\_45 t;  CREATE VIEW caliphr.CMS137v3\_GROUP\_COUNT\_79 AS  SELECT \*  FROM  (  SELECT \*  FROM caliphr.CMS137v3\_GROUP\_COUNT\_79\_DERIVED  ) CMS137v3\_GROUP\_COUNT\_79\_DERIVED\_CNTBASE  INNER JOIN  (  SELECT patient\_id AS cnt\_patient\_id, COUNT(\*) AS CNT  FROM caliphr.CMS137v3\_GROUP\_COUNT\_79\_DERIVED GROUP BY patient\_id  ) AS CMS137v3\_GROUP\_COUNT\_79\_DERIVED\_CNT  ON CMS137v3\_GROUP\_COUNT\_79\_DERIVED\_CNTBASE.patient\_id = CMS137v3\_GROUP\_COUNT\_79\_DERIVED\_CNT.cnt\_patient\_id  WHERE  CMS137v3\_GROUP\_COUNT\_79\_DERIVED\_CNT.CNT >= 2;  (Source Views Omitted) |
| Example ‑ Data Criterion with Negation **Human-readable representation:**  "Risk Category Assessment not done: Medical Reason" for "Falls Screening" during "Measurement Period" |
| **JSON representation:**  "RiskCategoryAssessmentFallsScreening\_precondition\_26": {  "id": "RiskCategoryAssessmentFallsScreening\_precondition\_26",  "title": "Falls Screening",  "description": "Risk Category Assessment: Falls Screening",  "codeListId": "2.16.840.1.113883.3.464.1003.118.12.1028",  "type": "risk\_category\_assessments",  "patientApiFunction": "procedures",  "definition": "risk\_category\_assessment",  "negation": true,  "sourceDataCriteria": "RiskCategoryAssessmentFallsScreening",  "negationCodeListId": "2.16.840.1.113883.3.526.3.1007",  "variable": false,  "isSourceDataDriteria": false,  "codeListXpath": "./\*/cda:code",  "temporalReferences": [  {  "type": "DURING",  "reference": "MeasurePeriod"  }  ]  } |
| **Generated SQL:**  CREATE VIEW caliphr.CMS139v3\_RiskCategoryAssessmentFallsScreening\_precondition\_26 AS  SELECT t35.\*  FROM caliphr.CMS139v3\_RiskCategoryAssessmentFallsScreening t35  WHERE  t35.negation\_code\_id IN  (  SELECT code\_id  FROM common.VALUE\_SET\_CODE  WHERE  value\_set\_id = 366  )  AND  /\* DURING \*/  t35.effective\_time\_start >= MILLIS\_EPOCH('2013-01-01')  AND t35.effective\_time\_end <= MILLIS\_EPOCH('2013-12-31');  (Source Views Omitted) |
| Example ‑ Generated SQL for all populations of a measure The following example shows all the SQL generated and executed to calculate all the populations, including Initial Patient Population (IPP), Denominator (DENOM), Numerator (NUMER), Denominator Exclusions (DENEX) or Denominator Exceptions (DENEXCEP). Once all the H2 views for the data criteria are generated, they are utilized in the SQL for the population preconditions.  **Human-readable representation:**   * **Initial Patient Population =**   + AND: "Patient Characteristic Birthdate: birth date" >= 23 year(s) starts before start of "Measurement Period"   + AND: "Patient Characteristic Birthdate: birth date" < 64 year(s) starts before start of "Measurement Period"   + AND: "Patient Characteristic Sex: Female"   + AND:     - OR: "Encounter, Performed: Office Visit"     - OR: "Encounter, Performed: Face-to-Face Interaction"     - OR: "Encounter, Performed: Preventive Care Services - Established Office Visit, 18 and Up"     - OR: "Encounter, Performed: Preventive Care Services-Initial Office Visit, 18 and Up"     - OR: "Encounter, Performed: Home Healthcare Services"     - during "Measurement Period" * **Denominator =**   + AND: "Initial Patient Population" * **Denominator Exclusions =**   + AND: "Procedure, Performed: Hysterectomy with No Residual Cervix" ends before or during "Measurement Period" * **Numerator =**   + AND: "Laboratory Test, Result: Pap Test (result)" <= 2 year(s) ends before or during "Measurement Period" * **Denominator Exceptions =**   + None |
| **JSON representation:**  "populationCriteria": [  {  "conjunction": true,  "id": "IPP",  "type": "IPP",  "title": "Initial Patient Population",  "hqmfId": "21D17EA5-67D7-4842-B7F6-3D28CC3E8A69",  "preconditions": [  {  "preconditions": [  {  "reference": "PatientCharacteristicBirthdateBirthDate\_precondition\_1"  },  {  "reference": "PatientCharacteristicBirthdateBirthDate\_precondition\_3"  },  {  "reference": "PatientCharacteristicSexFemale\_precondition\_5"  },  {  "preconditions": [  {  "reference": "EncounterPerformedOfficeVisit\_precondition\_7"  },  {  "reference": "EncounterPerformedFaceToFaceInteraction\_precondition\_9"  },  {  "reference": "EncounterPerformedPreventiveCareServicesEstablishedOfficeVisit18AndUp\_precondition\_11"  },  {  "reference": "EncounterPerformedPreventiveCareServicesInitialOfficeVisit18AndUp\_precondition\_13"  },  {  "reference": "EncounterPerformedHomeHealthcareServices\_precondition\_15"  }  ],  "conjunctionCode": "atLeastOneTrue"  }  ],  "conjunctionCode": "allTrue"  }  ]  },  {  "conjunction": true,  "id": "DENOM",  "type": "DENOM",  "title": "Denominator",  "hqmfId": "7E7CA78D-C3BD-43BD-89DB-7766E5D2213A",  "preconditions": []  },  {  "conjunction": true,  "id": "DENEX",  "type": "DENEX",  "title": "Denominator Exclusions",  "hqmfId": "44829070-FB59-4807-9BBD-A64B37511CD2",  "preconditions": [  {  "preconditions": [  {  "reference": "ProcedurePerformedHysterectomyWithNoResidualCervix\_precondition\_21"  }  ],  "conjunctionCode": "allTrue"  }  ]  },  {  "conjunction": true,  "id": "NUMER",  "type": "NUMER",  "title": "Numerator",  "hqmfId": "6558531E-97D5-44FB-97B7-E5523FB681B8",  "preconditions": [  {  "preconditions": [  {  "reference": "LaboratoryTestResultPapTest\_precondition\_24"  }  ],  "conjunctionCode": "allTrue"  }  ]  }  ] |
| **Generated SQL:**  CREATE TABLE caliphr.IPP\_CMS124v3(patient\_id integer);  INSERT INTO caliphr.IPP\_CMS124v3 (  (  /\* PatientCharacteristicBirthdateBirthDate\_precondition\_1 \*/  SELECT t15.patient\_id  FROM caliphr.CMS124v3\_PatientCharacteristicBirthdateBirthDate\_precondition\_1 t15  /\* allTrue \*/  INTERSECT  /\* PatientCharacteristicBirthdateBirthDate\_precondition\_3 \*/  SELECT t16.patient\_id  FROM caliphr.CMS124v3\_PatientCharacteristicBirthdateBirthDate\_precondition\_3 t16  /\* allTrue \*/  INTERSECT  /\* PatientCharacteristicSexFemale\_precondition\_5 \*/  SELECT t17.patient\_id  FROM caliphr.CMS124v3\_PatientCharacteristicSexFemale\_precondition\_5 t17  /\* allTrue \*/  INTERSECT  (  /\* EncounterPerformedOfficeVisit\_precondition\_7 \*/  SELECT t18.patient\_id  FROM caliphr.CMS124v3\_EncounterPerformedOfficeVisit\_precondition\_7 t18  /\* atLeastOneTrue \*/  UNION  /\* EncounterPerformedFaceToFaceInteraction\_precondition\_9 \*/  SELECT t19.patient\_id  FROM caliphr.CMS124v3\_EncounterPerformedFaceToFaceInteraction\_precondition\_9 t19  /\* atLeastOneTrue \*/  UNION  /\* EncounterPerformedPreventiveCareServicesEstablishedOfficeVisit18AndUp\_precondition\_11 \*/  SELECT t20.patient\_id  FROM caliphr.CMS124v3\_EncounterPerformedPreventiveCareServicesEstablishedOfficeVisit18AndUp\_precondition\_11 t20  /\* atLeastOneTrue \*/  UNION  /\* EncounterPerformedPreventiveCareServicesInitialOfficeVisit18AndUp\_precondition\_13 \*/  SELECT t21.patient\_id  FROM caliphr.CMS124v3\_EncounterPerformedPreventiveCareServicesInitialOfficeVisit18AndUp\_precondition\_13 t21  /\* atLeastOneTrue \*/  UNION  /\* EncounterPerformedHomeHealthcareServices\_precondition\_15 \*/  SELECT t22.patient\_id  FROM caliphr.CMS124v3\_EncounterPerformedHomeHealthcareServices\_precondition\_15 t22  )  ));  SELECT DISTINCT patient\_id  FROM caliphr.IPP\_CMS124v3;  CREATE TABLE caliphr.DENOM\_CMS124v3(patient\_id integer);  INSERT INTO caliphr.DENOM\_CMS124v3 (  SELECT t23.patient\_id  FROM caliphr.IPP\_CMS124v3 t23  );  SELECT DISTINCT patient\_id  FROM caliphr.DENOM\_CMS124v3;  CREATE TABLE caliphr.DENEX\_CMS124v3(patient\_id integer);  INSERT INTO caliphr.DENEX\_CMS124v3 (  SELECT t26.patient\_id  FROM caliphr.DENOM\_CMS124v3 t26  INTERSECT  (  /\* ProcedurePerformedHysterectomyWithNoResidualCervix\_precondition\_21 \*/  SELECT t27.patient\_id  FROM caliphr.CMS124v3\_ProcedurePerformedHysterectomyWithNoResidualCervix\_precondition\_21 t27  ));  SELECT DISTINCT patient\_id  FROM caliphr.DENEX\_CMS124v3;  CREATE TABLE caliphr.NUMER\_CMS124v3(patient\_id integer);  INSERT INTO caliphr.NUMER\_CMS124v3 (  SELECT t30.patient\_id  FROM caliphr.DENOM\_CMS124v3 t30  INTERSECT  (  /\* LaboratoryTestResultPapTest\_precondition\_24 \*/  SELECT t31.patient\_id  FROM caliphr.CMS124v3\_LaboratoryTestResultPapTest\_precondition\_24 t31  ));  DELETE FROM caliphr.NUMER\_CMS124v3  WHERE  patient\_id IN  (  SELECT patient\_id  FROM caliphr.DENEX\_CMS124v3  );  SELECT DISTINCT patient\_id  FROM caliphr.NUMER\_CMS124v3;  (Source Views Omitted) |
| Example ‑ Numerator with Specific Occurrences carried over from Denominator The following example shows a numerator that has a specific occurrence carried over from the denominator (o1), as well as adding two new occurrences (o2 and o3).  **Human-readable representation:**   * **Numerator 1 =**   + AND:     - OR: "Occurrence A of Encounter, Performed: Alcohol and Drug Dependence Treatment"     - OR: "Occurrence A of Encounter, Performed: Psych Visit - Psychotherapy"     - <= 14 day(s) starts after start of "Occurrence A of Diagnosis, Active: Alcohol and Drug Dependence" |
| **JSON representation:**  {  "conjunction": true,  "id": "NUMER",  "type": "NUMER",  "title": "Numerator",  "hqmfId": "BEA1444C-FA23-4741-8659-F8ABCA3644EE",  "preconditions": [  {  "preconditions": [  {  "preconditions": [  {  "reference": "OccurrenceAAlcoholAndDrugDependenceTreatment2\_precondition\_33"  },  {  "reference": "OccurrenceAPsychVisitPsychotherapy3\_precondition\_35"  }  ],  "conjunctionCode": "atLeastOneTrue"  }  ],  "conjunctionCode": "allTrue"  }  ]  } |
| **Generated SQL:**  CREATE TABLE caliphr.NUMER\_CMS137v3(patient\_id integer, ido1 integer, ido2 integer, ido3 integer)  INSERT INTO caliphr.NUMER\_CMS137v3 (  SELECT t33.patient\_id, t33.ido1, o2.id ido2, o3.id ido3  FROM caliphr.DENOM\_CMS137v3 t33  LEFT OUTER JOIN caliphr.CMS137v3\_OccurrenceAAlcoholAndDrugDependenceTreatment2 o2 ON t33.patient\_id = o2.patient\_id  LEFT OUTER JOIN caliphr.CMS137v3\_OccurrenceAPsychVisitPsychotherapy3 o3 ON t33.patient\_id = o3.patient\_id  INTERSECT  (  (  /\* OccurrenceAAlcoholAndDrugDependenceTreatment2\_precondition\_33 \*/  SELECT o2.patient\_id, o2.ido1, o2.ido2, o3.id ido3  FROM caliphr.CMS137v3\_OccurrenceAAlcoholAndDrugDependenceTreatment2\_precondition\_33 o2  LEFT OUTER JOIN caliphr.CMS137v3\_OccurrenceAPsychVisitPsychotherapy3 o3 ON o2.patient\_id = o3.patient\_id  /\* atLeastOneTrue \*/  UNION  /\* OccurrenceAPsychVisitPsychotherapy3\_precondition\_35 \*/  SELECT o3.patient\_id, o3.ido1, o2.id ido2, o3.ido3  FROM caliphr.CMS137v3\_OccurrenceAPsychVisitPsychotherapy3\_precondition\_35 o3  LEFT OUTER JOIN caliphr.CMS137v3\_OccurrenceAAlcoholAndDrugDependenceTreatment2 o2 ON o3.patient\_id = o2.patient\_id  )  ));  (Source Views Omitted) |
| Example ‑ Generated SQL for preconditions that contain NOT in populations For preconditions of the “AND NOT” variety below, the basic approach is to construct the set of all possible patients, and MINUS any patient that satisfies the condition. This set is then intersected with the preconditions that come before it.  **Human-readable representation:**   * **Initial Patient Population =**   + AND: "Occurrence A of Diagnosis, Active: Diabetes" starts before or during "Measurement Period"   + AND NOT: "Occurrence A of Diagnosis, Active: Diabetes" ends before start of "Measurement Period” |
| **JSON representation (simplified):**  {  "conjunction": true,  "id": "IPP",  "type": "IPP",  "title": "Initial Patient Population",  "hqmfId": "EDED90E9-E4FE-47E6-90AC-29D9AA3E861A",  "preconditions": [  {  "preconditions": [  {  "reference": "OccurrenceADiabetes1\_precondition\_4"  },  {  "preconditions": [  {  "reference": "OccurrenceADiabetes1\_precondition\_6"  }  ],  "conjunctionCode": "allFalse"  }  ],  "conjunctionCode": "allTrue"  }  ]  } |
| **Generated SQL (simplified):**  INSERT  INTO caliphr.IPP\_CMS122v3 (  (  /\* OccurrenceADiabetes1\_precondition\_4 \*/  SELECT o1.patient\_id, o1.ido1  FROM caliphr.CMS122v3\_OccurrenceADiabetes1\_precondition\_4 o1  /\* allTrue \*/  INTERSECT  /\* allFalse \*/  (  SELECT t26.patient\_id, o1.id ido1  FROM caliphr.PATIENT\_INFO t26  LEFT OUTER JOIN caliphr.CMS122v3\_OccurrenceADiabetes1 o1 ON t26.patient\_id = o1.patient\_id  MINUS  (  /\* OccurrenceADiabetes1\_precondition\_6 \*/  SELECT o1.patient\_id, o1.ido1  FROM caliphr.CMS122v3\_OccurrenceADiabetes1\_precondition\_6 o1  )  )  ))  (Source Views Omitted) |

# Time Relationships and Data Criteria

Data Criteria can be related to other Data Criteria or the Measure Period via time relationships. The following examples show how an encounter can have a temporal relationship with other Data Criteria or a Measure Period.

* + Patient had a lab test that occurred one year before the most recent encounter.
  + Patient has encounters during the Measure Period where a particular medication was requested.
  + Patient had a diagnosis of disease X within N years of immunization for disease X.

The following table is a list of available temporal relationships:

|  |  |  |
| --- | --- | --- |
| Concept Code | Print Name | Definition |
| CONCURRENT | concurrent with | A relationship in which the source act's effective time is the same as the target act's effective time. |
| DURING | occurs during | A relationship in which the source act's effective time is wholly within the target act's effective time (including end points, as defined in the act's effective times) |
| EAE | ends after end | A relationship in which the source act's effective time ends after the target act's effective time. |
| EAS | ends after start | A relationship in which the source act's effective time ends after the start of the target act. |
| EBE | ends before end | The source act ends before the end of the target act |
| EBS | ends before start | A relationship in which the source act's effective time ends before the start of the target act. |
| ECW | ends concurrent with | A relationship in which the source act's effective time ends with the end of the target act's effective time. |
| ECWS | ends concurrent with start | The source act ends when the target act starts |
| EDU | ends during | A relationship in which the source act ends within the target act's effective time (including end points, as defined in the act's effective times) |
| OVERLAP | overlaps with | A relationship in which the source act's effective time overlaps the target act's effective time in any way. |
| SAE | starts after end of | A relationship in which the source act starts after the end of the target act. |
| SAS | starts after start of | The source act starts after the start of the target act. |
| SBE | starts before end | The source act starts before the end of the target act. |
| SBS | starts before start of | A relationship in which the source act starts before the start of the target act. |
| SCW | starts concurrent with | A relationship in which the source act's effective time starts with the start of the target act's effective time |
| SCWE | starts concurrent with end | The source act starts when the target act ends. |
| SDU | starts during | A relationship in which the source act starts within the target act's effective time (including end points, as defined in the act's effective times) |

## Note how time differences were implemented in H2

The Quality Data Model (QDM) defines how time intervals should be calculated (for specifics, see [**Appendix B - Time Interval Calculation Conventions**](https://ecqi.healthit.gov/system/files/qdm_4_2.pdf)). As these definitions differ significantly from how dates and times are specified by default in SQL, a series of custom H2 functions were implemented to align with QDM definitions (in H2TemporalComparisonUtil.java). The hqmf\_util.js.coffee file of the hqmf2js project was used as a reference, and includes the following functions:

|  |
| --- |
| Function Name |
| MILLIS\_EPOCH |
| YEARS\_DIFFERENCE |
| MONTHS\_DIFFERENCE |
| MINUTES\_DIFFERENCE |
| HOURS\_DIFFERENCE |
| DAYS\_DIFFERENCE |
| WEEKS\_DIFFERENCE |

# Specific Occurrences

The design and implementation of Specific Occurrences was one of the most challenging and complex aspects of measure calculation to implement in SQL. A full discussion of specific occurrences is outside the scope of this document. Refer to HQMF spec and CMS Measure Logic Guidance documents at [this link](https://www.cms.gov/regulations-and-guidance/legislation/ehrincentiveprograms/ecqm_library.html) for more details. Below is a description how specific occurrence logic was implemented in SQL.

Specific Occurrences represent a single instance of an event, and all the logic of the measure must hold for that single event. Furthermore measures can reference multiple specific occurrences of the same type or of different types. The basic method taken in our SQL is to produce a set of candidate vales that evaluates the measure against every combination of specific occurrences. Through a series of inner and outer joins, the valid candidate combinations of occurrences remain.

# Patient-Based vs Episode-of-Care measures

Sections 2.1 and 2.2 of the [**Clinical Quality eMeasure Logic and Implementation Guidance**](https://www.cms.gov/Regulations-and-Guidance/Legislation/EHRIncentivePrograms/Downloads/2014_eCQM_Measure_Logic_Guidance.pdf) document from CMS makes a distinction between Patient-Based and Episode-of-Care measures. Briefly:

* Measures that evaluate the care of a patient, and assign the patient to membership in one or more populations are called patient‐based measures. The vast majority (57 out of 64) of the EP CQMs are patient‐based.
* Measures that evaluate the care during a patient‐provider encounter, sometimes called an episode of care, and assign the episode of care to one or more populations are called episode‐of‐care measures. 7 of the 64 of the EP measures are of this type.

As there is no clear indication within the HQMF file that specifies whether a measure is patient-based or episode-of-care, the corresponding measure.metadata file within the measure bundle is used to make the determination. Specifically, every measure has an episode\_of\_care variable. In the case of a value of true, the episode\_ids variable is also used as the indication of which data criteria constitute an episode, to be added to the SELECT DISTINCT clause generated for a given population.

The following is an example of the measure.metadata content of an episode-of-care measure, CMS157v3:

|  |
| --- |
| {  "nqf\_id": "0384",  "type": "ep",  "category": "Cancer",  "episode\_of\_care": true,  "continuous\_variable": false,  "episode\_ids": [  "OccurrenceARadiationTreatmentManagement1",  "OccurrenceAOfficeVisit2",  "OccurrenceAFaceToFaceInteraction5"  ],  "custom\_functions": null,  "force\_sources": null  }  After evaluating each population (like the IPP) the following SQL is used to gather all the episodes. (Note that ido6, ido2 and ido5 correspond to the data criteria represented by the episode ids.)  SELECT DISTINCT patient\_id, ido6, ido2, ido5  FROM caliphr.IPP\_CMS157v3;  Following this, some java logic is executed on the result set so that only one row for a given episode id is retained. |

# Continuous Variable Measures

Note that the current implementation only supports Proportion Measures. Continuous Variable Measures are deferred to a future release of CAliPHR.