Data Quality Check Type Definitions

Clair Blacketer

2024-12-24

Contents

1 Introduction

The DataQualityDashboard functions by applying 20 parameterized check types to a CDM instance, resulting in over 3,351 resolved, executed, and evaluated individual data quality checks. For example, one check type might be written as

*The number and percent of records with a value in the **cdmFieldName** field of the **cdmTableName** table less than **plausibleValueLow**.*

This would be considered an atemporal plausibility verification check because we are looking for implausibly low values in some field based on internal knowledge. We can use this check type to substitute in values for **cdmFieldName**, **cdmTableName**, and **plausibleValueLow** to create a unique data quality check. If we apply it to PERSON.YEAR_OF_BIRTH here is how that might look:

*The number and percent of records with a value in the **year_of_birth** field of the **PERSON** table less than **1850**.*

And, since it is parameterized, we can similarly apply it to DRUG_EXPOSURE.days_supply:

*The number and percent of records with a value in the **days_supply** field of the **DRUG_EXPOSURE** table less than **0**.*

Version 1 of the tool includes 20 different check types organized into Kahn contexts and categories (link to paper). Additionally, each data quality check type is considered either a table check, field check, or concept-level check. Table-level checks are those evaluating the table at a high-level without reference to individual fields, or those that span multiple event tables. These include checks making sure required tables are present or that at least some of the people in the PERSON table have records in the event tables. Field-level checks are those related to specific fields in a table. The majority of the check types in version 1 are field-level checks. These include checks evaluating primary key relationship and those investigating if the concepts in a field conform to the specified domain. Concept-level checks are related to individual concepts. These include checks looking for gender-specific concepts in persons of the wrong gender and plausible values for measurement-unit pairs.

This article will detail each check type, its name, check level, description, definition, and to which Kahn context, category, and subcategory it belongs.

1.1 measurePersonCompleteness

Name: measurePersonCompleteness **Level**: Table check **Context**: Validation **Category**: Completeness

Description: The number and percent of persons in the CDM that do not have at least one record in the **cdmTableName** table.

Definition: For each table indicated this check will count the number of persons from the PERSON table that do not have at least one record in the specified clinical event table. It may be that there are 100 persons listed in the PERSON table but only 30 of them have at least one record in the MEASUREMENT table. If the **measurePersonCompleteness** check is indicated for the MEASUREMENT table, the result will be 70%, meaning that 70% of the persons in the CDM instance do not have at least one record in MEASUREMENT.

1.2 cdmField

Name: cdmField **Level**: Field check **Context**: Verification **Category**: Conformance **Subcategory**: Relational

Description: A value indicating if all fields are present in the **cdmTableName** table.

Definition: For each table indicated this check will go through and determine if all fields are present as specified based on the CDM version. If the field is present, the resulting value will be 0; if the field is absent the resulting value will be 100.

1.3 isRequired

Name: isRequired **Level**: Field check **Context**: Validation **Category**: Conformance **Subcategory**: Relational

Description: The number and percent of records with a NULL value in the **cdmFieldName** of the **cdmTableName** that is considered not nullable

Definition: This check is meant to ensure that all NOT NULL constraints specified in the CDM version are followed. It will count up all records with a NULL value in the specified field of the specified table and return the percent of records in the table that violate the constraint.

1.4 cdmDatatype

Name: cdmDatatype **Level**: Field check **Context**: Verification **Category**: Conformance **Subcategory**: Value

Description: A yes or no value indicating if the **cdmFieldName** in the **cdmTableName** is the expected data type based on the specification.

Definition: At present this will check only that fields that are supposed to be integers are the expected datatype. For a given field, it will count the number of records with a non-null, non-integer value.

1.5 isPrimaryKey

Name: isPrimaryKey **Level**: Field check **Context**: Verification **Category**: Conformance **Subcategory**: Relational

Description: The number and percent of records that have a duplicate value in the **cdmFieldName** field of the **cdmTableName**.

Definition: This check will make sure that all primary keys as specified in the CDM version are truly unique values in the database. While this should be caught by primary key constraints, some database management systems such as redshift do not enforce these.

1.6 isForeignKey

Name: isForeignKey **Level**: Field check **Context**: Verification **Category**: Conformance **Subcategory**: Relational

Description: The number and percent of records that have a value in the **cdmFieldName** field in the **cdmTableName** table that does not exist in the **fkTableName** table.

Definition: This check will make sure that all foreign keys as specified in the CDM version have a value in the related primary key field. While this should be caught by foreign key constraints, some database management systems such as redshift do not enforce these.

1.7 fkDomain

Name: fkDomain **Level**: Field check **Context**: Verification **Category**: Conformance **Subcategory**: Value

Description: The number and percent of records that have a value in the **cdmFieldName** field in the **cdmTableName** table that do not conform to the **fkDomain** domain.

Definition: It is often the case that standard concept fields in the OMOP CDM should belong to a certain domain. All possible domains are listed in the vocabulary table DOMAIN and the expected domain for CDM fields are listed as part of the CDM documentation. For example, in the field PERSON.gender_concept_id all concepts in that field should conform to the *gender* domain. This check will search all concepts in a field and count the number of records that have concepts in the field that do not belong to the correct domain.

1.8 fkClass

Name: fkClass **Level**: Field check **Context**: Verification **Category**: Conformance **Subcategory**: Computational

Description: The number and percent of records that have a value in the **cdmFieldName** field in the **cdmTableName** table that do not conform to the **fkClass** class.

Definition: There is the occasional field in the OMOP CDM that expects not only concepts of a certain domain, but of a certain concept class as well. The best example is the drug_concept_id field in the DRUG_ERA table. Drug eras represent the span of time a person was exposed to a particular drug *ingredient* so all concepts in DRUG_ERA.drug_concept_id are of the drug domain and ingredient class. This check will search all concepts in a field and count the number of records that have a concept in the field that do not belong to the correct concept class.

1.9 isStandardValidConcept

Name: isStandardValidConcept **Level**: Field check **Context**: Verification **Category**: Conformance

Description: The number and percent of records that do not have a standard, valid concept in the *cdmFieldName* field in the *cdmTableName* table.

Definition: In order to standardize not only the structure but the vocabulary of the OMOP CDM, certain fields in the model require standard, valid concepts while other fields do not. For example, in the PERSON table, the field gender_concept_id MUST be a standard, valid concept: either 8532 or 8507. In contrast the field gender_source_concept_id can be any concept, standard or no. This check will count the number of records that have a concept in a given field that are not standard and valid.

1.10 measureValueCompleteness

Name: measureValueCompleteness **Level**: Field check **Context**: Verification **Category**:Completeness

Description: The number and percent of records with a NULL value in the *cdmFieldName* of the *cdmTableName.*

Definition: This check will count the number of records with a NULL value in a specified field. This is different from the isRequired check because it will run this calculation for all tables and fields whereas the isRequired check will only run for those fields deemed required by the CDM specification. Often the thresholds for failure are set at different levels between these checks as well.

1.11 standardConceptRecordCompleteness

Name: standardConceptRecordCompleteness **Level**: Field check **Context**: Verification **Category**: Completeness

Description: The number and percent of records with a value of 0 in the standard concept field *cdmFieldName* in the *cdmTableName* table.

Definition: It is important to understand how well source values were mapped to standard concepts. This check will count the number of records in a standard concept field (condition_concept_id, drug_concept_id, etc.) with a value of 0 rather a standard concept. NOTE for the field unit_concept_id in the MEASUREMENT and OBSERVATION tables both the numerator and denominator are limited to records where value_as_number is not null. This prevents over-inflation of the numbers and focuses the check to records that are eligible for a unit value.

1.12 sourceConceptRecordCompleteness

Name: sourceConceptRecordCompleteness **Level**: Field check **Context**: Verification **Category**: Completeness

Description: The number and percent of records with a value of 0 in the source concept field *cdmFieldName* in the *cdmTableName* table.

Definition: This check will count the number of records in a source concept field (condition_source_concept_id, drug_source_concept_id) with a value of 0. This is useful since source values that are represented by concepts in the vocabulary have automatic mappings to standard concepts. Using this check along with the standardConceptRecordCompletness check can help identify any vocabulary mapping issues during ETL.

1.13 sourceValueCompleteness

Name: sourceValueCompleteness **Level**: Field check **Context**: Verification **Category**: Completeness

Description: The number and percent of distinct source values in the **cdmFieldName** field of the **cdmTableName** table mapped to 0.

Definition: This check will look at all distinct source values in the specified field and calculate how many are mapped to 0. This should be used in conjunction with the standardConceptRecordCompletness check to identify any mapping issues in the ETL.

1.14 plausibleValueLow - (for Fields)

Name: plausibleValueLow **Level**: Field check **Context**: Verification **Category**: Plausibility **Subcategory**: Atemporal

Description: The number and percent of records with a value in the **cdmFieldName** field of the **cdmTableName** table less than **plausibleValueLow**.

Definition: This check will count the number of records that have a value in the specified field that is lower than some value. This is the field-level version of this check so it is not concept specific. For example, it will count the number of records that have an implausibly low value in the year_of_birth field of the PERSON table.

1.15 plausibleValueHigh - (for Fields)

Name: plausibleValueHigh **Level**: Field check **Context**: Verification **Category**: Plausibility **Subcategory**: Atemporal

Description: The number and percent of records with a value in the **cdmFieldName** field of the **cdmTableName** table greater than **plausibleValueHigh**.

Definition: This check will count the number of records that have a value in the specified field that is higher than some value. This is the field-level version of this check so it is not concept specific. For example, it will count the number of records that have an implausibly high value in the year_of_birth field of the PERSON table.

1.16 plausibleTemporalAfter

Warning: This check is reimplemented by the `plausibleStartBeforeEnd` and `plausibleAfterBirth` checks, and will be deprecated in the future.

Name: plausibleTemporalAfter **Level**: Field check **Context**: Verification **Category**: Plausibility **Subcategory**: Temporal

Description: The number and percent of records with a value in the **cdmFieldName** field of the **cdmTableName** that occurs prior to the date in the **plausibleTemporalAfterFieldName** field of the **plausibleTemporalAfterTableName** table.

Definition: This check is attempting to apply temporal rules to a CDM instance. For example, it will check to make sure that all visit records for a person in the VISIT_OCCURRENCE table occur after the person's birth.

1.17 plausibleDuringLife

Warning: This check is reimplemented by the `plausibleBeforeDeath` check, and will be deprecated in the future.

Name: plausibleDuringLife **Level**: Field check **Context**: Verification **Category**: Plausibility **Subcategory**: Temporal

Description: If yes, the number and percent of records with a date value in the **cdmFieldName** field of the **cdmTableName** table that occurs after death.

Definition: This check will calculate the number of records that occur after a person's death. This is called *plausibleDuringLife* because turning it on indicates that the specified dates should occur during a person's lifetime, like drug exposures, etc.

1.18 plausibleStartBeforeEnd

Name: plausibleStartBeforeEnd **Level**: Field check **Context**: Verification **Category**: Plausibility **Subcategory**: Temporal

Description: The number and percent of records with a value in the **cdmFieldName** field of the **cdmTableName** that occurs after the date in the **plausibleStartBeforeEndFieldName**.

Definition: This check is attempting to apply temporal rules within a table, specifically checking that all start dates are before the end dates. For example, in the VISIT_OCCURRENCE table it checks that the VISIT_OCCURRENCE_START_DATE is before VISIT_OCCURRENCE_END_DATE.

1.19 plausibleAfterBirth

Name: plausibleAfterBirth **Level**: Field check **Context**: Verification **Category**: Plausibility **Subcategory**: Temporal

Description: The number and percent of records with a date value in the **cdmFieldName** field of the **cdmTableName** table that occurs prior to birth.

Definition: This check will calculate the number of records that occur before a person's birth. If the `person.birth_datetime` is given this is used. If not, it is calculated from the `person.year_of_birth`, `person.month_of_birth`, and `person.day_of_birth`, taking the first month of the year, day of the month if the respective value is not given. For example, if only year of birth is given, the birth date is assumed to be January 1st of that year.

1.20 plausibleBeforeDeath

Name: plausibleBeforeDeath **Level**: Field check **Context**: Verification **Category**: Plausibility **Subcategory**: Temporal

Description: The number and percent of records with a date value in the **cdmFieldName** field of the **cdmTableName** table that occurs after death.

Definition: This check will calculate the number of records that occur more than 60 days after a person's death. The 60 day 'washout' period is to allow for administrative records after death.

1.21 plausibleValueLow - (for Concept + Unit combinations)

Name: plausibleValueLow **Level**: Concept check **Context**: Verification **Category**: Plausibility **Subcategory**: Atemporal

Description: For the combination of CONCEPT_ID **conceptId** (**conceptName**) and UNIT_CONCEPT_ID **unitConceptId** (**unitConceptName**), the number and percent of records that have a value less than **plausibleValueLow**.

Definition: This check will count the number of records that have a value in the specified field with the specified concept_id and unit_concept_id that is lower than some value. This is the concept-level version of this check so it is concept specific and therefore the denominator will only be the records with the specified concept and unit. For example, it will count the number of records that have an implausibly low value in the value_as_number field of the MEASUREMENT table where MEASUREMENT_CONCEPT_ID = 2212241 (Calcium; total) and UNIT_CONCEPT_ID = 8840 (milligram per deciliter). These implausible values were determined by a team of physicians and are meant to be *biologically implausible*, not just lower than the normal value.

1.22 plausibleValueHigh - (for Concept + Unit combinations)

Name: plausibleValueHigh **Level**: Concept check **Context**: Verification **Category**: Plausibility **Subcategory**: Atemporal

Description: For the combination of CONCEPT_ID **conceptId** (**conceptName**) and UNIT_CONCEPT_ID **unitConceptId** (**unitConceptName**), the number and percent of records that have a value higher than **plausibleValueHigh**.

Definition: This check will count the number of records that have a value in the specified field with the specified concept_id and unit_concept_id that is higher than some value. This is the concept-level version of this check so it is concept specific and therefore the denominator will only be the records with the specified concept and unit. For example, it will count the number of records that have an implausibly high value in the value_as_number field of the MEASUREMENT table where MEASUREMENT_CONCEPT_ID = 2212241 (Calcium; total) and UNIT_CONCEPT_ID = 8840 (milligram per deciliter). These implausible values were determined by a team of physicians and are meant to be *biologically implausible*, not just higher than the normal value.

1.23 plausibleGender

Warning: This check is reimplemented by the `plausibleGenderUseDescendants` check below, and will be deprecated in the future.

Name: plausibleGender **Level**: Concept check **Context**: Validation **Category**: Plausibility **Subcategory**: Atemporal

Description: For a CONCEPT_ID **conceptId** (**conceptName**), the number and percent of records associated with patients with an implausible gender (correct gender = **plausibleGender**).

Definition: This check will count the number of records that have an incorrect gender associated with a gender-specific concept_id. This check is concept specific and therefore the denominator will only be the records with the specified concept. For example it will count the number of records of prostate cancer that are associated with female persons.

1.24 plausibleGenderUseDescendants

Name: plausibleGenderUseDescendants **Level**: Concept check **Context**: Validation **Category**: Plausibility **Subcategory**: Atemporal

Description: For descendants of CONCEPT_ID **conceptId** (**conceptName**), the number and percent of records associated with patients with an implausible gender (correct gender = **plausibleGenderUseDescendants**).

Definition: This check will count the number of records that have an incorrect gender associated with a gender-specific concept_id. In this new implementation, the base concept_id is one or more broad ancestor concepts such as "Procedure on female genital system". Any record with a descendant of this concept_id will be checked for an associated person gender matching the gender-specific concept.

1.25 plausibleUnitConceptIds

Name: plausibleUnitConceptIds **Level**: Concept check **Context**: Verification **Category**: Plausibility **Subcategory**: Atemporal

Description: The number and percent of records for a given CONCEPT_ID **conceptId** (**conceptName**) with implausible units (i.e. UNIT_CONCEPT_ID NOT IN (**plausibleUnitConceptIds**))

Definition: This check will count the number of records for a given concept that do not use one of the allowable units, which is represented in the csv file as a quoted comma-separated list of unit_concept_ids. If no units are specified for **plausibleUnitConceptIds**, this check will count the number of records that have non-NULL units. This check is concept specific and therefore the denominator will only be the records with the specified concept. For example it will count the number of records or **Glomerular Filtration Rate** (CONCEPT_ID = 3030354) that do not use unit **mL/min/1.73.m2** (UNIT_CONCEPT_ID = 9117).