Process Design for the Enhanced Observational Medical Dataset Simulator (OSIM 2) v1.5.005

UBC®

United BioSource Corporation

Evidence Matters®

OBSERVATIONAL MEDICAL OUTCOMES PARTNERSHIP

A PUBLIC PRIVATE RESEARCH PARTNERSHIP OF THE FNIH

Last Revised: 03 January 2011

Table of Contents

Introduction to the Observational Medical Dataset Simulator Project

To facilitate its methodological research, the Observational Medical Outcomes Partnership (OMOP) will evaluate the performance of various analytical methods identifying drug-outcome associations across multiple disparate observational data sources (administrative claims and electronic health records). Methodological research typically requires some benchmark or 'gold standard' against which to measure performance. In this context, a desired gold standard would be a true causal relationship between a drug and a health outcome. Unfortunately, most observational data sources are poorly characterized, clinical observations may be insufficiently recorded or poorly validated, and actual 'truth' may not be absolutely determined. 'True relationships' between drugs and outcomes may be difficult to ascertain as these 'known associations' may be affected by issues including sample size, adequacy of data capture, and confounding.

Because of these issues and the desire to have a common, acceptable test set, the Partnership is designing and developing an automated procedure to construct simulated datasets to supplement the methods evaluation. The simulated datasets will be modeled after real observational data sources, but will be comprised of hypothetical persons with fictional drug exposure and health outcomes occurrence. The datasets will be constructed such that the relationships between the fictional drugs and fictional outcomes are well-characterized as 'true' and 'false' associations. That is, hypothetical persons will be created and assigned fictional drug exposure periods and instances of health outcomes based on random sampling from probability distributions that define the relationships between the fictional drugs and outcomes. The relationships created within the simulated datasets will be contrived, but will be representative of the types of relationships expected to be observed within real observational data sources.

The simulated datasets will only be used to perform statistical evaluations of the analytical methods offered to identify drug-outcome associations. The performance characteristics (sensitivity, specificity, positive and negative predictive value) of the analytical methods can then be empirically measured in terms of the known characteristics of the data will enable the classification of the drug-outcome relationships as 'true' or 'false' and methods will be executed to classify the drug-outcome pairs as 'positive' or 'negatives'. Because the simulated data will represent hypothetical patients, fictional drug classes and artificial outcomes types, there can be no clinical interpretations drawn from the data. *In addition, positive and negative predictive value will be defined and measured in terms of statistical associations, not necessarily based on medical or clinical relationships between drugs and outcomes (as statistical associations in the data could be caused by many factors).*

These same simulated datasets will be made publicly available to encourage methods development from the broader research community and to enable researchers to evaluate the performance of their methods for themselves against the findings from the methods tested in OMOP. Researchers will be encouraged to provide the methods implementation and summary results to the Research Core so that the methods can be incorporated into additional studies against the centralized and distributed datasets in conjunction with the simulation data. Because the simulated data structure, including table names, field names and data types, will conform to the Partnership's Common Data Model design, all methods implementations develop should be portable between the real and simulated datasets.

The successful development of the Observational Medical Dataset Simulator (OSIM; http://omop.fnih.org/OSIM) has enabled methods development across a broad research community. OSIM can construct many millions of hypothetical patients with drug exposure, background conditions, and known adverse events that can be used to benchmark methods performance. OSIM has provided access to large-scale data to methodologists, and facilitated the establishment of the OMOP Cup Competition. It has also advanced the OMOP Research Team's learnings about the complex interdependencies between clinical observations in real data, and how those relationships may influence a method's behavior in identifying true associations and discerning from false positive findings. Based on these learnings, the Principal Investigators have recommended that OMOP continue research and development into a second-generation simulated dataset procedure. OSIM 2 establishes a complementary model to the original OSIM program, applying an alternative design to accommodate additional complexities observed in real-world data, including advanced modeling of the correlations between drugs and conditions. OSIM 2 should allow for more direct comparisons between simulated data and real observational databases, and should enable greater methods evaluation by allowing assessment of how methods accommodate these complex interrelationships.

The OSIM 2 program further implements this vision. This document describes the process design for OSIM 2.

System Architecture

Design Environment

The Observational Medical Data Simulation program consists of an Oracle Package, written in the Oracle PL/SQL programming language. It compiles and runs on a wide variety of operating systems hosting an Oracle database.

High-Level Architecture Overview

Figure 1 below illustrates the high-level architecture and major components of the OSIM 2 Oracle package. Execution of the procedures is controlled by user-defined input parameters. The package itself consists of two main procedures that can be executed together or separately, as specified by input parameters. The final output tables contain Simulated Drugs, Conditions, and Persons produced by the simulation execution. A detailed description of the input parameters and output files can be found in the *Data Dictionary for the Observational Medical Dataset Simulator 2 (OSIM 2).*

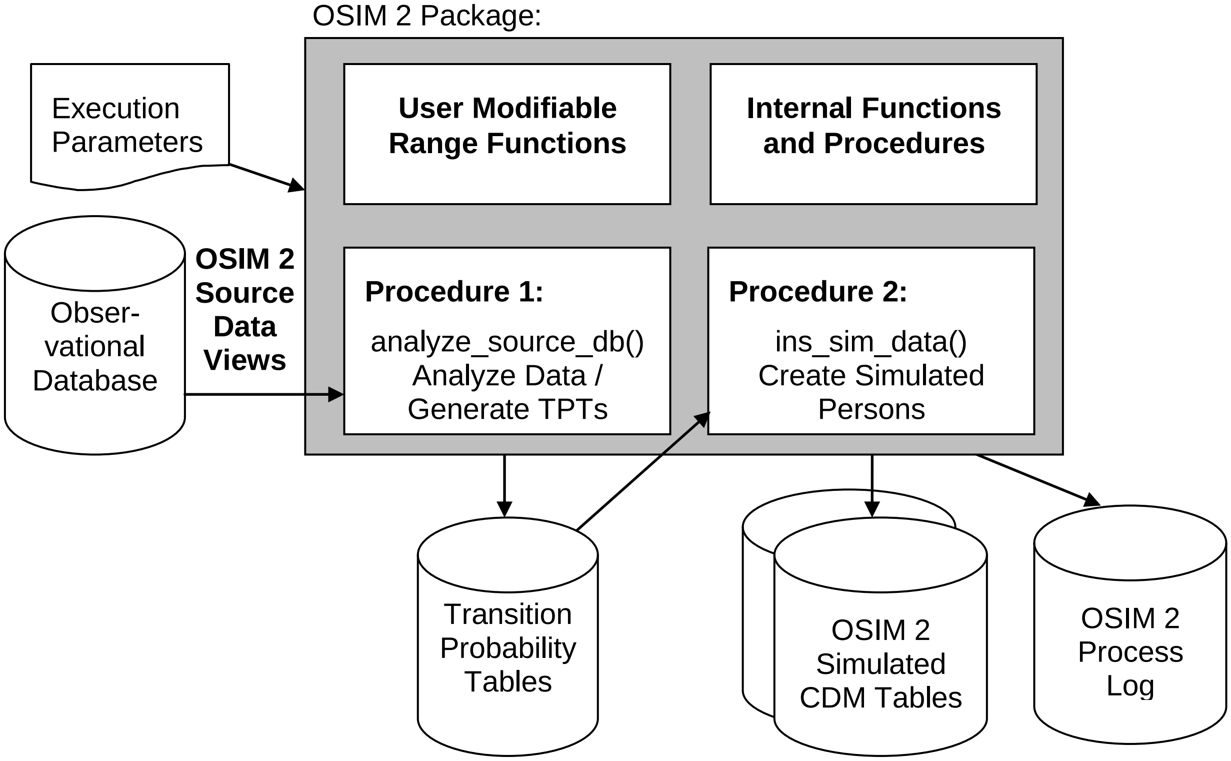


Figure 1: OSIM 2 High-Level Architecture

Performance Design Considerations

The two main performance considerations that have affected the design of OSIM are:

- OSIM may generate very large number of Simulated Persons (millions of persons)
- OSIM can run on any Oracle database server

To accommodate the efficient and timely generation of a large number of simulated records using a variety of high performance and lower-end platforms, the package has been designed for both local and distributed execution modes. In local execution mode, the Simulation Transition Probability Tables that control the characteristics of the Simulated Persons are generated in the same run as the Simulated Persons. In a distributed mode run, the Simulation Transition Probability Tables are generated first, in an initial execution. These Simulation Transition Probability Tables are used as input to multiple, distributed executions of the OSIM 2 simulation procedure to generate Simulated Persons. Since the distributed Simulated Person Tables were populated using the same input Tables, the simulated data can be concatenated into one large Simulated Person Table at the conclusion of the distributed runs. Figure 2 below illustrates a distributed execution mode of OSIM 2.

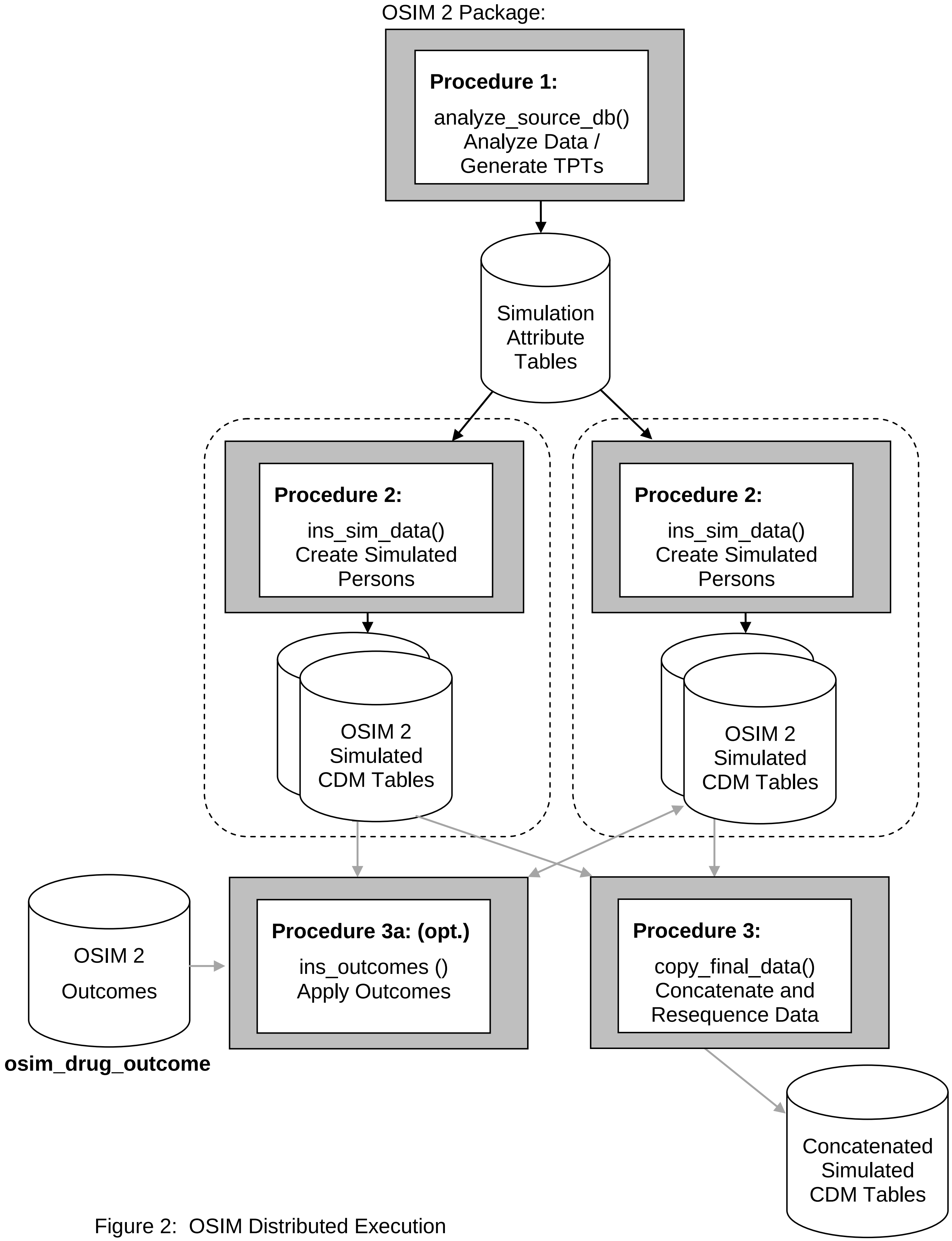


Figure 2: OSIM Distributed Execution

Detailed Description of OSIM 2 Components

Background

To allow the simulated dataset to approximate characteristics of real observational data, Procedure 1 (analyze_source_db()) performs a preliminary analysis of the CDM database. Attributes and transition probabilities are examined for:

Database Attributes

- Number of Persons in the database
- Number of Condition Eras in the database
- Number of Drug Eras in the database
- Drug and Condition Era Type (persistence window code)
- Minimum and Maximum Date of Era or Observation in the database

Transition Probabilities

The probability transitions are commonly expressed in formula form:

Fn(Result(s) | Strata)

- Fn(gender_concept_id)
- Fn(age_at_obs | gender_concept_id)
- Fn(cond_concept_count | gender_concept_id, age_at_obs)
- Fn(time_observed | gender_concept_id, age_at_obs, cond_count_bucket)
- Fn(cond_era_count | condition_concept_id, cond_count_bucket, time_remaining)
- Fn(drug_count | gender_concept_id, age_range, cond_count_bucket)
- Fn(drug_draw_count | condition_concept_id, interval_bucket, age_range, drug_count_bucket, cond_count_bucket)
- Fn(drug_era_count, total_exposure | drug_concept_id, drug_count_bucket, condition_count_bucket, age_range, time_remaining)
- Fn(total_duration | drug_concept_id, time_remaining, drug_era_count, total_exposure)
- Fn(days_until | condition_concept_id, age_range, time_remaining)
- Fn(condition2_concept_id, delta_days | gender_concept_id, age_range, cond_count_bucket, time_remaining, condition1_concept_id)
- Fn(drug_concept_id, delta_days | condition_concept_id, interval_bucket, gender_concept_id, age_range, condition_count_bucket, drug_count_bucket, day_cond_count)

These probability distributions, which are described in detail in the *Data Dictionary for the Observational Medical Dataset Simulator,* are inserted into tables for use by the simulation procedure to generate the Simulated Persons data.

OSIM 2 CDM Source Data Synonyms and Views

The OSIM 2 database analysis uses a set of read-only views to access the data in the source CDM database. These views can be modified to limit or refine the analysis, and in particular for choosing the persistence window of the eras. The views access the source CDM database through synonyms to the person, observation period, drug era, and condition era tables.

OSIM 2 Source Data Synonyms

The synonyms must be created in the target schema before running the OSIM 2 analysis.

```
--
-- Example OSIM 2 CDM Synonyms
--
CREATE OR REPLACE SYNONYM s_person FOR mslr_cdm.person;
/
CREATE OR REPLACE SYNONYM s_condition_era FOR mslr_cdm.omop_condition_era;
/
CREATE OR REPLACE SYNONYM s_observation_period FOR mslr_cdm.observation_period;
/
CREATE OR REPLACE SYNONYM s_drug_era FOR mslr_cdm.omop_drug_era;
/
COMMIT;
/
```

OSIM 2 Source Data Views

The views must be created in the target schema before running the OSIM 2 analysis. The views must be created in order because of inter-dependency.

VIEW v_src_person

```
--================================================================================
-- VIEW v_src_person
--================================================================================
CREATE OR REPLACE VIEW v_src_person
 (person_id, year_of_birth, gender_concept_id, race_concept_id,
  location_concept_id, source_person_key, source_gender_code,
  source_location_code, source_race_code) AS
SELECT /*+ NO_PARALLEL(person) */DISTINCT
  person.person_id,person.year_of_birth, person.gender_concept_id,
  person.race_concept_id, person.location_concept_id, person.source_person_key,
  person.source_gender_code, person.source_location_code, person.source_race_code
FROM s_person person
INNER JOIN s_observation_period period on person.person_id = period.person_id
WHERE person.year_of_birth IS NOT NULL
  AND period.observation_period_start_date IS NOT NULL
WITH READ ONLY;
```

VIEW v_src_person_strata

```
--================================================================================
-- VIEW v_src_person_strata
--================================================================================
CREATE OR REPLACE VIEW v_src_person_strata
 (person_id, year_of_birth, gender_concept_id, race_concept_id,
  location_concept_id, source_person_key, source_gender_code,
  source_location_code, source_race_code, observation_period_start_date,
  observation_period_end_date, age, obs_duration_days, condition_concepts,
  drug_concepts) AS
WITH drug_counts AS
 (SELECT
    person.person_id,
    COUNT (DISTINCT drug_concept_id) AS drugs
  FROM v_src_person person
  INNER JOIN s_drug_era drug_era ON person.person_id = drug_era.person_id
  WHERE drug_exposure_type = '7' -- SET PERSISTENCE WINDOW
  GROUP BY person.person_id),
cond_counts AS
 (SELECT
    person.person_id,
    COUNT (DISTINCT condition_concept_id) AS conditions
  FROM v_src_person person
  INNER JOIN s_condition_era condition_era
    ON person.person_id = condition_era.person_id
  WHERE condition_occurrence_type = '65' -- SET PERSISTENCE WINDOW
  GROUP BY person.person_id),
person_strata AS
 (SELECT /*+ NO_PARALLEL (person) */
    person.person_id, person.year_of_birth, person.gender_concept_id,
    person.race_concept_id, person.location_concept_id, person.source_person_key,
    person.source_gender_code, person.source_location_code, person.source_race_code,
    MIN(period.observation_period_start_date) AS observation_period_start_date,
    MAX(period.observation_period_end_date) AS observation_period_end_date,
    TO_NUMBER(TO_CHAR(MIN(period.observation_period_start_date),'yyyy'))
        - person.year_of_birth as age,
    MAX(period.observation_period_end_date)
        - MIN(period.observation_period_start_date) AS obs_duration_days
  FROM v_src_person person
  INNER JOIN s_observation_period period on person.person_id = period.person_id
  GROUP BY   person.person_id, person.year_of_birth, person.gender_concept_id,
    person.race_concept_id, person.location_concept_id, person.source_person_key,
    person.source_gender_code, person.source_location_code, person.source_race_code)
SELECT strata.person_id, strata.year_of_birth, strata.gender_concept_id,
  strata.race_concept_id, strata.location_concept_id, strata.source_person_key,
  strata.source_gender_code, strata.source_location_code, strata.source_race_code,
  strata.observation_period_start_date, strata.observation_period_end_date,
  strata.age, strata.obs_duration_days,
  NVL(cond.conditions,0) AS condition_concepts,
  NVL(drug.drugs,0) AS drug_concepts
FROM person_strata strata
  LEFT JOIN cond_counts cond ON strata.person_id = cond.person_id
  LEFT JOIN drug_counts drug ON strata.person_id = drug.person_id
WITH READ ONLY;
```

VIEW v_observation_period

```
--================================================================================
-- VIEW v_observation_period
--================================================================================
CREATE OR REPLACE VIEW v_src_observation_period
 (observation_period_id,   observation_period_start_date, observation_period_end_date,
  person_id, person_status_concept_id, rx_data_availability, dx_data_availability,
  hospital_data_availability, confidence) AS
SELECT /*+ NO_PARALLEL(obs) */ obs.observation_period_id,
  obs.observation_period_start_date,
  obs.observation_period_end_date,
  obs.person_id,
  obs.person_status_concept_id,
  obs.rx_data_availability,
  obs.dx_data_availability,
  obs.hospital_data_availability,
  obs.confidence
FROM s_observation_period obs
INNER JOIN v_src_person person ON obs.person_id = person.person_id
WITH READ ONLY;
```

VIEW v_src_condition_era1_ids

```
CREATE OR REPLACE VIEW v_src_condition_era1_ids
 (condition_occurrence_id, condition_occurrence_count) AS
SELECT /*+ NO_PARALLEL(cond) */ DISTINCT
  condition_era_id, cond.condition_occurrence_count
FROM s_condition_era cond
INNER JOIN v_src_person person ON cond.person_id = person.person_id
WHERE condition_occurrence_type = '65' -- SET PERSISTENCE WINDOW
WITH READ ONLY;
```

VIEW v_src_condition_era1

```
--================================================================================
-- VIEW v_src_condition_era1
--================================================================================
CREATE OR REPLACE VIEW v_src_condition_era1
  (condition_era_id, condition_era_start_date, person_id, condition_era_end_date,
   condition_concept_id, condition_occurrence_type, condition_occurrence_count) AS
SELECT /*+ NO_PARALLEL(cond) */
  condition_era_id, condition_era_start_date, cond.person_id, condition_era_end_date,
  condition_concept_id, condition_occurrence_type, condition_occurrence_count
FROM s_condition_era cond
INNER JOIN v_src_person person ON cond.person_id = person.person_id
WHERE condition_occurrence_type = '65' -- SET PERSISTENCE WINDOW
WITH READ ONLY;
```

VIEW v_src_first_conditions

```
CREATE OR REPLACE VIEW v_src_first_conditions
(person_id, condition_era_start_date, condition_concept_id) AS
SELECT DISTINCT
  cond.person_id,
  FIRST_VALUE(condition_era_start_date)
    OVER
    (PARTITION BY cond.person_id, cond.condition_concept_id
      ORDER BY cond.condition_era_start_date) AS condition_era_start_date,
  condition_concept_id
FROM v_src_condition_era1 cond
INNER JOIN s_person person  ON cond.person_id = person.person_id
GROUP BY cond.person_id, condition_era_start_date,  condition_concept_id
WITH READ ONLY;
```

VIEW v_all_conditions

```
CREATE OR REPLACE VIEW v_src_all_conditions
(person_id, gender_concept_id, age, condition_era_start_date, condition_concept_id) AS
  SELECT DISTINCT
    person.person_id,
    gender_concept_id,
    TO_NUMBER(TO_CHAR(condition_era_start_date,'yyyy')) - year_of_birth AS age,
    condition_era_start_date,
    condition_concept_id
  FROM v_src_condition_era1 cond
  INNER JOIN v_src_person person ON cond.person_id = person.person_id
WITH READ ONLY;
```

VIEW v_src_drug_era1

```
CREATE OR REPLACE VIEW v_src_drug_era1
  (drug_era_id, drug_era_start_date, drug_era_end_date, person_id,
   drug_exposure_type, drug_concept_id, drug_exposure_count) AS
SELECT /*+ NO_PARALLEL(drug) */
  drug_era_id, drug_era_start_date, drug_era_end_date, drug.person_id,
  drug_exposure_type, drug_concept_id, drug_exposure_count
FROM s_drug_era drug
INNER JOIN v_src_person person ON drug.person_id = person.person_id
WHERE drug_exposure_type = '7' -- SET PERSISTENCE WINDOW
WITH READ ONLY;
```

VIEW v_src_drug_era1

```
CREATE OR REPLACE VIEW v_src_drug_era1_ids
 (drug_exposure_id, drug_exposure_count) AS
SELECT /*+ NO_PARALLEL(cond) */ DISTINCT
  drug_era_id, drug.drug_exposure_count
FROM s_drug_era drug
INNER JOIN v_src_person person ON drug.person_id = person.person_id
WHERE drug_exposure_type = '7' -- SET PERSISTENCE WINDOW
WITH READ ONLY;
```

VIEW v_src_first_drugs

```
CREATE OR REPLACE VIEW v_src_first_drugs
(person_id, drug_era_start_date, drug_era_end_date, drug_concept_id) AS
SELECT DISTINCT
  drug.person_id,
  FIRST_VALUE(drug_era_start_date)
    OVER
    (PARTITION BY drug.person_id, drug.drug_concept_id
      ORDER BY drug.drug_era_start_date) AS drug_era_start_date,
  FIRST_VALUE(drug_era_end_date)
    OVER
    (PARTITION BY drug.person_id, drug.drug_concept_id
      ORDER BY drug.drug_era_start_date) AS drug_era_end_date,
  drug_concept_id
FROM v_src_drug_era1 drug
INNER JOIN s_person person ON drug.person_id = person.person_id
GROUP BY drug.person_id, drug_era_start_date,  drug_era_end_date, drug_concept_id
WITH READ ONLY;
```

OSIM 2 Source Data Views: Customization

The standard views can be modified to filter the data the simulation analysis will use to populate the probability tables. At the very minimum a single drug era persistence window and condition era persistence window must be selected by the views. The drug chosen drug era persistence wind can be different from the condition era persistence window.

Other filtering can be done in the views as well, limiting the analysis to a particular age range, gender, or person_id range.

Setting the Drug Era Persistence Window

The Drug Era Persistence Window is modified by changing the “WHERE” clause in the following data source views:

- v_src_person_strata
- v_src_drug_era1
- v_src_drug_era1_ids

0 Day Persistence Window

WHERE drug_exposure_type = ’6’

30 Day Persistence Window

WHERE drug_exposure_type = ’7’

Setting the Condition Era Persistence Window

The Condition Era Persistence Window is modified by changing the “WHERE” clause in the following data source views:

- v_src_person_strata
- v_src_condition_era1_ids
- v_src_condition_era1

0 Day Persistence Window

WHERE condition_occurrence_type = ’64’

30 Day Persistence Window

WHERE condition_occurrence_type = ’65’

Adding Criteria for Limiting Person Selection

Any additional criteria can be added to the view v_src_person to limit person selection by age, gender, or even to a set of drug or condition concepts (requiring an additional join drug and/or condition eras in the view).

Limit analysis to a range of person_ids

During OSIM 2 code development, the source database analysis was often limited to the first 150,000 persons of MSLR database to speed up both analysis and simulation tests.

```
--================================================================================
-- VIEW v_src_person
--================================================================================
CREATE OR REPLACE VIEW v_src_person
 (person_id, year_of_birth, gender_concept_id, race_concept_id,
  location_concept_id, source_person_key, source_gender_code,
  source_location_code, source_race_code) AS
SELECT /*+ NO_PARALLEL(person) */DISTINCT
  person.person_id,person.year_of_birth, person.gender_concept_id,
  person.race_concept_id, person.location_concept_id, person.source_person_key,
  person.source_gender_code, person.source_location_code, person.source_race_code
FROM s_person person
INNER JOIN s_observation_period period on person.person_id = period.person_id
WHERE person.year_of_birth IS NOT NULL
  AND period.observation_period_start_date IS NOT NULL
  AND person.person_id <= 1022865402
WITH READ ONLY;
```

User Modifiable Range Functions

The user-modifiable range functions are used by both the database analysis and simulation phases of OSIM 2. They specify the ranges of various strata to be bucketed together for the transition probabilities. The user has control over the categories and related ranges associated with some of the key distributions, via a set of functions in the OSIM2 Oracle package described below. The simulation must run with the same range functions that were used to create the transition tables.

FUNCTION age_bucket

The following CASE statement can be modified to change the default age categories for analysis. The default values are shown in the example below; range values can be changed based on desired age categories.

```
CASE TRUE
  WHEN age IS NULL THEN RETURN NULL;
  WHEN age < 6 THEN RETURN 6;
  WHEN age < 14 THEN RETURN 14;
  WHEN age < 20 THEN RETURN 20;
  WHEN age < 55 THEN RETURN 55;
  WHEN age < 70 THEN RETURN 70;
  ELSE RETURN 120;
END CASE;
```

FUNCTION condition_count_bucket

The following CASE statement can be modified to change the default categories which place the count of conditions into pre-defined bins. The default values are shown in the example below; the range values can be changed based on desired age categories.

```
CASE TRUE
  WHEN condition_count <= 2 THEN RETURN 2;
  WHEN condition_count <= 7 THEN RETURN 7;
  WHEN condition_count <= 25 THEN RETURN 25;
  ELSE RETURN 2000;
END CASE;
```

FUNCTION drug_count_bucket

The following CASE statement can be modified to change the default categories which place the count of drugs into pre-defined bins. The default values are shown in the example below; the range values can be changed based on desired age categories.

```
CASE TRUE
  WHEN drug_count <= 2 THEN RETURN 2;
  WHEN drug_count <= 7 THEN RETURN 7;
  WHEN drug_count <= 25 THEN RETURN 25;
  ELSE RETURN 2000;
END CASE;
```

FUNCTION time_observed_bucket

The following CASE statement can be modified to change the default categories which place the person’s observation period time remaining into pre-defined bins. The default values are shown in the example below, which are simply full semi-annual periods.

```
CASE TRUE
  WHEN days > 0 THEN RETURN FLOOR((1+days) / 182.625);
  ELSE RETURN 0;
END CASE;
```

FUNCTION round_days

The following CASE statement can be modified to change the default categories which place era duration and separating intervals (in days) into pre-defined bins. The default values are shown in the example below.

```
CASE TRUE
  WHEN days <= 75 THEN RETURN ROUND(days);
  ELSE RETURN ROUND(days - 15 + DBMS_RANDOM.VALUE * 30);
END CASE;
```

FUNCTION duration_days_bucket

The following CASE statement can be modified to change the default categories which place drug era duration intervals and time between condition days for drug simulation (in days) into pre-defined bins. The default values are shown in the example below.

```
CASE TRUE
  WHEN days <= 7 THEN RETURN 7;
  ELSE RETURN 8;
END CASE;
```

Procedure 1: Analyze CDM data and Populate Transition Probability Tables

The main purpose of **Procedure 1** is to thoroughly analyze the source CDM database for various attributes and probabilities and populate tables for the simulation to produce data with similar characteristics.

The high-level process for **Procedure 1** simply calls a series of 13 internal procedures, which each analyze and populate one of the attribute or probability tables.

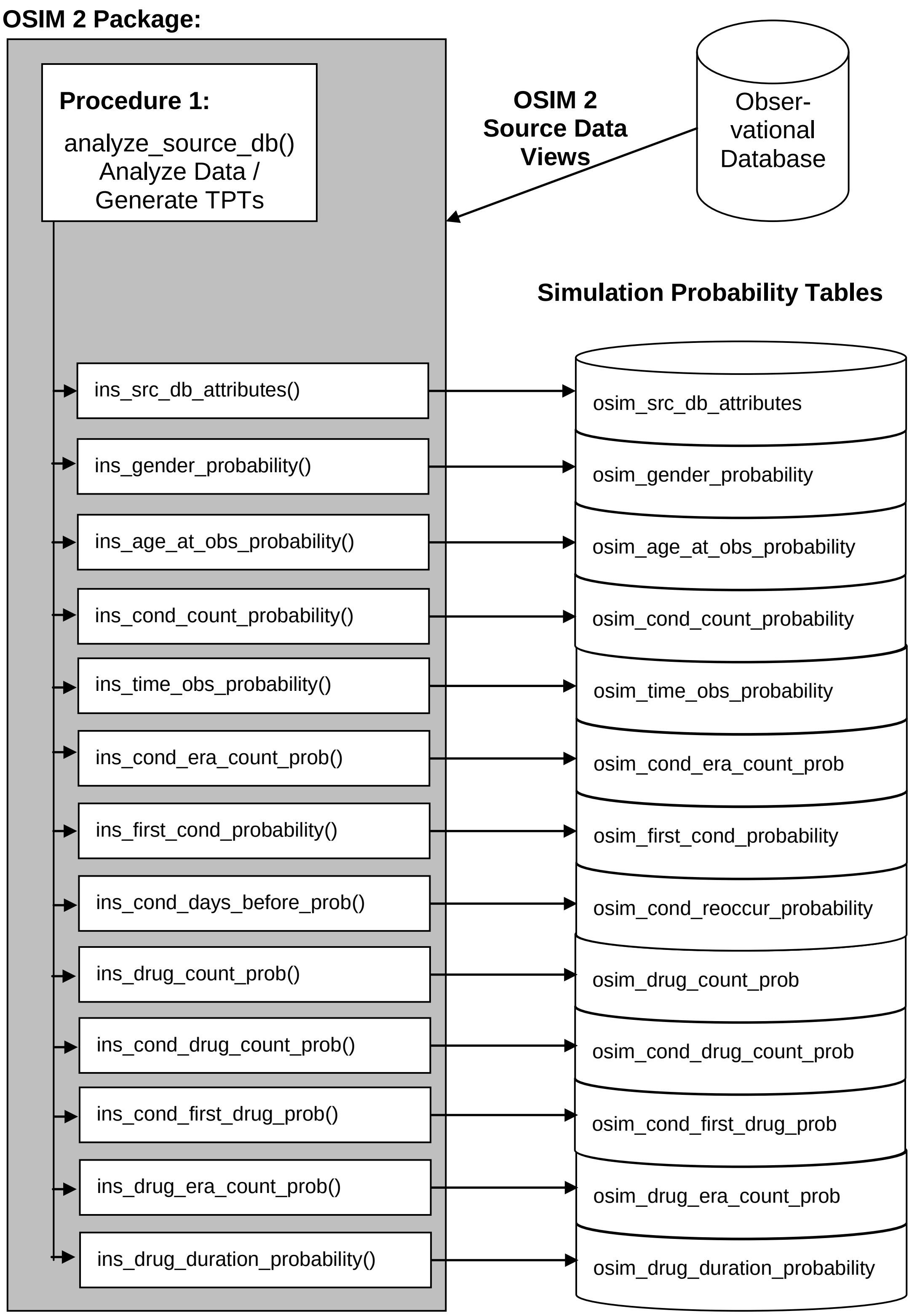
Each internal procedure generally performs four steps:

- Deletes All Current Data From Target Table
- Drops All Indexes From Target Table (For Quicker Insertion)
- Inserts Analysis Results Into Target Table
- Restores All Indexes For Target Table

Each internal procedure will post log messages for starting, number of affected rows, and ending to the Process Log Table.

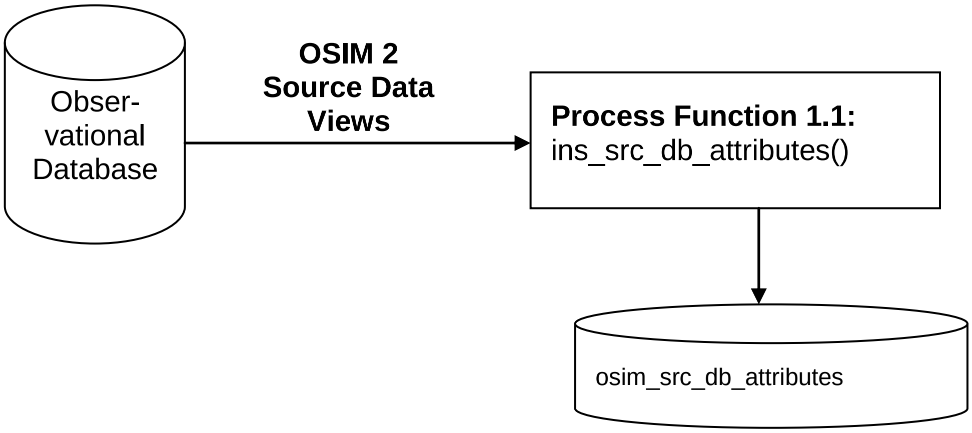
The overall process is illustrated in Figure 3, and the details of each function are described in the following section.

OSIM 2 Procedure 1: Analyze CDM data and Populate Transition Probability Tables High Level Process Design


OSIM 2 Package:
Procedure 1:
analyze_source_db()
Analyze Data /
Generate TPTs
OSIM 2 Source Data Views
Obser-vational Database
Simulation Probability Tables
ins_src_db_attributes()
ins_gender_probability()
ins_age_at_obs_probability()
ins_cond_count_probability()
ins_time_obs_probability()
ins_cond_era_count_prob()
ins_first_cond_probability()
ins_cond_days_before_prob()
ins_drug_count_prob()
ins_cond_drug_count_prob()
ins_cond_first_drug_prob()
ins_drug_era_count_prob()
ins_drug_duration_probability()
osim_src_db_attributes
osim_gender_probability
osim_age_at_obs_probability
osim_cond_count_probability
osim_time_obs_probability
osim_cond_era_count_prob
osim_first_cond_probability
osim_cond_reoccur_probability
osim_drug_count_prob
osim_cond_drug_count_prob
osim_cond_first_drug_prob
osim_drug_era_count_prob
osim_drug_duration_probability

Process Function 1.1: Analyze Source Database Attributes

The Analyze Source Database Attributes function counts and stores general database attributes from the source CDM database.



Field Name	Format	Description / Comment
db_min_date	Date	The minimum era or observation period start date in the CDM database being analyzed for simulation.
db_max_date	Date	The maximum era or observation period end date in the CDM database being analyzed for simulation.
person_count	Number	The number of persons in the CDM database being analyzed for simulation.
condition_era_count	Number	The number of 0 day persistence window condition eras in the CDM database being analyzed for simulation.
drug_era_count	Number	The number of 30 day persistence window drug eras in the CDM database being analyzed for simulation.
condition_occurrence_type	Text	The condition_occurrence_type of the analyzed condition eras.
drug_exposure_type	Text	The drug_exposure_type of the analyzed drug eras.

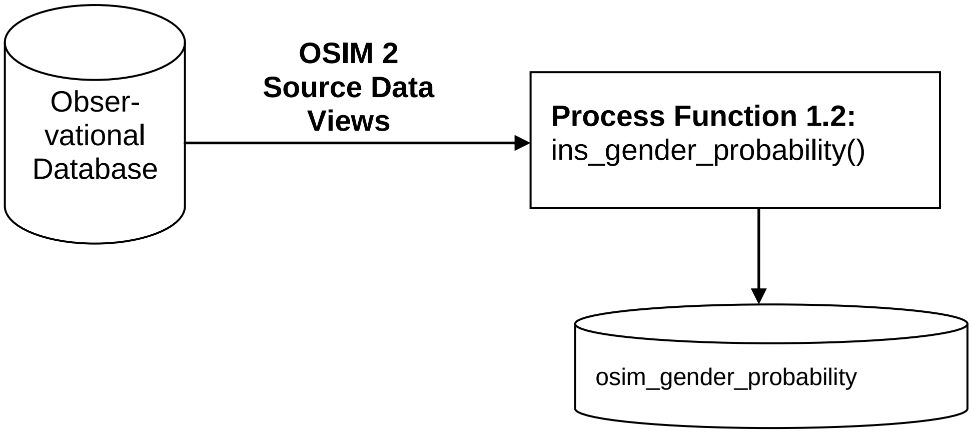
Process Description:

1) **Log** starting message
2) **Analyze** source database for minimum era or observation period start date
3) **Analyze** source database for maximum era or observation period start date
4) **Analyze** source database condition eras for occurrence type (persistence window)
5) **Analyze** source database drug eras for exposure type (persistence window)
6) **Count** distinct persons in source database
7) **Count** distinct condition eras in source database
8) **Count** distinct drug eras in source database
9) **Delete** all data in **osim_src_db_attributes** table
10) **Insert** counts into **osim_src_db_attributes** table
11) **Log** number of inserted rows
12) **Log** process complete message

Process Function 1.2: Analyze Gender Probability

Fn(gender_concept_id)

The Analyze Gender Probability function counts and calculates the probability of gender concept codes in the source database.



Field Name	Format	Description / Comment
gender_concept_id	Number	(Result) Distinct gender concept ID from the analyzed source CDM database.
n	Number	Number of persons in the source CDM database with the specified gender concept ID.
accumulated_probability	Float	Accumulating probability of gender concept ID.

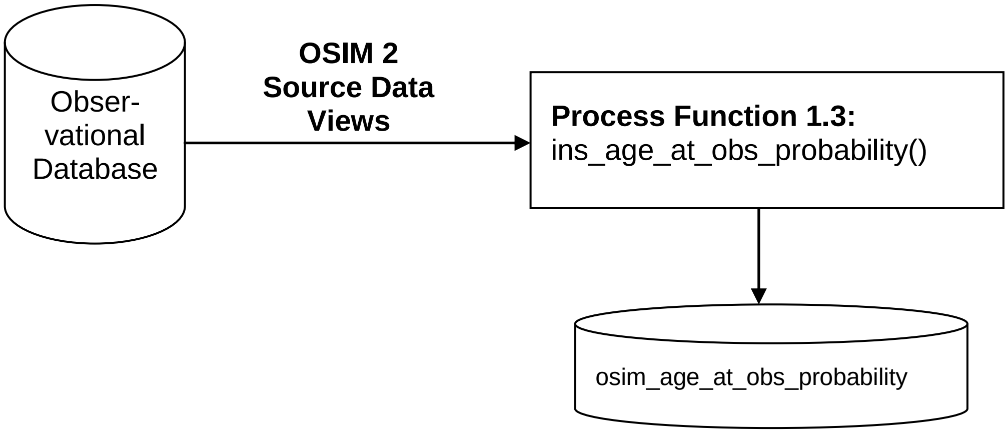
Process Description:

1) **Log** starting message
2) **Delete** all data in **osim_gender_probability** table
3) **Analyze** and **Insert** counts into **osim_gender_probability** table
4) **Log** number of inserted rows
5) **Update** all maximum accumulated_probability values to 1.0
6) **Log** process complete message

Process Function 1.3: Analyze Age Probability

Fn(age_at_obs | gender_concept_id)

The Analyze Age Probability function counts and calculates the probability of age at observation period start for each gender concept code in the source database.



Field Name	Format	Description / Comment
gender_concept_id	Number	(Stratum) Distinct gender concept ID from the analyzed source CDM database.
age_at_obs	Number	(Result) Age in whole years of person at beginning of observation period.
n	Number	Number of persons in the source CDM database with the specified gender concept ID and age at observation.
accumulated_probability	Float	Accumulating probability of age at observation for each gender concept ID.

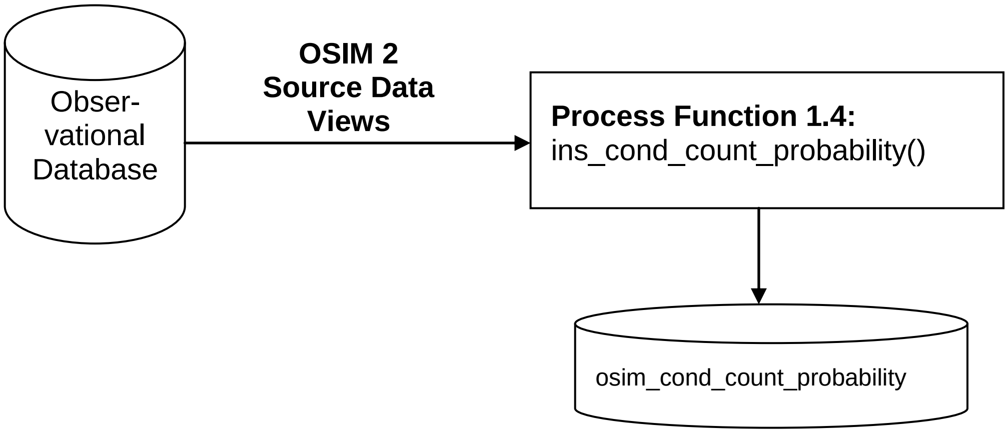
Process Description:

1) **Log** starting message
2) **Delete** all data in **osim_age_at_obs_probability** table
3) **Analyze** and **Insert** counts into **osim_age_at_obs_probability** table
4) **Log** number of inserted rows
5) **Update** all maximum accumulated_probability values to 1.0
6) **Log** process complete message

Process Function 1.4: Analyze Distinct Condition Count Probability

Fn(cond_concept_count | gender_concept_id, age_at_obs)

The Analyze Distinct Condition Count Probability function counts and calculates the probability for the number of distinct condition concepts for each age and gender concept code in the source database.



Field Name	Format	Description / Comment
gender_concept_id	Number	(Stratum) Distinct gender concept ID from the analyzed source CDM database.
age_at_obs	Number	(Stratum) Age in whole years of person at beginning of observation period.
cond_concept_count	Number	(Result) Count of distinct condition concepts.
n	Number	Number of persons in the source CDM database with the specified gender concept ID age at observation, and number of distinct conditions.
accumulated_probability	Float	Accumulating probability for the strata.

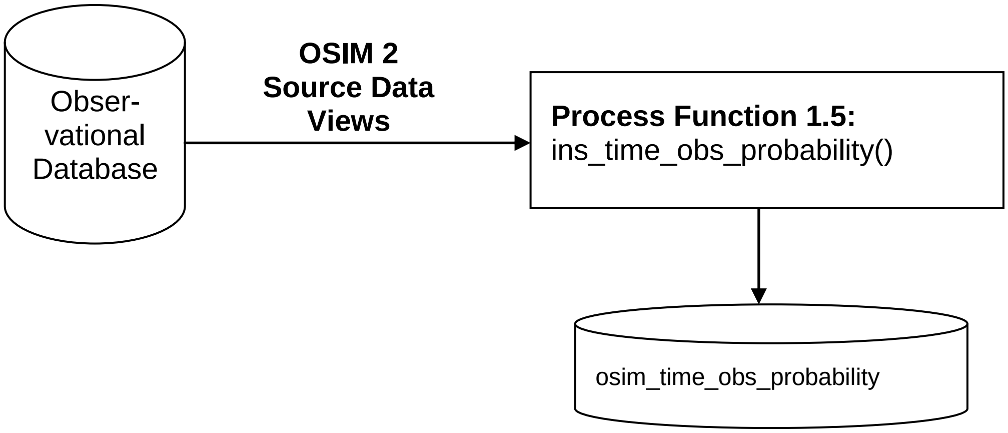
Process Description:

1) **Log** starting message
2) **Delete** all data in **osim_cond_count_probability** table
3) **Analyze** and **Insert** counts into **osim_cond_count_probability** table
4) **Log** number of inserted rows
5) **Update** all maximum accumulated_probability values to 1.0
6) **Log** process complete message

Process Function 1.5: Analyze Time Observed Probability

Fn(time_observed | gender_concept_id, age_at_obs, cond_count_bucket)

The Analyze Time Observed Probability function counts and calculates the probability of complete semi-annual (six month) periods between the minimum observation period start and maximum observation period end date for a person based on age range, gender, and number of distinct conditions range.



Field Name	Format	Description / Comment
gender_concept_id	Number	(Stratum) Distinct gender concept ID from the analyzed source CDM database.
age_at_obs	Number	(Stratum) Age in whole years of person at beginning of observation period.
cond_count_bucket	Number	(Stratum) Bucketed count of distinct condition concepts.
time_observed	Number	(Result) Person observation duration in full semi-years. A value of zero represents less than 6 months observed.
n	Number	Number of persons in the source CDM database with the specified gender concept ID age at observation, and number of distinct conditions.
accumulated_probability	Float	Accumulating probability for the strata.

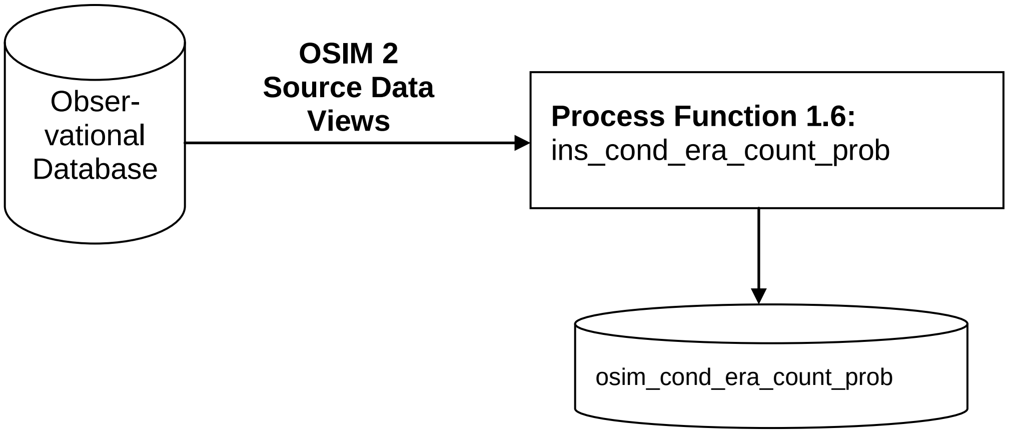
Process Description:

1) **Log** starting message
2) **Delete** all data in **osim_time_obs_probability** table
3) **Analyze** and **Insert** counts into **osim_time_obs_probability** table
4) **Log** number of inserted rows
5) **Update** all maximum accumulated_probability values to 1.0
6) **Log** process complete message

Process Function 1.6: Analyze Condition Era Count Probability

Fn(cond_era_count | condition_concept_id, cond_count_bucket, time_remaining)

The Analyze Condition Era Count Probability function counts and calculates the probability of the number of total condition eras for a particular condition concept, based on number of distinct conditions range, and complete semi-annual (six month) periods remaining until the maximum observation period end date.



Field Name	Format	Description / Comment
condition_concept_id	Number	(Stratum) Distinct condition concept.
condition_count_bucket	Number	(Stratum) Bucketed count of distinct condition concepts.
time_remaining	Number	(Stratum) Person observation duration in full semi-years. A value of zero represents less than 6 months observed.
cond_era_count	Number	(Result) Total number of condition eras for the condition concept.
n	Number	Number of occurrences in the source CDM database with the strata and results.
accumulated_probability	Float	Accumulating probability for the strata.

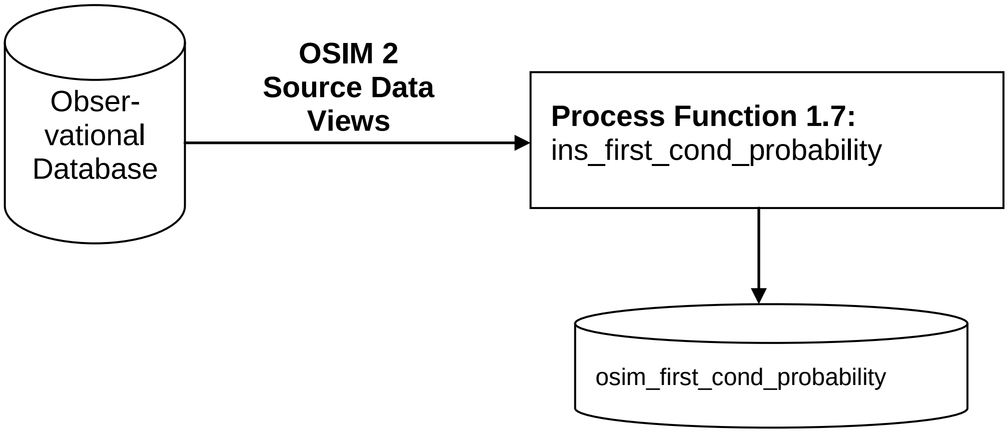
Process Description:

1) **Log** starting message
2) **Drop** all indexes for the **osim_cond_era_count_prob** table
3) **Delete** all data in **osim_cond_era_count_prob** table
4) **Analyze** and **Insert** counts into **osim_cond_era_count_prob** table
5) **Log** number of inserted rows
6) **Create** all indexes for the **osim_cond_era_count_prob** table
7) **Update** all maximum accumulated_probability values to 1.0
8) **Log** process complete message

Process Function 1.7: Analyze First Occurrence Condition Transition Probability

Fn(condition2_concept_id, delta_days | gender_concept_id, age_range, cond_count_bucket, time_remaining, condition1_concept_id)

The Analyze First Occurrence Condition Transition Probability function counts and calculates the probability of subsequent first occurrence condition concepts and the days between based on the prior condition concept, the number of distinct condition concepts range, time remaining in full semi-years until observation end, gender, and age range. It only analyzes the very first condition era for each condition concept for each person.



Field Name	Format	Description / Comment
gender_concept_id	Number	(Stratum) Distinct gender concept ID from the analyzed source CDM database.
age_range	Number	(Stratum) Age in whole years of person at beginning of observation period for initial condition (condition1_concept_id = -1); bucketed age at beginning of condition era for actual condition concepts.
cond_count_bucket	Number	(Stratum) Bucketed count of distinct condition concepts.
time_remaining	Number	(Stratum) Person observation duration in full semi-years. A value of zero represents less than 6 months observed.
condition1_concept_id	Number	(Stratum) Prior condition concept, -1 for observation period start.
condition2_concept_id	Number	(Result) Next condition concept.
delta_days	Number	(Result) Days from prior to next condition (see the round_days bucket function).

Field Name	Format	Description / Comment
n	Number	Number of occurrences in the source CDM database with the strata and results.
accumulated_probability	Float	Accumulating probability for the strata.

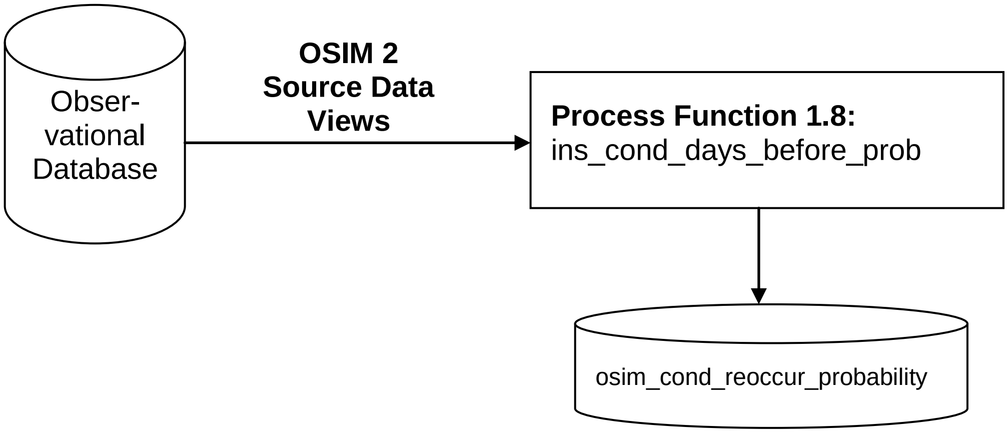
Process Description:

1) **Log** starting message
2) **Drop** all indexes for the **osim_first_cond_probability** table
3) **Delete** all data in **osim_first_cond_probability** table
4) **Analyze** and **Insert** counts into **osim_first_cond_probability** table
5) **Log** number of inserted rows
6) **Create** all indexes for the **osim_first_cond_probability** table
7) **Update** all maximum accumulated_probability values to 1.0
8) **Log** process complete message

Process Function 1.8: Analyze Days between Subsequent Conditions Probability

Fn(delta_days | condition_concept_id, age_range, time_remaining)

The Analyze Days between Subsequent Conditions Probability function counts and calculates the probability of the days between subsequent reoccurrences of the same condition based on the condition concept, the age range, and semi-years remaining in the observation period at the previous occurrence.



Field Name	Format	Description / Comment
condition_concept_id	Number	(Stratum) Distinct condition concept.
age_range	Number	(Stratum) Bucketed age at beginning of condition era.
time_remaining	Number	(Stratum) Person observation duration remaining in full semi-years at beginning of condition era. A value of zero represents less than 6 months observed.
delta_days	Number	(Result) Days from prior to next condition (see the round_days bucket function).
n	Number	Number of occurrences in the source CDM database with the strata and results.
accumulated_probability	Float	Accumulating probability for the strata.

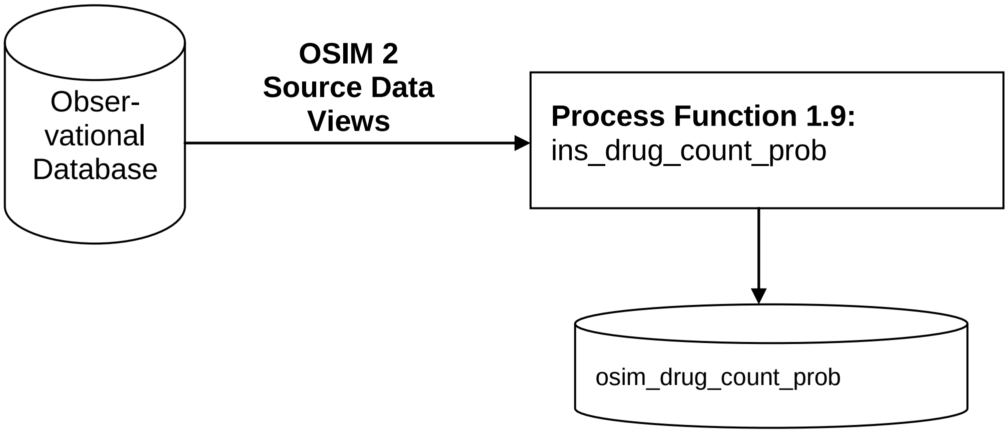
Process Description:

1) **Log** starting message
2) **Drop** all indexes for the **osim_cond_reoccur_probability** table
3) **Delete** all data in **osim_cond_reoccur_probability** table
4) **Analyze** and **Insert** counts into **osim_cond_reoccur_probability** table
5) **Log** number of inserted rows
6) **Create** all indexes for the **osim_cond_reoccur_probability** table
7) **Update** all maximum accumulated_probability values to 1.0
8) **Log** process complete message

Process Function 1.9: Analyze Distinct Drug Concept Count Probability

Fn(drug_count | gender_concept_id, age_bucket, condition_count_bucket)

The Analyze Distinct Drug Concept Count Probability function counts and calculates the probability of the distinct number of drug concepts based on age range, gender, and distinct condition count range.



Field Name	Format	Description / Comment
gender_concept_id	Number	(Stratum) Distinct gender concept ID from the analyzed source CDM database.
age_bucket	Number	(Stratum) Bucketed age at beginning of observation period.
condition_count_bucket	Number	(Stratum) Bucketed count of distinct condition concepts.
drug_count	Number	(Result) Number of distinct drug concepts.
n	Number	Number of occurrences in the source CDM database with the strata and results.
accumulated_probability	Float	Accumulating probability for the strata.

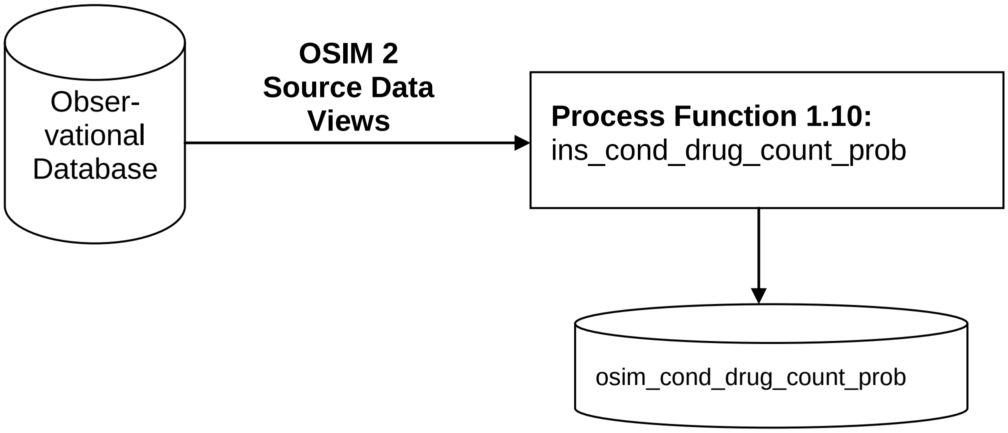
Process Description:

1) **Log** starting message
2) **Drop** all indexes for the **osim_drug_count_prob** table
3) **Delete** all data in **osim_drug_count_prob** table
4) **Analyze** and **Insert** counts into **osim_drug_count_prob** table
5) **Log** number of inserted rows
6) **Create** all indexes for the **osim_drug_count_prob** table
7) **Update** all maximum accumulated_probability values to 1.0
8) **Log** process complete message

Process Function 1.10: Analyze Drug Draw per Condition Count Probability

Fn(drug_count | condition_concept_id, interval_bucket, age_range, drug_count_bucket, cond_count_bucket)

The Analyze Drug Draw per Condition Count Probability function counts and calculates the probability of the number of drug draws the simulation should make for a given condition. When multiple condition eras occur on the same day, the probability of each is divided amongst the succeeding first occurrence drug eras in the gap to the next day with conditions.



Field Name	Format	Description / Comment
gender_concept_id	Number	(Stratum) Distinct gender concept ID from the analyzed source CDM database.
age_bucket	Number	(Stratum) Bucketed age at beginning of observation period.
condition_count_bucket	Number	(Stratum) Bucketed count of distinct condition concepts.
drug_count	Number	(Result) Number of distinct drug concepts.
n	Number	Number of occurrences in the source CDM database with the strata and results.
accumulated_probability	Float	Accumulating probability for the strata.

Process Description:

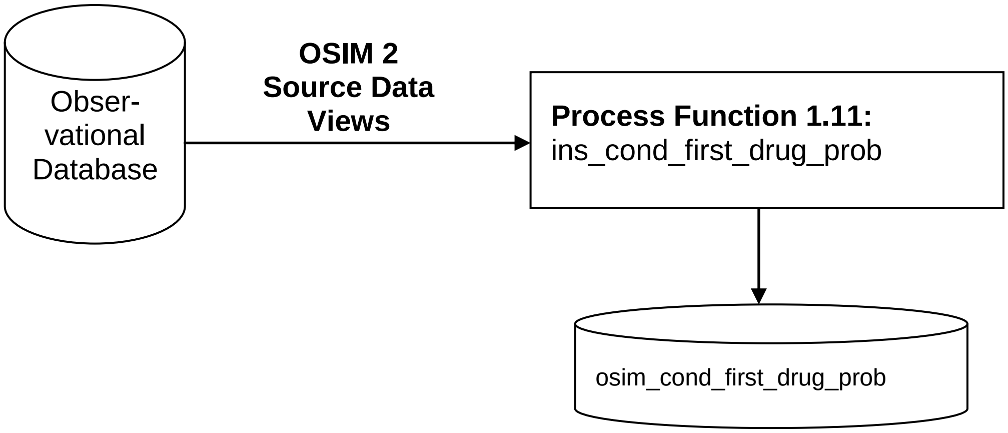
1) **Log** starting message
2) **Drop** all indexes for the **osim_cond_drug_count_prob** table
3) **Delete** all data in **osim_cond_drug_count_prob** table
4) **Analyze** and **Insert** counts into **osim_cond_drug_count_prob** table
5) **Log** number of inserted rows
6) **Create** all indexes for the **osim_cond_drug_count_prob** table
7) **Update** all maximum accumulated_probability values to 1.0
8) **Log** process complete message

Process Function 1.11: Analyze First Occurrence Drug Concept Transition From Condition Era

Fn(drug_count | condition_concept_id, interval_bucket, age_range, drug_count_bucket, cond_count_bucket)

The Analyze First Occurrence Drug Concept Transition From Condition Era function counts and calculates the probability of the transition from a condition era to the drug concept of a first occurrence drug era. The duration in days until the drug era occurrence is also returned. The transition is further stratified by the duration of the gap to the next day with condition(s) (or end of observation), gender, age range, distinct number of conditions range, and the number of conditions on the day preceding the gap.

For the initial gap, from the observation period start date, a condition concept of -1 is used; it may or may not coincide with other condition concepts on the same date. If no first occurrence drug eras occur in the gap, the drug concept of -1 is used to designate the probability of transitioning to no first occurrence drug era.


Obser-vational Database
OSIM 2 Source Data Views
Process Function 1.11: ins_cond_first_drug_prob
osim_cond_first_drug_prob

Field Name	Format	Description / Comment
condition_concept_id	Number	(Stratum) Distinct condition concept.
interval_bucket	Number	(Stratum) Bucketed number of days between subsequent days with conditions (see duration_days_bucket function).
gender_concept_id	Number	(Stratum) Distinct gender concept ID from the analyzed source CDM database.
age_bucket	Number	(Stratum) Bucketed age at beginning of condition era.
condition_count_bucket	Number	(Stratum) Bucketed count of distinct condition concepts.
drug_count_bucket	Number	(Stratum) Bucketed count of distinct drug concepts.
day_cond_count	Number	(Stratum) Number of distinct conditions on

Field Name	Format	Description / Comment
		the day of the condition preceding the gap.
drug_concept_id	Number	(Result) Drug concept occurring in the gap. A value of -1 is used to designate the probability of no drug.
delta_days	Number	(Result) Days from condition date until the drug era (see the round_days bucket function).
n	Number	Number of occurrences in the source CDM database with the strata and results.
accumulated_probability	Float	Accumulating probability for the strata.

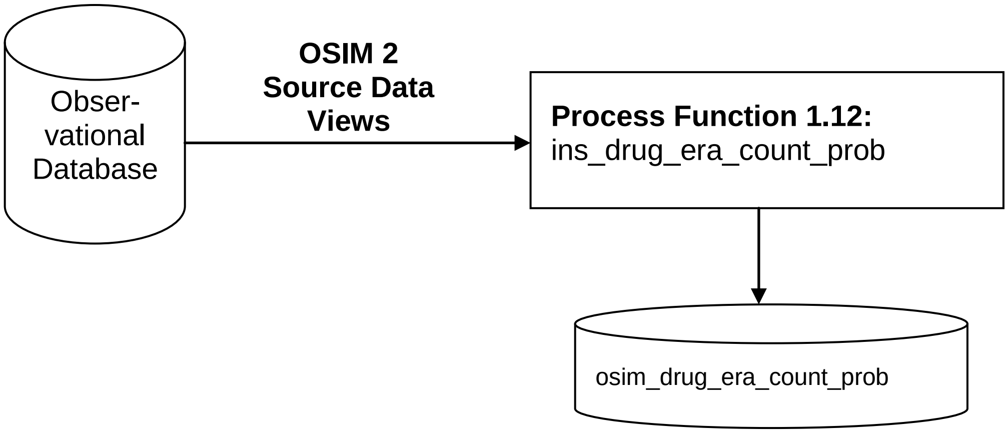
Process Description:

1) **Log** starting message
2) **Drop** all indexes for the **osim_cond_first_drug_prob** table
3) **Delete** all data in **osim_cond_first_drug_prob** table
4) **Analyze** and **Insert** counts into **osim_cond_first_drug_prob** table for days with multiple condition eras
5) **Log** number of inserted rows
6) **Analyze** and **Insert** counts into **osim_cond_first_drug_prob** table for days with a single condition era
7) **Log** number of inserted rows
8) **Create** all indexes for the **osim_cond_first_drug_prob** table
9) **Update** all maximum accumulated_probability values to 1.0
10) **Log** process complete message

Process Function 1.12: Analyze Drug Reoccurrence Count

Fn(drug_era_count, total_exposure | drug_concept_id, drug_count_bucket, condition_count_bucket, age_range, time_remaining)

The Analyze Drug Reoccurrence Count function counts and calculates the probability of the number of drug eras and the total drug exposure length (combined duration of all the drug eras for the drug) based on the drug concept, the ranged number of distinct drugs, the ranged number of distinct conditions, age range, and full semi-years remaining in the observation period.



Field Name	Format	Description / Comment
drug_concept_id	Number	(Stratum) Distinct drug concept.
drug_count_bucket	Number	(Stratum) Bucketed count of distinct drug concepts.
condition_count_bucket	Number	(Stratum) Bucketed count of distinct condition concepts.
age_range	Number	(Stratum) Bucketed age at beginning of condition era.
time_remaining	Number	(Stratum) Person observation duration remaining in full semi-years at beginning of first drug era. A value of zero represents less than 6 months observed.
drug_era_count	Number	(Result) Count of total drug eras for the drug concept.
total_exposure	Number	(Result) Bucketed total number of exposure days for drug concept (see round_days function).
n	Number	Number of occurrences in the source CDM database with the strata and results.
accumulated_probability	Float	Accumulating probability for the strata.

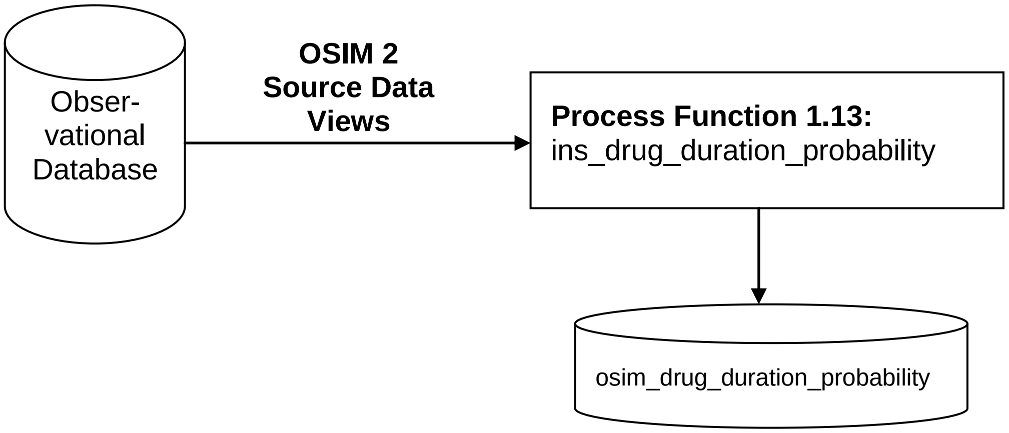
Process Description:

1) **Log** starting message
2) **Drop** all indexes for the **osim_drug_era_count_prob** table
3) **Delete** all data in **osim_drug_era_count_prob** table
4) **Analyze** and **Insert** counts into **osim_drug_era_count_prob** table
5) **Log** number of inserted rows
6) **Create** all indexes for the **osim_drug_era_count_prob** table
7) **Update** all maximum accumulated_probability values to 1.0
8) **Log** process complete message

Process Function 1.13: Analyze Total Drug Duration Probability

Fn(total_duration | drug_concept_id, time_remaining, drug_era_count, total_exposure, total_duration)

The Analyze Total Drug Duration Probability function counts and calculates the probability of the number of total duration of a drug (the start of the first drug era to the end of the last drug era) for a given drug further stratified by semi-years remaining in observation period, number of drug eras for the drug, and the total exposure to the drug (sum of the drug era lengths).



Field Name	Format	Description / Comment
drug_concept_id	Number	(Stratum) Distinct drug concept.
time_remaining	Number	(Stratum) Person observation duration remaining in full semi-years at beginning of first drug era. A value of zero represents less than 6 months observed.
drug_era_count	Number	(Stratum) Count of total drug eras for the drug concept.
total_exposure	Number	(Stratum) Bucketed total number of exposure days for drug concept (see round_days function).
total_duration	Number	(Result) Bucketed total duration for exposure from start of initial exposure to end of final exposure for drug concept (see round_days function).
n	Number	Number of occurrences in the source CDM database with the strata and results.
accumulated_probability	Float	Accumulating probability for the strata.

Process Description:

1) **Log** starting message
2) **Drop** all indexes for the **osim_drug_duration_probability** table
3) **Delete** all data in **osim_drug_duration_probability** table
4) **Analyze** and **Insert** counts into **osim_drug_duration_probability** table
5) **Log** number of inserted rows
6) **Create** all indexes for the **osim_drug_duration_probability** table
7) **Update** all maximum accumulated_probability values to 1.0
8) **Log** process complete message

Procedure 2: Generate Simulated Persons

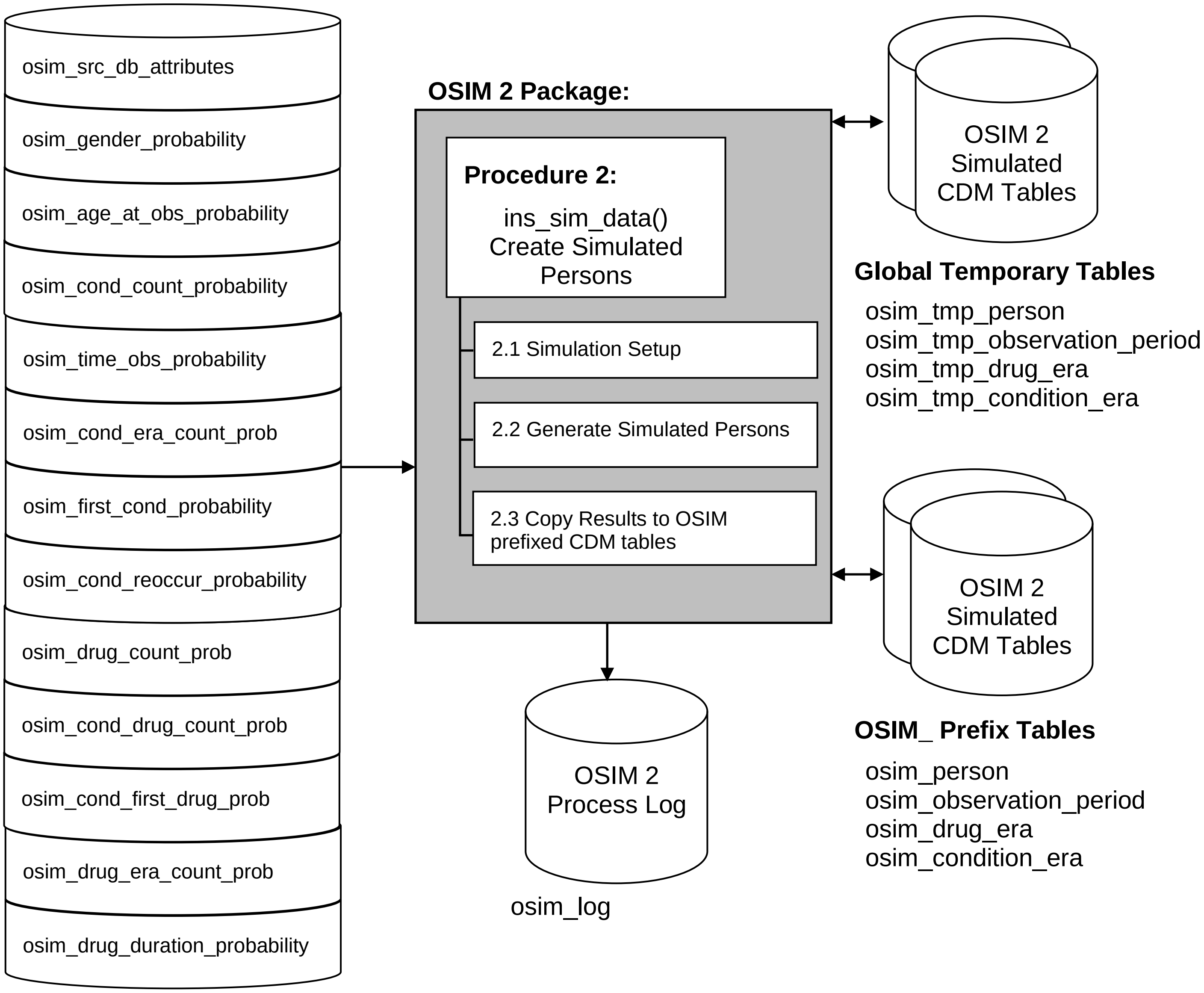
The main purpose of **Procedure 2** is to generate the Simulated Persons, including Observation Period, Drug Eras, and Condition Eras. The simulated persons are created from the Transition Probability Tables populated by **Procedure 1**.

The high-level process for **Procedure 2** comprises 6 functions:

- Simulate Setup Functions
- Generate Simulated Persons and Condition Eras
- Generate First Occurrence Drug Eras
- Generate Subsequent Drug Eras
- Copy Results to OSIM prefixed CDM tables

The overall process is illustrated below and the details of each function are described in the following sections.

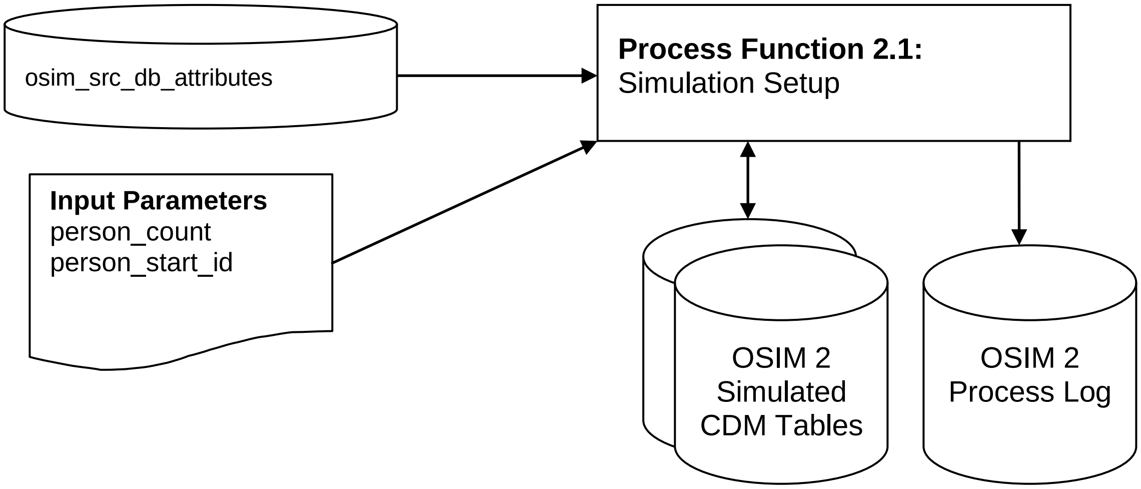
OSIM 2 Procedure 2: Generate Simulated Persons High Level Process Design



Process Function 2.1: Simulation Setup

In order to start the simulation, **Procedure 2** performs a few standard initialization tasks:

- Log Starting Message
- Drop OSIM 2 Simulated CDM Tables Indexes
- Retrieve Source Database Attributes
- Process Input Parameters
- Retrieve maximum IDs

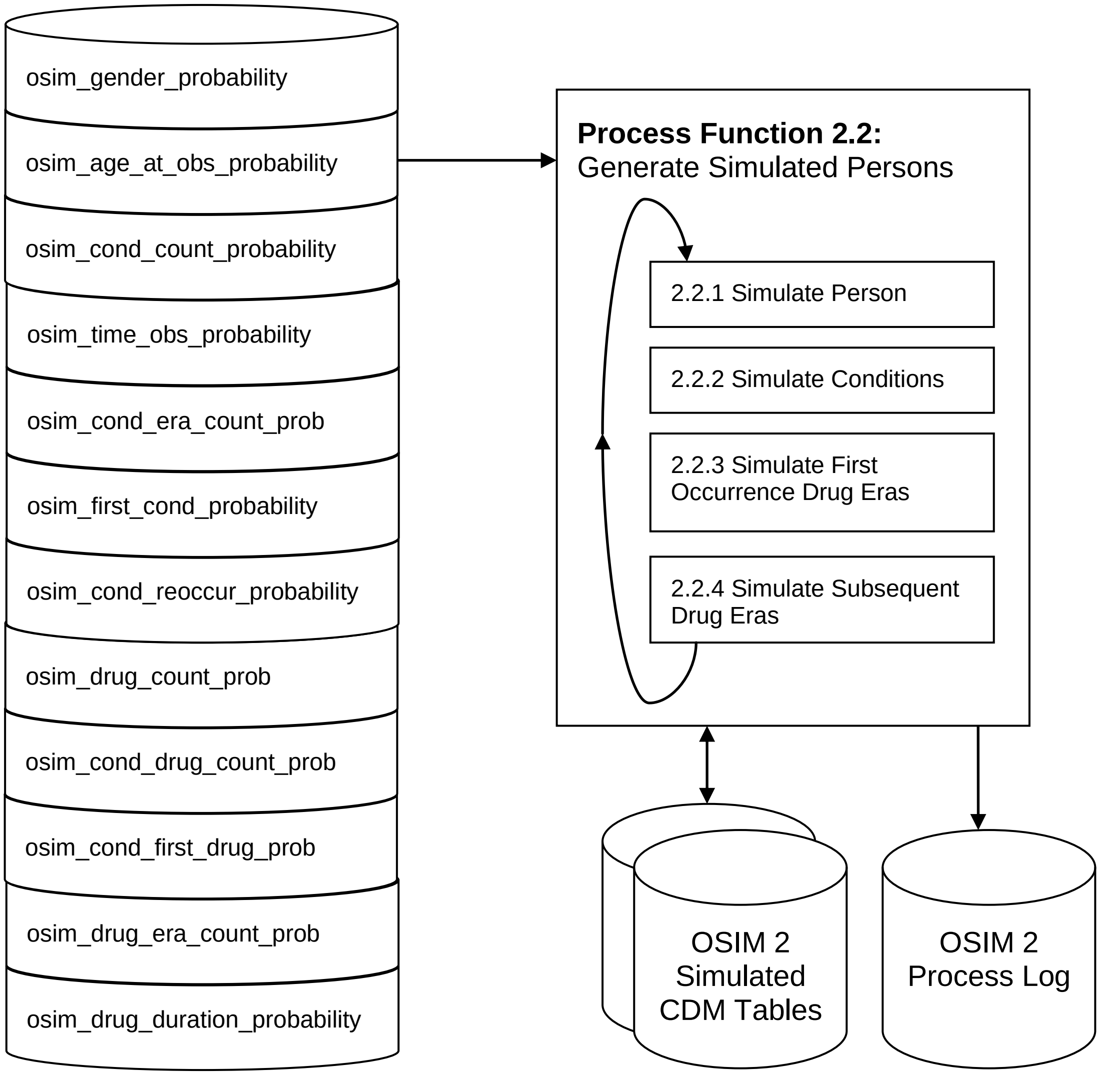


Process Description:

1) **Log** starting message
2) **Drop** all indexes for the **osim_person, osim_observation_period, osim_drug_era,** and **osim_condition_era** tables
3) **Retrieve** general source database attributes from **osim_src_db_attributes** table
4) **If person_start_id Input Parameter = 0**
   Then Set person_id to the maximum **person_id** value from **osim_person** table
5) **If person_start_id** Input Parameter **<> 0**
   Then Set person_id to the **person_start_id** Input Parameter - 1
6) **Set drug_era_id** to the maximum **drug_era_id** value from **osim_drug_era** table
7) **Set condition_era_id** to the maximum **condition_era_id** value from **osim_condition_era** table

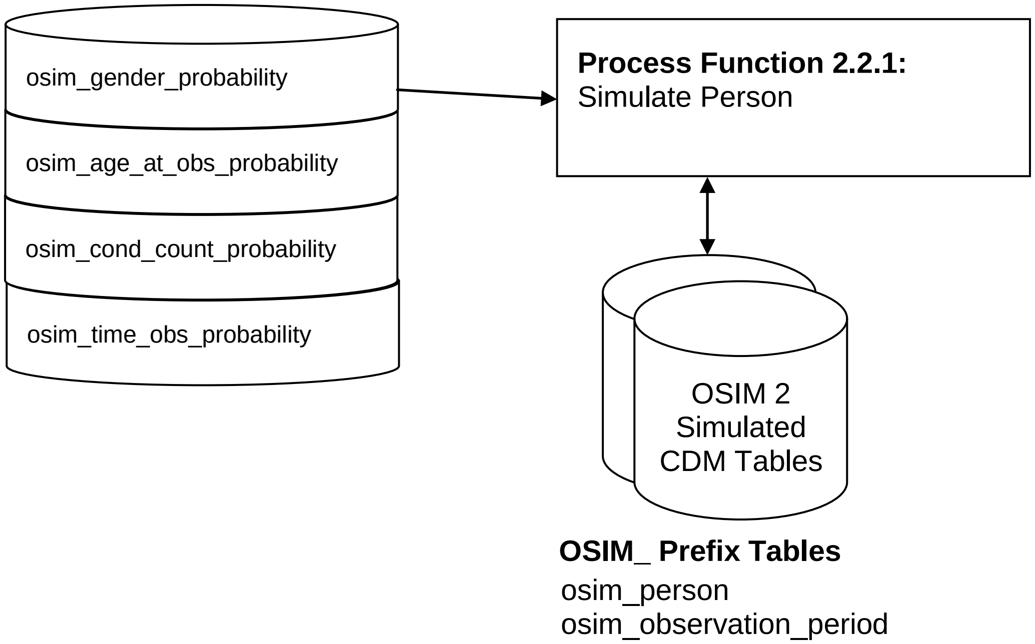
Process Function 2.2: Generate Simulated Persons

The simulation begins a loop to simulate a single person at a time, the observation period, and all condition and drug eras.


osim_gender_probability
osim_age_at_obs_probability
osim_cond_count_probability
osim_time_obs_probability
osim_cond_era_count_prob
osim_first_cond_probability
osim_cond_reoccur_probability
osim_drug_count_prob
osim_cond_drug_count_prob
osim_cond_first_drug_prob
osim_drug_era_count_prob
osim_drug_duration_probability
Process Function 2.2:
Generate Simulated Persons
2.2.1 Simulate Person
2.2.2 Simulate Conditions
2.2.3 Simulate First Occurrence Drug Eras
2.2.4 Simulate Subsequent Drug Eras
OSIM 2 Simulated CDM Tables
OSIM 2 Process Log

Process Function 2.2.1: Simulate Person

Simulate a person from the Transition Probability Tables.

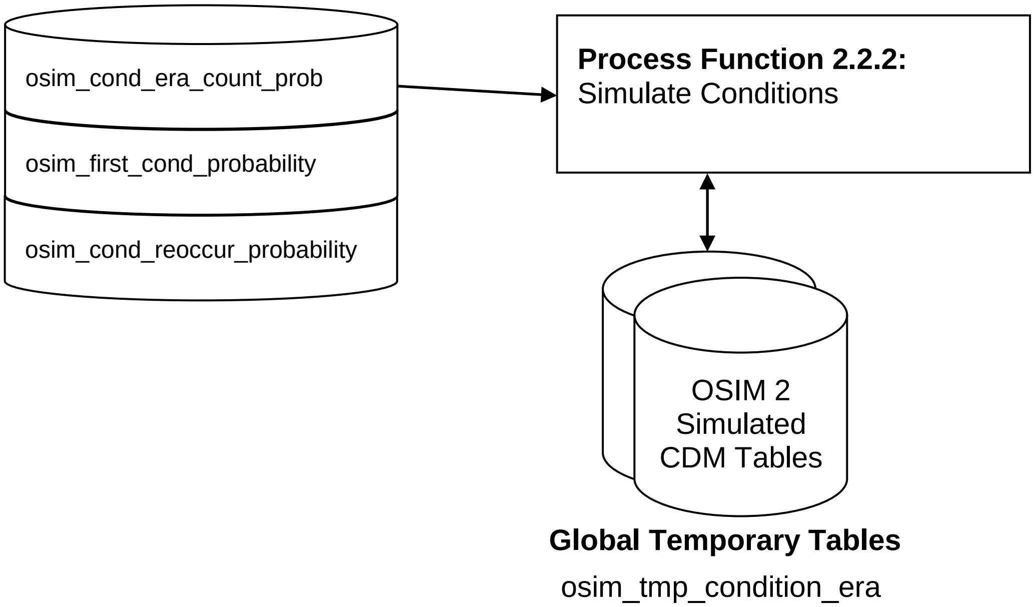

osim_gender_probability
osim_age_at_obs_probability
osim_cond_count_probability
osim_time_obs_probability
Process Function 2.2.1: Simulate Person
OSIM 2 Simulated CDM Tables
OSIM_ Prefix Tables
osim_person
osim_observation_period

Process Description:

1) **Random Draw** for **gender_concept_id** from **osim_gender_probability** table
2) **Random Draw** for **age** from **osim_age_at_obs_probability** table
3) **Random Draw** for **Distinct Condition Count** from **osim_cond_count_probability** table
4) **Random Draw** for **Time Observed** from **osim_time_obs_probability** table
5) **Increment person_id**
6) **Insert Person** into **osim_tmp_person** table
7) **Insert Observation Period** into **osim_tmp_observation_period** table
8) **Purge Condition Concept** array
9) **Purge Drug Concept** array
10) **Set** initial **Condition Concept** to -1

Process Function 2.2.2: Simulate Conditions

Simulate all condition eras for a person from the Transition Probability Tables.


osim_cond_era_count_prob
osim_first_cond_probability
osim_cond_reoccur_probability
Process Function 2.2.2:
Simulate Conditions
OSIM 2
Simulated
CDM Tables
Global Temporary Tables
osim_tmp_condition_era

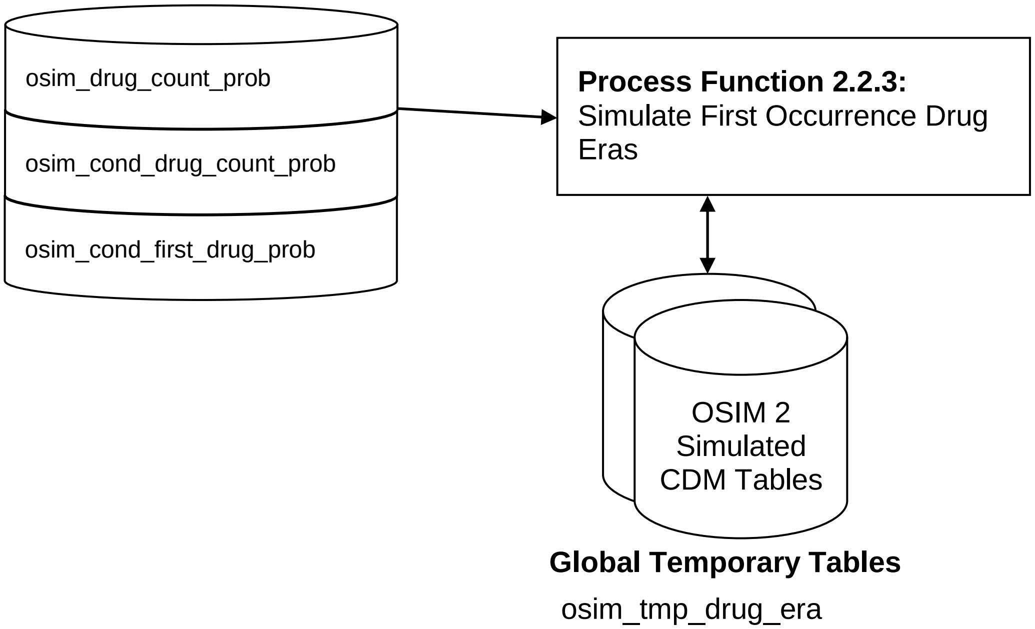
Process Description:

LOOP UNTIL Distinct Condition Count is reached

1) **Random Draw** for next **Conditon Concept** and **Days Until** from **osim_first_cond_probability** table **Until Condition Concept** Not In **Condition Concept** array
2) **Add Condition Concept** to **Condition Concept** array
3) **Random Draw** for **Condition Era Count** from **osim_cond_era_count_prob** table
4) **Insert Condition Era** into **osim_condition_era** table
5) **While Condition Era Count < Drawn Era Count**
    a) **Random Draw** for **Days Until Next Era** from **osim_cond_reoccur_probability** table
    b) **Insert Condition Era** into **osim_condition_era** table

Process Function 2.2.3: Simulate First Occurrence Drug Eras

Simulate all first occurrence drug eras for a person from the Transition Probability Tables.

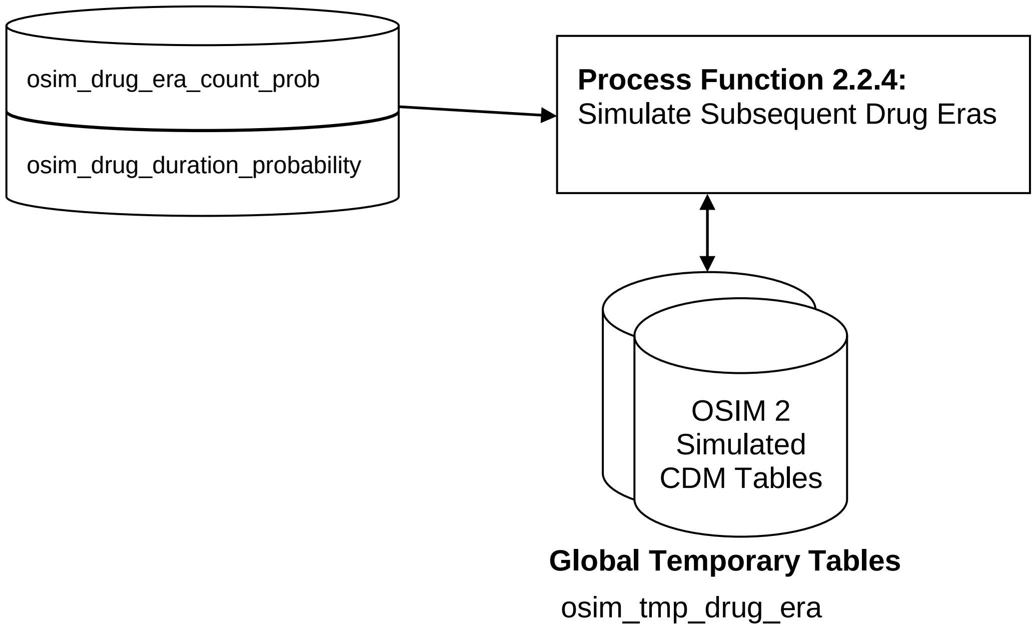

osim_drug_count_prob
osim_cond_drug_count_prob
osim_cond_first_drug_prob
Process Function 2.2.3: Simulate First Occurrence Drug Eras
OSIM 2 Simulated CDM Tables
Global Temporary Tables
osim_tmp_drug_era

Process Description:

```
1)  Random Draw  for Distinct Drug Count from osim_drug_count_prob table
2)  LOOP UNTIL Distinct Drug Count is reached
      a) For Each Condition Era
          i)  Random Draw for next Number of Drug Draws from
              osim_cond_drug_count_prob
          ii) Loop Number of Drug Draws
              (1) Random Draw Drug Concept and Days Until from
                  osim_cond_first_drug_prob
              (2) If Drug Concept Not In Drug Concept  array
                  (a) Insert Drug Era into osim_drug_era table
                  (b) Add Drug Concept  to Drug Concept  array
```

Process Function 2.2.4: Simulate Subsequent Drug Eras

Simulate all subsequent drug eras for a person from the Transition Probability Tables.

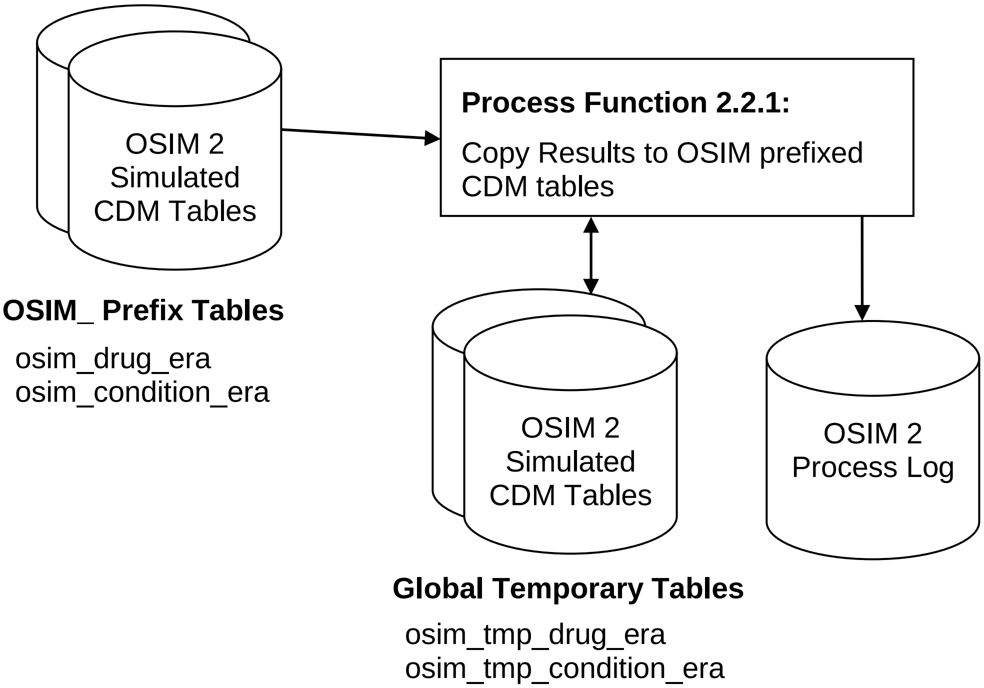

osim_drug_era_count_prob
osim_drug_duration_probability
Process Function 2.2.4:
Simulate Subsequent Drug Eras
OSIM 2
Simulated
CDM Tables
Global Temporary Tables
osim_tmp_drug_era

Process Description:

1) **For Each Drug Era**
    - a) **Random Draw** for **Drug Era Count** and **Total Exposure** from **osim_drug_era_count_prob** table
    - b) **Random Draw** for **Drug Duration** from **osim_drug_duration_probability** table
        - i) **While Actual Drug Era Count** < **Drawn Drug Era Count**
            - (1) **Insert Drug Era** into **osim_drug_era** table with random **Gap**

Process Function 2.3: Copy Results to OSIM prefixed CDM tables

Simulate a person from the Transition Probability Tables.



Process Description:

1) **Copy rows** from **osim_tmp_condition_era** to **osim_condition_era**
2) **Copy rows** from **osim_tmp_condition_era** to **osim_condition_era**
3) **Commit**
4) **Log** complete message

OSIM 2 Optional Procedure 3a: Outcomes

Drug-related Outcomes

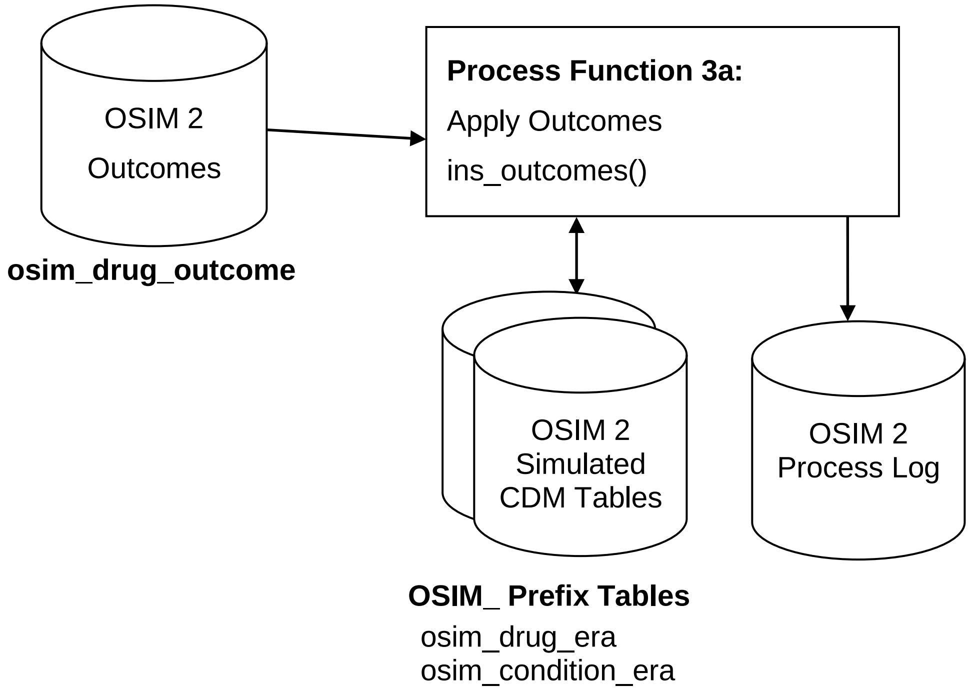
Increased risk in harmful outcomes following drug exposure could be indicative of potential drug adverse reactions that warrant further consideration. The simulation procedure will incorporate the relationship between drugs and outcomes by introducing additional cases of the condition into the sample based on the attributable risk of the condition due to the drug.

The osim_drug_outcome table can be manually populated with known condition / drug outcome effects. Running the optional outcome procedure will apply the rules, and either remove or insert condition eras to obtain the specified relative risk.

TABLE OSIM_DRUG_OUTCOME

Field Name	**Format**	**Description / Comment**
RISK_OR_BENEFIT	Text	'risk' or 'benefit' outcome type
DRUG_CONCEPT_ID	Number	Outcome Drug concept ID
CONDITION_CONCEPT_ID	Number	Outcome condition concept ID
RELATIVE_RISK	Float	Percentage of simulated persons with the drug expected to have outcome
OUTCOME_RISK_TYPE	Text	• **first exposure –** outcomes are only added to first drug exposures • **any exposure –** outcomes may be added to any drug exposure • **insidious –** outcomes are randomly added on a date during any exposure • **accumulative –** outcomes are added during any drug exposure, with accumulating probability over time
OUTCOME_ONSET_DAYS_MIN	Number	Minimum days from drug exposure start date for outcome to occur; this column can be set for any outcome_risk_type
OUTCOME_ONSET_DAYS_MIN	Number	Maximum days from drug exposure start date for outcome to occur; this column only applies to first and any exposure types; value must be >= outcome_onset_days_min

Process Function 3a: Apply Outcomes



Process Description:

1) **Log** starting message
2) **For Each Outcome** in **osim_drug_outcome** table
    a) **Count Rows** with **Outcome**
    b) **Log** outcome rules
    c) **Log** outcome counts
    d) **Calculate Affected Count** of rows with **Outcome from** condition_era
    e) **If Affected Count** < 0 and **risk_or_benefit** = **‘**benefit’
        i) **Randomly Delete Affected Count Rows** with **Outcome**
        ii) **Log** affected rows message
    f) **If Affected Count** >= 0 and **risk_or_benefit** = **‘**risk’
        i) If **outcome_risk_type** = ‘first exposure’
            (1) **Insert Affected Count Rows into** condition_era following first drug exposure
            (2) Set **outcomes_insert_count** = Inserted Row Count
        ii) If **outcome_risk_type** = ‘any exposure’
            (1) **Insert Affected Count Rows into** condition_era following any drug exposure
            (2) Set **outcomes_insert_count** = Inserted Row Count
        iii) If **outcome_risk_type** = ‘accumulative’
            (1) **Insert Affected Count Rows into** condition_era following first drug exposure with accumulating risk during drug exposure

(2) **Set outcomes_insert_count** = Inserted Row Count

iv) If **outcome_risk_type** = ‘insideous’

(1) **Insert Affected Count Rows into** condition_era randomly during any drug exposure

(2) Set **outcomes_insert_count** = Inserted Row Count

v) **Log** affected row count message

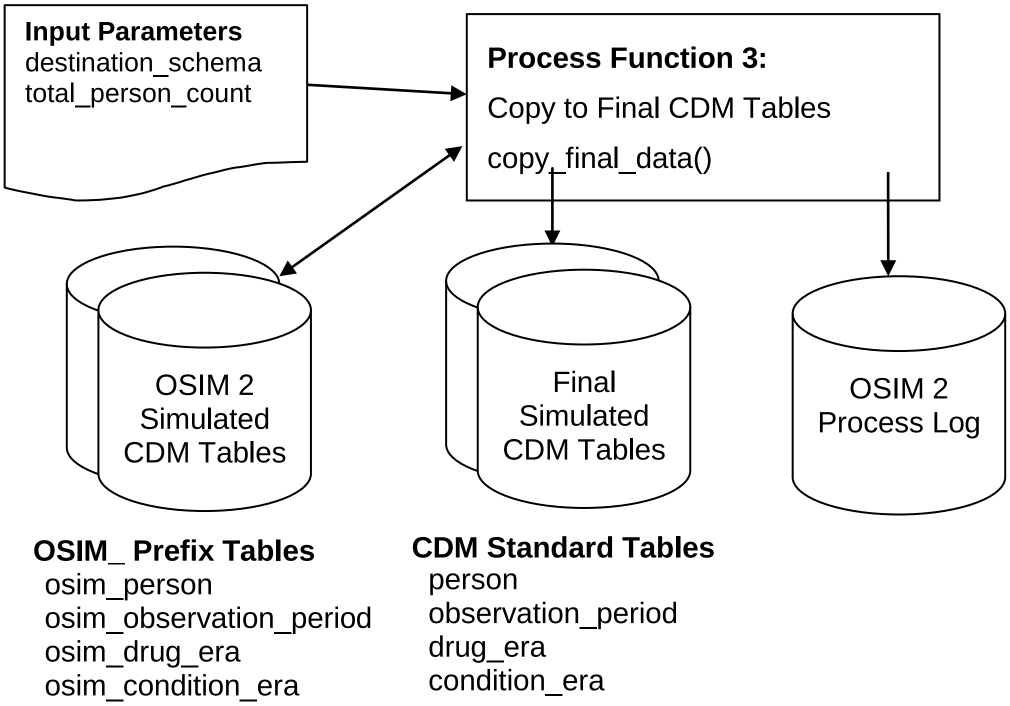
g) **Commit**

3) **Log** complete message

OSIM 2 Procedure 3: Copy to Final CDM Tables

Copy simulated data from 'osim_' prefixed tables to standardized CDM tables in the specified schema. An additional parameter of minimum persons created can be set for parallel runs. The copy will not occur until the entire person count is reached and the last parallel simulation calls the procedure.

Process Function 3: Final copy



Process Description:

1) **Count persons** in **osim_person**
2) **If persons count** = **total_person_count** (input parameter)
   a) **Create** osim_ Indexes
   b) **Log** starting message
   c) **Create** Dynamic SQL to **Copy osim_person to (destination_schema).person**
   d) **Execute Copy**
   e) **Log** copy counts
   f) **Create** Dynamic SQL to **Copy osim_observation_period to (destination_schema).observation_period**
   g) **Execute Copy**
   h) **Log** copy counts

i) **Create** Dynamic SQL to **Copy osim_condition_era to (destination_schema).condition_era**

j) **Execute Copy**

k) **Log** copy counts

l) **Create** Dynamic SQL to **Copy osim_drug_era to (destination_schema).drug_era**

m) **Execute Copy**

n) **Log** copy counts

o) **Log** copy complete message

Revision History

Revision Date	Person Resp.	Reason for Revision
23 Dec 2010	R. Murray	Draft Version
30 Dec 2010	R Murray	Final Version
03 Jan 2011	R Murray	Updated Corporate Logo