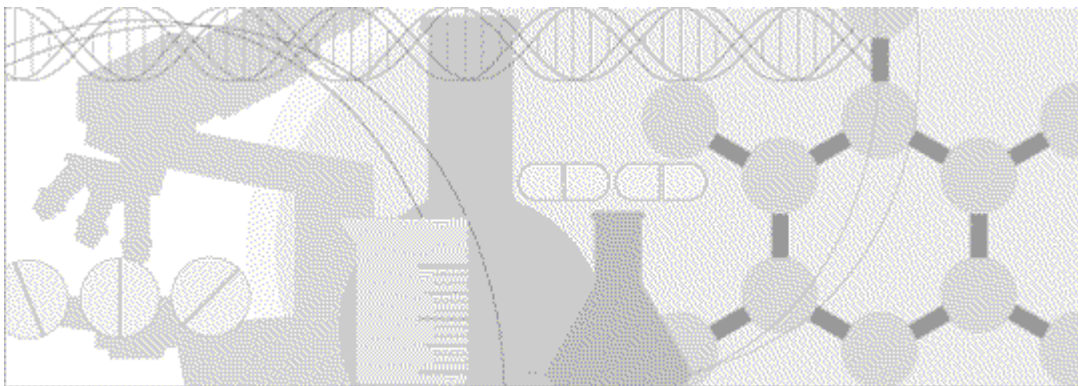


Transferring Data from the Common Data Model

Oracle[®] Health Sciences Empirica Healthcare Analysis
Release 1.0.2



ORACLE[®]

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

This documentation may include references to materials, offerings, or products that were previously offered by Phase Forward Inc. Certain materials, offerings, services, or products may no longer be offered or provided. Oracle and its affiliates cannot be held responsible for any such references should they appear in the text provided.

Contents

About this guide	v
Overview of this guide.....	vi
Audience	vi
Documentation	vii
Documentation accessibility.....	vii
If you need assistance.....	viii
Finding Empirica Healthcare Analysis information and patches on My Oracle Support.....	viii
Finding Oracle documentation	ix
Chapter 1 Introduction	1
Requirements for the OMOP Common Data Model.....	2
Required software for the source database server.....	2
Requirements for source data.....	2
Required columns.....	2
Tables that are copied from the source database.....	4
Overview of the installation and data transfer process.....	5
Before you install.....	7
Errors and failures.....	7
Tables that track the progress of the installation and data transfer.....	8
Chapter 2 Install the ETL utility	11
Decisions to make before you begin.....	12
Prerequisites and information to collect before you begin	13
Preparing the Empirica Healthcare Analysis database server.....	15
Setting installation and data transfer settings.....	16
Installing the ETL utility	25
How do I know whether the installation was successful?	26
What do I do if an error or other issue occurs during installation?.....	27
Editing the etl.properties file and installing the ETL utility again.....	28
Chapter 3 Transfer data and import configurations	31
Transferring data to the Empirica Healthcare Analysis database.....	32
Monitoring the data transfer.....	33
What do I do if an issue occurs during the data transfer?.....	35
Resuming the data transfer	36
Installing and transferring data again.....	36
Determining whether the data transfer process was successful	38
How do I confirm that data was imported?	38
How do I confirm that validation was successful?	38
How do I confirm that data was transformed correctly?	39
How do I confirm that configurations were created?.....	39
Importing and validating configurations	40
Appendix A Exporting source data (Only for importing using files)	43
Exporting source data and making the data files accessible.....	44
Appendix B Transformation details	47
Columns that are added when data is transformed	48
How added columns are populated.....	49

Configurations that are created 51
Views that are created 52
Tables that are created 53

Appendix C Configuration details 55

About this appendix 56
Indexes 57
DM_CONFIGS 58
SOURCE_INFO_ERA 59
SOURCE_INFO_EXP 60
DRILLDOWN_MAP_ERA 61
DRILLDOWN_MAP_EXP 63
CFG_ERA 65
CFG_EXP 70

About this guide

In this preface

Overview of this guide.....	vi
Documentation	vii
If you need assistance.....	viii

Overview of this guide

This guide describes how to use the Empirica Healthcare Analysis ETL utility to transfer data in the OMOP Common Data Model (CDM) Version 4 from a source system to the Empirica Healthcare Analysis database for use in Empirica Healthcare Analysis release 1.0.2 or later.

Audience

This guide is for anyone who prepares clinical data that conforms to the OMOP Common Data Model (CDM) for use in the Empirica Healthcare Analysis database.

You should be comfortable working in SQL*Plus and a command shell and reviewing log files and status tables. Additionally, you should be familiar with the OMOP Common Data Model and how your data is stored.

Documentation

The product documentation is available from the following locations:

- **Oracle Software Delivery Cloud** (<https://edelivery.oracle.com>)—The complete documentation set.
- **My Oracle Support** (<https://support.oracle.com>)—*Release Notes* and *Known Issues*.
- **Oracle Technology Network** (<http://www.oracle.com/technetwork/documentation>)—The most current documentation set, excluding the *Release Notes* and *Known Issues*.

All documents may not be updated for every Empirica Healthcare Analysis release. Therefore, the version numbers for the documents in a release may differ.

Document	Description	Last updated
<i>Release Notes</i>	The <i>Release Notes</i> document provides descriptions of enhancements and bug fixes as well as system requirements.	1.0.2
<i>Known Issues</i>	The <i>Known Issues</i> document provides detailed information about the known issues in this release, along with workarounds, if available.	1.0.2
<i>User Guide</i>	The <i>User Guide</i> describes how to use the Empirica Healthcare Analysis application to perform epidemiologic and statistical analyses of healthcare and administrative claims data.	1.0.1
<i>Installation Guide</i>	The <i>Installation Guide</i> document describes how to install the Empirica Healthcare Analysis software.	1.0.1
<i>Secure Configuration Guide</i>	The <i>Secure Configuration Guide</i> provides guidance and recommendations on securely installing, configuring, and managing the Empirica Healthcare Analysis software and its system components.	1.0.2
<i>Transferring Data from the Common Data Model</i>	The <i>Transferring Data from the Common Data Model</i> document describes how to use the Empirica Healthcare Analysis ETL utility to transfer data in the OMOP Common Data Model (CDM) Version 4 from a source system to the Empirica Healthcare Analysis database.	1.0.2
<i>Third Party Licenses and Notices</i>	The <i>Third Party Licenses and Notices</i> document includes licenses and notices for third party technology that may be included with the Empirica Healthcare Analysis software.	1.0.2

Documentation accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

If you need assistance

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Finding Empirica Healthcare Analysis information and patches on My Oracle Support

The latest information about the Empirica Healthcare Analysis application is on the Oracle Support self-service website, My Oracle Support. Before you install and use the Empirica Healthcare Analysis application, check My Oracle Support for the latest information, including *Release Notes* and *Known Issues*, alerts, white papers, bulletins, and patches.

Creating a My Oracle Support account

You must register at My Oracle Support to obtain a user name and password before you can enter the site.

- 1 Open a browser to <https://support.oracle.com>.
- 2 Click the **Register** link.
- 3 Follow the instructions on the registration page.

Finding information and articles

- 1 Sign in to My Oracle Support at <https://support.oracle.com>.
- 2 If you know the ID number of the article you need, enter the number in the text box at the top right of any page, and then click the magnifying glass icon or press **Enter**.
- 3 To search the knowledge base, click the **Knowledge** tab, and then use the options on the page to search by:
 - Product name or family.
 - Keywords or exact terms.

Finding patches

You can search for patches by patch ID or number, product, or family.

- 1 Sign in to My Oracle Support at <https://support.oracle.com>.
- 2 Click the **Patches & Updates** tab.
- 3 Enter your search criteria and click **Search**.
- 4 Click the patch ID number.

The system displays details about the patch. You can view the Read Me file before downloading the patch.

- 5 Click **Download**, and then follow the instructions on the screen to download, save, and install the

patch files.

Finding Oracle documentation

The Oracle website contains links to Oracle user and reference documentation. You can view or download a single document or an entire product library.

Finding Oracle Health Sciences documentation

For Oracle Health Sciences applications, go to the Oracle Health Sciences Documentation page at <http://www.oracle.com/technetwork/documentation/hsgbu-clinical-407519.html>.

Note: Always check the Oracle Health Sciences Documentation page to ensure you have the most up-to-date documentation.

Finding other Oracle documentation

- 1 Do one of the following:
 - Go to <http://www.oracle.com/technology/documentation/index.html>.
 - Go to <http://www.oracle.com>, point to the **Support** tab, and then click **Product Documentation**.
- 2 Scroll to the product you need, and click the link.

CHAPTER 1

Introduction

In this chapter

Requirements for the OMOP Common Data Model.....	2
Overview of the installation and data transfer process.....	5
Tables that track the progress of the installation and data transfer.....	8

Requirements for the OMOP Common Data Model

You can transfer data from the OMOP Common Data Model Version 4. You can find details of the data model standard in the following document:

OMOP CDM Version 4.0 Specification April 2012.pdf

This document is available from the following location:

<http://omop.org/CDM>

Required software for the source database server

The source database must use one of the following:

- Oracle Database 11g Release 2 (11.2.0.4).
- Oracle Database 12c Release 1 (12.1.0.2).

Requirements for source data

- All PERSON_ID values that you import must have 9 or fewer digits, or transformation fails.
- At least one OBSERVATION_PERIOD row must exist for each person.
- If you varied from the default Common Data Model Version 4 and added an Age column to the PERSON table, and you want to import this column, the name of the column must:
 - Contain 30 or fewer characters.
 - Begin with an alphabetic character.
 - Contain only the following characters:
 - Alphanumeric characters
 - Underscores (_)
 - Dollar signs (\$)
 - Number signs (#)

Required columns

For each of the following source tables, the Empirica Healthcare Analysis application requires the following columns and requires that they be non-null. All columns except DRUG_EXPOSURE.DAYS_SUPPLY are also required by the OMOP Common Data Model.

The validation step in the ETL utility checks whether NOT NULL constraints exist on each of the following columns. If the constraints do not exist, the ETL utility creates them in the transferred data.

Source table	Columns
PERSON	PERSON_ID GENDER_CONCEPT_ID YEAR_OF_BIRTH
OBSERVATION_PERIOD	PERSON_ID OBSERVATION_PERIOD_START_DATE OBSERVATION_PERIOD_END_DATE
DRUG_ERA	PERSON_ID DRUG_CONCEPT_ID DRUG_ERA_START_DATE DRUG_ERA_END_DATE
DRUG_EXPOSURE	PERSON_ID DRUG_CONCEPT_ID DRUG_EXPOSURE_START_DATE DAYS_SUPPLY
CONDITION_ERA	PERSON_ID CONDITION_CONCEPT_ID CONDITION_ERA_START_DATE
CONDITION_OCCURRENCE	PERSON_ID CONDITION_CONCEPT_ID CONDITION_START_DATE
PROCEDURE_OCCURENCE	PERSON_ID PROCEDURE_CONCEPT_ID PROCEDURE_DATE

Tables that are copied from the source database

Your source data and vocabulary information in the OMOP Common Data Model (CDM) might be in separate accounts or in the same account. One account is created to hold all CDM data in the destination database, which is the Empirica Healthcare Analysis database.

The following table lists the tables that you can transfer to the account that is created by the ETL utility in the Empirica Healthcare Analysis database. The Empirica Healthcare Analysis application supports only these tables.

When you set installation and data transfer settings, you specify the tables to transfer.

Table	Contains	Contents of the table	Source account
PERSON	Patient information	One record per patient.	Data
OBSERVATION_PERIOD	Date ranges during which data has been collected	Might contain more than one record per patient.	Data
DRUG_ERA	Drug information	One row per each span of time when a person is exposed to a drug.	Data
DRUG_EXPOSURE	Drug information	One record for each dispensation of a drug.	Data
CONDITION_ERA	Condition information	Information about the chronological period of condition occurrence combining individual occurrences into a single era.	Data
CONDITION_OCCURRENCE	Condition information	One record for each condition.	Data
PROCEDURE_OCCURRENCE	Procedure information	One record for each procedure.	Data
CONCEPT	Coded field translations	One record for each code. All codes are stored in the same table.	Vocabulary

Overview of the installation and data transfer process

The Empirica Healthcare Analysis ETL utility moves data and vocabulary information that is in the OMOP Common Data Model Version 4 from a source database to a new data account in the Empirica Healthcare Analysis database. The new data account is the destination account.

The ETL utility includes two execution scripts, one for starting the ETL process and another for resuming the ETL process, which you use only if an issue occurs while data is transferring. Both scripts create log files with processing and error information.

The ETL utility does not have a graphical user interface. You install and execute the ETL utility in SQL*Plus.

Additionally, the ETL utility allows you to move data one time. It does not allow for periodic updates of data, though you can run the ETL utility again to transfer data into a different destination account.

Transferring the data involves the following steps.

Step 1. Install the ETL utility

You install the Empirica Healthcare Analysis ETL utility by running a SQL script (`install_etl.sql`) while connected to the destination database, which is the Empirica Healthcare Analysis database. The installation typically takes no more than several minutes.

The installation of the ETL utility accomplishes the following tasks:

- 1 A destination account that will contain the transferred and transformed data is created in the Empirica Healthcare Analysis database.
- 2 If you are importing using a database link, database links are created from the destination account to the source database account(s).
- 3 The ETL utility creates required tables and packages in the destination account.

After the installation is complete, you can start the extract, transfer, and load process by executing the Empirica Healthcare Analysis ETL utility.

Step 2. Use the ETL utility to extract, load, and transfer the data

You transfer data by executing the ETL utility. If you have data for millions of patients, this step can take several hours or longer.

The data transfer process consists of the following steps. A step must complete successfully before the next step starts. After a step completes, the ETL utility starts the next step.

Step	What the ETL utility does
1. Import the data	Transfers data and vocabulary from the source database to the Empirica Healthcare Analysis database. The method of importing data varies, depending on whether you import using a database link or from data files.

Step	What the ETL utility does
2. Validate the data	<p>Validates that the tables that were imported into the destination account have the correct structure and information for use by the Empirica Healthcare Analysis application.</p> <p>Validation checks that the following requirements are met:</p> <ul style="list-style-type: none">• Tables were created for the tables that you imported.• Required columns exist in each table, and columns do not contain null values.• Foreign Key constraints exist. <p>For the drug, condition, and procedure tables, the PERSON_ID has a corresponding row in the PERSON table.</p>
3. Transform the data	<p>Transforms the data in the data tables in the Empirica Healthcare Analysis database by adding columns, populating values, and deriving additional items as necessary. Coded fields are translated using the CONCEPT table and stored in the data tables in the Empirica Healthcare Analysis database.</p>
4. Create configurations	<p>Creates default configurations in the Empirica Healthcare Analysis database. The ETL utility creates the tables that are needed to define one or more configurations.</p>

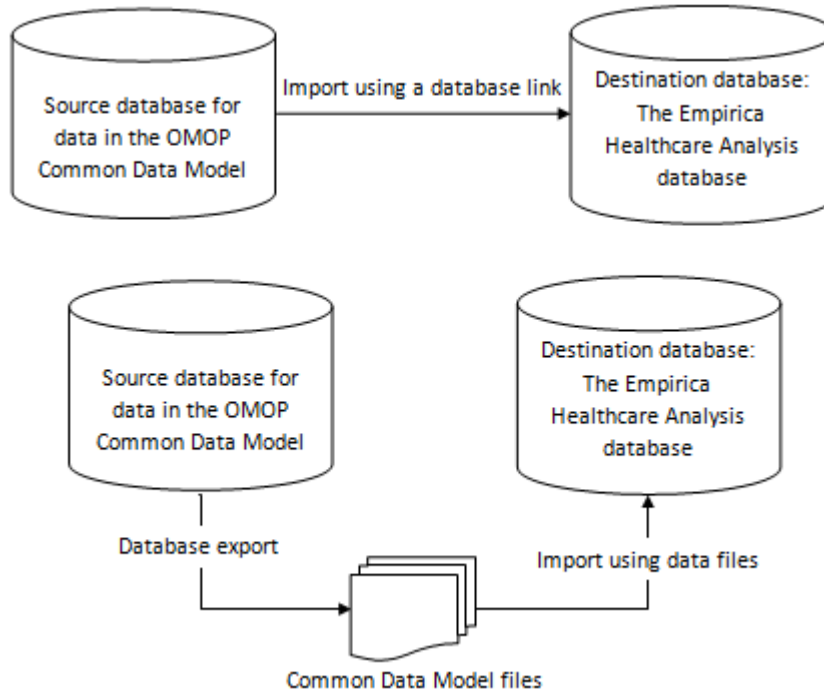
Step 3. Import configurations in the Empirica Healthcare Analysis application so that users can access the data

After the data transfer is complete, you must import and validate the configurations that are created in the database into the Empirica Healthcare Analysis application so that the data is accessible to users. After the configurations are validated in the Empirica Healthcare Analysis application, users can access the data.

Before you install

Before installing the Empirica Healthcare Analysis ETL utility, you must decide how you will transfer data, either by using a database link or by exporting to and then importing from data files. You specify this decision and other options in a properties file. Data is transferred using the Oracle Data Pump technology either across a database link or using exported files.

The following images describe your options for transferring data. Typically, the source and destination databases are hosted on different servers across a network, sometimes in different domains. However, the databases can be on the same machine and even within the same database instance.



Errors and failures

If an error occurs during installation, you can edit the properties file to correct the error, drop the destination account, and then install again.

If an error occurs during the data transfer, you can either resume the process or edit the properties file, install the ETL utility again, and start over, depending on the cause of the error.

This document contains troubleshooting information to help in case of errors or failures.

Tables that track the progress of the installation and data transfer

The following tables, which the ETL utility creates in the destination account, allow you to track the progress of your installation and data transfer.

Do not modify these tables directly.

ETL_PROPERTIES

This table saves processing options. Properties are added to the file when you install and execute the ETL utility.

The ETL utility uses the information in this table to run or resume the execution of the ETL utility, if it is interrupted.

Primary Key: NAME

Field	Description
NAME	Name of a property from the etl.properties file.
VALUE	Value of the property.

ETL_STATUS

This table saves information about the steps that have executed or are currently executing.

Rows are added to the table during each step of the data transfer.

Primary Key: STEP_NAME

Field	Description
STEP_NAME	Name of each step that is performed in the data transfer.
START_DATE	Date and time when the step began.
END_DATE	Date and time when the step finished, for a step that either completed or failed.
STATUS	Status of the step: <ul style="list-style-type: none"> • InProgress • Completed • Stopped • Failed • Skipped
NEXT_PERSON_ID	For the transformation process, the first PERSON_ID of the next batch to be processed.

Field	Description
PCT_COMPLETE	For import steps, an indicator of the percentage of the table that has been transferred.

CHAPTER 2

Install the ETL utility

In this chapter

- Decisions to make before you begin12
- Prerequisites and information to collect before you begin13
- Preparing the Empirica Healthcare Analysis database server15
- Setting installation and data transfer settings16
- Installing the ETL utility25
- How do I know whether the installation was successful?26
- What do I do if an error or other issue occurs during installation?27

Decisions to make before you begin

Before you install the Empirica Healthcare Analysis ETL utility, you must determine the method for transferring data from the source database to the Empirica Healthcare Analysis database. You have the following options. Both options use the Oracle Data Pump technology to transfer the data to the Empirica Healthcare Analysis database.

- Import across a database link.

A remote database link can save local disk space on the Empirica Healthcare Analysis server. If you have limited space on the Empirica Healthcare Analysis server, consider using this option.

Keep in mind that ETL activity is subject to network traffic and performance while the ETL execution script runs over a database link. If you experience network latency issues, or if you are running other applications in the destination database, importing across a database link could be a lengthy process.

- Import using export files created by the Oracle Data Pump technology.

You create the export files at the source instance and move the files to the Empirica Healthcare Analysis database server. Complete this step at your convenience before executing the ETL utility.

You have the following options for storing the files:

- Save them on the source server, and then transfer them to the Empirica Healthcare Analysis server.
- Export the files to an external hard drive and manually move it to the Empirica Healthcare Analysis server.
- Export the files to a network file location that is accessible to both servers.

The export files that you create could be very large. Make sure that you have sufficient space for storing the files.

If you disregard the time spent transferring data files to a locally accessible folder on the Empirica Healthcare Analysis server, a data file import might be faster than a database link import.

For information about exporting source data, see *Exporting source data and making the data files accessible* (on page 44).

Prerequisites and information to collect before you begin

For information about the requirements for the OMOP Common Data Model, see *Requirements for the OMOP Common Data Model* (on page 2).

Information to collect for all installations

- The name of the tablespace in the Empirica Healthcare Analysis database where the imported data for the destination account will be stored. Oracle recommends using separate tablespaces for the imported data and the main Empirica Healthcare Analysis data.
- User name for each of the following accounts:
 - Data account in the source database.
This account contains data and sometimes vocabulary information.
 - Vocabulary account in the source database, if the vocabulary account is separate from the data account.
This account contains vocabulary information if the information is not included in the data account.
- The names of the tablespaces used by the accounts in the source database.
- Connect string to the Empirica Healthcare Analysis database from the server on which you will be running the ETL utility.

Additional information to collect for importing using a database link

- Password for each of the following accounts:
 - Data account in the source database.
 - Vocabulary account in the source database, if the account exists.
- Value of the `global_names` initialization parameter in the Empirica Healthcare Analysis database.
To determine the value, in SQL*Plus, type the following command:


```
SQL> show parameter global_names;
```
- If the value of the `global_names` initialization parameter is `TRUE`, obtain the global name of the **source database** using the following command:


```
SQL > select * from global_name;
```
- Connect string from the Empirica Healthcare Analysis database to the source database.
The connect string is either a net service name or an EZCONNECT connection.
If the value of the Empirica Healthcare Analysis database parameter `global_names` is `TRUE`, the net service name used to connect from the Empirica Healthcare Analysis database to the source database must use the source `global_name`.

Additional information to collect for importing data using files

You created the data files and import directory when you exported source data. For more information, see *Exporting source data and making the data files accessible* (on page 44).

- Name of the data file you exported from the source database.
- (If the vocabulary account is separate from the data account) Name of the vocabulary file you exported from the source database.
- Location of the import directory on the Empirica Healthcare Analysis database server. You can use an existing directory or create a new directory. For instructions on creating a directory, see *Exporting source data (Only for importing using files)* (on page 43).

Prerequisites for all installations

- Empirica Healthcare Analysis Release 1.0.2 or later must be installed.
- The data you are transferring must use the OMOP Common Data Model Version 4.
- The Empirica Healthcare Analysis database must have sufficient available tablespace for the data and vocabulary information that is being transferred.

The Empirica Healthcare Analysis database requires approximately 2.5 times the amount of space required for the source tables. You can use an existing tablespace or a newly created tablespace.

- You must have access to a privileged account, such as sys or system, for logging in to the source database.

The account must have privileges to create and edit a user and grant basic privileges.

- If you are working on a network that limits your connection time, the limit must be greater than the time required for each step to complete.

Different factors affect the time required for each step. Oracle recommends working on a network with no limits for connection times.

Additional prerequisites for installations using a database link

If the value of the Empirica Healthcare Analysis database parameter `global_names` is `TRUE`, the following prerequisites apply:

- The source data account and the source vocabulary accounts must be in a single account. If they are not, you must copy the `CONCEPT` table in the source vocabulary account into the data account before installing.
- The source database global name must contain a domain component, for example, `orcl.us.acme.com`.

Preparing the Empirica Healthcare Analysis database server

Perform the following steps regardless of how you plan to import data.

- 1 Obtain the installation ZIP file, Healthcare-CDM4_ETL_XXX.zip, where XXX is the build number.
- 2 Unpack the ZIP file into an empty folder in one of the following locations:
 - The Empirica Healthcare Analysis database server.
 - A location where you can reach the Empirica Healthcare Analysis database server.

Setting installation and data transfer settings

You set installation and data transfer settings in the `etl.properties` file. The values that you specify determine how the ETL utility is installed and how your data is moved from the source database to the Empirica Healthcare Analysis database.

You must enclose all values in single quotation marks, for example, **'Y'**. To provide a blank value, type two single quotes (`''`).

Do not comment out any lines in the `etl.properties` file.

- 1 In a command shell, set your current directory to the location where you unpacked the installation ZIP file, `Healthcare-CDM4_ETL_xxx.zip`, where `xxx` refers to the build number.

For example:

```
$ cd /u01/stage/cdm4etl
```

- 2 Type the following command to create a copy of the `template_etl.properties` file, and rename the file to `etl.properties`:

```
$ cp template_etl.properties etl.properties
```

- 3 Open the `etl.properties` file in a text editor, such as `vi`.
- 4 Provide values for the following settings. These settings are used when the ETL utility is installed.

Note: If you provide a non-blank value for the `source_connect` property, the installation attempts to use a database link for the import, even if you specify non-blank values for properties required by a file import. If `source_connect` is blank, the installation attempts to import using data files.

Property	Non-blank value required	Value to provide
<code>source_connect</code>	Yes, if you plan to install using a database link.	<p>Connect string from the Empirica Healthcare Analysis database to the source database.</p> <p>The connect string must be one of the following:</p> <ul style="list-style-type: none"> • A net service name that is specified in the <code>tnsnames.ora</code> file on the Empirica Healthcare Analysis database server and that indicates how to connect to the source server. • An <code>EZCONNECT</code> connection string in the following format: hostname:port/service <p>If you provide a non-blank value, the ETL utility imports data using a database link.</p> <p>Example for importing using a database link: <code>'ORCLSRC'</code></p> <p>Value to use for importing using files: <code>''</code></p>

Property	Non-blank value required	Value to provide
source_global_name	Yes, if you plan to install using a database link, and if the global_names initialization parameter in the Empirica Healthcare Analysis database is set to TRUE.	Global name of the source database. Example when global_names is TRUE: 'ORCL.world' Value to use for importing using a database link when the global_names initialization parameter in the Empirica Healthcare Analysis database is set to FALSE: " Value to use for importing using files: "
source_schema	Yes	Name of the account that contains the data tables in the source database. Example: 'CDM_SOURCE'
vocab_schema	Yes	Name of the account that contains the CONCEPT table in the source database. If the CONCEPT table is in the same account as the data, provide the name of the account that contains the data tables. Example: 'CDM_VOCABULARY'
source_tablespaces	Yes	Comma-delimited list of the tablespace(s) that are in the source database and that contain the data and vocabulary accounts that you are importing. Example: 'CDM4DATA, CDM4VOC'
dest_connect	Yes	Connect string for the Empirica Healthcare Analysis database. Use the TNS connection that you use to connect to the Empirica Healthcare Analysis database from the machine where you install the ETL utility. Example: 'ORCLDEST'

Property	Non-blank value required	Value to provide
dest_schema	Yes	<p>Name of the destination account that will be created in the Empirica Healthcare Analysis database to hold the data and vocabulary information from the source database.</p> <p>The name must:</p> <ul style="list-style-type: none"> • Not already exist. • Contain 30 or fewer characters. • Begin with an alphabetic character. <p>The name can contain only the following characters:</p> <ul style="list-style-type: none"> • Alphanumeric characters • Underscores (_) • Dollar signs (\$) • Number signs (#). <p>Example: 'CDM_DEST'</p>
dest_tablespace	Yes	<p>Name of the default tablespace for the destination account.</p> <p>Example: 'CDMDEST'</p>
healthcare_master	Yes	<p>Name of the existing account for the Empirica Healthcare Analysis database. You created this account when you installed the Empirica Healthcare Analysis application.</p> <p>This account is different from the destination account that you are creating to hold the data that you transfer. You specify the name of the destination account using the dest_schema property.</p> <p>Example: 'HEALTHCARE'</p>

- 5 Provide values for the following settings. These settings are used when data is imported into the Empirica Healthcare Analysis database.

PERSON and OBSERVATION_PERIOD tables are always imported.

Property	Non-blank value required	Value to provide
data_filename	Yes, if you plan to import using data files.	Name of the file containing the data to be moved to the Empirica Healthcare Analysis database. Example: 'data_export.dmp' Value to use for importing using a database link: "
vocab_filename	Yes, if you plan to import using data files.	Name of the file containing the vocabulary information. If the CONCEPT table is contained in the data export file, use the value that you provided for data_filename. Example: 'voc_export.dmp' Value to use for importing using a database link: "
import_directory	Yes, if you plan to install using data files.	Name of the directory from which you are importing data. Example: 'ETL_DIR' Value to use for importing using a database link: "
load_drug_era	Yes	'Y' or 'N', indicating whether to import the DRUG_ERA table. You are required to import at least one drug table. Therefore, at least one of the load_drug_era and load_drug_exposure properties must be set to 'Y'.
load_drug_exposure	Yes	'Y' or 'N', indicating whether to import the DRUG_EXPOSURE table. You are required to import at least one drug table. At least one of the load_drug_era and load_drug_exposure properties must be set to 'Y'.
load_condition_era	Yes	'Y' or 'N', indicating whether to import the CONDITION_ERA table. You are required to import exactly one condition table. Therefore, if load_condition_era is 'Y', load_condition_occurrence must be 'N', and vice versa.

Property	Non-blank value required	Value to provide
load_condition_occurrence	Yes	'Y' or 'N', indicating whether to import the CONDITION_OCCURRENCE table. You are required to import exactly one condition table. Therefore, if load_condition_occurrence is 'Y', load_condition_era must be 'N', and vice versa.
load_procedure_occurrence	Yes	'Y' or 'N', indicating whether to import the PROCEDURE_OCCURRENCE table.

- 6 Provide values for the following settings. These settings are used when data is transformed for use in the Empirica Healthcare Analysis database.

Property	Non-blank value required	Value to provide
commit_count	Yes	Size of the batch in which records in the tables are processed based on the PERSON_ID column in each table. This value must be a number that is greater than or equal to 100. A smaller number can increase the time of the data transfer but allows you to check the progress in the ETL_STATUS table. Consider setting this value to 1 percent of the rows in the PERSON table. Example: '100000'

Property	Non-blank value required	Value to provide
compute_age	Yes	<p>'Y' or 'N', indicating whether to populate the AGE and AGE_GROUP columns in the PERSON table.</p> <ul style="list-style-type: none"> • If the value is 'Y', values for the AGE column are populated during the transformation process in one of the following ways: <ul style="list-style-type: none"> ▪ Values are derived using the default age calculation provided by the ETL utility when age_column is ". The default age calculation subtracts YEAR_OF_BIRTH from the year part of ENROLLMENT_START_DATE. ▪ Values are populated using an Age column that you specify using the age_column property. If you varied from the default Common Data Model Version 4 and your PERSON table includes a column for age, you can use this column in place of the age calculation. Specify the column name in the age_column property, which appears next in this table. • If the value is 'N', values for the AGE_GROUP column are derived based upon the value in the AGE column. <p>Note: If the value is 'N' and you specify a column for age_column, the AGE column is populated with values from the column, but AGE_GROUP is not calculated.</p>

Property	Non-blank value required	Value to provide
age_column	No	<p>In the PERSON table, the name of the column that contains age. The value of age must be specified in years.</p> <p>If you do not have an age column, but you want the configuration to include a calculated age column, the transformation process can create and derive AGE if you set compute_age, which appears before this property in this table, to 'Y', and set age_column to ". Age is computed in years.</p> <p>If you provide a non-blank value:</p> <ul style="list-style-type: none"> • The column datatype must be numeric. • The column name must: <ul style="list-style-type: none"> ▪ Contain 30 or fewer characters. ▪ Begin with an alphabetic character. <p>The column name can contain only the following characters:</p> <ul style="list-style-type: none"> • Alphanumeric characters • Underscores (_) • Dollar signs (\$) • Number signs (#). <p>Example: AGE_AT_ENROLLMENT</p>
parallel_update	Yes	<p>'Y' or 'N', indicating whether to use parallel updates during the transformation process.</p> <p>The update statements used to perform the data transformation generally perform better using parallel execution. However, in some cases, bottlenecks can occur, so the utility allows you to turn off parallel execution.</p>

- 7 Provide values for the following settings. These settings are used when configurations are created in the Empirica Healthcare Analysis database.

Property	Non-blank value required	Value to provide
config_name_era	Yes	<p>Name to use for the Drug Era configuration. The name must not be used by an existing configuration.</p> <p>The Drug Era configuration is created only when the load_drug_era property is set to 'Y'.</p> <p>If load_drug_era is set to 'Y', you must provide a value.</p> <p>The values for config_name_era and config_name_exposure must:</p> <ul style="list-style-type: none"> • Be different. • Contain 100 or fewer characters. <p>The values can contain only the following characters:</p> <ul style="list-style-type: none"> • Alphanumeric characters. • Underscores (_). • Parentheses ([or]). • Blank space. • Plus sign (+) or minus sign (-). <p>Example: 'CDM Era'</p>

Property	Non-blank value required	Value to provide
config_name_exposure	Yes	<p>Name to use for the Drug Exposure configuration. The name must not be used by an existing configuration.</p> <p>The Drug Exposure configuration is created only when the load_drug_exposure property is set to 'Y'.</p> <p>If load_drug_exposure is set to 'Y', you must provide a non-blank value.</p> <p>The values for config_name_era and config_name_exposure must:</p> <ul style="list-style-type: none"> • Be different. • Contain 100 or fewer characters. <p>The values can contain only the following characters:</p> <ul style="list-style-type: none"> • Alphanumeric characters. • Underscores (_). • Parentheses ([or]). • Blank space. • Plus sign (+) or minus sign (-). <p>Example: 'CDM Exposure'</p>
parallel_ddl	Yes	<p>'Y' or 'N', indicating whether to use parallel processes for creating tables and indexes.</p> <p>The performance for creating tables and indexes is generally better when you use parallel execution. However, in some cases, bottlenecks can occur, so the utility allows you to turn off parallel execution.</p>

Installing the ETL utility

- 1 In a command shell, set your current directory to the location where you unpacked the installation ZIP file, Healthcare-CDM4_ETL_xxx.zip, where xxx refers to the build number.

For example:

```
$ cd /u01/stage/cdm4etl
```

- 2 In the command shell, log in to the Empirica Healthcare Analysis database using a privileged user, such as sys or system. The user must have privileges to create a user and grant privileges to the user.

For example:

```
$ sqlplus <privileged_user_name>@<connect_string>
```

You are prompted to type a password for the user.

- 3 Type a password, and press **Enter**.
- 4 In SQL*Plus, execute the installation script, install_etl.sql. This file is in the installation kit.

```
SQL>@install_etl.sql
```

If you set properties so that you import using a database link, the installation prompts you to provide and confirm the passwords for the data and vocabulary accounts in the source database.

For all installations, the installation prompts you to create a password for the destination account. Remember this value for later when you check whether the installation was successful and execute the ETL utility.

Executing the installation script writes a log file, install_etl.log, in the current directory.

If you stop the installation, you can change the etl.properties file, if needed, drop the destination account, and install again. For more information, see *Editing the etl.properties file and installing the ETL utility again* (on page 28).

How do I know whether the installation was successful?

The installation of the ETL utility typically takes no more than a couple minutes.

- 1 In the current directory, open the `install_etl.log` file, and make sure that:
 - The last entry in the file says `Installation Complete`.
 - The file contains no errors.
- 2 Review the `ETL_STATUS` table for errors:
 - a Log in to the Empirica Healthcare Analysis database using the destination account. You provided the name for the destination account in the `etl.properties` file and supplied a password during installation.
 - b In SQL*Plus, review the status table using the following command:

```
SQL>SELECT * FROM ETL_STATUS;
```
 - c Make sure **Completed** appears for the status of the Install step.

What do I do if an error or other issue occurs during installation?

You must address any errors that occur during installation before attempting to transfer data. Typically, you address errors by changing the value of one or more properties, dropping the destination account, and rerunning the `install_etl.sql` script.

Issue	Resolution
<p>The error ORA-39002: invalid operation occurred with one or both of the following errors:</p> <ul style="list-style-type: none"> ORA-39070: unable to open the log file ORA-39087: directory name <DirectoryName> is invalid error 	<p>Make sure that:</p> <ul style="list-style-type: none"> The specified directory in the database points to an existing and accessible location. The <code>import_directory</code> parameter is specified correctly. <p>Make any necessary changes. For more information, see <i>Editing the etl.properties file and installing the ETL utility again</i> (on page 28).</p>
<p>You are importing using a database link, and the console output or the installation log file says:</p> <pre>*** Skipping Database link creation ***</pre>	<p>Check that the <code>source_connect</code> property in the <code>etl.properties</code> file has a valid value.</p> <p>To check the values of the properties in the file, run the following command in SQL*Plus:</p> <pre>select name, value from <dest_account>.etl_properties</pre> <p>If the property does not have a valid value, correct the value. For more information, see <i>Editing the etl.properties file and installing the ETL utility again</i> (on page 28).</p>
<p>The console output or the installation log file says:</p> <p>Database Link <LinkName> is unusable</p>	<p>Possible causes:</p> <ul style="list-style-type: none"> You provided the wrong passwords for the import accounts. You entered an invalid value in the <code>source_connect</code> property in the properties file. The <code>source_connect</code> value you put in the properties file doesn't exist in your <code>tnsnames.ora</code> file. <p>Install the ETL utility again with the correct values. For more information, see <i>Editing the etl.properties file and installing the ETL utility again</i> (on page 28).</p>
<p>You are importing using a file import, but the installation attempts to import using a database link.</p>	<p>Make sure that the value of the <code>source_connect</code> property is blank value (two single quotation marks with no space between them: "").</p> <p>Additionally, make sure you specified values for the <code>import_directory</code> and <code>import_files</code> when you exported source data and made the files accessible.</p>

Editing the etl.properties file and installing the ETL utility again

If you need to install the ETL utility subsequent times after the first installation, you must first drop the destination account.

Step 1. Edit the etl.properties file.

- 1 In a command shell, set your current directory to the location where you unpacked the installation ZIP file, Healthcare-CDM4_ETL_XXX.zip, where XXX refers to the build number.

For example:

```
$ cd /u01/stage/cdm4etl
```

- 2 Open the etl.properties file in a text editor, such as vi.
- 3 Provide values for the settings in the etl.properties files. For descriptions of each property, see *Setting installation and data transfer settings* (on page 16).

Step 2. Drop the destination account.

- 1 In the command shell, log in to the Empirica Healthcare Analysis database using a privileged user, such as sys or system. The user must have privileges to create a user and grant privileges to the user.

For example:

```
$ sqlplus <privileged_user_name>@<connect_string>
```

You are prompted to type a password for the user.

- 2 Type a password, and press **Enter**.
- 3 Drop the destination account using the following command:

Note: Drop the destination account, which the ETL utility created, not the account that was created when you installed the Empirica Healthcare Analysis application. You must type the user name for the account correctly. You cannot recover a dropped account.

```
SQL> drop user <user_name> cascade;
```

Example:

```
SQL> drop user CDM_DEST cascade;
```

Step 3. Install the ETL utility again.

- 1 In a command shell, set your current directory to the location where you unpacked the installation ZIP file, Healthcare-CDM4_ETL_XXX.zip, where XXX refers to the build number.

For example:

```
$ cd /u01/stage/cdm4etl
```

- 2 In the command shell, log in to the Empirica Healthcare Analysis database using a privileged user, such as sys or system. The user must have privileges to create a user and grant privileges to the user.

For example:

```
$ sqlplus <privileged_user_name>@<connect_string>
```

You are prompted to type a password for the user.

- 3 Type a password, and press **Enter**.
- 4 In SQL*Plus, execute the installation script, `install_etl.sql`. This file is in the installation kit.
`SQL>@install_etl.sql`

If you set properties so that you import using a database link, the installation prompts you to provide and confirm the passwords for the data and vocabulary accounts in the source database.

For all installations, the installation prompts you to create a password for the destination account. Remember this value for later when you check whether the installation was successful and execute the ETL utility.

The `install_etl.log` file, which is in the current directory, is overwritten when you reinstall.

CHAPTER 3

Transfer data and import configurations

In this chapter

Transferring data to the Empirica Healthcare Analysis database.....	32
Monitoring the data transfer	33
What do I do if an issue occurs during the data transfer?.....	35
Determining whether the data transfer process was successful	38
Importing and validating configurations	40

Transferring data to the Empirica Healthcare Analysis database

If you have data for millions of patients, this process can take several hours or longer.

The destination account, which you specified in the `etl.properties` file and which was created during the installation of the ETL utility, pulls data from one of the following locations:

- For imports that use a database link, the destination account pulls data from the source database using the database link that the installation created.
- For imports that use exported files, the destination account pulls data from the export files you created.

The tables from the source account(s) are imported into the destination account with the same names and structure used in the source tables. The tables are then transformed, and the configurations required by the Empirica Healthcare Analysis application are created. For details, see *Transformation details* (on page 47) and *Configuration details* (on page 55).

- 1 In a command shell, set your current directory to the location where you unpacked the installation ZIP file, `Healthcare-CDM4_ETL_xxx.zip`, where `xxx` refers to the build number.

For example:

```
$ cd /u01/stage/cdm4etl
```

- 2 In the command shell, log in to the Empirica Healthcare Analysis database using the destination account, which was created during installation of the ETL utility. You provided the name of the account in the `etl.properties` file and supplied the password during installation.

For example:

```
$ sqlplus <account_name>@<connect_string>
```

You are prompted to type a password for the destination account.

- 3 Type a password, and press **Enter**.
- 4 In SQL*Plus, execute the ETL script:

```
SQL>@start_etl
```

The ETL utility begins the data transfer process. A log file, `start_etl.log`, is written to the current directory.

Monitoring the data transfer

You check the progress of each step of the data transfer, including whether each step is successful, by querying the ETL_STATUS table in the destination account. The results of the query include:

- The steps that have been or are being performed.
- The status of each step.
- The number of PERSON_IDs that have been processed for the transformation steps.

You can check the progress at any time during and after the data transfer process.

The imports of the data and vocabulary accounts are completed sequentially. Each table is imported separately, so that in case of failure, resuming the process imports only the tables that have not been successfully imported.

Best practices for monitoring the data transfer

- After starting the data transfer, check to make sure that the Import Person step completes.

The time to complete the Import Person step varies, depending on the size of the data. For example, a data set with patients in the low millions might take a few minutes, though the time varies depending on many factors. If you are importing using files, you can use metrics gathered from the file export to estimate the time needed for the file import.

The import of other tables with patients in the low millions might take fifteen minutes for each table.

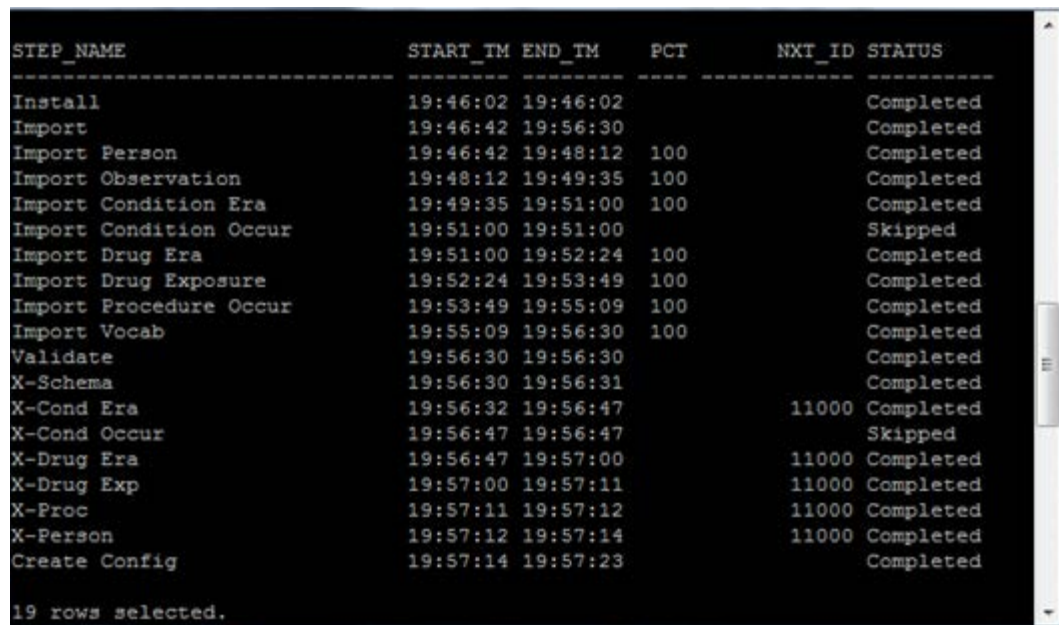
- If you are working on a network that limits your connection time, determine when your connection will time out so that you can reconnect and resume the data transfer.
- Oracle recommends checking the status of the data transfer at least every two to three hours to make sure that the process has not been interrupted.

To monitor the data transfer:

- 1 In SQL*Plus, connect to the Empirica Healthcare Analysis database using the destination account.
- 2 Query the ETL_STATUS table in the destination account:

```
select step_name, to_char(start_date, 'HH24:MI:SS') start_tm,
       to_char(end_date, 'HH24:MI:SS') end_tm, pct_complete pct, next_person_id
       nxt_id, status from etl_status order by start_date;
```

The status of each step appears. In the following example, all steps are completed. If you query the ETL_STATUS table while the data transfer is in progress, the STEP_NAME column lists only the jobs that have completed or started.



STEP_NAME	START_TM	END_TM	PCT	NXT_ID	STATUS
Install	19:46:02	19:46:02			Completed
Import	19:46:42	19:56:30			Completed
Import Person	19:46:42	19:48:12	100		Completed
Import Observation	19:48:12	19:49:35	100		Completed
Import Condition Era	19:49:35	19:51:00	100		Completed
Import Condition Occur	19:51:00	19:51:00			Skipped
Import Drug Era	19:51:00	19:52:24	100		Completed
Import Drug Exposure	19:52:24	19:53:49	100		Completed
Import Procedure Occur	19:53:49	19:55:09	100		Completed
Import Vocab	19:55:09	19:56:30	100		Completed
Validate	19:56:30	19:56:30			Completed
X-Schema	19:56:30	19:56:31			Completed
X-Cond Era	19:56:32	19:56:47		11000	Completed
X-Cond Occur	19:56:47	19:56:47			Skipped
X-Drug Era	19:56:47	19:57:00		11000	Completed
X-Drug Exp	19:57:00	19:57:11		11000	Completed
X-Proc	19:57:11	19:57:12		11000	Completed
X-Person	19:57:12	19:57:14		11000	Completed
Create Config	19:57:14	19:57:23			Completed

19 rows selected.

- 3 Review the STATUS column to check the progress of the data transfer.

What do I do if an issue occurs during the data transfer?

If a step fails or the data transfer is interrupted, the error appears in the SQL*Plus window. The reason for failure determines your next step. Next steps include:

- Resuming the data transfer.

For more information, see *Resuming the data transfer* (on page 36).

- Editing the etl.properties file, dropping the destination account, installing again, and starting the data transfer over again.

For more information, see *Installing and transferring data again* (on page 36).

Issue	Resolution
You see in the SQL*Plus window that the data transfer has been interrupted.	Reconnect to the network, and resume the data transfer.
A step, such as an import step, the validate step, a transformation step, or the Create Config step fails. A transformation step begins with X-, such as X-Schema.	<ol style="list-style-type: none"> Query the ETL_STATUS table to determine the last step that was processed. The last step that was processed has a status of InProgress. Review start_etl.log for errors or messages, and correct the errors. Edit the properties file and start over, or resume the data transfer: <ul style="list-style-type: none"> If you change the value of a property in the etl.properties file, you must drop the destination account, install the ETL utility again, and start the data transfer over again. If your fix does not involve changing the etl.properties file, resume the data transfer.
The log file says that a table creation issue has occurred.	<p>Specify the correct source tablespace.</p> <p>How to troubleshoot:</p> <ol style="list-style-type: none"> Check that the source_tablespaces property in the etl.properties file includes a comma-delimited list of the names of both tablespaces in the source account. Edit the properties file and start again. If you change the value of a property in the etl.properties file, you must drop the destination account, install the ETL utility again, and start the data transfer over again.
A step fails with an out-of-space error.	If the process failed because you ran out of tablespace for the destination account, fix the tablespace issue, and resume the data transfer.
The data transfer was successful, but some column names were renamed.	If you added columns to the OMOP Common Data Model, the column names might have collided with the columns added by the ETL utility. If a collision occurred, the existing source column names were renamed in the destination account. Columns are renamed to SRC_<original_name>. You do not need to take any corrective steps.

Resuming the data transfer

If a step in the data transfer is interrupted, you can resume the process. Resuming continues the ETL process after the last successfully completed step.

When you resume, the results are saved to a different log file, `resume_etl.log`, in the current directory. The `resume_etl.log` file is overwritten each time you resume.

- 1 In a command shell, set your current directory to the location where you unpacked the installation ZIP file, `Healthcare-CDM4_ETL_xxx.zip`, where `xxx` refers to the build number.

For example:

```
$ cd /u01/stage/cdm4etl
```

- 2 In the command shell, log in to the Empirica Healthcare Analysis database using the destination account, which was created during installation of the ETL utility. You provided the name of the account in the `etl.properties` file and supplied the password during installation.

For example:

```
$ sqlplus <account_name>@<connect_string>
```

You are prompted to type a password for the destination account.

- 3 Type a password, and press **Enter**.
- 4 Run the resume ETL script:

```
SQL>@resume_etl
```

Installing and transferring data again

If an issue occurs during data transfer and the fix involves changing the `etl.properties` file, you must install the ETL utility again, and then start the data transfer process again from the beginning.

Step 1: Install the ETL utility again.

For instructions, see *Editing the etl.properties file and installing the ETL utility again* (on page 28).

Step 2: Transfer data again.

- 1 In a command shell, set your current directory to the location where you unpacked the installation ZIP file, `Healthcare-CDM4_ETL_xxx.zip`, where `xxx` refers to the build number.

For example:

```
$ cd /u01/stage/cdm4etl
```

- 2 If you created one or more configurations in the Empirica Healthcare Analysis database during your last attempt at data transfer, provide different names for the configurations:
 - a Open the `etl.properties` file in a text editor, such as `vi`.
 - b Edit the properties as needed. For descriptions of each property, see *Setting installation and data transfer settings* (on page 16).

- 3 Execute the ETL script:

```
SQL>@start_etl
```

The start_etl.log file is overwritten in the current directory.

Determining whether the data transfer process was successful

How do I confirm that data was imported?

- 1 In SQL*Plus, using the destination account, run the following command to view the ETL_STATUS table:

```
select step_name, to_char(start_date, 'HH24:MI:SS') start_tm,  
       to_char(end_date, 'HH24:MI:SS') end_tm, pct_complete pct, next_person_id  
       nxt_id, status from <dest_account>.etl_status;
```

- 2 Make sure that one row exists for each of the following values in the STEP_NAME column:
 - Import.
 - Import Vocab.
 - Each of the seven data tables, with a value of **Import <table_name>**.

If any rows are missing, the import was not successful.

- 3 Make sure that:

- Each of the steps has one of the following values for STATUS:
 - **Completed** and a PCT value (for percent complete) of 100.
 - **Skipped**, for tables that were not loaded.

If the STATUS is Failed or the PCT value is less than 100, the import was not successful.

- The START_TM and END_TM columns have values.

If values are missing, the import was not successful.

How do I confirm that validation was successful?

- 1 In SQL*Plus, using the destination account, run the following command to view the ETL_STATUS table:

```
select step_name, to_char(start_date, 'HH24:MI:SS') start_tm,  
       to_char(end_date, 'HH24:MI:SS') end_tm, pct_complete pct, next_person_id  
       nxt_id, status from <dest_account>.etl_status;
```

- 2 Make sure that:

- The STEP_NAME column contains a Validate step.
- The step has a STATUS of Completed.
- A value exists in the END_TM column for the step.

If the Validate step is missing or has a STATUS of Failed, validation was not successful.

How do I confirm that data was transformed correctly?

- 1 In SQL*Plus, using the destination account, run the following command to view the ETL_STATUS table:

```
select step_name, to_char(start_date, 'HH24:MI:SS') start_tm,
       to_char(end_date, 'HH24:MI:SS') end_tm, pct_complete pct, next_person_id
       nxt_id, status from <dest_account>.etl_status;
```

- 2 Make sure that one row exists for each table. The name of each imported table is X-<table>, such as X-Schema, and appears in the STEP_NAME column.

Additionally:

- Each row has a STATUS of Completed or Skipped.
- The END_TM columns have values.

For tables that were not imported, the STATUS is Skipped. You can disregard these rows.

If a row is missing from the table, or a row has a STATUS of Failed, or an END_TM column is missing a value, data was not transformed correctly.

To understand how data is transformed, see *Transformation details* (on page 47).

How do I confirm that configurations were created?

- 1 In SQL*Plus, using the destination account, run the following command to view the ETL_STATUS table:

```
select step_name, to_char(start_date, 'HH24:MI:SS') start_tm,
       to_char(end_date, 'HH24:MI:SS') end_tm, pct_complete pct, next_person_id
       nxt_id, status from <dest_account>.etl_status;
```

- 2 Make sure that:

- A Create Config row exists.
- The step has a STATUS of Completed.
- A value exists in the END_TM column for the step.


If the Create Config row is missing or has a STATUS of Failed, the configurations were not created.

Importing and validating configurations

After data is transferred successfully, you must import and validate configurations in the Empirica Healthcare Analysis application so that users can begin accessing the imported data. After the data transfer, you have one or two configurations:

- If you imported the DRUG_ERA table, you have a Drug Era configuration.
- If you imported the DRUG_EXPOSURE table, you have a Drug Exposure configuration.

When you validate configurations in the Empirica Healthcare Analysis application, the validation process applies grants to the destination account.

- 1 Log in to the Empirica Healthcare Analysis application as a superuser, such as **admin**, or a user with Manage Configurations privileges.
- 2 Import the configurations that were created:
 - a Click **Settings**, and click **Manage Configurations**.
The Manage Configurations page appears.
 - b Click **Import Configurations**.
The Add/Import Configurations page appears, listing the names of accounts containing added or imported configurations.
 - c Click **Import Configurations from a Different Account**.
Fields appear on the page.
 - d Type the name of the destination account.
 - e (Optional) Chose a database group to which to add the configurations.
 - f Click **OK**.
The configurations in the data account are imported, and their values in the **Is Valid** column are **No**.
- 3 Validate the configurations:
 - a Click **Settings**, and click **Manage Configurations**.
The Manage Configurations page appears.
 - b Select the row menu for a configuration () , and select **Edit**.
The Modify Configuration page appears.
 - c Click **Validate Now**.
The validation process runs, and one of the following results occurs:
 - The configuration is valid and has no errors or warnings.
 - The configuration is valid with warnings.
 - The configuration is invalid with errors and possibly warnings.
 - d If the validation generates warnings, review the warnings. To ignore them, select **Ignore Warnings**.
If no errors were found and you ignore the warnings, the configuration is validated.

If both warnings and errors were found and you ignore the warnings, the page reappears with the same warning messages and error messages.

- e Address any issues found.

To understand the details of the configurations, see *Configuration details* (on page 55).

APPENDIX A

Exporting source data (Only for importing using files)

In this appendix

Exporting source data and making the data files accessible44

Exporting source data and making the data files accessible

Perform the following steps only if you plan to import data using files. These steps export the data in the source database and make the files accessible to the Empirica Healthcare Analysis database.

These steps create separate export files for the data and vocabulary accounts and include directions on how to modify commands if the vocabulary account is part of the data account.

Perform these steps before you set installation and data transfer settings.

Step 1. Create the export directory in the source database.

You can use the default DATA_PUMP_DIR directory or create a new directory. Oracle recommends creating a new directory.

- 1 Using SQL*Plus, log in to the source database as SYS.
- 2 Create the directory, pointing to a valid and accessible location on the server:


```
SQL>create or replace directory <export_directory> as '<server_path>';
```

where:

- *<export_directory>* is the name of the directory to which you are exporting data.
- *<server_path>* is the path to the server location that exported files will be written to.

Example:

```
SQL>create or replace directory ETL_SRC_DIR as '/u01/oradata/xxx';
```

Step 2. Give privileges to the source user accounts for the export directory.

- Use the following commands to give privileges to the user accounts in the source database that will be exported.

If the vocabulary account is the same as the data account, omit the second command.

```
SQL>GRANT READ, WRITE ON DIRECTORY <export_directory> TO
<data_schema_name>;

SQL>GRANT READ, WRITE ON DIRECTORY <export_directory> TO
<vocabulary_schema_name>;
```

where:

- *<export_directory>* is the name of the directory to which you are exporting data.
- *<data_schema_name>* is the name of the account for the data tables in the source database.
- *<vocabulary_schema_name>* is the name of the account for the vocabulary tables in the source database.

Example:

```
SQL>GRANT READ, WRITE ON DIRECTORY ETL_SRC_DIR TO CDM_SOURCE;

SQL>GRANT READ, WRITE ON DIRECTORY ETL_SRC_DIR TO CDM_VOCABULARY;
```

Step 3. Export the source data, either all of the tables or the tables of your choosing.

If you export all the tables, tables that are not needed are exported. For instance, you need only one condition table.

Export the tables along with their indexes and constraints. You do not need to export other objects in the account.

- To export all the data, use the following commands.

If the vocabulary account is the same as the data account, omit the second command.

You are prompted to provide a password for each account.

Commands for Oracle 11g:

```
$expdp <data_schema_name>@<connect_string> directory=<export_directory>
dumpfile=<dump_file_name>.dmp logfile=<log_file_name>
```

```
$expdp <vocabulary_schema_name>@<connect_string>
directory=<export_directory> dumpfile=<dump_file_name>.dmp
logfile=<log_file_name>
```

Commands for Oracle 12c:

```
$expdp <data_schema_name>@<connect_string>
dumpfile=<export_directory>:<dump_file_name>.dmp
logfile=<export_directory>:<log_file_name>
```

```
$expdp <vocabulary_schema_name>@<connect_string>
dumpfile=<export_directory>:<dump_file_name>.dmp
logfile=<export_directory>:<log_file_name>
```

where:

- *<data_schema_name>* is the name of the account for the data tables in the source database.
- *<vocabulary_schema_name>* is the name of the account for the vocabulary tables in the source database.
- *<connect_string>* is the connect string for the source database, one of the following:
 - A net service name that is specified in the tnsnames.ora file on the Empirica Healthcare Analysis database server and that indicates how to connect to the source server.
 - An EZCONNECT connection string in the following format: **hostname:port/service**
- *<export_directory>* is the name of the directory to which you are exporting data.
- *<dump_file_name>* is the name to use for the dump file that is created.
- *<log_file_name>* is the name to use for the log file that is created.

Example for Oracle 11g:

```
$expdp cdm_source@ORCL directory=ETL_SRC_DIR dumpfile=data_export.dmp
logfile=data_export.log
```

```
$expdp cdm_vocabulary@ORCL directory=ETL_SRC_DIR dumpfile=voc_export.dmp
logfile=voc_export.log
```

Example for Oracle 12g:

```
$expdp cdm_source@ORCL dumpfile=ETL_SRC_DIR:data_export.dmp
logfile=ETL_SRC_DIR:data_export.log
```

```
$expdp cdm_vocabulary@ORCL dumpfile=ETL_SRC_DIR:voc_export.dmp
logfile=ETL_SRC_DIR:voc_export.log
```

- To export only a set of specific tables:
 - a Decide upon the data tables to export. For a list of the options, see *Tables that are copied from the source database* (on page 4).
 - The following tables are required for all exports: PERSON, OBSERVATION_PERIOD
 - You must export at least one of the following tables: DRUG_ERA, DRUG_EXPOSURE
 - You must export exactly one of the following tables: CONDITION_ERA, CONDITION_OCCURRENCE
 - The following table is optional: PROCEDURE_OCCURRENCE
 - b If you have a vocabulary account, decide whether to export only the CONCEPT table.
 - c Add a space to the last letter of each of the previous commands, and add the following text:

```
TABLES=<table_list>
```

where:

- *<table_list>* is the comma-delimited list of tables to export from the source database.

Example:

```
$expdp cdm_source/cdm_source@ORCL directory=ETL_SRC_DIR
dumpfile=data_export.dmp logfile=data_export.log
TABLES=PERSON,OBSERVATION_PERIOD,DRUG_ERA,DRUG_EXPOSURE,CONDITION_ERA,PR
OCEDURE_OCCURRENCE
```

```
$expdp cdm_vocabulary/cdm_vocabulary@ORCL directory=ETL_SRC_DIR
dumpfile=voc_export.dmp logfile=voc_export.log TABLES=CONCEPT
```

Step 4. Move the export files you created in step 3 to the import directory location on the Empirica Healthcare Analysis database server.

You can move the files using a tool of your choice, such as an sFTP client.

Step 5. Create a database directory on the Empirica Healthcare Analysis database server machine.

- 1 Using SQL*Plus, log in to the Empirica Healthcare Analysis database as SYS.
- 2 Create a directory, pointing to a valid and accessible location on the server:

```
SQL>create or replace directory <import_directory> as '<server_path>';
```

where:

- *<import_directory>* is the name of the directory from which you are importing data.
- *<server_path>* is the path to the server location that will contain the data files you will import from.

Example:

```
SQL>create or replace directory ETL_SRC_DIR as '/u01/oradata/xxx';
```


APPENDIX B

Transformation details

In this appendix

Columns that are added when data is transformed	48
How added columns are populated	49
Configurations that are created.....	51
Views that are created	52
Tables that are created.....	53

Columns that are added when data is transformed

The following columns are added in the Empirica Healthcare Analysis database and populated during the data transformation step.

Table	Added column	Nulls allowed?	Based on
All Tables	PATIENT_ID	No	PERSON_ID
PERSON	GENDER_CD	No	GENDER_CONCEPT_ID
PERSON	RACE_CD	Yes	RACE_CONCEPT_ID
PERSON	ETHNICITY_CD	Yes	ETHNICITY_CONCEPT_ID
PERSON	ENROLLMENT_START_DATE	No	Observation Periods
PERSON	ENROLLMENT_END_DATE	No	Observation Periods
PERSON	AGE	Yes	ENROLLMENT_START_DATE, YEAR_OF_BIRTH
PERSON	AGE_GROUP	Yes	AGE
DRUG_ERA, DRUG_EXPOSURE	DRUG_TYPE_CD	Yes	DRUG_TYPE_CONCEPT_ID
DRUG_ERA	DURATION	No	DRUG_ERA_START_DATE DRUG_ERA_END_DATE
DRUG_EXPOSURE	RELEVANT_CONDITION_CD	Yes	RELEVANT_CONDITION_CONCEPT_ID
CONDITION_ERA, CONDITION_OCCURRENCE	CONDITION_TYPE_CD	Yes	CONDITION_TYPE_CONCEPT_ID
PROCEDURE_OCCURRENCE	PROCEDURE_TYPE_CD	Yes	PROCEDURE_TYPE_CONCEPT_ID
PROCEDURE_OCCURRENCE	RELEVANT_CONDITION_CD	Yes	RELEVANT_CONDITION_CONCEPT_ID

How added columns are populated

Column	How it is populated
PATIENT_ID	<p>PATIENT_ID is derived from PERSON_ID by converting to a VARCHAR2. The string is zero-filled to maintain sort order in the Empirica Healthcare Analysis application. A leading 1 is prepended to the value because the Empirica Healthcare Analysis application requires this format.</p> <p>The maximum number of allowed digits in PATIENT_ID is 10 and in PERSON_ID is 9.</p>
(Multiple columns used for coded fields)	<p><value>_CD fields, such as RACE_CD, are derived by selecting the CONCEPT_NAME from the CONCEPT table, based on the value of the corresponding <value>_concept_id field.</p>
ENROLLMENT_START_DATE ENROLLMENT_END_DATE	<p>The SELECTED_OBSERVATION_PERIOD view is created and contains one row for each person, with the OBSERVATION_START_DATE and OBSERVATION_END_DATE for the longest enrollment period (as determined by subtracting START_DATE from END_DATE). If more than one record exists with the longest duration, the earliest observation period is selected.</p>
AGE	<p>Age, in years, is either derived or populated:</p> <ul style="list-style-type: none"> • If compute_age in the etl.properties field is set to 'Y', this value is either derived by subtracting YEAR_OF_BIRTH from the year part of ENROLLMENT_START_DATE, or populated with values from the database field that you specify in age_column. • If compute_age is set to 'N' and you specify a column name in age_column in the etl.properties file, this field is populated with values from the database field that you specify. • If compute_age is set to 'N' and you do not specify a column name in age_column, this field is null.

Column	How it is populated
AGE_GROUP	<p>AGE_GROUP is derived from AGE when compute_age in the etl.properties file is set to 'Y'. Each patient has one of the following values:</p> <ul style="list-style-type: none">'00-04''05-14''15-17''18-24''25-34''35-44''45-64';'65+''UNK' (if AGE is null) <p>When compute_age is set to 'N', this field is null.</p>
DURATION	<p>Duration is derived by subtracting DRUG_ERA_START_DATE from DRUG_ERA_END_DATE + 1.</p>

Configurations that are created

The settings in the `etl.properties` file determine the configurations that are created:

- If you import the `DRUG_ERA` table, the CDM Era configuration is created.
- If you import the `DRUG_EXPOSURE` table, the CDM Exposure configuration is created.

Views that are created

For drug, condition, and procedure tables, views are created with the following naming convention:

<table_name>_VIEW

Each view joins the base table with CONCEPT to translate the drug, condition, and procedure concept_ids into the associated codes and names.

- Coded values appear in the views in one of the following additional columns:
 - DRUG_CD, for drug concept_ids.
 - CONDITION_CD, for condition concept_ids.
 - PROCEDURE_CD, for procedure concept_ids.
- Text descriptions appear in the views in one of the following additional columns:
 - DRUG_NAME, for drug concept_ids.
 - CONDITION_NAME, for condition concept_ids.
 - PROCEDURE_NAME, for procedure concept_ids.

Tables that are created

The following tables are created and populated in the destination account:

- DM_CONFIGS
- SOURCE_INFO_ERA
- SOURCE_INFO_EXP
- DRILLDOWN_MAP_ERA
- DRILLDOWN_MAP_EXP
- CFG_ERA
- CFG_EXP

For details, see *Configuration details* (on page 55).

APPENDIX C

Configuration details

In this appendix

- About this appendix56
- Indexes.....57
- DM_CONFIGS.....58
- SOURCE_INFO_ERA.....59
- SOURCE_INFO_EXP.....60
- DRILLDOWN_MAP_ERA61
- DRILLDOWN_MAP_EXP.....63
- CFG_ERA65
- CFG_EXP70

About this appendix

This appendix includes the contents of the tables that are created in the Empirica Healthcare Analysis database.

Indexes

During data transfer, indexes are created on the following tables.

Table	Columns on which indexes are created
PERSON	PATIENT_ID
DRUG_ERA	PATIENT_ID DRUG_ERA_START_DATE DRUG_CONCEPT_ID
DRUG_EXPOSURE	PATIENT_ID DRUG_EXPOSURE_START_DATE DRUG_CONCEPT_ID
CONDITION_ERA	PATIENT_ID CONDITION_ERA_START_DATE CONDITION_CONCEPT_ID
CONDITION_OCCURRENCE	PATIENT_ID CONDITION_START_DATE CONDITION_CONCEPT_ID
PROCEDURE_OCCURRENCE	PATIENT_ID PROCEDURE_DATE PROCEDURE_CONCEPT_ID

The transformation process creates additional single-column indexes, one for each configuration variable, based on a column that exists in the base tables.

DM_CONFIGS

The DM_CONFIGS table is created in the destination account.

This table contains one row for each configuration that is created.

Config name	Config table	Description	Drilldown map	Source info table
config_name_era	CFG_ERA	CDM using DRUG_ERA	DRILLDOWN_MAP_ERA	SOURCE_INFO_ERA
config_name_exposure	CFG_EXP	CDM using DRUG_EXPOSURE	DRILLDOWN_MAP_EXP	SOURCE_INFO_EXP

SOURCE_INFO_ERA

The SOURCE_INFO_ERA table is created in the destination account.

This table has one row and identifies the table sources for the CDM Era configuration.

Description	Source	Demog table	Drug table	Event table	Procedure table
CDM Era	<Destination Schema>	PERSON	DRUG_ERA_VIEW	One of the following: CONDITION_ERA_VIEW, if you imported the CONDITION_ERA table. CONDITION_OCCURRENCE_VIEW, if you imported the CONDITION_OCCURRENCE table.	PROCEDURE_OCCURRENCE_VIEW

SOURCE_INFO_EXP

The SOURCE_INFO_EXP table is created in the destination account.

This table has one row and identifies the table sources for the CDM Exposure configuration.

Description	Source	Demog table	Drug table	Event table	Procedure table
CDM Exposure	<Destination Schema>	PERSON	DRUG_EXPOSURE_VIEW	One of the following: CONDITION_ERA_VIEW, if you imported the CONDITION_ERA table. CONDITION_OCCURRENCE_VIEW, if you imported the CONDITION_OCCURRENCE table.	PROCEDURE_OCCURRENCE_VIEW

DRILLDOWN_MAP_ERA

The DRILLDOWN_MAP_ERA table is created in the destination account.

This table identifies the columns to display for each of the types of data in the Empirica Healthcare Analysis application. You can change this information in the Empirica Healthcare Analysis application. For more information, see the *User Guide*.

The PROCEDURES row has null values for Table, Display Columns, and Order By if you did not load procedures.

Info type	Table	Display columns	Order by columns	Section title	CASE_COL
LIST	PERSON	PATIENT_ID AS "Patient Id"			PATIENT_I D
		GENDER_CD AS "Gender"			
		AGE as "Age"			
		ENROLLMENT_STA RT_DATE			
		ENROLLMENT_EN D_DATE			
CASE	PERSON	PATIENT_ID AS "Patient Id"		Patient Information	PATIENT_I D
		GENDER_CD AS "Gender"			
		RACE_CD AS "Race"			
		AGE as "Age"			
		ENROLLMENT_STA RT_DATE			
		ENROLLMENT_EN D_DATE			

Info type	Table	Display columns	Order by columns	Section title	CASE_COL
EVENT	If you imported the CONDITION_ERA table: CONDITION_ERA_VIEW If you imported the CONDITION_OCCURRENCE table: CONDITION_OCCURRENCE_VIEW	PATIENT_ID AS "Patient Id" If you imported the CONDITION_ERA table: CONDITION_ERA_START_DATE as "Condition Date" If you imported the CONDITION_OCCURRENCE table: CONDITION_CD AS "Condition Code" CONDITION_NAME AS "Condition Name"	If you imported the RA table: RA_START_DATE If you imported the OCCURRENCE table: CONDITION_START_DATE CONDITION_S TART_DATE	Diagnoses	PATIENT_ID
DRUG	DRUG_ERA_VIEW	PATIENT_ID AS "Patient Id" DRUG_ERA_START_DATE as "Start Date" DRUG_CD as "Drug Code" DRUG_NAME AS "Drug Name" DURATION	DRUG_ERA_START_DATE	Drugs/Prescriptions	PATIENT_ID
PROCEDURES	PROCEDURE_OCCURRENCE_VIEW	PATIENT_ID AS "Patient Id" PROCEDURE_DATE PROCEDURE_CD PROCEDURE_NAME	PROCEDURE_DATE	Procedures	PATIENT_ID

DRILLDOWN_MAP_EXP

The DRILLDOWN_MAP_EXP table is created in the destination account.

This table identifies the columns to display for each of the types of data in the Empirica Healthcare Analysis application. You can change this information in the Empirica Healthcare Analysis application. For more information, see the *User Guide*.

The PROCEDURES row has null values for Table, Display Columns, and Order By if you did not load procedures.

Info type	Table	Display columns	Order by columns	Section title	CASE_COL
LIST	PERSON	PATIENT_ID AS "Patient Id"			PATIENT_I D
		GENDER_CD AS "Gender"			
		AGE as "Age"			
		ENROLLMENT_STA RT_DATE			
		ENROLLMENT_EN D_DATE			
CASE	PERSON	PATIENT_ID AS "Patient Id"		Patient Information	PATIENT_I D
		GENDER_CD AS "Gender"			
		RACE_CD AS "Race"			
		AGE as "Age"			
		ENROLLMENT_STA RT_DATE			
		ENROLLMENT_EN D_DATE			

Info type	Table	Display columns	Order by columns	Section title	CASE_COL
EVENT	<p>If you imported the CONDITION_ERA table: CONDITION_ERA_VIEW</p> <p>If you imported the CONDITION_OCCURRENCE table: CONDITION_OCCURRENCE_VIEW</p>	<p>PATIENT_ID AS "Patient Id"</p> <p>If you imported the CONDITION_ERA table: CONDITION_ERA_START_DATE as "Condition Date"</p> <p>If you imported the CONDITION_OCCURRENCE table: CONDITION_CD AS "Condition Code"</p> <p>CONDITION_NAME AS "Condition Name"</p>	<p>If you imported the RA table: RA_START_DATE</p> <p>If you imported the CONDITION_OCCURRENCE table: CONDITION_START_DATE</p>	Diagnoses	PATIENT_ID
DRUG	DRUG_EXPOSURE_VIEW	<p>PATIENT_ID AS "Patient Id"</p> <p>DRUG_EXPOSURE_START_DATE as "Start Date"</p> <p>DRUG_CD as "Drug Code"</p> <p>DRUG_NAME AS "Drug Name"</p> <p>DAYS_SUPPLY</p>	DRUG_EXPOSURE_START_DATE	Drugs/Prescriptions	PATIENT_ID
PROCEDURES	PROCEDURE_OCCURRENCE_VIEW	<p>PATIENT_ID AS "Patient Id"</p> <p>PROCEDURE_DATE</p> <p>PROCEDURE_CD</p> <p>PROCEDURE_NAME</p>	PROCEDURE_DATE	Procedures	PATIENT_ID

CFG_ERA

The CFG_ERA table is created in the destination account.

The table contains the following rows in the following tables. The Patient Id table row indicates whether a Patient Id table is required and created for the variable. Patient Id tables and Index tables are created as needed because the Empirica Healthcare Analysis application requires them.

Rows created based on the PERSON table

The following rows appear in this table for all imports.

Variable	Description	Table	Column	Variable type	Selection type	Patient Id table
Patient Id	Unique identifier of individual	PERSON	PATIENT_ID	Patient ID	Free Text	
Gender	Patient gender	PERSON	GENDER_CD	Gender Stratification	Distinct Value List	Yes
Race	Patient race	PERSON	RACE_CD	Other	Distinct Value List	
Ethnicity	Patient ethnicity	PERSON	ETHNICITY_CD	Other	Distinct Value List	
Enrollment Start	Enrollment start date	PERSON	ENROLLMENT_START_DATE	Other (Enrollment Start Date)	Continuous	
Enrollment End	Enrollment end date	PERSON	ENROLLMENT_END_DATE	Other (Enrollment End Date)	Continuous	
Year of birth	Year of birth	PERSON	YEAR_OF_BIRTH	Other	Continuous	
Month of birth	Month of birth	PERSON	MONTH_OF_BIRTH	Other	Continuous	
Day of Birth	Day of birth	PERSON	DAY_OF_BIRTH	Other	Continuous	
Age	Age at enrollment start	PERSON	AGE	Age (Age at enroll-start)	Continuous	
Age Group	Patient age group	PERSON	AGE_GROUP	Other Stratification	Distinct Value List	Yes

Notes:

- The Age Group variable has a report_id_table only when compute_age in the etl.properties file is set to 'Y'.
- For variables of type Drug, Event, and Procedure, the prefix attribute is **D**, **E**, and **P**, respectively.

Rows created based on the DRUG_ERA table

If you imported the DRUG_ERA table, the following rows appear in this table.

Variable	Description	Table	Column	Variable type	Selection type	Patient Id table
Patient Id (Drug)	Unique identifier of individual	DRUG_ERA_VIEW	PATIENT_ID	Patient ID	Free Text	
Drug Start Date	Drug start date	DRUG_ERA_VIEW	DRUG_ERA_START_DATE	Record Date	Continuous	
Drug End Date	Drug end date	DRUG_ERA_VIEW	DRUG_ERA_END_DATE	Other	Continuous	
Drug Code	Drug code	DRUG_ERA_VIEW	DRUG_CD	Other	Distinct Value List	
Drug Name	Drug name	DRUG_ERA_VIEW	DRUG_NAME	Drug	Distinct Value List	Yes
Drug Label	Drug name label	DRUG_ERA_VIEW	DRUG_NAME	Other (Drug Label)	Distinct Value List	
Drug Exposure Type	Drug exposure type	DRUG_ERA_VIEW	DRUG_TYPE_CD	Other	Distinct Value List	
Duration	Length of drug era (days)	DRUG_ERA_VIEW	DURATION	Record Duration	Continuous	
Drug Exposure Count	Drug exposure count	DRUG_ERA_VIEW	DRUG_EXPOSURE_COUNT	Other	Continuous	

Rows created based on the CONDITION_ERA table

If you imported the CONDITION_ERA table, the following rows appear in this table.

Variable	Description	Table	Column	Variable type	Selection type	Patient Id table
Patient Id (Condition)	Unique identifier of individual	CONDITION_ERA_VIEW	PATIENT_ID	Patient ID	Free Text	
Condition Start Date	Condition start date	CONDITION_ERA_VIEW	CONDITION_ERA_START_DATE	Record Date	Continuous	
Condition End Date	Condition end date	CONDITION_ERA_VIEW	CONDITION_ERA_END_DATE	Other	Continuous	
Condition Code	Condition code	CONDITION_ERA_VIEW	CONDITION_CODE	Other	Distinct Value List	
Condition Name	Condition name	CONDITION_ERA_VIEW	CONDITION_NAME	Event	Distinct Value List	Yes
Condition Label	Condition name label	CONDITION_ERA_VIEW	CONDITION_NAME	Other (Event Label)	Distinct Value List	
Condition Type	Condition type	CONDITION_ERA_VIEW	CONDITION_TYPE_CD	Other	Distinct Value List	
Occurrence Count	Condition occurrence count	CONDITION_ERA_VIEW	CONDITION_OCCURRENCE_COUNT	Other	Continuous	

Rows created based on the CONDITION_OCCURRENCE table

If you imported the CONDITION_OCCURRENCE table, the following rows appear in this table.

Variable	Description	Table	Column	Variable type	Selection type	Patient Id table
Patient Id (Condition)	Unique identifier of individual	CONDITION_OCCURRENCE_VIEW	PATIENT_ID	Patient ID	Free Text	
Condition Start Date	Condition start date	CONDITION_OCCURRENCE_VIEW	CONDITION_START_DATE	Record Date	Continuous	
Condition End Date	Condition end date	CONDITION_OCCURRENCE_VIEW	CONDITION_END_DATE	Other	Continuous	

Variable	Description	Table	Column	Variable type	Selection type	Patient Id table
Condition Code	Condition code	CONDITION_OCCURRENCE_VIEW	CONDITION_CD	Other	Distinct Value List	
Condition Name	Condition name	CONDITION_OCCURRENCE_VIEW	CONDITION_NAME	Event	Distinct Value List	Yes
Condition Label	Condition name label	CONDITION_OCCURRENCE_VIEW	CONDITION_NAME	Other (Event Label)	Distinct Value List	
Condition Type	Condition type	CONDITION_OCCURRENCE_VIEW	CONDITION_TYPE_CD	Other	Distinct Value List	
Stop Reason (Condition)	Stop reason	CONDITION_OCCURRENCE_VIEW	STOP_REASON	Other	Free Text	

Rows created based on the PROCEDURE_OCCURRENCE table

If you imported the PROCEDURE_OCCURRENCE table, the following rows appear in this table.

Name	Description	Table	Column	Variable type	Selection type	Patient Id Table
Patient Id (Procedure)	Unique identifier of individual	PROCEDURE_OCCURRENCE_VIEW	PATIENT_ID	Patient ID	Free Text	
Procedure Date	Procedure date	PROCEDURE_OCCURRENCE_VIEW	PROCEDURE_DATE	Record Date	Continuous	
Procedure Code	Procedure code	PROCEDURE_OCCURRENCE_VIEW	PROCEDURE_CD	Other	Distinct Value List	
Procedure Name	Procedure name	PROCEDURE_OCCURRENCE_VIEW	PROCEDURE_NAME	Event (Procedure)	Distinct Value List	Yes
Procedure Label	Procedure name label	PROCEDURE_OCCURRENCE_VIEW	PROCEDURE_NAME	Other (Proc Label)	Distinct Value List	
Procedure Type	Procedure type	PROCEDURE_OCCURRENCE_VIEW	PROCEDURE_TYPE_CD	Other	Distinct Value List	

Name	Description	Table	Column	Variable type	Selection type	Patient Id Table
Relevant Condition Code (Procedure)	Relevant condition code	PROCEDURE_OCCURRENCE_VIEW	RELEVANT_CONDITION_CD	Other	Distinct Value List	

CFG_EXP

The CFG_EXP table is created in the destination account.

The table contains the following rows in the following tables. The Patient Id table row indicates whether a Patient Id table is required and created for the variable. Patient Id tables and Index tables are created as needed because the Empirica Healthcare Analysis application requires them.

Rows created based on the PERSON table

The following rows appear in this table for all imports.

Variable	Description	Table	Column	Variable type	Selection type	Patient Id table
Patient Id	Unique identifier of individual	PERSON	PATIENT_ID	Patient ID	Free Text	
Gender	Patient gender	PERSON	GENDER_CD	Gender Stratification	Distinct Value List	Yes
Race	Patient race	PERSON	RACE_CD	Other	Distinct Value List	
Ethnicity	Patient ethnicity	PERSON	ETHNICITY_CD	Other	Distinct Value List	
Enrollment Start	Enrollment start date	PERSON	ENROLLMENT_START_DATE	Other (Enrollment Start Date)	Continuous	
Enrollment End	Enrollment end date	PERSON	ENROLLMENT_END_DATE	Other (Enrollment End Date)	Continuous	
Year of birth	Year of birth	PERSON	YEAR_OF_BIRTH	Other	Continuous	
Month of birth	Month of birth	PERSON	MONTH_OF_BIRTH	Other	Continuous	
Day of Birth	Day of birth	PERSON	DAY_OF_BIRTH	Other	Continuous	
Age	Age at enrollment start	PERSON	AGE	Age (Age at enroll-start)	Continuous	
Age Group	Patient age group	PERSON	AGE_GROUP	Other Stratification	Distinct Value List	Yes

Notes:

- The Age Group variable has a report_id_table only when compute_age in the etl.properties file is set to 'Y'.
- For variables of type Drug, Event, and Procedure, the prefix attribute is **D**, **E**, and **P**, respectively.

Rows created based on the DRUG_EXPOSURE table

The following rows appear in this table if you imported the DRUG_EXPOSURE table.

Variable	Description	Table	Column	Variable type	Selection type	Patient Id table
Patient Id (Drug)	Unique identifier of individual	DRUG_EXPOSURE_VIEW	PATIENT_ID	Patient ID	Free Text	
Drug Start Date	Drug start date	DRUG_EXPOSURE_VIEW	DRUG_EXPOSURE_START_DATE	Record Date	Continuous	
Drug End Date	Drug end date	DRUG_EXPOSURE_VIEW	DRUG_EXPOSURE_END_DATE	Other	Continuous	
Drug Code	Drug code	DRUG_EXPOSURE_VIEW	DRUG_CD	Other	Distinct Value List	
Drug Name	Drug name	DRUG_EXPOSURE_VIEW	DRUG_NAME	Drug	Distinct Value List	Yes
Drug Label	Drug name label	DRUG_EXPOSURE_VIEW	DRUG_NAME	Other (Drug Label)	Distinct Value List	
Drug Exposure Type	Drug exposure type	DRUG_EXPOSURE_VIEW	DRUG_TYPE_CD	Other	Distinct Value List	
Days Supply	Days supply	DRUG_EXPOSURE_VIEW	DAYS_SUPPLY	Record Duration	Continuous	
Stop Reason (Drug)	Stop reason	DRUG_EXPOSURE_VIEW	STOP_REASON	Other	Free Text	
Refills	Refill count	DRUG_EXPOSURE_VIEW	REFILLS	Other	Continuous	
Quantity	Quantity	DRUG_EXPOSURE_VIEW	QUANTITY	Other	Continuous	
SIG	SIG	DRUG_EXPOSURE_VIEW	SIG	Other	Free Text	
Relevant Condition Code	Relevant condition code	DRUG_EXPOSURE_VIEW	RELEVANT_CONDITION_CD	Other	Distinct Value List	

Rows created based on the **CONDITION_ERA** table

If you imported the **CONDITION_ERA** table, the following rows appear in this table.

Variable	Description	Table	Column	Variable type	Selection type	Patient Id table
Patient Id (Condition)	Unique identifier of individual	CONDITION_ERA_VIEW	PATIENT_ID	Patient ID	Free Text	
Condition Start Date	Condition start date	CONDITION_ERA_VIEW	CONDITION_ERA_START_DATE	Record Date	Continuous	
Condition End Date	Condition end date	CONDITION_ERA_VIEW	CONDITION_ERA_END_DATE	Other	Continuous	
Condition Code	Condition code	CONDITION_ERA_VIEW	CONDITION_CD	Other	Distinct Value List	
Condition Name	Condition name	CONDITION_ERA_VIEW	CONDITION_NAME	Event	Distinct Value List	Yes
Condition Label	Condition name label	CONDITION_ERA_VIEW	CONDITION_NAME	Other (Event Label)	Distinct Value List	
Condition Type	Condition type	CONDITION_ERA_VIEW	CONDITION_TYPE_CD	Other	Distinct Value List	
Occurrence Count	Condition occurrence count	CONDITION_ERA_VIEW	CONDITION_OCCURRENCE_COUNT	Other	Continuous	

Rows created based on the **CONDITION_OCCURRENCE** table

If you imported the **CONDITION_OCCURRENCE** table, the following rows appear in this table.

Variable	Description	Table	Column	Variable type	Selection type	Patient Id table
Patient Id (Condition)	Unique identifier of individual	CONDITION_OCCURRENCE_VIEW	PATIENT_ID	Patient ID	Free Text	
Condition Start Date	Condition start date	CONDITION_OCCURRENCE_VIEW	CONDITION_START_DATE	Record Date	Continuous	
Condition End Date	Condition end date	CONDITION_OCCURRENCE_VIEW	CONDITION_END_DATE	Other	Continuous	

Variable	Description	Table	Column	Variable type	Selection type	Patient Id table
Condition Code	Condition code	CONDITION_OCCURRENCE_VIEW	CONDITION_CD	Other	Distinct Value List	
Condition Name	Condition name	CONDITION_OCCURRENCE_VIEW	CONDITION_NAME	Event	Distinct Value List	Yes
Condition Label	Condition name label	CONDITION_OCCURRENCE_VIEW	CONDITION_NAME	Other (Event Label)	Distinct Value List	
Condition Type	Condition type	CONDITION_OCCURRENCE_VIEW	CONDITION_TYPE_CD	Other	Distinct Value List	
Stop Reason (Condition)	Stop reason	CONDITION_OCCURRENCE_VIEW	STOP_REASON	Other	Free Text	

Rows created based on the PROCEDURE_OCCURRENCE table

If you imported the PROCEDURE_OCCURRENCE table, the following rows appear in this table.

Name	Description	Table	Column	Variable type	Selection type	Patient Id Table
Patient Id (Procedure)	Unique identifier of individual	PROCEDURE_OCCURRENCE_VIEW	PATIENT_ID	Patient ID	Free Text	
Procedure Date	Procedure date	PROCEDURE_OCCURRENCE_VIEW	PROCEDURE_DATE	Record Date	Continuous	
Procedure Code	Procedure code	PROCEDURE_OCCURRENCE_VIEW	PROCEDURE_CD	Other	Distinct Value List	
Procedure Name	Procedure name	PROCEDURE_OCCURRENCE_VIEW	PROCEDURE_NAME	Event (Procedure)	Distinct Value List	Yes
Procedure Label	Procedure name label	PROCEDURE_OCCURRENCE_VIEW	PROCEDURE_NAME	Other (Proc Label)	Distinct Value List	
Procedure Type	Procedure type	PROCEDURE_OCCURRENCE_VIEW	PROCEDURE_TYPE_CD	Other	Distinct Value List	

Name	Description	Table	Column	Variable type	Selection type	Patient Id Table
Relevant Condition Code (Procedure)	Relevant condition code	PROCEDUR E_OCCURRE NCE_VIEW	RELEVANT_C ONDITION_C D	Other	Distinct Value List	