# Oracle® AI Database

## Database Sample Schemas

26ai
G43102-01
October 2025

Oracle AI Database Database Sample Schemas, 26ai

G43102-01



Primary Author: Richard Butner

Contributing Authors: Apoorva Srinivas, Lavanya Jayapalan, Amith R. Kumar, Tulika Das, Roza Leyderman, David Austin , Christian Bauwens, Vimmika Dinesh, Mark Drake, Nancy Greenberg, Deepti Kamal, Diana Lorentz, Nagavalli Pataballa

# Contents

## Index

# List of Tables

# About This Content

This guide is a primary source of information about the sample database schemas that are used for examples in Oracle AI Database documentation.

### Audience

This guide is intended for all users of Oracle AI Database.

### 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.

### 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.

### Conventions

The following text conventions are used in this document.

| Convention | Meaning |
|---|---|
| **boldface** | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary. |
| *italic* | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values. |
| `monospace` | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

1

# Introduction to Sample Schemas

The sample database schemas can be used for product documentation, courseware, software development, and application demos.

## About the Sample Schemas

The sample database schemas provide a common platform for examples in each release of Oracle AI Database. The sample schemas are a set of interlinked database schemas. This set provides the following approaches to complexity:

- Schema Human Resources (`hr`) is useful for introducing basic topics. An extension to this schema supports Oracle Internet Directory demos.
- Schema Customer Orders (`co`) is a modern schema useful for demos of e-commerce transactions. It allows the storage of semi-structured data using JSON.
- Schema Sales History (`sh`) is designed to allow for demos with large amounts of data. An extension to this schema provides support for advanced analytic processing.

The following schemas are no longer updated, but are still available:

- Schema Order Entry (`oe`) is useful for dealing with matters of intermediate complexity. Many data types are available in this schema, including nonscalar data types.
- Schema Online Catalog (`oc`) is a collection of object-relational database objects built inside the `oe` schema.
- Schema Product Media (`pm`) is dedicated to print media data types.

> **Note**
>
> The Business Intelligence (`bi`), Information Exchange (`ix`), and Shipping (`qs`) schemas are no longer available.

## Design Principles for Sample Schemas

The sample database schemas were created with the following design principles in mind:

- **Simplicity and ease of use**. The `hr` schema is intentionally simple. It provides a graduated path from simple to intermediate levels of database use.
- **Relevance for typical users**. The base schemas and their extensions bring to the foreground the functionality that customers typically use. Only the most commonly used database objects are built automatically in the schemas. The entire set of schemas provides a foundation upon which you can expand to illustrate additional functionality.
- **Extensibility**. The sample schemas provide a logical and physical foundation for adding objects to demonstrate functionality beyond the fundamental scope.

- **Relevance**. The sample schemas are designed to be applicable to e-business and other significant industry trends (for example, XML). When this goal conflicts with the goal of simplicity, schema extensions are used to showcase the trends in focus.

# Customer Benefits of Sample Schemas

Benefits provided by the sample schemas include the following:

- **Continuity of context**. When encountering the same set of tables everywhere, users, students, and developers can spend less time becoming familiar with the schema and more time understanding or explaining the technical concepts.
- **Usability**. Customers can use these schemas in the seed database to run examples that are shown in Oracle AI Database documentation and training materials. This first-hand access to examples facilitates both conceptual understanding and application development.
- **Quality**. Through central maintenance and testing of both the creation scripts that build the sample schemas and the examples that run against the schemas, the quality of Oracle AI Database documentation and training materials is enhanced.

# Overview of the Sample Schemas

The Oracle AI Database sample schemas are based on a fictitious sample company that sells goods through various channels. The company operates worldwide to fill orders for products. It has several divisions, each of which is represented by a sample database schema.

## HR Sample Schema

The Human Resources division tracks information about the company employees and facilities. In the Human Resource (`hr`) records, each employee has an identification number, e-mail address, job identification code, salary, and manager. Some employees earn commissions in addition to their salary.

The company also tracks information about jobs within the organization. Each job has an identification code, job title, and a minimum and maximum salary range for the job. Some employees have been with the company for a long time and have held different positions within the company. When an employee resigns, the duration the employee was working, the job identification number, and the department are recorded.

The sample company is regionally diverse, so it tracks the locations of its warehouses and departments. Each employee is assigned to a department, and each department is identified either by a unique department number or a short name. Each department is associated with one location, and each location has a full address that includes the street name, postal code, city, state or province, and the country code.

In places where the departments and warehouses are located, the company records details such as the country name, currency symbol, currency name, and the region where the country is located geographically.

## CO Sample Schema

The Customer Orders division tracks customer, product, store, and order data. The Customer Orders (`co`) schema records the details of transactions made by a retail application.

The `co` schema highlights features such as JSON support.

The company sells a variety of products, which are maintained in the `products` table. Each product has a unique identification number, name, price, details stored in a JSON object and product image details.

The orders placed by the customer are tracked in the `orders` table using the order identification number, date and time when the order was placed, customer details, order status and the store information.

The details of the products in a particular order are also tracked in the `order_items` table using the order identification number. Details of the products, price at the time of purchase, quantity and shipment are recorded.

The information of a customer placing an order is tracked in the `customers` table. Each customer has an identification number, name, and email address that is used for communication of the orders.

The customers can purchase the products in stores or online through the company's website. The information for all of the stores and their corresponding physical and virtual addresses is tracked in the `stores` table. The store information is also recorded in the order details.

The shipment details of the orders placed such as the delivery address, customer details, store information and the shipment status are stored in the `shipments` table.

An `inventory` table stores the details of each product such as the quantity available at each store.

## SH Sample Schema

The Sales division tracks business statistics to facilitate business decisions. The sample company does a high volume of business, so it runs business statistics reports to aid in decision making. Many of these reports are time-based and nonvolatile. That is, they analyze past data trends. The company loads data into its data warehouse regularly to gather statistics for these reports. These reports include annual, quarterly, monthly, and weekly sales figures by product. These reports are stored by using the schema Sales History (`sh`).

The company also runs reports on distribution channels through which its sales are delivered. When the company runs special promotions on its products, it analyzes the impact of the promotions on sales. It also analyzes sales by geographical area.

## OE Sample Schema

> **Note**
>
> The `oe` schema is no longer updated, but is still available.

The company sells several products, such as computer hardware and software, music, clothing, and tools. The company maintains information about these products, such as product identification numbers, the category into which the product falls, order entry (`oe`), the weight group (for shipping purposes), the warranty period if applicable, the supplier, the availability status of the product, a list price, a minimum price at which a product will be sold, and a URL address for manufacturer information. Inventory information is also recorded for all products, including the warehouse where the product is available and the quantity on hand. Because products are sold worldwide, the company maintains the names of the products and their descriptions in several languages.

The company maintains warehouses in several locations to fulfill customer needs. Each warehouse has a warehouse identification number, name, facility description, and location identification number.

Customer information is also tracked. Each customer has an identification number. Customer records include customer name, street name, city or province, country, phone numbers (up to five phone numbers for each customer), and postal code. Some customers place orders through the Internet, so e-mail addresses are also recorded. Because of language differences among customers, the company records the native language and territory of each customer.

The company places a credit limit on its customers, to limit the amount of products they can purchase at one time. Some customers have an account manager, and this information is also recorded.

When a customer places an order, the company tracks the date of the order, how the order was placed, the current status of the order, shipping mode, total amount of the order, and the sales representative who helped place the order. The sales representative may or may not be the same person as the account manager for a customer. If an order is placed over the Internet, no sales representative is recorded. In addition to order information, the company also tracks the number of items ordered, the unit price, and the products ordered.

The `oe` schema also contains XML purchase-order documents. You can access these documents by using SQL to query the `purchaseorder` table, or by querying the public views `RESOURCE_VIEW` and `PATH_VIEW`.

## OC Sample Schema

> **Note**
>
> The `oe` schema is no longer updated, but is still available.

The Online Catalog (`oc`) subschema of database schema `oe` addresses an online catalog merchandising scenario. The same customers and products are used in `oc` as in schema `oe` proper, but subschema `oc` organizes the products into a hierarchy of parent categories and subcategories. This hierarchy corresponds to the arrangement on an e-commerce portal site, where users navigate to specific products by drilling down through increasingly specialized categories of products.

## PM Sample Schema

> **Note**
>
> The `pm` schema is no longer updated, but is still available.

The company stores print information about its products in a database. The Product Media (`pm`) schema is used to store such information. Examples of such information are:

- Press release texts
- Print media advertisements
- Other promotional texts and translations

2

# Installation of the Sample Schemas

To use the sample schemas, download the installable `.zip` file from the GitHub repository.

## Installing the Sample Schemas

You can install the sample schemas independently.

Each sample schema comes with its own set of scripts, including an installation script. The following steps describe how to install the `hr` schema, but you can complete similar steps to install any of the sample schemas.

> **Note**
>
> - You use a client program such as SQL*Plus or SQLcl to install the schemas. You can also use SQL Developer, Visual Studio Code with the Oracle SQL Developer extension, or any other client program that supports running SQL scripts and connecting to the database as a privileged user with rights to create and drop other users.
> - To install or uninstall the `sh` schema, you must use SQLcl, SQL Developer, or Visual Studio Code with the Oracle SQL Developer extension. You cannot use SQL*Plus to install the `sh` schema.
> - When you install any of the sample schemas, you are prompted to drop (remove) any previously installed schema with the same name, and the default answer is `yes`. If you accept the default, the previous schema is dropped and a fresh schema is installed.
> - Do not use the sample schemas for your personal or business data and applications. The sample schemas are meant to be used for demonstration purposes only.

1. To find the latest version of the sample schemas installation scripts, go to the following GitHub location :

   https://github.com/oracle/db-sample-schemas/releases/latest

   Previous versions of the sample schemas are available at the following location:

   https://github.com/oracle-samples/db-sample-schemas/releases
2. Clone the GitHub repository, or download the `.zip` file from GitHub and then extract the files.
3. Navigate to the folder that contains the schema that you want to install. In this case, navigate to the `human_resources` folder.
   (The `co` schema is in the `customer_orders` folder, and the `sh` schema is in the `sales_history` folder.)

4. From the client program such as SQLcl or SQL*Plus, connect to the database as a privileged user with rights to create and drop other users. For example, connect as `SYSTEM` or `ADMIN`. To install the `sh` schema, you must use SQLcl or SQL Developer.
   For more information on connecting to a database by using SQLcl, see Connecting to a Database.

   For more information on connecting to a database by using SQL*Plus, see Starting SQL*Plus.
5. Run the `*_install.sql` installation script. In this case, run `hr_install.sql` to create the `hr` schema.
   - Enter a secure password for the schema.
   - Enter a tablespace to use as the default for the schema. If you do not enter a tablespace, the default database tablespace is used.
   - Type `yes` or `no` to indicate whether to overwrite an existing schema. The default answer is `yes`.
6. Examine the script output and the log file to verify that the installation completed successfully.

# Resetting Sample Schemas

Typically, there is no difference between installing a sample schema for the first time or reinstalling it over a previously installed version. By default, the `*_install.sql` scripts drop the schema users and all of their objects from the previous installation.

Therefore, to reset the sample schemas, follow the steps mentioned in the section Installing the Sample Schemas.

# Uninstalling Sample Schemas

To uninstall a sample schema, run the corresponding `*_uninstall.sql` script from a client program such as SQL*Plus or SQLcl.

1. Connect to the database as a privileged user with rights to create and drop other users.

   For example, connect as `SYSTEM` or `ADMIN`.
2. Run the `*_uninstall.sql` uninstall script.

   For example, to uninstall the `hr` schema, run the `hr_uninstall.sql` script.

3

# Schema Diagrams

The following diagrams illustrate the structures of the sample database schemas.

## Sample Schema Diagrams

Figure 3-1 illustrates sample schemas `hr` and `oe` and their relationship. For information about the scripts and table descriptions for these schemas, see HR Schema and OE Schema.

Figure 3-2 illustrates schema `co`. For information about the scripts and table description for this schema, see CO Schema

Figure 3-3 illustrates schema `sh`. For information about the scripts and table description for this schema, see SH Schema.

Figure 3-4 illustrates schema `pm`. For information about the scripts and table description for this schema, see PM Schema.

**Figure 3-1 Sample Schemas HR and OE**

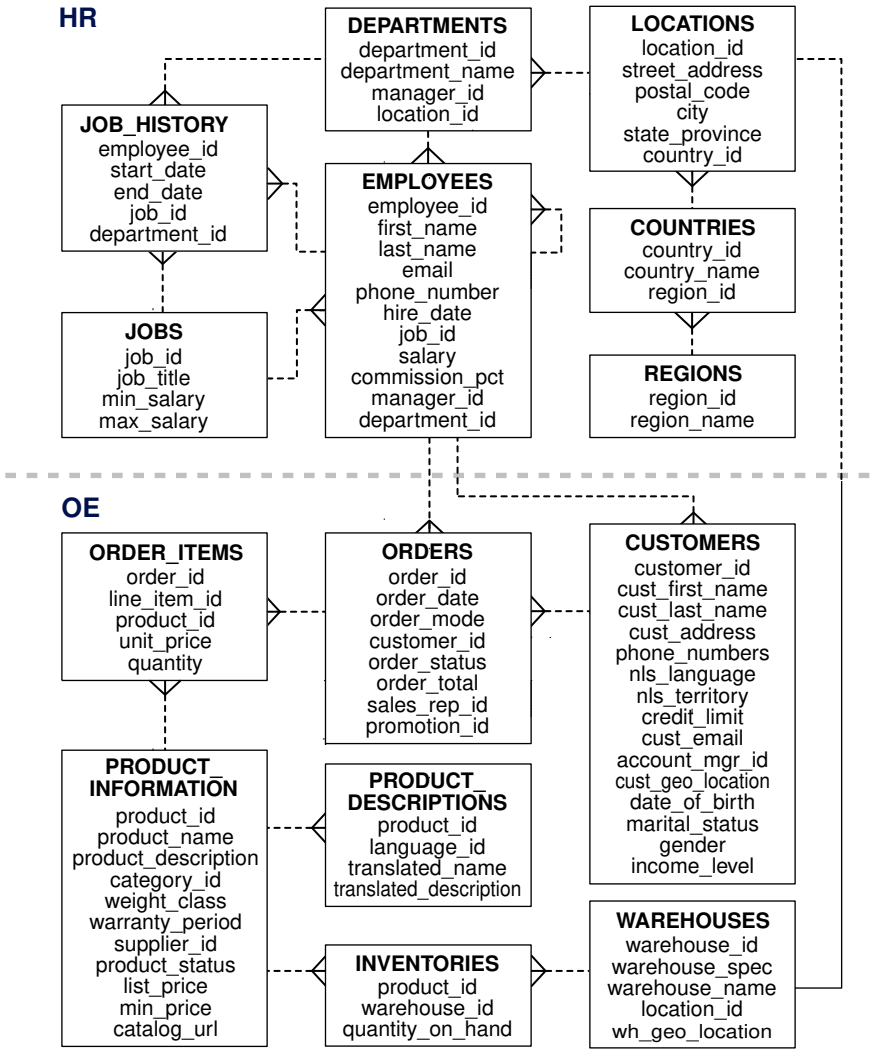


**Figure 3-2 Sample Schema CO**

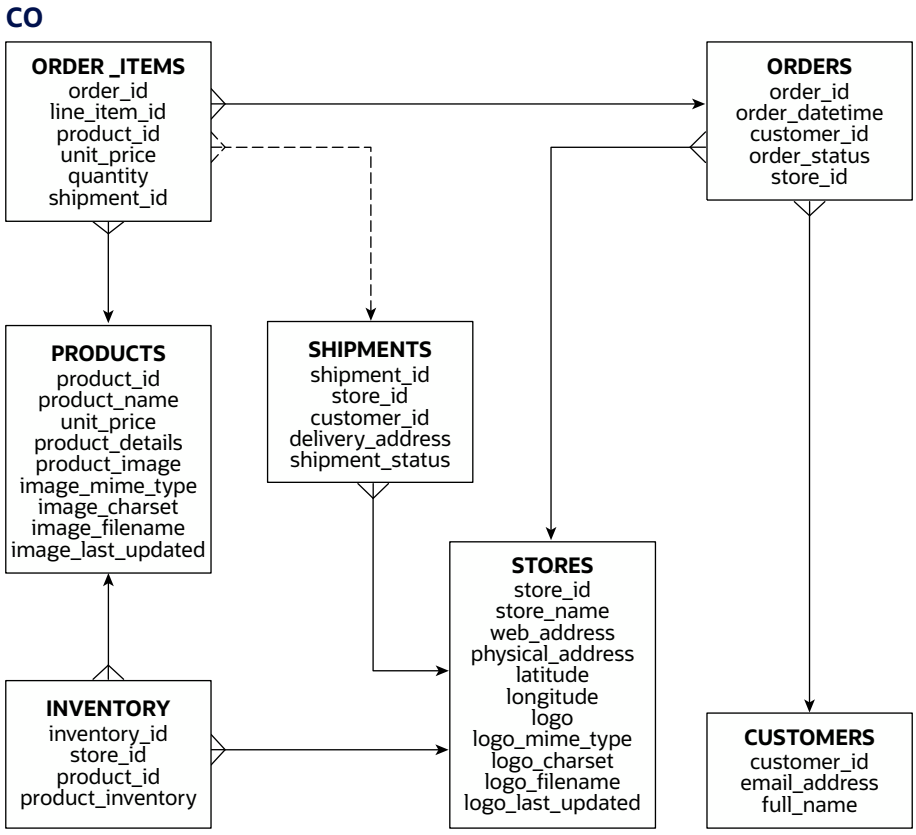


**Figure 3-3 Sample Schema SH**

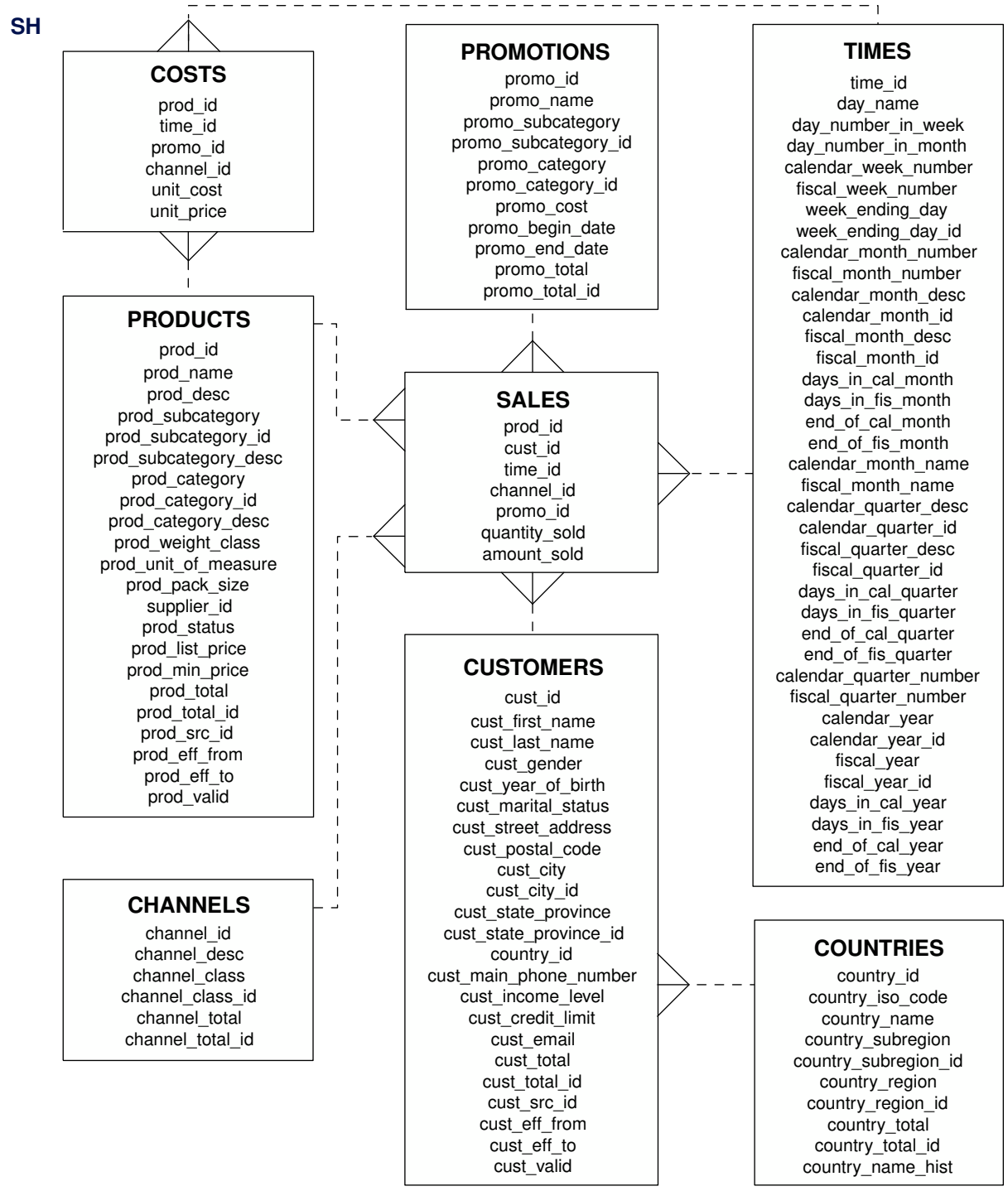
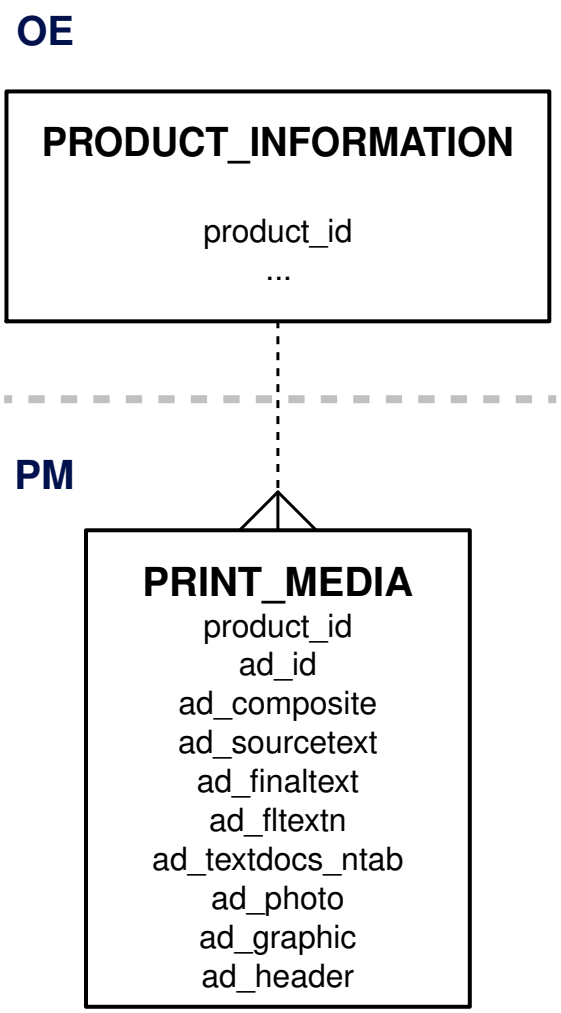


**Figure 3-4 Sample Schemas OE and PM**



4

# Sample Schema Scripts and Object Descriptions

Each of the sample schemas has two primary scripts:

- The *xx*`_install.sql` script (where *xx* is the schema abbreviation) resets and creates all objects and data for a particular schema.
- The *xx*`_uninstall.sql` script (where *xx* is the schema abbreviation) removes all objects from a particular schema.

## HR Sample Schema Scripts and Objects

The following tables list the names of the scripts that create the human resources (`hr`) schema and describe the objects in the schema. Table 4-1 lists the `hr` scripts in alphabetical order, while Table 4-2 lists the `hr` objects.

**Table 4-1 HR Sample Schema Scripts**

| Script Name | Description |
|---|---|
| `hr_code.sql` | Creates procedural objects in the schema |
| `hr_create.sql` | Creates the `hr` objects |
| `hr_install.sql` | Main script for schema `hr`; calls other scripts |
| `hr_populate.sql` | Populates the objects |
| `hr_uninstall.sql` | Uninstalls the schema |

**Table 4-2 HR Sample Schema Objects**

| Object Type | Objects |
|---|---|
| Index | `country_c_id_pk`, `dept_id_pk`, `dept_location_ix`, `emp_department_ix`, `emp_email_uk`, `emp_emp_id_pk`, `emp_job_ix`, `emp_manager_ix`, `emp_name_ix`, `jhist_department_ix`, `jhist_employee_ix`, `jhist_emp_id_st_date_pk`, `jhist_job_ix`, `job_id_pk`, `loc_city_ix`, `loc_country_ix`, `loc_id_pk`, `loc_state_province_ix`, `reg_id_pk` |
| Procedure | `add_job_history`, `secure_dml` |
| Sequence | `departments_seq`, `employees_seq`, `locations_seq` |
| Table | `countries`, `departments`, `employees`, `jobs`, `job_history`, `locations`, `regions` |
| Trigger | `secure_employees`, `update_job_history` |
| View | `emp_details_view` |

## HR Sample Schema Table Descriptions

The following tables describe the columns of each table of the `hr` sample schema.

# Table HR.COUNTRIES

**Table 4-3 HR.COUNTRIES Table Description**

| Column Name | Null? | Type |
|---|---|---|
| COUNTRY_ID | NOT NULL | CHAR(2) |
| COUNTRY_NAME | | VARCHAR2(60) |
| REGION_ID | | NUMBER |

# Table HR.DEPARTMENTS

**Table 4-4 HR.DEPARTMENTS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| DEPARTMENT_ID | NOT NULL | NUMBER(4) |
| DEPARTMENT_NAME | NOT NULL | VARCHAR2(30) |
| MANAGER_ID | | NUMBER(6) |
| LOCATION_ID | | NUMBER(4) |

# Table HR.EMPLOYEES

**Table 4-5 HR.EMPLOYEES Table Description**

| Column Name | Null? | Type |
|---|---|---|
| EMPLOYEE_ID | NOT NULL | NUMBER(6) |
| FIRST_NAME | | VARCHAR2(20) |
| LAST_NAME | NOT NULL | VARCHAR2(25) |
| EMAIL | NOT NULL | VARCHAR2(25) |
| PHONE_NUMBER | | VARCHAR2(20) |
| HIRE_DATE | NOT NULL | DATE |
| JOB_ID | NOT NULL | VARCHAR2(10) |
| SALARY | | NUMBER(8,2) |
| COMMISSION_PCT | | NUMBER(2,2) |
| MANAGER_ID | | NUMBER(6) |

**Table 4-5 (Cont.) HR.EMPLOYEES Table Description**

| Column Name | Null? | Type |
|---|---|---|
| DEPARTMENT_ID | | NUMBER(4) |

# Table HR.JOBS

**Table 4-6 HR.JOBS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| JOB_ID | NOT NULL | VARCHAR2(10) |
| JOB_TITLE | NOT NULL | VARCHAR2(35) |
| MIN_SALARY | | NUMBER(6) |
| MAX_SALARY | | NUMBER(6) |

# Table HR.JOB_HISTORY

**Table 4-7 HR.JOB_HISTORY Table Description**

| Column Name | Null? | Type |
|---|---|---|
| EMPLOYEE_ID | NOT NULL | NUMBER(6) |
| START_DATE | NOT NULL | DATE |
| END_DATE | NOT NULL | DATE |
| JOB_ID | NOT NULL | VARCHAR2(10) |
| DEPARTMENT_ID | | NUMBER(4) |

# Table HR.LOCATIONS

**Table 4-8 HR.LOCATIONS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| LOCATION_ID | NOT NULL | NUMBER(4) |
| STREET_ADDRESS | | VARCHAR2(40) |
| POSTAL_CODE | | VARCHAR2(12) |
| CITY | NOT NULL | VARCHAR2(30) |
| STATE_PROVINCE | | VARCHAR2(25) |
| COUNTRY_ID | | CHAR(2) |

## Table HR.REGIONS

**Table 4-9 HR.REGIONS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| REGION_ID | NOT NULL | NUMBER |
| REGION_NAME | | VARCHAR2(25) |

# CO Sample Schema Scripts and Objects

The following tables list the names of the scripts that create the customer orders (co) schema and describe the objects in the schema.

Table 4-10 lists the co scripts in alphabetical order, while Table 4-11 lists the co objects.

**Table 4-10 CO Sample Schema Scripts**

| Script Name | Description |
|---|---|
| co_create.sql | Creates the co objects |
| co_install.sql | Main script for schema co; calls other scripts |
| co_populate.sql | Populates the objects |
| co_uninstall.sql | Uninstalls the schema |

**Table 4-11 CO Sample Schema Objects**

| Object Type | Objects |
|---|---|
| Index | customers_name_i, orders_customer_id_i, orders_store_id_i, products_pk, stores_pk, store_name_u, customers_pk, customers_email_u, orders_pk, order_items_pk, order_items_product_u, inventory_product_id_i, inventory_pk, inventory_store_product_u, shipments_store_id_i, shipments_customer_id_i, shipments_pk |
| Table | customers, stores, products, orders, order_items, shipments, inventory |
| View | customer_order_products, store_orders, product_reviews, product_orders |

# CO Sample Schema Table Descriptions

The following tables describe the columns of each table of the co sample schema.

If the value of a primary key column is NULL, an integer is generated by default for all of the co Sample Schema tables.

- Table CO.CUSTOMERS
- Table CO.STORES

# Table CO.CUSTOMERS

The table below lists `CO.CUSTOMERS` table description.

**Table 4-12 CO.CUSTOMERS Table Description**

| Column Name | Nulls Allowed? | Type |
|---|---|---|
| CUSTOMER_ID | NO | INTEGER |
| EMAIL_ADDRESS | NO | VARCHAR2(255 CHAR) |
| FULL_NAME | NO | VARCHAR2(255 CHAR) |

# Table CO.STORES

The table below lists `CO.STORES` table description.

**Table 4-13 Table CO.STORES Table Description**

| Column Name | Nulls Allowed? | Type |
|---|---|---|
| STORE_ID | NO | INTEGER |
| STORE_NAME | NO | VARCHAR2(255 CHAR) |
| WEB_ADDRESS | YES | VARCHAR2(100 CHAR) |
| PHYSICAL_ADDRESS | YES | VARCHAR2(512 CHAR) |
| LATITUDE | YES | NUMBER(9,6) |
| LONGITUDE | YES | NUMBER(9,6) |
| LOGO | YES | BLOB |
| LOGO_MIME_TYPE | YES | VARCHAR2(512 CHAR) |
| LOGO_FILENAME | YES | VARCHAR2(512 CHAR) |
| LOGO_CHARSET | YES | VARCHAR2(512 CHAR) |
| LOGO_LAST_UPDATED | YES | DATE |

> **Note**
>
> The table has a constraint to verify either `WEB_ADDRESS` or `PHYSICAL ADDRESS` is `NOT NULL`.

# Table CO.PRODUCTS

The table below lists `CO.PRODUCTS` table description.

**Table 4-14 Table CO.PRODUCTS Table Description**

| Column Name | Nulls Allowed? | Type |
|---|---|---|
| PRODUCT_ID | NO | INTEGER |
| PRODUCT_NAME | NO | VARCHAR2(255 CHAR) |
| UNIT_PRICE | YES | NUMBER(10,2) |
| PRODUCT_DETAILS | YES | BLOB |
| PRODUCT_IMAGE | YES | BLOB |
| IMAGE_MIME_TYPE | YES | VARCHAR2(512 CHAR) |
| IMAGE_FILENAME | YES | VARCHAR2(512 CHAR) |
| IMAGE_CHARSET | YES | VARCHAR2(512 CHAR) |
| IMAGE_LAST_UPDATED | YES | DATE |

## Table CO.ORDERS

The table below lists CO.ORDERS table description.

**Table 4-15 Table CO.ORDERS Table Description**

| Column Name | Nulls Allowed? | Type |
|---|---|---|
| ORDER_ID | NO | INTEGER |
| ORDER_TMS | NO | TIMESTAMP(6) |
| CUSTOMER_ID | NO | INTEGER |
| ORDER_STATUS | NO | VARCHAR2(10 CHAR) |
| STORE_ID | NO | INTEGER |

## Table CO.ORDER_ITEMS

The table below lists CO.ORDER_ITEMS table description.

**Table 4-16 Table CO.ORDER_ITEMS Table Description**

| Column Name | Nulls Allowed? | Type |
|---|---|---|
| ORDER_ID | NO | INTEGER |
| LINE_ITEM_ID | NO | INTEGER |
| PRODUCT_ID | NO | INTEGER |
| UNIT_PRICE | NO | NUMBER(10,2) |
| QUANTITY | NO | INTEGER |
| SHIPMENT_ID | YES | INTEGER |

## Table CO.SHIPMENTS

The table below lists CO.SHIPMENTS table description.

**Table 4-17 Table CO.SHIPMENTS Table Description**

| Column Name | Nulls Allowed? | Type |
|---|---|---|
| SHIPMENT_ID | NO | INTEGER |
| STORE_ID | NO | INTEGER |
| CUSTOMER_ID | NO | INTEGER |
| DELIVERY_ADDRESS | NO | VARCHAR2(512 CHAR) |
| SHIPMENT_STATUS | NO | VARCHAR2(100 CHAR) |

## Table CO.INVENTORY

The table below lists CO.INVENTORY table description.

**Table 4-18 Table CO.INVENTORY Table Description**

| Column Name | Nulls Allowed? | Type |
|---|---|---|
| INVENTORY_ID | NO | INTEGER |
| STORE_ID | NO | INTEGER |
| PRODUCT_ID | NO | INTEGER |
| PRODUCT_INVENTORY | NO | INTEGER |

# SH Sample Schema Scripts and Objects

The following tables list the names of the scripts that create the Sales History (sh) schema and describe the objects in the schema. Table 4-19 lists the sh scripts in alphabetical order, while Table 4-20 lists the sh objects.

**Table 4-19 SH Sample Schema Scripts**

| Script Name | Description |
|---|---|
| sh_create.sql | Creates the objects in the schema |
| sh_install.sql | Main script for schema sh; calls other scripts |
| sh_populate.sql | Populates the objects |
| sh_uninstall.sql | Uninstalls the schema |

**Table 4-20 SH Sample Schema Objects**

| Object Type | Objects |
|---|---|
| Dimension | channels_dim, customers_dim, products_dim, promotions_dim, times_dim |
| Index | channels_pk, countries_pk, customers_gender_bix, customers_marital_bix, customers_pk, customers_yob_bix, dr$sup_text_idx$x, fw_psc_s_mv_chan_bix, fw_psc_s_mv_promo_bix, fw_psc_s_mv_subcat_bix, fw_psc_s_mv_wd_bix, products_pk, products_prod_cat_ix, products_prod_status_bix, products_prod_subcat_ix, promo_pk, sup_text_idx, times_pk |

**Table 4-20 (Cont.) SH Sample Schema Objects**

| Object Type | Objects |
|---|---|
| Partitioned Index | costs_prod_bix, costs_time_bix, sales_channel_bix, sales_cust_bix, sales_prod_bix, sales_promo_bix, sales_time_bix |
| Materialized View | cal_month_sales_mv, fweek_pscat_sales_mv |
| Table | channels, countries, customers, products, promotions, times |
| Partitioned Table | costs, sales |
| View | profits |

# SH Sample Schema Table Descriptions

The following tables describe the columns of each table of the sh sample schema.



## Table SH.CHANNELS

**Table 4-21 SH.CHANNELS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| CHANNEL_ID | NOT NULL | NUMBER |
| CHANNEL_DESC | NOT NULL | VARCHAR2(20) |
| CHANNEL_CLASS | NOT NULL | VARCHAR2(20) |
| CHANNEL_CLASS_ID | NOT NULL | NUMBER |
| CHANNEL_TOTAL | NOT NULL | VARCHAR2(13) |
| CHANNEL_TOTAL_ID | NOT NULL | NUMBER |

## Table SH.COSTS

**Table 4-22 SH.COSTS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| PROD_ID | NOT NULL | NUMBER |
| TIME_ID | NOT NULL | DATE |

**Table 4-22 (Cont.) SH.COSTS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| PROMO_ID | NOT NULL | NUMBER |
| CHANNEL_ID | NOT NULL | NUMBER |
| UNIT_COST | NOT NULL | NUMBER(10,2) |
| UNIT_PRICE | NOT NULL | NUMBER(10,2) |

# Table SH.COUNTRIES

**Table 4-23 SH.COUNTRIES Table Description**

| Column Name | Null? | Type |
|---|---|---|
| COUNTRY_ID | NOT NULL | NUMBER |
| COUNTRY_ISO_CODE | NOT NULL | CHAR(2) |
| COUNTRY_NAME | NOT NULL | VARCHAR2(40) |
| COUNTRY_SUBREGION | NOT NULL | VARCHAR2(30) |
| COUNTRY_SUBREGION_ID | NOT NULL | NUMBER |
| COUNTRY_REGION | NOT NULL | VARCHAR2(20) |
| COUNTRY_REGION_ID | NOT NULL | NUMBER |
| COUNTRY_TOTAL | NOT NULL | VARCHAR2(11) |
| COUNTRY_TOTAL_ID | NOT NULL | NUMBER |

# Table SH.CUSTOMERS

**Table 4-24 SH.CUSTOMERS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| CUST_ID | NOT NULL | NUMBER |
| CUST_FIRST_NAME | NOT NULL | VARCHAR2(20) |
| CUST_LAST_NAME | NOT NULL | VARCHAR2(40) |
| CUST_GENDER | NOT NULL | CHAR(1) |
| CUST_YEAR_OF_BIRTH | NOT NULL | NUMBER(4) |
| CUST_MARITAL_STATUS | | VARCHAR2(20) |
| CUST_STREET_ADDRESS | NOT NULL | VARCHAR2(40) |
| CUST_POSTAL_CODE | NOT NULL | VARCHAR2(10) |
| CUST_CITY | NOT NULL | VARCHAR2(30) |
| CUST_CITY_ID | NOT NULL | NUMBER |
| CUST_STATE_PROVINCE | NOT NULL | VARCHAR2(40) |
| CUST_STATE_PROVINCE_ID | NOT NULL | NUMBER |
| COUNTRY_ID | NOT NULL | NUMBER |

**Table 4-24 (Cont.) SH.CUSTOMERS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| CUST_MAIN_PHONE_NUMBER | NOT NULL | VARCHAR2(25) |
| CUST_INCOME_LEVEL | | VARCHAR2(30) |
| CUST_CREDIT_LIMIT | | NUMBER |
| CUST_EMAIL | | VARCHAR2(50) |
| CUST_TOTAL | NOT NULL | VARCHAR2(14) |
| CUST_TOTAL_ID | NOT NULL | NUMBER |
| CUST_SRC_ID | | NUMBER |
| CUST_EFF_FROM | | DATE |
| CUST_EFF_TO | | DATE |
| CUST_VALID | | VARCHAR2(1) |

# Table SH.PRODUCTS

**Table 4-25 SH.PRODUCTS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| PROD_ID | NOT NULL | NUMBER(6) |
| PROD_NAME | NOT NULL | VARCHAR2(50) |
| PROD_DESC | NOT NULL | VARCHAR2(4000) |
| PROD_SUBCATEGORY | NOT NULL | VARCHAR2(50) |
| PROD_SUBCATEGORY_ID | NOT NULL | NUMBER |
| PROD_SUBCATEGORY_DESC | NOT NULL | VARCHAR2(2000) |
| PROD_CATEGORY | NOT NULL | VARCHAR2(50) |
| PROD_CATEGORY_ID | NOT NULL | NUMBER |
| PROD_CATEGORY_DESC | NOT NULL | VARCHAR2(2000) |
| PROD_WEIGHT_CLASS | NOT NULL | NUMBER(3) |
| PROD_UNIT_OF_MEASURE | | VARCHAR2(20) |
| PROD_PACK_SIZE | NOT NULL | VARCHAR2(30) |
| SUPPLIER_ID | NOT NULL | NUMBER(6) |
| PROD_STATUS | NOT NULL | VARCHAR2(20) |
| PROD_LIST_PRICE | NOT NULL | NUMBER(8,2) |
| PROD_MIN_PRICE | NOT NULL | NUMBER(8,2) |
| PROD_TOTAL | NOT NULL | VARCHAR2(13) |
| PROD_TOTAL_ID | NOT NULL | NUMBER |
| PROD_SRC_ID | | NUMBER |
| PROD_EFF_FROM | | DATE |
| PROD_EFF_TO | | DATE |
| PROD_VALID | | VARCHAR2(1) |

# Table SH.PROMOTIONS

**Table 4-26 SH.PROMOTIONS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| PROMO_ID | NOT NULL | NUMBER(6) |
| PROMO_NAME | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY | NOT NULL | VARCHAR2(30) |
| PROMO_SUBCATEGORY_ID | NOT NULL | NUMBER |
| PROMO_CATEGORY | NOT NULL | VARCHAR2(30) |
| PROMO_CATEGORY_ID | NOT NULL | NUMBER |
| PROMO_COST | NOT NULL | NUMBER(10,2) |
| PROMO_BEGIN_DATE | NOT NULL | DATE |
| PROMO_END_DATE | NOT NULL | DATE |
| PROMO_TOTAL | NOT NULL | VARCHAR2(15) |
| PROMO_TOTAL_ID | NOT NULL | NUMBER |

# Table SH.SALES

**Table 4-27 SH.SALES Table Description**

| Column Name | Null? | Type |
|---|---|---|
| PROD_ID | NOT NULL | NUMBER(6) |
| CUST_ID | NOT NULL | NUMBER |
| TIME_ID | NOT NULL | DATE |
| CHANNEL_ID | NOT NULL | NUMBER(1) |
| PROMO_ID | NOT NULL | NUMBER(6) |
| QUANTITY_SOLD | NOT NULL | NUMBER(3) |
| AMOUNT_SOLD | NOT NULL | NUMBER(10,2) |

# Table SH.TIMES

**Table 4-28 SH.TIMES Table Description**

| Column Name | Null? | Type |
|---|---|---|
| TIME_ID | NOT NULL | DATE |
| DAY_NAME | NOT NULL | VARCHAR2(9) |
| DAY_NUMBER_IN_WEEK | NOT NULL | NUMBER(1) |
| DAY_NUMBER_IN_MONTH | NOT NULL | NUMBER(2) |
| CALENDAR_WEEK_NUMBER | NOT NULL | NUMBER(2) |
| FISCAL_WEEK_NUMBER | NOT NULL | NUMBER(2) |

**Table 4-28 (Cont.) SH.TIMES Table Description**

| Column Name | Null? | Type |
|---|---|---|
| WEEK_ENDING_DAY | NOT NULL | DATE |
| WEEK_ENDING_DAY_ID | NOT NULL | NUMBER |
| CALENDAR_MONTH_NUMBER | NOT NULL | NUMBER(2) |
| FISCAL_MONTH_NUMBER | NOT NULL | NUMBER(2) |
| CALENDAR_MONTH_DESC | NOT NULL | VARCHAR2(8) |
| CALENDAR_MONTH_ID | NOT NULL | NUMBER |
| FISCAL_MONTH_DESC | NOT NULL | VARCHAR2(8) |
| FISCAL_MONTH_ID | NOT NULL | NUMBER |
| DAYS_IN_CAL_MONTH | NOT NULL | NUMBER |
| DAYS_IN_FIS_MONTH | NOT NULL | NUMBER |
| END_OF_CAL_MONTH | NOT NULL | DATE |
| END_OF_FIS_MONTH | NOT NULL | DATE |
| CALENDAR_MONTH_NAME | NOT NULL | VARCHAR2(9) |
| FISCAL_MONTH_NAME | NOT NULL | VARCHAR2(9) |
| CALENDAR_QUARTER_DESC | NOT NULL | CHAR(7) |
| CALENDAR_QUARTER_ID | NOT NULL | NUMBER |
| FISCAL_QUARTER_DESC | NOT NULL | CHAR(7) |
| FISCAL_QUARTER_ID | NOT NULL | NUMBER |
| DAYS_IN_CAL_QUARTER | NOT NULL | NUMBER |
| DAYS_IN_FIS_QUARTER | NOT NULL | NUMBER |
| END_OF_CAL_QUARTER | NOT NULL | DATE |
| END_OF_FIS_QUARTER | NOT NULL | DATE |
| CALENDAR_QUARTER_NUMBER | NOT NULL | NUMBER(1) |
| FISCAL_QUARTER_NUMBER | NOT NULL | NUMBER(1) |
| CALENDAR_YEAR | NOT NULL | NUMBER(4) |
| CALENDAR_YEAR_ID | NOT NULL | NUMBER |
| FISCAL_YEAR | NOT NULL | NUMBER(4) |
| FISCAL_YEAR_ID | NOT NULL | NUMBER |
| DAYS_IN_CAL_YEAR | NOT NULL | NUMBER |
| DAYS_IN_FIS_YEAR | NOT NULL | NUMBER |
| END_OF_CAL_YEAR | NOT NULL | DATE |
| END_OF_FIS_YEAR | NOT NULL | DATE |

# OE Sample Schema Scripts and Objects

The following tables list the names of the scripts that create the Order Entry (oe) sample schema and describe the objects in the schema. Table 4-29 lists the oe scripts in alphabetical order, while Table 4-30 lists the oe objects. Note that language-specific statements for product names and descriptions are stored in the following files, where each file represents a different

language: oe_p_us.sql, oe_p_ar.sql, oe_p_cs.sql, oe_p_d.sql, oe_p_dk.sql, oe_p_e.sql, oe_p_el.sql, oe_p_esa.sql, oe_p_f.sql, oe_p_frc.sql, oe_p_hu.sql, oe_p_i.sql, oe_p_iw.sql, oe_p_ja.sql, oe_p_ko.sql, oe_p_n.sql, oe_p_nl.sql, oe_p_pl.sql, oe_p_pt.sql, oe_p_ptb.sql, oe_p_ro.sql, oe_p_ru.sql, oe_p_s.sql, oe_p_sf.sql, oe_p_sk.sql, oe_p_th.sql, oe_p_tr.sql, oe_p_zhs.sql, oe_p_zht.sql.

**Table 4-29 OE Sample Schema Scripts**

| Script Name | Description |
|---|---|
| oc_comnt.sql | Adds comments to the online catalog (oc) subschema wherever possible |
| oc_cre.sql | Creates subschema oc |
| oc_drop.sql | Drops subschema oc |
| oc_main.sql | Main script for subschema oc |
| oc_popul.sql | Populates the object tables |
| oe_analz.sql | Gathers statistics on the oe objects |
| oe_comnt.sql | Creates comments for the objects in the schema |
| oe_cre.sql | Creates the oe objects |
| oe_drop.sql | Drops schema oe and all its objects |
| oe_idx.sql | Creates indexes on the oe tables |
| oe_main.sql | Main script for the oe schema; calls other scripts |
| oe_views.sql | Creates the oe schema views |

**Table 4-30 OE Sample Schema Objects**

| Object Type | Objects |
|---|---|
| Index | customers_pk, cust_account_manager_ix, cust_email_ix, cust_lname_ix, cust_upper_name_ix, inventory_ix, inv_product_ix, item_order_ix, item_product_ix, order_items_pk, order_items_uk, order_pk, ord_customer_ix, ord_order_date_ix, ord_sales_rep_ix, prd_desc_pk, product_information_pk, prod_name_ix, prod_supplier_ix, promo_id_pk, reference_is_unique, sys_c003584, sys_c003587, sys_c003588, sys_c003589, sys_c003590, warehouses_pk, whs_location_ix |
| Function | get_phone_number_f |
| Sequence | orders_seq |
| Synonym | countries, departments, employees, jobs, job_history, locations |
| Table | customers, inventories, orders, order_items, product_descriptions, product_information, warehouses |
| Trigger | insert_ord_line, orders_items_trg, orders_trg |

**Table 4-30 (Cont.) OE Sample Schema Objects**

| Object Type | Objects |
|---|---|
| Type | catalog_typ, category_typ, composite_category_typ, corporate_customer_typ, customer_typ, cust_address_typ, inventory_list_typ, inventory_typ, leaf_category_typ, order_item_list_typ, order_item_typ, order_list_typ, order_typ, phone_list_typ, product_information_typ, product_ref_list_typ, subcategory_ref_list_typ, sys_yoid0000046073$, sys_yoid0000046075$, sys_yoid0000046077$, sys_yoid0000046079$, sys_yoid0000046081$, warehouse_typ, xdbpo_actions_type, xdbpo_action_collection, xdbpo_action_type, xdbpo_lineitems_type, xdbpo_lineitem_collection, xdbpo_lineitem_type, xdbpo_part_type, xdbpo_rejection_type, xdbpo_shipinstructions_type, xdbpo_type |
| Type Body | catalog_typ, composite_category_typ, leaf_category_typ |
| View | account_managers, bombay_inventory, customers_view, deptview, oc_corporate_customers, oc_customers, oc_inventories, oc_orders, oc_product_information, orders_view, products, product_prices, sydney_inventory, toronto_inventory |

# OE Sample Schema Table Descriptions

The following tables describe the columns of each table of the oe sample schema.



## Table OE.CUSTOMERS

**Table 4-31 OE.CUSTOMERS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| CUSTOMER_ID | NOT NULL | NUMBER(6) |
| CUST_FIRST_NAME | NOT NULL | VARCHAR2(20) |
| CUST_LAST_NAME | NOT NULL | VARCHAR2(20) |
| CUST_ADDRESS | | CUST_ADDRESS_TYP |
| PHONE_NUMBERS | | PHONE_LIST_TYP |
| NLS_LANGUAGE | | VARCHAR2(3) |
| NLS_TERRITORY | | VARCHAR2(30) |

**Table 4-31 (Cont.) OE.CUSTOMERS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| CREDIT_LIMIT | | NUMBER(9,2) |
| CUST_EMAIL | | VARCHAR2(30) |
| ACCOUNT_MGR_ID | | NUMBER(6) |
| CUST_GEO_LOCATION | | MDSYS.SDO_GEOMETRY |
| DATE_OF_BIRTH | | DATE |
| MARITAL_STATUS | | VARCHAR2(20) |
| GENDER | | VARCHAR2(1) |
| INCOME_LEVEL | | VARCHAR2(20) |

# Table OE.INVENTORIES

**Table 4-32 OE.INVENTORIES Table Description**

| Column Name | Null? | Type |
|---|---|---|
| PRODUCT_ID | NOT NULL | NUMBER(6) |
| WAREHOUSE_ID | NOT NULL | NUMBER(3) |
| QUANTITY_ON_HAND | NOT NULL | NUMBER(8) |

# Table OE.ORDERS

**Table 4-33 OE.ORDERS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| ORDER_ID | NOT NULL | NUMBER(12) |
| ORDER_DATE | NOT NULL | TIMESTAMP(6) WITH LOCAL TIME ZONE |
| ORDER_MODE | | VARCHAR2(8) |
| CUSTOMER_ID | NOT NULL | NUMBER(6) |
| ORDER_STATUS | | NUMBER(2) |
| ORDER_TOTAL | | NUMBER(8,2) |
| SALES_REP_ID | | NUMBER(6) |
| PROMOTION_ID | | NUMBER(6) |

# Table OE.ORDER_ITEMS

**Table 4-34 OE.ORDER_ITEMS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| ORDER_ID | NOT NULL | NUMBER(12) |

**Table 4-34 (Cont.) OE.ORDER_ITEMS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| LINE_ITEM_ID | NOT NULL | NUMBER(3) |
| PRODUCT_ID | NOT NULL | NUMBER(6) |
| UNIT_PRICE | | NUMBER(8,2) |
| QUANTITY | | NUMBER(8) |

# Table OE.PRODUCT_DESCRIPTIONS

**Table 4-35 OE.PRODUCT_DESCRIPTIONS Table Description**

| Column Name | Null? | Type |
|---|---|---|
| PRODUCT_ID | NOT NULL | NUMBER(6) |
| LANGUAGE_ID | NOT NULL | VARCHAR2(3) |
| TRANSLATED_NAME | NOT NULL | NVARCHAR2(50) |
| TRANSLATED_DESCRIPTION | NOT NULL | NVARCHAR2(2000) |

# Table OE.PRODUCT_INFORMATION

**Table 4-36 OE.PRODUCT_INFORMATION Table Description**

| Column Name | Null? | Type |
|---|---|---|
| PRODUCT_ID | NOT NULL | NUMBER(6) |
| PRODUCT_NAME | | VARCHAR2(50) |
| PRODUCT_DESCRIPTION | | VARCHAR2(2000) |
| CATEGORY_ID | | NUMBER(2) |
| WEIGHT_CLASS | | NUMBER(1) |
| WARRANTY_PERIOD | | INTERVAL YEAR(2) TO MONTH |
| SUPPLIER_ID | | NUMBER(6)) |
| PRODUCT_STATUS | | VARCHAR2(20) |
| LIST_PRICE | | NUMBER(8,2) |
| MIN_PRICE | | NUMBER(8,2) |
| CATALOG_URL | | VARCHAR2(50) |

# Table OE.WAREHOUSES

**Table 4-37 OE.WAREHOUSES Table Description**

| Column Name | Null? | Type |
|---|---|---|
| WAREHOUSE_ID | NOT NULL | NUMBER(3) |

**Table 4-37 (Cont.) OE.WAREHOUSES Table Description**

| Column Name | Null? | Type |
|---|---|---|
| WAREHOUSE_SPEC | | SYS.XMLTYPE |
| WAREHOUSE_NAME | | VARCHAR2(35) |
| LOCATION_ID | | NUMBER(4) |
| WH_GEO_LOCATION | | MDSYS.SDO_GEOMETRY |

Column `warehouse_spec` of table `OE.warehouses` contains `XMLType` data. This data is not based on any XML schema, which means that it can take any form. However, the actual data in column `warehouse_spec` at the outset (before any changes you might have made to it) has a top-level element `Warehouse` with the following child elements:

- `Building`, with text node `Owned` or `Rented`
- `Area`, with text node a number (representing, for example, square feet)
- `Docks`, with text node the number of loading docks (for example, 1, 2, or 3)
- `DockType`, with text node empty or `Rear Load` or `Side Load`
- `WaterAccess`, with text node `Y` or `N`
- `RailAccess`, with text node `Y` or `N`
- `Parking`, with text node `Street` or `Lot`
- `VClearance` (vertical clearance), with text node a number followed by a linear unit (for example, `11.5 ft`)

> **See Also**
>
> *Oracle XML DB Developer's Guide* for examples using the `XMLType` data in column `warehouse_spec`

## Table OE.PURCHASEORDER

Table `OE.purchaseorder` is an object-relational table with `XMLType` data. The data conforms to XML schema `purchaseOrder.xsd`.

# PM Sample Schema Scripts and Objects

This following tables list the names of the scripts that create the Product Media (`pm`) schema and describe the objects in the schema. Table 4-38 lists the `pm` scripts in alphabetical order, while Table 4-39 lists the `pm` objects. Note that the SQL*Loader data file `pm_p_lob.dat` contains hard-coded absolute path names that have been set during installation. Before attempting to load the data in a different environment, you must first edit the path names in this file.

**Table 4-38 PM Schema Scripts**

| Script Name | Description |
|---|---|
| pm_analz.sql | Gathers statistics on the pm objects |
| pm_cre.sql | Creates the pm objects |
| pm_drop.sql | Drops schema pm and all its objects |
| pm_main.sql | Main script for schema pm, which calls other scripts |
| pm_p_lob.sql, | Populates the objects in the schema |

**Table 4-39 PM Sample Schema Objects**

| Object Type | Objects |
|---|---|
| Index | printmedia_pk, sys_c003538 |
| Table | print_media |
| Type | adheader_typ, textdoc_tab, textdoc_typ |

# PM Sample Schema Table Descriptions

The following table describes the columns of the print_media table in the pm sample schema.



## Table PM.PRINT_MEDIA

**Table 4-40 PM.PRINT_MEDIA Table Description**

| Column Name | Null? | Type |
|---|---|---|
| PRODUCT_ID | NOT NULL | NUMBER(6) |
| AD_ID | NOT NULL | NUMBER(6) |
| AD_COMPOSITE | | BLOB |
| AD_SOURCETEXT | | CLOB |
| AD_FINALTEXT | | CLOB |
| AD_FLTEXTN | | NCLOB |
| AD_TEXTDOCS_NTAB | | TEXTDOC_TAB |
| AD_PHOTO | | BLOB |
| AD_GRAPHIC | | BINARY FILE LOB |
| AD_HEADER | | ADHEADER_TYP |

# Index

## E



## F



## G



## H



## I



## J



## L



## M



## N



## O

## P



## Q



## R



## S



## T

## U



## W