

Share




DATABASE, SQL AND PL/SQL

Document Entities

Generate logical models and entity relationship diagrams easily from an existing database with Oracle SQL Developer Data Modeler.

By Jeff Smith

May/June 2014

Without adequate design documentation, it can be difficult for your Oracle Database application's stakeholders to understand how the database entities are defined and how they interact with one another. A *logical model*—represented by an *entity relationship diagram* (ERD)—clarifies these definitions and relationships. For example, an ERD can show at a glance that multiple departments can be associated with an employee or that a country can include one or more locations. Oracle SQL Developer Data Modeler—automatically included as an extension in Oracle SQL Developer (and also available in )—makes it easy to produce ERDs and share them with others who need to review or approve the database design.

This column steps you through the process of creating a *relational* data model from an existing database and then converting the relational model to a logical model and a resulting ERD, such as the one in Figure 1. After you generate the diagram, you'll see how you can customize your diagrams and share them in various formats with your colleagues.

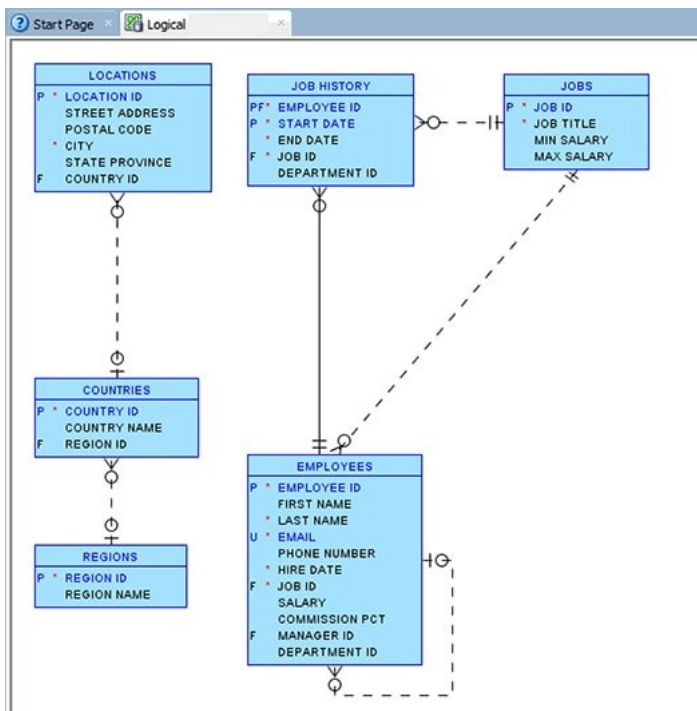


Figure 1: Entity relationship diagram

Getting Started

To work through the example, you need an Oracle Database instance with the sample HR schema that's available in the default database installation. You also need version 4.0 of Oracle SQL Developer, in which you access Oracle SQL Developer Data Modeler through the Data Modeler submenu, as shown in Figure 2. Alternatively, you can use the standalone Oracle SQL Developer Data Modeler. The modeling functionality is identical in the two implementations, and both are available as free downloads from Oracle Technology Network.

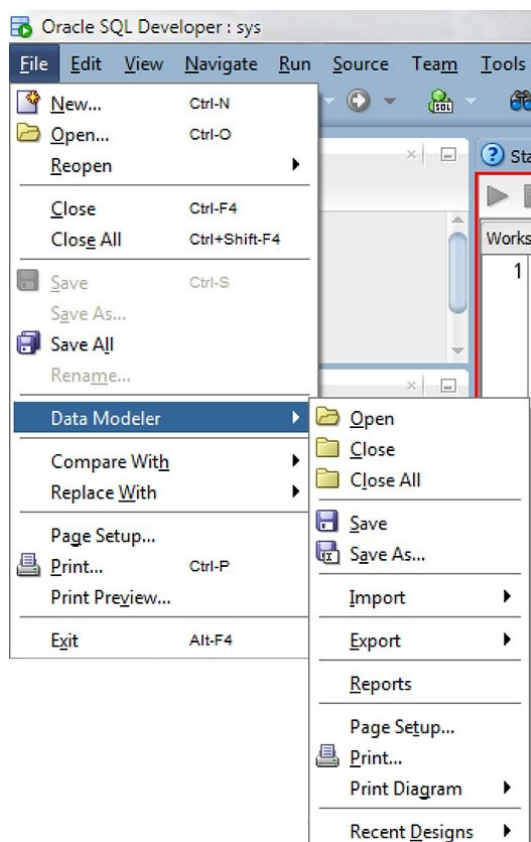


Figure 2: The Data Modeler submenu in Oracle SQL Developer

In Oracle SQL Developer, select **View -> Data Modeler -> Browser**. In the **Browser** panel, select the **Relational Models** node, right-click, and select **New Relational Model** to open a blank model diagram panel. You're now starting at the same place as someone who's using the standalone Oracle SQL Developer Data Modeler.

Importing Your Data Dictionary

A *design* in Oracle SQL Developer Data Modeler consists of one logical model and one or more relational and physical models. To begin the process of creating your design, you must import the schema information from your existing database. Select **File** -> **Data Modeler** -> **Import** -> **Data Dictionary** to open the Data Dictionary Import wizard.

Click **Add** to open the **New** -> **Select Database Connection** dialog box, and connect as the HR user. (For detailed information on creating a connection from Oracle SQL Developer, see ["Making Database Connections,"](#) in the May/June 2008 issue of *Oracle Magazine*.)

Select your connection, and click **Next**. You see a list of schemas from which you can import. Type HR in the **Filter** box to narrow the selection list. Select the checkbox next to **HR**, and click **Next**.

In the **Tables** tab, click the **Select All** toolbar button to select all tables for import, as shown in Figure 3.

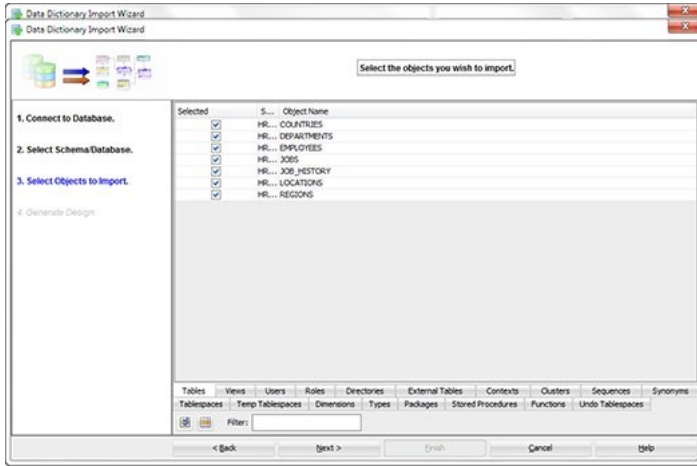


Figure 3: Selecting all tables for import

Now, open the **Views** tab and select the **HR** checkbox. For this exercise, you'll import only the tables and views from the HR schema. For your own applications, you can cycle through each object-type tab and select any objects you want to include in your model and diagrams.

Click **Next**. Oracle SQL Developer Data Modeler summarizes the types and numbers of objects to be imported. Click **Finish** to execute the import and generate the design.

Close the **View Log** and take a look at your new relational model and diagram, which should look similar to Figure 4.

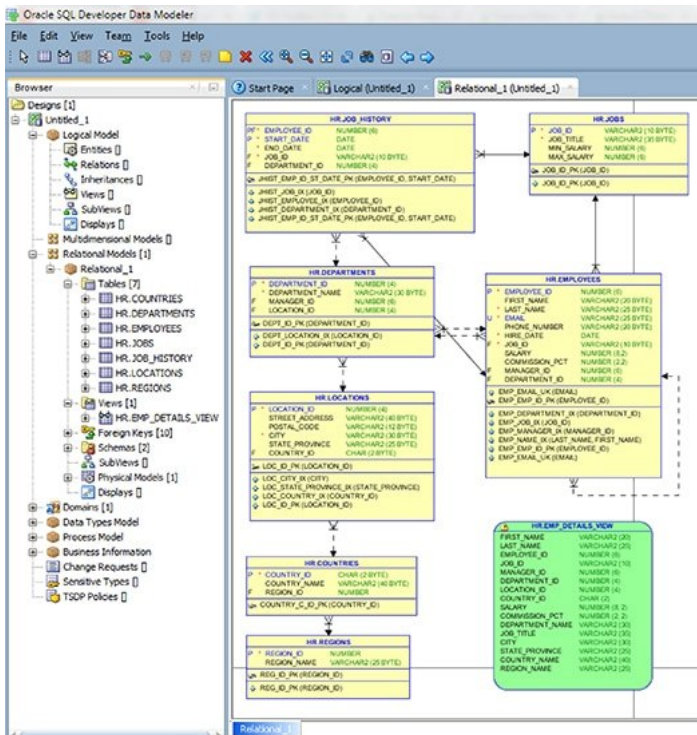


Figure 4: Relational model and diagram for HR

The diagram you've generated is not an ERD. Logical models are higher abstractions. An ERD represents entities and their attributes and relations, whereas a relational or physical model represents tables, columns, and foreign keys. For example, a logical "person" entity might be implemented physically in the database as one or more tables.

To generate a logical model from your relational model, you'll use the Engineer to Logical Model feature of Oracle SQL Developer Data Modeler. Before continuing, save your relational diagram with a name of your choosing (**File -> Data Modeler -> Save**). The same name will be used for the design.

Engineer to Logical Model

In the **Browser** tree, expand the **Relational Models** node, right-click **Relational_1**, and select **Engineer to Logical Model** to open the Engineer to Logical Model dialog box. By default, everything in the relational-model tree (on the left) is brought over to the logical-model tree (on the right) for you. Note that as you select an item in either tree, its details appear in the bottom panel. You can also view the objects in the **Tabular View** tab.

When you are ready to generate your logical model, click **Engineer** and wait for the ERD to appear. You now have a logical model and diagram. Note that the JOBS_HISTORY table is the JOBS HISTORY entity in the logical model.

Before you share your diagram, you can customize its appearance and the information it shows.

Customizing the Diagram

To resize the entity objects to be only large enough to show their names and attributes, right-click in the blank portion of the diagram and select **Resize Objects to Visible**.

You can use your mouse to select, drag, and drop entities and their relationships to wherever you want them to appear. Oracle SQL Developer Data Modeler also offers four automatically generated diagram object layouts. Right-click in the diagram space again, and select one of the **Auto Layout** options.

Oracle SQL Developer Data Modeler supports three styles of logical-design notation:

- Barker (the default)
- Bachman
- Information Engineering

To change the notation style, right-click in the diagram space and select **Bachman Notation**. Note that Bachman notation adds the attribute datatype information to the diagram. Resize the objects to visible, and cycle through the **Auto Layout** options to optimize the spacing and display of the entities again.

Your business users may have conflicting requirements for the look and feel of their diagrams. You can create multiple *displays* in Oracle SQL Developer Data Modeler. Each display is an independent visual presentation of your model diagram. Now you'll create one display for each notation style.

In the Logical Model tree in the **Browser**, right-click the **Displays** node and select **New Display** to create a new diagram display. You can use the buttons below your ERD to toggle between this new display (labeled **Display_1**) and the original diagram (labeled **Logical**). Right-click in **Display_1**, and change the notation style back to Barker.

Expand the **Displays** node in the **Browser** tree, right-click on the **Display_1** node, and select **Properties**. Change the name to Barker, and click **OK**. Repeat the process in the preceding paragraph to create another new display. This time change the notation style to Information Engineering and reflect this change in the display name.

Finally, you'll add a legend that will help document the diagram by showing your OS username, creation date, and other properties of the design. Right-click in the diagram space in any one of the displays, and select **Show -> Legend** to add the legend to your diagram. Drag and drop the legend to an acceptable location in the diagram.

You are now ready to share your ERDs with your colleagues.

Exporting an ERD

You'll export the Information Engineering display to a PDF document. Right-click in the diagram space, and select **Show -> Page Grid** to activate a marker that shows where your PDF page breaks will occur. For readability, make sure that your diagram items don't bleed from one page into another; drag and move them if necessary.

Next use the **Zoom In** button in the Oracle SQL Developer Data Modeler toolbar to resize the diagram to an optimal size. When the entities are easy to see, select **File -> Data Modeler -> Print Diagram -> To PDF File**. Choose a filename and directory location, and click **Save**.

Locate the generated PDF in its saved location. You can view the ERD in a PDF reader, as shown in Figure 5, and share the file with your colleagues.

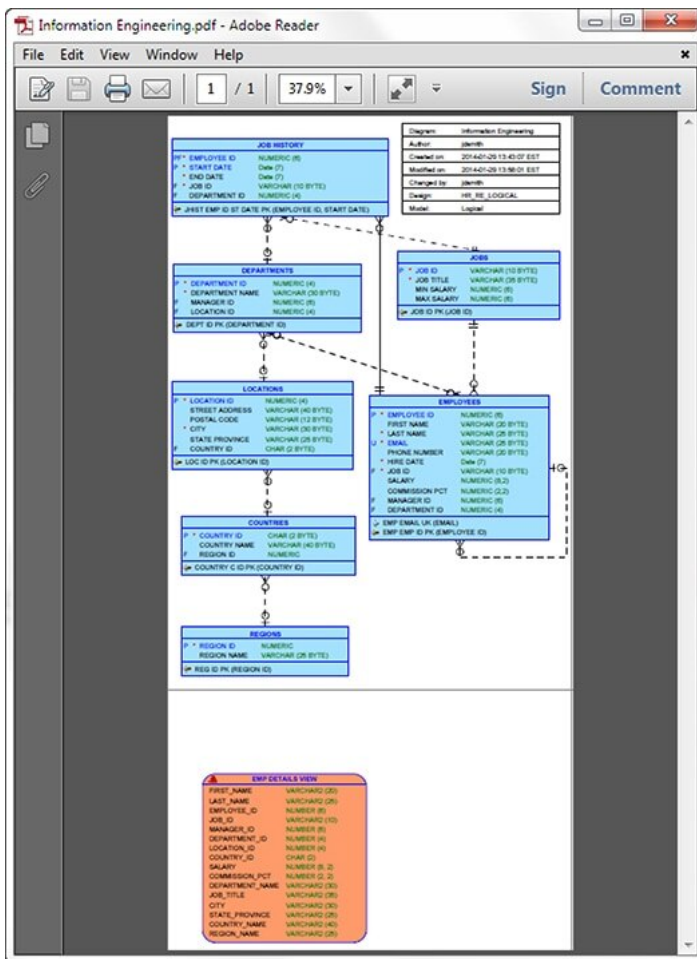


Figure 5: The exported diagram matches the size shown in Oracle SQL Developer Data Modeler.

Conclusion

This column has stepped you through the process of using the Data Dictionary Import wizard and the Engineer to Logical Model feature in Oracle SQL Developer Data Modeler to generate an ERD from existing schema objects. You've seen how the import wizard enables you to choose objects from multiple schemas and object types from an existing database. You can now convert a relational model to a logical model, customize your diagrams, and then share them as PDFs. Oracle SQL Developer Data Modeler makes database design documentation and diagram generation a straightforward task without requiring expensive third-party tools.

Next Steps

LEARN more about

- [Oracle SQL Developer](#)
- [Oracle SQL Developer Data Modeler](#)

DISCUSS

- [Oracle SQL Developer](#)
- [Oracle SQL Developer Data Modeler](#)
- PARTICIPATE in the [Oracle SQL Developer Exchange](#)

DOWNLOAD

- [Oracle SQL Developer](#)
- [Oracle SQL Developer Data Modeler](#)

Photography by Aaron Burson, Unsplash



Jeff Smith

Jeff Smith, a senior principal product manager in Oracle's Database Development Tools group, is responsible for Oracle SQL Developer, Oracle SQL Developer Data Modeler, SQLcl, and Oracle REST Data Services.



Related

DATABASE, SQL AND PL/SQL

Modeling and Accessing Relational Data

Part 2 in a series on the basics of the relational database and SQL

.NET

Distinctive Database Development

Use Entity Framework with Oracle Data Provider for .NET.

IT INNOVATION

Modeling Tables and Compone

Maintain your data-centric business serv diagrams.



Contact Us

About Oracle

Cloud

Events

Top Tasks

News

Key Topics



Integrated Cloud & Platform Services