East West University
Department of Computer Science and Engineering

CSE302: Lab 07
Course Instructor: Md Al-Imran

Entity Relationship Modeling using Data Modeler in SQLDeveloper

Lab Objective

Familiarize students with Entity-Relationship Model.

Lab Outcome

After completing this lab successfully, students will be able to:

1. Understand E-R Model.
2. Understand and use Data Modeler tool for E-R modeling.

Psychomotor Learning Levels

This lab involves activities that encompass the following learning levels in psychomotor domain.

Level	Category	Meaning	Keywords
P1	Imitation	Copy action of another; observe and replicate.	Relate, Repeat, Choose, Copy, Follow, Show, Identify, Isolate.
P2	Manipulation	Reproduce activity from instruction or memory	Copy, response, trace, Show, Start, Perform, Execute, Recreate.

Lab Instructions

Follow the instructor during the class.

You are given lab lecture note based on Oracle Data Modeler documentation as attached material.

2 () Data Modeler Tutorial: Modeling for a Small Database

In this tutorial, you will use Data Modeler to create models for a simplified library database, which will include entities for books, patrons (people who have library cards), and transactions (checking a book out, returning a book, and so on).

This tutorial uses the same entities as for the tutorial provided with the SQL Developer online help. The model is deliberately oversimplified and would not be adequate for any actual public or organizational library. For more advanced tutorials and other materials, see Section 1.11, "For More Information About Data Modeling" (data_modeling.htm#CHDGFCJI).

If the instructions do not mention a particular dialog box, tab, or field, then do not specify anything for it.

This simplified tutorial uses only a subset of the possible steps for the Top-Down Modeling (data_modeling.htm#BABIFGCJ) approach. (For information about the approaches, see Section 1.4, "Approaches to Data Modeling" (data_modeling.htm#BABGFFDB).)

You will perform the following major steps:

1. Develop the Logical Model.
2. Develop the Relational Model.
3. Generate DDL.
4. Save the Design.

()

2.1 Develop the Logical Model

The logical model for the database includes three entities: Books (describes each book in the library), Patrons (describes each person who has a library card), and Transactions (describes each transaction involving a patron and a book). However, before you create the entities, create some domains that will make the entity creation (and later DDL generation) more meaningful and specific.

To start developing the logical model, go to Section 2.1.1, "Adding Domains".

()

Table of Contents

(toc.htm) Oracle SQL Developer Data Modeler User's Guide (toc.htm)
- Preface (preface.htm)
- Data Modeler Concepts and Usage (data_modeling.htm#DMDUG25000)
- Data Modeler Tutorial: Modeling for a Small Database (tut_data_modeling.htm#DMDUG36166)
- Data Modeler Dialog Boxes (dialogs_data_modeling.htm#DMDUG36000)

Download

Categories
- Home (../../index.htm)

2.1.1 Adding Domains

In planning for your data needs, you have determined that several kinds of fields will occur in multiple kinds of records, and many fields can share a definition. For example, you have decided that:

- The first and last names of persons can be up to 25 characters each.
- Street address lines can be up to 40 characters.
- City names can be up to 25 characters.
- State codes (United States) are 2-character standard abbreviations.
- Zip codes (United States postal codes) can be up to 10 characters (*nnnnn-nnnn*).
- Book identifiers can be up to 20 characters.
- Other identifiers are numeric, with up to 7 digits (no decimal places).
- Titles (books, articles, and so on) can be up to 50 characters.

You therefore decide to add appropriate domains, so that you can later use them to specify data types for attributes when you create the entities. (These added domains will also be available after you exit Data Modeler and restart it later.)

1. Click **Tools**, then **Domains Administration**.
2. In the Domains Administration (dialogs_data_modeling.htm#BABBFGAA) dialog box, add domains with the following definitions. Click **Add** to start each definition, and click **Apply** after each definition.

Name	Logical Type	Other Information
Person Name	VARCHAR	Size: 25
Address Line	VARCHAR	Size: 40
City	VARCHAR	Size: 25
State	VARCHAR	Size: 2
Zip	VARCHAR	Size: 10
Book Id	VARCHAR	Size: 20
Numeric Id	NUMERIC	Precision: 7, Scale: 0
Title	VARCHAR	Size: 50

3. When you have finished defining these domains, click **Save**. This creates a file named `defaultdomains.xml` in the `datamodeler/domains` directory or `datamodeler\domains` folder under the location where you installed Data Modeler.

4. Optionally, copy the `defaultdomains.xml` file to a new location (not under the Data Modeler installation directory), and give it an appropriate name, such as `library_domains.xml`. You can then import domains from that file when you create other designs.

5. Click **Close** to close the dialog box.

6. Go to Section 2.1.2, "Creating the Books Entity".

()

2.1.2 Creating the Books Entity

The Books entity describes each book in the library. Create the Books entity as follows:

1. In the main area (right side) of the Data Modeler window, click the Logical tab.

2. Click the New Entity icon.

3. Click in the logical model pane in the main area; and in the Logical pane press, diagonally drag, and release the mouse button to draw an entity box. The Entity Properties (dialogs_data_modeling.htm#BABHACHD) dialog box is displayed.

4. Click **General** on the left, and specify as follows:

   Name: Books

5. Click **Attributes** on the left, and use the Add (+) icon to add the following attributes, one at a time. (For datatypes, select from the Domain types except for Rating, which is a Logical type.)

Name	Datatype	Other Information and Notes
book_id	Domain: Book Id	Primary UID (unique identifier). (The Dewey code or other book identifier.)
title	Domain: Title	M (mandatory, that is, must not be null).
author_last_name	Domain: Person Name	M (mandatory, that is, must not be null).
author_first_name	Domain: Person Name	(Author's first name; not mandatory, but enter it if the author has a first name.)
rating	Logical type: NUMERIC (Precision=2, Scale=0)	(Librarian's personal rating of the book, from 1 (poor) to 10 (great).)

6. Click **OK** to finish creating the Books entity.

7. Go to Section 2.1.3, "Creating the Patrons Entity".

2.1.3 Creating the Patrons Entity

The Patrons entity describes each library patron (that is, each person who has a library card and is thus able to borrow books). Create the Patrons entity as follows:

1. In the main area (right side) of the Data Modeler window, click the Logical tab.

2. Click the New Entity icon.

3. Click in the logical model pane in the main area; and in the Logical pane press, diagonally drag, and release the mouse button to draw an entity box. (Suggestion: draw the box to the right of the Books box.) The Entity Properties (dialogs_data_modeling.htm#BABHACHD) dialog box is displayed.

4. Click **General** on the left, and specify as follows:

   Name: Patrons

5. Click **Attributes** on the left, and use the Add (+) icon to add the following attributes, one at a time. (For datatypes, select from the Domain types, except for location, which uses the structured type SDO_GEOMETRY.)

Attribute Name	Type	Other Information and Notes
patron_id	Domain: Numeric Id	Primary UID (unique identifier). (Unique patron ID number, also called the library card number.)
last_name	Domain: Person Name	M (mandatory, that is, must not be null). 25 characters maximum.
first_name	Domain: Person Name	(Patron's first name.)
street_address	Domain: Address Line	(Patron's street address.)
city	Domain: City	(City or town where the patron lives.)
state	Domain: State	(2-letter code for the state where the patron lives.)
zip	Domain: Zip	(Postal code where the patron lives.)
location	Structured type: SDO_GEOMETRY	Oracle Spatial and Graph geometry object representing the patron's geocoded address.

6. Click **OK** to finish creating the Patrons entity.

7. Go to Section 2.1.4, "Creating the Transactions Entity".

()

2.1.4 Creating the Transactions Entity

The Transactions entity describes each transaction that involves a patron and a book, such as someone checking out or returning a book. Each record is a single transaction, regardless of how many books the patron brings to the library desk. For example, a patron returning two books and checking out three books causes five transactions to be recorded (two returns and three checkouts). Create the Transactions entity as follows:

1. In the main area (right side) of the Data Modeler window, click the Logical tab.

2. Click the New Entity icon.

3. Click in the logical model pane in the main area; and in the Logical pane press, diagonally drag, and release the mouse button to draw an entity box. (Suggestion: Draw the box below and centered between the Books and Patrons boxes.) The Entity Properties (dialogs_data_modeling.htm#BABHACHD) dialog box is displayed.

4. Click **General** on the left, and specify as follows:

   Name: Transactions

5. Click **Attributes** on the left, and use the Add (+) icon to add the following attributes, one at a time. (For datatypes, select from the Domain types, except for transaction_date, which uses a Logical type.)

Attribute Name	Type	Other Information and Notes
transaction_id	Domain: Numeric Id	Primary UID (unique identifier). (Unique transaction ID number)
transaction_date	Logical type: Datetime	M (mandatory, that is, must not be null). Date and time of the transaction.
transaction_type	Domain: Numeric Id	M (mandatory, that is, must not be null). (Numeric code indicating the type of transaction, such as 1 for checking out a book.)

   Note that you do not explicitly define the patron_id and book_id attributes, because these will be automatically added to the Transactions entity after you create relations between the entities (see Section 2.1.5); they will be added as foreign keys when you generate the relational model (see Section 2.2).

6. Click **OK** to finish creating the Transactions entity.

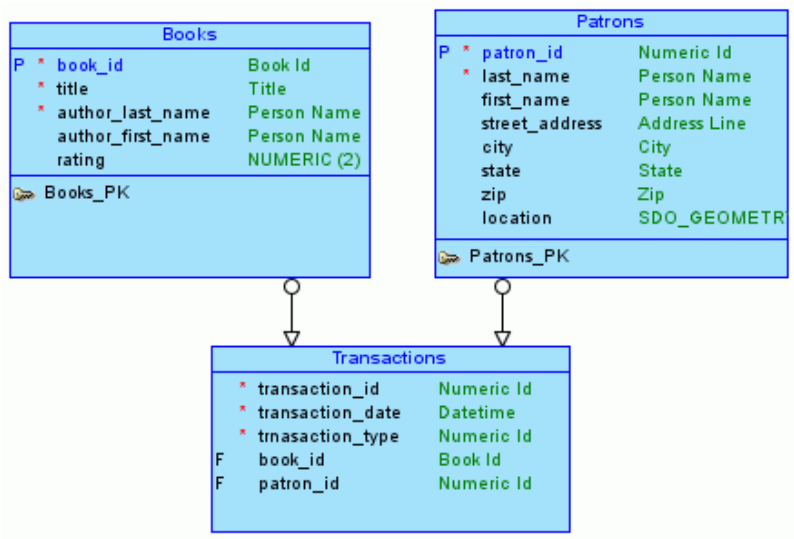
7. Go to Section 2.1.5, "Creating Relations Between Entities".

2.1.5 Creating Relations Between Entities

Relations show the relationships between entities: one-to-many, many-to-one, or many-to-many. The following relationships exist between the entities:

- Books and Transactions: one-to-many. Each book can be involved in multiple sequential transactions. Each book can have zero or one active checkout transactions; a book that is checked out cannot be checked out again until after it has been returned.
- Patrons and Transactions: one-to-many. Each patron can be involved in multiple sequential and simultaneous transactions. Each patron can check out one or many books in a visit to the library, and can have multiple active checkout transactions reflecting several visits; each patron can also return checked out books at any time.

Create the relationships as follows. When you are done, the logical model pane in the main area should look like the following figure. Note that for this figure, Bachman notation is used (you can change to Barker by clicking View, then Logical Diagram Notation, then Barker Notation).



1. In the logical model pane in the main area, arrange the entity boxes as follows: Books on the left, Patrons on the right, and Transactions either between Books and Patrons or under them and in the middle. (If the pointer is still cross-hairs, click the Select icon at the top left to change the pointer to an arrow.)

   Suggestion: Turn off auto line routing for this exercise: right-click in the Logical pane, and ensure that Auto Route is not checked.

2. Click the New 1:N Relation icon.

3. Click first in the Books box, then in the Transactions box. A line with an arrowhead is drawn from Books to Transactions.

4. Click the New 1:N Relation icon.

5. Click first in the Patrons box, then in the Transactions box. A line with an arrowhead is drawn from Patrons to Transactions.

6. Optionally, double-click a line (or right-click a line and select Properties) and view the Relation Properties (dialogs_data_modeling.htm#BABDJCBE) information.

7. Go to Section 2.2, "Develop the Relational Model".

()

2.2 Develop the Relational Model

The relational model for the library tutorial database consists of tables that reflect the entities of the logical model (Books, Patrons, and Transactions) and all attributes of each entity. In the simplified data model for this tutorial, a single relational model reflects the entire logical model; however, for other data models you can create one or more relational models, each reflecting all or a subset of the logical model. (To have a relational model reflect a subset of the logical model, use the "filter" feature in the dialog box for engineering a relational model.)

Develop the relational model as follows:

1. With the logical model selected, click the Engineer to Relational Model icon, or right-click the logical model in the navigator, then select **Engineer to Relational Model**. The Engineering (dialogs_data_modeling.htm#BABGEAGD) dialog box is displayed.

2. Accept all defaults (do not filter), and click **Engineer**. This causes the Relational_1 model to be populated with tables and other objects that reflect the logical model.

3. Expand the Relational Models node in the object browser on the left side of the window, and expand Relational_1 and optionally nodes under it that contain any entries (such as Tables and Columns), to view the objects created.

4. Change the name of the relational model from Relational_1 to something more meaningful for diagram displays, such as Library (relational). Specifically, right-click Relational_1 in the hierarchy display, select **Properties**, in the General pane of the Model Properties - <name> (Relational) (dialogs_data_modeling.htm#BABIHFFE) dialog box specify **Name** as `Library (relational)` , and click **OK**.

5. Go to Section 2.3, "Generate DDL".

()

2.3 Generate DDL

Generate Data Definition Language (DDL) statements that you can use to create database objects that reflect the models that you have designed. The DDL statements will implement the physical model (type of database, such as Oracle Database 11*g*) that you specify.

Develop the physical model as follows:

1. Optionally, view the physical model before you generate DDL statements:

   a. With the relational model selected and expanded, right-click the Physical Models node and select **New**. A dialog box is displayed for selecting the type of database for which to create the physical model.

b. Specify the type of database (for example, Oracle Database 11*g*), and click **OK**. A physical model reflecting the type of database is created under the Physical Models node.

c. Expand the Physical Models node under the Library relational model, and expand the newly created physical model and the Tables node under it, to view the table objects that were created.

Table of Contents

Download

Categories

2. Click **File**, then **Export**, then **DDL File**.

3. Select the database type (for example, Oracle Database 11g) and click **Generate**. The DDL Generation Options (dialogs_data_modeling.htm#BABDHGHC) dialog box is displayed.

4. Accept all defaults, and click **OK**. A DDL file editor is displayed, with SQL statements to create the tables and add constraints. (Although you can edit statements in this window, do not edit any statements for this tutorial exercise.)

5. Click **Save** to save the statements to a .sql script file (for example, `create_library_objects.sql`) on your local system.

   Later, run the script (for example, using a database connection and SQL Worksheet in SQL Developer) to create the objects in the desired database.

6. Click **Close** to close the DDL file editor.

7. Go to Section 2.4, "Save the Design".

()

2.4 Save the Design

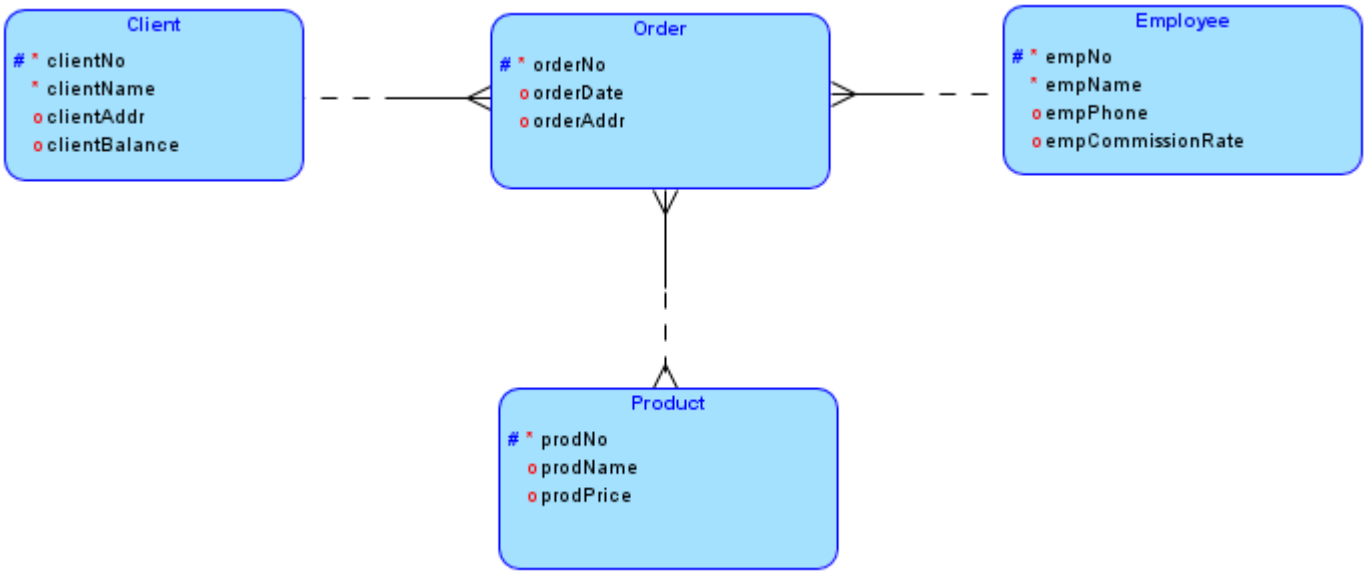
Save the design by clicking **File**, then **Save**. Specify the location and name for the XML file to contain the basic structural information (for example, `library_design.xml`). A directory or folder structure will also be created automatically to hold the detailed information about the design, as explained in Section 1.3.1, "Database Design" (data_modeling.htm#BABCFHIH).

Continue creating and modifying design objects, if you wish. When you are finished, save the design again if you have made any changes, then exit Data Modeler by clicking **File**, then **Exit**.

You can later open the saved design and continue working on it, as explained in Section 1.6, "Saving, Opening, Exporting, and Importing Designs" (data_modeling.htm#BABFAFEG).

Exercise:

Draw the following ER Diagram



- Create **four entities** with appropriate attributes first.
- Create the following **three relationships**:
    - **Client to Order: one to many.** A client can give many orders. An order can be placed by one client.
    - **Employee to Order: one to many.** An employee can process many orders. An order is processed by one employee.
    - **Product to Order: many to many.** A product can be included in many orders. An order may have many products.