Oracle Data Modeling and Relational Database Design

Activity Guide

D56497GC10

Edition 1.0

September 2010

D67009

ORACLE®

Author

Marcie Young

Technical Contributors and Reviewers

Sue Harper
Philip Stoyanov
Nancy Greenberg
Rick Green
Brian Pottle
Anjula Subbiahpillai
Gerry Jurrens
Nick Donatone
David Lapoint
Tom Provenzano
Mike Ritz
Tim Trauernicht
Zhicheng Xu
Ron Berry
David Lyons
Kim Bell
Maria Billings
Steve Friedberg
Bryan Roberts
Priyanka Sharma
Matthew Gregory
Angelika Krupp

Editors

Daniel Milne
Vijayalakshmi Narasimhan

Graphic Designer

Rajiv Chandrabhanu

Publishers

Shaik Basha
Jayanthy Keshavamurthy

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS
The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

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

Appendix A
Practices and Solutions

Table of Contents

Practices for Lesson 1

Practice 1-1: Identify the Modeling Approach

In this practice, you review the following case study and identify the models that need to be developed and what approach should be used.

Starlight DVD and Games Rentals is a successful company that rents DVDs and games. Last year's turnover almost doubled to $6.5 million with a profile margin of just over 10%, which is high in this type of business. The number of rentals increased 80% and the customer base tripled to 30,000. The number of available movie titles grew by 8%, reflecting the increase in titles coming out on DVD. The number of copies in the stores increased by 45% to 63,000. The renting of games is still an immature business (less than 5% of turnover) and the number of game titles is steady as new games replace those removed from the catalog.

Starlight already has an information system to track rentals, but this needs enhancement to support proposed changes to the business. A team of business consultants has carried out a study and produced a report outlining proposals. The following is an excerpt from the report:

- Starlight attracts new customers easily and the number of rentals is growing rapidly. The customer base is not stable, however, which is a cause for concern. We suggest implementing some new concepts that should attract new customers and retain existing customers.
- The main idea is to introduce the concept of membership at Starlight. Members will pay a membership fee (the amount to be decided) that entitles them to certain privileges that are valid for a period of time (length to be decided). Initially, there will be three types of membership (gold, silver, and bronze) although more may be introduced later. Bronze membership is free and replaces the current way of handling rentals. Silver and gold memberships incur a fee but entitle the member to privileges, such as a discount on the usual rental price for a type of title, a free DVD after a certain number of rentals, the right to take out more than the usual number of DVDs at the same time, and the right to take out more "new release" titles than usual.
- In the current situation, only individuals ("customers") can rent. It is proposed to extend membership to companies as well.
- A membership will be of a certain type, last for a specified period, and will be renewed periodically. The type of membership cannot be changed. If someone wants to upgrade from a bronze to a silver membership, the bronze membership is terminated and a silver membership is created. Members will receive an automatic renewal or upgrade offer about a month before the membership is due to expire if they have rented a tape or game during the previous four weeks. Bronze memberships that do not lead to a rental during the rental period are terminated automatically. Silver and gold members receive a membership card only after paying the appropriate fee. Members can terminate their membership by request at any time.
- Another cause for concern is the number of "lost" DVDs. We propose some new procedures to address this problem. Every customer seeking membership will have to provide proof of identity as part of the registration process. A photocopy of the identification will be stored. A membership card, containing the membership number,

type, expiration date, and a photograph of the holder, will be provided. This card will have to be shown for every rental transaction.

- In addition, the store manager will be able to produce a weekly report identifying DVDs that are more than two weeks overdue. Members who have the DVDs will be requested to return them immediately. The request will be repeated until the DVDs are returned. Members who fail to return DVDs, or who repeatedly break the rules by keeping DVDs for two weeks or more, will have their membership terminated.
- Starlight management emphasizes that all amendments to the information system must be flexible enough to deviate from fees and time frames in individual cases.

Practices for Lesson 2

Practice 2-1: Identify Types of Business Direction Information

In this practice, you gain experience in identifying the types of business direction information. Read through each of the following notes and mark in the table the type of business direction information that it is (business objective, key performance indicator, critical success factor, or problem).

Note	Business Objective	Key Performance Indicator	Critical Success Factor	Assumption	Problem
Store clerks do not know when a membership is due for renewal, and cannot, therefore, proactively encourage customers to renew their membership.					
To reduce the number of non-returned DVDs from 5% to <1%					
Staff needs to be alerted when a DVD is more than two weeks overdue.					
Store clerks cannot easily identify DVDs that are seriously overdue (more than two weeks late).					
To increase rentals by 25% annually through the introduction of membership privileges					
Our current system uses Oracle Database 9*i* and is on UNIX.					
The number of seriously overdue DVDs should be <3%.					
Starlight must stock a very wide range of DVDs and sufficient copies of the latest DVDs to ensure that membership is attractive.					
Store clerks are unable to chase overdue DVDs.					
Starlight would like to upgrade to Oracle Database 11*g* on Linux.					

Practices for Lesson 3

Practice 3-1: Create a Data Flow Diagram

In this practice, you build a data flow diagram for the Starlight DVD Rental case study. As stated previously, the business consultants have recommended the introduction of a tiered membership scheme for Starlight. This necessitates some changes to the way the company deals with its customers. The following extract from the consultants' report describes the proposals regarding the creation of memberships:

- There will be three types of memberships: gold, silver, and bronze. Gold and silver memberships incur a fee, but entitle the member to certain privileges. Bronze memberships replace the current situation.
- Customers must provide some form of identification before a membership is created.
- The identification document will be photocopied and filed.
- A membership card will be issued and must be shown each time a DVD or game is rented.
- A unique number is allocated to each new membership
- The fee for gold and silver membership must be paid before the membership card can be issued.

Perform the following in groups assigned by your instructor:

1. Create processes, external agents, information stores, and information flows.
2. Label all objects on your diagram.
3. Define the events that trigger each process.
4. Define the outcome of each process and specify its type.

Practices for Lesson 4

Practice 4-1: Build a Data Flow Diagram in Oracle SQL Developer Data Modeler

In this practice, you use Oracle SQL Developer Data Modeler to build the data flow diagram that you created in Practice 3-1.

Practices for Lesson 5

Practice 5-1: Decompose a Process in Your Data Flow Diagram

In this practice, you decompose a process and add a transformation process in the DFD that you created in the previous practice. Decompose the Gather Membership Information process to handle the following requirements:

- Verify that the customer has entered all the required information on the form.
- Determine whether the customer is a current customer, and compare whether the customer is upgrading the membership or staying at the same Bronze level.

Create a transformation process that loads the membership information into your CRM system so that CRM activities can be used to communicate with customers.
Incorporate the above requirements by revising your DFD to show the decomposition in Oracle SQL Developer Data Modeler.

Practices for Lesson 6

Practice 6-1: Identify Entities and Attributes

In this practice, you identify and model the entities and attributes for Starlight DVD Rental. Write a brief description of each entity. Show at least two attributes for each entity.

For your convenience, here is a summary of part of the business analyst's report:

- Starlight Rentals wants to introduce memberships (Gold, Silver and Bronze).
- A membership can be for an individual customer or an organization.
- An organization may be a part of another organization
- Starlight Rentals must keep track of the reasons why memberships terminate.
- The actual fee that is paid by a member can be less than the standard fee.
- The type of membership cannot be modified. An upgrade always leads to termination of the current membership and creation of a new one.
- Memberships can be renewed annually
- A membership can start at any time and is valid for a period of a year.
- Silver and Gold memberships entitle members to privileges.
- One privilege is a type-dependent discount percentage of each rental.

Practice 6-2: Identify Entities and Attributes

In this practice, you identify and model the entities in the following set of information requirements:

District Hotel Manager, Sharon Ferguson, would like to manage the information that her company keeps about their hotels, guests, and rooms. Sharon is responsible for multiple hotels in various locations. Sharon would like to know what rooms she has in which hotels and what the price is for each room. She would also like to know which rooms have been reserved for a particular date by a guest.

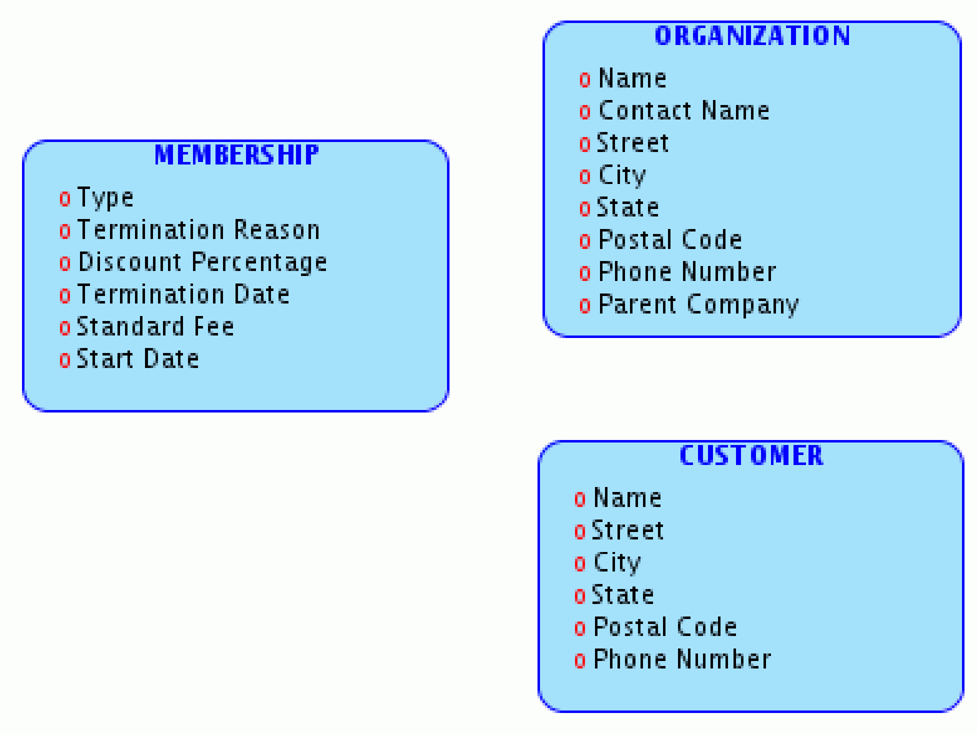
Perform the following tasks:

1. Examine the nouns. Are they things of significance?
2. Name each entity.
3. Is there information of interest about the entity that the business needs to hold?
4. Diagram each entity and its attributes.

Practices for Lesson 7

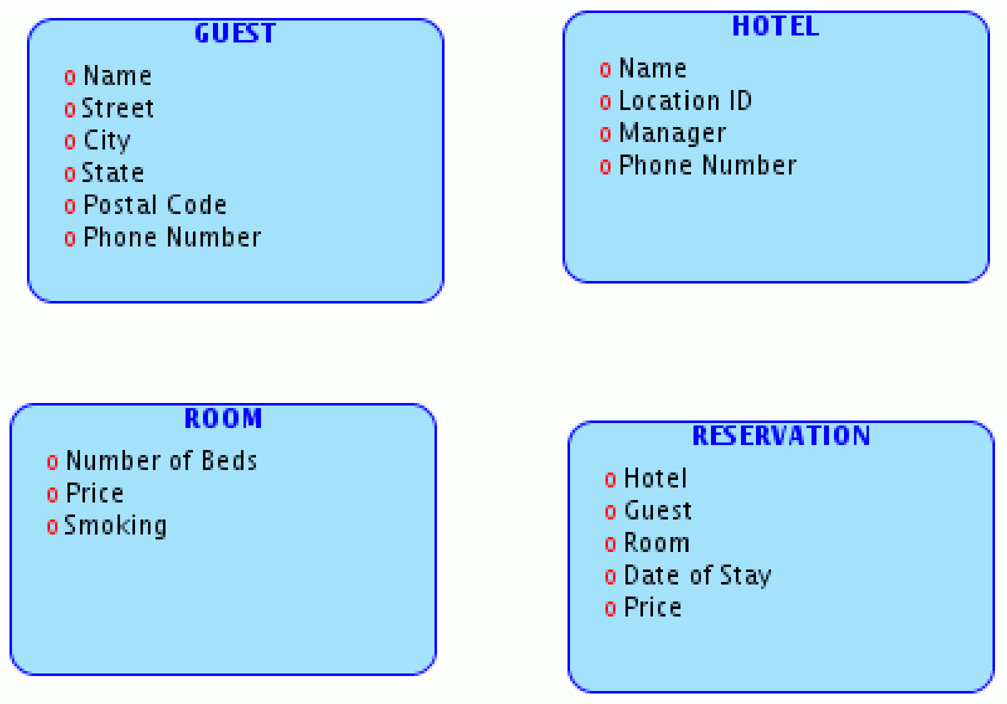
Practice 7-1: Analyze and Model Relationships

In this practice, you analyze and model the relationships for the following entities, which you created in the DVD Membership practice (Practice 6-1). Use a relationship matrix to track the existence of relationships between the entities, and then draw the relationships in the diagram.



Practice 7-2: Analyze and Model Relationships

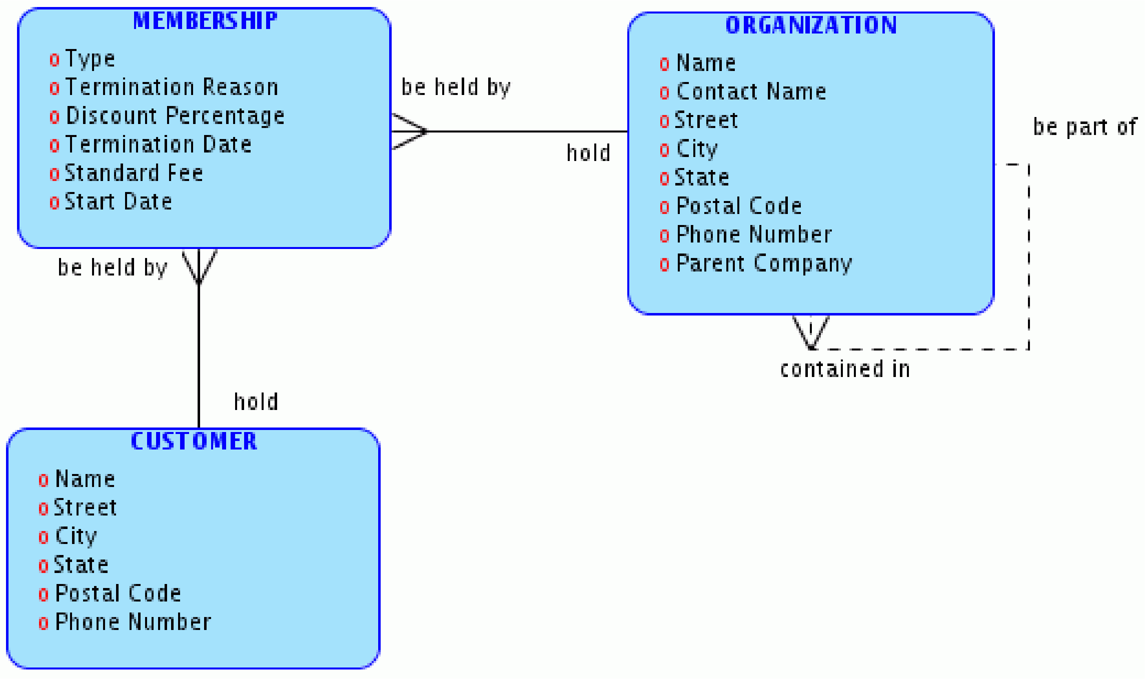
In this practice, you analyze and model the relationships for the following entities, which you created in Practice 6-2. Use a relationship matrix to track the existence of relationships between entities, and then draw the relationships in the diagram.



Practices for Lesson 8

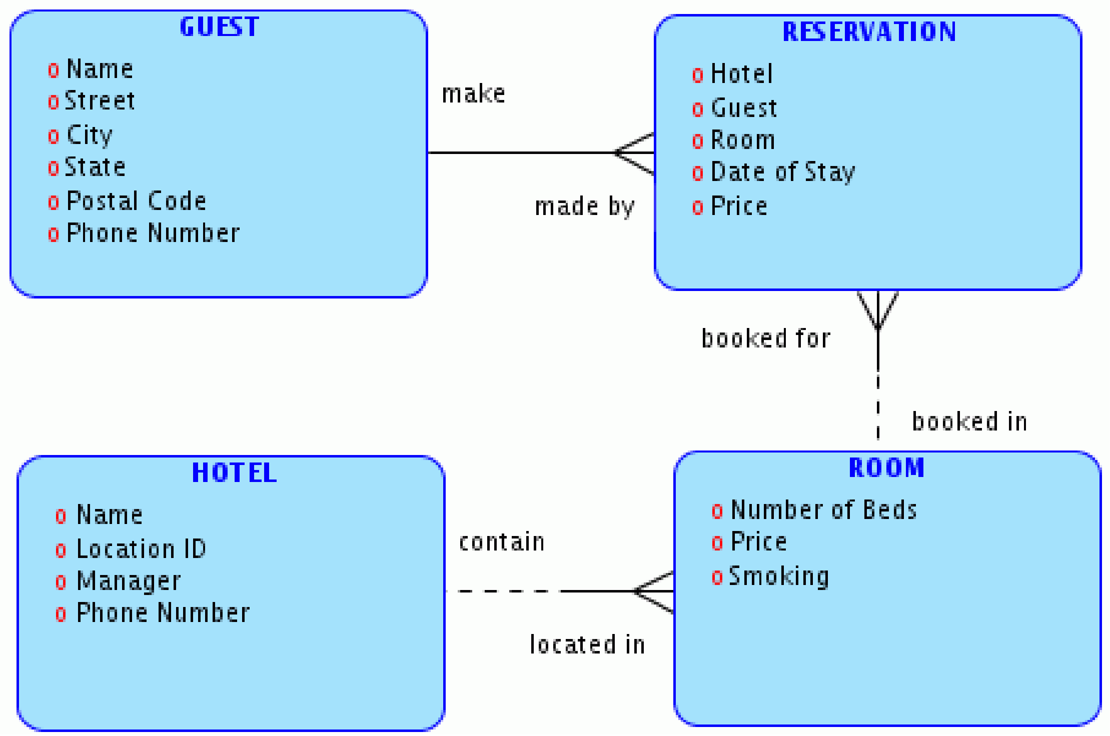
Practice 8-1: Identify Unique Identifiers

In this practice, you identify the unique identifiers that you created in the DVD Membership practice (Practice 7-1).



Practice 8-2: Identify Unique Identifiers

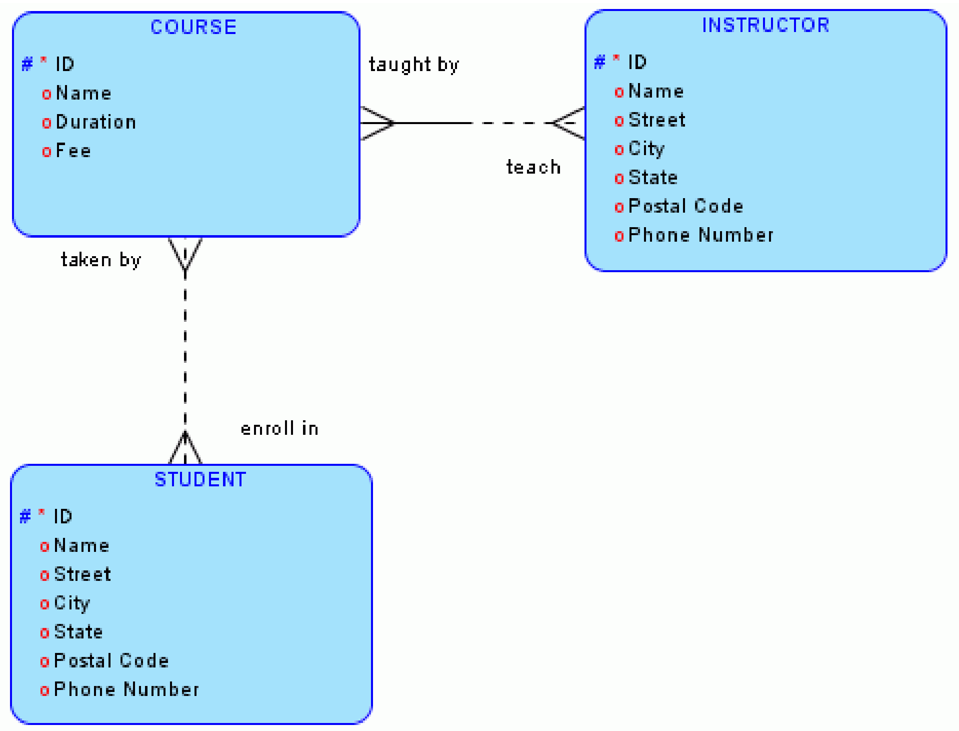
In this practice, you identify unique identifiers for the ERD you created in Practice 7-2.

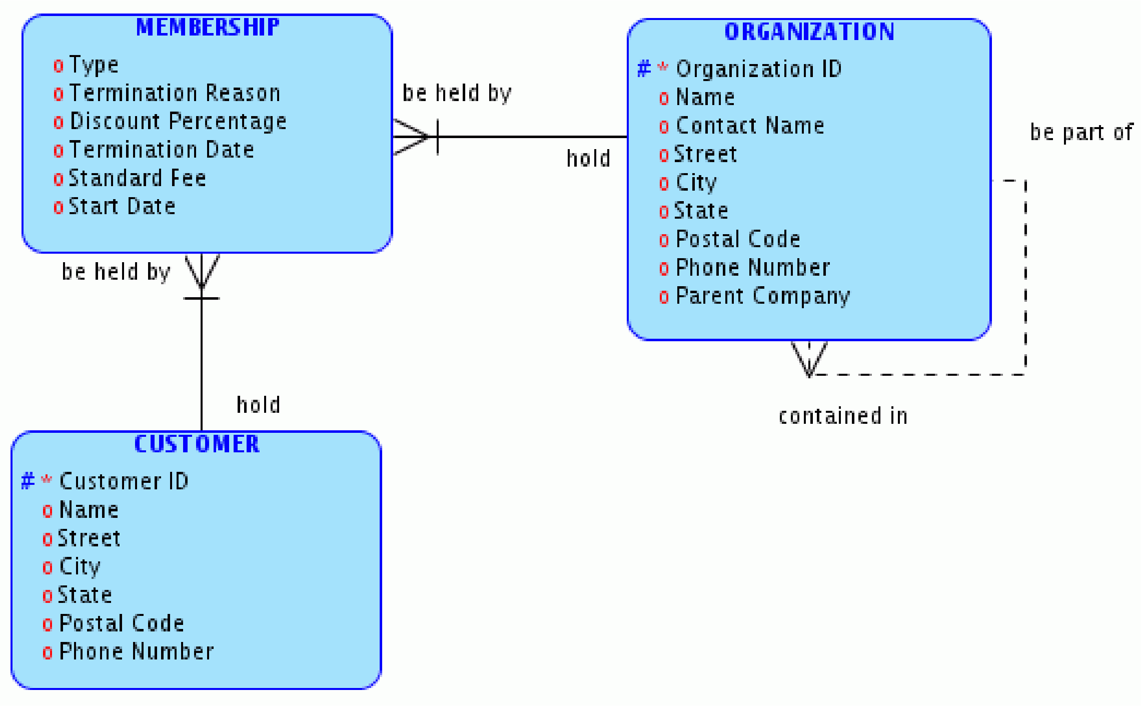


Practices for Lesson 9

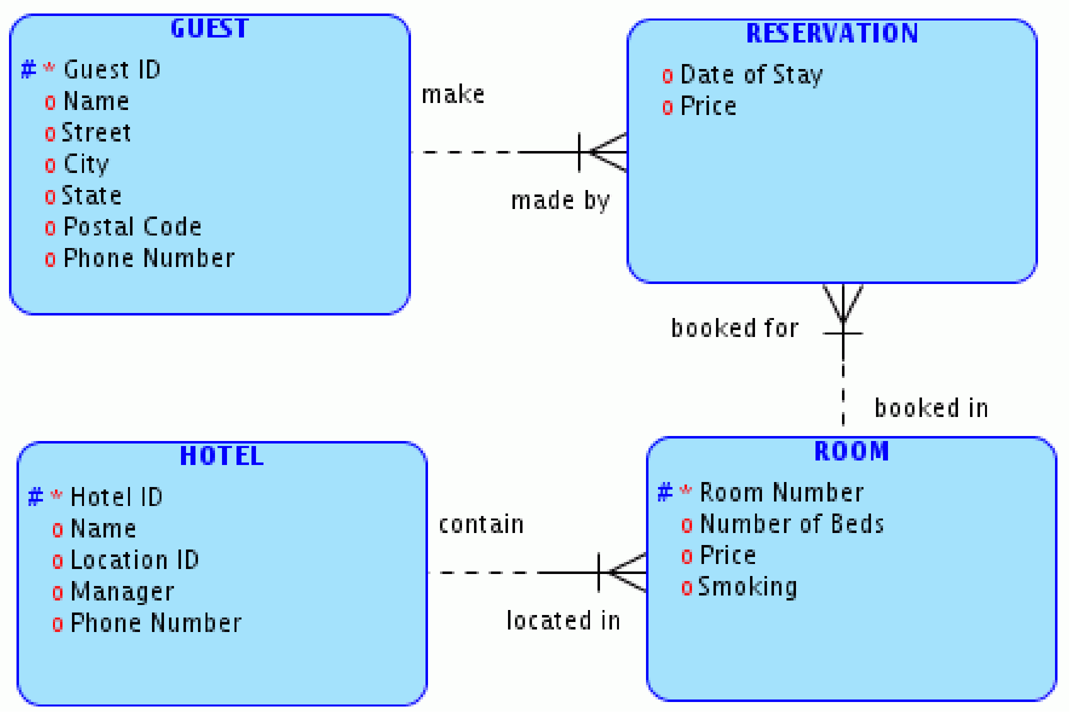
Practice 9-1: Build an ERD in Oracle SQL Developer Data Modeler

In this practice, you build the following ERDs in Oracle SQL Developer Data Modeler. Build a subview and display for each ERD.





Practice 9-1: Build an ERD in Oracle SQL Developer Data Modeler (continued)



Practices for Lesson 10

Practice 10-1: Develop and Validate Your ERD

In this practice, you develop and validate an ERD in Oracle SQL Developer Data Modeler using the following information requirements:

The Midwest Oracle User's Group has grown to include over 200 members. This is a volunteer organization and its records are a mess. They need to automate their membership records. For each member, the following information must be kept: the member's name, title, mailing address, office phone number, type of membership (individual or corporate), and whether or not the member is current on dues. The dues are collected on a yearly basis, and everyone's dues are due in January.

The User's Group would also like to know what company a member works for. Keeping this information current is difficult because members are always changing companies; however, only the current employer of the member must be tracked. Members come from many different companies, including Coors, EG&G, and Storage Tech. A few members are unemployed. For each company, the following information must be kept: the company name, address, and type of business. There are a standard set of business code types. Only the main company address for each company is needed.

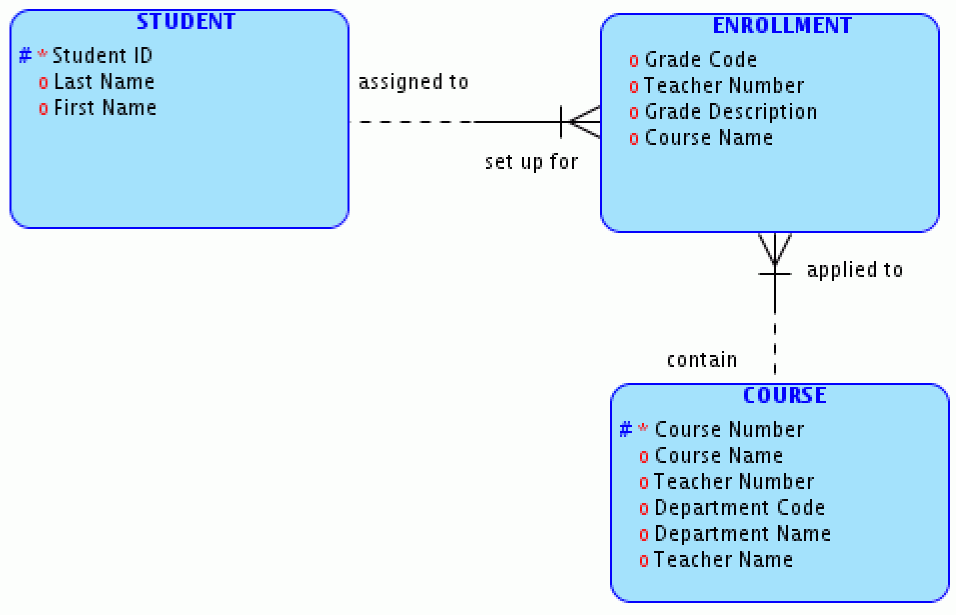
The User's Group holds various events during the year, and they want to track information about each event. Some of their annual events include the September Meeting, the November Meeting, the annual Training Day in January, and their April Meeting. They also hold special events each year. For example, they held a special Data Modeling day last May where an Oracle representative came and spoke about Data Modeling concepts. They hold their events at several locations around town, including at AT&T, at Redrocks Community college, and at D.U. Bank. They want to track each event's date, an optional description of the event, number of attendees, where it was held, how much money they spent on it, and any comments on the event. They treat all comments as if they came from anonymous submitters. A set of comments is just a free form text statement of any length. They number each set of comments, and they frequently get multiple sets of comments for an event.

The User's Group tracks which members attended which events. Some members are very active, and others attend very infrequently or just enjoy receiving the newsletter.
In addition to the above, the User's Group must track what type of computer platform each member uses. The User's Group has a unique, three-digit system identification for each type of platform. For example, 001 is for XP, 002 is for Vista, 003 is for Linux, and 004 is for UNIX.
The User's Group would also like to track which application areas each member is interested in (for example, accounting, human resources, oil and gas, pharmaceuticals, and health systems). The applications should be portable, so the User's Group does not need to know which platforms they run on.

Practices for Lesson 11

Practice 11-1: Normalize an ERD

For the following ERD, evaluate each entity against the rules of normalization, identify the misplaced attributes, and explain which rule of normalization each misplaced attribute violates. Then redraw the diagram in Oracle SQL Developer Data Modeler.



Practice 11-2: Validate ERD for Normalization

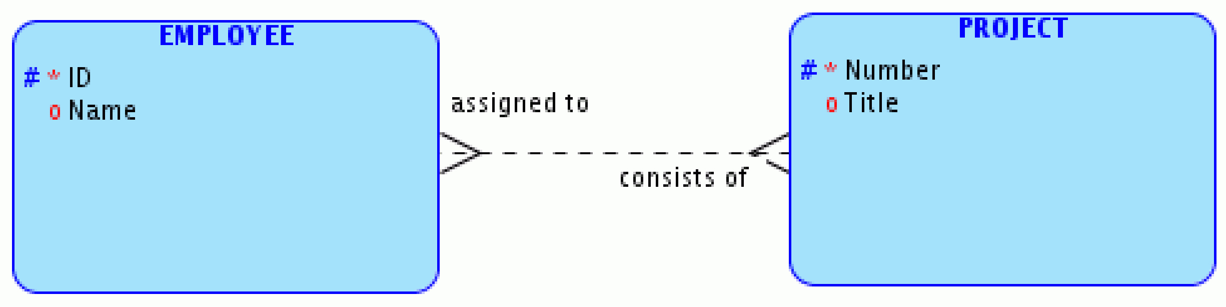
For the following ERD, evaluate the following unnormalized data and develop an entity relationship diagram in SQL Developer Data Modeler that is normalized to third normal form.

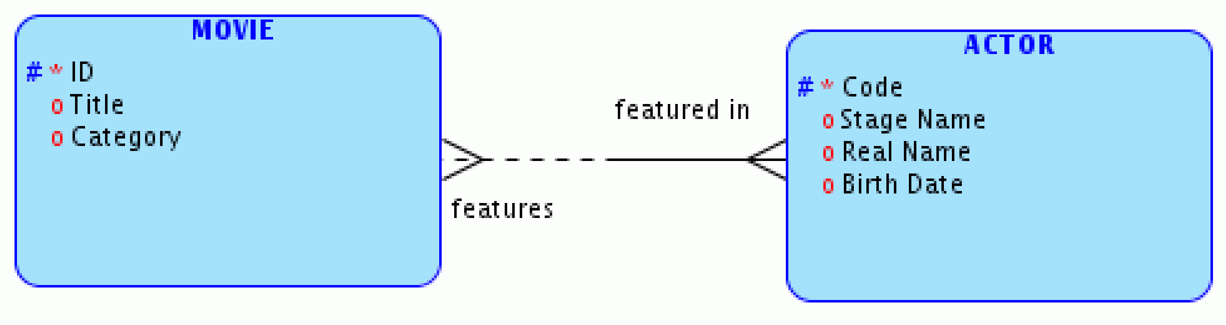
User ID	User Name	Message ID	Recorded Date	Subject	Text	Server ID	Server Name
2301	Smith	54101	05/07	Meeting Today	There is…	3786	IMAP05
2301	Smith	54098	07/12	Promotions	I like to…	3786	IMAP05
2301	Smith	54445	10/06	Next Assignment	Your next…	3786	IMAP05
5607	Jones	54512	06/07	Lunch?	Can You…	6001	IMAP08
5607	Jones	54101	05/07	Meeting Today	There is…	6001	IMAP08
5607	Jones	54660	12/01	Jogging Today?	Can you…	6001	IMAP08
7773	Walsh	54101	05/07	Meeting Today	There is…	9988	EMEA01
7773	Walsh	54554	03/17	Stock Quote	The latest…	9988	EMEA01
0022	Patel	54101	05/07	Meeting Today	There is…	9988	EMEA01
0022	Patel	54512	06/07	Lunch?	Can you…	9988	EMEA01

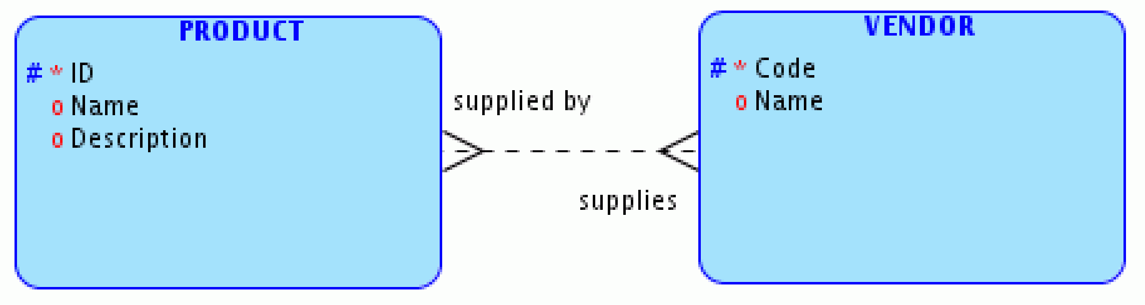
Practices for Lesson 12

Practice 12-1: Resolve M:M Relationships

In this practice, you resolve the following M:M relationships. Add additional attributes in the intersection entity. Perform the work in Oracle SQL Developer Data Modeler. Start in a new logical model.







Practice 12-2: Model Hierarchical Data

In this practice, model the entities, relationships, attributes, and unique identifiers for the hierarchy of a hotel. The hotel has many floors, many suites on each floor, and many rooms within each suite.

Practice 12-3: Model Hierarchical Data and Recursive Relationships

Develop two ERDs to represent the following situation. Develop one as a hierarchical structure and one as a recursive structure. Put your model into Oracle SQL Developer Data Modeling.

Curves Dynamics sells products throughout the United States. They are divided into four major sales regions: the Northern, Eastern, Southern, and Western regions. Each sales region has a unique region code. Each sales region is then divided into sales districts. For example, the Western Region is divided into the Rocky Mountain, Northwest, Pacific Coast, and Pacific districts. Each district has a unique district code.

Each district is made up of sales territories. The Rocky Mountain district is composed of three territories: Wyoming-Montana, Colorado, and Utah-New Mexico. The Northwest district is made up of two territories: the Washington and Oregon-Idaho territories. The Pacific Coast district is composed of two territories: the California and Nevada territories. The Pacific district includes the Hawaii territory and the Alaska territory. Each territory has a unique territory code.

Each sales territory is broken down into sales areas. For example, Colorado is made up of two sales areas: the Front Range and the Western Slope sales areas. Each sales area has a unique sales-area code.

Each salesperson is responsible for one or more sales areas, and has a specific sales quota. Each sales manager is responsible for one or more sales districts and sales directors who are responsible for one or more sales regions. Each sales manager is responsible for the territories within his districts. Employees' responsibilities do not overlap. A sales area is always the responsibility of a single salesperson, and managers and directors' responsibilities do not overlap. Sometimes salespersons, managers, and directors will be on leave or special assignments and will not have sales area responsibilities. All sales personnel are identified by their employee IDs.

Practice 12-4: Examine Exclusive Relationships

Determine how each of the following exclusive relationships should be modeled and then create them in Oracle SQL Developer Data Modeler.

1. ABC Insurance Company would like to keep track of information on its insurance policies. A policy must be either for a property or for a vehicle. Each policy has a policy number, premium, and renewal date. If it is a property policy, the property ID, type, and value must be stored. If it is a vehicle policy, the vehicle identification number (VIN), year, model, and make must be stored.

2. The XYZ Corporation would like to keep track of information about their Commercial versus Private customers. Both types of customer have some attributes in common, such as ID, name, and address. However, XYZ Corporation wants to store some attributes for Commercial customers, such as State Tax ID and Account Number, that they do want to store for Private customers. And for Private customers, they want to store the customers' Social Security numbers (SSNs) and credit card numbers, which they do not need for Commercial customers. A customer can only be a Commercial customer or a Private customer, not both.

Practice 12-5: Examine Exclusive Relationships

Develop an ERD for the following information requirements:

The Right-Way Rental Truck Company rents small moving trucks and trailers for local and one-way usage. There are 347 rental offices across the western United States. The rental inventory includes a total of 5,750 vehicles, including various types of trucks and trailers. The data that needs to be tracked is rental agreements and vehicle assignments. Each rental office rents vehicles that they have in inventory, to customers ready to take possession of the vehicle. Reservations are not taken, and speculation on when the customer will return the rented vehicles is not tracked. The central office oversees the vehicle distribution, and directs transfers of vehicles from one rental office to another.

Each rental office has an office name like “Madison Right-Way” and address. Each office also has a unique three-digit office number. Each office is a home office for some vehicles, and each vehicle is based out of a single home office.

Each vehicle has a vehicle ID, state of registration, and a license plate registration number. There are five different types of vehicles: 36-foot trucks, 24-foot trucks, 10-foot trucks, 8-foot covered trailers, and 6-foot open trailers, each with a type code. For all vehicles, a last maintenance date and expiration date of its registration needs to be tracked. In addition, for trucks, the current odometer reading, the gas tank capacity, and whether or not it has a working radio needs to be stored. For long moves, customers really prefer a radio. The current mileage is logged before the truck is rented, and then again when it is returns.

Most rental agreements are for individual customers, but a rental agreement can be either for an individual or company. A small percentage of trucks is rented to companies. Each company is assigned a company number and the name and address of the company is tracked. The corporate sales group handles all the information separately.

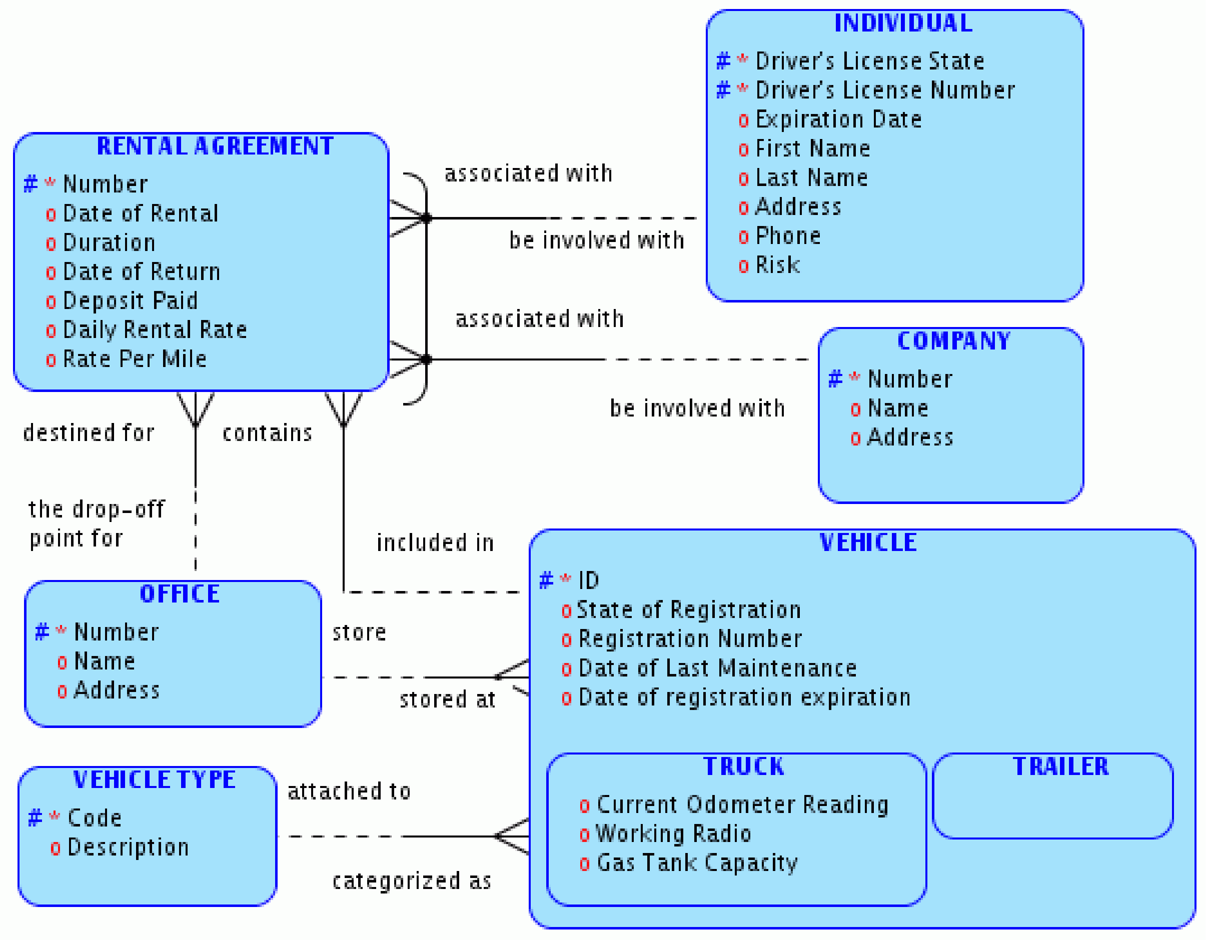
For each individual customer, the following information is tracked: name, home phone, address, and driver’s license state, number and expiration date. If a customer damaged a vehicle, abandoned it, or did not fully pay the bill, the customer is tagged as a poor risk, and the customer may not rent again.

Only a single individual or company can obtain a rental agreement, and a separate rental agreement is written for each vehicle. Customers can rent two or more vehicles at the same time. Each rental agreement is identified by the originating rental office number and a rental agreement number. In addition, the rental date, anticipated duration of the rental, the originating rental office, the drop-off rental office, the amount of the deposit paid, the quoted daily rental rate, and the quoted rate per mile is tracked. For trailers, there is not a mileage charge.

Practices for Lesson 13

Practice 13-1: Create and Assign Data Types

In this practice, you define the data types for each attribute in the following model. Create a data type model with structured types and then use them within your data model for addresses. Create two domains for id_6 and name_50 and assign them to appropriate attributes in your model. Run Design Rules to make sure that all attributes that have a data type defined.



Practices for Lesson 14

Practice 14-1: Develop and Validate Your ERD

In this practice, you develop and validate an ERD in Oracle SQL Developer Data Modeler using the following information requirements:

Cascade Cruises has decided that their manual system of booking passengers onto their ships will not hold up when they get a new one next month. They currently have two ships (not including the new one) and will probably expand to five or six by 2012. They are named “Goodsea” and “Goodwind,” and the new one will be called “Goodsky.” Each ship has a specific passenger capacity and registry. Registry is the country that the ship is registered with. They do not need to worry about tonnage or draft or anything else about the ship.

Each year Cascade puts out a brochure with information about their cruises. Every cruise has a name and duration in number of days. They offer three, seven, eleven, and fourteen day cruises. Each cruise also has a specific ship assigned to it; some people want to go on only the newer ships. For each cruise, there are different port stops. A three day cruise will have only one stop, always on the second day of the cruise; a seven day cruise will have three port stops. Cascade varies port stops depending on where the cruise originates. For example, the Los Angeles, CA cruises go to Mexico ports like Cabo San Lucas and Acapulco, the Miami cruises go to the Bahamas and the Virgin Islands, and the Anchorage cruises make stops all through Alaska. Depending upon the length of each cruise, the cruise will make port calls on different days out.

Passengers book a given cruise, which has a certain length and number of ports. Depending on the cruise that they pick, customers are informed of the available cabins. After the passenger chooses a cabin, they can get a price. The price also depends on the number of people in the cabin and the “class” of the cabin. After the cabin is booked, it is then removed from the list of available cabins unless the passenger indicated that they want to share with someone else. If the cabin can hold four people, and they are traveling alone, then the price will be cheaper if they share. After passengers are booked and a deposit is received, the travel agent who made the reservation will receive the commission for the cruise.

Practice 14-2: Develop and Validate Your ERD (Optional)

In this practice, you develop and validate an ERD in Oracle SQL Developer Data Modeler using the following information requirements:

Larry Craig, a senior partner at Craig and Associates, a large diversified law firm, handles a wide variety of cases including traffic violations, domestic disputes, civil suits, and homicide cases.

The firm has just hired a database administrator to organize and track various data became the firm grew faster than expected and they have so many cases to manage documents for.

The firm is made up of departments such as criminal, misdemeanor, and civil. Each case is assigned a particular department for administrative purposes. Attorneys are also assigned to a particular department, but this is only for billing/payroll purposes because an attorney can work on cases in other departments.

A history of the case is maintained that includes a log of events and the date that the event becomes effective. Cases have to be identifiable by a unique number, which appears on the list with every event date and event description. Events have special codes like 011 Meet with client, 012 Status conference, or 013 Preliminary Trial. Every case must have an event status.

The firm would like to keep track of important information associated with a case, including the department to which it is assigned and a brief description. After a case has been closed, it may be reopened at some future date. A new case number should be assigned to reopened cases, but the previous case number should be maintained so that the history of a case can be maintained.

The firm would like to keep track of what role a particular party plays in the context of a particular case. The kinds of people that may be involved in cases include judges, eyewitnesses, defendants, and attorneys. For example, the firm is working for the defendant on a murder case. One attorney is assigned to the case, there is a judge presiding over the case, and there is an eyewitness. As a result, there are four people who are party to the case. Parties should be identified by a unique ID, name, and date of birth. A given party can serve in different roles in different cases, but a party can only serve in one role on a given case.

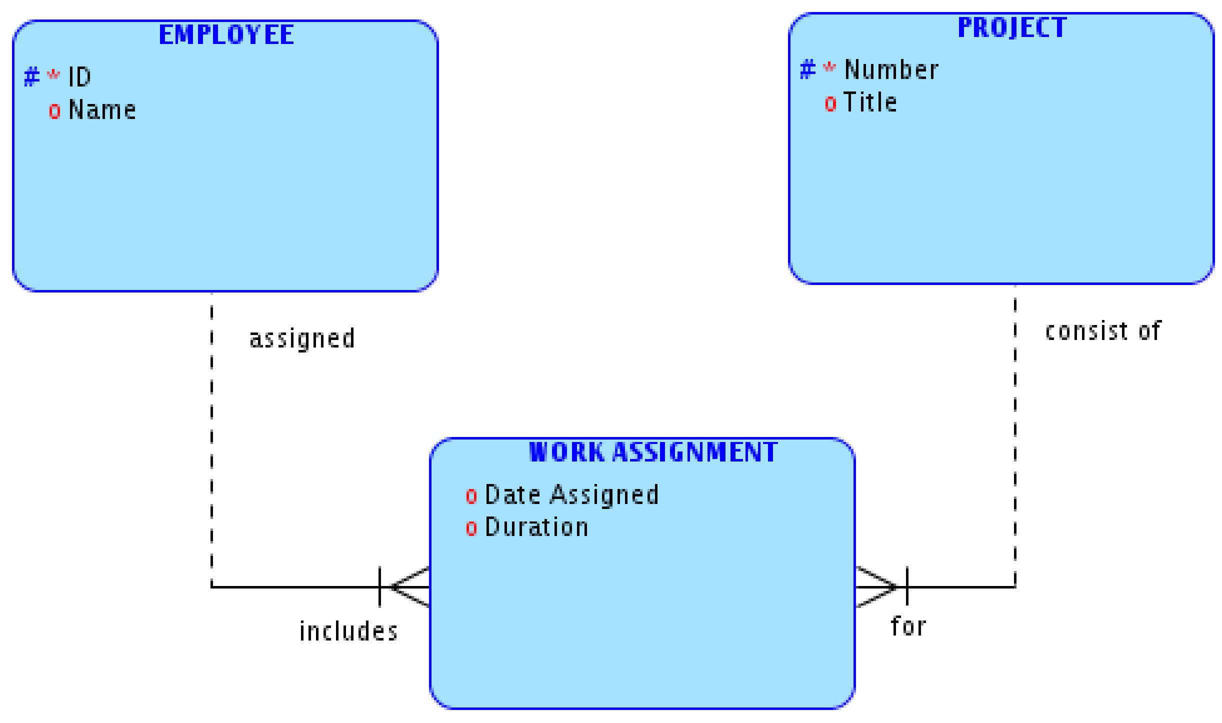
Practices for Lesson 15

Practice 15-1: Create an Initial Relational Model

For the following ERD, perform the following tasks:

1. Make sure that all attributes have a logical data type assigned.
2. Create a glossary with abbreviations for all the words in the model.
3. Associate the glossary with the model.
4. Add a short name and preferred abbreviation for each entity.
5. Run Design Rules to make sure that you do not have any errors. (Warnings are OK.)
6. Engineer the model to a relational model.
7. Change the Name template for primary keys and foreign keys to have PK and FK as prefixes rather than suffixes.
8. Add a prefix called OU_ for table names and apply it to the model.

NOTE: You can open `sol12-1a.xml` as the starting point for this practice.

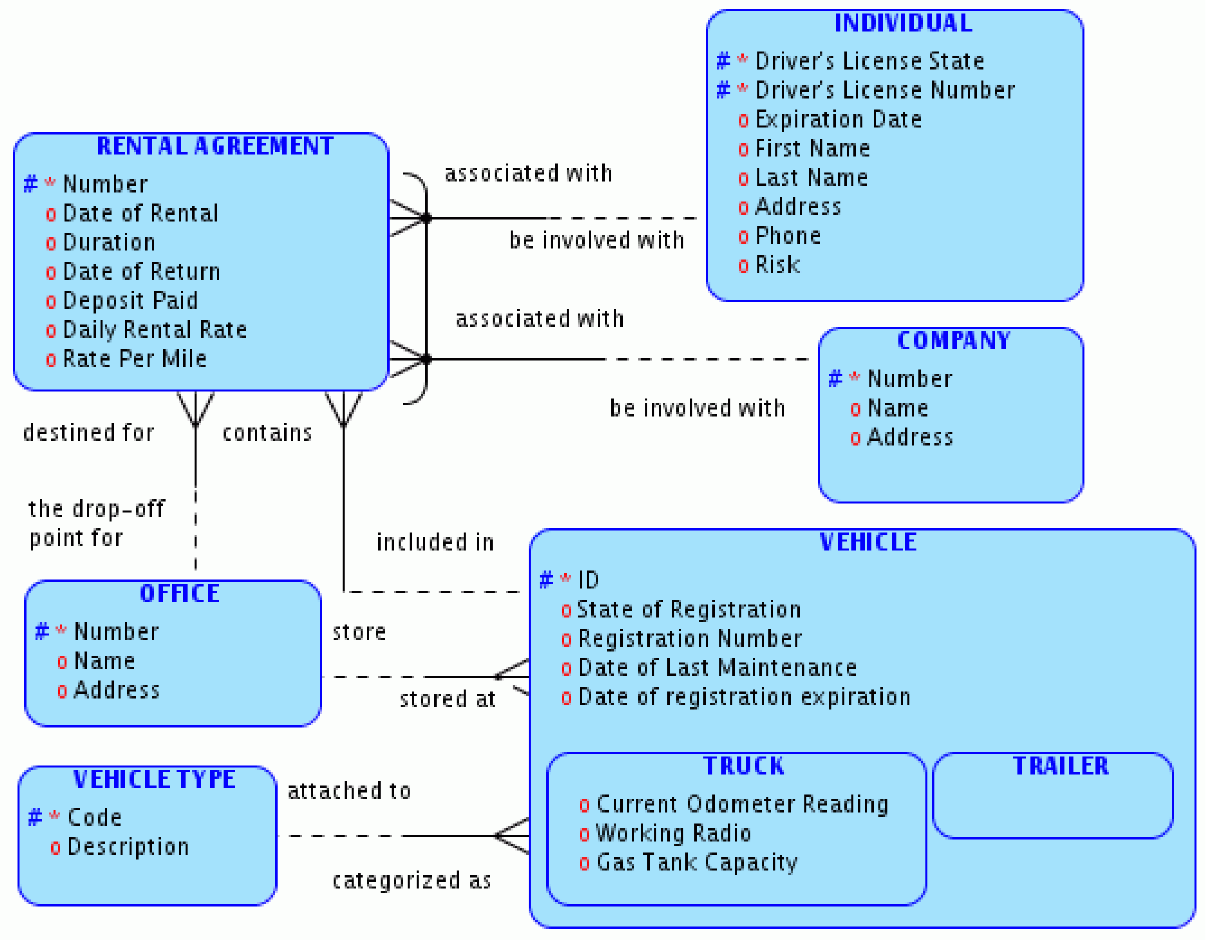


Practice 15-2: Forward Engineer a Model

For the following ERD that you created in Practice 13-1, perform the following tasks:

1. Review the data types assigned to each attribute. Note that this is the same model that you created domains and a data type model for.
2. Run Design Rules to make sure that you do not have any errors. (Warnings are OK.)
3. Make sure that the FWD engineering strategy is set to Single Table for the entity type hierarchy.
4. Engineer the model to a relational model.
5. Review the results.
6. Create another relational model.
7. Change the engineering strategy for the entity type hierarchy to Table for each entity. Engineer the model again. What differences do you see?

Hint: Open `sol13-1.xml` as your starting point or open the model that you created in Practice 13-1.

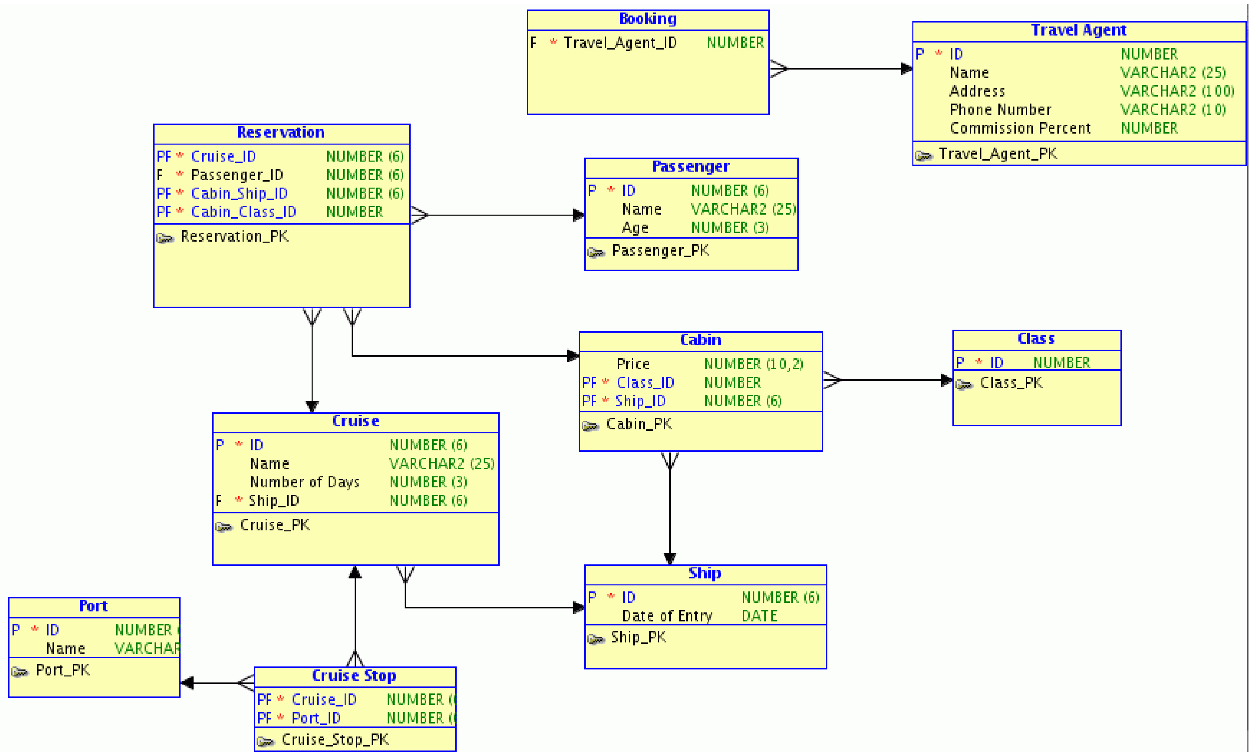


Practices for Lesson 16

Practice 16-1: Analyze Your Relational Model

For the following relational model, which is based on the Cascade Cruise case study in Practice 14-1, add or modify existing design components based on the following requirements.

A. Want to know how many passengers were on a particular cruise for each month
B. Want to quickly see the cruises that a particular ship has made
C. Want to know how well each cruise did as far as revenue
D. Want to know the total commission that each travel agent made
E. Want to quickly see the average age of passengers on a particular cruise



Practices for Lesson 17

Practice 17-1: Denormalize Your Relational Model

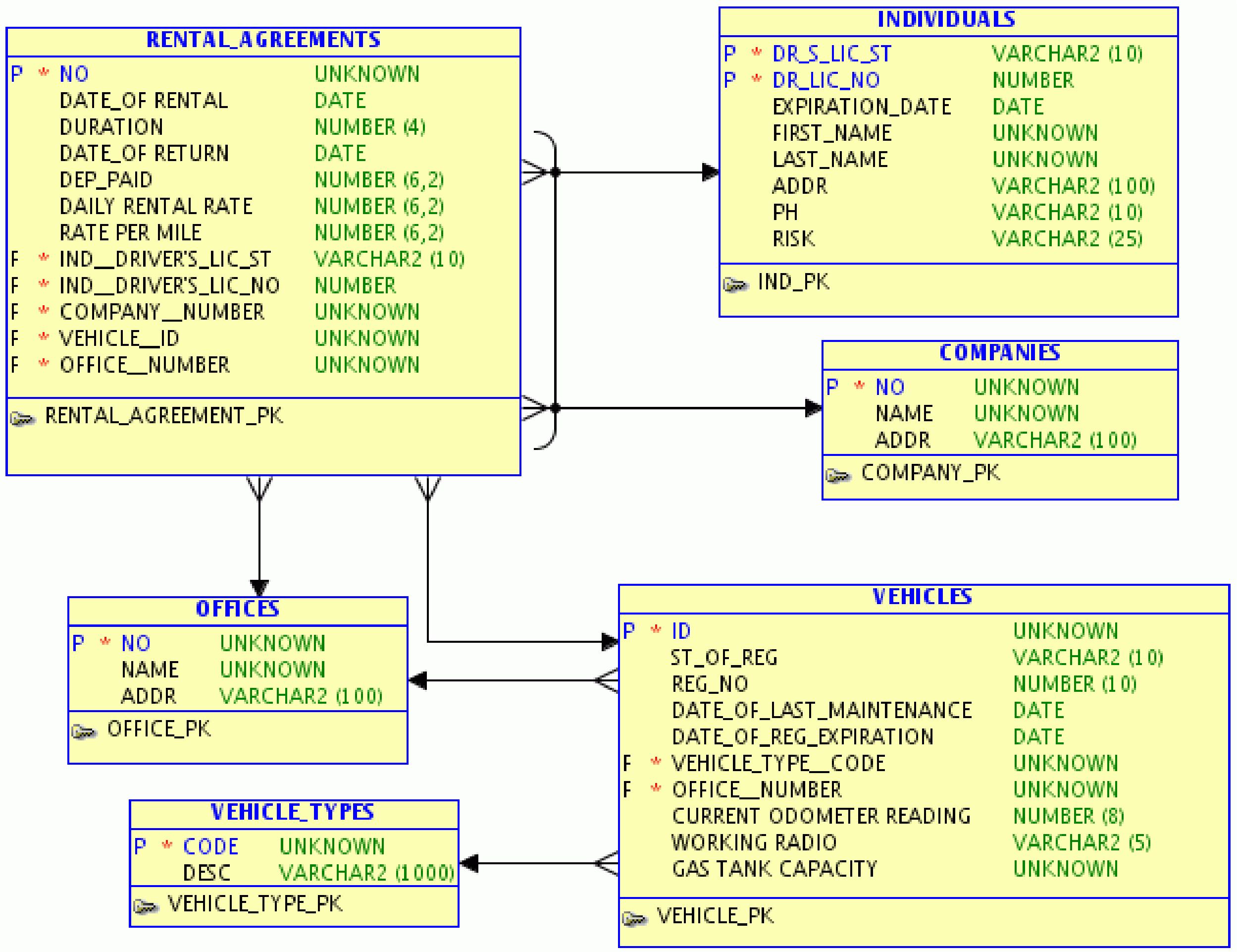
For the following relational model, denormalize the model based on the following set of requirements:

A. Vehicle types are relatively static (compact, full-size, SUV, minivan, etc) and you want to validate a list of vehicles by type.

B. You frequently want to find out which rental agreements have been made by which offices.

C. You want to keep a history of which vehicles have been rented. However, you want to be able to query which vehicles have been rented between a set of dates.

D. You want to keep track of the total amount for a rental paid by an individual or a company.



RENTAL_AGREEMENTS			
P	*	NO	UNKNOWN
		DATE_OF RENTAL	DATE
		DURATION	NUMBER (4)
		DATE_OF RETURN	DATE
		DEP_PAID	NUMBER (6,2)
		DAILY RENTAL RATE	NUMBER (6,2)
		RATE PER MILE	NUMBER (6,2)
F	*	IND__DRIVER'S_LIC_ST	VARCHAR2 (10)
F	*	IND__DRIVER'S_LIC_NO	NUMBER
F	*	COMPANY__NUMBER	UNKNOWN
F	*	VEHICLE__ID	UNKNOWN
F	*	OFFICE__NUMBER	UNKNOWN
		RENTAL_AGREEMENT_PK	

INDIVIDUALS			
P	*	DR_S_LIC_ST	VARCHAR2 (10)
P	*	DR_LIC_NO	NUMBER
		EXPIRATION_DATE	DATE
		FIRST_NAME	UNKNOWN
		LAST_NAME	UNKNOWN
		ADDR	VARCHAR2 (100)
		PH	VARCHAR2 (10)
		RISK	VARCHAR2 (25)
		IND_PK	

COMPANIES			
P	*	NO	UNKNOWN
		NAME	UNKNOWN
		ADDR	VARCHAR2 (100)
		COMPANY_PK	

OFFICES			
P	*	NO	UNKNOWN
		NAME	UNKNOWN
		ADDR	VARCHAR2 (100)
		OFFICE_PK	

VEHICLE_TYPES			
P	*	CODE	UNKNOWN
		DESC	VARCHAR2 (1000)
		VEHICLE_TYPE_PK	

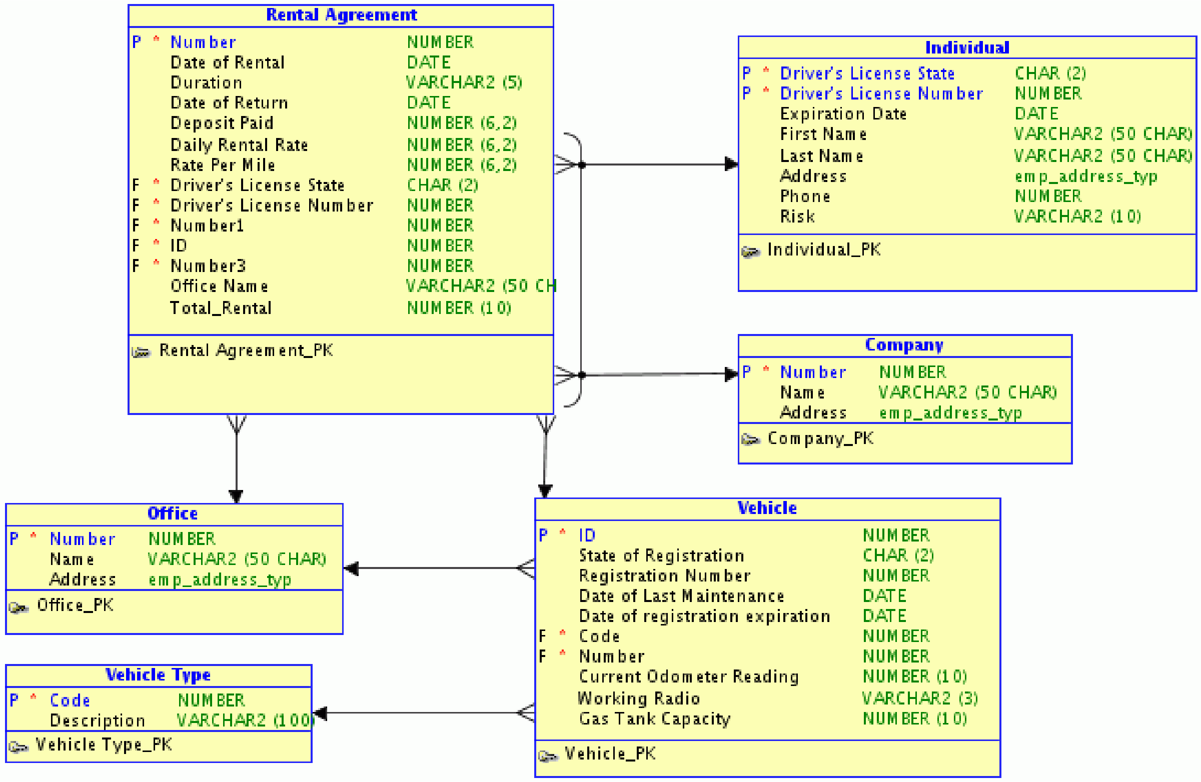
VEHICLES			
P	*	ID	UNKNOWN
		ST_OF_REG	VARCHAR2 (10)
		REG_NO	NUMBER (10)
		DATE_OF_LAST_MAINTENANCE	DATE
		DATE_OF_REG_EXPIRATION	DATE
F	*	VEHICLE_TYPE__CODE	UNKNOWN
F	*	OFFICE__NUMBER	UNKNOWN
		CURRENT ODOMETER READING	NUMBER (8)
		WORKING RADIO	VARCHAR2 (5)
		GAS TANK CAPACITY	UNKNOWN
		VEHICLE_PK	

Practices for Lesson 18

Practice 18-1: Create a Physical Model

In this practice, you perform the following:

1. Create a physical model for the Rental Agreement practice (in Practice 17-1) for Relational_1.
2. Change the name of Relational_1 to Vehicle Rentals.
3. Create a tablespace and a user for the model.
4. Add a partition to the Rental Agreement table so that you can access the rental agreements for a given month very quickly.



Rental Agreement

P	*	Number	NUMBER
		Date of Rental	DATE
		Duration	VARCHAR2 (5)
		Date of Return	DATE
		Deposit Paid	NUMBER (6,2)
		Daily Rental Rate	NUMBER (6,2)
		Rate Per Mile	NUMBER (6,2)
F	*	Driver's License State	CHAR (2)
F	*	Driver's License Number	NUMBER
F	*	Number1	NUMBER
F	*	ID	NUMBER
F	*	Number3	NUMBER
		Office Name	VARCHAR2 (50 CH
		Total_Rental	NUMBER (10)

Rental Agreement_PK

Individual

P	*	Driver's License State	CHAR (2)
P	*	Driver's License Number	NUMBER
		Expiration Date	DATE
		First Name	VARCHAR2 (50 CHAR)
		Last Name	VARCHAR2 (50 CHAR)
		Address	emp_address_typ
		Phone	NUMBER
		Risk	VARCHAR2 (10)

Individual_PK

Company

P	*	Number	NUMBER
		Name	VARCHAR2 (50 CHAR)
		Address	emp_address_typ

Company_PK

Office

P	*	Number	NUMBER
		Name	VARCHAR2 (50 CHAR)
		Address	emp_address_typ

Office_PK

Vehicle

P	*	ID	NUMBER
		State of Registration	CHAR (2)
		Registration Number	NUMBER
		Date of Last Maintenance	DATE
		Date of registration expiration	DATE
F	*	Code	NUMBER
F	*	Number	NUMBER
		Current Odometer Reading	NUMBER (10)
		Working Radio	VARCHAR2 (3)
		Gas Tank Capacity	NUMBER (10)

Vehicle_PK

Vehicle Type

P	*	Code	NUMBER
		Description	VARCHAR2 (100

Vehicle Type_PK

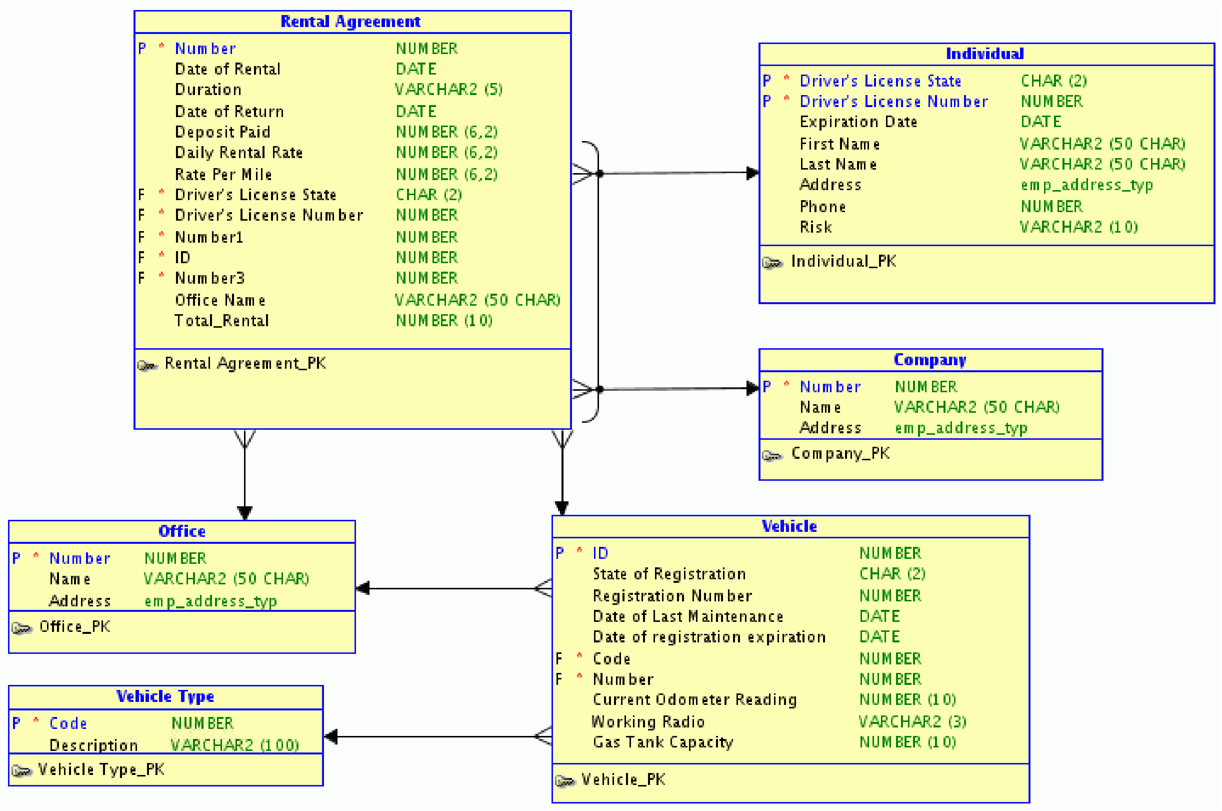
Practices for Lesson 19

Practice 19-1: Generate DDL

In this practice, you generate the DDL for your database by using the model from the last practice (Practice 18-1).

Keep the following in mind:

- Every table must have a prefix of `V_`.
- If there are any errors, make appropriate changes to the model and regenerate the DDL.



Rental Agreement

P	*	Number	NUMBER
		Date of Rental	DATE
		Duration	VARCHAR2 (5)
		Date of Return	DATE
		Deposit Paid	NUMBER (6,2)
		Daily Rental Rate	NUMBER (6,2)
		Rate Per Mile	NUMBER (6,2)
F	*	Driver's License State	CHAR (2)
F	*	Driver's License Number	NUMBER
F	*	Number1	NUMBER
F	*	ID	NUMBER
F	*	Number3	NUMBER
		Office Name	VARCHAR2 (50 CHAR)
		Total_Rental	NUMBER (10)

Rental Agreement_PK

Individual

P	*	Driver's License State	CHAR (2)
P	*	Driver's License Number	NUMBER
		Expiration Date	DATE
		First Name	VARCHAR2 (50 CHAR)
		Last Name	VARCHAR2 (50 CHAR)
		Address	emp_address_typ
		Phone	NUMBER
		Risk	VARCHAR2 (10)

Individual_PK

Company

P	*	Number	NUMBER
		Name	VARCHAR2 (50 CHAR)
		Address	emp_address_typ

Company_PK

Office

P	*	Number	NUMBER
		Name	VARCHAR2 (50 CHAR)
		Address	emp_address_typ

Office_PK

Vehicle Type

P	*	Code	NUMBER
		Description	VARCHAR2 (100)

Vehicle Type_PK

Vehicle

P	*	ID	NUMBER
		State of Registration	CHAR (2)
		Registration Number	NUMBER
		Date of Last Maintenance	DATE
		Date of registration expiration	DATE
F	*	Code	NUMBER
F	*	Number	NUMBER
		Current Odometer Reading	NUMBER (10)
		Working Radio	VARCHAR2 (3)
		Gas Tank Capacity	NUMBER (10)

Vehicle_PK

Practices for Lesson 20

Practice 20-1: Re-Engineer the HR Schema

Perform the following tasks in Oracle SQL Developer Data Modeler:

1. Import all the tables in the HR schema from the data dictionary.
2. Reverse engineer to create a Logical Model.
3. Add the DEPENDENTS entity with the following attributes: ID, Name, and Birthdate. Create a relationship between EMPLOYEES and DEPENDENTS. ID is the unique identifier. You may import the `domains.xml` file from the solutions directory, and use the domains in the file as the datatypes for some of your new attributes.
4. Add COST CENTER to the DEPARTMENTS entity.
5. In the EMPLOYEES entity, move HIRE_DATE above the EMAIL attribute.
6. Create a 1:M relationship between EMPLOYEES and DEPENDENTS.
7. Forward engineer to a new relational model.
8. Compare the relational model with what is currently in the database
   Hint: Run the import from the data dictionary.
9. Preview the DDL. Were the correct ALTER statements generated?
10. Generate some data modeler reports to make sure that there are no violations and that the metadata looks good.
    Hint: Create a reporting user in Oracle SQL Developer and export the model to a reporting schema in Oracle SQL Developer Data Modeler.

Practices for Lesson 21

Practice 21-1: Build a Multidimensional Model

In this practice, you build a multidimensional model. Perform the work in Oracle SQL Developer Data Modeler:

1) Import `sh_cre_all.sql` to create the relational model.
2) Examine the dimensions in the physical model.
3) Reverse engineer the relational model to create the logical model.
4) Engineer the Oracle model to create the multidimensional model.
5) Examine the results.

Solutions for All Lessons

Solution 1-1: Identify Modeling Approach

There is no one right answer. You must evaluate the existing rental tracking database to determine whether it would be better to start with a new model or re-engineer the database that already exists.

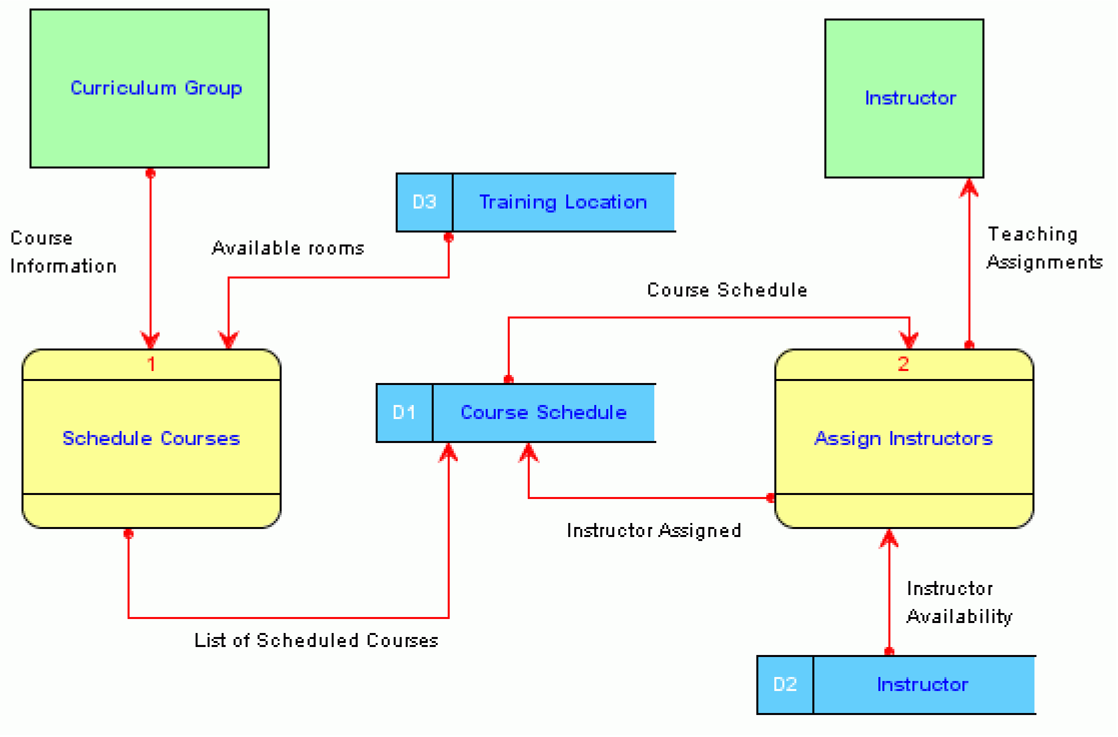
Solution 2-1: Identify Types of Business Direction Information

One solution for this practice is as follows:

Note	**Business Objective**	**Key Performance Indicator**	**Critical Success Factor**	**Assumption**	**Problem**
Store clerks do not know when a membership is due for renewal, and cannot, therefore, proactively encourage customers to renew their membership.					X
To reduce the number of non-returned DVDs from 5% to <1%.	X				
Staff needs to be alerted when a DVD is more than two weeks overdue.			X		
Store clerks cannot easily identify DVDs that are seriously overdue (more than two weeks late).					X
To increase rentals by 25% annually through the introduction of membership privileges.	X				
Our current system uses Oracle Database 9*i* and is on UNIX.				X	
The number of seriously overdue DVDs should be <3%.		X			
Starlight must stock a very wide range of DVDs and sufficient copies of the latest DVDs to ensure that membership is attractive.			X		
Store clerks are unable to chase overdue DVDs.					X
Starlight would like to upgrade to Oracle Database 11*g* on Linux.				X	

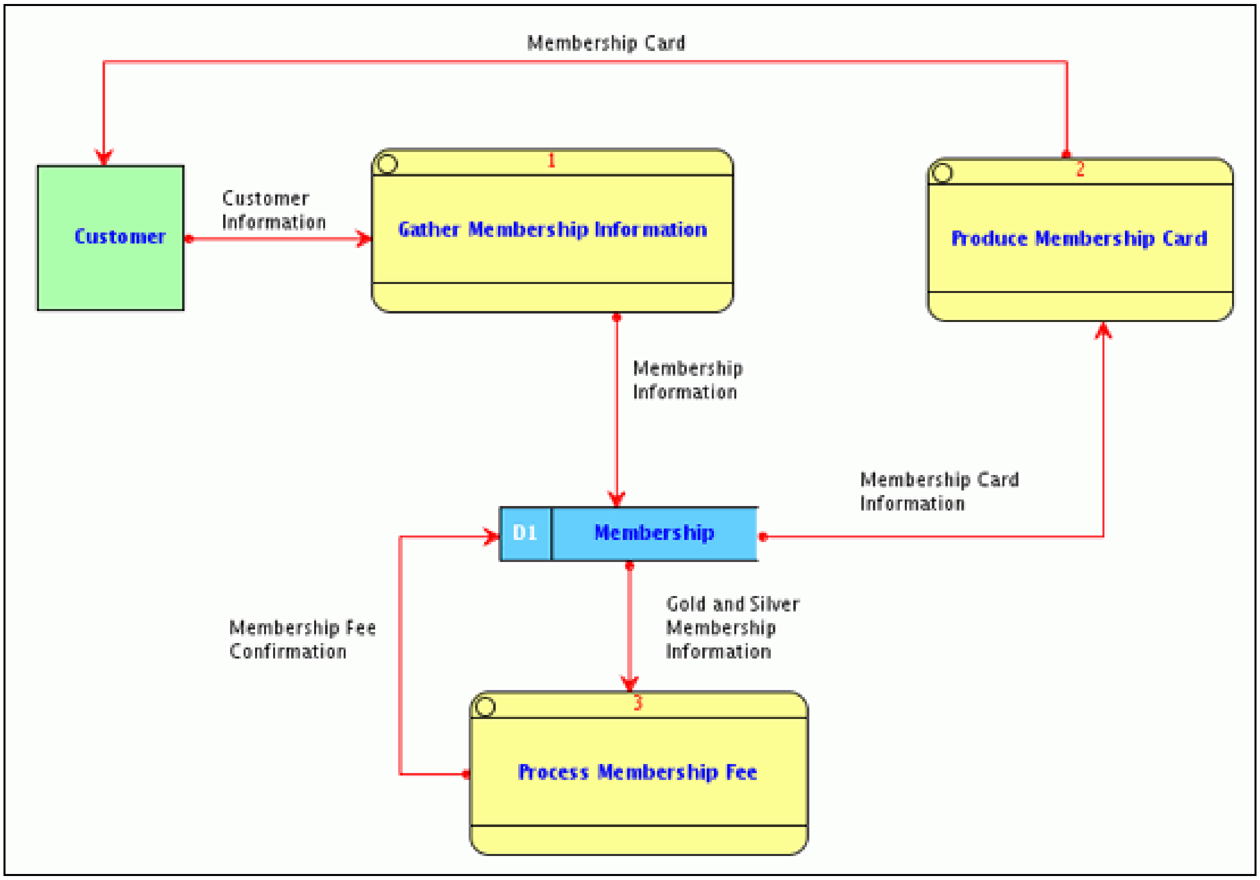
Solution: Lesson 3 Class Practice: Create a Data Flow Diagram

One possible solution to this practice is as follows:



Solution 3-1: Create a Data Flow Diagram

One possible solution to this practice is as follows:



Solution 4-1: Build a Data Flow Diagram in Oracle SQL Developer Data Modeler

To create and populate the DFD, perform the following steps:

1) Open Oracle SQL Developer Data Modeler.

2) Expand the Process Model node.

3) Right-click Data Flow Diagrams and select New Data Flow Diagram.

4) Select the New Process icon and click anywhere in the white space of the Data Flow Diagram.

5) Enter `Gather Membership Information` for the name, and click OK.

6) Click in the white space of the Data Flow Diagram to create a second process.

7) Enter `Produce Membership Card` for the name, and click OK.

8) Click one more time in the white space of the Data Flow Diagram.

9) Enter `Process Membership Fee` for the name, and click OK.

10) Select the New Information Store icon, and click in the white space of the Data Flow Diagram.

11) Enter `Membership` for the name, and click OK.

12) Select the New External Agent icon, and click in the white space of the Data Flow Diagram.

13) Enter `Customer`, for the name and click OK.

14) Select the New Flow → icon.

15) Click the Customer external agent and then click the Gather Membership Information process.

16) Double-click the information flow that you just created.

17) Enter `Customer Information` for the flow name and click OK. If the flow name is not displayed, perform the following steps:

    a) Select Tools > General Options.

    b) Select the Diagram node.

    c) Select the check box for Show Flow Name, and click OK.

18) Select the New Flow → icon.

19) Click the Gather Membership Information process, and then click the Membership information store.

20) Double-click the information flow that you just created.

Solution 4-1: Build a Data Flow Diagram in Oracle SQL Developer Data Modeler (continued)

21) Enter `Membership Information` for the flow name, and click OK.

22) Select the New Flow → icon.

23) Click the Membership information store, and then click the Process Membership Fee process.

24) Double-click the information flow that you just created.

25) Enter `Gold and Silver Membership Information` for the flow name, and click OK.

26) Select the New Flow → icon.

27) Click the Process Membership Fee process, and then click the Membership information store.

28) Double-click the information flow that you just created.

29) Enter `Membership Fee Confirmation` for the flow name, and click OK.

30) Select the New Flow → icon.

31) Click the Membership information store, and then click the Produce Membership Card process.

32) Double-click the information flow that you just created.

33) Enter `Membership Card Information` for the flow name, and click OK.

34) Select the New Flow → icon.

35) Click the Produce Membership Card process, and then click the Customer external agent.

36) Double-click the information flow that you just created.

37) Enter `Membership Card` for the flow name, and click OK.

38) You may want to move the objects around to maximize the room on your screen. See the steps in the slides to learn how to create straight lines and elbows.
Note: First turn off Auto Route by right-clicking the white space and then deselecting Auto Route.

39) You also want to specify the event for each process. Double-click Gather Membership Information.

40) In the Process Properties dialog box that appears, select the Events property in the navigator, and click the Add + icon.

41) In the Add Event dialog box, make sure that New Event is selected, enter `Customer fills out Membership Form` for the name, and select Person for the Type. Then click OK twice.

Solution 4-1: Build a Data Flow Diagram in Oracle SQL Developer Data Modeler (continued)

42) Double-click Produce Membership Card.

43) Select the Events property in the navigator, and click the Add ➕ icon.

44) Make sure that Available Event is selected, select “Customer fills out Membership” in the Events list, and click OK twice.

45) Double-click Process Membership Fee.

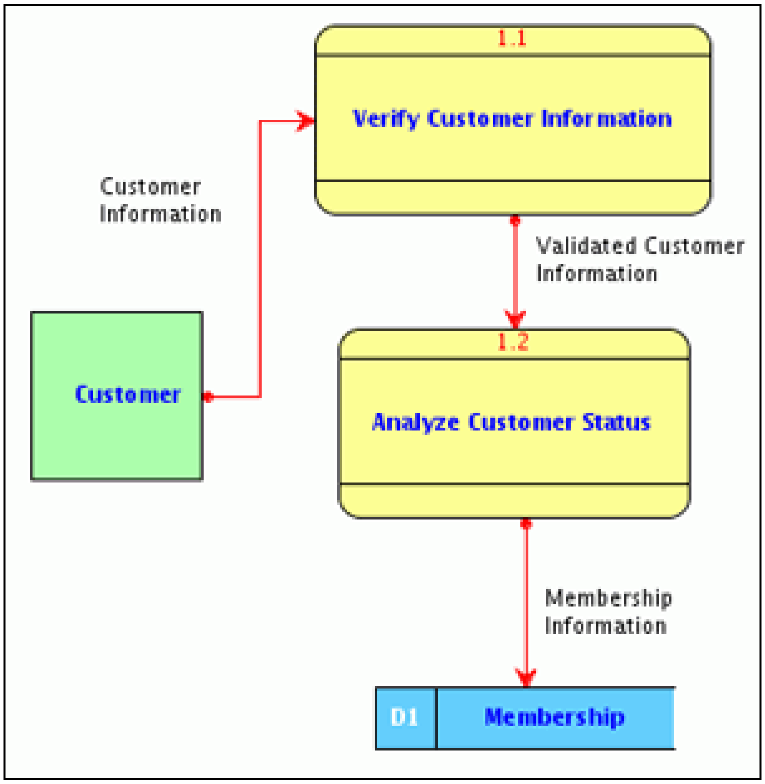
46) Select the Events property in the navigator, and click the Add ➕ icon.

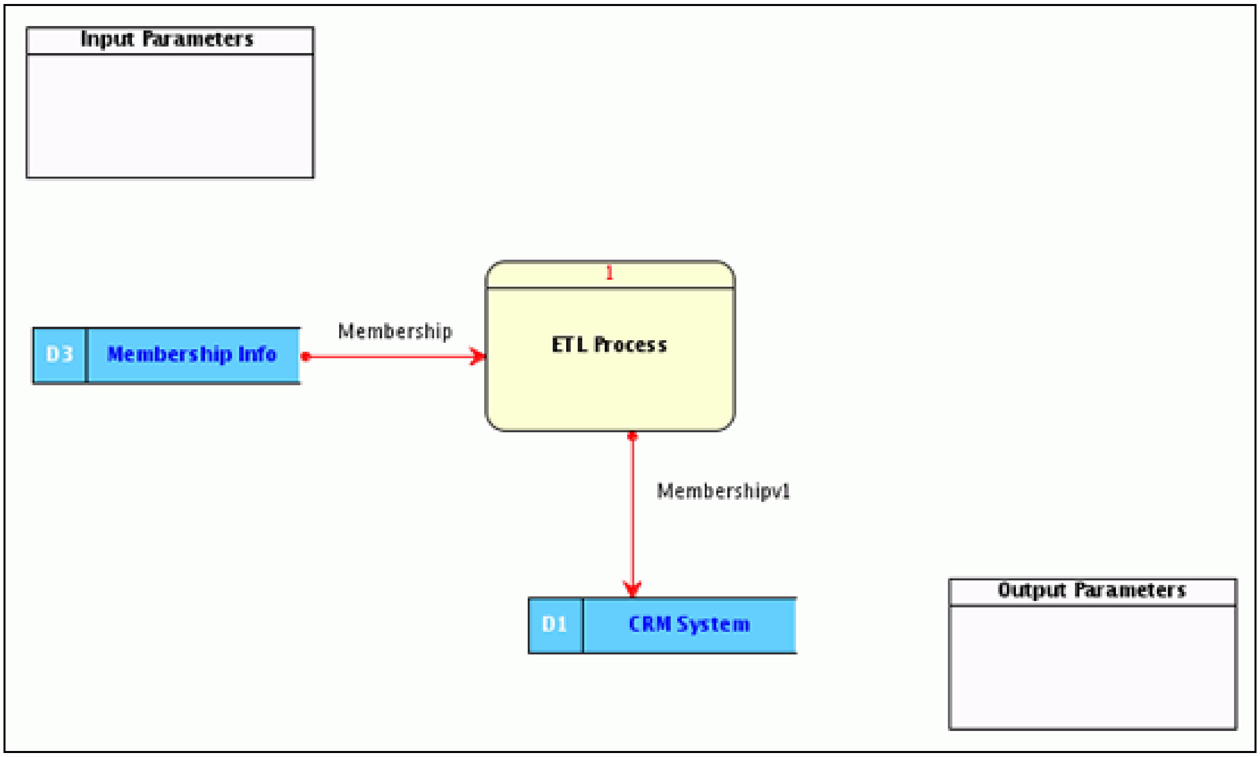
47) Make sure that New Event is selected, enter `New Membership Information` for the name, and select Clock/Calendar for the type. Then click OK twice.

48) Your data flow diagram is now complete for this practice.

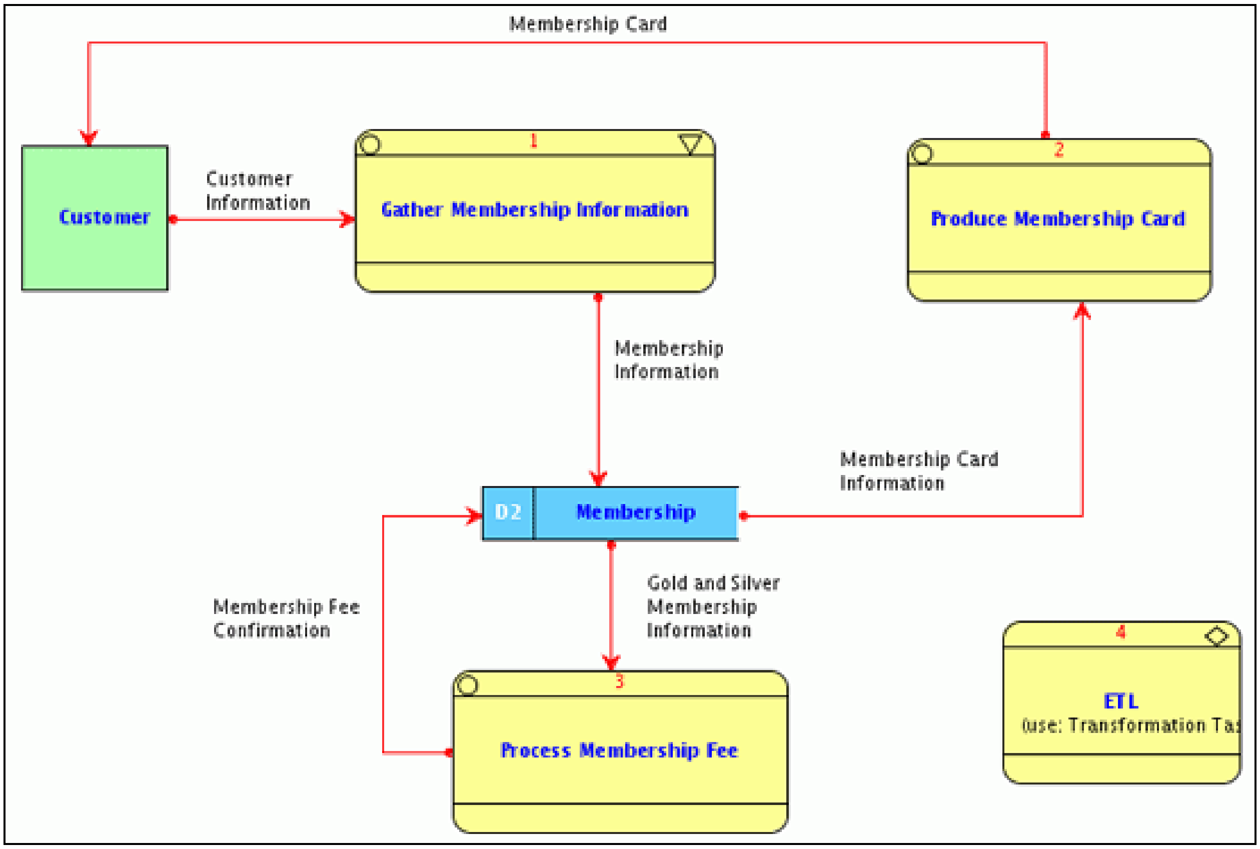
Solution 5-1: Decompose a Process in Your Data Flow Diagram

One solution to this practice is as follows:





Solution 5-1: Decompose a Process in Your Data Flow Diagram (continued)



The steps required to produce a lower-level DFD for the Gather Membership Information process are as follows:

1) Open your solution to Practice 4-1, or open `sol4-1.xml` from the solutions directory.

2) Double-click the Gather Membership Information process.

3) Change the type to Composite, and click OK.

4) Right-click the Gather Membership Information process, and select Go To Diagram.

5) Notice that the Customer External Agent and Membership Information Store were brought automatically into the lower-level DFD. Select the New Process icon and click anywhere in the white space of the data flow diagram.

6) Enter `Verify Customer Information` for the name, and click OK.

7) With the New Process icon selected, click in the white space of the data flow diagram.

8) Enter `Analyze Customer Status` for the name, and click OK.

Solution 5-1: Decompose a Process in Your Data Flow Diagram (continued)

9) Select the New Flow icon.

10) Click the Customer external agent, and then click the Verify Customer Information process.

11) Double-click the information flow that you just created.

12) Enter Customer Information for the flow name, and click OK.

13) Select the New Flow icon.

14) Click the Verify Customer Information process, and then click the Analyze Customer Status process.

15) Double-click the information flow that you just created.

16) Enter `Validated Customer Information` for the flow name, and click OK.

17) Select the New Flow icon.

18) Click the Analyze Customer Status process, and then click the Membership information store.

19) Double-click the information flow that you just created.

20) Enter `Membership Information` for the flow name, and click OK.

The steps required to produce the transformation process are as follows:

21) Expand the Process Model node in the object browser.

22) Right-click Transformation Packages and select New Package.

23) Expand Transformation Packages.

24) Right-click Transformation Package_1 and select New Transformation Task.

25) Select the New Transformation icon and click in the white space of the diagram.

26) Enter `ETL Process` for the name, and click OK.

27) Select the New Information Store icon, and click in the white space of the diagram.

28) Enter `Membership Info` for the name and click OK.

29) With the New Information Store icon selected, click the white space of the diagram to create another information store.

30) Enter `CRM System` for the name, and click OK.

31) Select the New Flow icon.

32) Click the Membership Info information store, and then click the ETL Process process.

Solution 5-1: Decompose a Process in Your Data Flow Diagram (continued)

33) Double-click the information flow that you just created.

34) Enter `Membership` for the flow name and click OK.

35) Select the New Flow → icon.

36) Click the ETL Process process, and then click the CRM System information store.

37) Double-click the information flow that you just created.

38) Enter `Membership` for the flow name, and click OK.

39) Notice that the name of the flow was changed to be a unique value because you already have a flow with the name Membership.

40) Now that the transformation task is created, you can create a transformation task process on your DFD. Switch to your Gather Membership Information DFD tab.

41) Select the New Process icon, and click in the white space of the diagram.

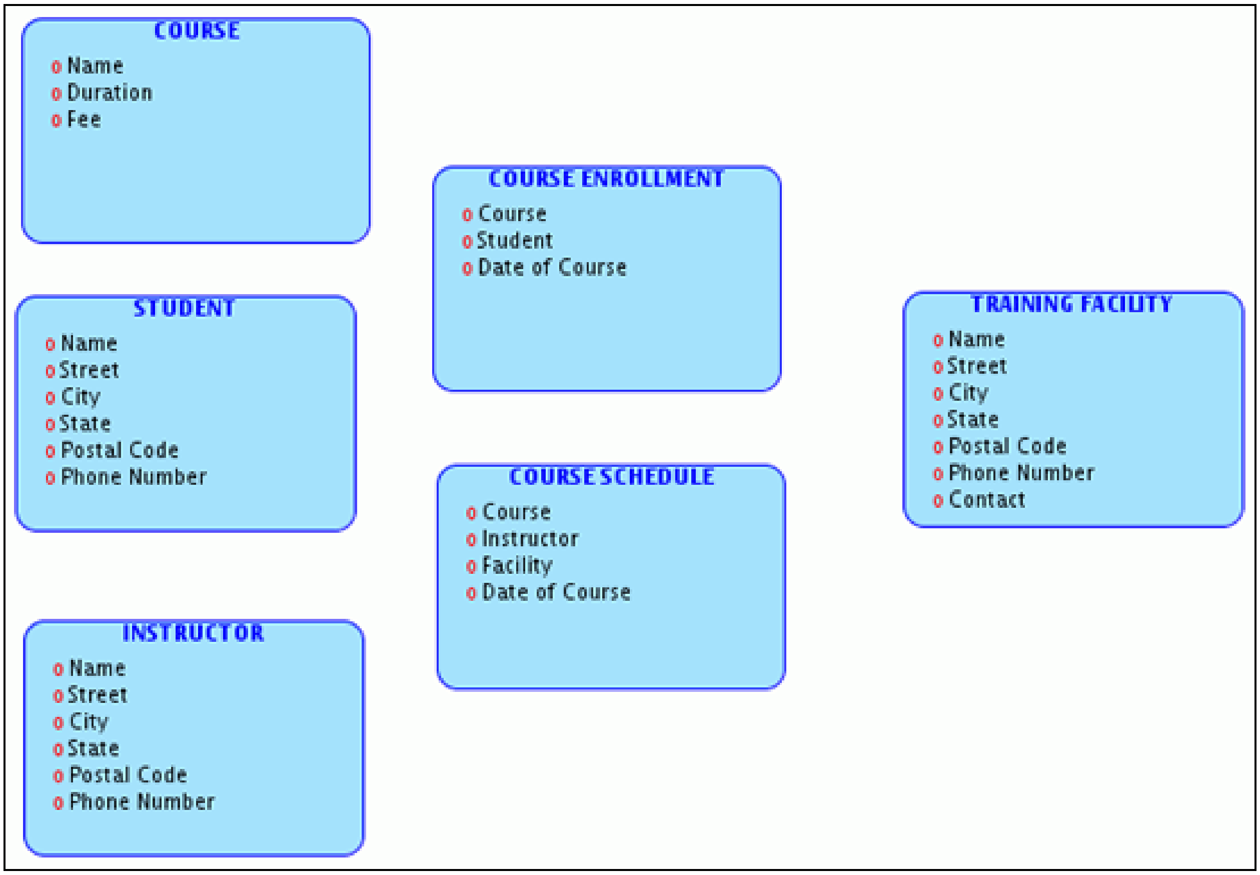
42) Enter `ETL` for the name, select Use Transformation Task for the type, and then click the button for Use Transformation task and select TransformationPackage_1.Transformation_Task_1, and then click OK twice.

43) The solution to this process has been completed.

.

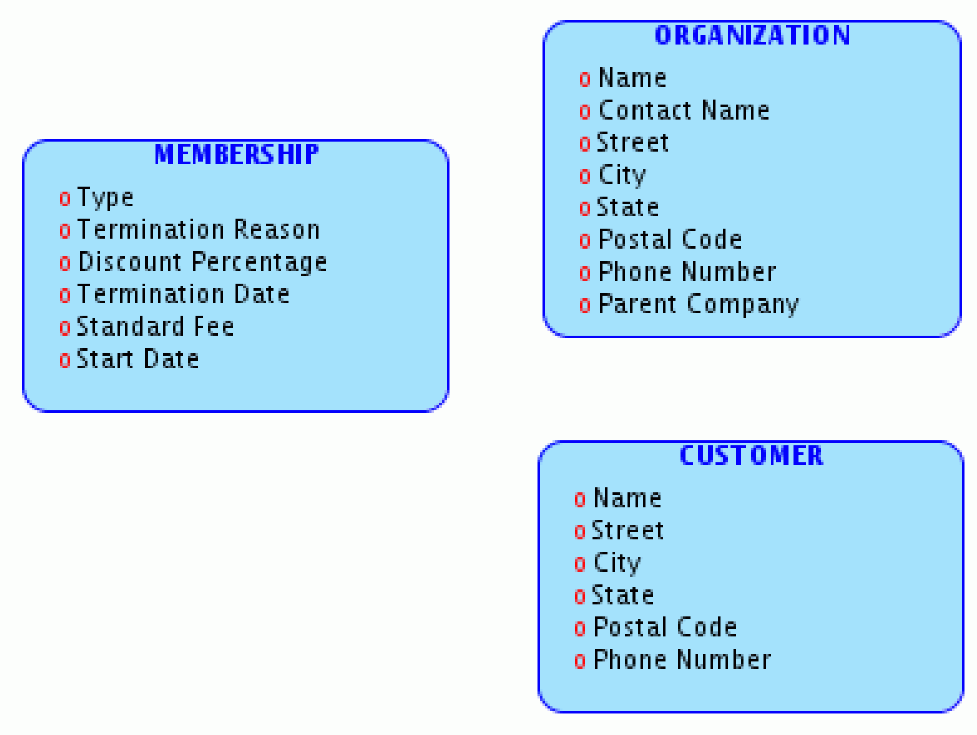
Solution: Lesson 6 Class Practice: Identify Entities and Attributes

One possible solution for this practice is:



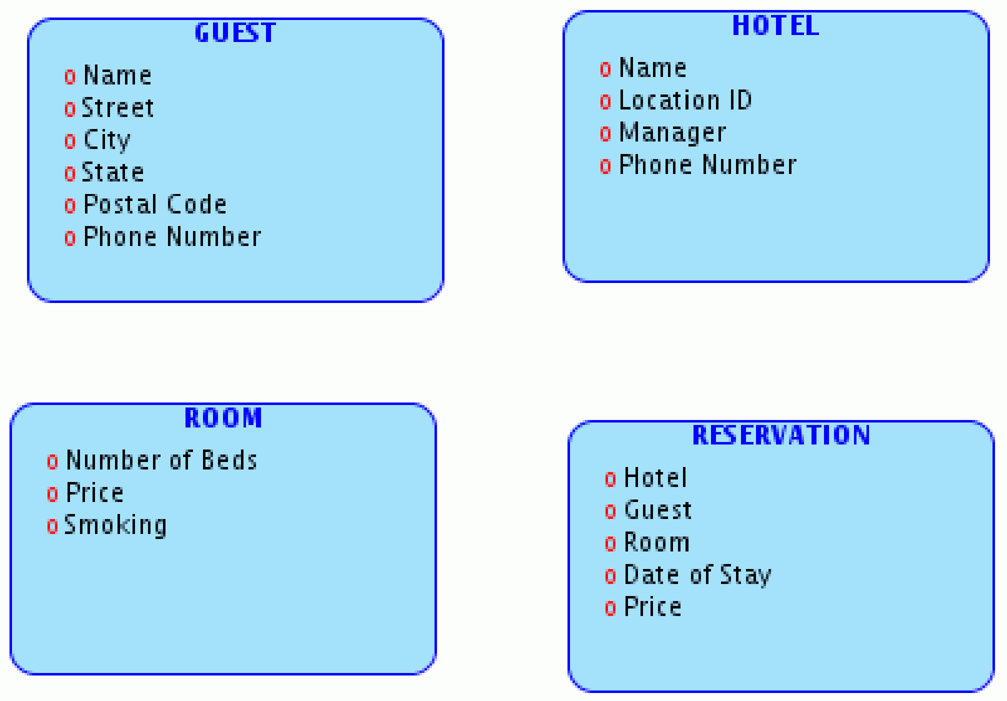
Solution 6-1: Identify Entities and Attributes

One possible solution for this practice is:



Solution 6-2: Identify Entities and Attributes

One possible solution for this practice is:



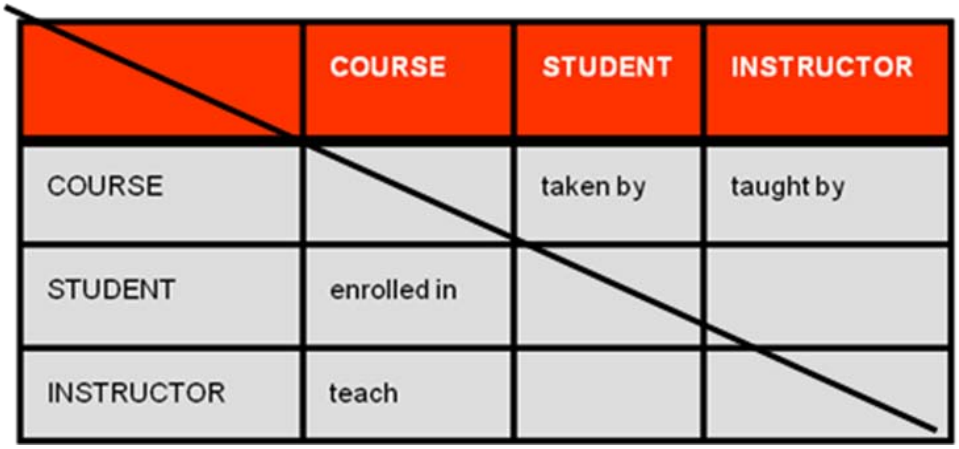
Solution: Lesson 7 Class Practice: Define Business Rules

The solution for this practice is as follows:

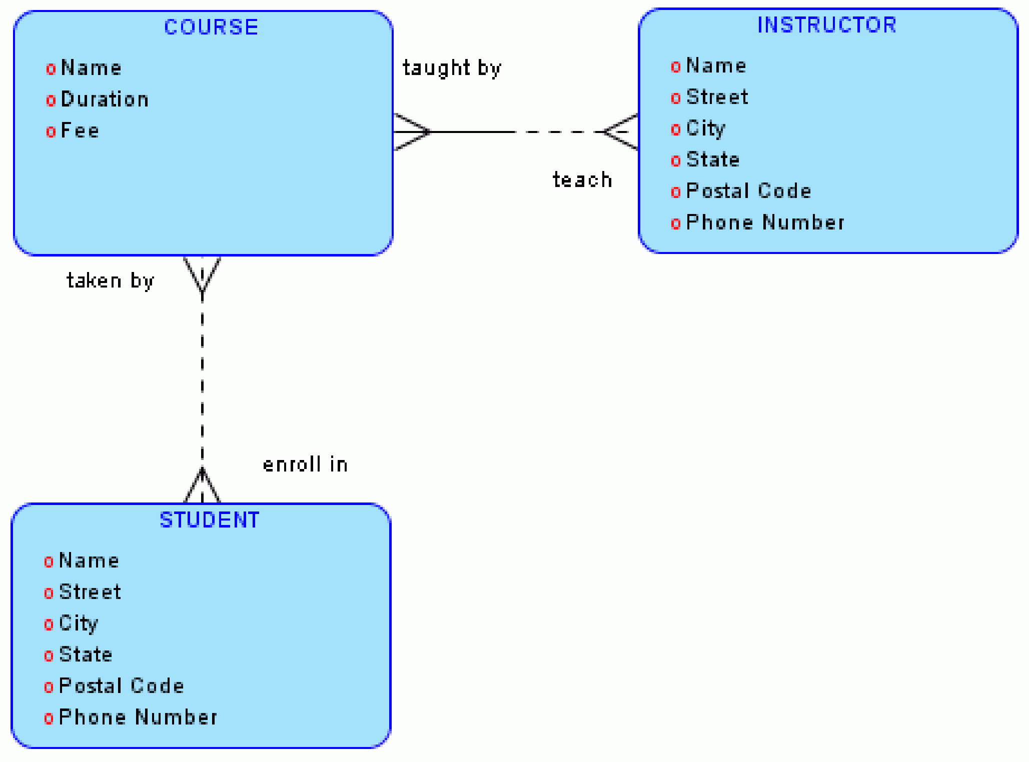
An ORDER must be issued for one or more ITEMs
An ITEM may be contained on one or more ORDERs
An ORDER must be placed by one and only one CUSTOMER
A CUSTOMER may place one or more ORDERs
An ITEM may be stored in one and only one WAREHOUSE
A WAREHOUSE may store one or more ITEMs

Solution: Lesson 7 Class Practice: Build a Relationship Matrix

The solution to this practice is as follows:

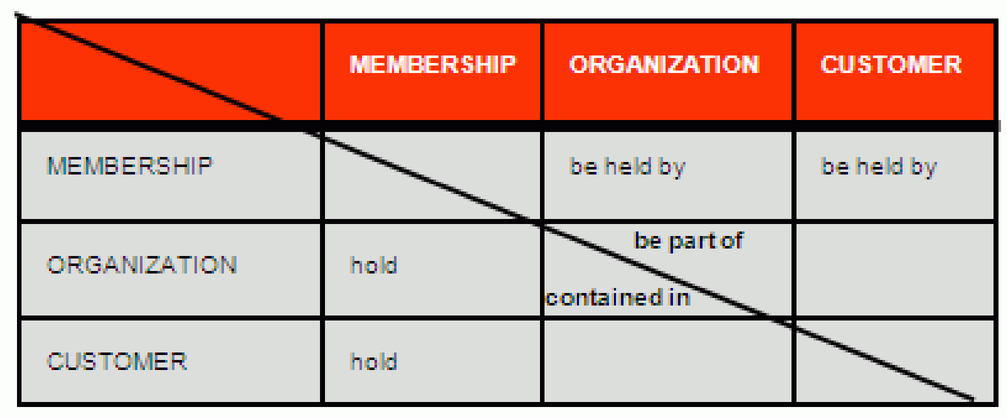


	COURSE	STUDENT	INSTRUCTOR
COURSE		taken by	taught by
STUDENT	enrolled in		
INSTRUCTOR	teach		

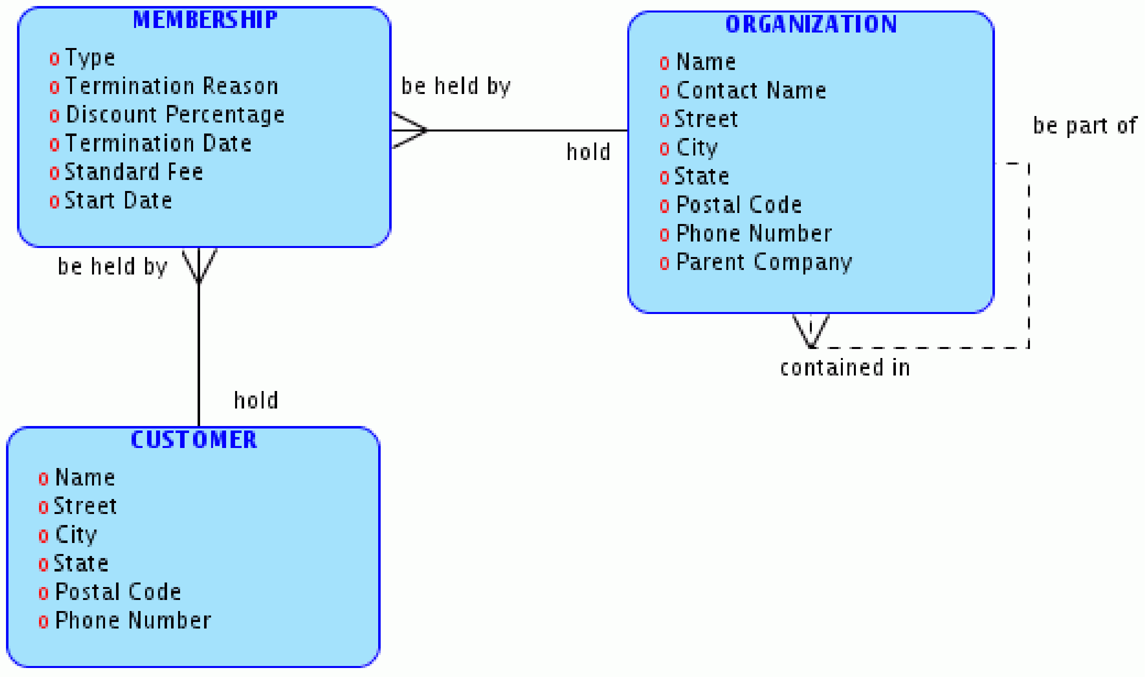


Solution 7-1: Analyze and Model Relationships

One solution to this practice is:

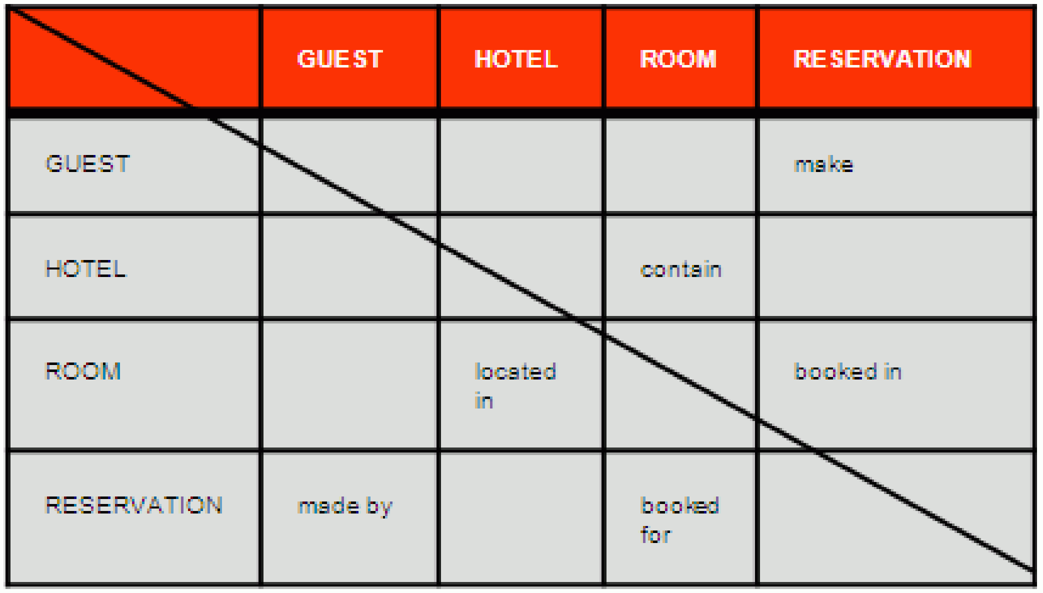


	MEMBERSHIP	ORGANIZATION	CUSTOMER
MEMBERSHIP		be held by	be held by
ORGANIZATION	hold	be part of contained in	
CUSTOMER	hold		

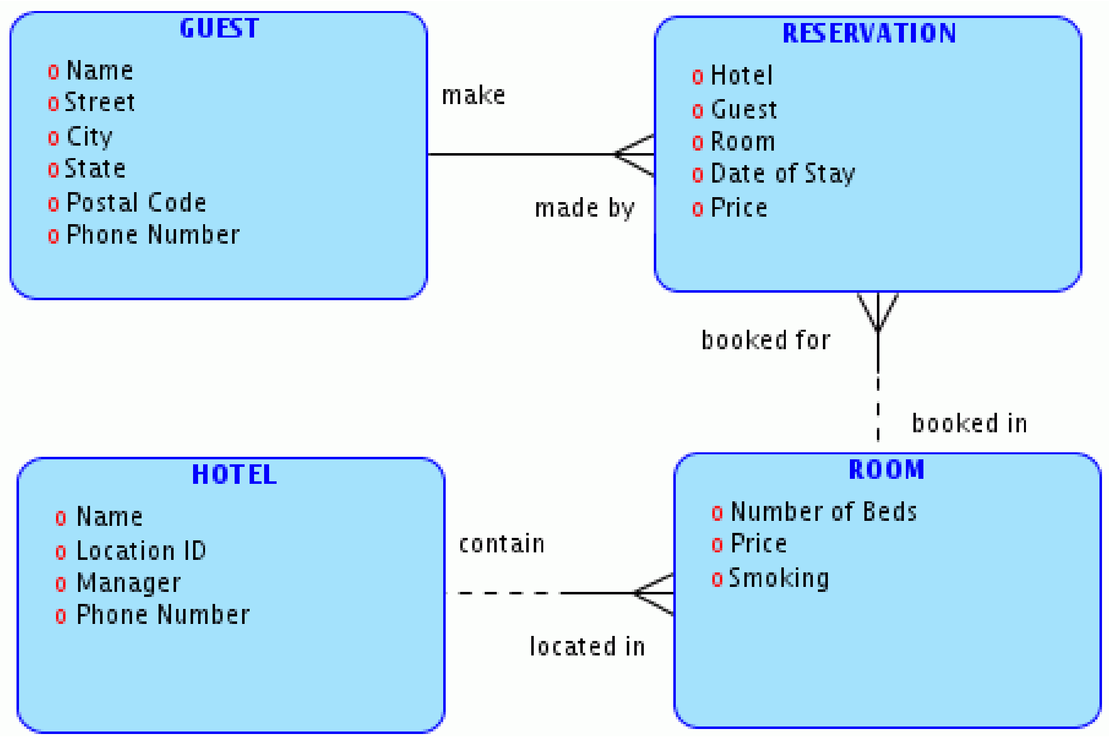


Solution 7-2: Analyze and Model Relationships

One solution to this practice is:

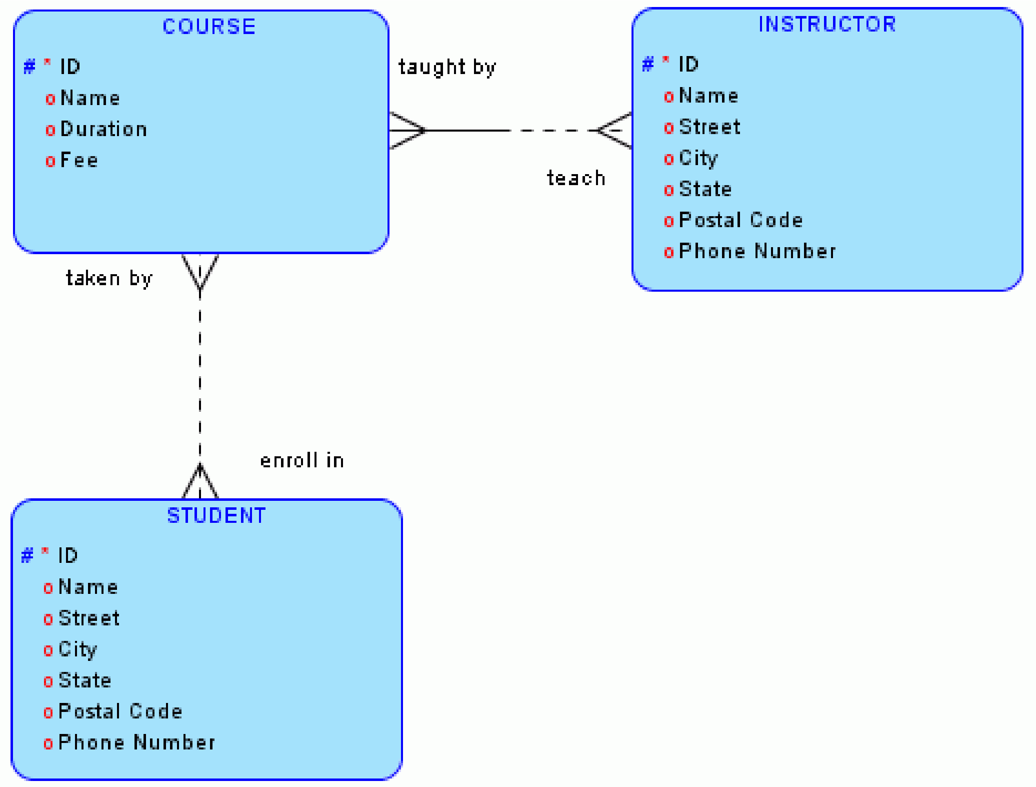


	GUEST	HOTEL	ROOM	RESERVATION
GUEST				make
HOTEL			contain	
ROOM		located in		booked in
RESERVATION	made by		booked for	



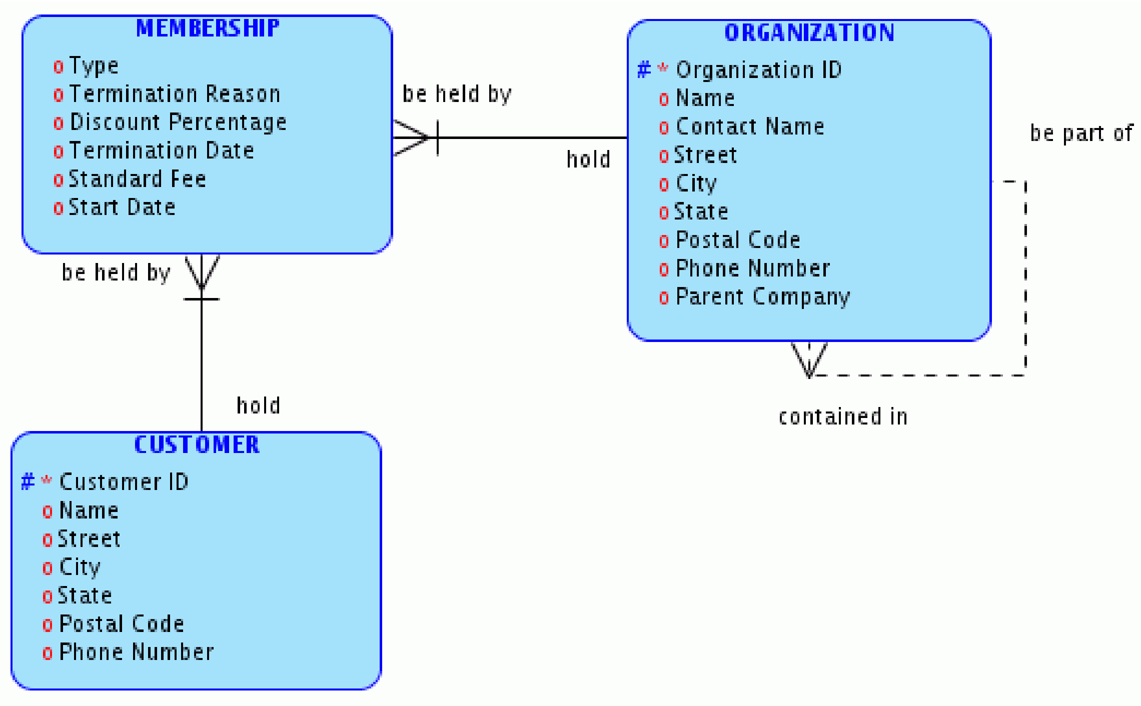
Solution: Lesson 8 Class Practice: Specify Unique Identifiers

A possible solution to this practice is as follows:



Solution 8-1: Identify Unique Identifiers

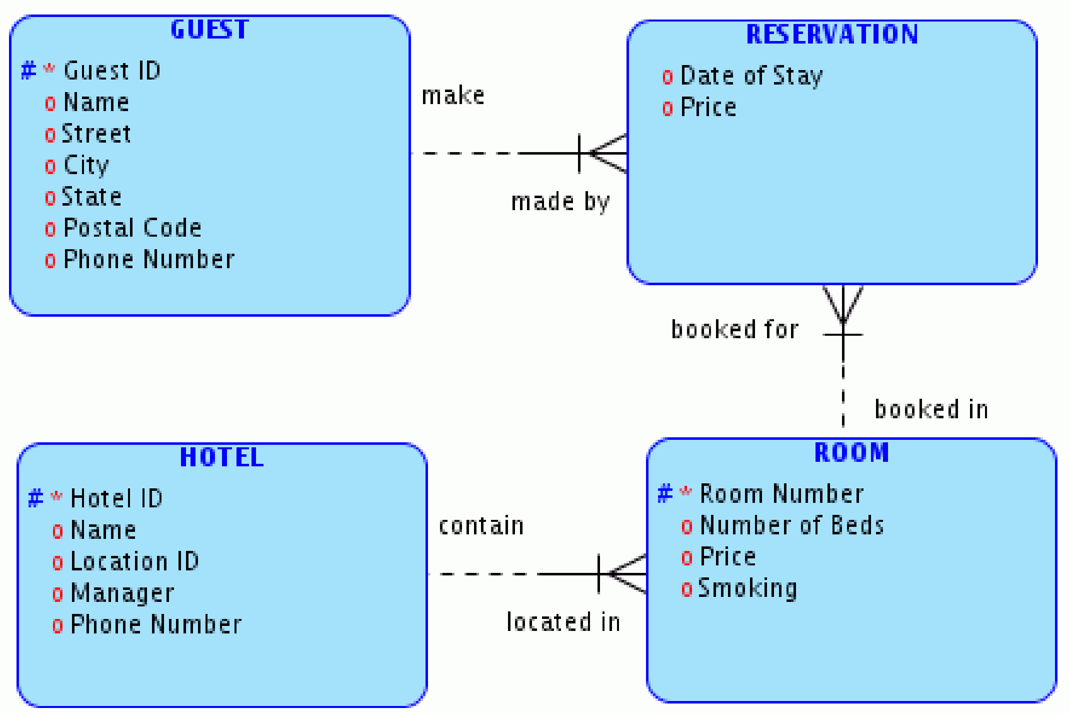
One possible solution to this practice is:



Note that the unique identifier for the MEMBERSHIP entity is the composite of the unique identifiers for both CUSTOMER and ORGANIZATION.

Solution 8-2: Identify Unique Identifiers

One possible solution to this practice is:



Solution 9-1: Build an ERD in Oracle SQL Developer Data Modeler

The steps required to build the class exercise from Lesson 8 are as follows:

1) Close any models that you currently have open. Select File > Close.
2) Click the Logical tab.
3) Select the New Entity ⚙ icon, and click anywhere in the white space of the entity relationship diagram.
4) Enter `Course` for the name, and click Attributes in the left navigator.
5) Select the Add ➕ icon.
6) Enter `Course ID` for the name, select the Primary UID check box, and then click the Add ➕ icon.
7) Enter `Name` for the name, and then click the Add ➕ icon.
8) Enter `Duration` for the name, and then click the Add ➕ icon.
9) Enter `Fee` for the name, and then click OK.
10) Click in the white space of the entity relationship diagram.
11) Enter `Instructor` for the name, and click Attributes in the left navigator.
12) Select the Add ➕ icon.
13) Enter `Instructor ID` for the name, select the Primary UID check box, and then click the Add ➕ icon.
14) Enter `Name` for the name, and then click the Add ➕ icon.
15) Enter `Street` for the name, and then click the Add ➕ icon.
16) Enter `City` for the name, and then click the Add ➕ icon.
17) Enter `State` for the name, and then click the Add ➕ icon.
18) Enter `Postal Code` for the name, and then click the Add ➕ icon.
19) Enter `Phone Number` for the name, and then click OK.
20) Click in the white space of the entity relationship diagram.
21) Enter `Student` for the name, and click Attributes in the left navigator.
22) Select the Add ➕ icon.

Solution 9-1: Build an ERD in Oracle SQL Developer Data Modeler (continued)

23) Enter `Student ID` for the name, select the Primary UID check box, and then click the Add + icon.

24) Enter `Name` for the name, and then click the Add + icon.

25) Enter `Street` for the name, and then click the Add + icon.

26) Enter `City` for the name, and then click the Add + icon.

27) Enter `State` for the name, and then click the Add + icon.

28) Enter `Postal Code` for the name, and then click the Add + icon.

29) Enter `Phone Number` for the name, and then click OK.

30) Your entities and attributes have been created. Click the New 1:N Relation → icon.

31) Click the INSTRUCTOR entity, and then click the COURSE entity

32) Double-click the relationship that you just created.

33) Select the Cardinality property in the left navigator.

34) Enter `teaches` for “Name on Source” and `taught by` for “Name on Target,” and click OK.

35) If your relationship names do not appear, perform the following steps:

    a) Select Tools > General Options.

    b) Expand Diagram, and select Logical Model.

    c) Select the Show Source/Target Name check box, and click OK.

36) Create another relationship between COURSE and STUDENT. Select the New M:N Relation ⇆ icon.

37) Click the COURSE entity, and then click the STUDENT entity.

38) Double-click the relationship that you just created.

39) Select the Cardinality property in the left navigator.

40) Enter `taken by` for “Name on Source” and `enrolled in` for “Name on Target,” and click OK.

41) To create a subview of the entities, select all the objects on the diagram and right-click an entity, select “Create SubView from selected.”

42) You may need to move some entities around to maximize space on your screen and to minimize the crossing of lines.

Solution 9-1: Build an ERD in Oracle SQL Developer Data Modeler (continued)

43) You may also want to straighten your lines or create an elbow to move a line. To do this, remember to turn off Auto Route first.

To produce the diagram from Practice 8-1 in Oracle SQL Developer Data Modeler, you go through the same steps as above. In this case, however, you want to create some identifying relationships and a recursive relationship.
You can create an identifying relationship in one of two ways:

1) To create the identifying relationship between CUSTOMER and MEMBERSHIP using the 1:N identifying relationship icon, perform the following steps:

   a) Select the 1:N identifying relationship icon.

   b) Click the CUSTOMER entity, and then click the MEMBERSHIP entity.

   c) Double-click the relationship that you just created, and click the Cardinality property in the left navigator.

   d) Enter `hold` for “Name of Source” and `be held by` for “Name of Target,” and click OK.

2) To create the identifying relationship between ORGANIZATION and MEMBERSHIP by specifying the UID in the relationship, perform the following steps:

   a) Select the 1:N relationship icon.

   b) Select the ORGANIZATION entity, and then click the MEMBERSHIP entity.

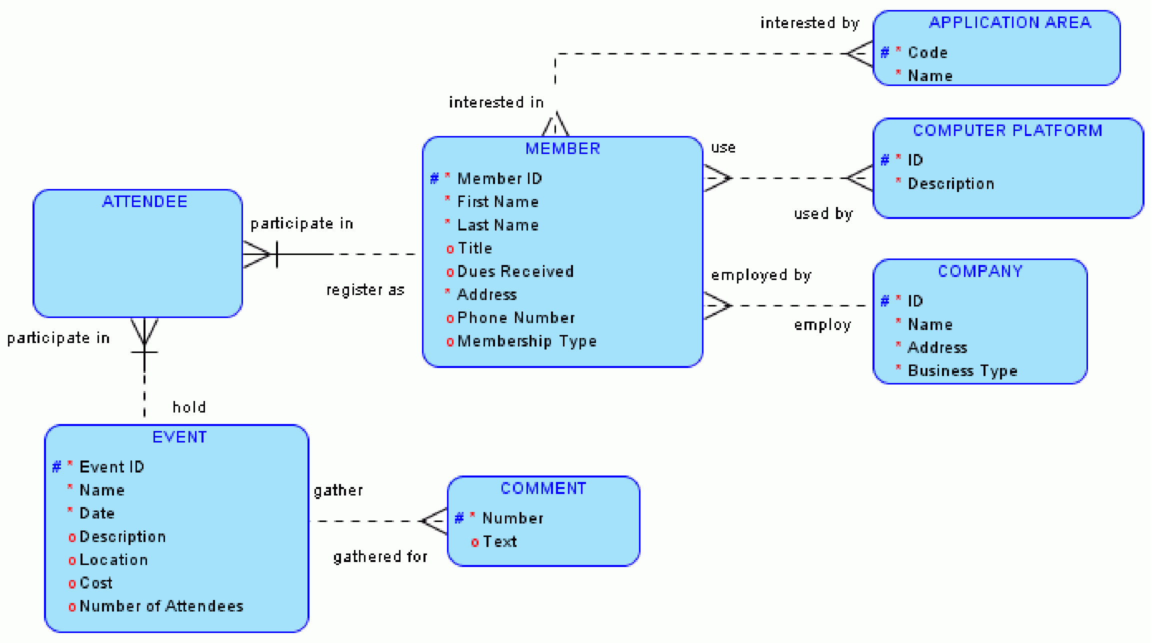
   c) Double-click the relationship that you just created, and click the Cardinality property in the left navigator.

   d) Enter `hold` for “Name of Source” and `be held by` for “Name of Target,” select the Identifying check box, and then click OK.

To create the recursive relationship, select the icon for the relationship that you want to create, in this case, the 1:M relationship, and click the ORGANIZATION entity two times. Double-click the relationship to change the cardinality source and target names.
To produce the diagram for Practice 8-2 in Oracle SQL Developer, perform the steps already discussed above.

Solution 10-1: Develop and Validate Your ERD

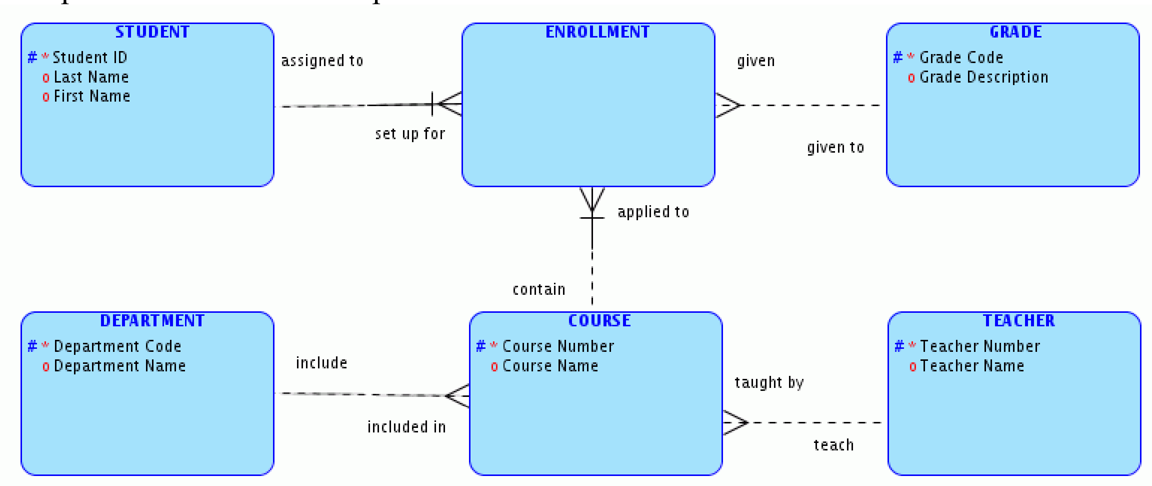
One possible solution to this practice is:



You should be able to create the diagram in Oracle SQL Developer Data Modeler based on the previous practice. The only task not discussed is how to designate an attribute as mandatory. This is done from the Entity properties window when the attribute is selected. There is a check box called Mandatory. This must be checked for all attributes that must have a value.

Solution 11-1: Normalize an ERD

One possible solution to this practice is:



1) The grade information is a repeating group in ENROLLMENT; therefore, you create its own entity GRADE and put the grade code and description in that new entity.

2) The teacher information is also a repeating group in COURSE; therefore, you create its own entity TEACHER and put the teacher number and name in that new entity.

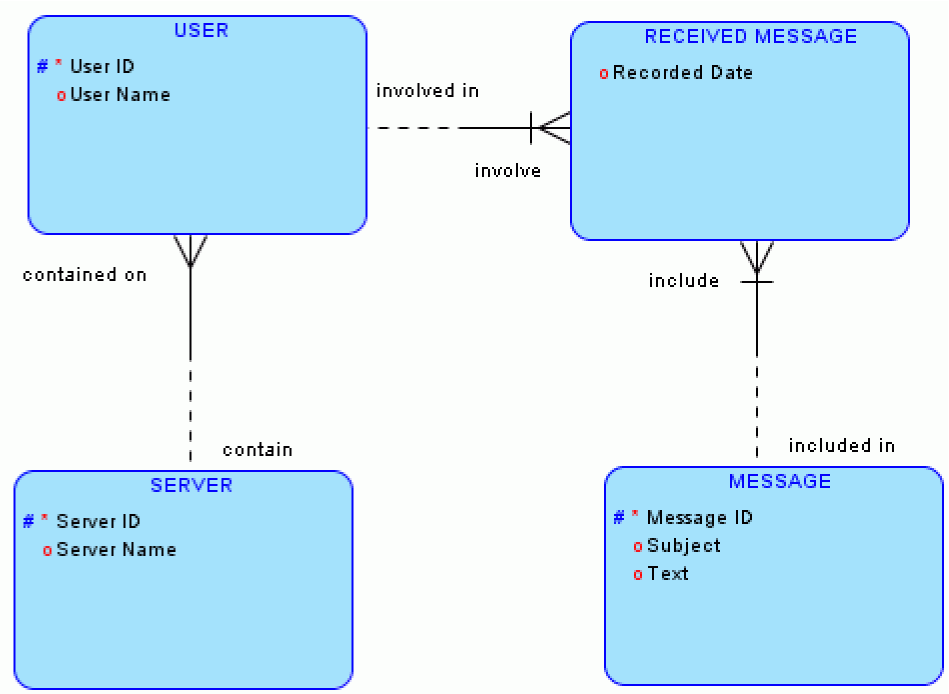
3) The Teacher number in enrollment can be eliminated because it is already in the new TEACHER entity.

4) The Department Name depends on the Department Code; therefore, you can create its own entity DEPARTMENTS and move the department code and name to that new entity.

5) In the new entities, define the unique identifiers accordingly.

Solution 11-2: Validate ERD for Normalization

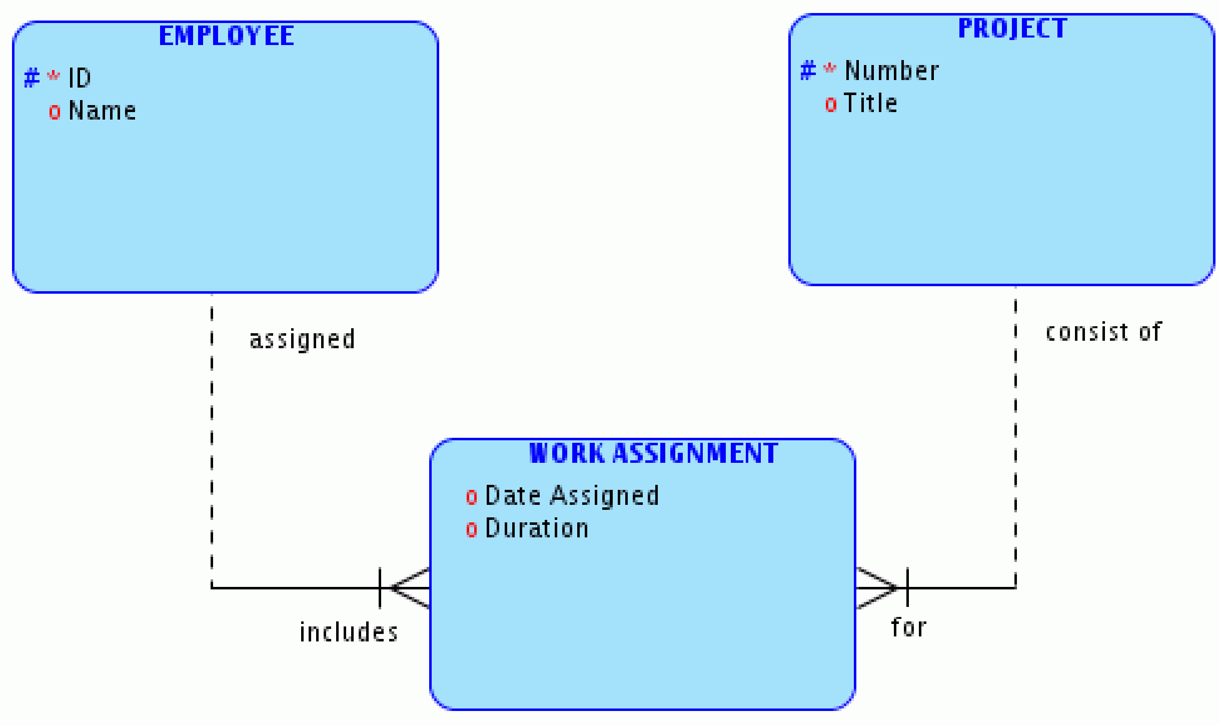
One possible solution to this practice is:

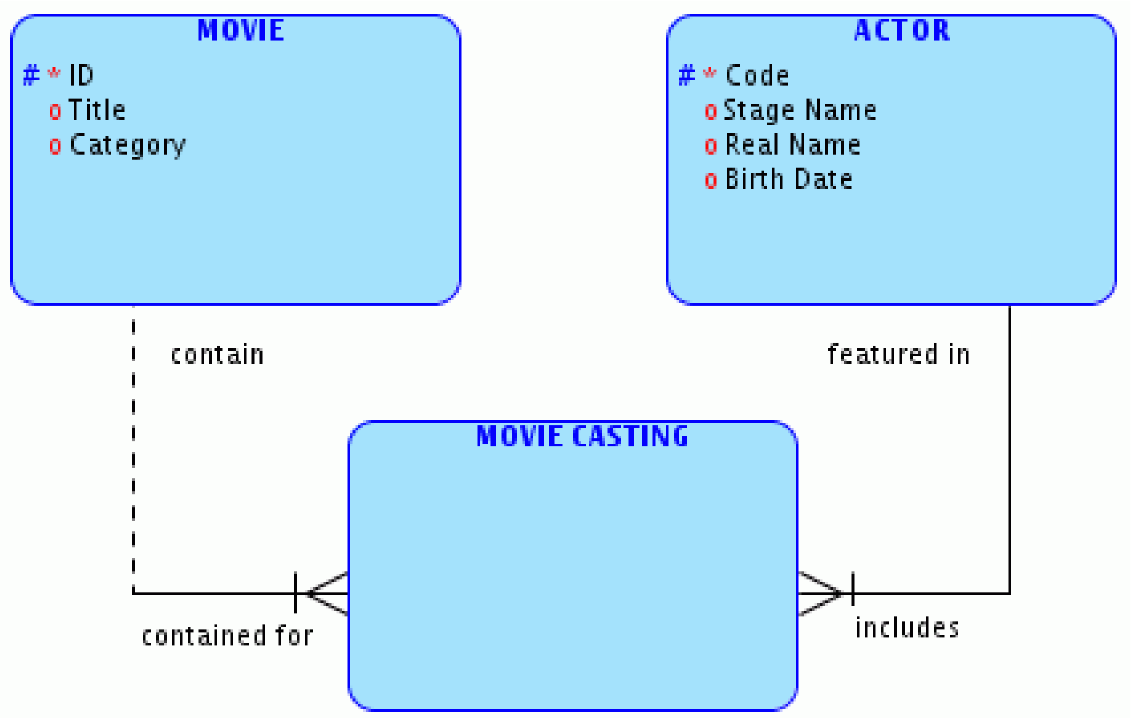


1) For first normal form, you needed to identify that there were multiple messages received for a given user. So USER and MESSAGE were created. There was an M:M between those two entities.
2) For second normal form, you needed to identify that not all the attributes in MESSAGE were dependent on the key; therefore, you must resolve the M:M to create the RECEIVED MESSAGE entity.
3) For third normal form, you needed to identify that a Server Name only depended on the Server ID; therefore, you could create a new entity and establish a relationship with the User entity.

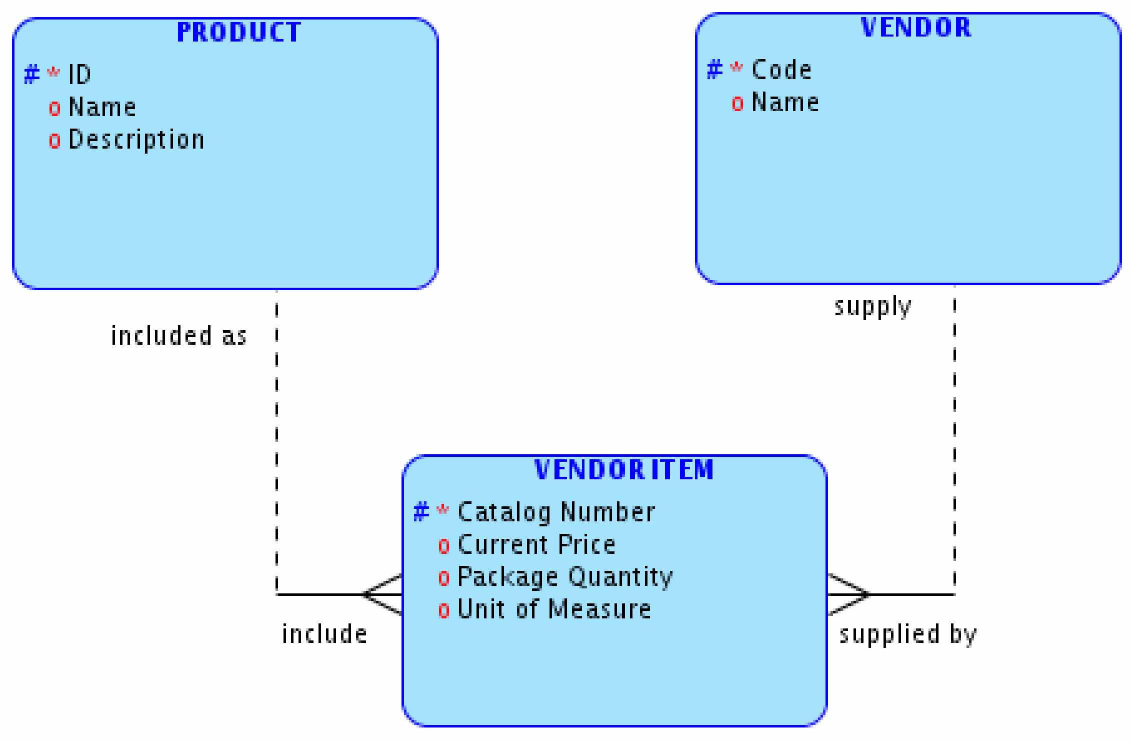
Solution 12-1: Resolve M:M Relationships

One possible solution to this practice is:





Solution 12-1: Resolve M:M Relationships (continued)

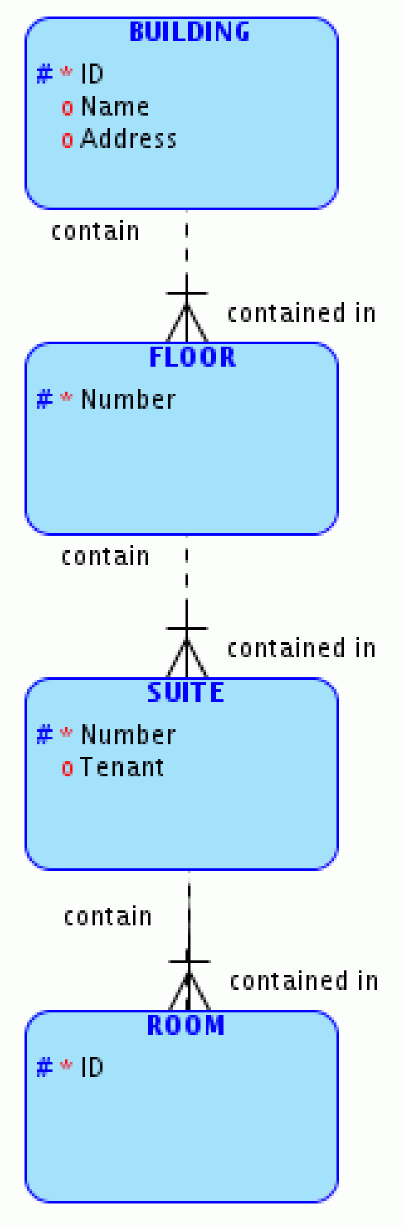


To create the resolved entity, delete the existing relationship, create the new entity and attributes, delete the attributes that you moved to the new entity, and then establish the new relationships with the new entity.

Note that the last example does not have identifying relationships with the originating entity because Catalog Number uniquely identifies a Vendor Item.

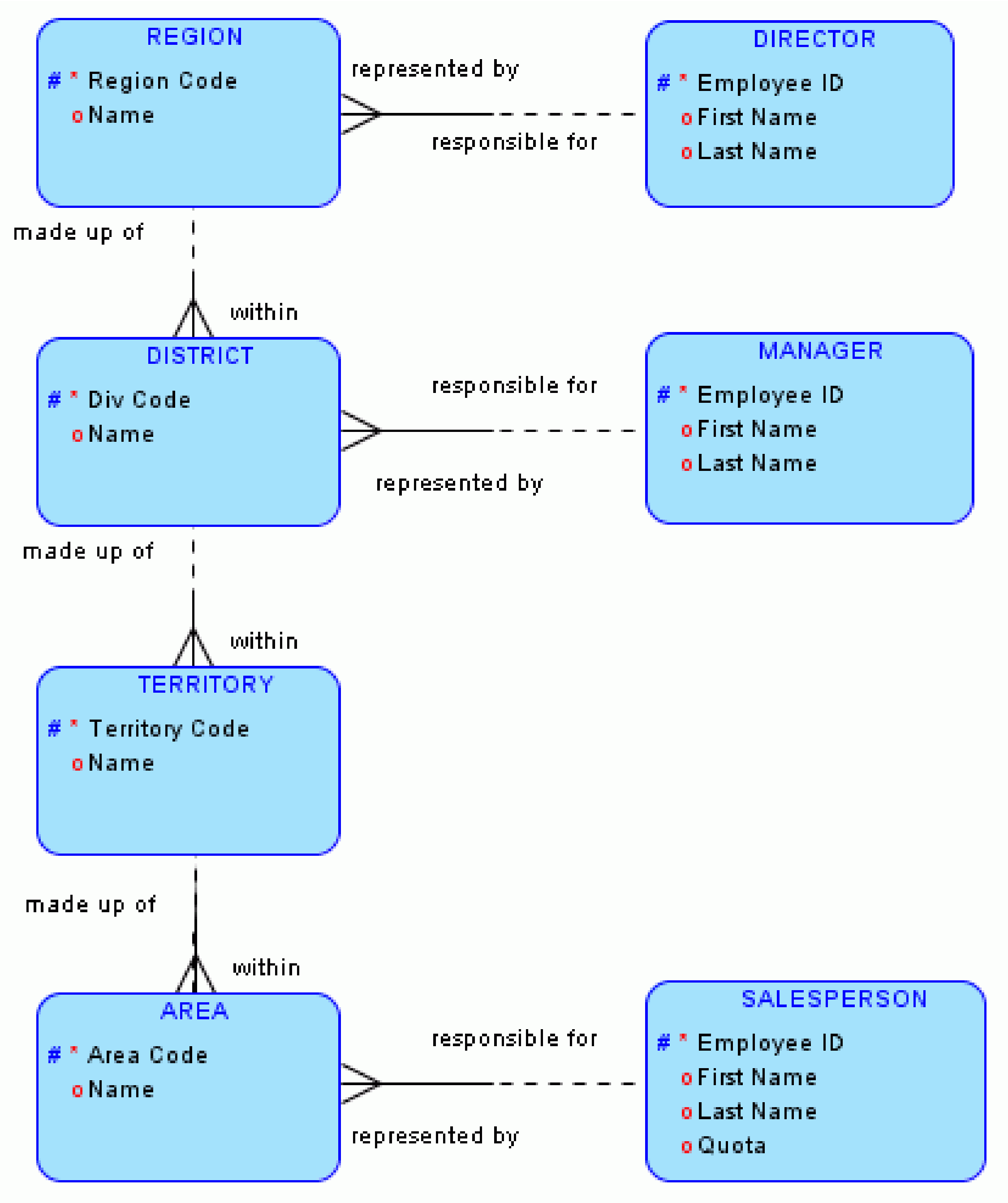
Solution 12-2: Model Hierarchical Data

One possible solution for this practice is:

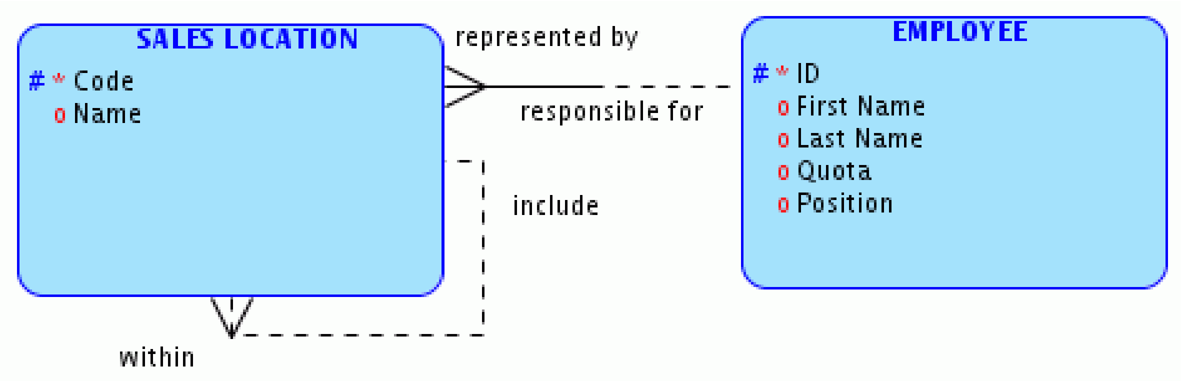


Solution 12-3: Model Hierarchical Data and Recursive Relationships

One possible solution to this practice is:

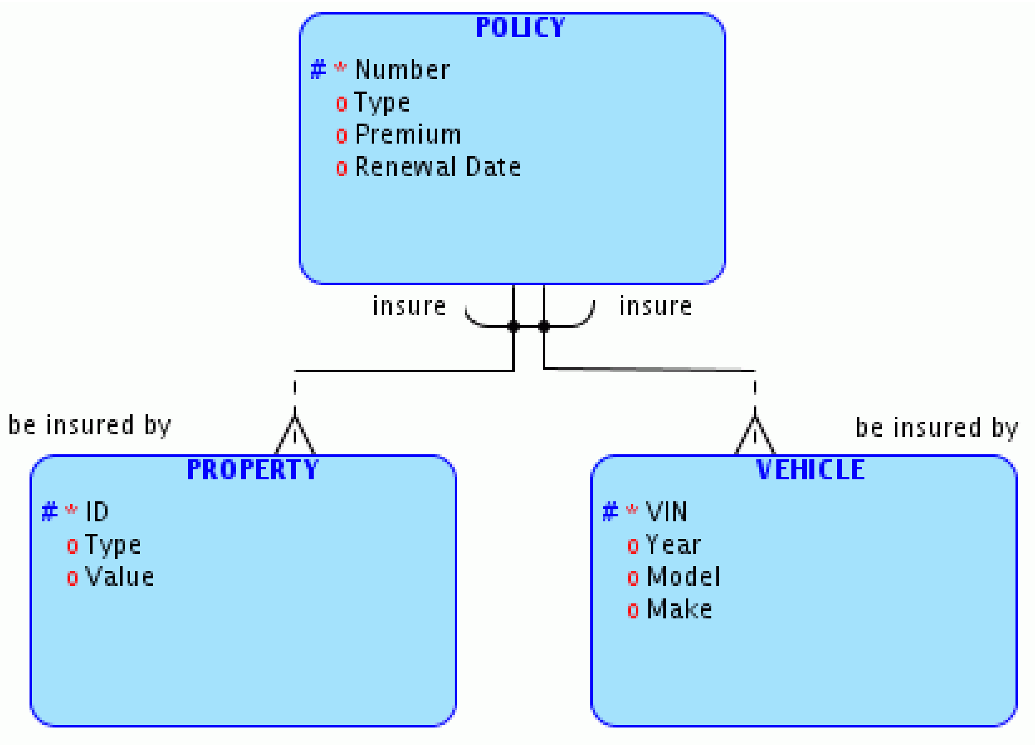


Solution 12-3: Model Hierarchical Data and Recursive Relationships (continued)



Solution 12-4: Examine Exclusive Relationships

One possible solution to this practice is:



To create the exclusive relationship shown above, perform the following steps:

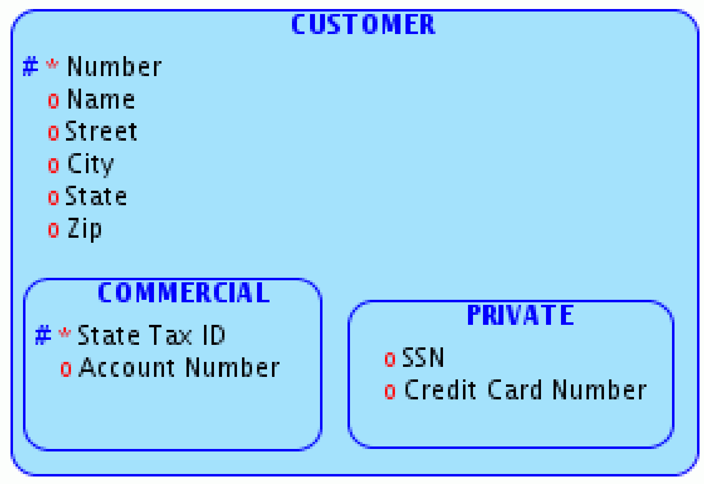
1) Create the entities, attributes, unique identifiers, and relationships as you have done previously.

2) Ctrl-select both relationships and the POLICY entity.

3) After you select all three, the New Arc exclusive relationship icon will become available. Select it.

4) An arc is created for both relationships with the POLICY entity.

Solution 12-4: Examine Exclusive Relationships (continued)

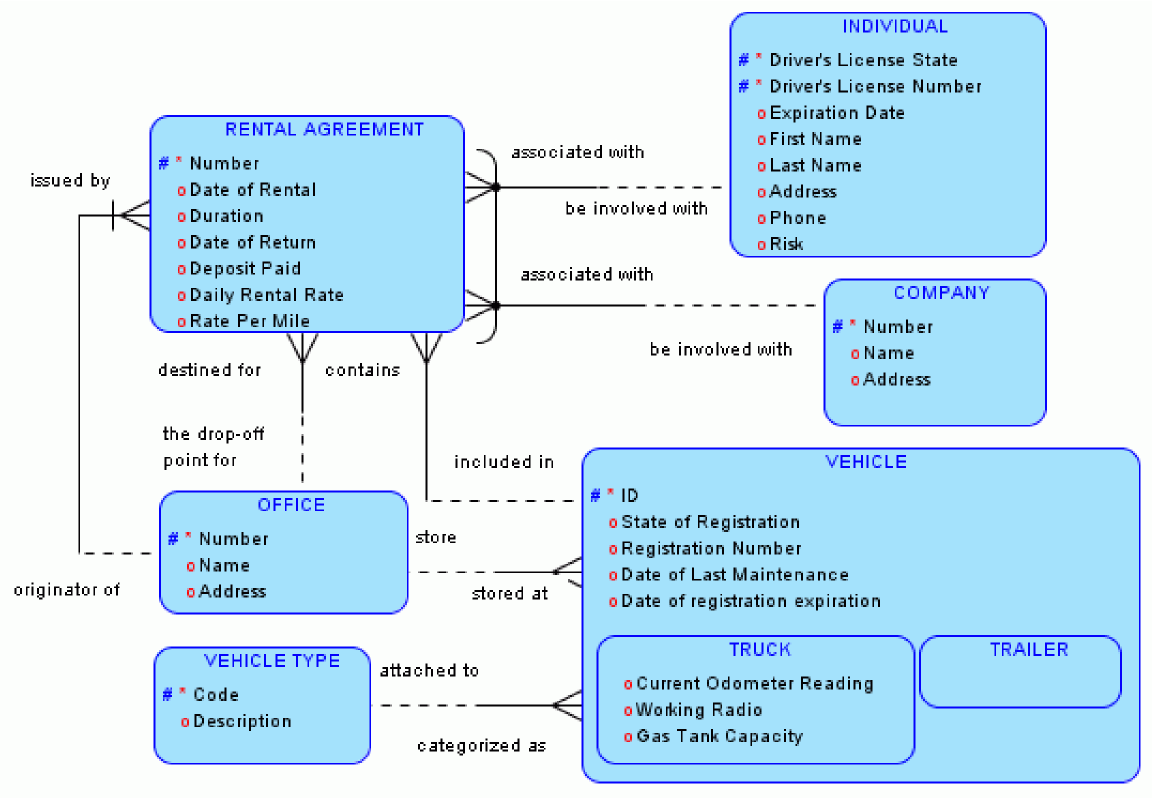


To create the entity type hierarchy shown above, perform the following steps:

1) Create the entities, attributes, and unique identifiers as you have done previously.
2) Double-click the COMMERCIAL entity, select CUSTOMER for the super type, and click OK.
3) Double-click the PRIVATE entity, select CUSTOMER for the super type, and click OK.
4) Right-click the background of the diagram, and select Box-in-Box Presentation.
5) You may need to make the super type entity bigger and the subtype entities smaller to see all the objects.

Solution 12-5: Examine Exclusive Relationships

One possible solution for this practice is:



Solution 13-1: Create and Assign Data Types

To create the data type model, perform the following steps:

1) Open the solution to Practice 12-5 (`sol12-5.xml`).

2) Right-click DataTypes, and select Show.

3) Select the New Structured Type icon, and click in the white space of the diagram.

4) Enter `emp_address_typ` for the name, and click the Attributes property in the left navigator.

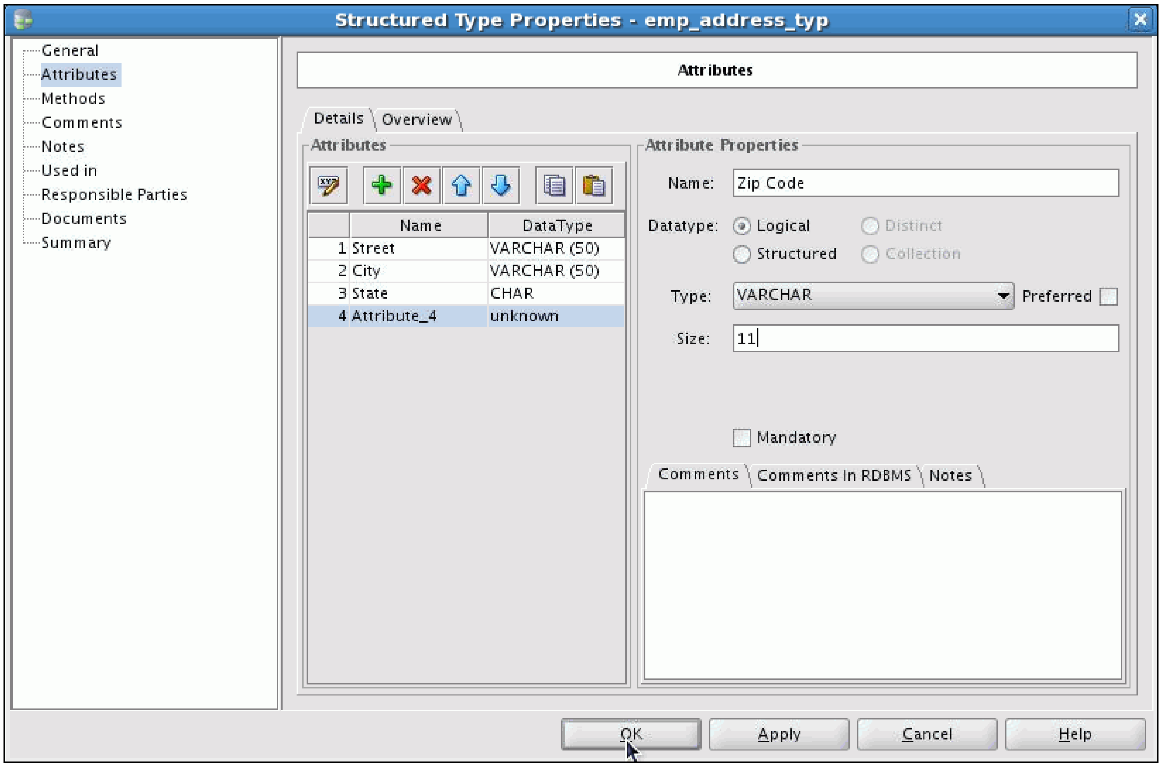
5) Click the Add icon.

6) Enter `Street` for Name, select VARCHAR for Type, and enter `50` for Size. Then click the Add icon.

7) Enter `City` for Name, select VARCHAR for Type, and enter `50` for Size. Then click the Add icon.

8) Enter `State` for Name, select CHAR for Type, and enter `2` for Size. Then click the Add icon.

9) Enter `Zip Code` for Name, select VARCHAR for Type, and enter `11` for Size. Then click OK.



	Name	DataType
1	Street	VARCHAR (50)
2	City	VARCHAR (50)
3	State	CHAR
4	Attribute_4	unknown

Name: Zip Code

Datatype: ● Logical ○ Distinct ○ Structured ○ Collection

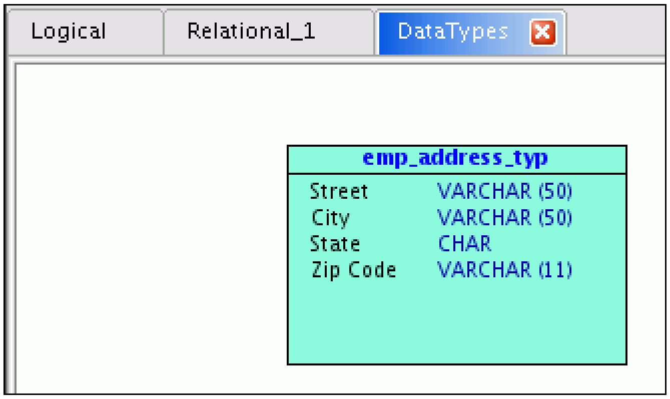
Type: VARCHAR Preferred □

Size: 11

□ Mandatory

10) Your structured type was created successfully. Now you can use it in your model.

Solution 13-1: Create and Assign Data Types (continued)



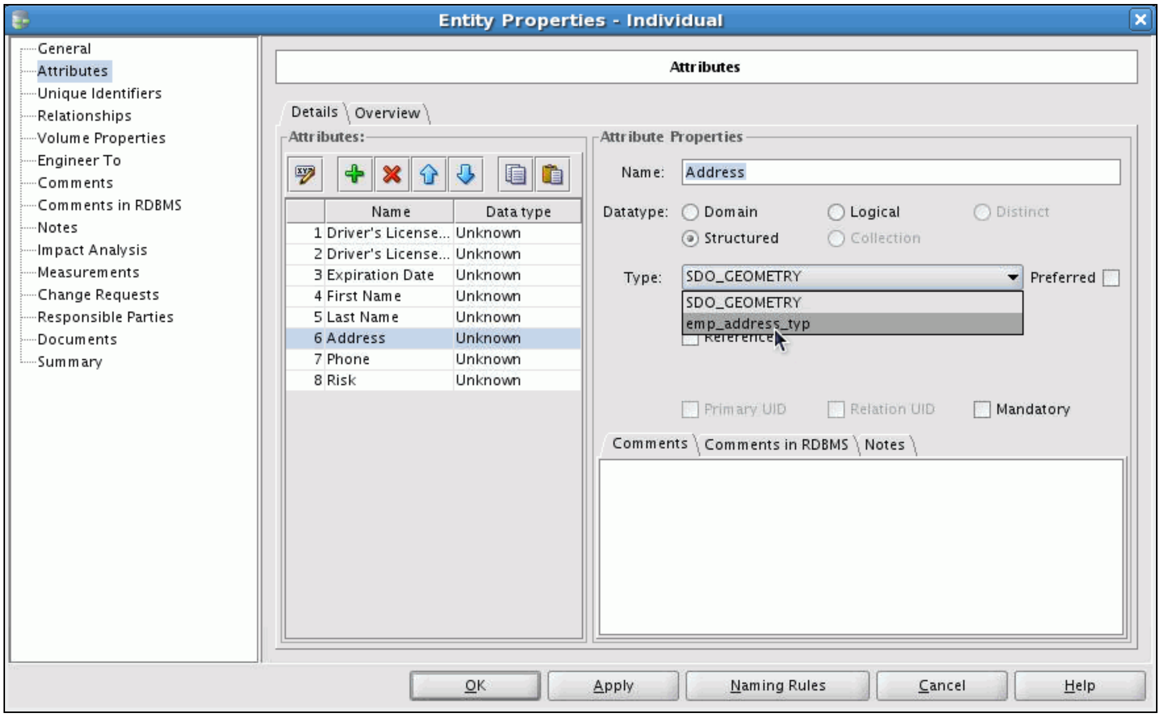
emp_address_typ

Street	VARCHAR (50)
City	VARCHAR (50)
State	CHAR
Zip Code	VARCHAR (11)

11) Click the Logical tab, and double-click the INDIVIDUAL entity.

12) Select the Attributes property in the left navigator.

13) Select the Address attribute, select the Structured option for Datatype, select `emp_address_typ` in the list of Types, and click OK.



	Name	Data type
1	Driver's License...	Unknown
2	Driver's License...	Unknown
3	Expiration Date	Unknown
4	First Name	Unknown
5	Last Name	Unknown
6	Address	Unknown
7	Phone	Unknown
8	Risk	Unknown

Name: Address

Datatype: ○ Domain ○ Logical ○ Distinct ● Structured ○ Collection

Type: SDO_GEOMETRY Preferred □

SDO_GEOMETRY
emp_address_typ

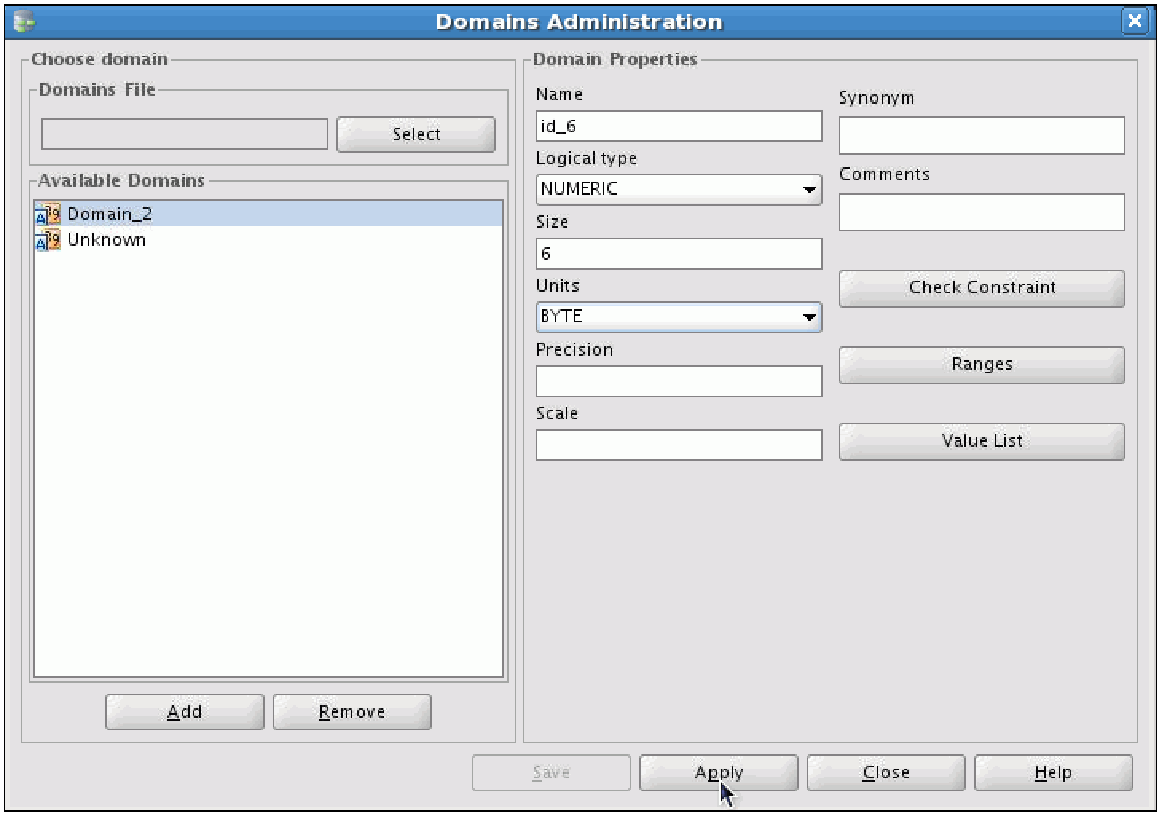
14) Perform the same steps for the other Address attributes in your model.

You can also create some domains and assign them to various attributes in your model.

1) Select Tools > Domains Administration.

2) Under Available Domains, click Add.

3) Enter `id_6` for Name, select NUMERIC for Logical type, enter `6` for Size, and select BYTE for Units. Then click Apply.



Name: id_6
Logical type: NUMERIC
Size: 6
Units: BYTE
Precision:
Scale:
Synonym:
Comments:

4) Under Available Domains, click Add.

5) Enter `Name_50` for Name, select VARCHAR for Logical type, enter `50` for Size, and select CHAR for Units. Then click Apply.

6) Notice that your two domains appear in the Available Domains list. Click Save to save them to the file.

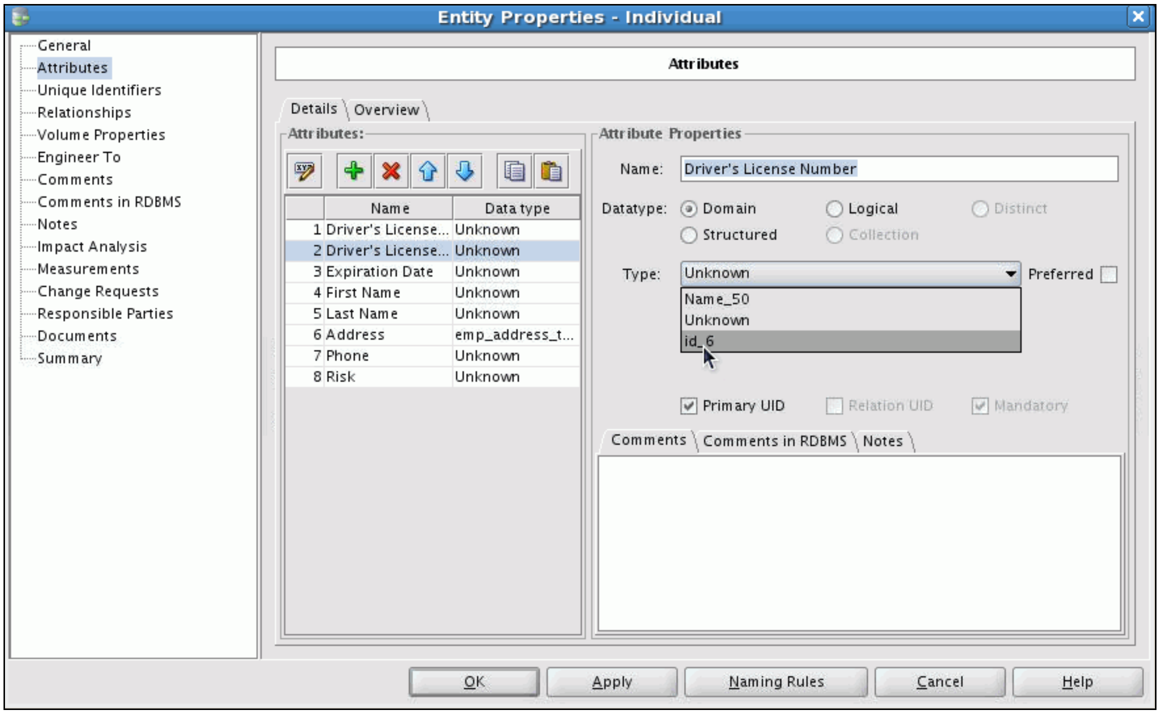
7) Click Close.

8) Double-click the INDIVIDUAL entity, and select the Attributes property in the left navigator.

9) Select the Driver's License Number attribute.

10) Make sure that Domain is selected for Datatype, select the id_6 domain in the list of Types, and click Apply.

Solution 13-1: Create and Assign Data Types (continued)



	Name	Data type
1	Driver's License...	Unknown
2	Driver's License...	Unknown
3	Expiration Date	Unknown
4	First Name	Unknown
5	Last Name	Unknown
6	Address	emp_address_t...
7	Phone	Unknown
8	Risk	Unknown

Name: Driver's License Number

Datatype: ● Domain ○ Logical ○ Distinct ○ Structured ○ Collection

Type: Unknown ▾ Preferred ☐

Name_50
Unknown
id_6

☑ Primary UID ☐ Relation UID ☑ Mandatory

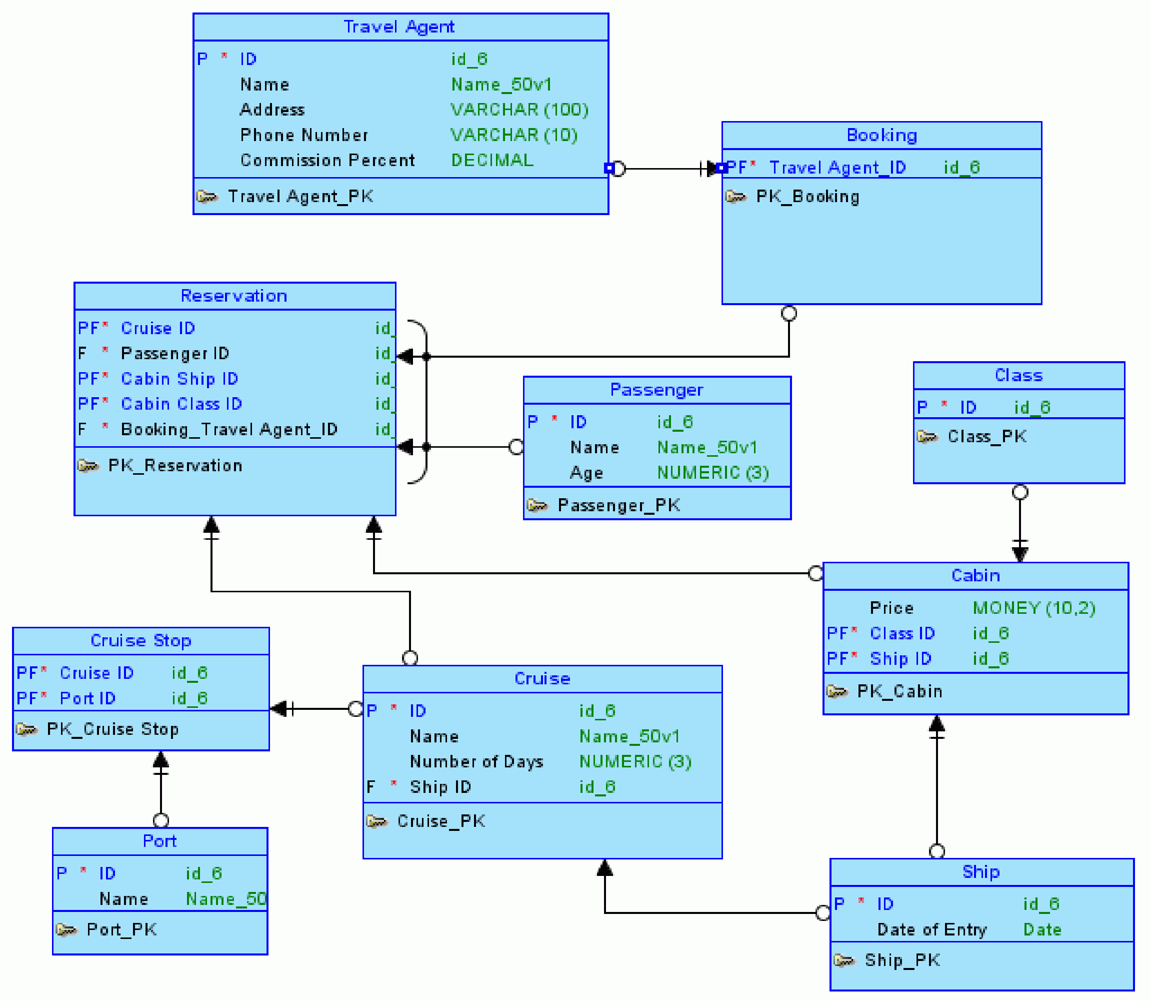
11) You can now select the First Name attribute and assign the Name_50 domain.

12) You can assign many of the attributes that have the same datatype and size using the domains throughout the entire model.

13) If the attribute does not use the structured type or domain that you have defined, you can also create a logical type by selecting the Logical option (in the Datatype section of the Entity Properties dialog box) and assigning the type specifically for that attribute.

Solution 14-1: Develop and Validate Your ERD

One possible solution may be the following:



This model is displayed with Bachman notation without the relationship labels. To view the Bachman Notation, click the white space of the diagram and select Bachman notation.

The attributes based on a relation (represented with the PF or F) are displayed in Bachman notation. You can change the names so that they are more meaningful. For example, in the Cabin entity, the attribute name ID for the relation between Cabin and Class has been changed to Class ID. To perform this task you can perform the following steps:

1) Select Tools > General Options.
2) Expand Model and select Logical.
3) Deselect “Keep as the Name of the Originating Attribute” and click OK.
4) Double-click an entity, and select the Attributes property in the left navigator.

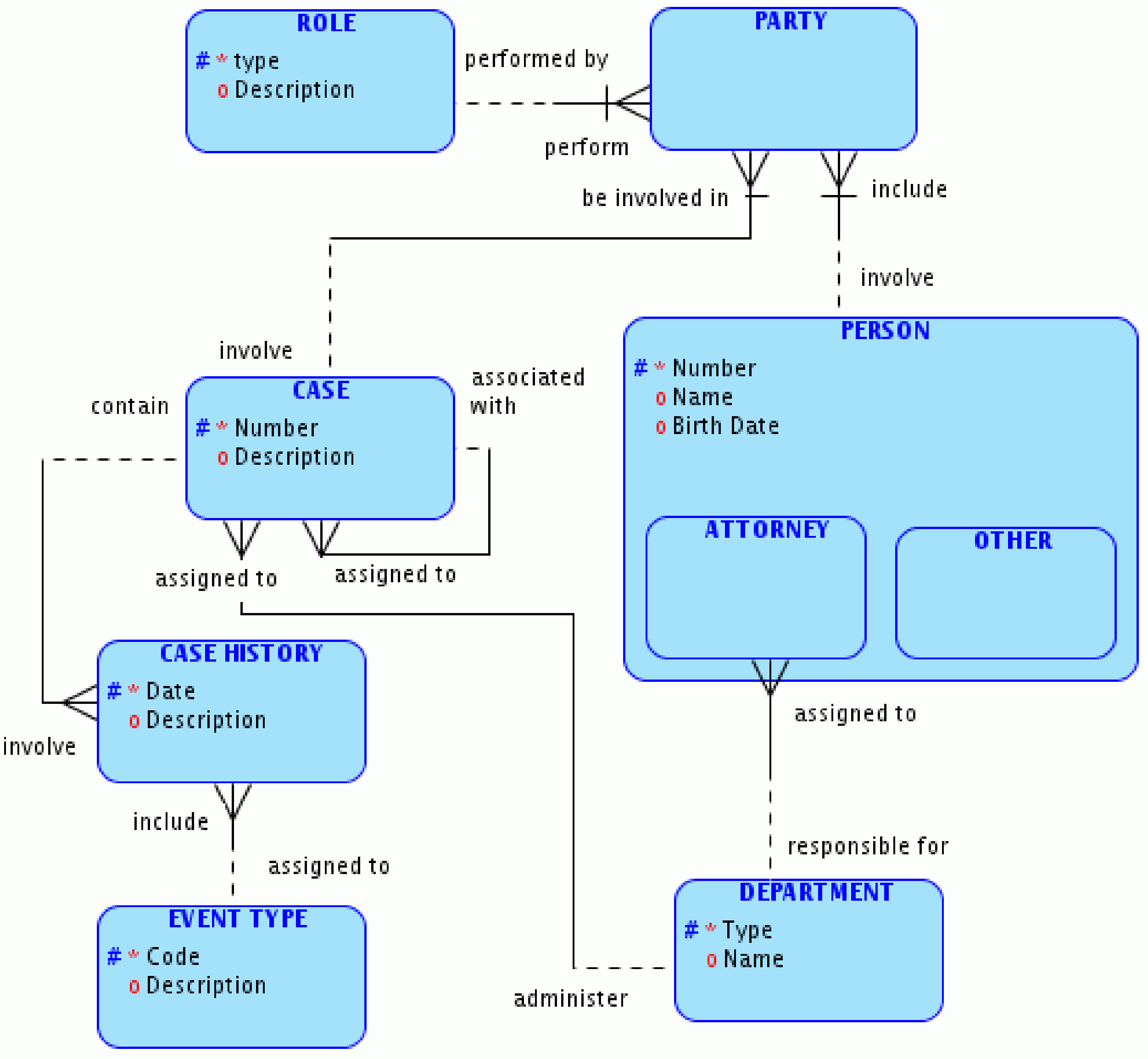
Solution 14-1: Develop and Validate Your ERD (continued)

5) Change the name of the attribute for the relation and click OK.

Some of the attributes in this model are based on a domain. The Domain file has been provided in the solutions directory and can be imported before the solution file is open so that the domains are represented.

Solution 14-2: Develop and Validate Your ERD

One possible solution for this practice is:

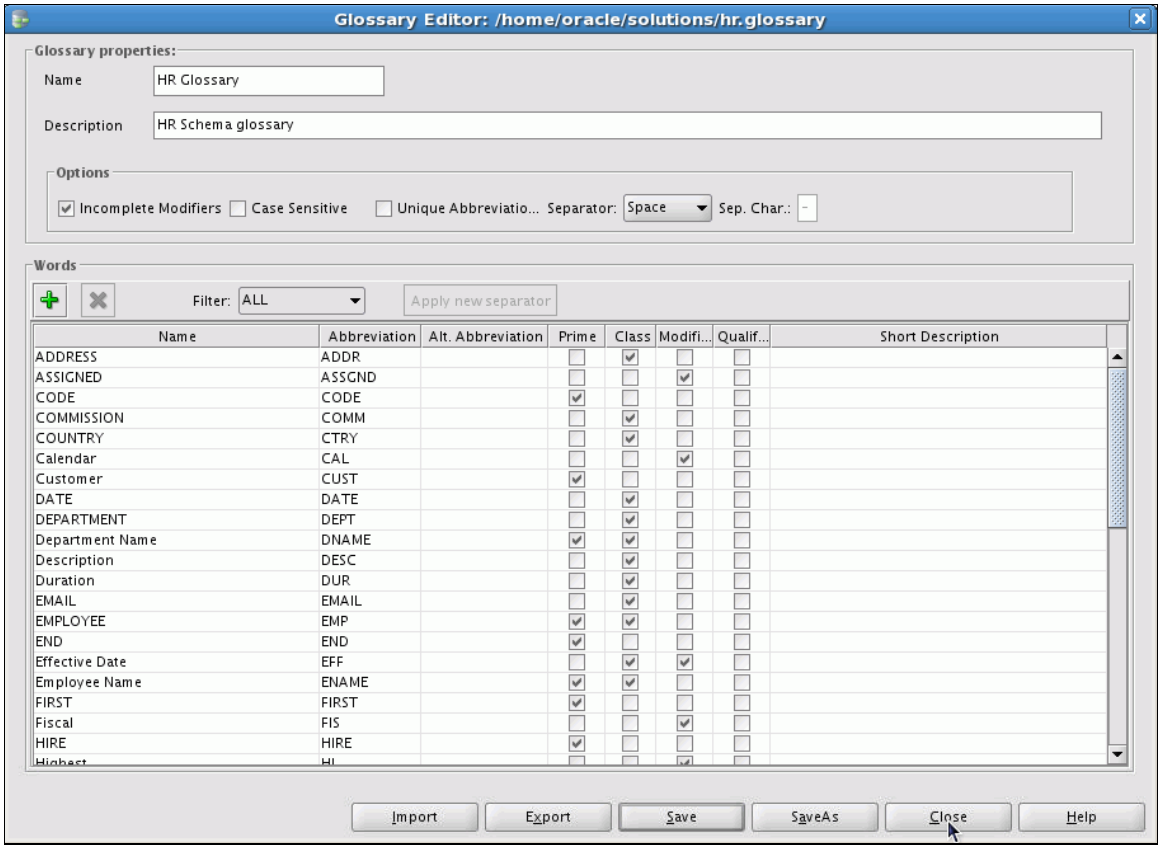


There are no detailed instructions for the solution of this practice because you should be familiar with how to perform the tasks necessary to complete the model.

Solution 15-1: Create an Initial Relational Model

One possible solution is as follows:

1) Open `sol12-1a.xml`.
2) Select Tools > Glossary Editor.
3) Select the `hr.glossary` file in the `/home/oracle/solutions` directory.
4) Review the Words list. When done, click Close.



Glossary Editor: /home/oracle/solutions/hr.glossary

Glossary properties:

Name: HR Glossary

Description: HR Schema glossary

Options

☑ Incomplete Modifiers ☐ Case Sensitive ☐ Unique Abbreviatio... Separator: Space Sep. Char.: -

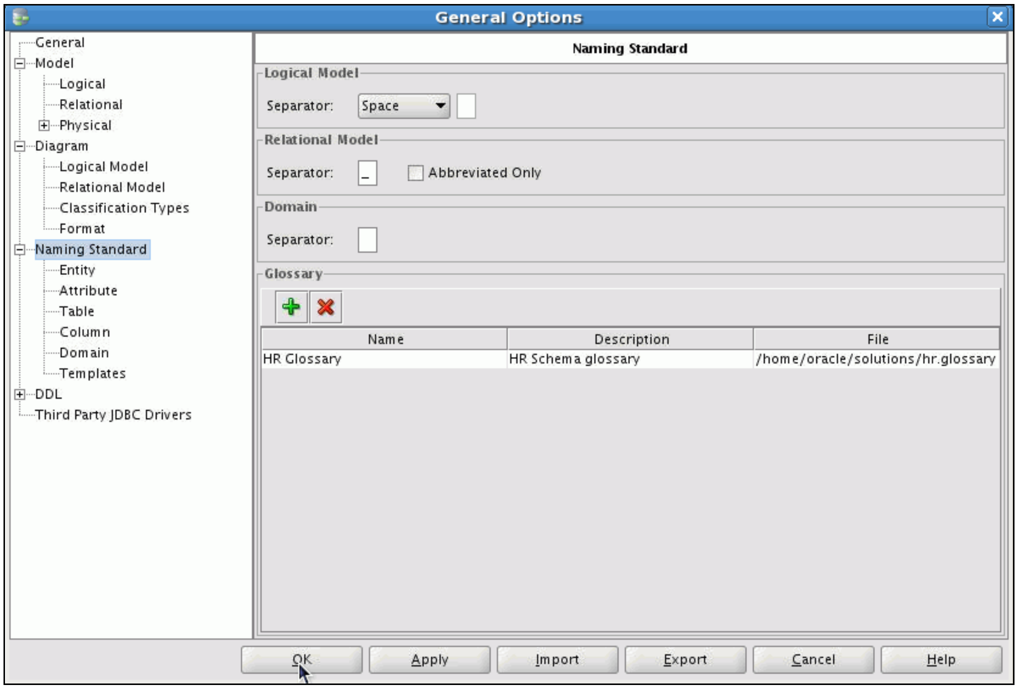
Words

Filter: ALL

Name	Abbreviation	Alt. Abbreviation	Prime	Class	Modifi...	Qualif...	Short Description
ADDRESS	ADDR		☐	☑	☐	☐	
ASSIGNED	ASSGND		☐	☐	☑	☐	
CODE	CODE		☑	☐	☐	☐	
COMMISSION	COMM		☐	☑	☐	☐	
COUNTRY	CTRY		☐	☑	☐	☐	
Calendar	CAL		☐	☐	☑	☐	
Customer	CUST		☑	☐	☐	☐	
DATE	DATE		☐	☑	☐	☐	
DEPARTMENT	DEPT		☐	☑	☐	☐	
Department Name	DNAME		☑	☑	☐	☐	
Description	DESC		☐	☑	☐	☐	
Duration	DUR		☐	☑	☐	☐	
EMAIL	EMAIL		☐	☑	☐	☐	
EMPLOYEE	EMP		☑	☑	☐	☐	
END	END		☑	☐	☐	☐	
Effective Date	EFF		☐	☑	☑	☐	
Employee Name	ENAME		☑	☑	☐	☐	
FIRST	FIRST		☑	☐	☐	☐	
Fiscal	FIS		☐	☐	☑	☐	
HIRE	HIRE		☑	☐	☐	☐	
Highest	HI		☐	☐	☑	☐	

5) Select Tools > General Options.
6) Select Naming Standard.
7) Select the Add ✚ icon under Glossary.
8) Select the `hr.glossary` file in the `/home/oracle/solutions` directory, and click OK.

Solution 15-1: Create an Initial Relational Model (continued)



Naming Standard

Logical Model

Separator: Space

Relational Model

Separator: _ □ Abbreviated Only

Domain

Separator:

Glossary

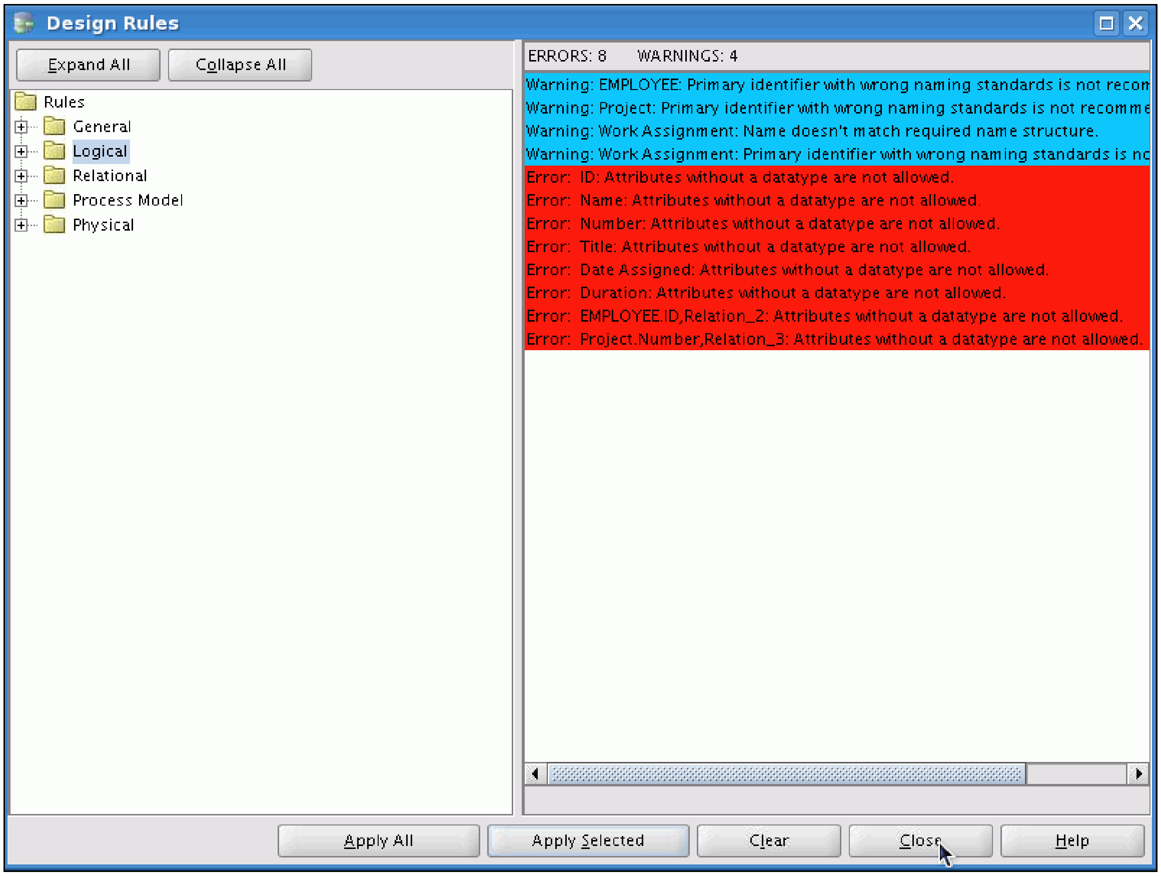
Name	Description	File
HR Glossary	HR Schema glossary	/home/oracle/solutions/hr.glossary

9) Select Tools > Design Rules.

10) Select Logical and click Apply Selected.

11) Review the results. Evaluate the warnings and errors. Errors should be fixed before engineering. If you only have warnings, you may proceed with the engineer. Click Close.

Solution 15-1: Create an Initial Relational Model (continued)



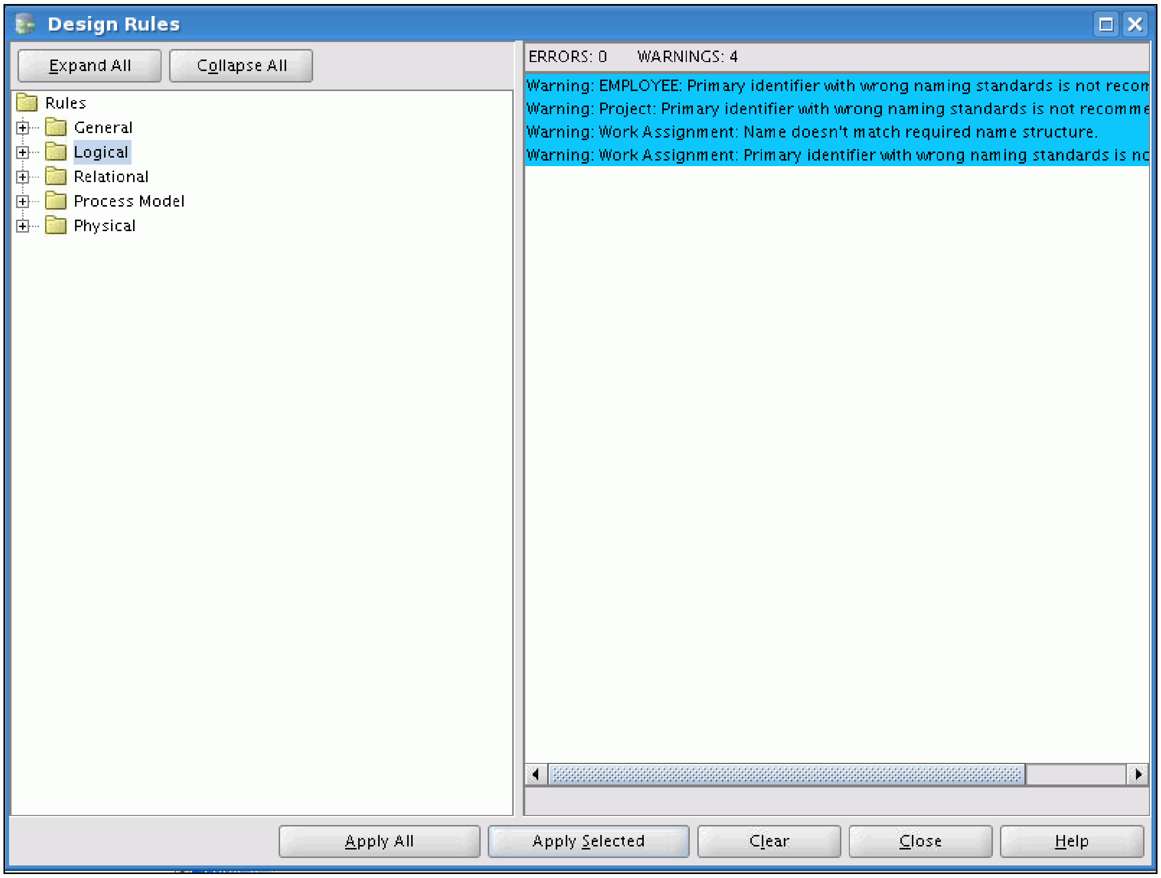
ERRORS: 8 WARNINGS: 4

Warning: EMPLOYEE: Primary identifier with wrong naming standards is not recom
Warning: Project: Primary identifier with wrong naming standards is not recomme
Warning: Work Assignment: Name doesn't match required name structure.
Warning: Work Assignment: Primary identifier with wrong naming standards is nc
Error: ID: Attributes without a datatype are not allowed.
Error: Name: Attributes without a datatype are not allowed.
Error: Number: Attributes without a datatype are not allowed.
Error: Title: Attributes without a datatype are not allowed.
Error: Date Assigned: Attributes without a datatype are not allowed.
Error: Duration: Attributes without a datatype are not allowed.
Error: EMPLOYEE.ID,Relation_2: Attributes without a datatype are not allowed.
Error: Project.Number,Relation_3: Attributes without a datatype are not allowed.

12) Correct the errors by adding the data types indicated in the table below to each of the attributes in each entity (as you did in Lesson 13). You should know how to do this so the steps are not provided. When done, rerun Design Rules so that you see what is shown in the following screenshot.

Entity	Attribute	Datatype
EMPLOYEE	ID	NUMERIC(6)
	Name	VARCHAR (50)
PROJECTS	Number	NUMERIC (6)
	Title	VARCHAR (50)
WORK ASSIGNMENTS	Date Assigned	Date
	Duration	NUMERIC(3)

Solution 15-1: Create an Initial Relational Model (continued)

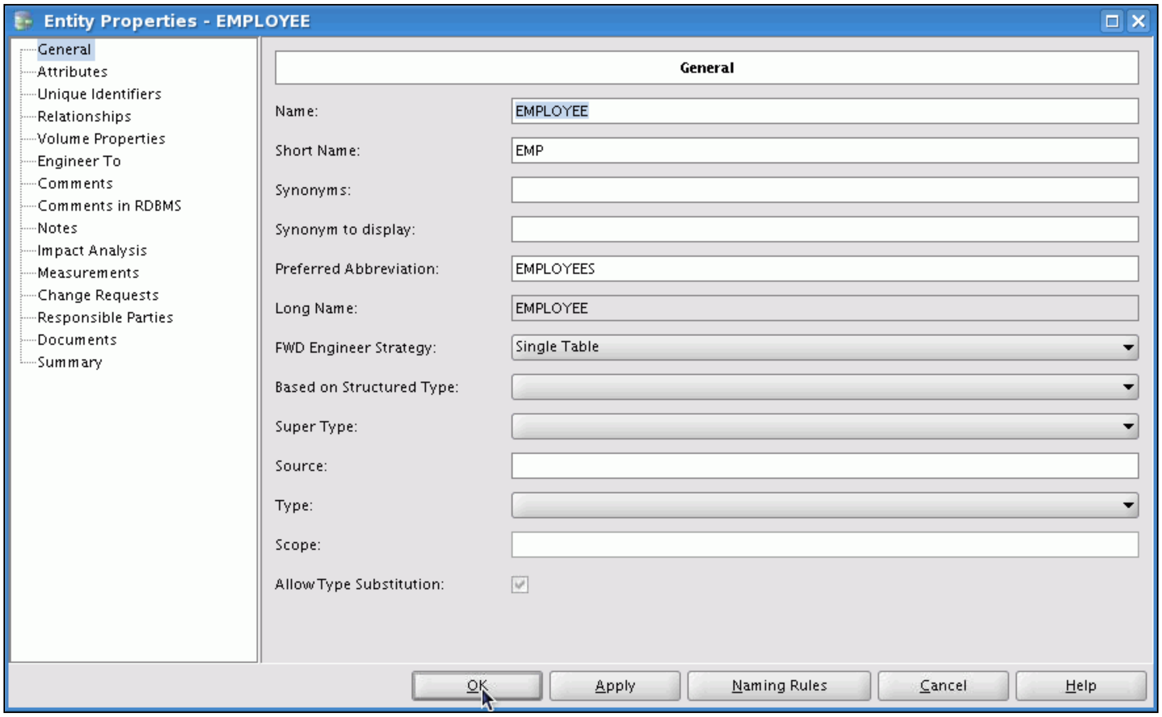


ERRORS: 0 WARNINGS: 4

Warning: EMPLOYEE: Primary identifier with wrong naming standards is not recom
Warning: Project: Primary identifier with wrong naming standards is not recomme
Warning: Work Assignment: Name doesn't match required name structure.
Warning: Work Assignment: Primary identifier with wrong naming standards is nc

13) Add a Short Name and Preferred Abbreviation for each entity. Double-click the EMPLOYEE entity, enter `EMP` for Short Name and `EMPLOYEES` for Preferred Abbreviation, and click OK.

Solution 15-1: Create an Initial Relational Model (continued)



General

Name:	EMPLOYEE
Short Name:	EMP
Synonyms:	
Synonym to display:	
Preferred Abbreviation:	EMPLOYEES
Long Name:	EMPLOYEE
FWD Engineer Strategy:	Single Table
Based on Structured Type:	
Super Type:	
Source:	
Type:	
Scope:	
Allow Type Substitution:	☑

14) Make the following assignments for the other entities:

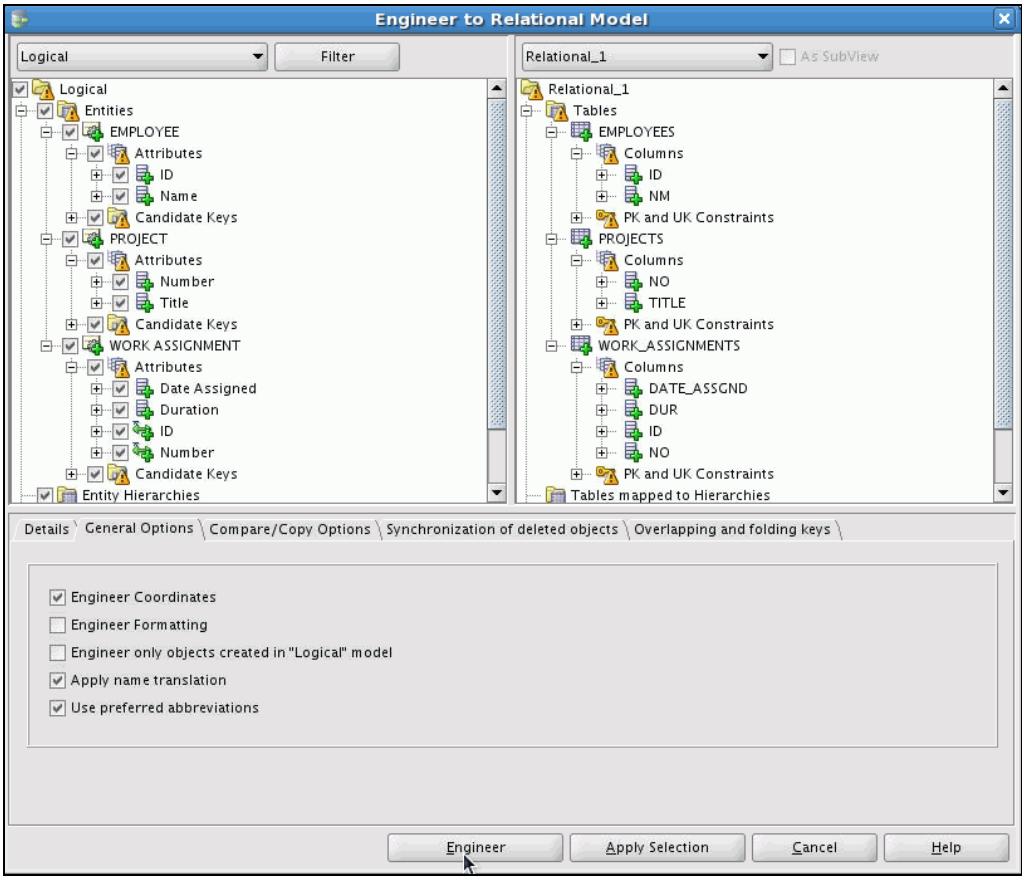
Entity	Short Name	Preferred Abbreviation
PROJECT	PROJ	PROJECTS
WORK ASSIGNMENT	WRKASSGN	WORK_ASSIGNMENTS

15) Now you are ready to engineer your model. Click the “Engineer to Relational Model” » icon.

16) Select the General Options tab in the lower panel, and select “Apply name translation.”

17) Expand the entities to see what the mappings are. If satisfied, click Engineer; otherwise, click Cancel, fix any issues, and try it again.

Solution 15-1: Create an Initial Relational Model (continued)



Engineer to Relational Model

Logical ▾ | Filter | Relational_1 ▾ | □ As SubView

```
☑ Logical
  ☑ Entities
    ☑ EMPLOYEE
      ☑ Attributes
        ☑ ID
        ☑ Name
      ☑ Candidate Keys
    ☑ PROJECT
      ☑ Attributes
        ☑ Number
        ☑ Title
      ☑ Candidate Keys
    ☑ WORK ASSIGNMENT
      ☑ Attributes
        ☑ Date Assigned
        ☑ Duration
        ☑ ID
        ☑ Number
      ☑ Candidate Keys
  ☑ Entity Hierarchies
```

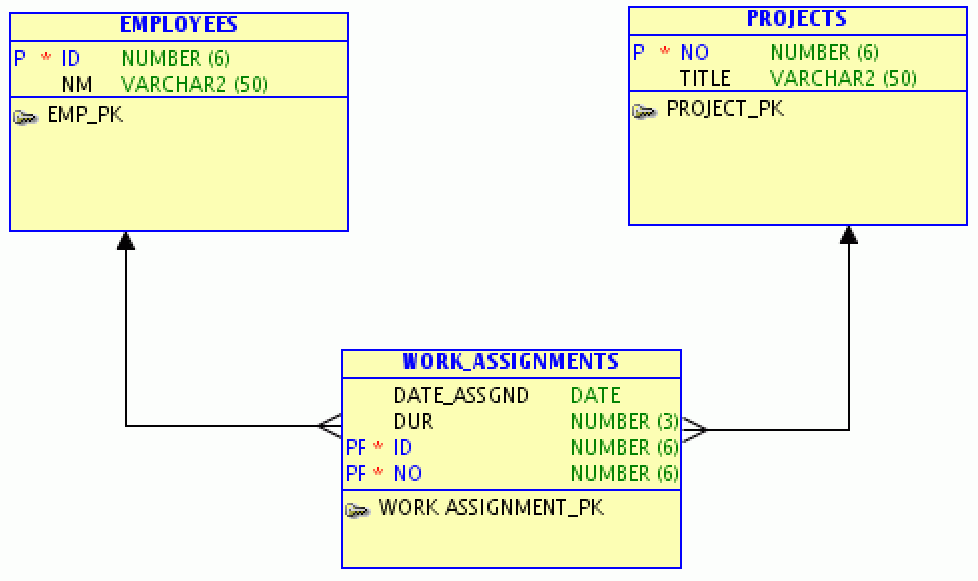
```
Relational_1
  Tables
    EMPLOYEES
      Columns
        ID
        NM
      PK and UK Constraints
    PROJECTS
      Columns
        NO
        TITLE
      PK and UK Constraints
    WORK_ASSIGNMENTS
      Columns
        DATE_ASSGND
        DUR
        ID
        NO
      PK and UK Constraints
  Tables mapped to Hierarchies
```

Details | General Options | Compare/Copy Options | Synchronization of deleted objects | Overlapping and folding keys

☑ Engineer Coordinates
□ Engineer Formatting
□ Engineer only objects created in "Logical" model
☑ Apply name translation
☑ Use preferred abbreviations

Engineer | Apply Selection | Cancel | Help

18) The result relational model is displayed. You may need to move some of the foreign key relation lines to make it look like the following illustration. Also, your column names may be a little different depending on whether you had the “Keep as the name of the Originating Attribute” option set for the FK Attribute name synchronization general option.



EMPLOYEES

P	*	ID	NUMBER (6)
		NM	VARCHAR2 (50)

EMP_PK

PROJECTS

P	*	NO	NUMBER (6)
		TITLE	VARCHAR2 (50)

PROJECT_PK

WORK_ASSIGNMENTS

		DATE_ASSGND	DATE
		DUR	NUMBER (3)
PF	*	ID	NUMBER (6)
PF	*	NO	NUMBER (6)

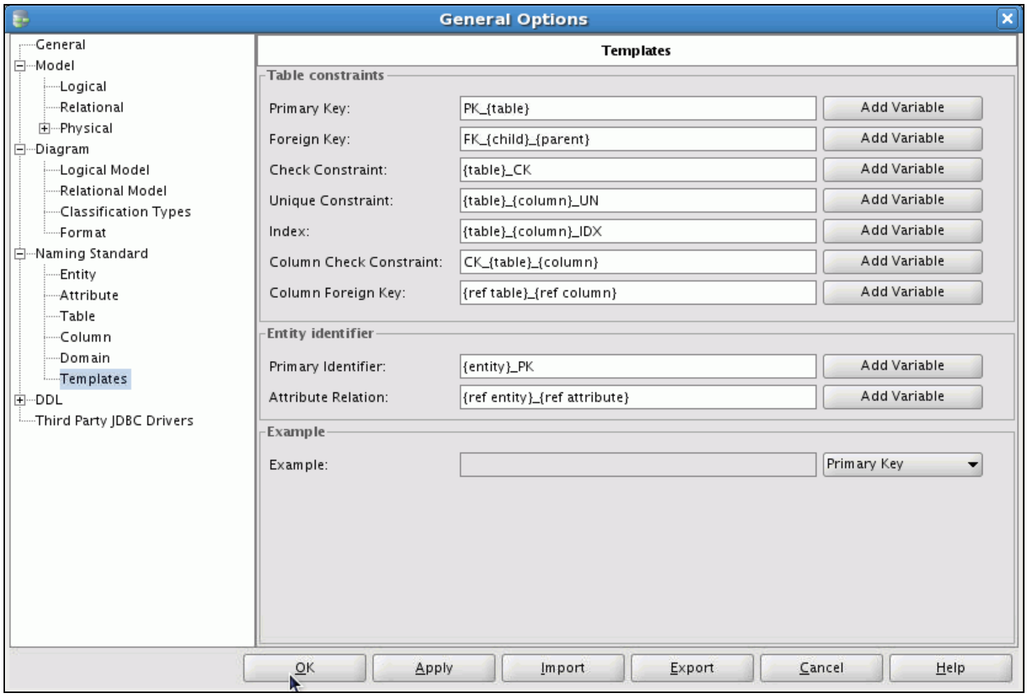
WORK ASSIGNMENT_PK

Solution 15-1: Create an Initial Relational Model (continued)

19) At this point, you want to change the names of the primary keys and foreign key so that PK and FK are prefixes rather than suffixes. Select Tools > General Options.

20) Expand Naming Standards, and select Templates.

21) Change the primary key and foreign key table constraints so that PK and FK are prefixes instead of suffixes as shown in the following screenshot. When done, click OK.



Templates

Table constraints

Primary Key: PK_{table}

Foreign Key: FK_{child}_{parent}

Check Constraint: {table}_CK

Unique Constraint: {table}_{column}_UN

Index: {table}_{column}_IDX

Column Check Constraint: CK_{table}_{column}

Column Foreign Key: {ref table}_{ref column}

Entity identifier

Primary Identifier: {entity}_PK

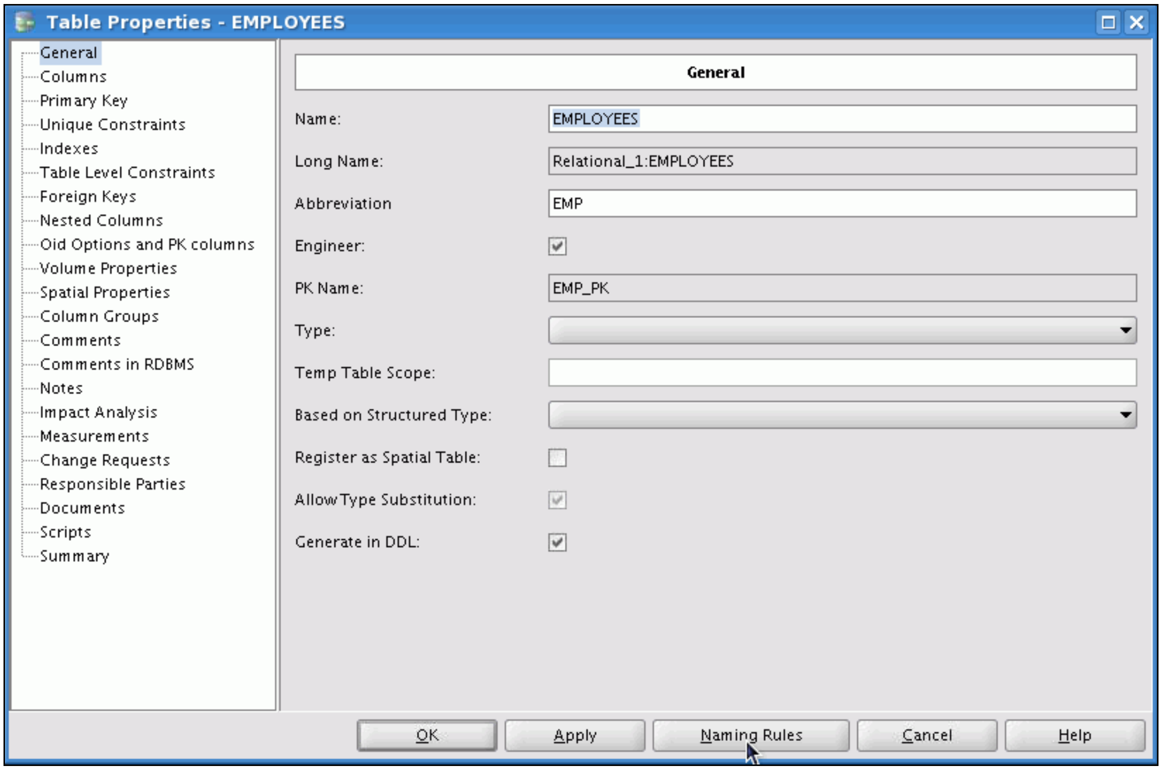
Attribute Relation: {ref entity}_{ref attribute}

Example

Example: Primary Key

22) To apply the naming change to the EMPLOYEES table, double-click the table to open the properties window. Click the Naming Rules button.

Solution 15-1: Create an Initial Relational Model (continued)



Name: EMPLOYEES

Long Name: Relational_1:EMPLOYEES

Abbreviation: EMP

Engineer: ☑

PK Name: EMP_PK

Type:

Temp Table Scope:

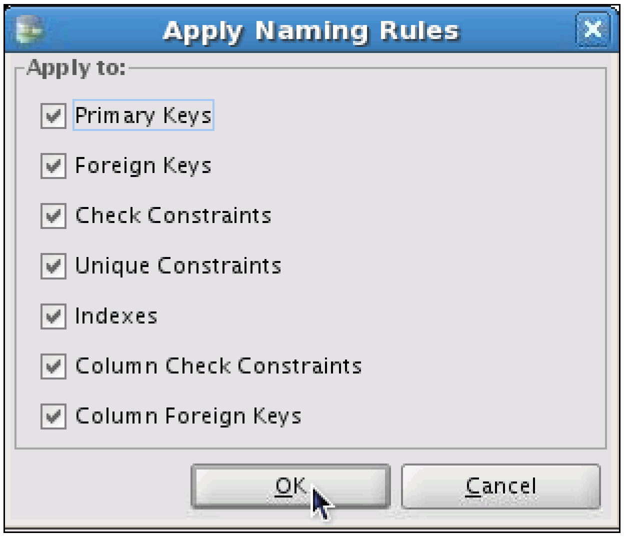
Based on Structured Type:

Register as Spatial Table: ☐

Allow Type Substitution: ☑

Generate in DDL: ☑

23) Click OK to apply the naming rules.



Apply to:

☑ Primary Keys

☑ Foreign Keys

☑ Check Constraints

☑ Unique Constraints

☑ Indexes

☑ Column Check Constraints

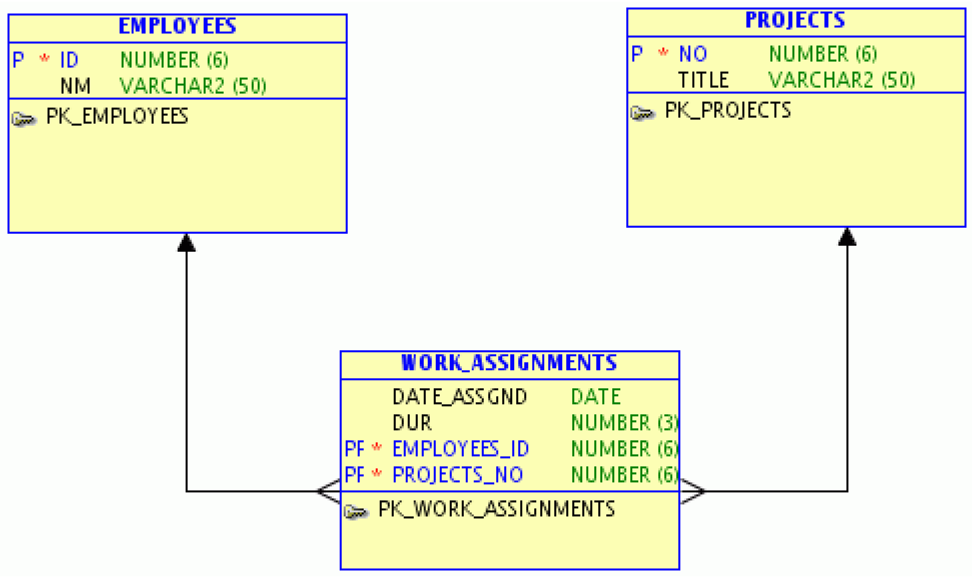
☑ Column Foreign Keys

24) Click OK.

Note that the name of the primary key for EMPLOYEES was EMP_PK (which included the short name and the suffix _PK), and that it changed to PK_EMPLOYEES (which includes PK_ for the prefix and the table name EMPLOYEES) after the template was applied.

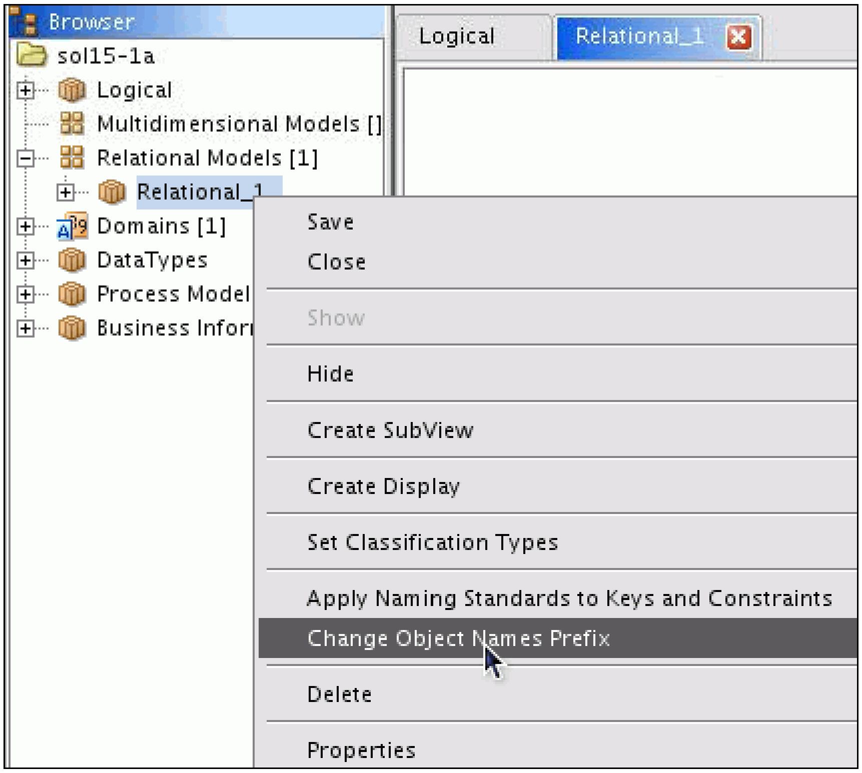
25) Perform the previous steps for each entity so that the result looks as follows.

Solution 15-1: Create an Initial Relational Model (continued)



26) Lastly, you want to add the OU_ prefix to each table. Expand Relational Models in the object browser.

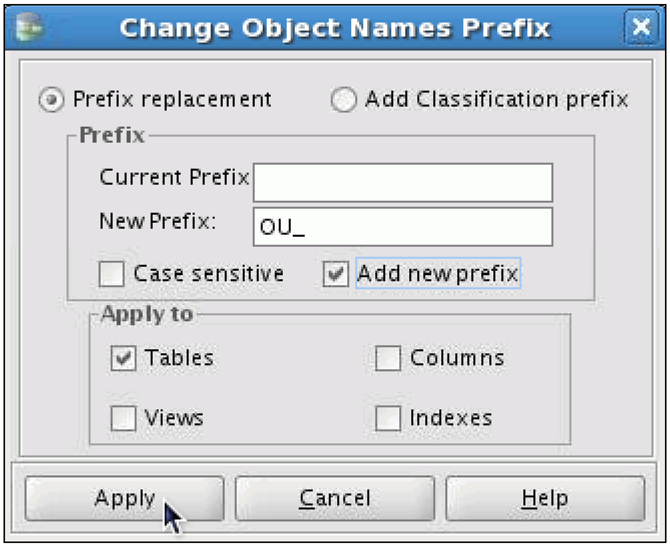
27) Right-click Relational_1 and select Change Object Names Prefix.



28) Enter `OU_` in the New Prefix field and select the "Add new prefix" check box.

29) Make sure that Tables is checked for "Apply to," and click Apply.

Solution 15-1: Create an Initial Relational Model (continued)



Change Object Names Prefix

● Prefix replacement ○ Add Classification prefix

Prefix

Current Prefix:

New Prefix: OU_

☐ Case sensitive ☑ Add new prefix

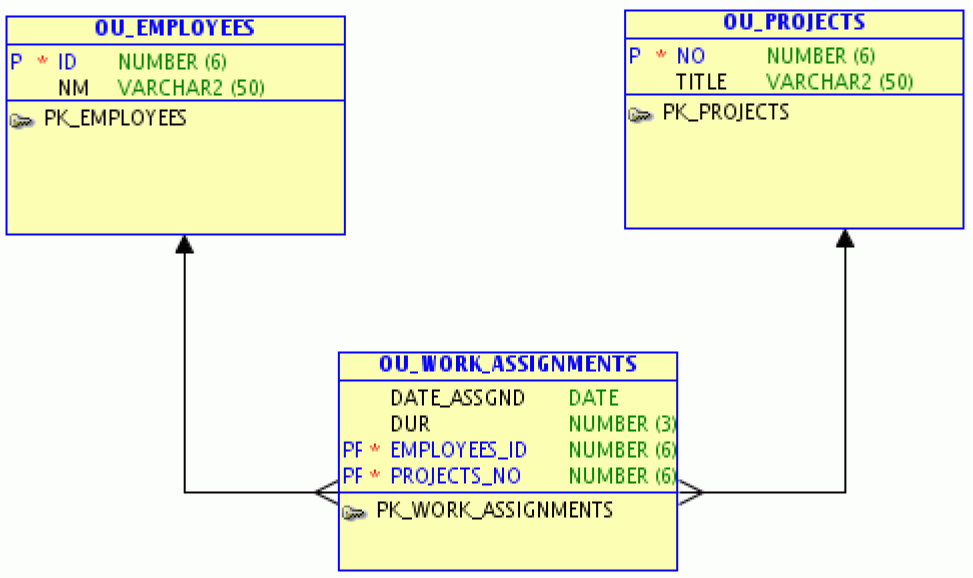
Apply to

☑ Tables ☐ Columns

☐ Views ☐ Indexes

30) Click OK to confirm that three changes were made.

31) Your relational model should look as follows.



Solution 15-2: Forward Engineer a Model

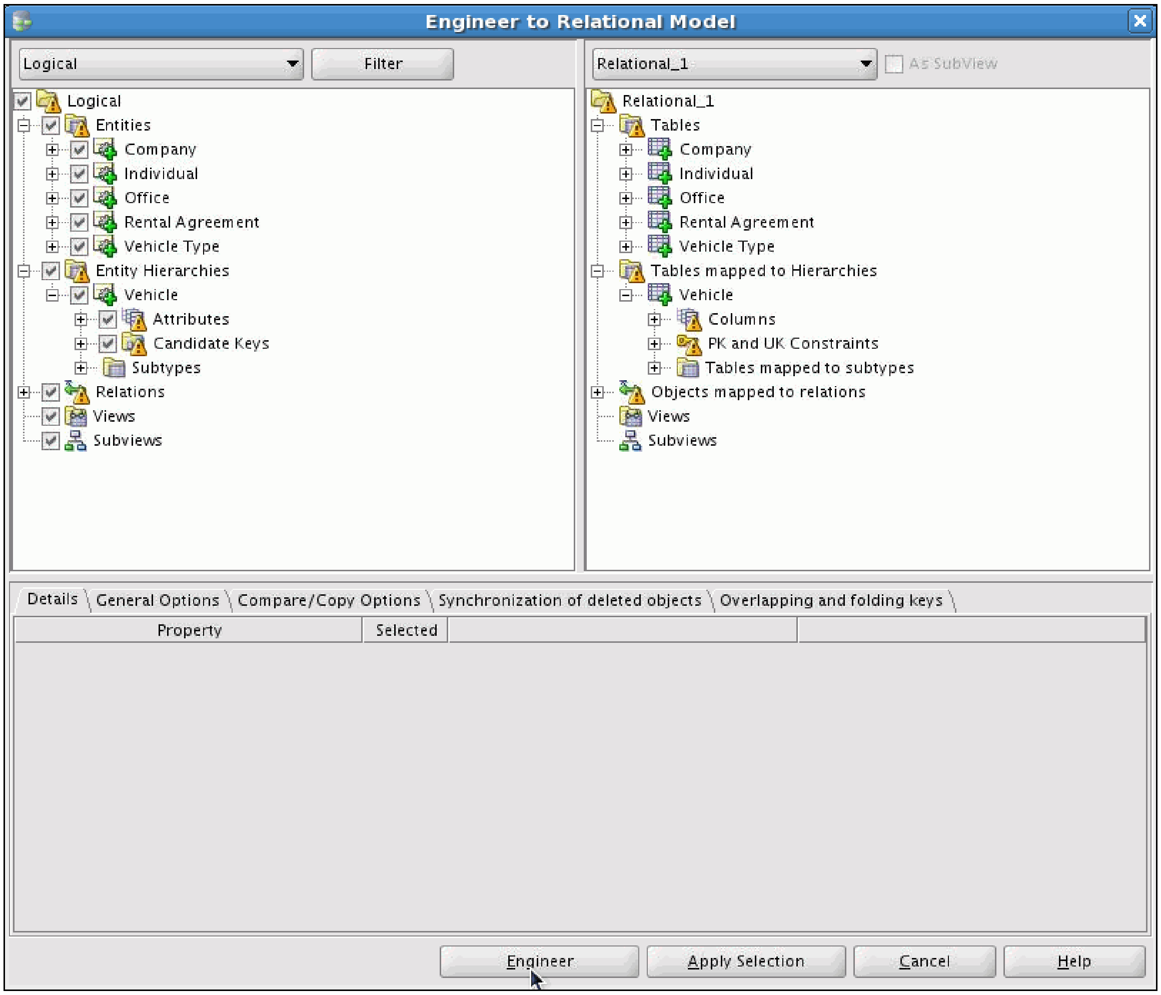
One possible solution to this practice is:

1) Open the solution for Practice 13-1 (`sol13-1.xml`).

2) Double-click the Vehicle super type entity. Make sure that FWD Strategy is set to Single Table.

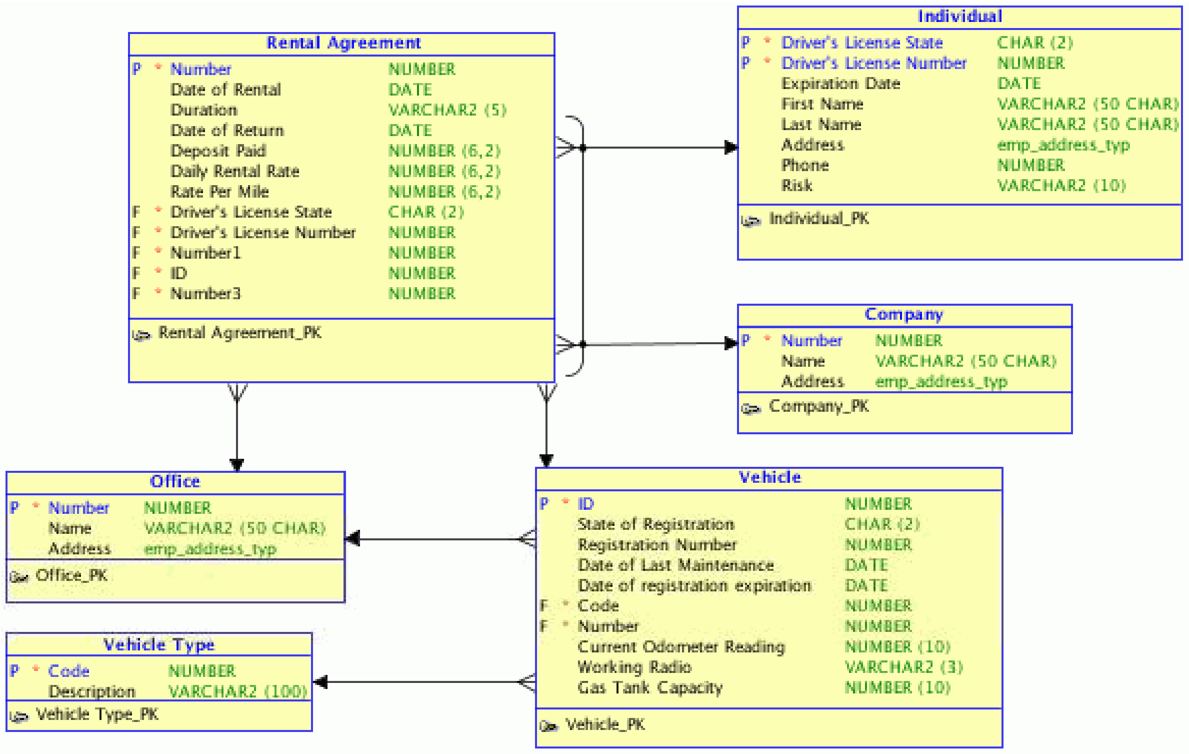
3) Click the “Engineer to Relational Model” icon.

4) Expand the nodes to make sure that the engineering process produces the correct results, and then click Engineer.



5) A relational model similar to the following will be displayed.

Solution 15-2: Forward Engineer a Model (continued)



Rental Agreement

P *	Number	NUMBER
	Date of Rental	DATE
	Duration	VARCHAR2 (5)
	Date of Return	DATE
	Deposit Paid	NUMBER (6,2)
	Daily Rental Rate	NUMBER (6,2)
	Rate Per Mile	NUMBER (6,2)
F *	Driver's License State	CHAR (2)
F *	Driver's License Number	NUMBER
F *	Number1	NUMBER
F *	ID	NUMBER
F *	Number3	NUMBER

Rental Agreement_PK

Individual

P *	Driver's License State	CHAR (2)
P *	Driver's License Number	NUMBER
	Expiration Date	DATE
	First Name	VARCHAR2 (50 CHAR)
	Last Name	VARCHAR2 (50 CHAR)
	Address	emp_address_typ
	Phone	NUMBER
	Risk	VARCHAR2 (10)

Individual_PK

Company

P *	Number	NUMBER
	Name	VARCHAR2 (50 CHAR)
	Address	emp_address_typ

Company_PK

Office

P *	Number	NUMBER
	Name	VARCHAR2 (50 CHAR)
	Address	emp_address_typ

Office_PK

Vehicle Type

P *	Code	NUMBER
	Description	VARCHAR2 (100)

Vehicle Type_PK

Vehicle

P *	ID	NUMBER
	State of Registration	CHAR (2)
	Registration Number	NUMBER
	Date of Last Maintenance	DATE
	Date of registration expiration	DATE
F *	Code	NUMBER
F *	Number	NUMBER
	Current Odometer Reading	NUMBER (10)
	Working Radio	VARCHAR2 (3)
	Gas Tank Capacity	NUMBER (10)

Vehicle_PK

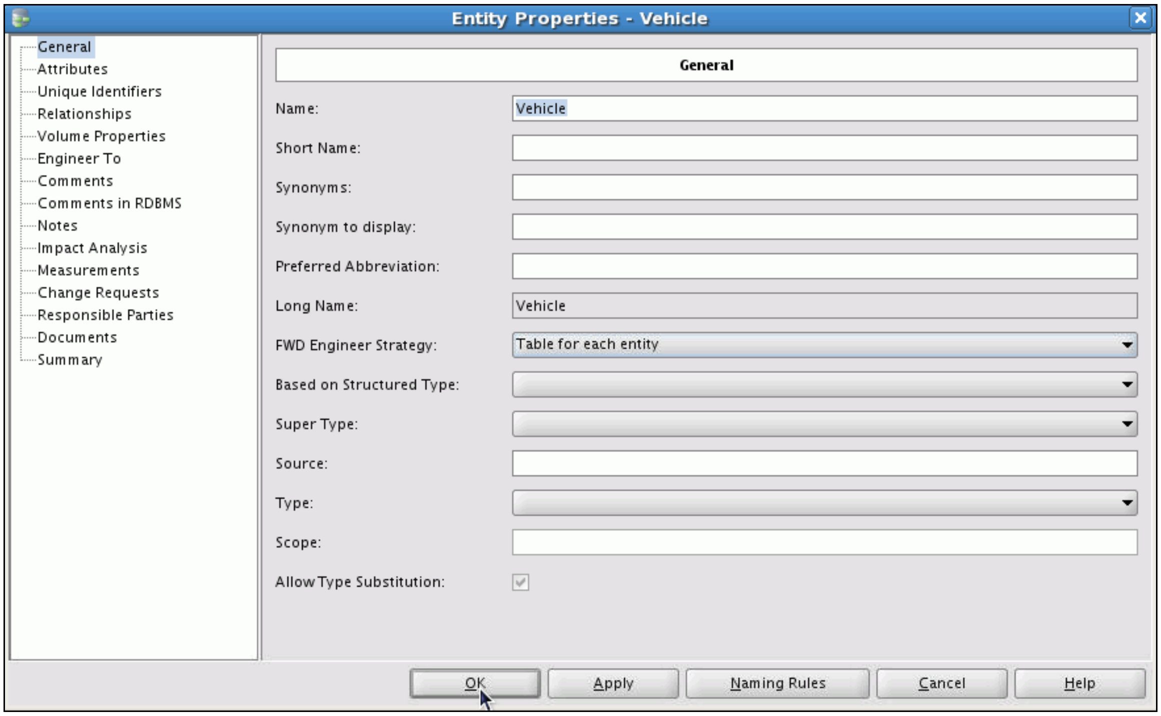
6) You can create another relational model, change the FWD strategy for the entity type hierarchy, and then engineer again. In the object browser, right-click Relational Model and select New Relational Model. A new relational model, Relational_2, is created.

7) Click the Logical tab.

8) To change the FWD engineering strategy for the entity type hierarchy, double-click the Vehicle entity.

9) Change FWD Engineering Strategy to “Table for each entity,” and click OK.

Solution 15-2: Forward Engineer a Model (continued)



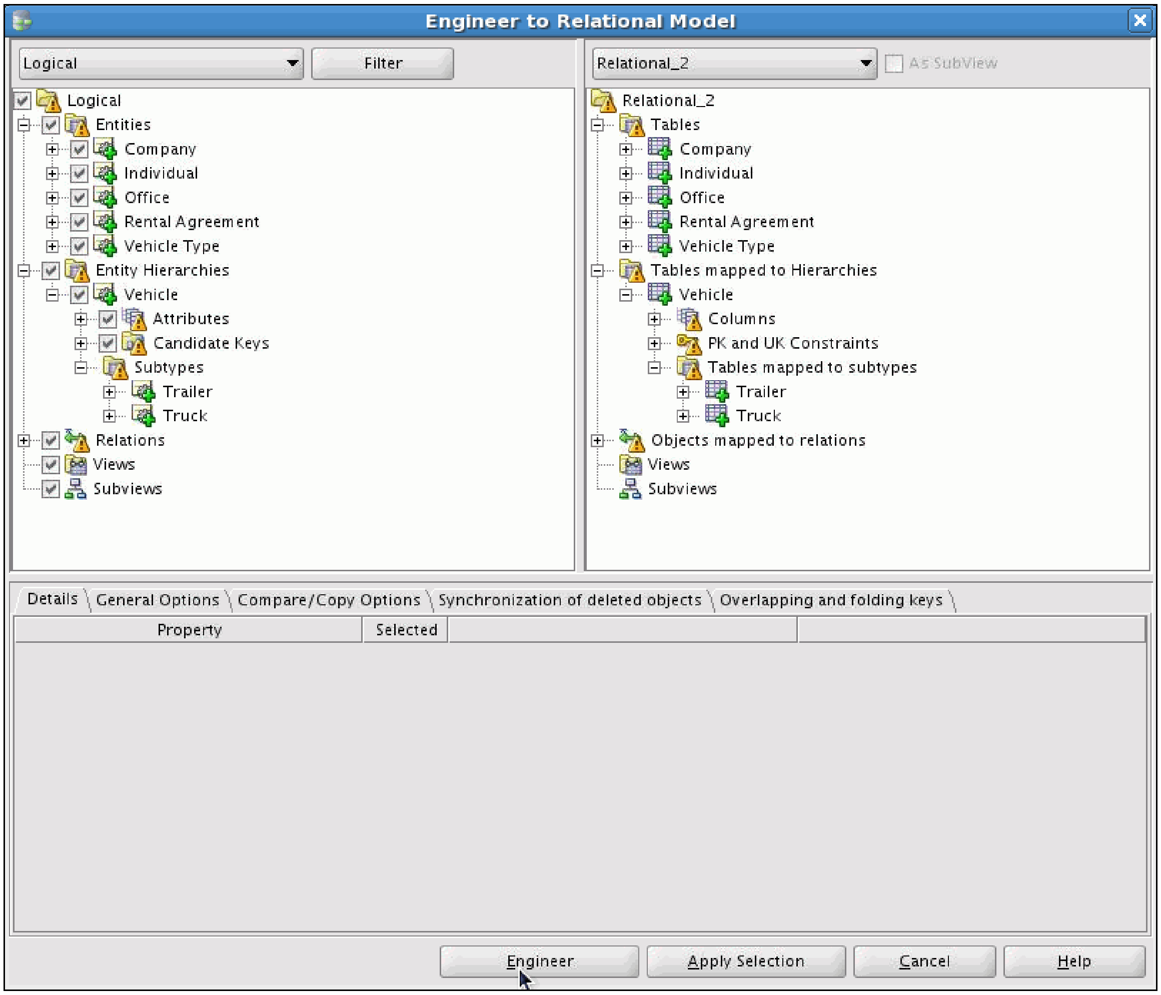
General

Name:	Vehicle
Short Name:	
Synonyms:	
Synonym to display:	
Preferred Abbreviation:	
Long Name:	Vehicle
FWD Engineer Strategy:	Table for each entity
Based on Structured Type:	
Super Type:	
Source:	
Type:	
Scope:	
Allow Type Substitution:	☑

10) Click the “Engineer to Relational Model” » icon.

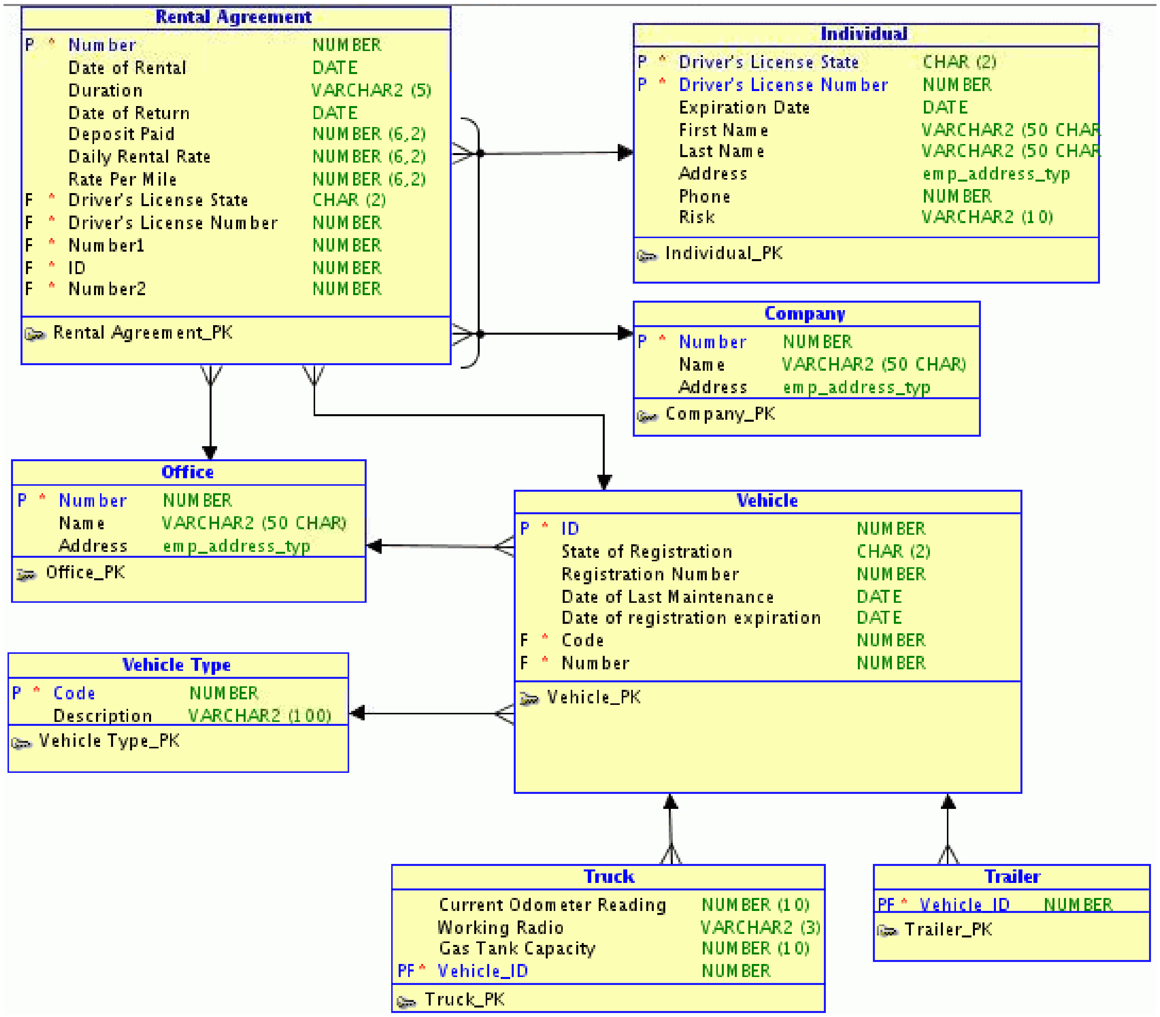
11) Make sure that the Relational Model is set to Relational_2. Expand the nodes to see what will happen when you click Engineer. Notice in this case that two tables will be created for the entity type hierarchy. Click Engineer.

Solution 15-2: Forward Engineer a Model (continued)



12) A relational model similar to the following is displayed.

Solution 15-2: Forward Engineer a Model (continued)



Rental Agreement

P *	Number	NUMBER
	Date of Rental	DATE
	Duration	VARCHAR2 (5)
	Date of Return	DATE
	Deposit Paid	NUMBER (6,2)
	Daily Rental Rate	NUMBER (6,2)
	Rate Per Mile	NUMBER (6,2)
F *	Driver's License State	CHAR (2)
F *	Driver's License Number	NUMBER
F *	Number1	NUMBER
F *	ID	NUMBER
F *	Number2	NUMBER

Rental Agreement_PK

Individual

P *	Driver's License State	CHAR (2)
P *	Driver's License Number	NUMBER
	Expiration Date	DATE
	First Name	VARCHAR2 (50 CHAR
	Last Name	VARCHAR2 (50 CHAR
	Address	emp_address_typ
	Phone	NUMBER
	Risk	VARCHAR2 (10)

Individual_PK

Company

P *	Number	NUMBER
	Name	VARCHAR2 (50 CHAR)
	Address	emp_address_typ

Company_PK

Office

P *	Number	NUMBER
	Name	VARCHAR2 (50 CHAR)
	Address	emp_address_typ

Office_PK

Vehicle

P *	ID	NUMBER
	State of Registration	CHAR (2)
	Registration Number	NUMBER
	Date of Last Maintenance	DATE
	Date of registration expiration	DATE
F *	Code	NUMBER
F *	Number	NUMBER

Vehicle_PK

Vehicle Type

P *	Code	NUMBER
	Description	VARCHAR2 (100)

Vehicle Type_PK

Truck

	Current Odometer Reading	NUMBER (10)
	Working Radio	VARCHAR2 (3)
	Gas Tank Capacity	NUMBER (10)
PF*	Vehicle_ID	NUMBER

Truck_PK

Trailer

PF *	Vehicle_ID	NUMBER

Trailer_PK

Solution 16-1: Analyze Your Relational Model

For the following relational model, add or modify existing design components based on the following requirements.

A. Want to know how many passengers were on a particular cruise for each month

To accomplish this requirement, you could create a view on the Passenger and Cruise tables. Perform the following steps:

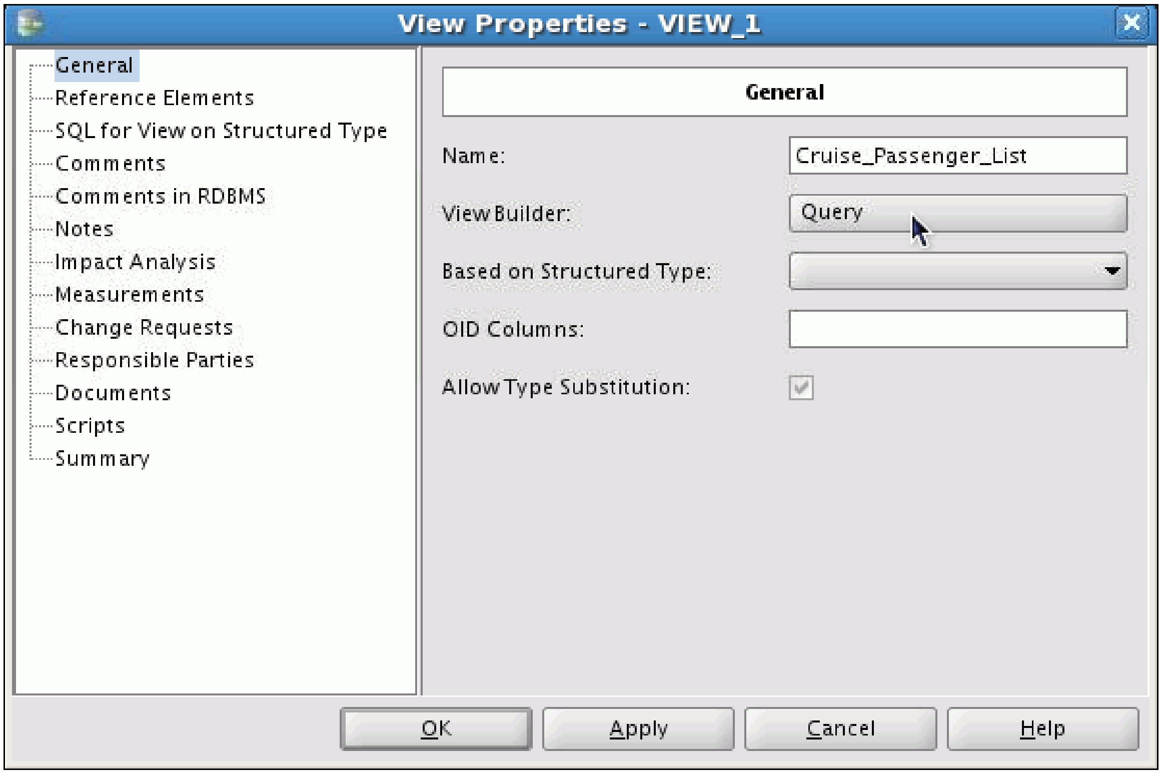
1) Open `sol14-1.xml`.

2) Forward engineer the model to create the relational model.

3) Add columns to the Cruise table to store start_date and end_date. In the Relational Model diagram, double-click the Cruise table and select the Columns property in the left navigator. Add the two columns, and click OK.

4) Add a view on the Passenger and Cruise tables. Click the New View icon and click in the white space of the relational diagram.

5) Enter `Cruise_Passenger_List` for Name, and click the Query button (because you are going to base this view on a query).



General

Name: Cruise_Passenger_List

View Builder: Query

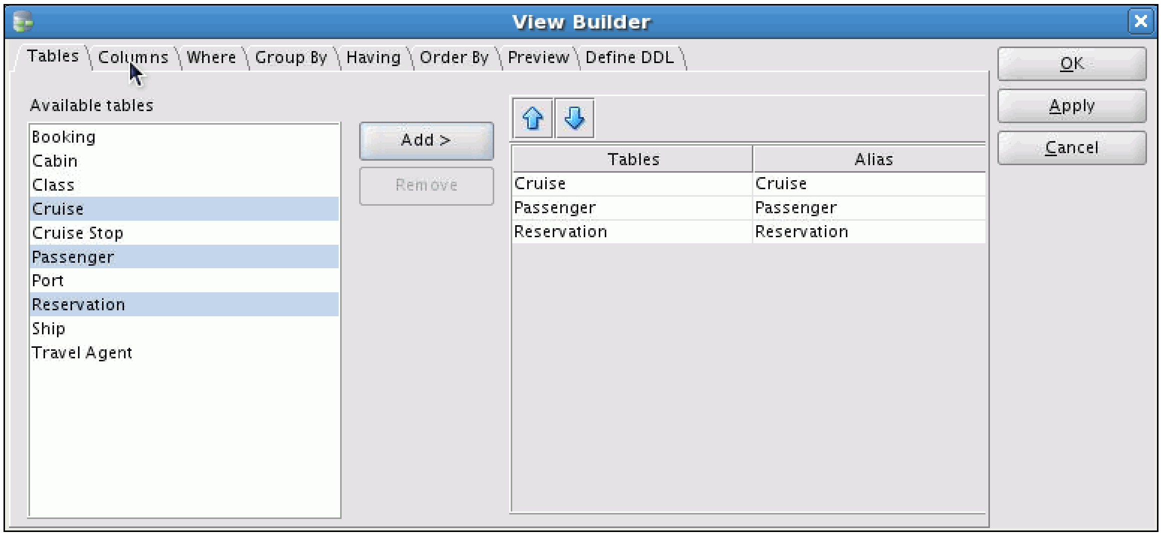
Based on Structured Type:

OID Columns:

Allow Type Substitution: ☑

6) Select the Cruise, Passenger, and Reservation tables, and click Add. Then click the Columns tab.

Solution 16-1: Analyze Your Relational Model (continued)

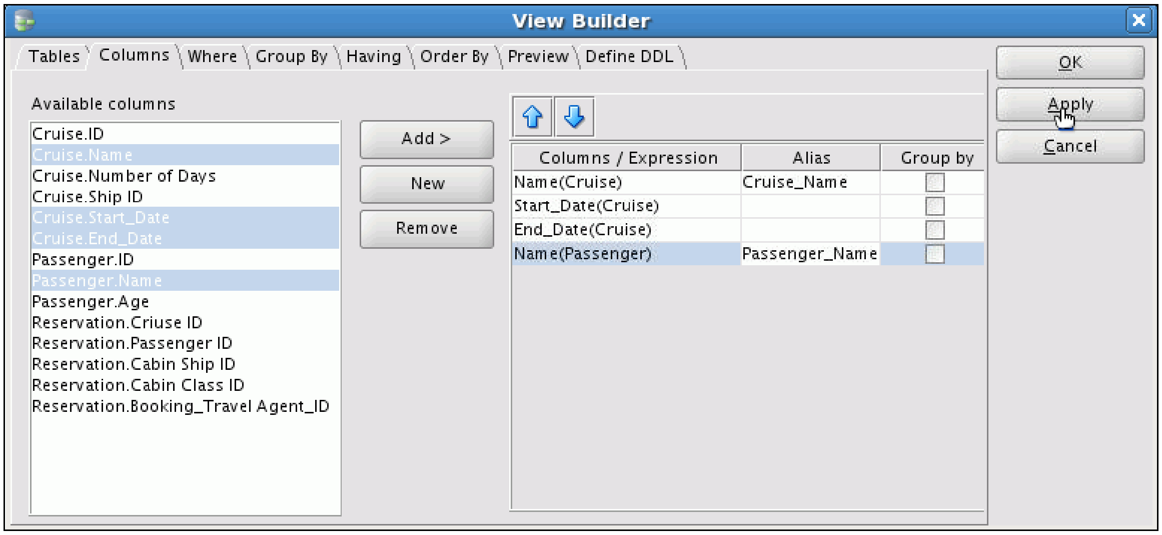


Available tables

- Booking
- Cabin
- Class
- Cruise
- Cruise Stop
- Passenger
- Port
- Reservation
- Ship
- Travel Agent

Tables	Alias
Cruise	Cruise
Passenger	Passenger
Reservation	Reservation

7) Select Name from both the Cruise and Passenger tables and Start_Date and End_Date from the Cruise table. Because the Name column in both Cruise and Passenger are the same, you must create an alias for each in the alias field. Select the Name(Cruise) column in the list, and then double-click the Alias field and change it to Cruise_Name. Do the same for Name(Passenger) only call it `Passenger_Name`. Then click Apply.



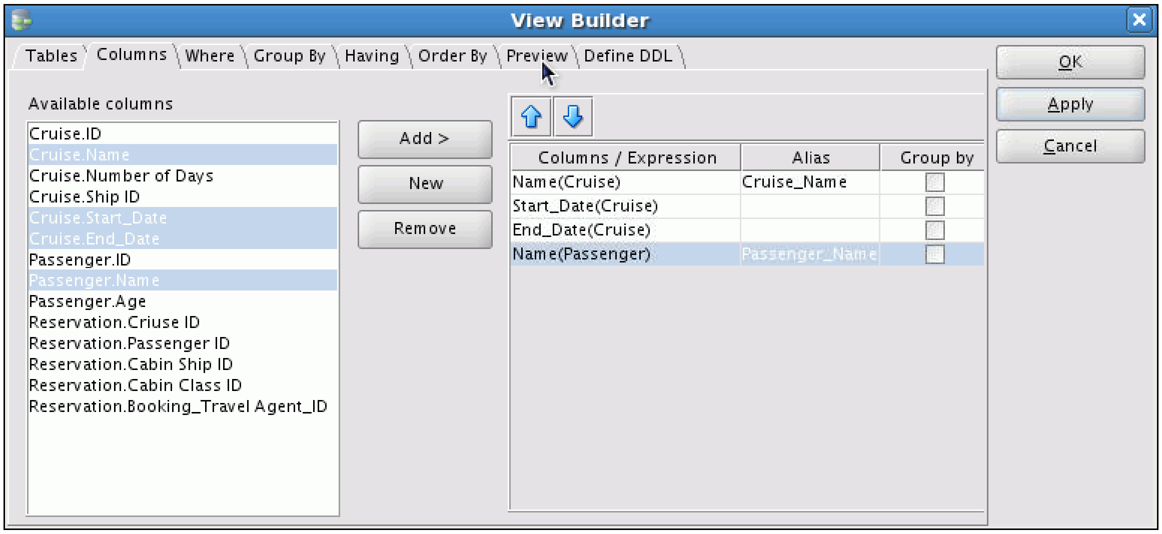
Available columns

- Cruise.ID
- Cruise.Name
- Cruise.Number of Days
- Cruise.Ship ID
- Cruise.Start_Date
- Cruise.End_Date
- Passenger.ID
- Passenger.Name
- Passenger.Age
- Reservation.Cruise ID
- Reservation.Passenger ID
- Reservation.Cabin Ship ID
- Reservation.Cabin Class ID
- Reservation.Booking_Travel Agent_ID

Columns / Expression	Alias	Group by
Name(Cruise)	Cruise_Name	□
Start_Date(Cruise)		□
End_Date(Cruise)		□
Name(Passenger)	Passenger_Name	□

8) To review the DDL that will be generated click the Preview tab.

Solution 16-1: Analyze Your Relational Model (continued)

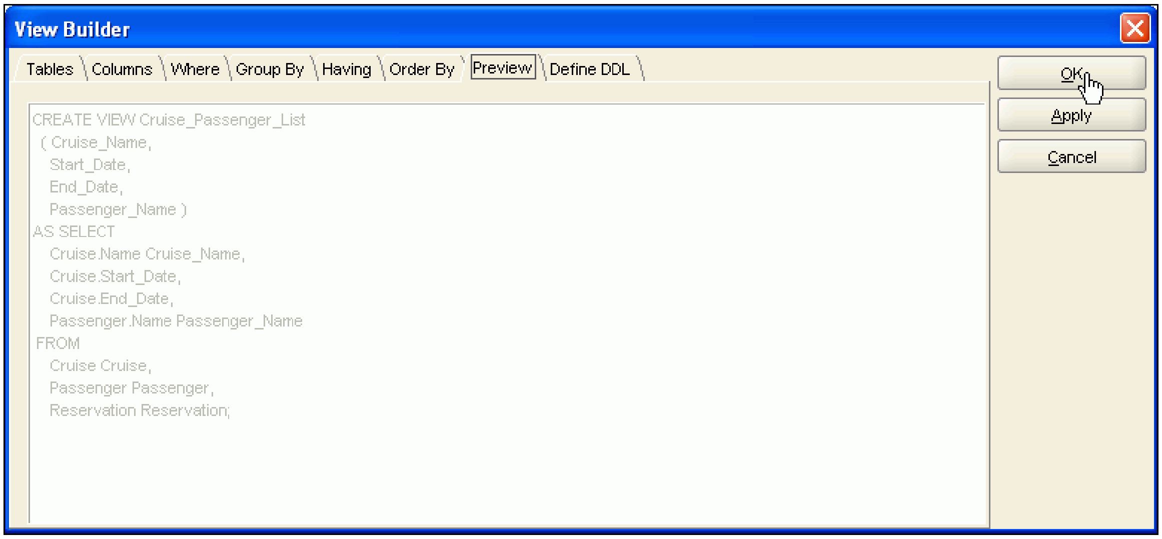


Available columns:

- Cruise.ID
- Cruise.Name
- Cruise.Number of Days
- Cruise.Ship ID
- Cruise.Start_Date
- Cruise.End_Date
- Passenger.ID
- Passenger.Name
- Passenger.Age
- Reservation.Criuse ID
- Reservation.Passenger ID
- Reservation.Cabin Ship ID
- Reservation.Cabin Class ID
- Reservation.Booking_Travel Agent_ID

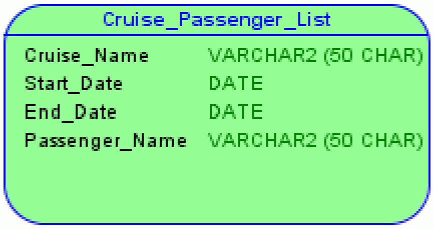
Columns / Expression	Alias	Group by
Name(Cruise)	Cruise_Name	□
Start_Date(Cruise)		□
End_Date(Cruise)		□
Name(Passenger)	Passenger_Name	□

9) Review the DDL. If you do not see the alias names, click Apply. When done, click OK.



```
CREATE VIEW Cruise_Passenger_List
 ( Cruise_Name,
   Start_Date,
   End_Date,
   Passenger_Name )
AS SELECT
   Cruise.Name Cruise_Name,
   Cruise.Start_Date,
   Cruise.End_Date,
   Passenger.Name Passenger_Name
FROM
   Cruise Cruise,
   Passenger Passenger,
   Reservation Reservation;
```

10) Click OK to create the view.



Cruise_Passenger_List	
Cruise_Name	VARCHAR2 (50 CHAR)
Start_Date	DATE
End_Date	DATE
Passenger_Name	VARCHAR2 (50 CHAR)

Solution 16-1: Analyze Your Relational Model (continued)

B. Want to quickly see the cruises that a particular ship has made

To accomplish this requirement, you could create an index on the SHIP_ID foreign key in the Cruise table. Perform the following steps:

1) Add an index on the SHIP_ID foreign key in the Cruise table. Double-click the Cruise table.
2) Select the Indexes property in the left navigator.
3) Click the Add + icon.
4) Select the Ship_ID column, and click the arrow to move the column over to the right. Then click OK.

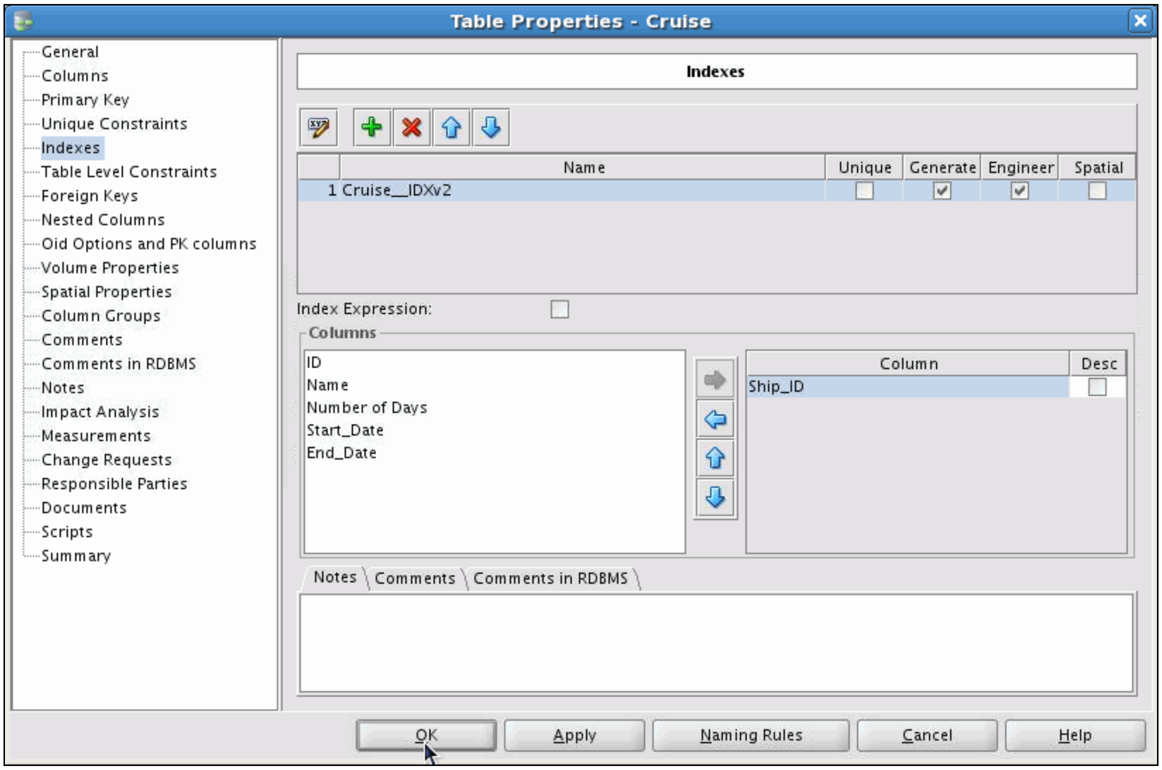


Table Properties - Cruise

- General
- Columns
- Primary Key
- Unique Constraints
- Indexes
- Table Level Constraints
- Foreign Keys
- Nested Columns
- Oid Options and PK columns
- Volume Properties
- Spatial Properties
- Column Groups
- Comments
- Comments in RDBMS
- Notes
- Impact Analysis
- Measurements
- Change Requests
- Responsible Parties
- Documents
- Scripts
- Summary

Indexes

	Name	Unique	Generate	Engineer	Spatial
1	Cruise__IDXv2	□	☑	☑	□

Index Expression: □

Columns

ID
Name
Number of Days
Start_Date
End_Date

Column	Desc
Ship_ID	□

5) Your index was created.

C. Want to know how well each cruise did as far as revenue

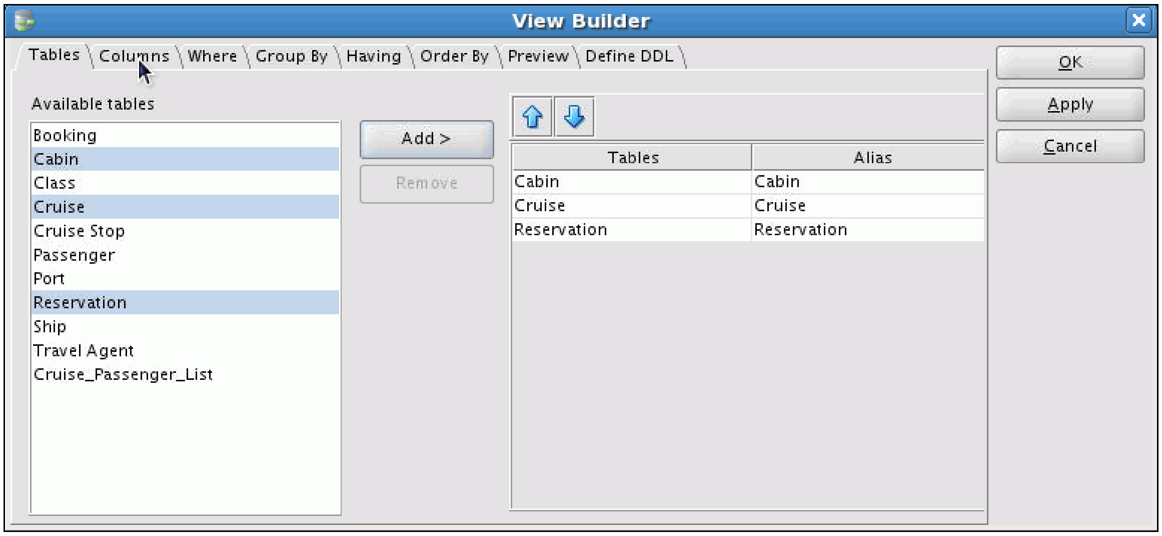
To accomplish this requirement, you could create a view between Reservation, Cruise, and Cabin where you sum the total cabins for a particular cruise. Perform the following steps:

1) Add a view on Reservation, Cruise, and Cabin. Click the New View icon and click in the white space of the relational diagram.

Solution 16-1: Analyze Your Relational Model (continued)

2) Enter `Cruise_Revenue` for Name, and click the Query button (because you are going to base this view on a query).

3) Select the Cabin, Cruise, and Reservation tables, and click Add. Then click the Columns tab.

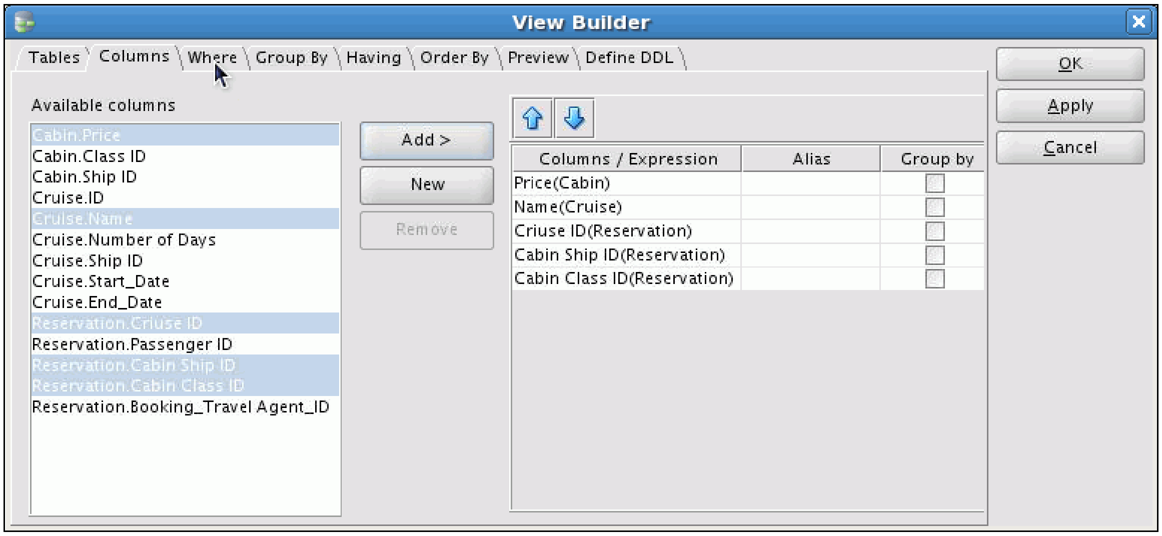


Available tables

- Booking
- Cabin
- Class
- Cruise
- Cruise Stop
- Passenger
- Port
- Reservation
- Ship
- Travel Agent
- Cruise_Passenger_List

Tables	Alias
Cabin	Cabin
Cruise	Cruise
Reservation	Reservation

4) Select the Price, Name, Cruise_ID, Cabin_Ship_ID, and Cabin_Class_ID columns and click Add to move them to the right side of the window. Then click the Where tab.



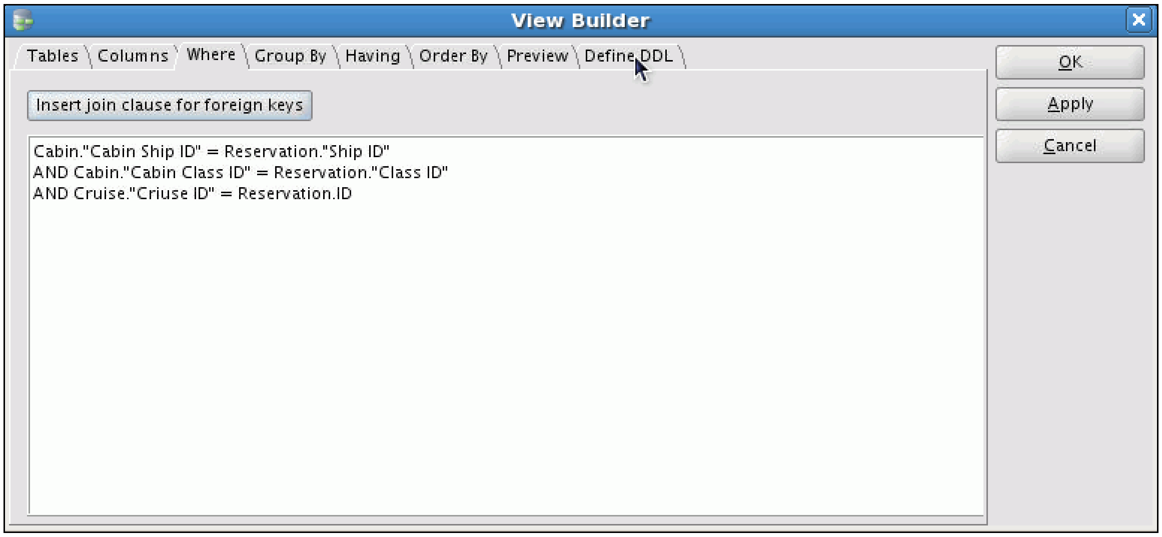
Available columns

- Cabin.Price
- Cabin.Class ID
- Cabin.Ship ID
- Cruise.ID
- Cruise.Name
- Cruise.Number of Days
- Cruise.Ship ID
- Cruise.Start_Date
- Cruise.End_Date
- Reservation.Cruise ID
- Reservation.Passenger ID
- Reservation.Cabin Ship ID
- Reservation.Cabin Class ID
- Reservation.Booking_Travel Agent_ID

Columns / Expression	Alias	Group by
Price(Cabin)		□
Name(Cruise)		□
Criuse ID(Reservation)		□
Cabin Ship ID(Reservation)		□
Cabin Class ID(Reservation)		□

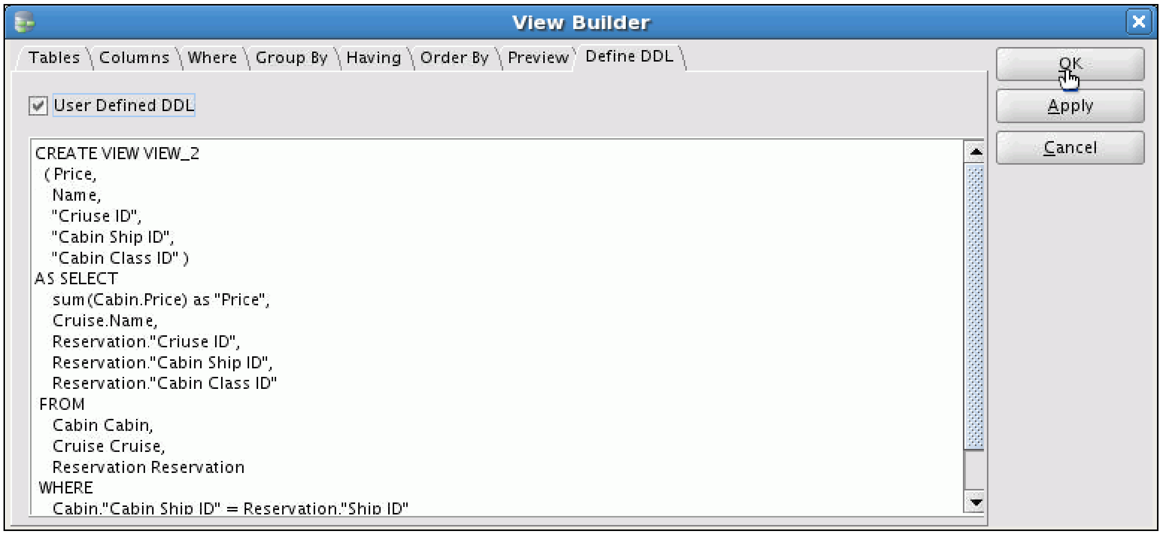
5) To create the Where clause for the join, click "Insert join clause for foreign keys," and then click the Define DDL tab.

Solution 16-1: Analyze Your Relational Model (continued)



```
Cabin."Cabin Ship ID" = Reservation."Ship ID"
AND Cabin."Cabin Class ID" = Reservation."Class ID"
AND Cruise."Criuse ID" = Reservation.ID
```

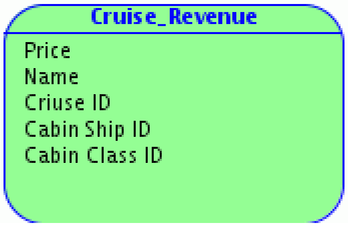
6) You want to modify the DDL to sum the total for price. Select User Defined DDL. Under AS SELECT, change the Price DDL to `sum(Cabin.Price) as "Price"`. Then click OK.



☑ User Defined DDL

```
CREATE VIEW VIEW_2
 ( Price,
   Name,
   "Criuse ID",
   "Cabin Ship ID",
   "Cabin Class ID" )
AS SELECT
   sum(Cabin.Price) as "Price",
   Cruise.Name,
   Reservation."Criuse ID",
   Reservation."Cabin Ship ID",
   Reservation."Cabin Class ID"
 FROM
   Cabin Cabin,
   Cruise Cruise,
   Reservation Reservation
 WHERE
   Cabin."Cabin Ship ID" = Reservation."Ship ID"
```

7) Click OK to create the view.



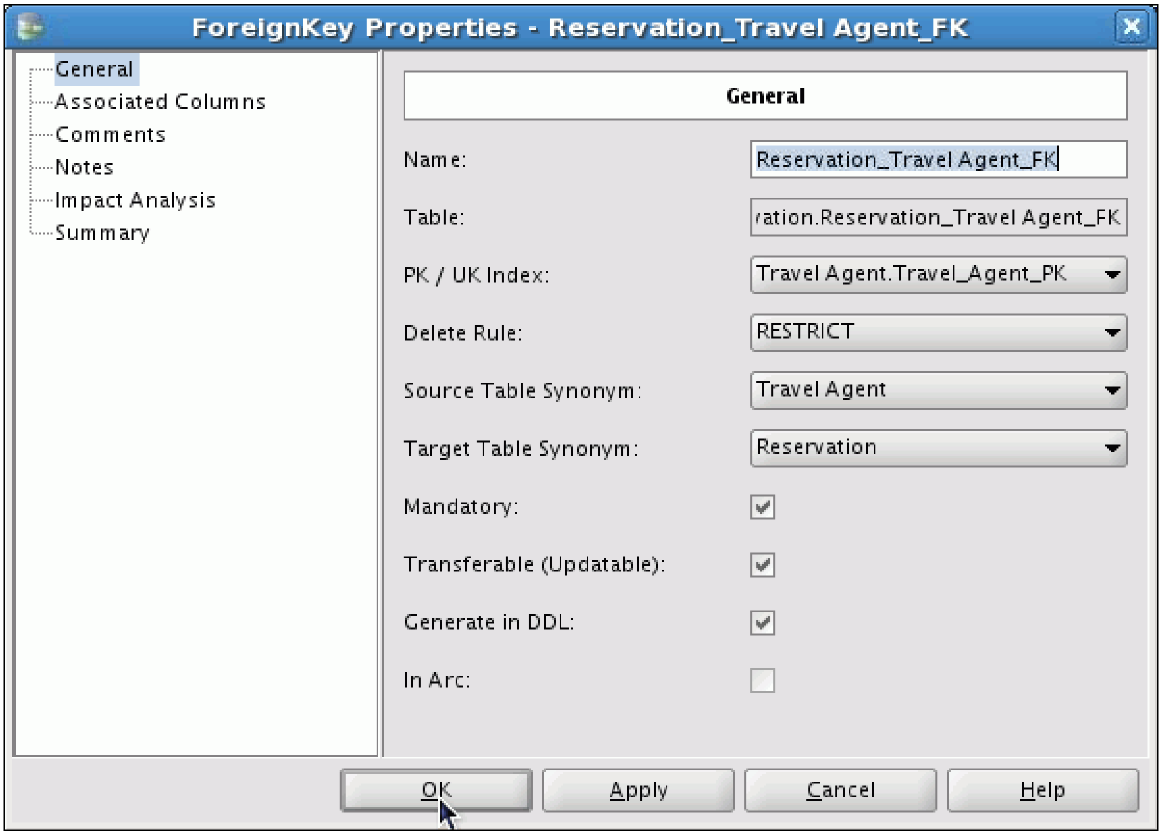
Solution 16-1: Analyze Your Relational Model (continued)

D. Want to know the total commission that each travel agent made

To accomplish this requirement, you could create an index on booking for each travel agent, and then create a view to calculate commission for each booking by reservation. Perform the following steps:

1) Create a foreign key between Travel Agent and Reservation. Select the New Foreign Key icon, and then click the Travel Agent table and then the Reservation table.

2) When the Foreign Key Properties window appears, click OK.



ForeignKey Properties - Reservation_Travel Agent_FK

- General
- Associated Columns
- Comments
- Notes
- Impact Analysis
- Summary

General

Name:	Reservation_Travel Agent_FK
Table:	ration.Reservation_Travel Agent_FK
PK / UK Index:	Travel Agent.Travel_Agent_PK
Delete Rule:	RESTRICT
Source Table Synonym:	Travel Agent
Target Table Synonym:	Reservation
Mandatory:	☑
Transferable (Updatable):	☑
Generate in DDL:	☑
In Arc:	☐

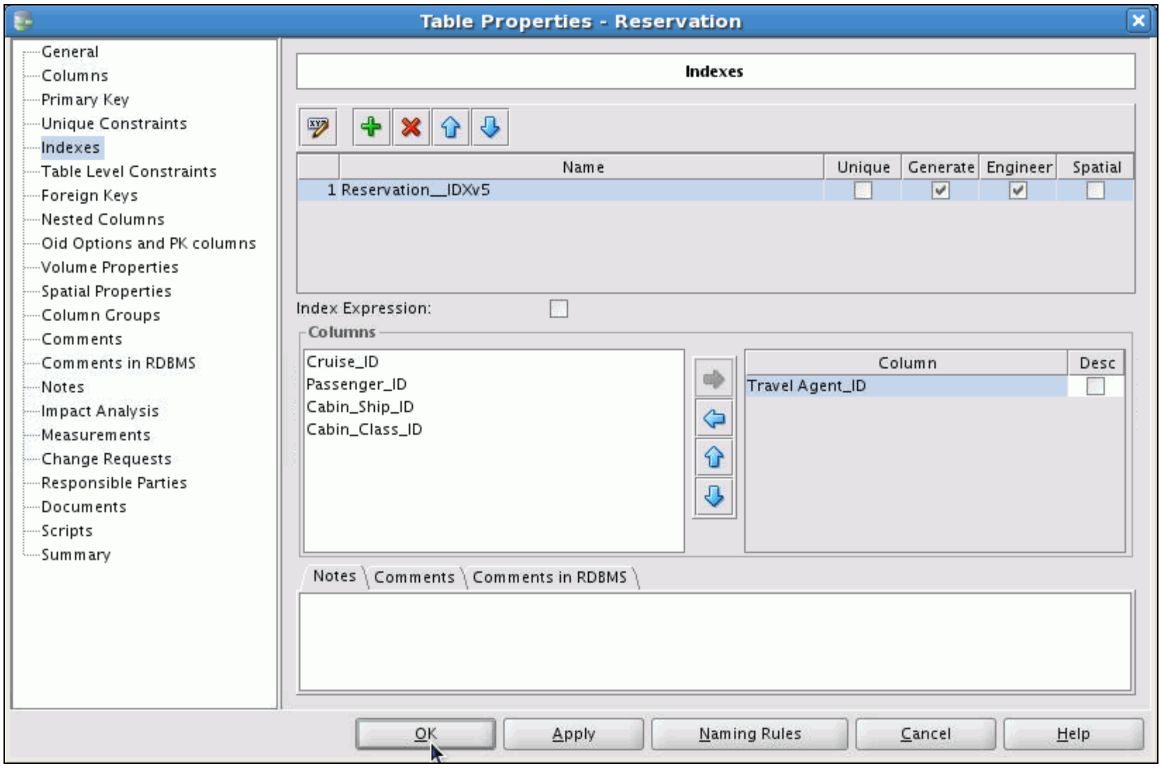
3) Double-click the Reservation table to open its properties window.

4) Select the Indexes property in the left navigator.

5) Click the Add icon.

6) Select the Travel_Agent_ID column, and click the Move icon. Then click OK.

Solution 16-1: Analyze Your Relational Model (continued)



	Name	Unique	Generate	Engineer	Spatial
1	Reservation__IDXv5	☐	☑	☑	☐

Index Expression: ☐

Columns

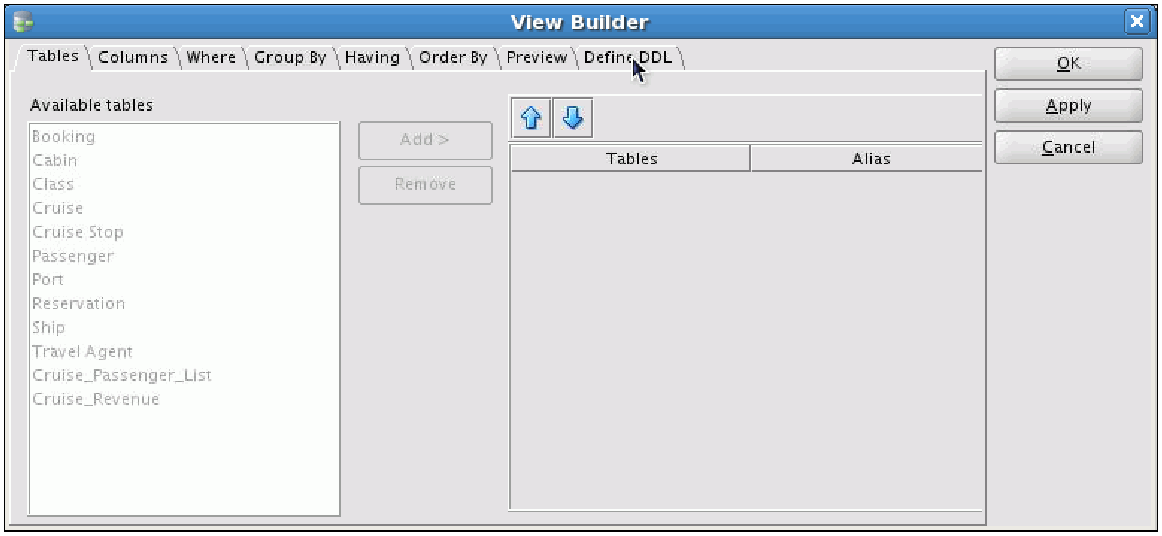
Cruise_ID
Passenger_ID
Cabin_Ship_ID
Cabin_Class_ID

Column	Desc
Travel Agent_ID	☐

7) Now you can create the view. Click the New View icon and click in the white space of the relational diagram.

8) Enter `Travel_Agent_Commission` for Name, and click the Query button (because you are going to base this view on a query).

9) If you want to just type the query, click Define DDL.



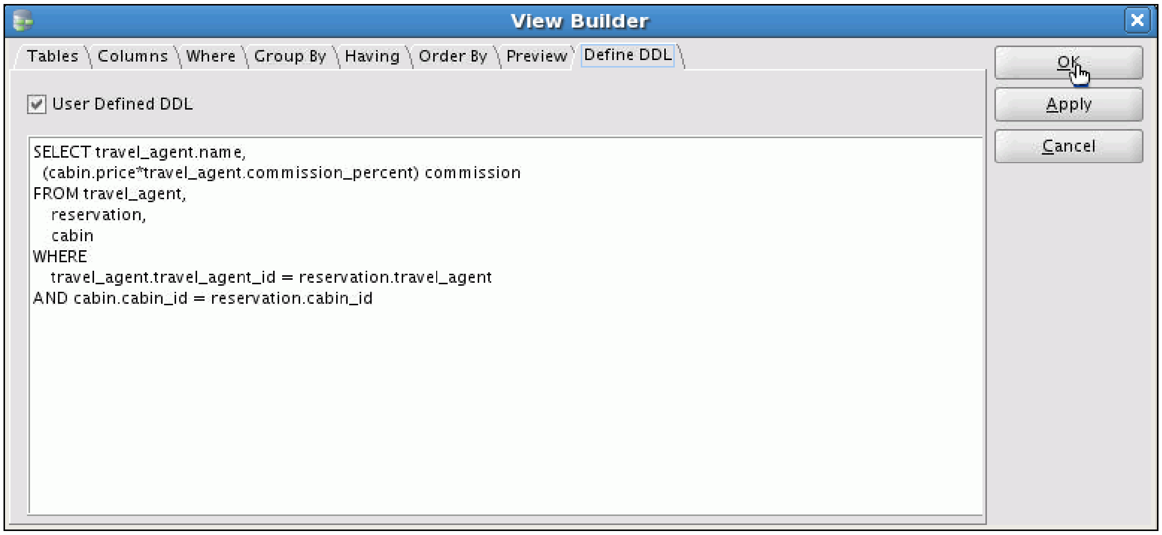
Booking
Cabin
Class
Cruise
Cruise Stop
Passenger
Port
Reservation
Ship
Travel Agent
Cruise_Passenger_List
Cruise_Revenue

10) Select the User Defined DDL check box, enter the following query, and click OK.

```
SELECT travel_agent.name,
  (cabin.price*travel_agent.commission_percent) commission
FROM travel_agent,
```

Solution 16-1: Analyze Your Relational Model (continued)

```
          reservation,
          cabin
     WHERE
          travel_agent.travel_agent_id = reservation.travel_agent
     AND cabin.cabin_id = reservation.cabin_id
```



☑ User Defined DDL

```
SELECT travel_agent.name,
 (cabin.price*travel_agent.commission_percent) commission
FROM travel_agent,
   reservation,
   cabin
WHERE
   travel_agent.travel_agent_id = reservation.travel_agent
AND cabin.cabin_id = reservation.cabin_id
```

11) Click OK to create the view. Notice that columns are not shown on the diagram because you typed in the query instead of using the wizard. To see the query, you can double-click to see the properties.

Travel_Agent_Commission

E. Want to quickly see the average age of passengers on a particular cruise

To accomplish this requirement, you could create a view between Passenger and Reservation, and calculate the average age. See whether you can perform this task on your own. You can view a possible answer in the solution `sol16-1.xml` file.

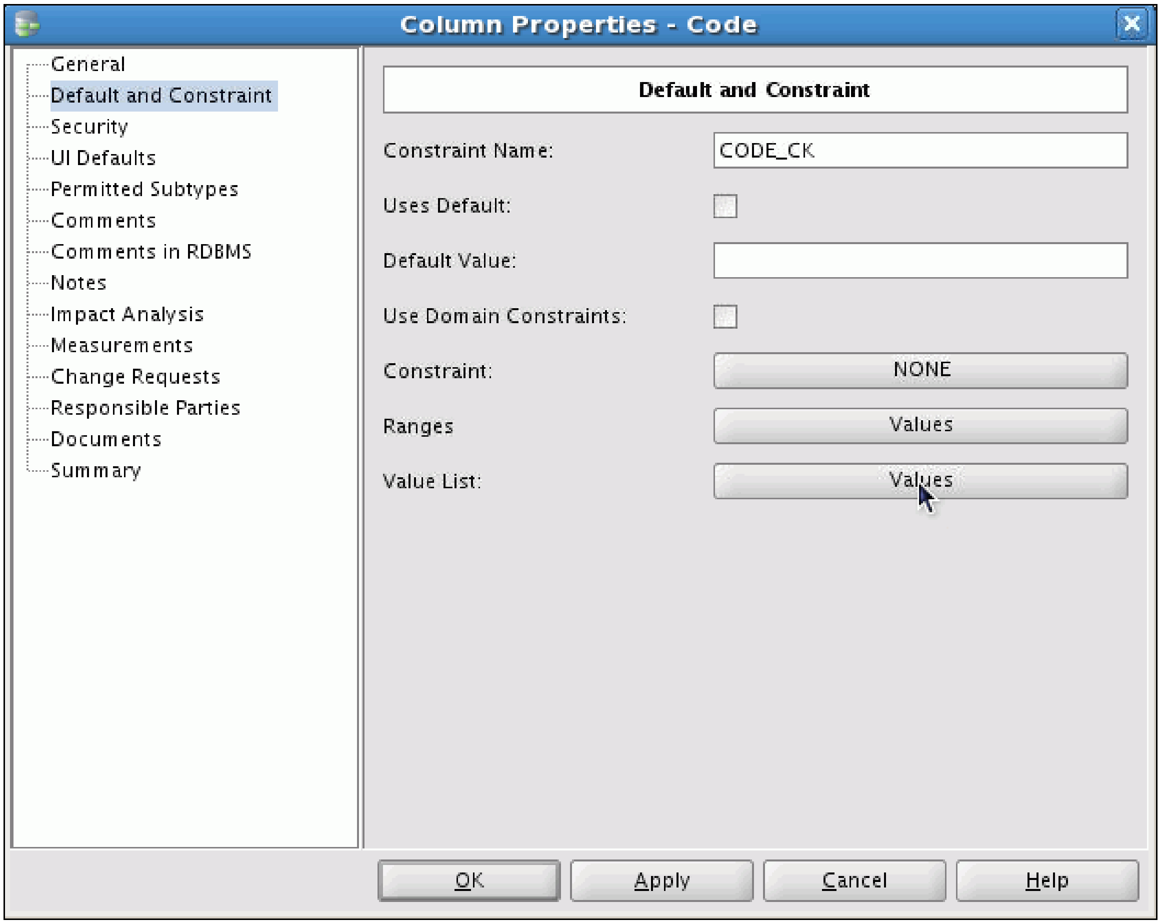
Solution 17-1: Denormalize Your Relational Model

Below is one possible solution to this practice.

A. Vehicle types are relatively static (compact, full-size, SUV, minivan, etc) and you want to show a list of vehicles by type.

To accomplish this requirement, you could add a check constraint with a list of values for vehicle type code. Perform the following steps:

1) Open the solution to Practice 15-2 (`sol15-2.xml`).
2) Click the Relational_1 tab.
3) Double-click the Vehicle Type table, and select the Column property in the left navigator.
4) Double-click the Code column to open the Column Properties window.
5) Select the "Default and Constraint" property in the left navigator.
6) Enter `CODE_CK` for the name, deselect the User Domain Constraints check box, and click the Values button for Value List.



Column Properties - Code

Default and Constraint

Constraint Name: CODE_CK

Uses Default: □

Default Value:

Use Domain Constraints: □

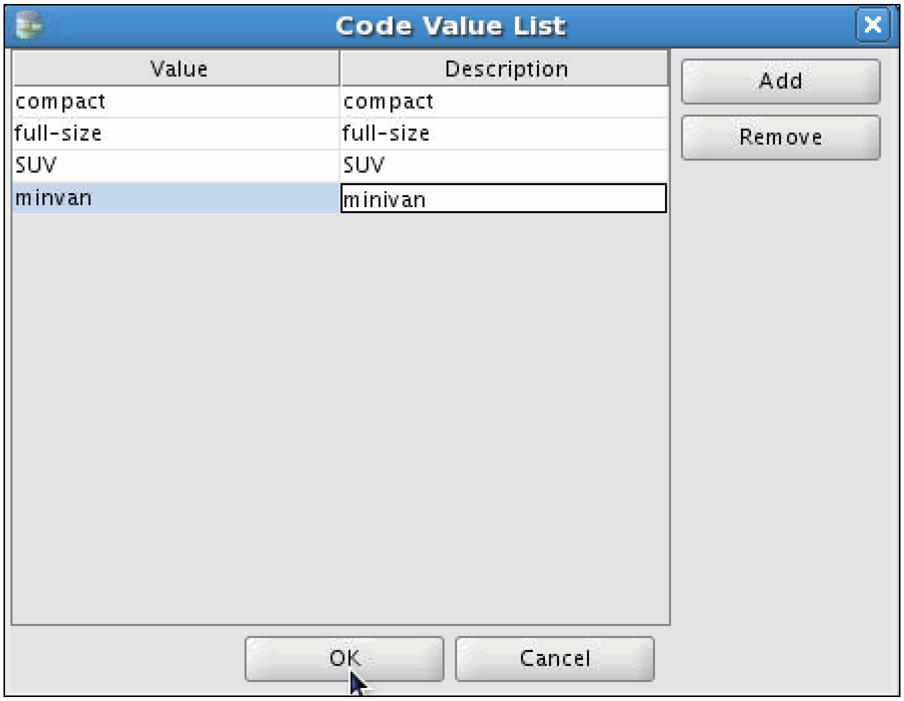
Constraint: NONE

Ranges: Values

Value List: Values

7) Enter some values by clicking the Add button and double-clicking in each field to add your values. Then click OK when done adding all the values.

Solution 17-1: Denormalize Your Relational Model (continued)



Value	Description
compact	compact
full-size	full-size
SUV	SUV
minvan	minivan

8) Click OK two more times to return to the diagram.

B. You frequently want to find out which rental agreements have been made by which offices.

To accomplish this requirement, you could add the office name to the rental agreements table so that you do not need to perform a join. You should already know how to do this in your diagram. If you want to view the solution, open the `sol17-1.xml` file.

C. You want to keep a history of which vehicles have been rented. However, you want to be able to query which vehicles have been rented between a set of dates.

This requirement has already been met by the current model because the rental agreement table already stores the date of rental and date of return.

D. You want to keep track of the total amount for a rental paid by an individual or a company.

To accomplish this requirement, you could add a total rental column to the rental agreement table; the total rental can be calculated by the date of rental and date of return and the daily rental rate. The value would be stored in the column, and the logic to determine the value would be handled by the application code.

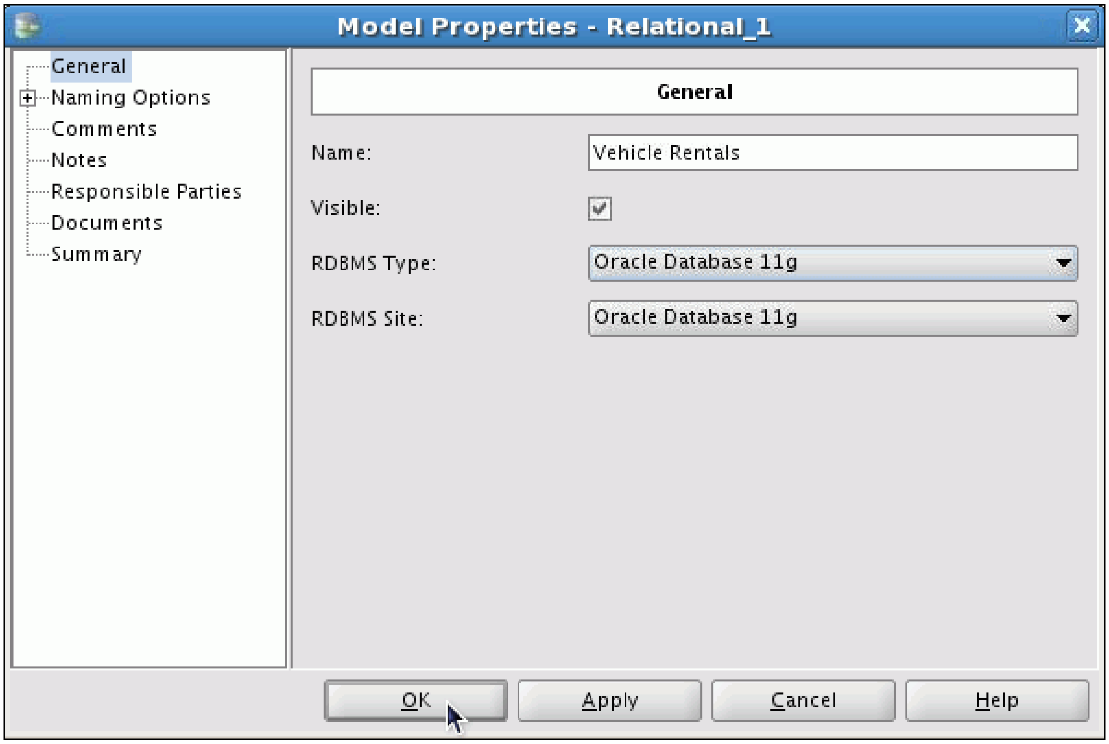
Solution 18-1: Create a Physical Model

Below is one possible solution to this practice.

1) Open the solution to the previous Practice 17-1 (`sol17-1.xml`).

2) Expand Relational Models and double-click Relational_1.

3) Change the name to `Vehicle Rentals`, make sure that RDBMS Type is set to Oracle Database 11g, and click OK.



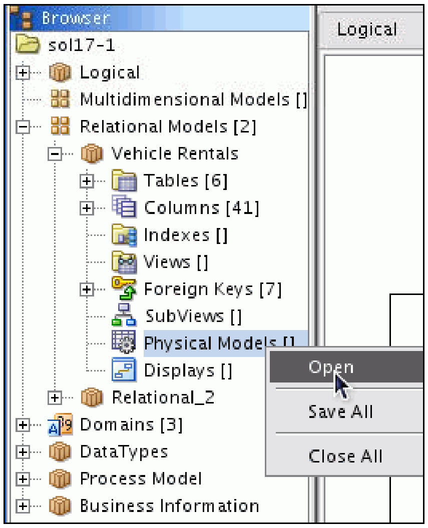
Name: Vehicle Rentals

Visible: ☑

RDBMS Type: Oracle Database 11g

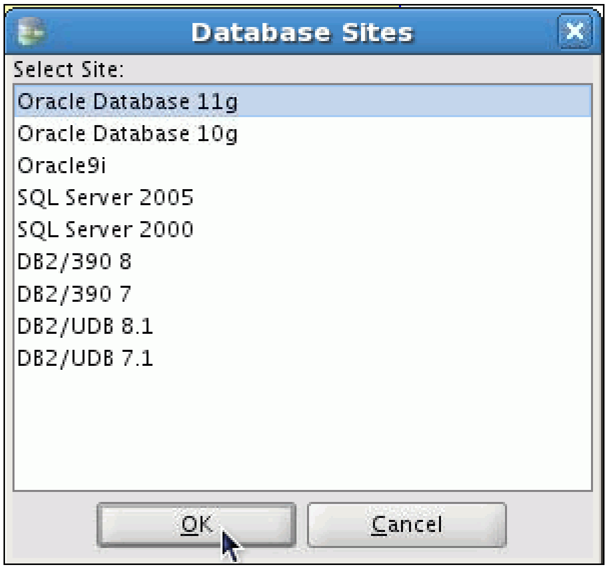
RDBMS Site: Oracle Database 11g

4) Expand Vehicle Rentals, right-click Physical Model, and select Open.



5) Select Oracle Database 11g from the list, and click OK.

Solution 18-1: Create a Physical Model (continued)



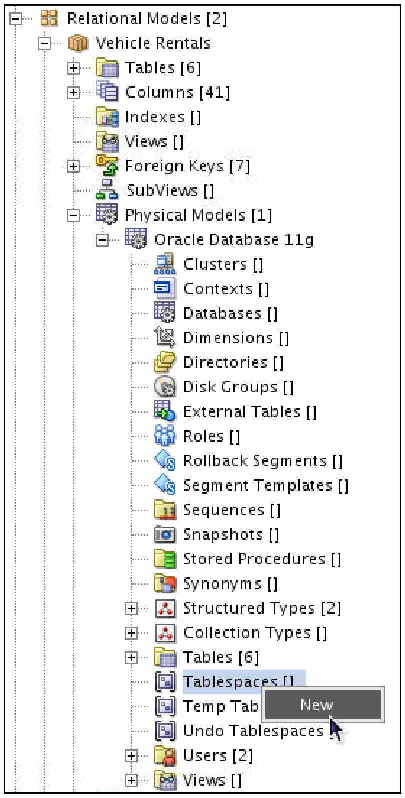
Database Sites

Select Site:

- Oracle Database 11g
- Oracle Database 10g
- Oracle9i
- SQL Server 2005
- SQL Server 2000
- DB2/390 8
- DB2/390 7
- DB2/UDB 8.1
- DB2/UDB 7.1

6) Your physical model was created successfully. Expand Physical Model > Oracle Database 11g.

7) You want to create a tablespace. Right-click Tablespace and select New.



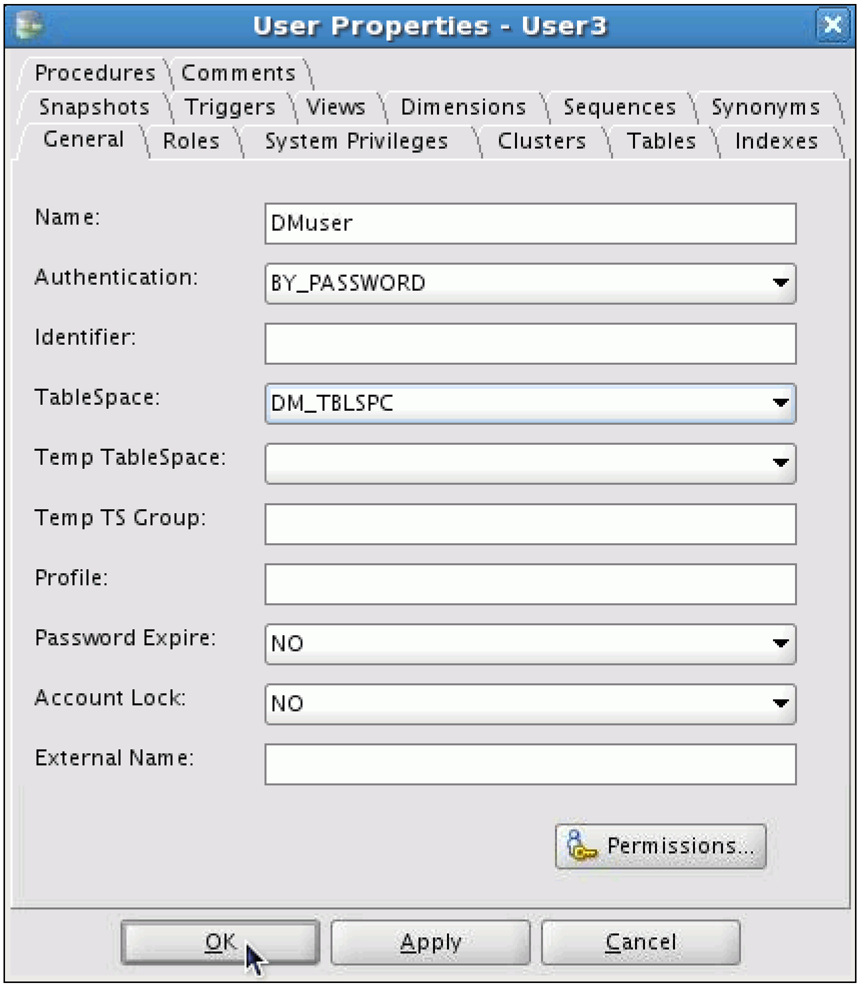
```
Relational Models [2]
  Vehicle Rentals
    Tables [6]
    Columns [41]
    Indexes []
    Views []
    Foreign Keys [7]
    SubViews []
    Physical Models [1]
      Oracle Database 11g
        Clusters []
        Contexts []
        Databases []
        Dimensions []
        Directories []
        Disk Groups []
        External Tables []
        Roles []
        Rollback Segments []
        Segment Templates []
        Sequences []
        Snapshots []
        Stored Procedures []
        Synonyms []
        Structured Types [2]
        Collection Types []
        Tables [6]
        Tablespaces []
        Temp Tab
        Undo Tablespaces
        Users [2]
        Views []
```

8) Enter `DM_TBLSPC` for the name, and click OK.

9) You want to also create a new user. Right-click Users and select New.

10) Enter `DMuser` for the name, select DM_TBLSPAC for the tablespace, and click OK.

Solution 18-1: Create a Physical Model (continued)

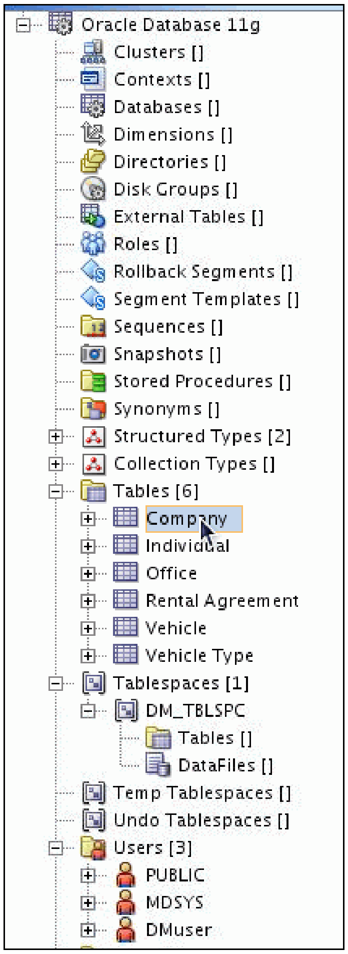


User Properties - User3

Name:	DMuser
Authentication:	BY_PASSWORD
Identifier:	
TableSpace:	DM_TBLSPC
Temp TableSpace:	
Temp TS Group:	
Profile:	
Password Expire:	NO
Account Lock:	NO
External Name:	

11) Your user was created. Now you can assign your tables to the new user. Expand Tables, and double-click the first table, Company.

Solution 18-1: Create a Physical Model (continued)

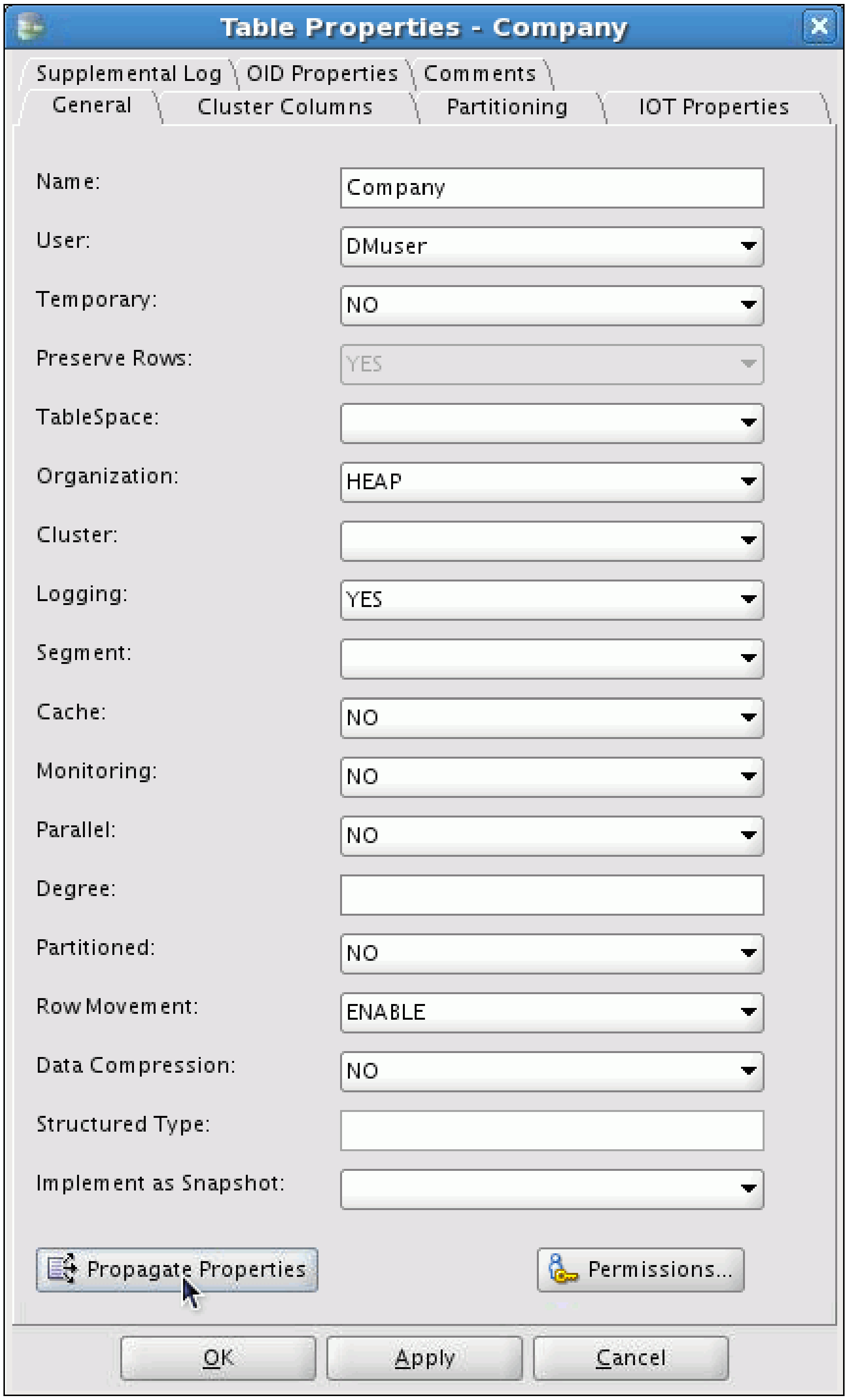


- Oracle Database 11g
  - Clusters []
  - Contexts []
  - Databases []
  - Dimensions []
  - Directories []
  - Disk Groups []
  - External Tables []
  - Roles []
  - Rollback Segments []
  - Segment Templates []
  - Sequences []
  - Snapshots []
  - Stored Procedures []
  - Synonyms []
  - Structured Types [2]
  - Collection Types []
  - Tables [6]
    - Company
    - Individual
    - Office
    - Rental Agreement
    - Vehicle
    - Vehicle Type
  - Tablespaces [1]
    - DM_TBLSPC
      - Tables []
      - DataFiles []
  - Temp Tablespaces []
  - Undo Tablespaces []
  - Users [3]
    - PUBLIC
    - MDSYS
    - DMuser

12) Select DMuser in the list of users.

13) You want to propagate this change to the other tables. Click Propagate Properties.

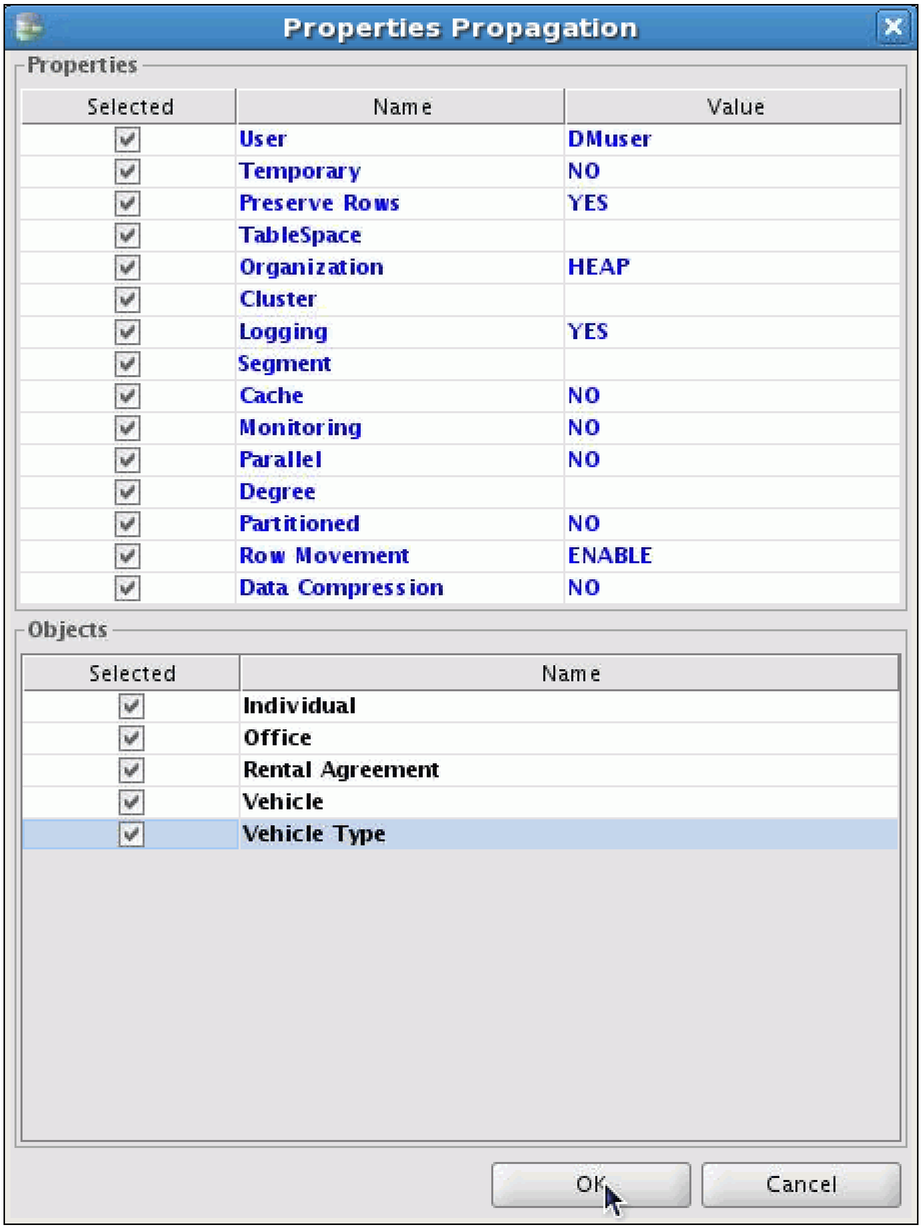
Solution 18-1: Create a Physical Model (continued)



Name:	Company
User:	DMuser
Temporary:	NO
Preserve Rows:	YES
TableSpace:	
Organization:	HEAP
Cluster:	
Logging:	YES
Segment:	
Cache:	NO
Monitoring:	NO
Parallel:	NO
Degree:	
Partitioned:	NO
Row Movement:	ENABLE
Data Compression:	NO
Structured Type:	
Implement as Snapshot:	

14) Under Objects, select the check box in front of each of the other tables, and click OK.

Solution 18-1: Create a Physical Model (continued)



Properties Propagation

Properties

Selected	Name	Value
☑	User	DMuser
☑	Temporary	NO
☑	Preserve Rows	YES
☑	TableSpace	
☑	Organization	HEAP
☑	Cluster	
☑	Logging	YES
☑	Segment	
☑	Cache	NO
☑	Monitoring	NO
☑	Parallel	NO
☑	Degree	
☑	Partitioned	NO
☑	Row Movement	ENABLE
☑	Data Compression	NO

Objects

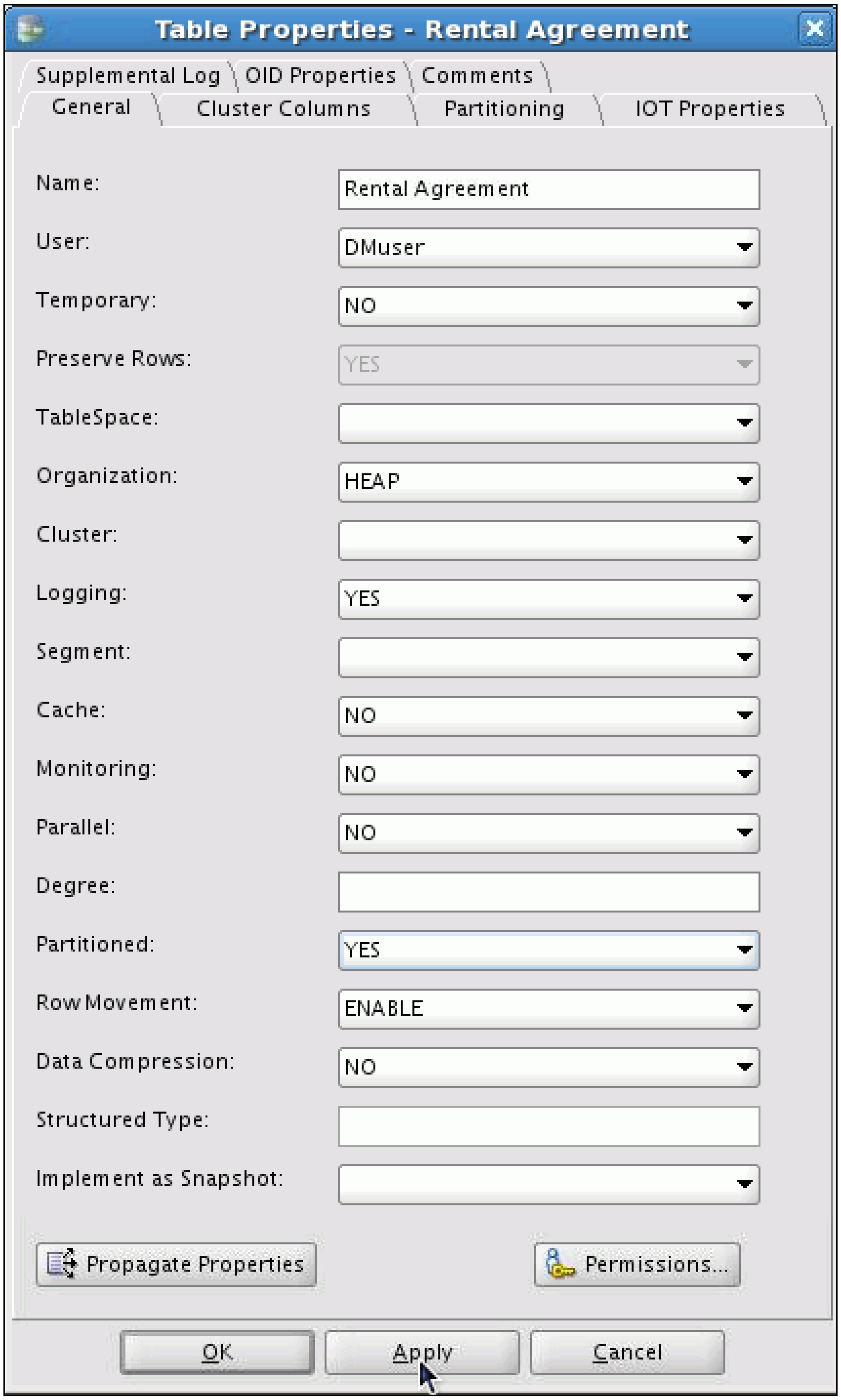
Selected	Name
☑	Individual
☑	Office
☑	Rental Agreement
☑	Vehicle
☑	Vehicle Type

15) Click OK.

16) Notice that the username qualifies each of the tables in the list. Double-click the DMuser.Rental Agreements table.

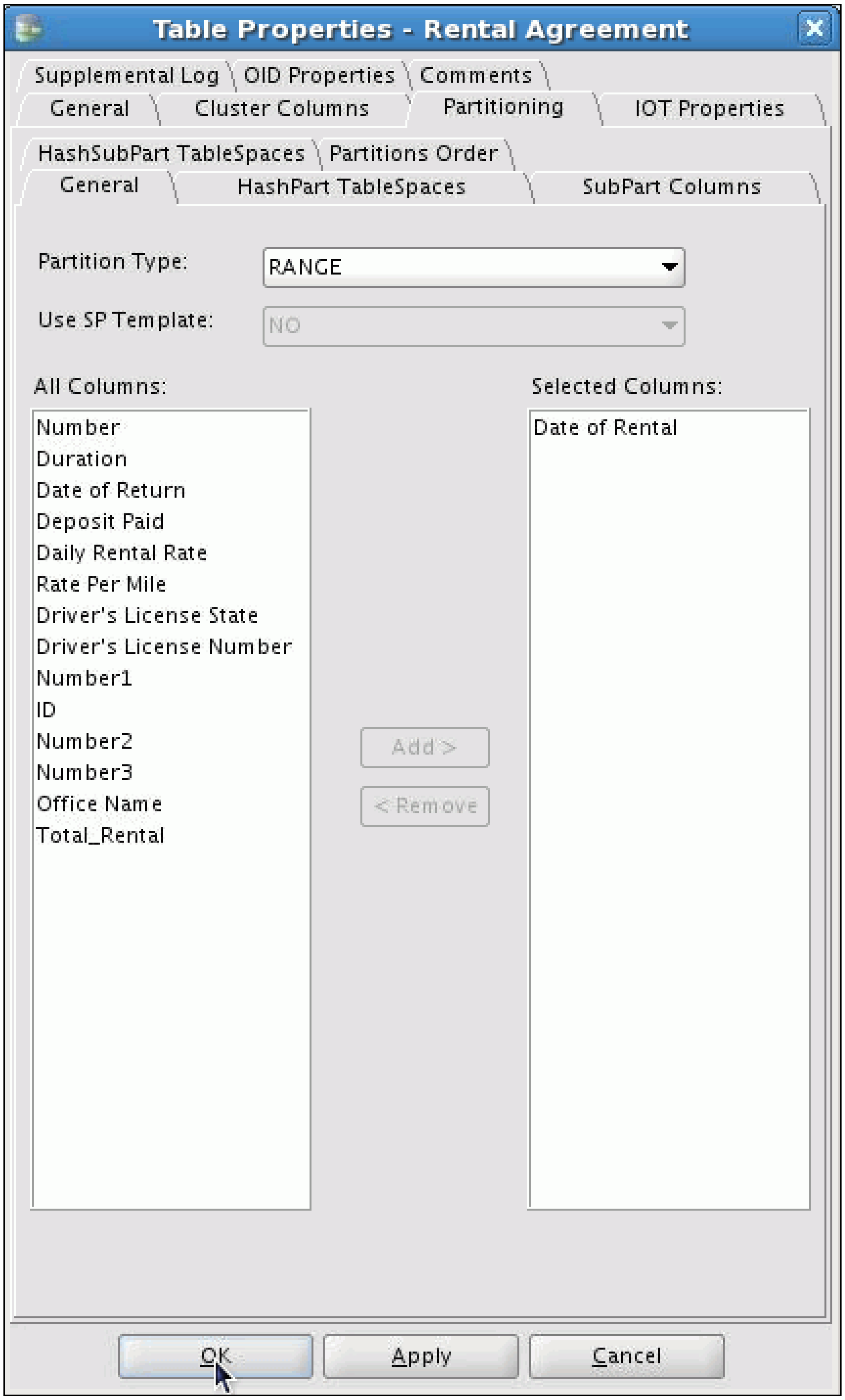
17) You want to create a partition for the Rental Agreement table so that you can access the rental agreements for a given month very quickly. You must first designate this table a partitioned table. Select Yes for Partitioned, and click Apply.

Solution 18-1: Create a Physical Model (continued)



Name:	Rental Agreement
User:	DMuser
Temporary:	NO
Preserve Rows:	YES
TableSpace:	
Organization:	HEAP
Cluster:	
Logging:	YES
Segment:	
Cache:	NO
Monitoring:	NO
Parallel:	NO
Degree:	
Partitioned:	YES
Row Movement:	ENABLE
Data Compression:	NO
Structured Type:	
Implement as Snapshot:	

18) Click the Partitioning tab. You can define what type of partition and the column(s) it will be based on. You will create a Range partition based on "Date of Rental." Select "Date of Rental," click Add to move it to the Selected Columns area, and click OK.



Partition Type: RANGE

Use SP Template: NO

All Columns:

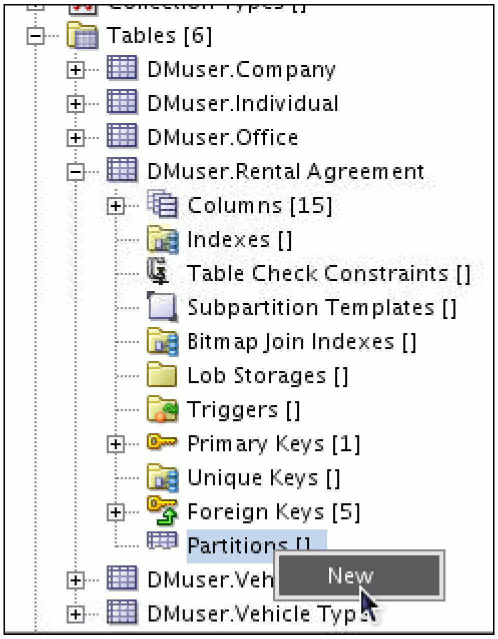
Number
Duration
Date of Return
Deposit Paid
Daily Rental Rate
Rate Per Mile
Driver's License State
Driver's License Number
Number1
ID
Number2
Number3
Office Name
Total_Rental

Selected Columns:

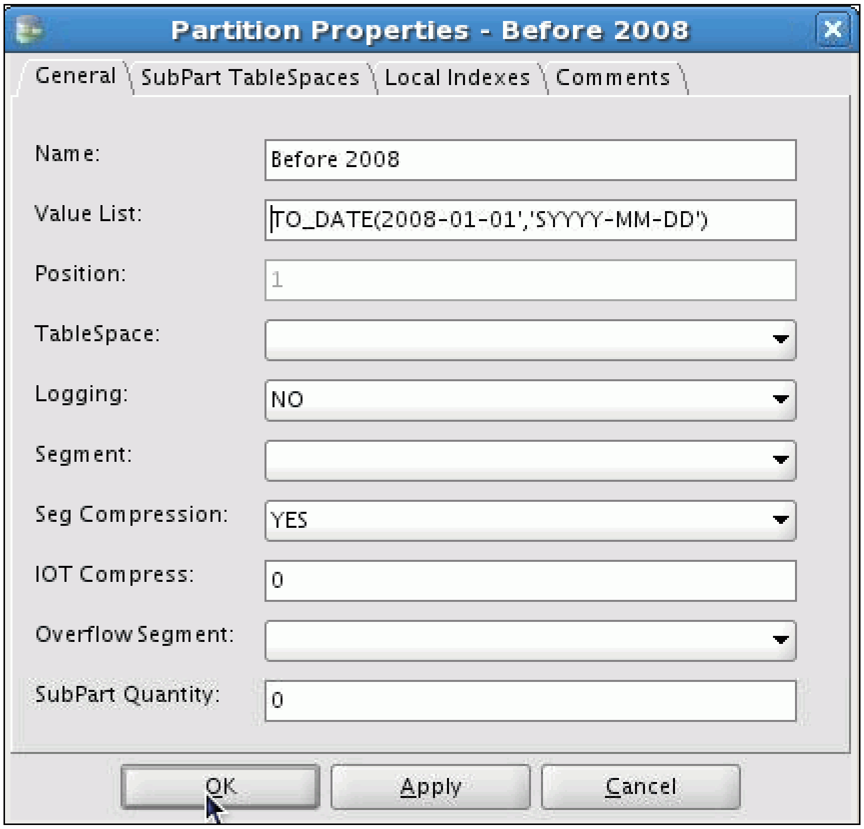
Date of Rental

19) Now you can create the partition itself. Expand the DMuser.Rental Agreement table, right-click Partition, and select New.

Solution 18-1: Create a Physical Model (continued)



20) You can create a partition for rentals before 2008. Enter `Before 2008` in the Name field, enter the expression `TO_DATE(2008-01-01','SYYYY-MM-DD')` in the Value List field, and click OK.

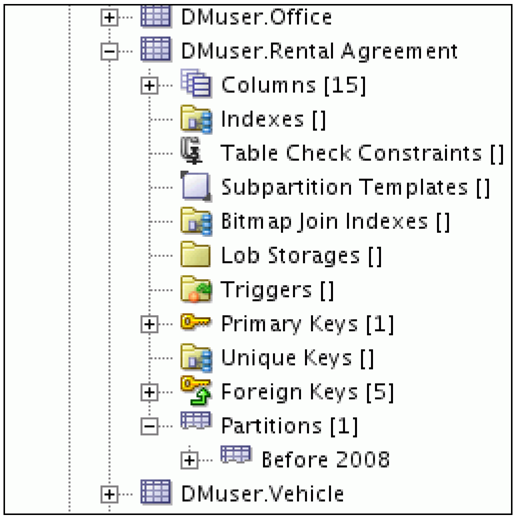


Partition Properties - Before 2008

Name:	Before 2008
Value List:	TO_DATE(2008-01-01','SYYYY-MM-DD')
Position:	1
TableSpace:	
Logging:	NO
Segment:	
Seg Compression:	YES
IOT Compress:	0
Overflow Segment:	
SubPart Quantity:	0

21) Your partition was created successfully.

Solution 18-1: Create a Physical Model (continued)



- DMuser.Office
- DMuser.Rental Agreement
  - Columns [15]
  - Indexes []
  - Table Check Constraints []
  - Subpartition Templates []
  - Bitmap Join Indexes []
  - Lob Storages []
  - Triggers []
  - Primary Keys [1]
  - Unique Keys []
  - Foreign Keys [5]
  - Partitions [1]
    - Before 2008
- DMuser.Vehicle

Solution 19-1: Generate DDL

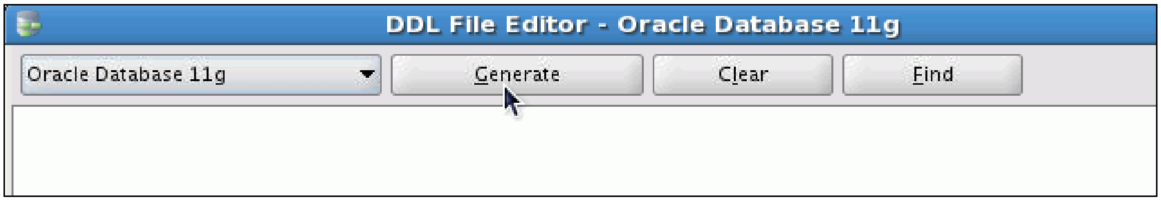
Below is one possible solution to this practice.

1) Open the solution to the previous Practice 18-1 (`sol18-1.xml`).

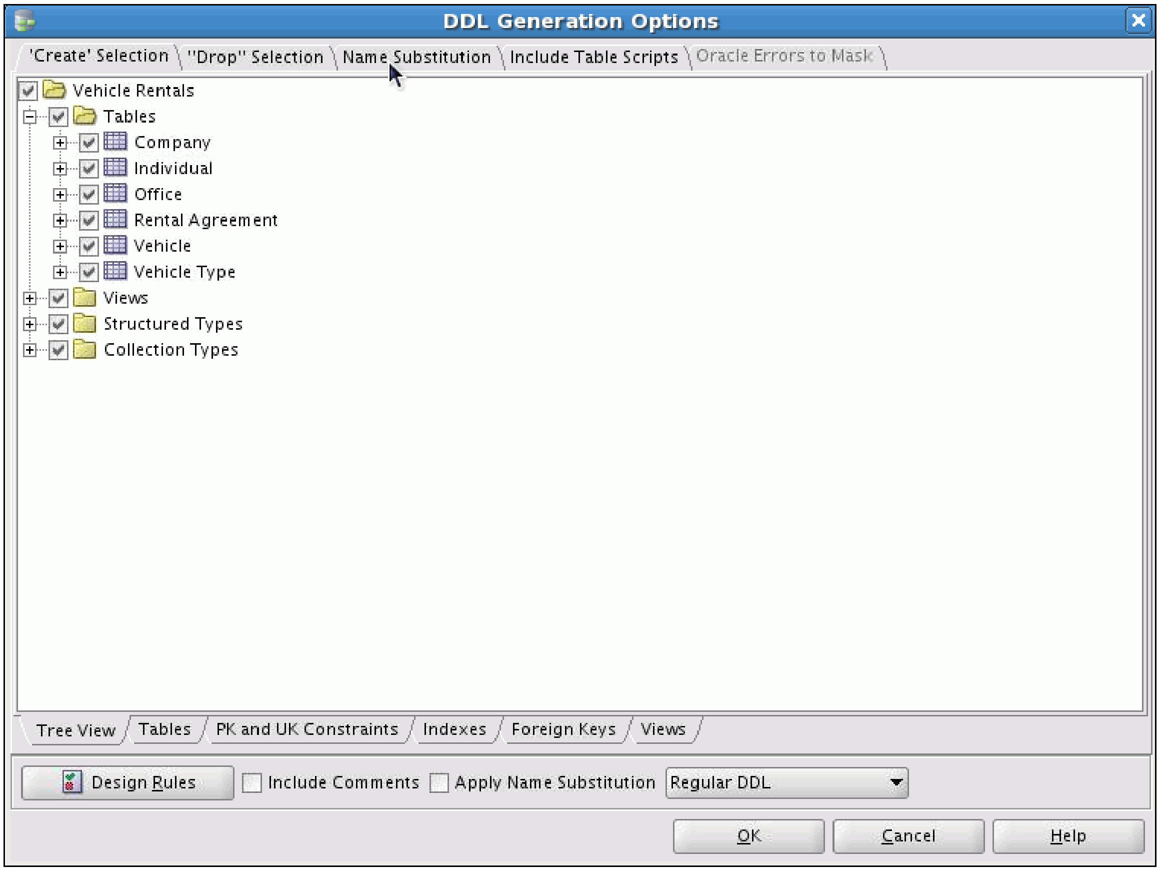
2) Make sure that the relational model's Vehicle Rentals tab is selected.

3) Click the DDL Generation icon.

4) Make sure that Oracle Database 11g is selected, and click Generate.



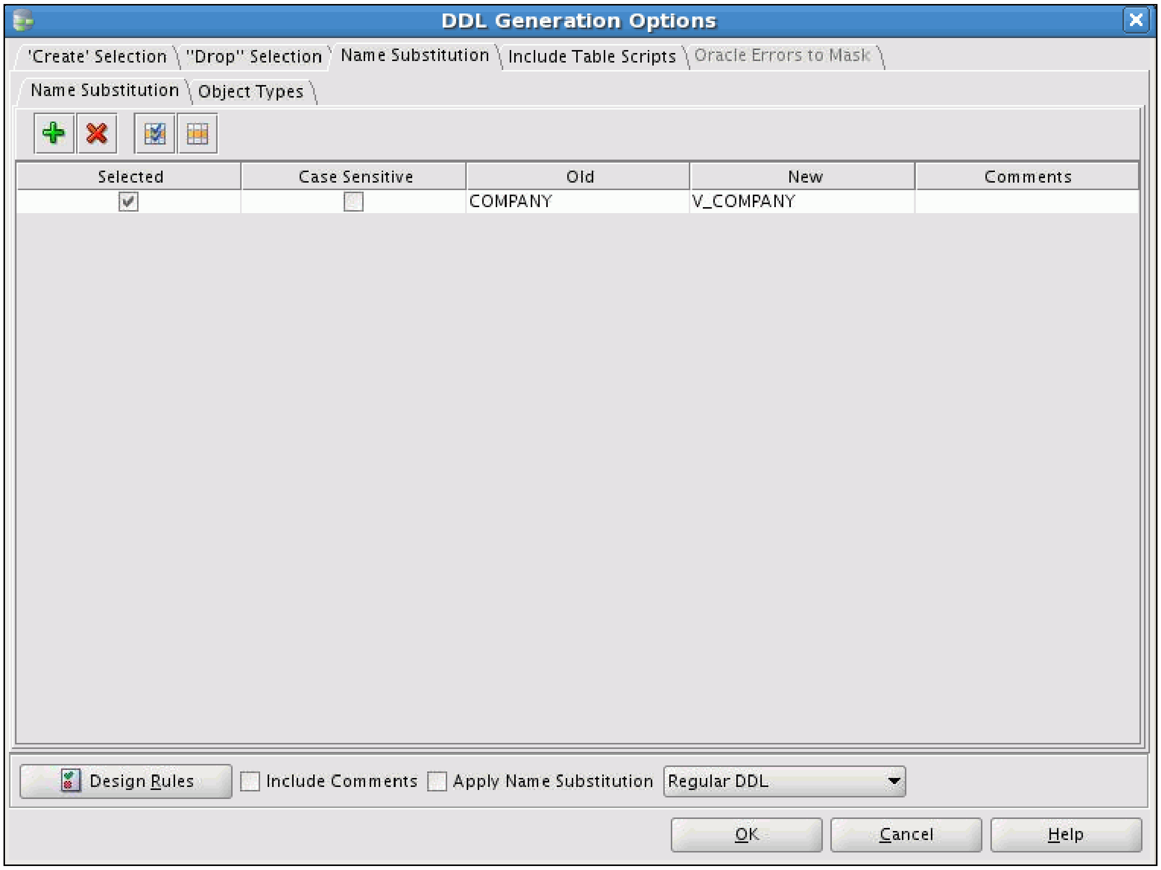
5) Expand Tables to see the list of tables that will be generated. Click the Name Substitution tab.



6) You want the Company and Individual tables to be generated with a 'V_' prefix. Click the Add icon.

7) Click the Selected check box and enter `COMPANY` for Old and `V_COMPANY` for New, and click the Add icon.

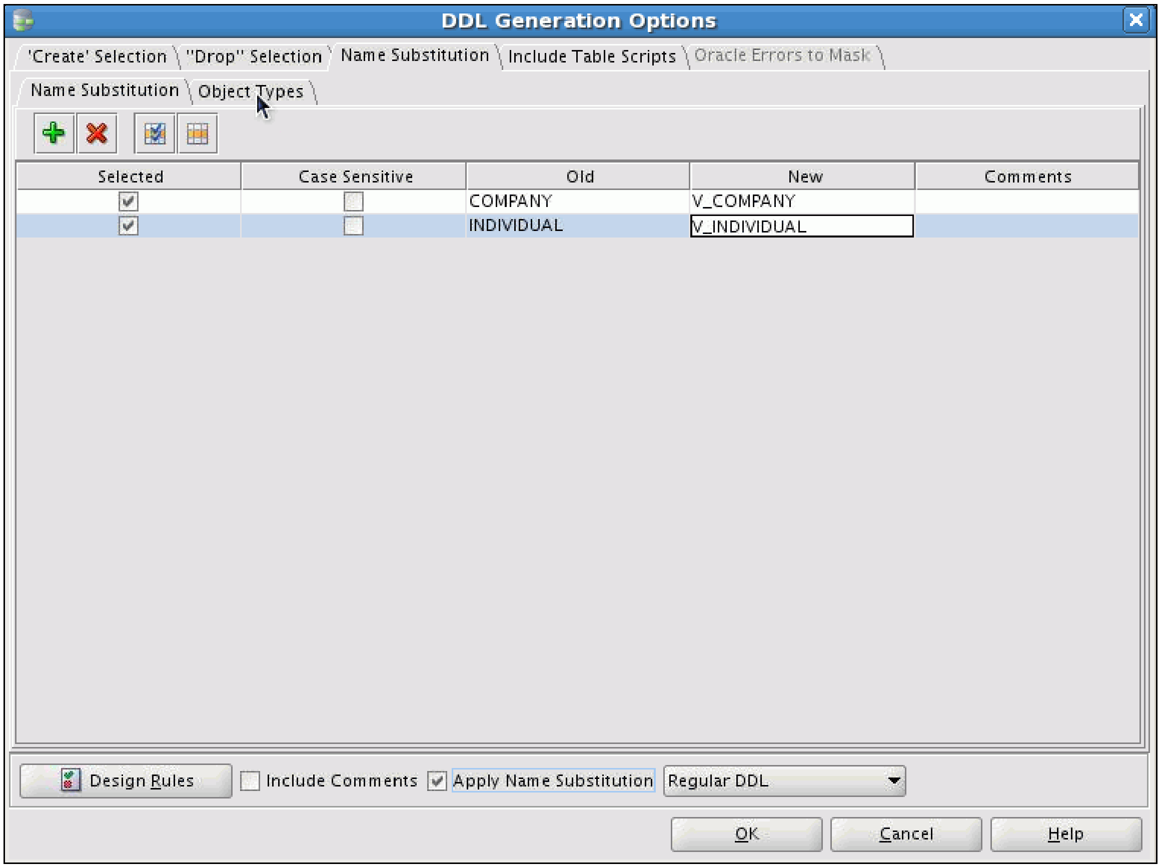
Solution 19-1: Generate DDL (continued)



Selected	Case Sensitive	Old	New	Comments
☑	□	COMPANY	V_COMPANY	

8) Click the Select check box, and enter `INDIVIDUAL` for Old and `V_INDIVIDUAL` for New. Click Apply Name Substitution, and then click the Object Types tab.

Solution 19-1: Generate DDL (continued)

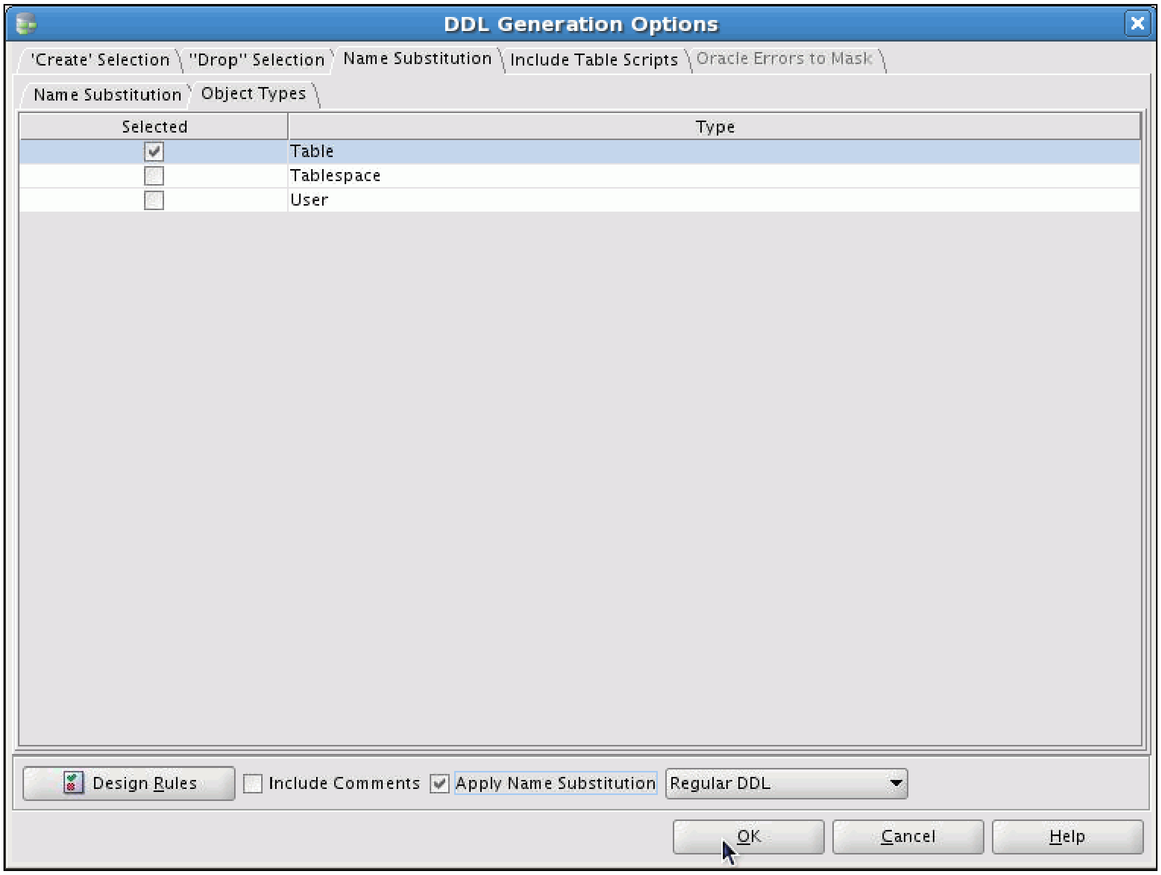


Selected	Case Sensitive	Old	New	Comments
☑	□	COMPANY	V_COMPANY	
☑	□	INDIVIDUAL	V_INDIVIDUAL	

□ Include Comments ☑ Apply Name Substitution Regular DDL

9) Make sure that the Table check box is selected, and click OK.

Solution 19-1: Generate DDL (continued)

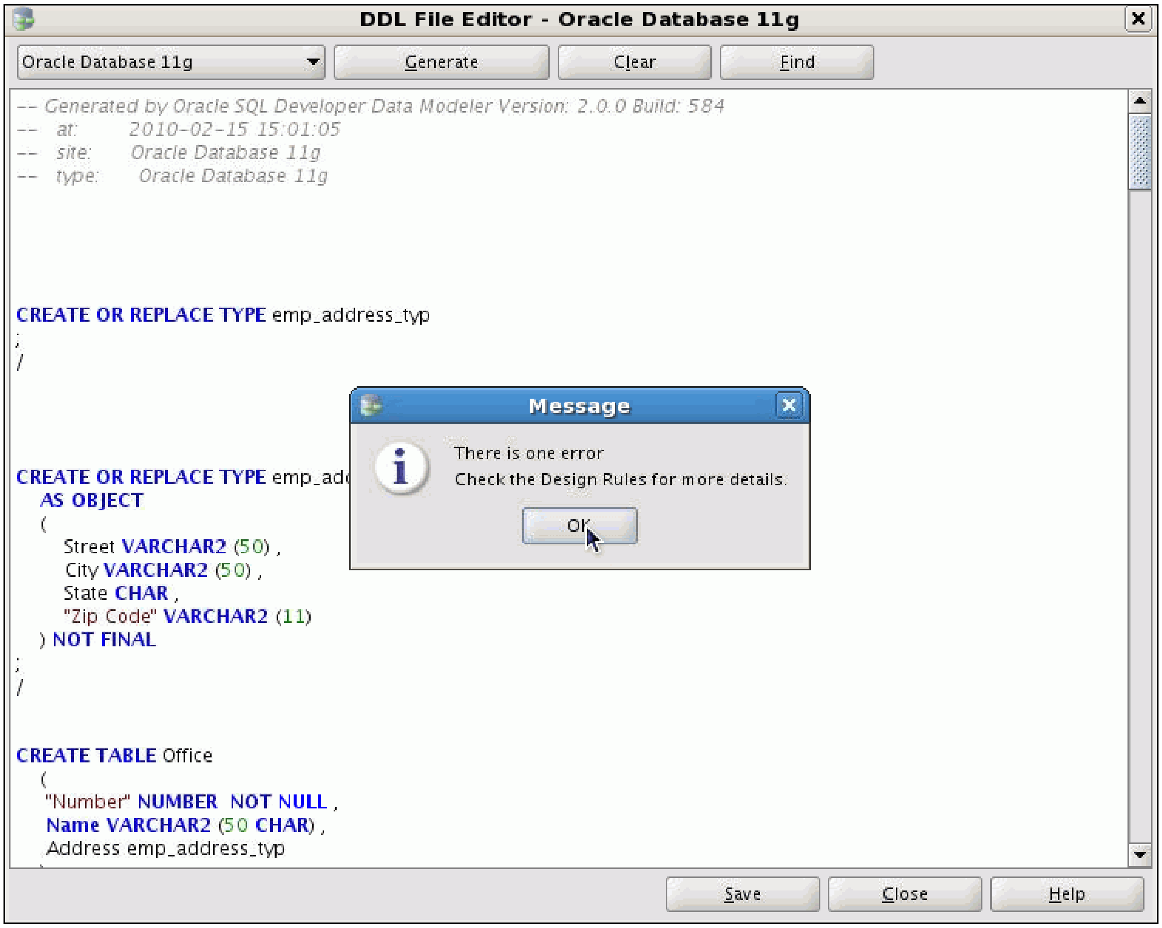


Selected	Type
☑	Table
□	Tablespace
□	User

□ Include Comments ☑ Apply Name Substitution Regular DDL

10) The DDL was generated. You received one error. Click OK.

Solution 19-1: Generate DDL (continued)



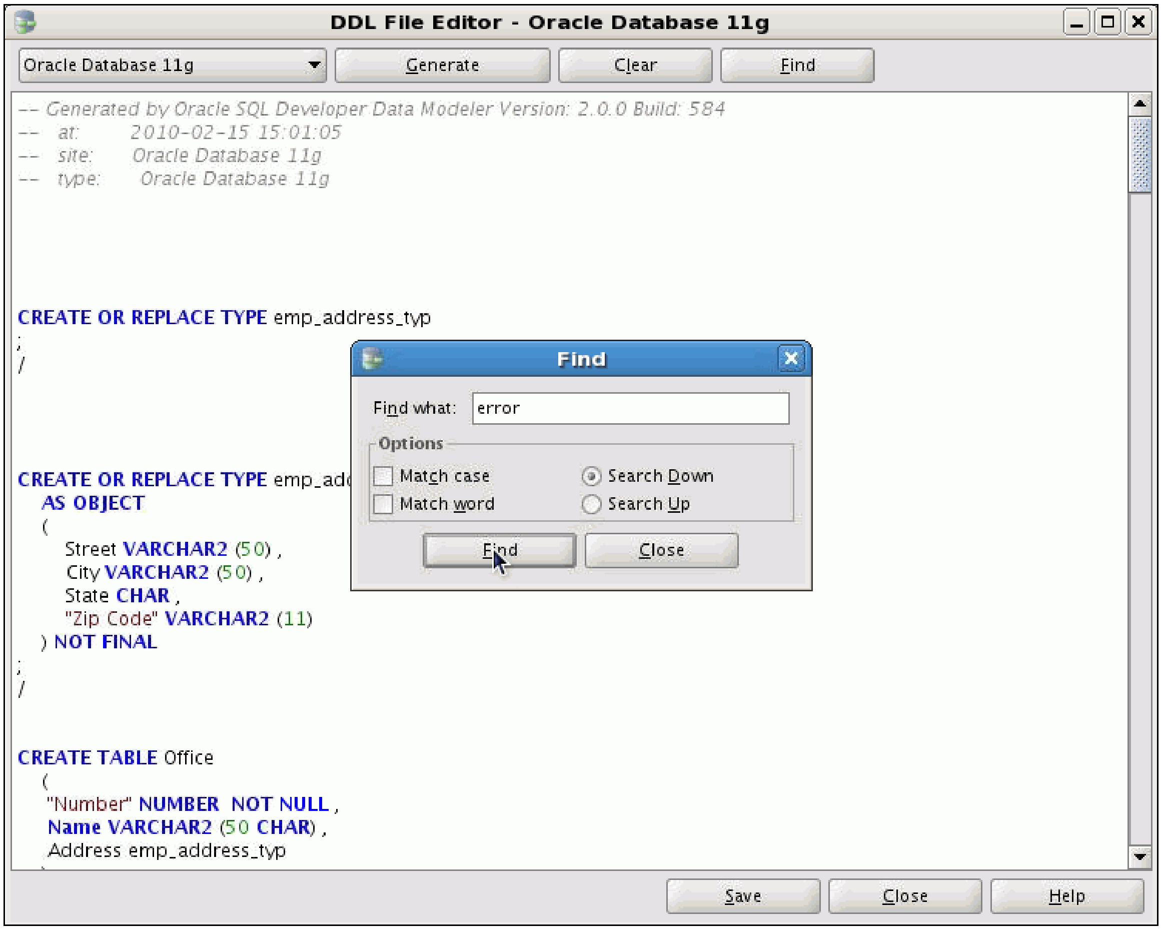
```
-- Generated by Oracle SQL Developer Data Modeler Version: 2.0.0 Build: 584
--   at:        2010-02-15 15:01:05
--   site:      Oracle Database 11g
--   type:      Oracle Database 11g



CREATE OR REPLACE TYPE emp_address_typ
;
/



CREATE OR REPLACE TYPE emp_add
   AS OBJECT
   (
      Street VARCHAR2 (50) ,
      City VARCHAR2 (50) ,
      State CHAR ,
      "Zip Code" VARCHAR2 (11)
   ) NOT FINAL
;
/


CREATE TABLE Office
   (
    "Number" NUMBER  NOT NULL ,
    Name VARCHAR2 (50 CHAR) ,
    Address emp_address_typ
```

There is one error
Check the Design Rules for more details.

11) You can search for the error. Click the Find button.

12) Enter `error` in the “Find what” field, and click Find.

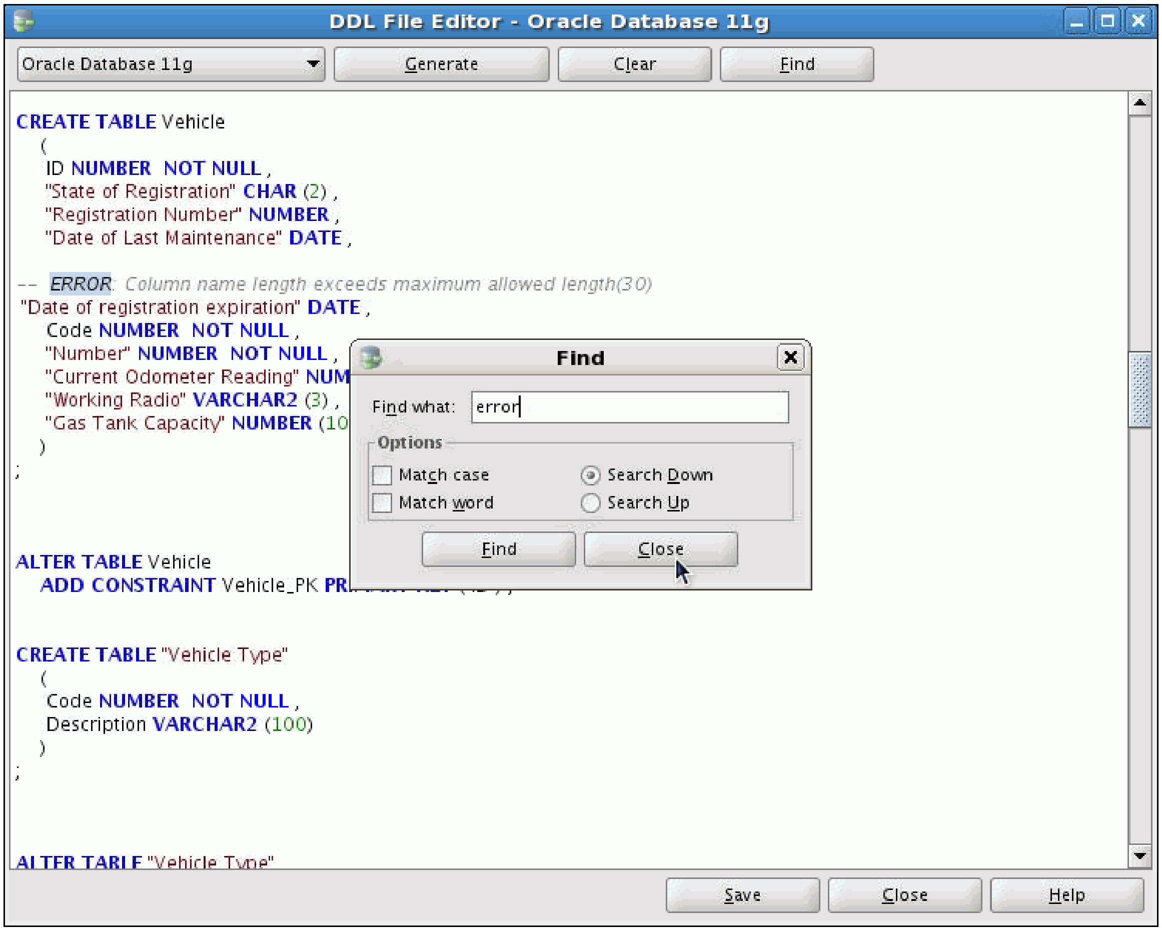
Solution 19-1: Generate DDL (continued)



```
-- Generated by Oracle SQL Developer Data Modeler Version: 2.0.0 Build: 584
--   at:        2010-02-15 15:01:05
--   site:      Oracle Database 11g
--   type:      Oracle Database 11g



CREATE OR REPLACE TYPE emp_address_typ
;
/



CREATE OR REPLACE TYPE emp_add
   AS OBJECT
   (
     Street VARCHAR2 (50) ,
     City VARCHAR2 (50) ,
     State CHAR ,
     "Zip Code" VARCHAR2 (11)
   ) NOT FINAL
;
/


CREATE TABLE Office
   (
    "Number" NUMBER  NOT NULL ,
    Name VARCHAR2 (50 CHAR) ,
    Address emp_address_typ
```

13) The error is displayed. It is due to a column length being too long. Click Close in the Find dialog box.

Solution 19-1: Generate DDL (continued)



```
CREATE TABLE Vehicle
    (
     ID NUMBER  NOT NULL ,
     "State of Registration" CHAR (2) ,
     "Registration Number" NUMBER ,
     "Date of Last Maintenance" DATE ,

-- ERROR: Column name length exceeds maximum allowed length(30)
 "Date of registration expiration" DATE ,
     Code NUMBER  NOT NULL ,
     "Number" NUMBER  NOT NULL ,
     "Current Odometer Reading" NUM
     "Working Radio" VARCHAR2 (3) ,
     "Gas Tank Capacity" NUMBER (10
    )
;


ALTER TABLE Vehicle
    ADD CONSTRAINT Vehicle_PK PR


CREATE TABLE "Vehicle Type"
    (
     Code NUMBER  NOT NULL ,
     Description VARCHAR2 (100)
    )
;


ALTER TABLE "Vehicle Type"
```

Find what: error

Options

□ Match case ● Search Down

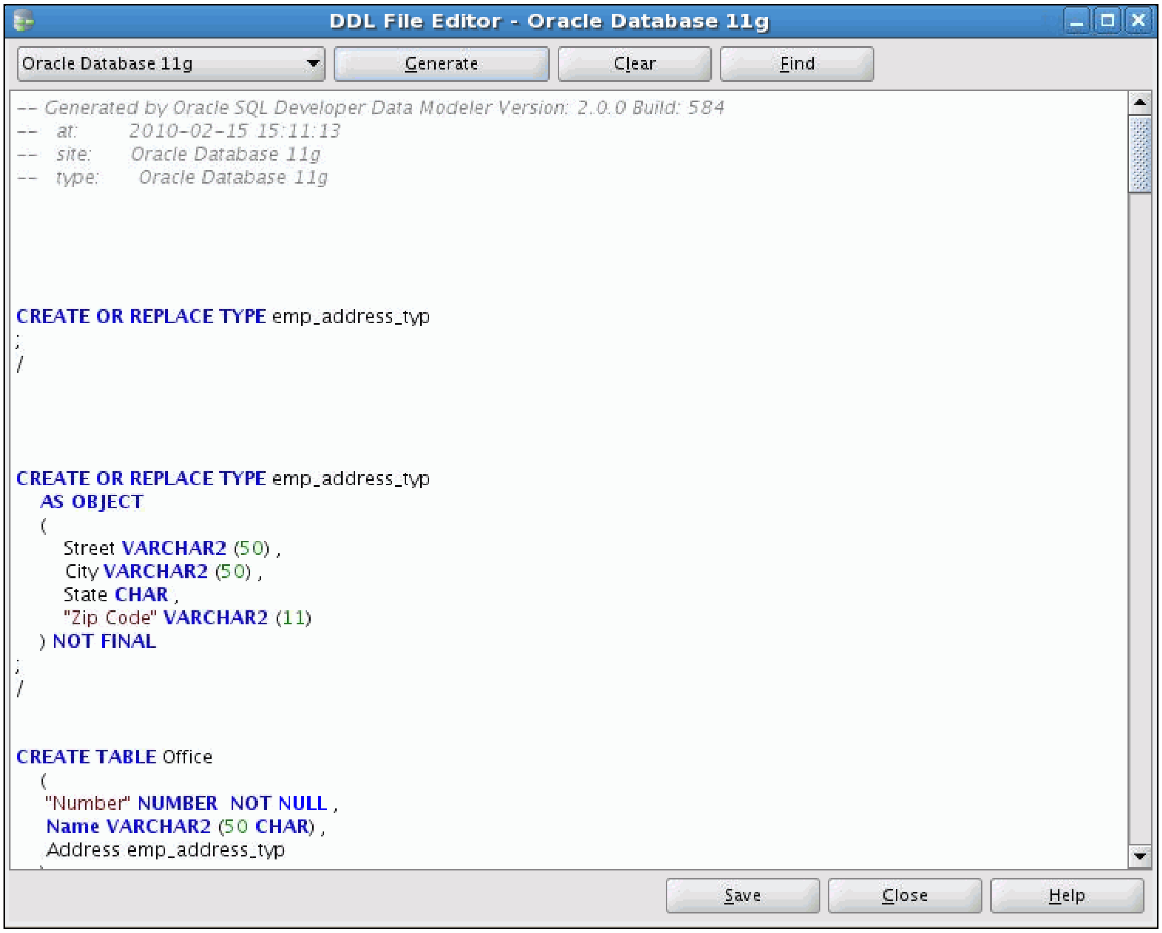
□ Match word ○ Search Up

14) Click Close.

15) Go back to your relational model and change the name of the column for “Date of registration expiration” to `Reg Exp Date`.

16) Regenerate the DDL until there are no errors.

Solution 19-1: Generate DDL (continued)



```
-- Generated by Oracle SQL Developer Data Modeler Version: 2.0.0 Build: 584
--   at:        2010-02-15 15:11:13
--   site:      Oracle Database 11g
--   type:      Oracle Database 11g




CREATE OR REPLACE TYPE emp_address_typ
;
/




CREATE OR REPLACE TYPE emp_address_typ
  AS OBJECT
  (
    Street VARCHAR2 (50) ,
    City VARCHAR2 (50) ,
    State CHAR ,
    "Zip Code" VARCHAR2 (11)
  ) NOT FINAL
;
/


CREATE TABLE Office
  (
   "Number" NUMBER  NOT NULL ,
   Name VARCHAR2 (50 CHAR) ,
   Address emp_address_typ
```

17) Review the DDL to make sure that all the tasks you completed previously are included (for example, the CODE_CK constraint).

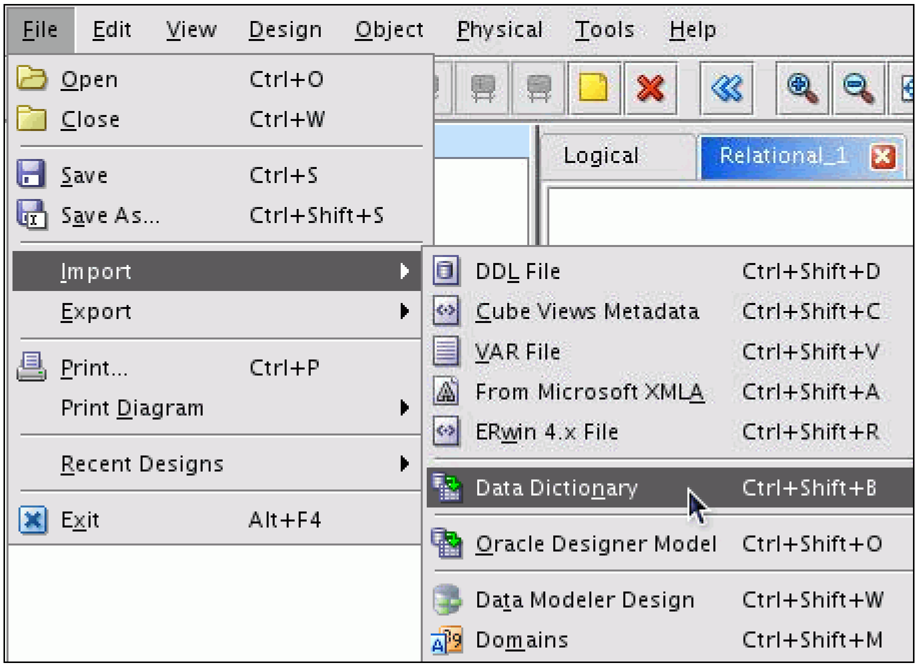
18) Verify that the Company and Individual tables have the ‘V_’ prefix.

Solution 20-1: Re-Engineer the HR Schema

Below is one possible solution to this practice.

1) Close all open models.

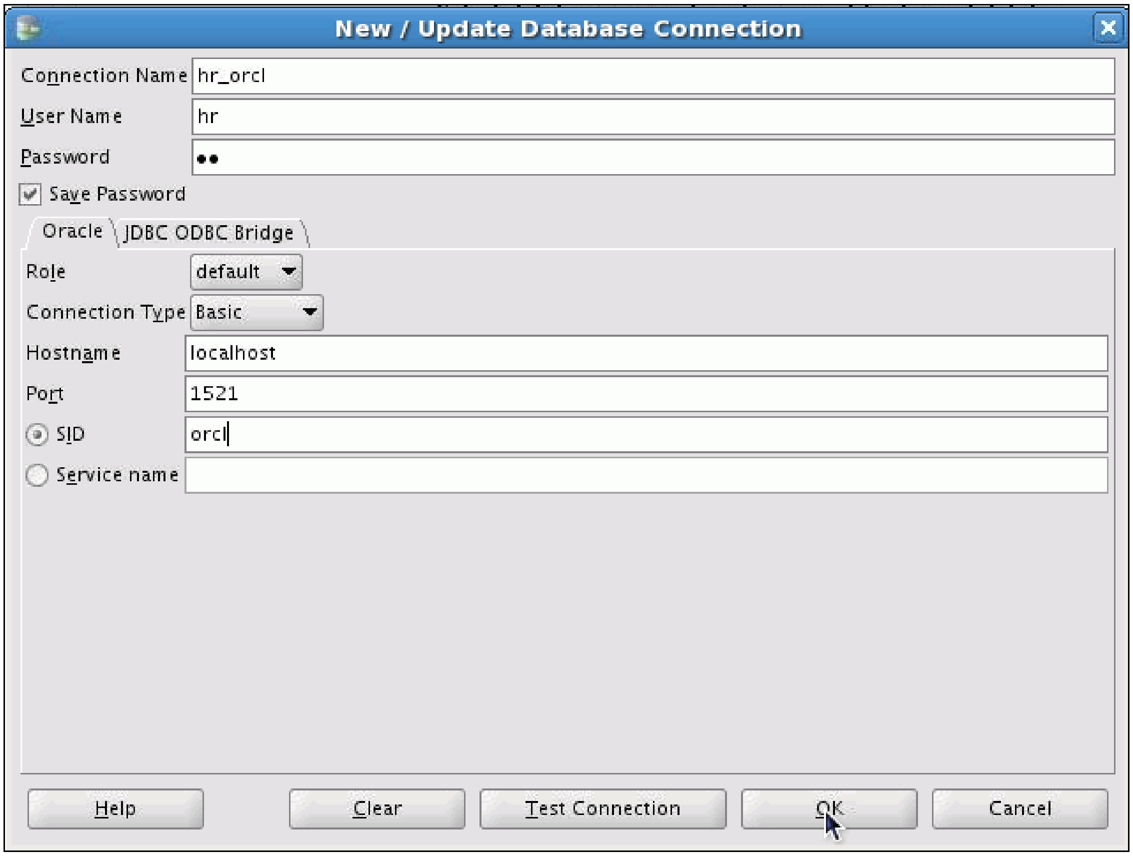
2) Select File > Import, and select Data Dictionary.



3) You want to create a connection to your database. Click the Add button.

4) Enter `hr_orcl` for Connection Name, `hr` for User Name and for Password, specify `orcl` for the SID, and click OK.

Solution 20-1: Re-Engineer the HR Schema (continued)



New / Update Database Connection

Connection Name: hr_orcl

User Name: hr

Password: ●●

☑ Save Password

Oracle | JDBC ODBC Bridge

Role: default

Connection Type: Basic

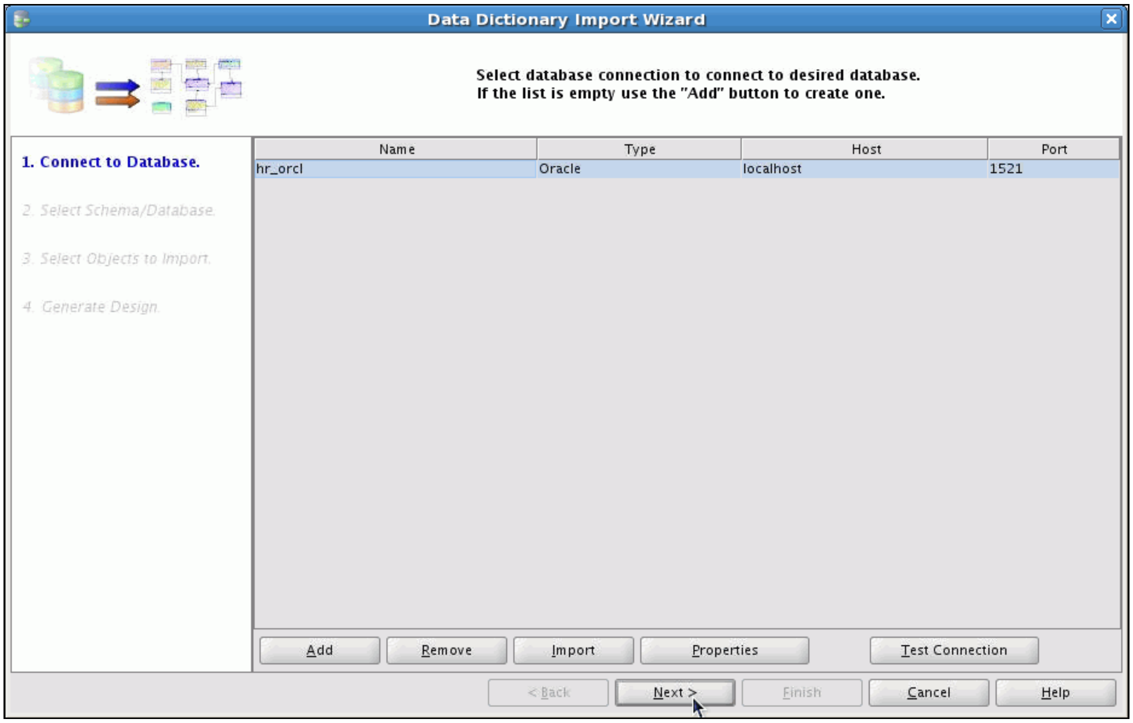
Hostname: localhost

Port: 1521

● SID: orcl

○ Service name:

5) Select the hr_orcl connection that you just created, and click Next.



Data Dictionary Import Wizard

Select database connection to connect to desired database.
If the list is empty use the "Add" button to create one.

1. Connect to Database.

2. Select Schema/Database.

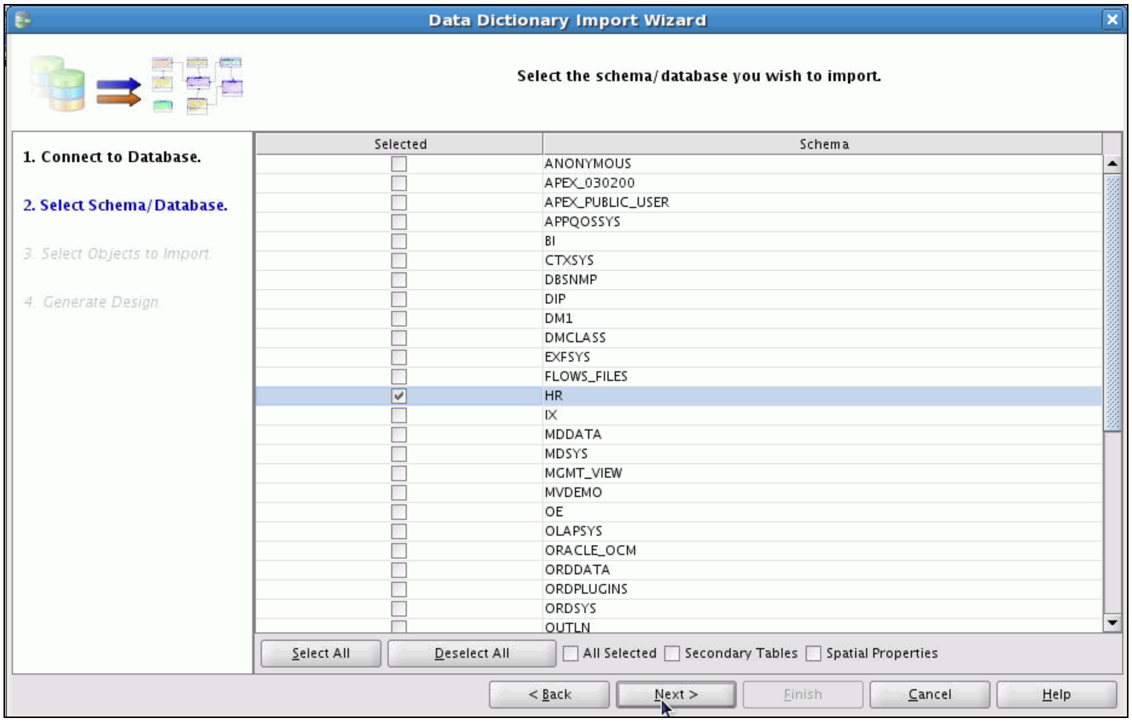
3. Select Objects to Import.

4. Generate Design.

Name	Type	Host	Port
hr_orcl	Oracle	localhost	1521

6) Select the HR schema, and click Next. Note: If you select another schema that the HR user does not have access to, you will see no objects when you click Next.

Solution 20-1: Re-Engineer the HR Schema (continued)



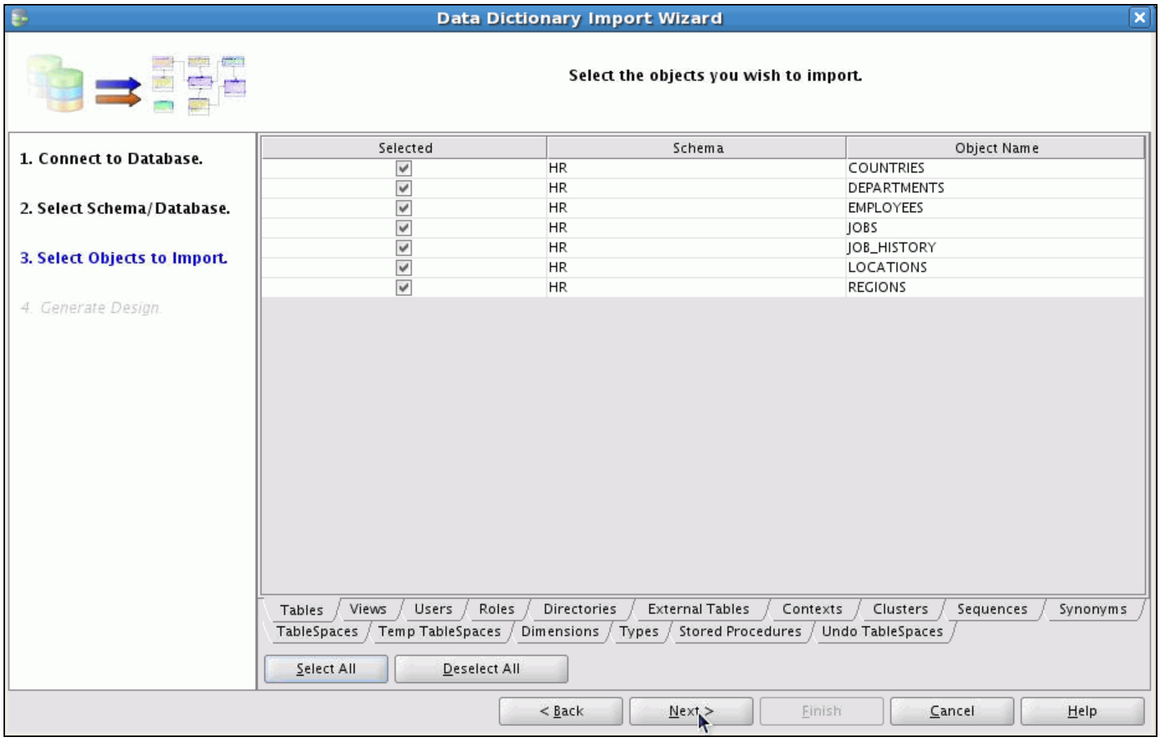
Select the schema/database you wish to import.

1. Connect to Database.
2. Select Schema/Database.
3. Select Objects to Import.
4. Generate Design.

Selected	Schema
□	ANONYMOUS
□	APEX_030200
□	APEX_PUBLIC_USER
□	APPQOSSYS
□	BI
□	CTXSYS
□	DBSNMP
□	DIP
□	DM1
□	DMCLASS
□	EXFSYS
□	FLOWS_FILES
☑	HR
□	IX
□	MDDATA
□	MDSYS
□	MGMT_VIEW
□	MVDEMO
□	OE
□	OLAPSYS
□	ORACLE_OCM
□	ORDDATA
□	ORDPLUGINS
□	ORDSYS
□	OUTLN

□ All Selected □ Secondary Tables □ Spatial Properties

7) Click Select All to select all the HR tables, and click Next.



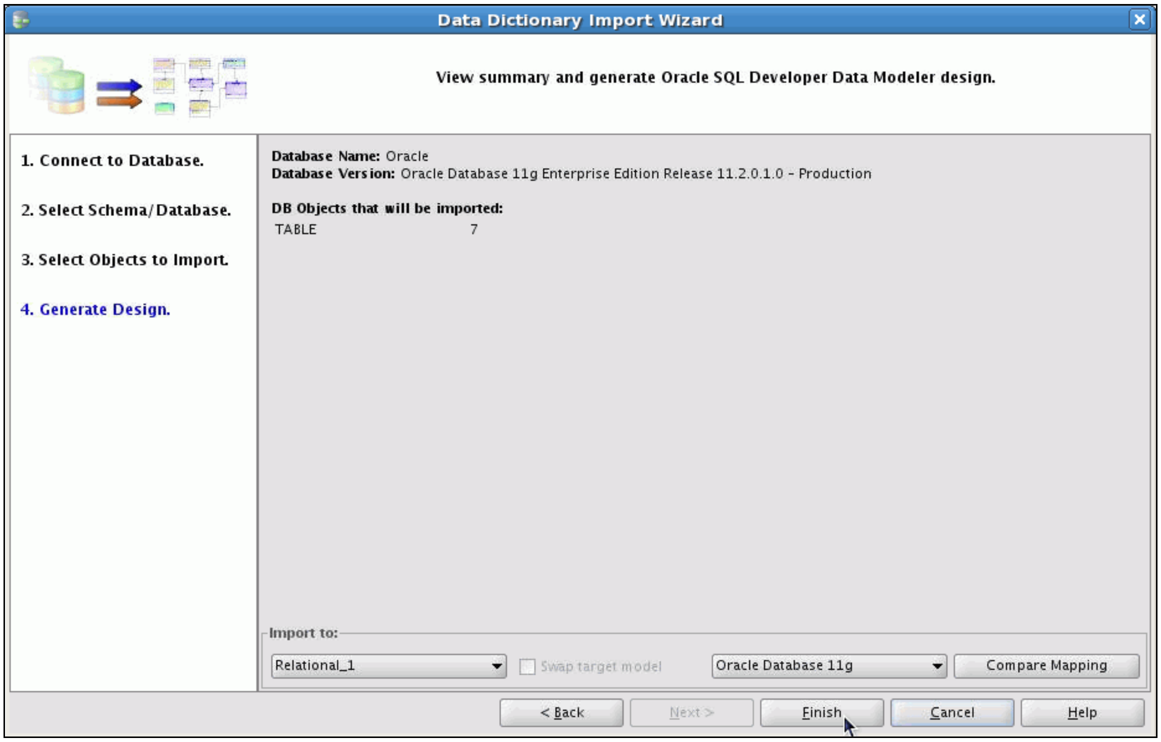
Select the objects you wish to import.

1. Connect to Database.
2. Select Schema/Database.
3. Select Objects to Import.
4. Generate Design.

Selected	Schema	Object Name
☑	HR	COUNTRIES
☑	HR	DEPARTMENTS
☑	HR	EMPLOYEES
☑	HR	JOBS
☑	HR	JOB_HISTORY
☑	HR	LOCATIONS
☑	HR	REGIONS

8) Click Finish.

Solution 20-1: Re-Engineer the HR Schema (continued)



View summary and generate Oracle SQL Developer Data Modeler design.

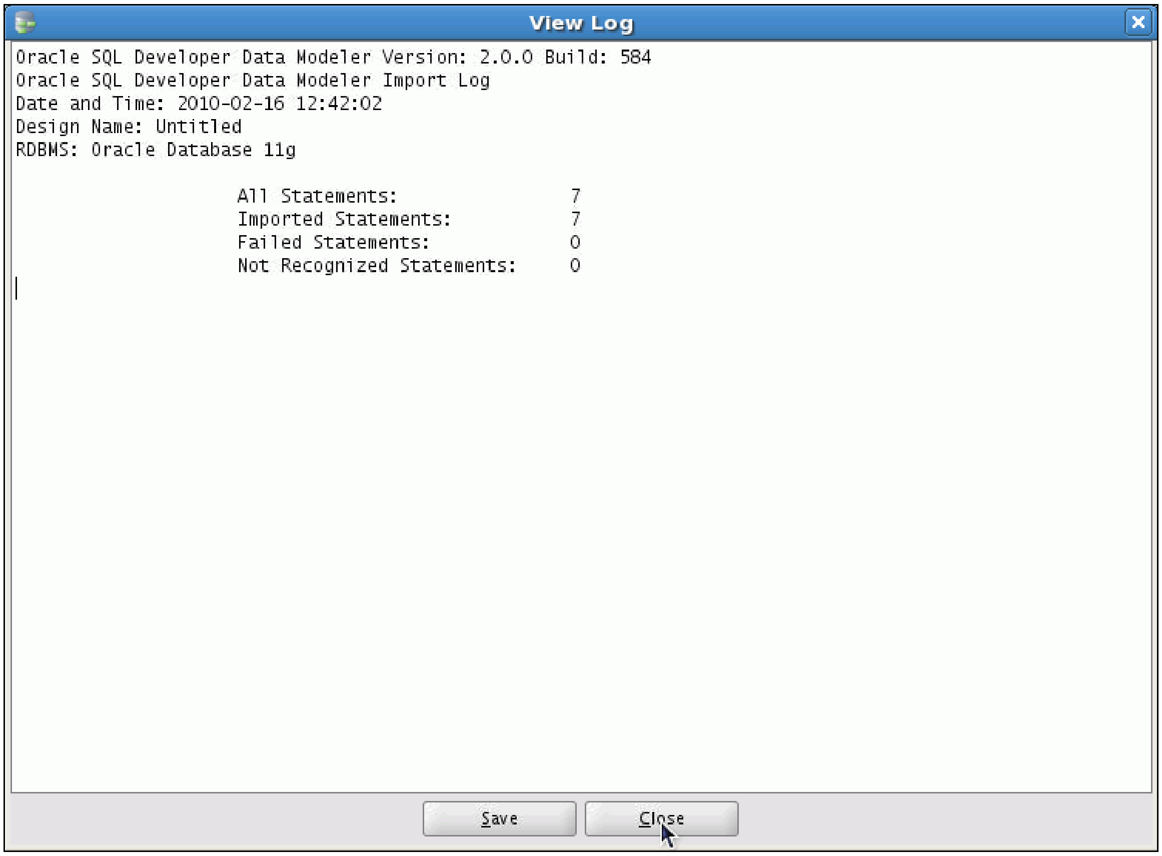
Database Name: Oracle
Database Version: Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production

DB Objects that will be imported:

TABLE	7

Import to: Relational_1 □ Swap target model Oracle Database 11g

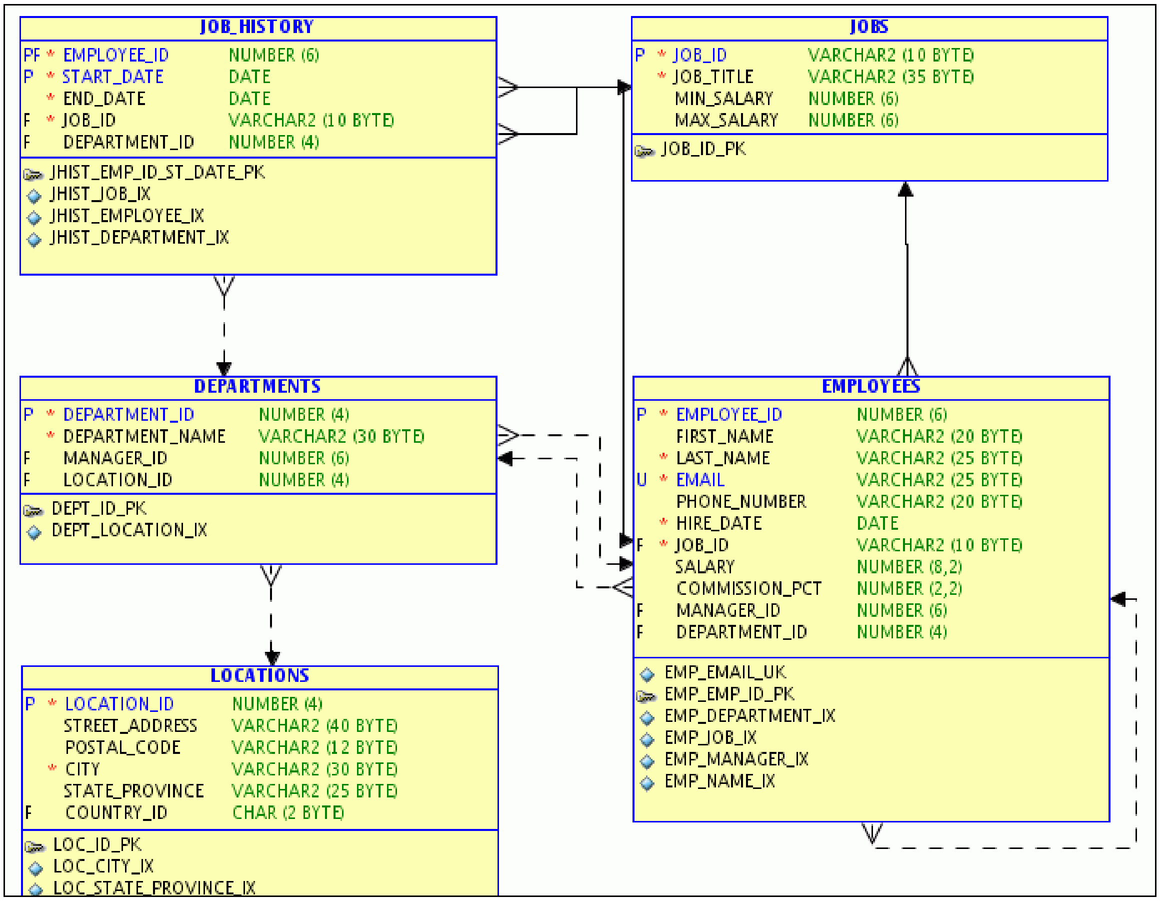
9) Seven tables were imported successfully. Click Close.



```
Oracle SQL Developer Data Modeler Version: 2.0.0 Build: 584
Oracle SQL Developer Data Modeler Import Log
Date and Time: 2010-02-16 12:42:02
Design Name: Untitled
RDBMS: Oracle Database 11g

                    All Statements:                 7
                    Imported Statements:            7
                    Failed Statements:              0
                    Not Recognized Statements:      0
```

10) Your relational model is displayed.

Solution 20-1: Re-Engineer the HR Schema (continued)



JOB_HISTORY

PF	* EMPLOYEE_ID	NUMBER (6)
P	* START_DATE	DATE
	* END_DATE	DATE
F	* JOB_ID	VARCHAR2 (10 BYTE)
F	DEPARTMENT_ID	NUMBER (4)

JHIST_EMP_ID_ST_DATE_PK
JHIST_JOB_IX
JHIST_EMPLOYEE_IX
JHIST_DEPARTMENT_IX

JOBS

P	* JOB_ID	VARCHAR2 (10 BYTE)
	* JOB_TITLE	VARCHAR2 (35 BYTE)
	MIN_SALARY	NUMBER (6)
	MAX_SALARY	NUMBER (6)

JOB_ID_PK

DEPARTMENTS

P	* DEPARTMENT_ID	NUMBER (4)
	* DEPARTMENT_NAME	VARCHAR2 (30 BYTE)
F	MANAGER_ID	NUMBER (6)
F	LOCATION_ID	NUMBER (4)

DEPT_ID_PK
DEPT_LOCATION_IX

EMPLOYEES

P	* EMPLOYEE_ID	NUMBER (6)
	FIRST_NAME	VARCHAR2 (20 BYTE)
	* LAST_NAME	VARCHAR2 (25 BYTE)
U	* EMAIL	VARCHAR2 (25 BYTE)
	PHONE_NUMBER	VARCHAR2 (20 BYTE)
	* HIRE_DATE	DATE
F	* JOB_ID	VARCHAR2 (10 BYTE)
	SALARY	NUMBER (8,2)
	COMMISSION_PCT	NUMBER (2,2)
F	MANAGER_ID	NUMBER (6)
F	DEPARTMENT_ID	NUMBER (4)

EMP_EMAIL_UK
EMP_EMP_ID_PK
EMP_DEPARTMENT_IX
EMP_JOB_IX
EMP_MANAGER_IX
EMP_NAME_IX

LOCATIONS

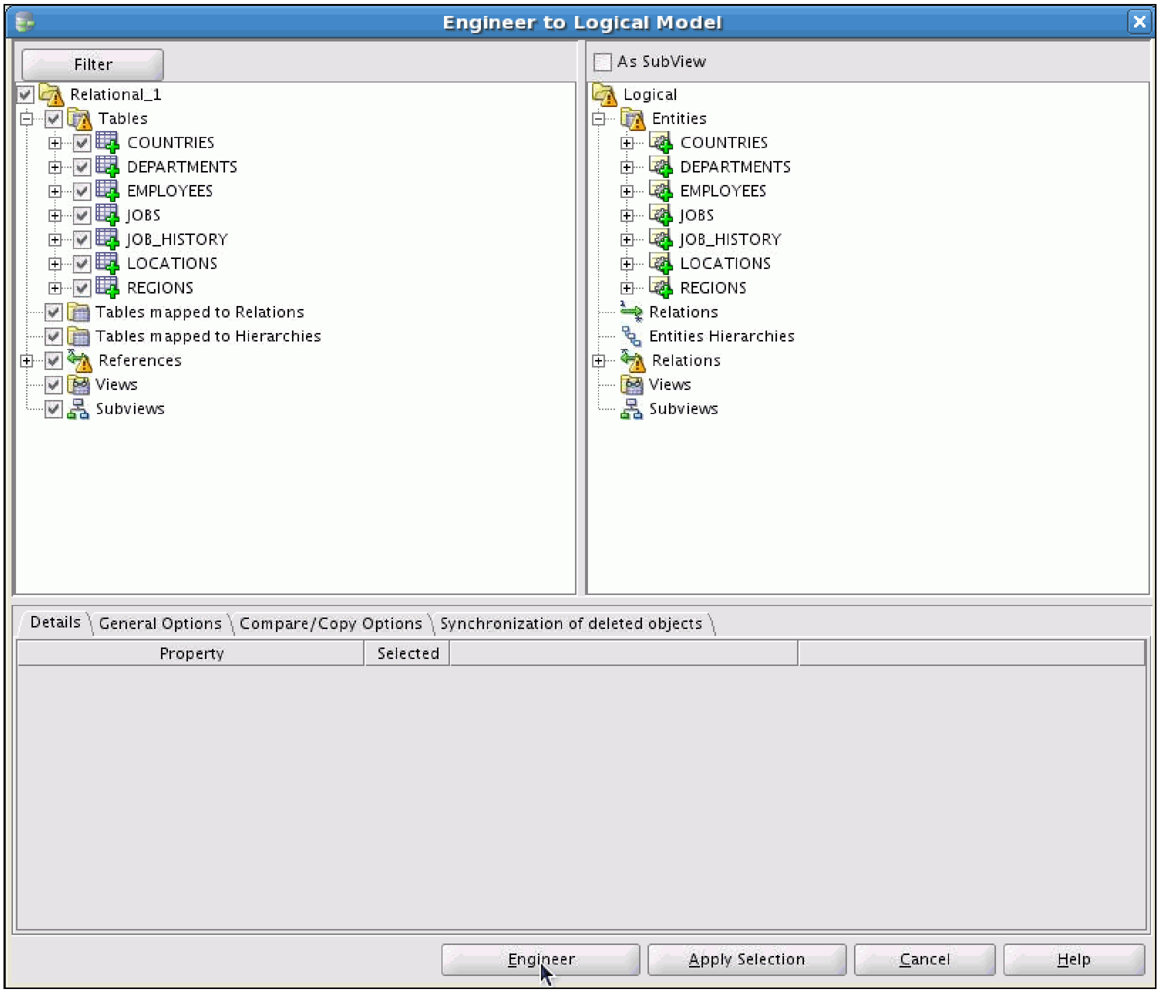
P	* LOCATION_ID	NUMBER (4)
	STREET_ADDRESS	VARCHAR2 (40 BYTE)
	POSTAL_CODE	VARCHAR2 (12 BYTE)
	* CITY	VARCHAR2 (30 BYTE)
	STATE_PROVINCE	VARCHAR2 (25 BYTE)
F	COUNTRY_ID	CHAR (2 BYTE)

LOC_ID_PK
LOC_CITY_IX
LOC_STATE_PROVINCE_IX

11) You can now reverse engineer to create the logical model. Click the “Engineer to Logical Model” « icon.

12) Expand the Table node to see the mapping. Click Engineer.

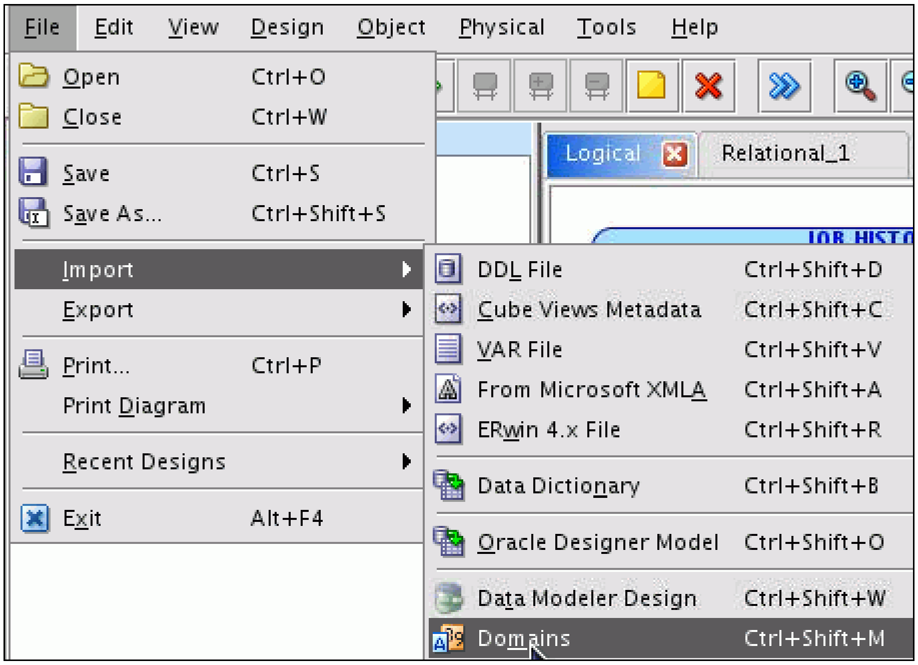
Solution 20-1: Re-Engineer the HR Schema (continued)



13) Your logical model is displayed.

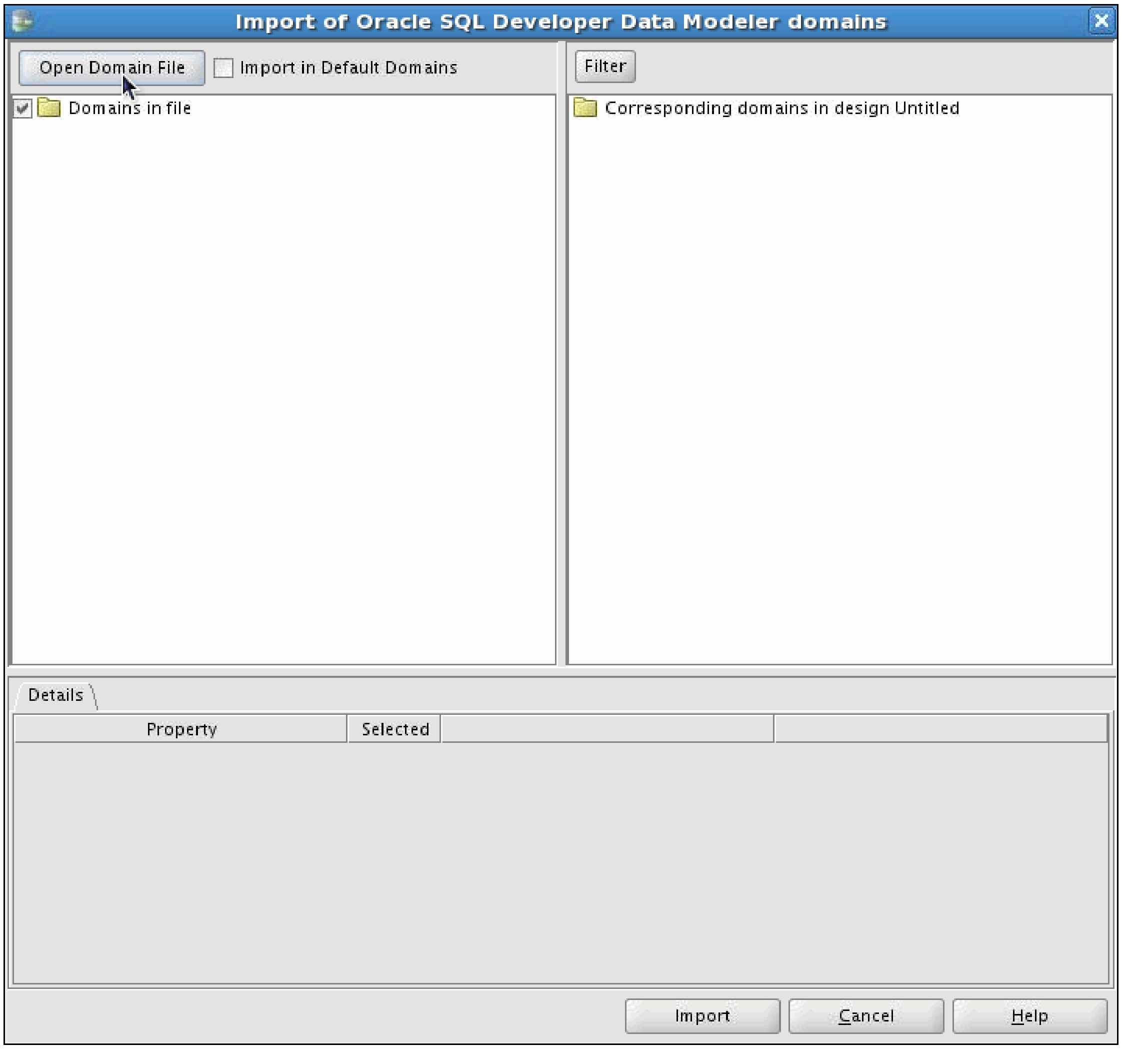
14) You want to create some attributes that use the domains contained in the `domains.xml` file in the `/home/oracle/solutions` directory. You will need to import this domain file. Select File > Import > Domains.

Solution 20-1: Re-Engineer the HR Schema (continued)



15) Click Open Domain File.

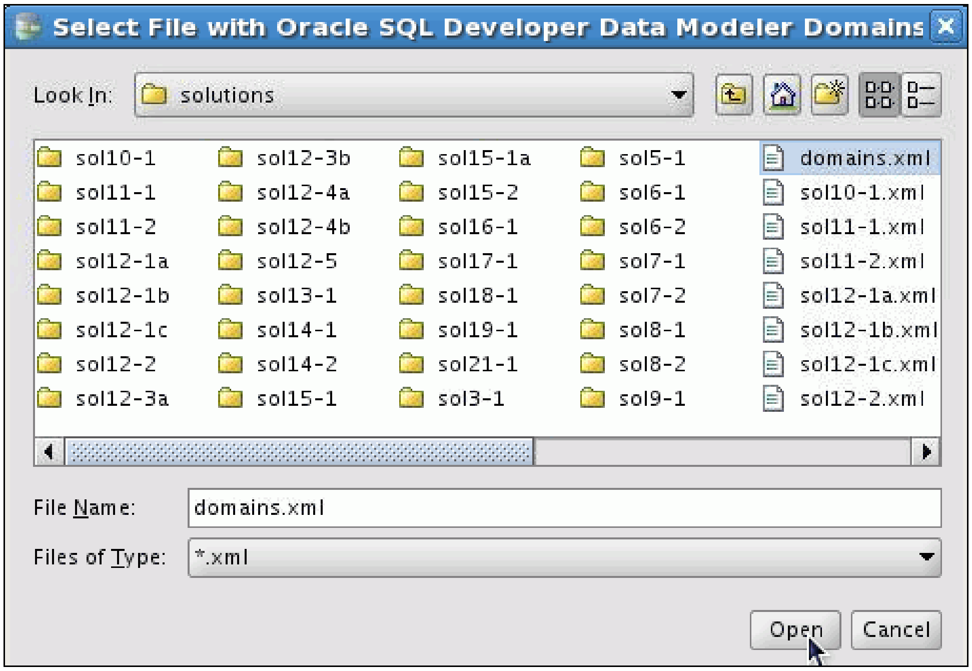
Solution 20-1: Re-Engineer the HR Schema (continued)


Import of Oracle SQL Developer Data Modeler domains
Open Domain File
□ Import in Default Domains
Filter
☑ Domains in file
Corresponding domains in design Untitled
Details
Property
Selected
Import
Cancel
Help

16) Select the `/home/oracle/solutions` directory, enter `*.xml` in the File Name field, and press Enter.

17) Select `domains.xml`, and click Open.

Solution 20-1: Re-Engineer the HR Schema (continued)

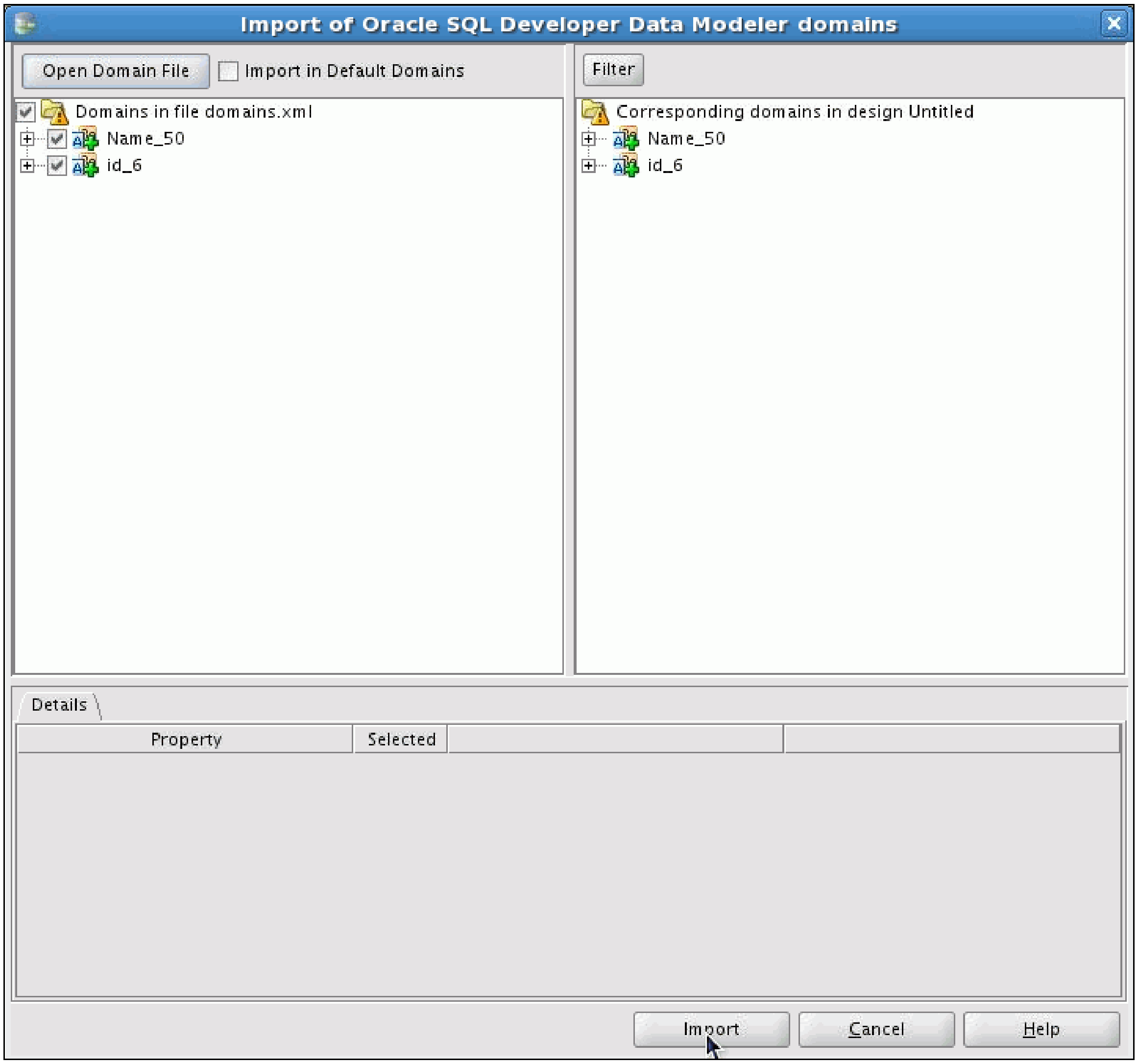


File Name: domains.xml

Files of Type: *.xml

18) The list of domains in the XML file is displayed. Click Import.

Solution 20-1: Re-Engineer the HR Schema (continued)



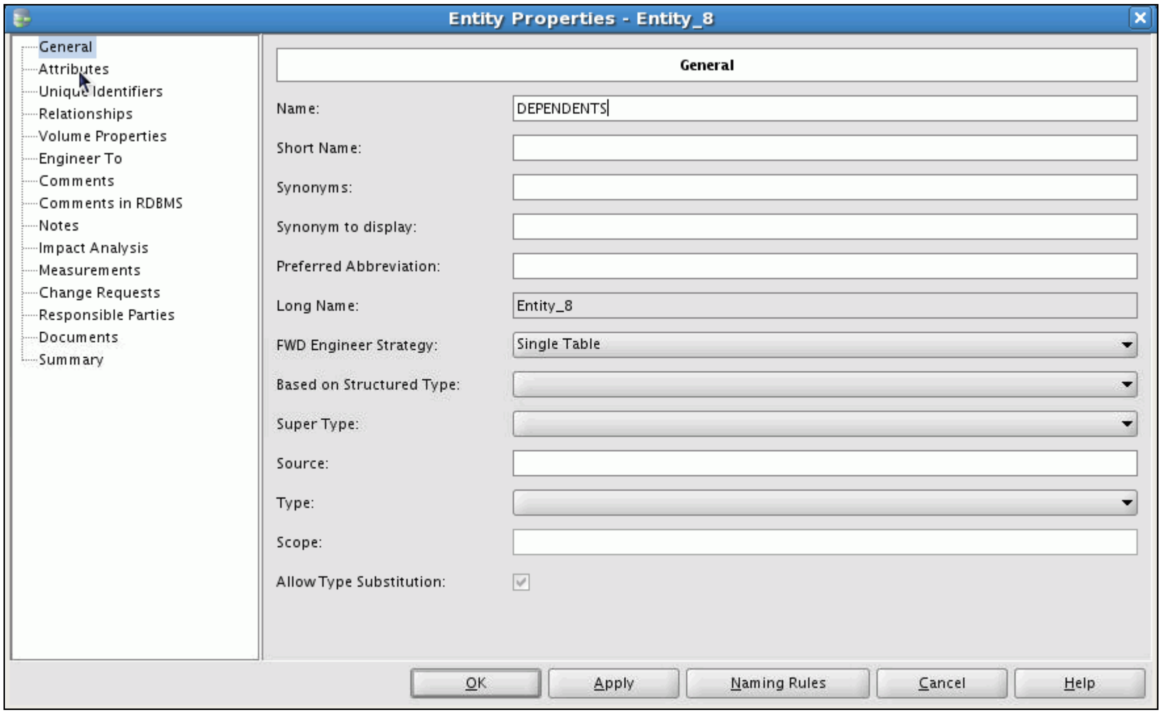
Import of Oracle SQL Developer Data Modeler domains

Open Domain File
□ Import in Default Domains
Filter
☑ Domains in file domains.xml
☑ Name_50
☑ id_6
Corresponding domains in design Untitled
Name_50
id_6
Details
Property
Selected
Import
Cancel
Help

19) Now you will create a new entity. Click the New Entity icon and click in the white space of the diagram.

20) Enter DEPENDENTS for Name, and select the Attributes property in the left navigator.

Solution 20-1: Re-Engineer the HR Schema (continued)



General

Name: DEPENDENTS

Short Name:

Synonyms:

Synonym to display:

Preferred Abbreviation:

Long Name: Entity_8

FWD Engineer Strategy: Single Table

Based on Structured Type:

Super Type:

Source:

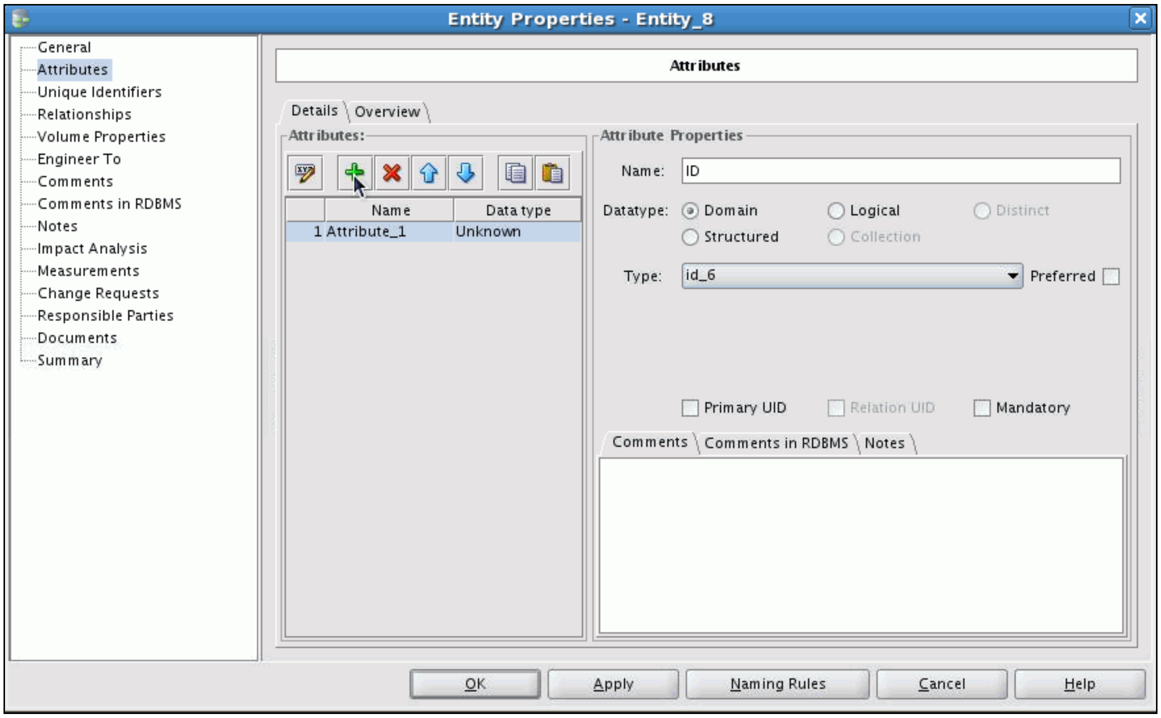
Type:

Scope:

Allow Type Substitution: ☑

21) Click the Add ➕ icon.

22) Enter `ID` for Name, select the id_6 domain for Type, and click the Add ➕ icon.



Attributes

Attributes:

	Name	Data type
1	Attribute_1	Unknown

Attribute Properties

Name: ID

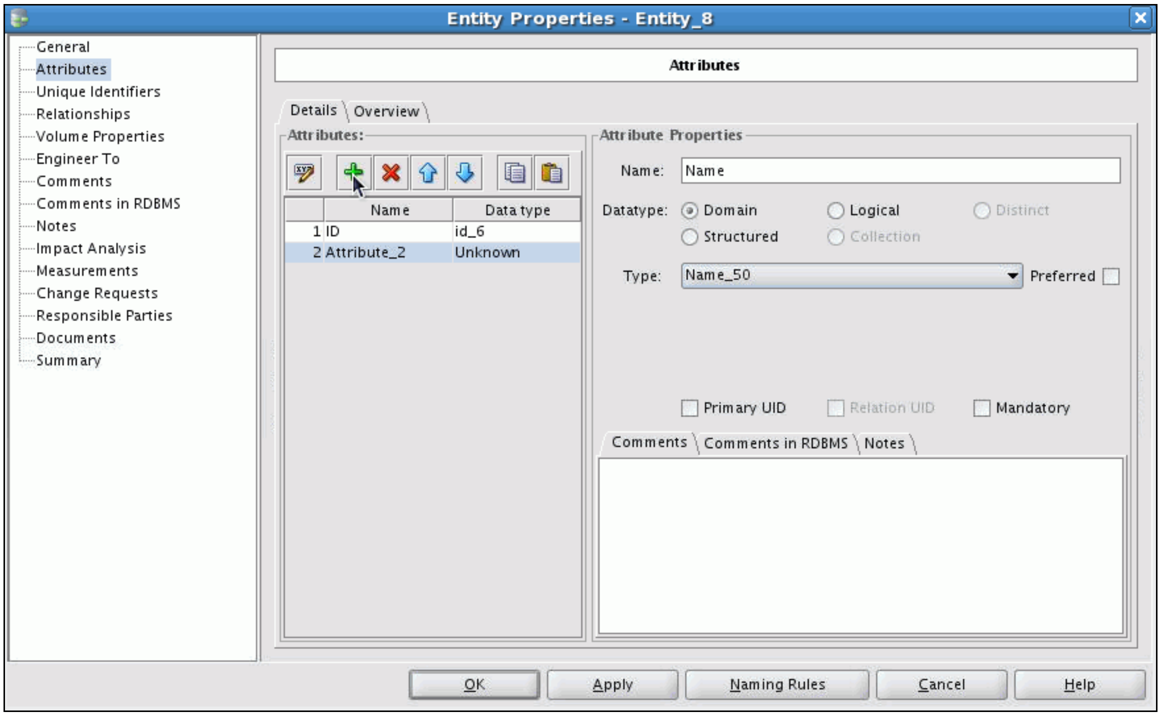
Datatype: ● Domain ○ Logical ○ Distinct ○ Structured ○ Collection

Type: id_6 Preferred □

□ Primary UID □ Relation UID □ Mandatory

23) Enter `Name` for Name, select the Name_50 domain for Type, and click the Add ➕ icon.

Solution 20-1: Re-Engineer the HR Schema (continued)



	Name	Data type
1	ID	id_6
2	Attribute_2	Unknown

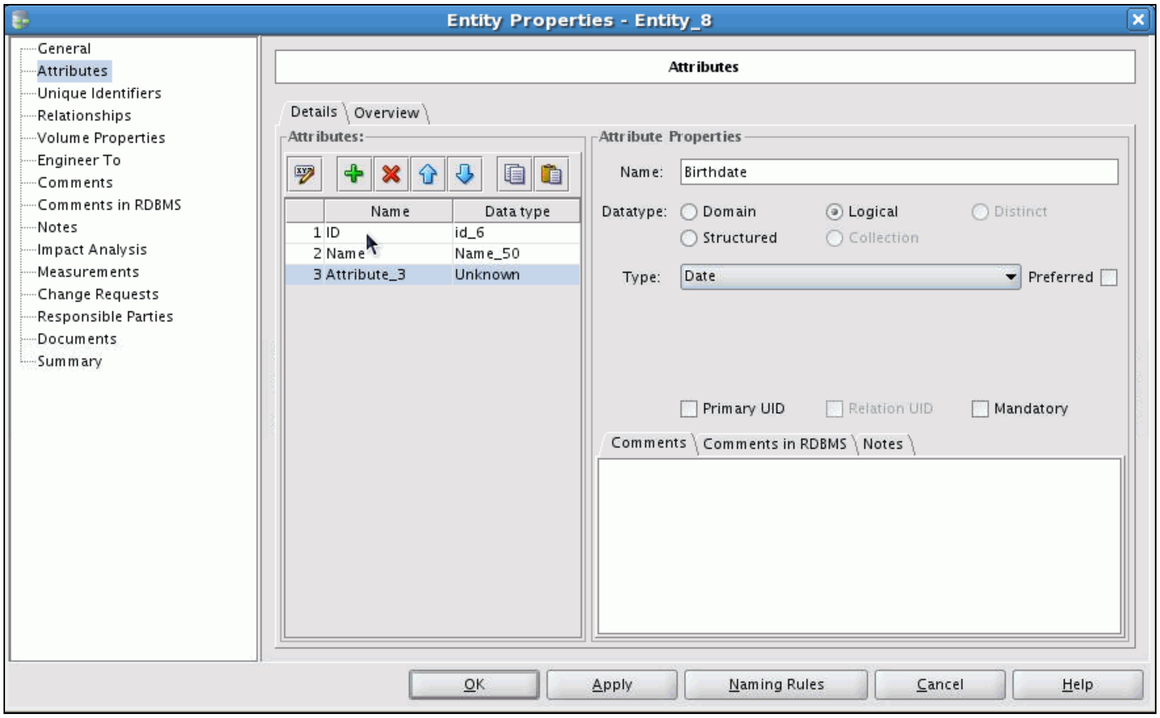
Name: Name

Datatype: ● Domain ○ Logical ○ Distinct ○ Structured ○ Collection

Type: Name_50 □ Preferred

□ Primary UID □ Relation UID □ Mandatory

24) Enter `Birthdate` for Name, select Logical for Datatype, and select Date for Type. To specify ID as the unique identifier, select ID in the Attributes list.



	Name	Data type
1	ID	id_6
2	Name	Name_50
3	Attribute_3	Unknown

Name: Birthdate

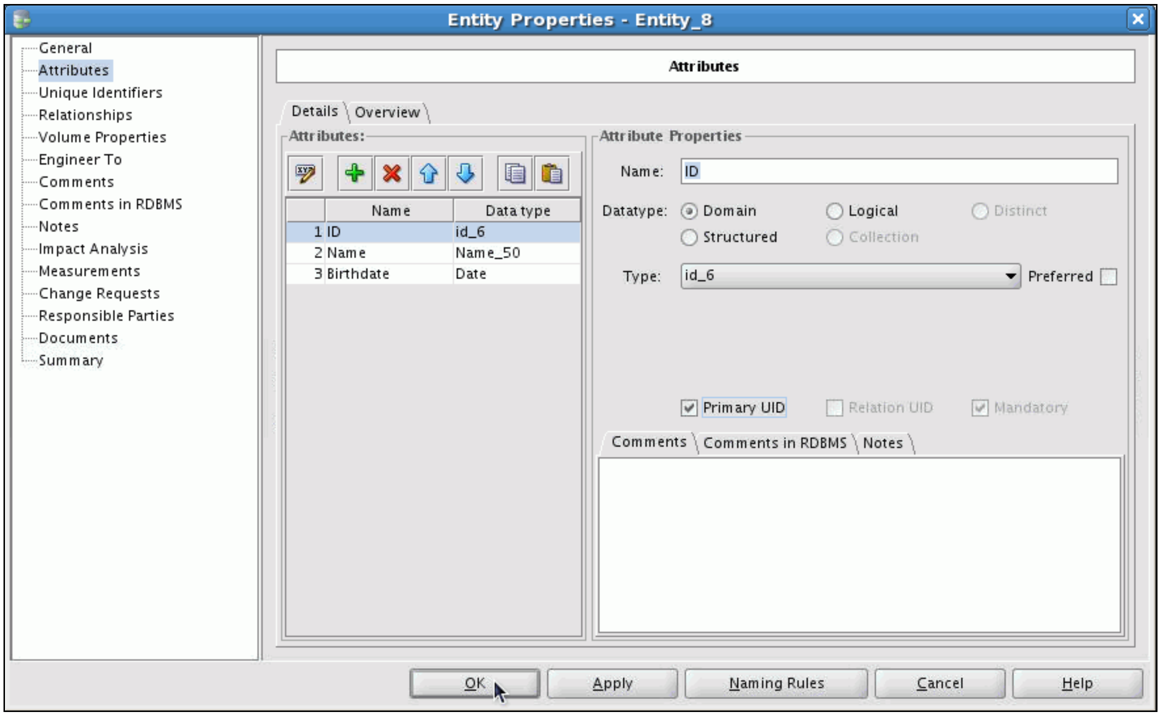
Datatype: ○ Domain ● Logical ○ Distinct ○ Structured ○ Collection

Type: Date □ Preferred

□ Primary UID □ Relation UID □ Mandatory

25) Select the Primary UID check box and click OK to create the entity and attributes.

Solution 20-1: Re-Engineer the HR Schema (continued)



	Name	Data type
1	ID	id_6
2	Name	Name_50
3	Birthdate	Date

Name: ID

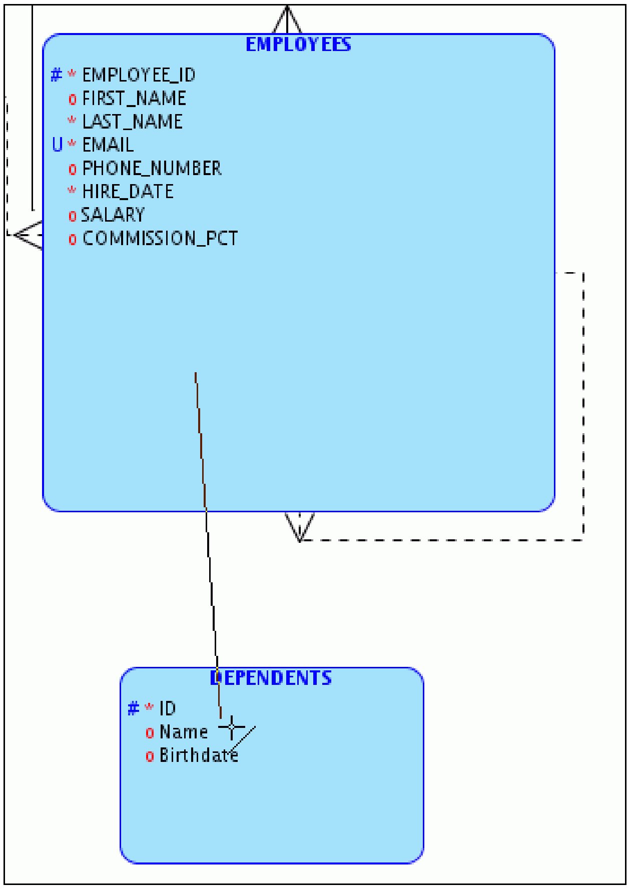
Datatype: ● Domain ○ Logical ○ Distinct ○ Structured ○ Collection

Type: id_6 Preferred □

☑ Primary UID □ Relation UID ☑ Mandatory

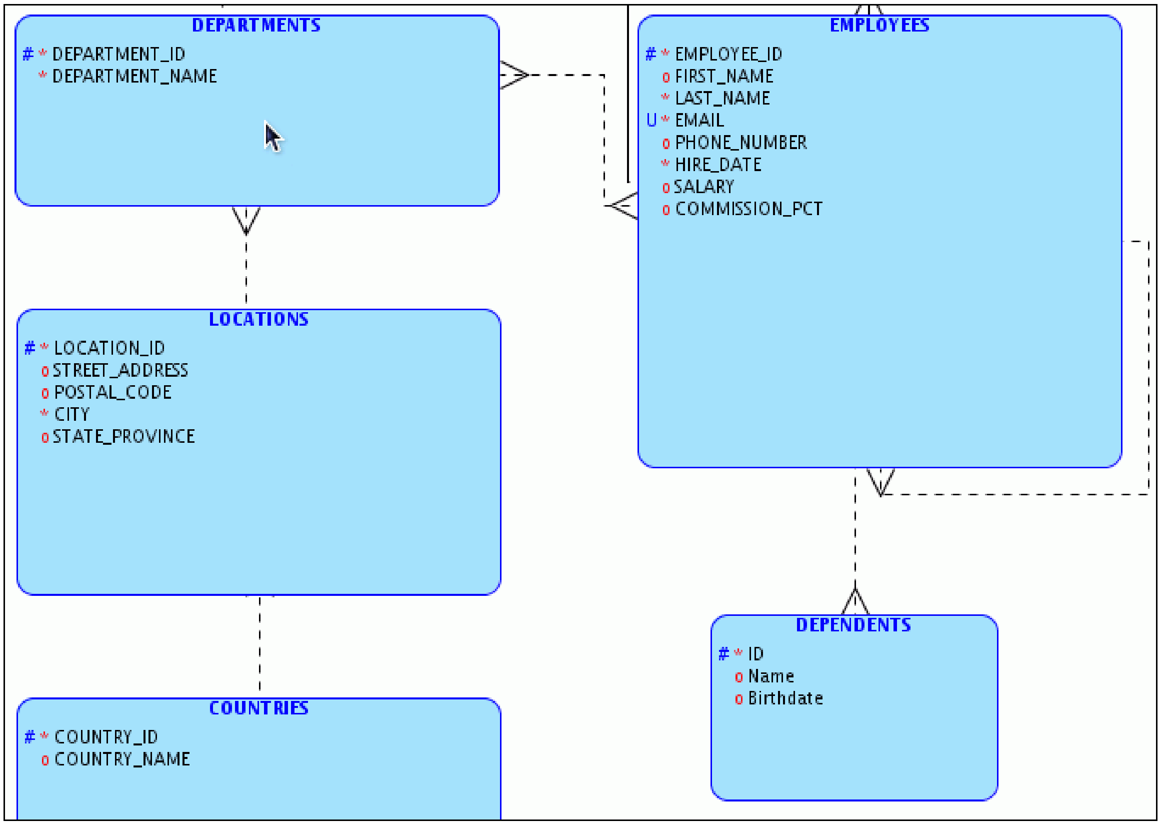
26) You want to create a 1:M relationship between EMPLOYEES and DEPENDENTS. Select the 1:M icon, and click EMPLOYEES and then DEPENDENTS.

Solution 20-1: Re-Engineer the HR Schema (continued)



27) The 1:M relationship is created. You can make some more changes to the logical model. Double-click the DEPARTMENTS entity.

Solution 20-1: Re-Engineer the HR Schema (continued)

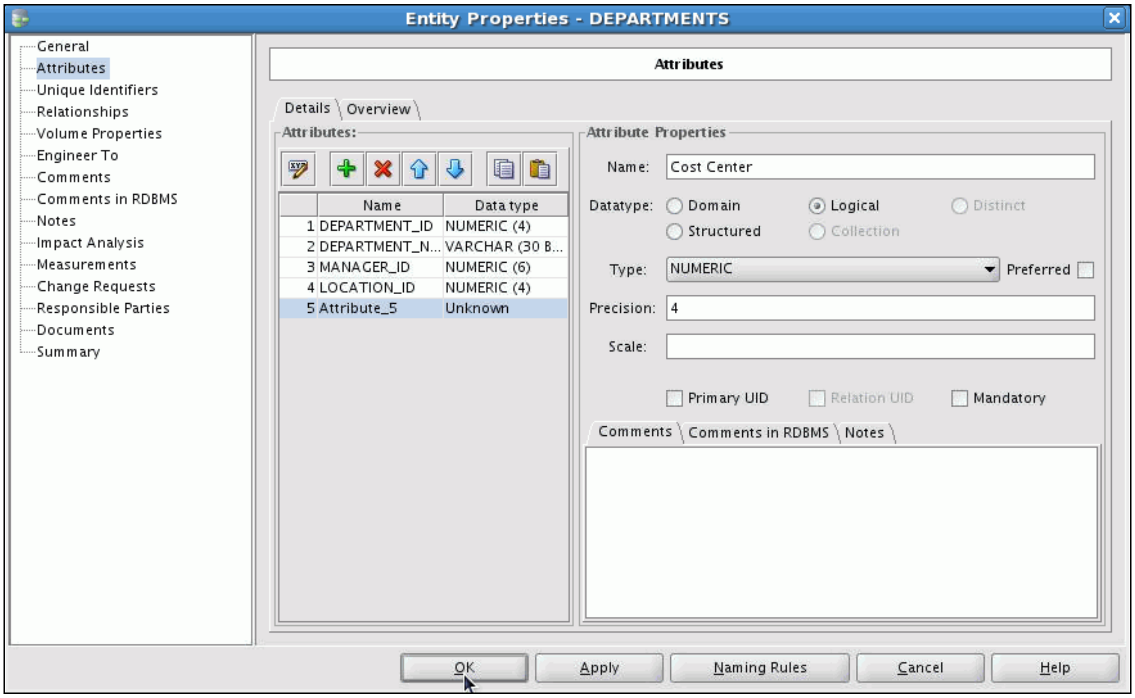


28) Select the Attributes property in the left navigator.

29) Click Add ✚ icon.

30) Enter `Cost Center` for Name, select Logical for Datatype, select NUMERIC for Type, and enter `4` for Precision. Then click OK.

Solution 20-1: Re-Engineer the HR Schema (continued)


Entity Properties - DEPARTMENTS

Attributes

Attributes:

	Name	Data type
1	DEPARTMENT_ID	NUMERIC (4)
2	DEPARTMENT_N...	VARCHAR (30 B...
3	MANAGER_ID	NUMERIC (6)
4	LOCATION_ID	NUMERIC (4)
5	Attribute_5	Unknown

Attribute Properties

Name: Cost Center

Datatype: ○ Domain ● Logical ○ Distinct
○ Structured ○ Collection

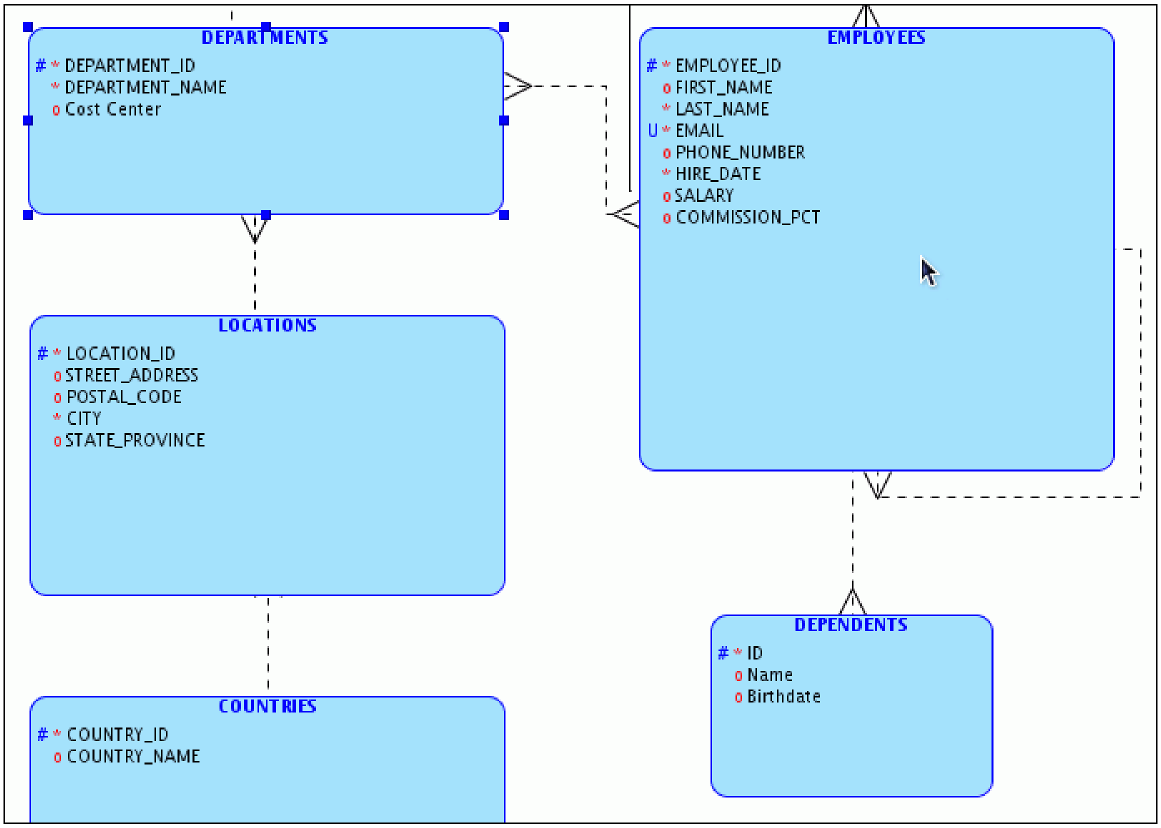
Type: NUMERIC Preferred □

Precision: 4

Scale:

□ Primary UID □ Relation UID □ Mandatory

31) Double-click the EMPLOYEES table.


DEPARTMENTS
* DEPARTMENT_ID
* DEPARTMENT_NAME
o Cost Center

EMPLOYEES
* EMPLOYEE_ID
o FIRST_NAME
* LAST_NAME
U * EMAIL
o PHONE_NUMBER
* HIRE_DATE
o SALARY
o COMMISSION_PCT

LOCATIONS
* LOCATION_ID
o STREET_ADDRESS
o POSTAL_CODE
* CITY
o STATE_PROVINCE

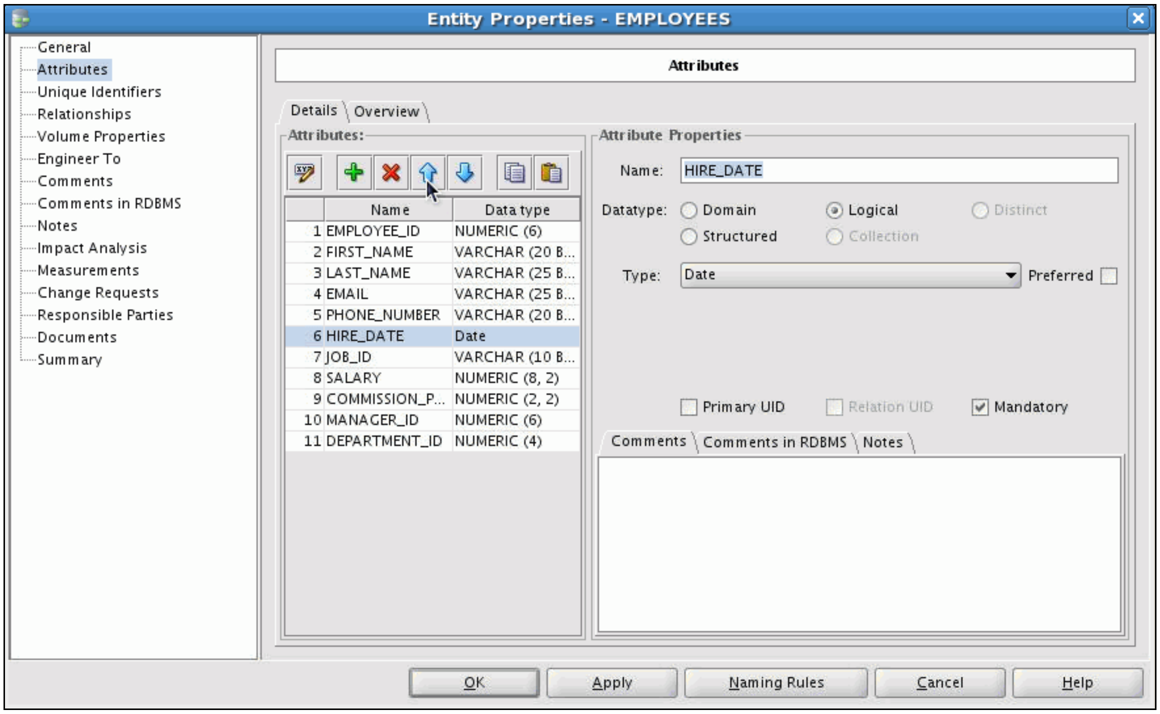
DEPENDENTS
* ID
o Name
o Birthdate

COUNTRIES
* COUNTRY_ID
o COUNTRY_NAME

32) Select the Attributes property in the left navigator.

Solution 20-1: Re-Engineer the HR Schema (continued)

33) Select the HIRE_DATE attribute, and click the Move Up ⇧ icon two times so that the HIRE_DATE attribute appears before the EMAIL attribute.



	Name	Data type
1	EMPLOYEE_ID	NUMERIC (6)
2	FIRST_NAME	VARCHAR (20 B...
3	LAST_NAME	VARCHAR (25 B...
4	EMAIL	VARCHAR (25 B...
5	PHONE_NUMBER	VARCHAR (20 B...
6	HIRE_DATE	Date
7	JOB_ID	VARCHAR (10 B...
8	SALARY	NUMERIC (8, 2)
9	COMMISSION_P...	NUMERIC (2, 2)
10	MANAGER_ID	NUMERIC (6)
11	DEPARTMENT_ID	NUMERIC (4)

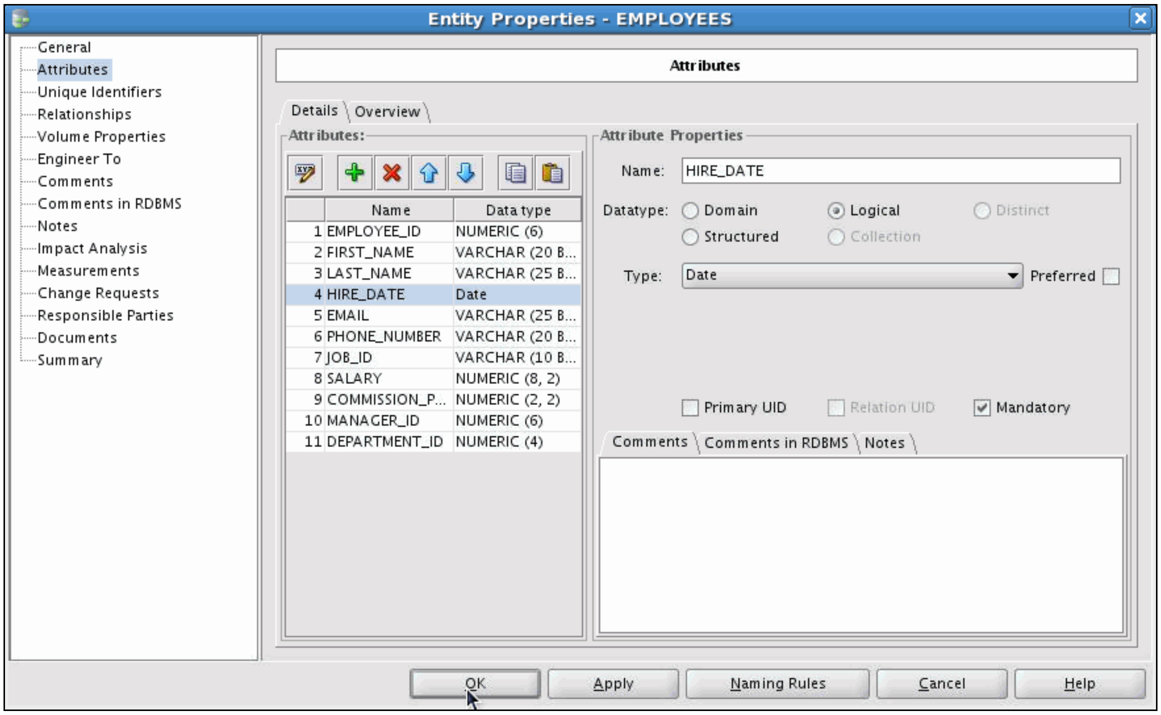
Name: HIRE_DATE

Datatype: ○ Domain ● Logical ○ Distinct ○ Structured ○ Collection

Type: Date Preferred □

□ Primary UID □ Relation UID ☑ Mandatory

34) Click OK.



	Name	Data type
1	EMPLOYEE_ID	NUMERIC (6)
2	FIRST_NAME	VARCHAR (20 B...
3	LAST_NAME	VARCHAR (25 B...
4	HIRE_DATE	Date
5	EMAIL	VARCHAR (25 B...
6	PHONE_NUMBER	VARCHAR (20 B...
7	JOB_ID	VARCHAR (10 B...
8	SALARY	NUMERIC (8, 2)
9	COMMISSION_P...	NUMERIC (2, 2)
10	MANAGER_ID	NUMERIC (6)
11	DEPARTMENT_ID	NUMERIC (4)

Name: HIRE_DATE

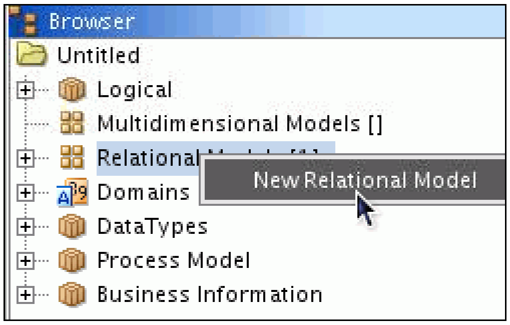
Datatype: ○ Domain ● Logical ○ Distinct ○ Structured ○ Collection

Type: Date Preferred □

□ Primary UID □ Relation UID ☑ Mandatory

35) Now you will forward engineer these changes to a new relational model and then compare the relational model to what is in the database. Right-click Relational Model in the object browser, and select New Relational Model.

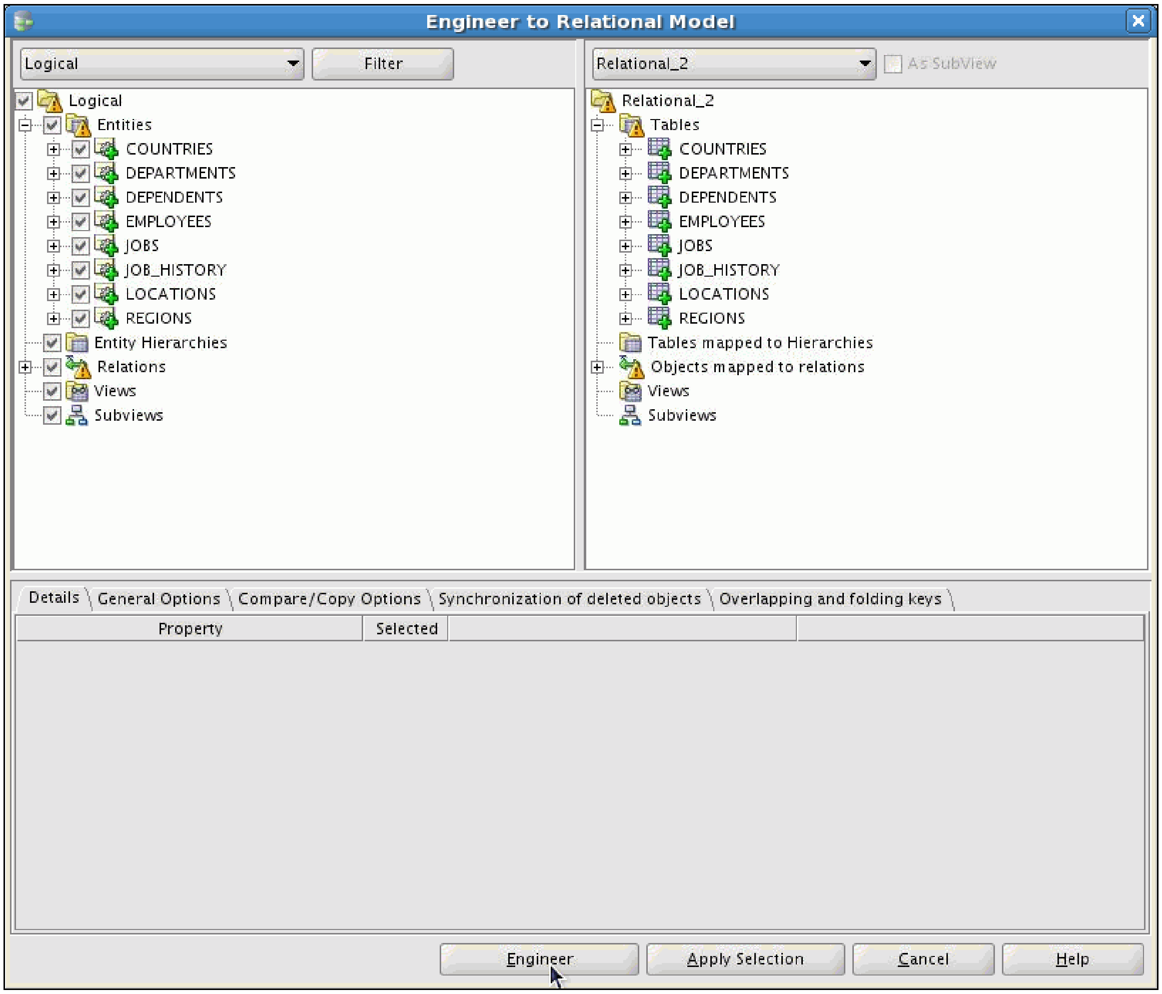
Solution 20-1: Re-Engineer the HR Schema (continued)



36) Click the Logical tab to return to your entity relationship diagram.

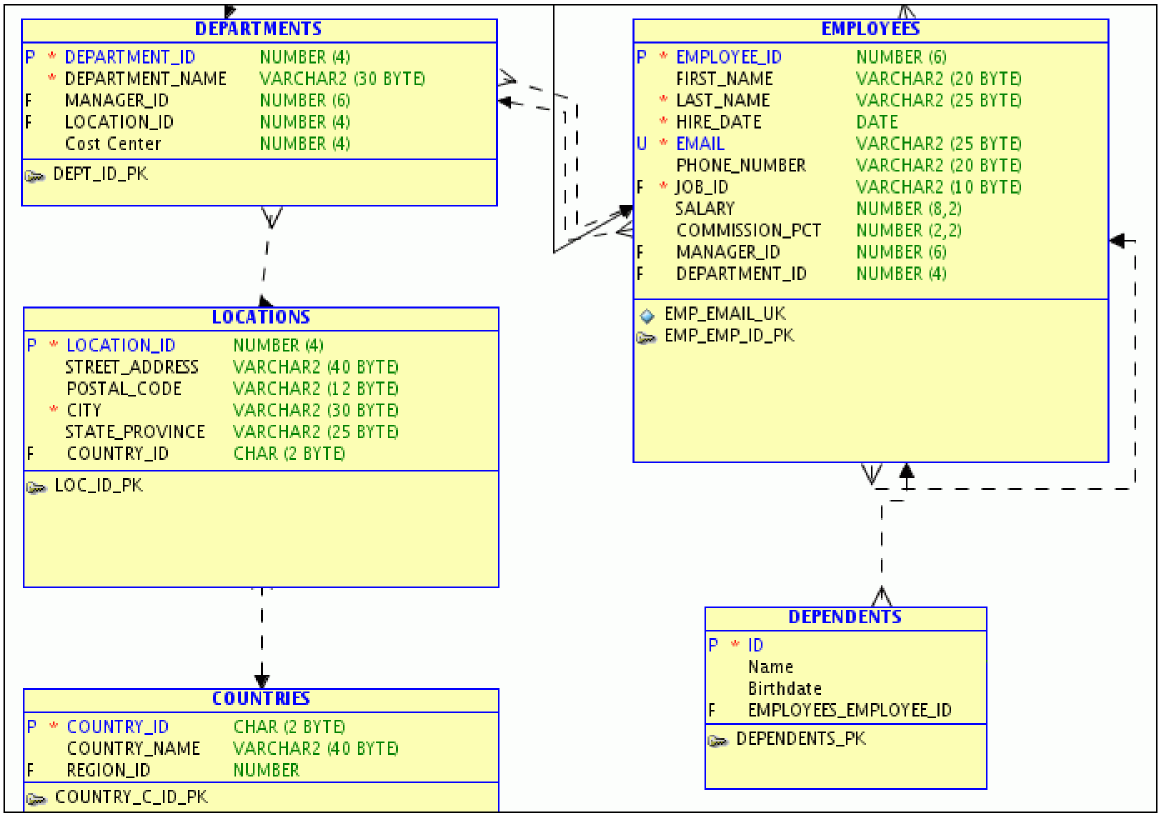
37) Click the “Engineer to Relational Model” ≫ icon.

38) Make sure that Relational_2 is selected at the top right. You can expand some of the nodes to see what will be created in the new relational model. Then click Engineer.



39) Your relational model was created with your modifications.

Solution 20-1: Re-Engineer the HR Schema (continued)



DEPARTMENTS

P	*	DEPARTMENT_ID	NUMBER (4)
	*	DEPARTMENT_NAME	VARCHAR2 (30 BYTE)
F		MANAGER_ID	NUMBER (6)
F		LOCATION_ID	NUMBER (4)
		Cost Center	NUMBER (4)

DEPT_ID_PK

EMPLOYEES

P	*	EMPLOYEE_ID	NUMBER (6)
		FIRST_NAME	VARCHAR2 (20 BYTE)
	*	LAST_NAME	VARCHAR2 (25 BYTE)
	*	HIRE_DATE	DATE
U	*	EMAIL	VARCHAR2 (25 BYTE)
		PHONE_NUMBER	VARCHAR2 (20 BYTE)
F	*	JOB_ID	VARCHAR2 (10 BYTE)
		SALARY	NUMBER (8,2)
		COMMISSION_PCT	NUMBER (2,2)
F		MANAGER_ID	NUMBER (6)
F		DEPARTMENT_ID	NUMBER (4)

EMP_EMAIL_UK
EMP_EMP_ID_PK

LOCATIONS

P	*	LOCATION_ID	NUMBER (4)
		STREET_ADDRESS	VARCHAR2 (40 BYTE)
		POSTAL_CODE	VARCHAR2 (12 BYTE)
	*	CITY	VARCHAR2 (30 BYTE)
		STATE_PROVINCE	VARCHAR2 (25 BYTE)
F		COUNTRY_ID	CHAR (2 BYTE)

LOC_ID_PK

DEPENDENTS

P	*	ID
		Name
		Birthdate
F		EMPLOYEES_EMPLOYEE_ID

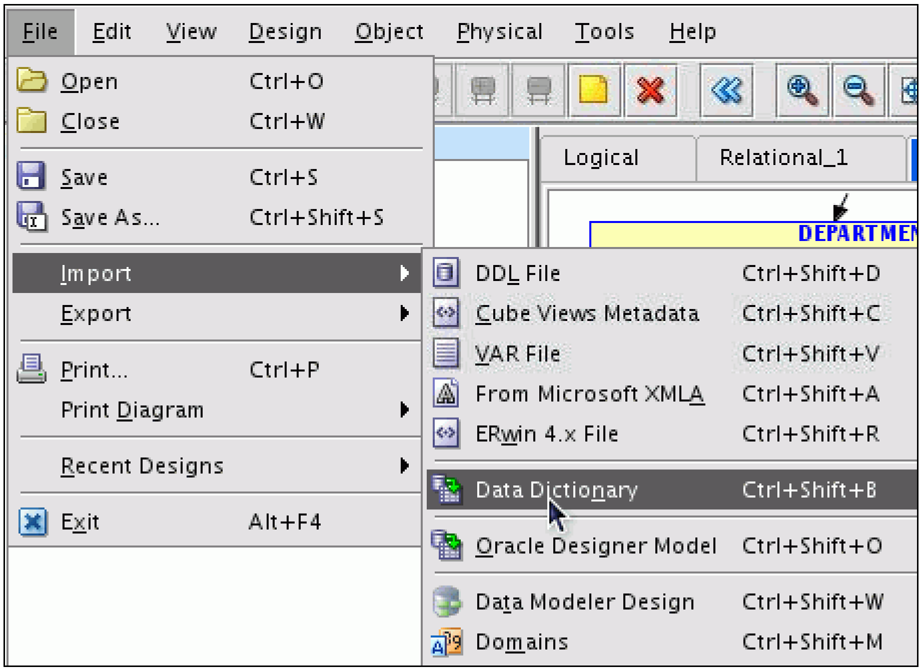
DEPENDENTS_PK

COUNTRIES

P	*	COUNTRY_ID	CHAR (2 BYTE)
		COUNTRY_NAME	VARCHAR2 (40 BYTE)
F		REGION_ID	NUMBER

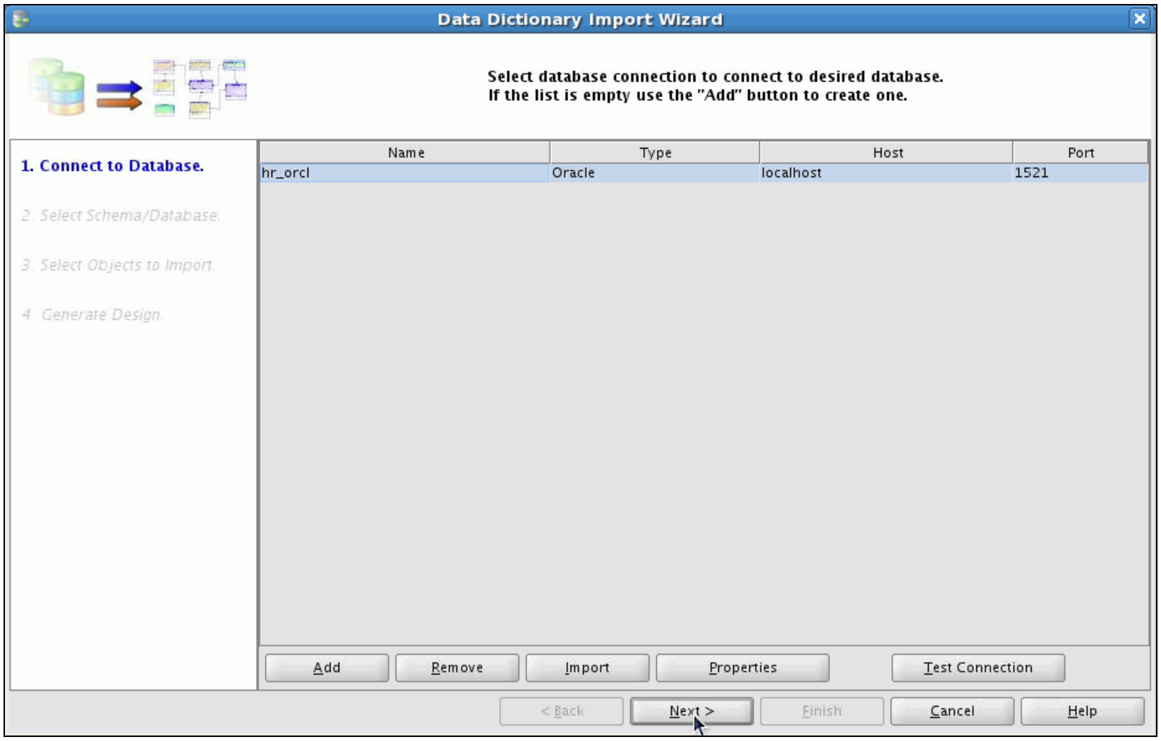
COUNTRY_C_ID_PK

40) You want to compare this model with what is in the database. To do that, you import from the data dictionary. Select File > Import > Data Dictionary.



41) Select the hr_orcl connection that you created earlier in the practice, and click Next.

Solution 20-1: Re-Engineer the HR Schema (continued)



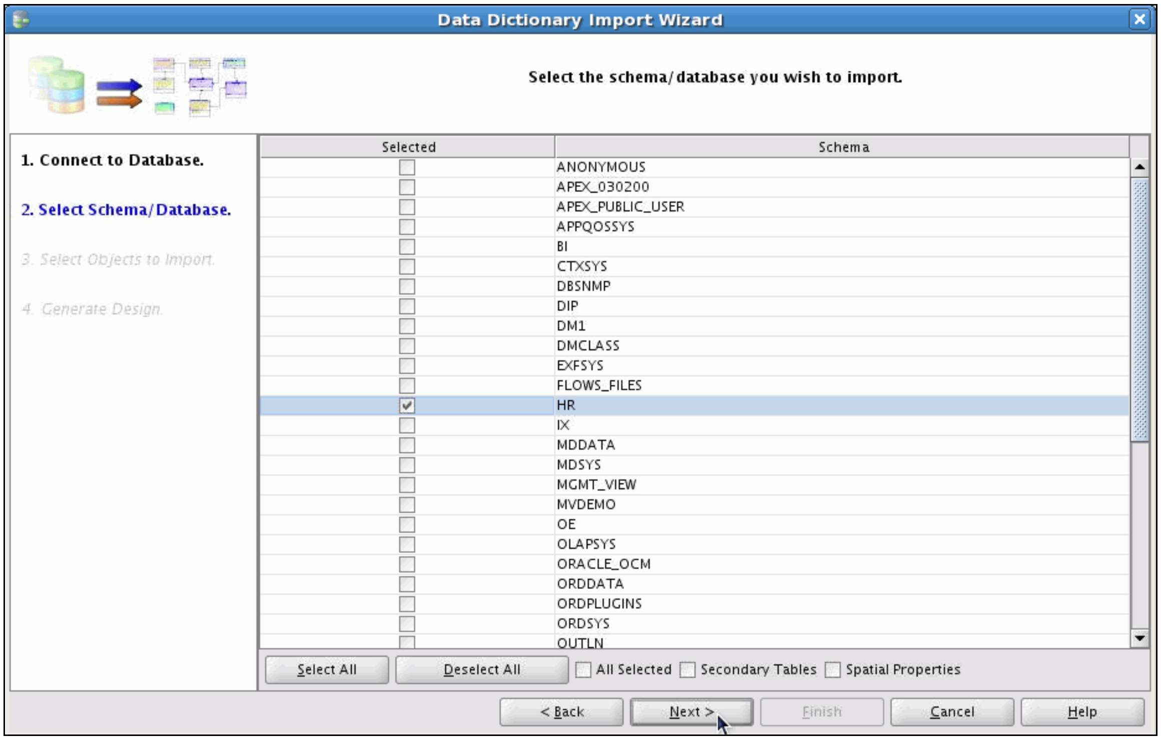
Data Dictionary Import Wizard

Select database connection to connect to desired database.
If the list is empty use the "Add" button to create one.

1. Connect to Database.
2. Select Schema/Database.
3. Select Objects to Import.
4. Generate Design.

Name	Type	Host	Port
hr_orcl	Oracle	localhost	1521

42) Select the HR schema, and click Next.



Data Dictionary Import Wizard

Select the schema/database you wish to import.

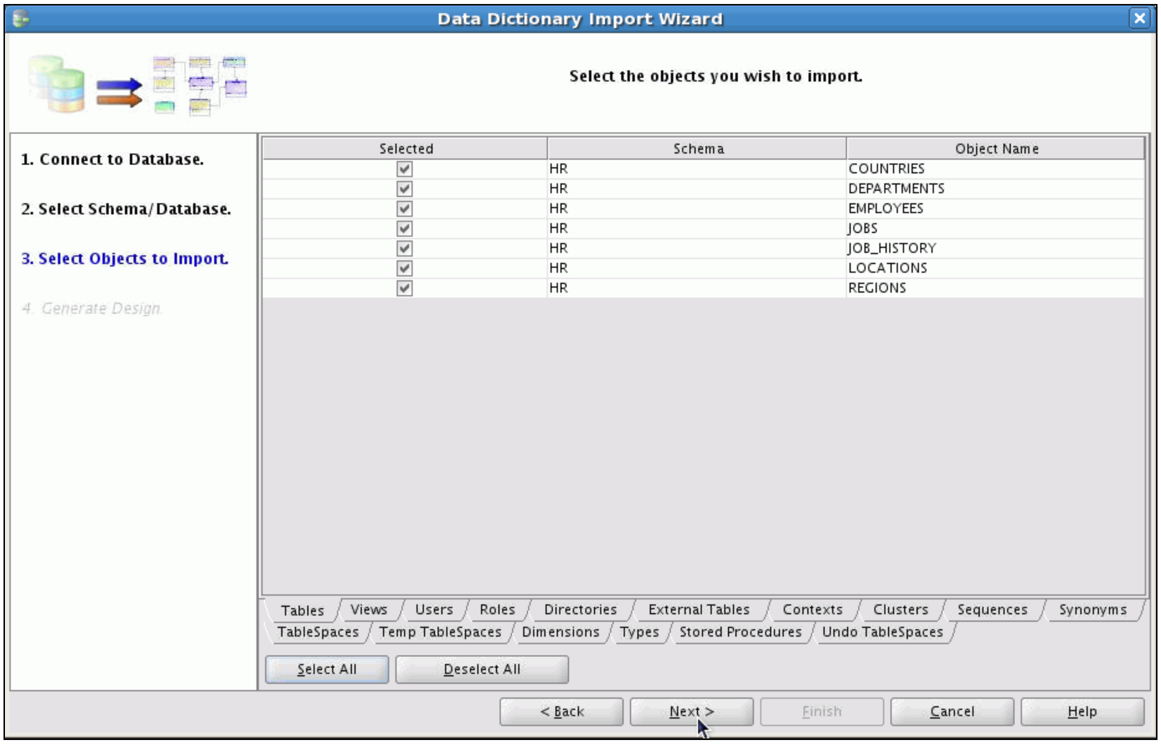
1. Connect to Database.
2. Select Schema/Database.
3. Select Objects to Import.
4. Generate Design.

Selected	Schema
□	ANONYMOUS
□	APEX_030200
□	APEX_PUBLIC_USER
□	APPQOSSYS
□	BI
□	CTXSYS
□	DBSNMP
□	DIP
□	DM1
□	DMCLASS
□	EXFSYS
□	FLOWS_FILES
☑	HR
□	IX
□	MDDATA
□	MDSYS
□	MGMT_VIEW
□	MVDEMO
□	OE
□	OLAPSYS
□	ORACLE_OCM
□	ORDDATA
□	ORDPLUGINS
□	ORDSYS
□	OUTLN

□ All Selected □ Secondary Tables □ Spatial Properties

43) Click Select All to select all the tables, and click Next.

Solution 20-1: Re-Engineer the HR Schema (continued)

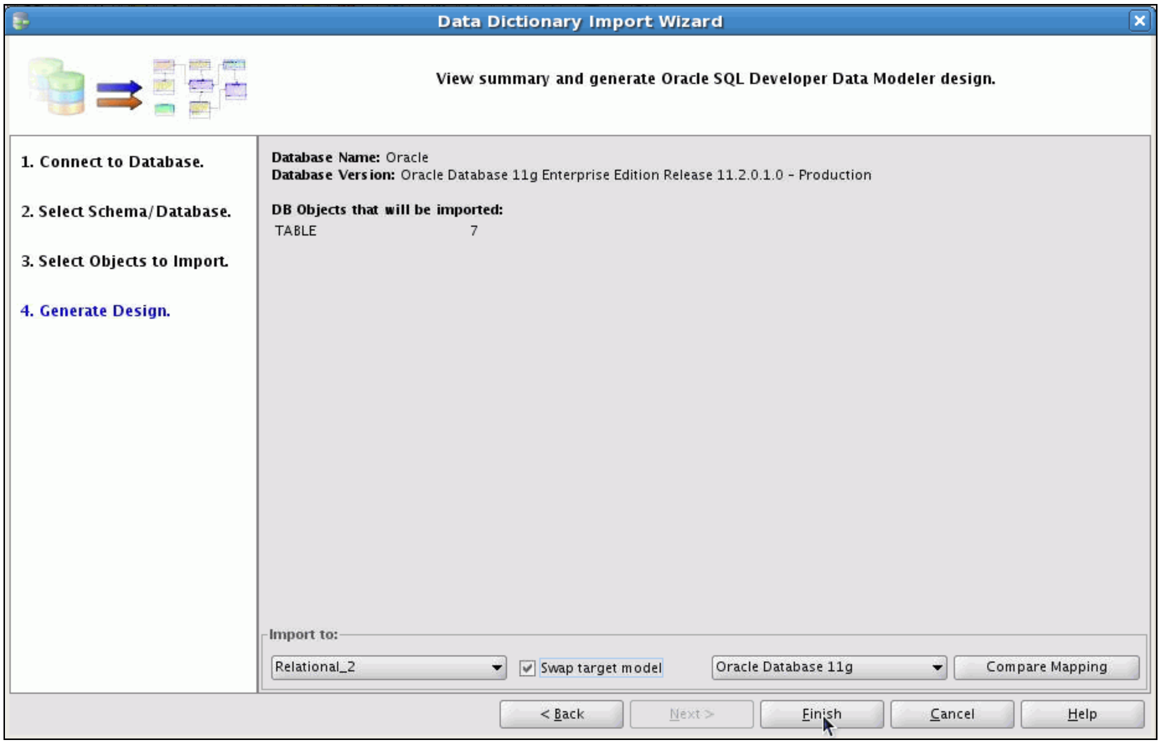


Data Dictionary Import Wizard
Select the objects you wish to import.
1. Connect to Database.
2. Select Schema/Database.
3. Select Objects to Import.
4. Generate Design.

Selected	Schema	Object Name
☑	HR	COUNTRIES
☑	HR	DEPARTMENTS
☑	HR	EMPLOYEES
☑	HR	JOBS
☑	HR	JOB_HISTORY
☑	HR	LOCATIONS
☑	HR	REGIONS

44) *This step is important:* In order to compare the relational model against the database, select Relational_2 for “Import to” and click the “Swap target model” check box. Then click Finish.

Solution 20-1: Re-Engineer the HR Schema (continued)


Data Dictionary Import Wizard
View summary and generate Oracle SQL Developer Data Modeler design.
1. Connect to Database.
2. Select Schema/Database.
3. Select Objects to Import.
4. Generate Design.
Import to:
< Back
Next >
Finish
Cancel
Help
Compare Mapping

Database Name: Oracle
Database Version: Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production

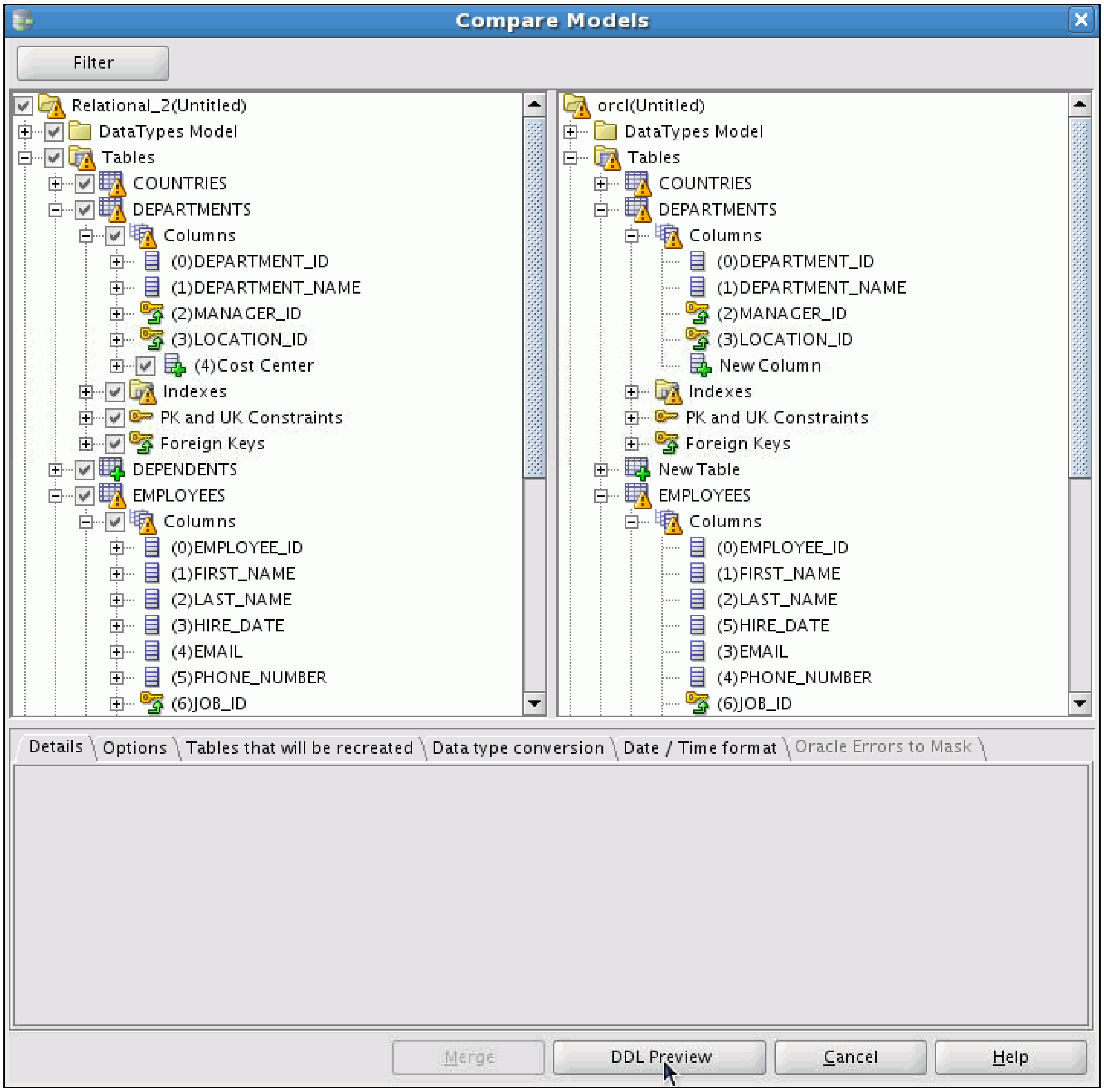
DB Objects that will be imported:

TABLE 7

Import to: Relational_2 ☑ Swap target model Oracle Database 11g

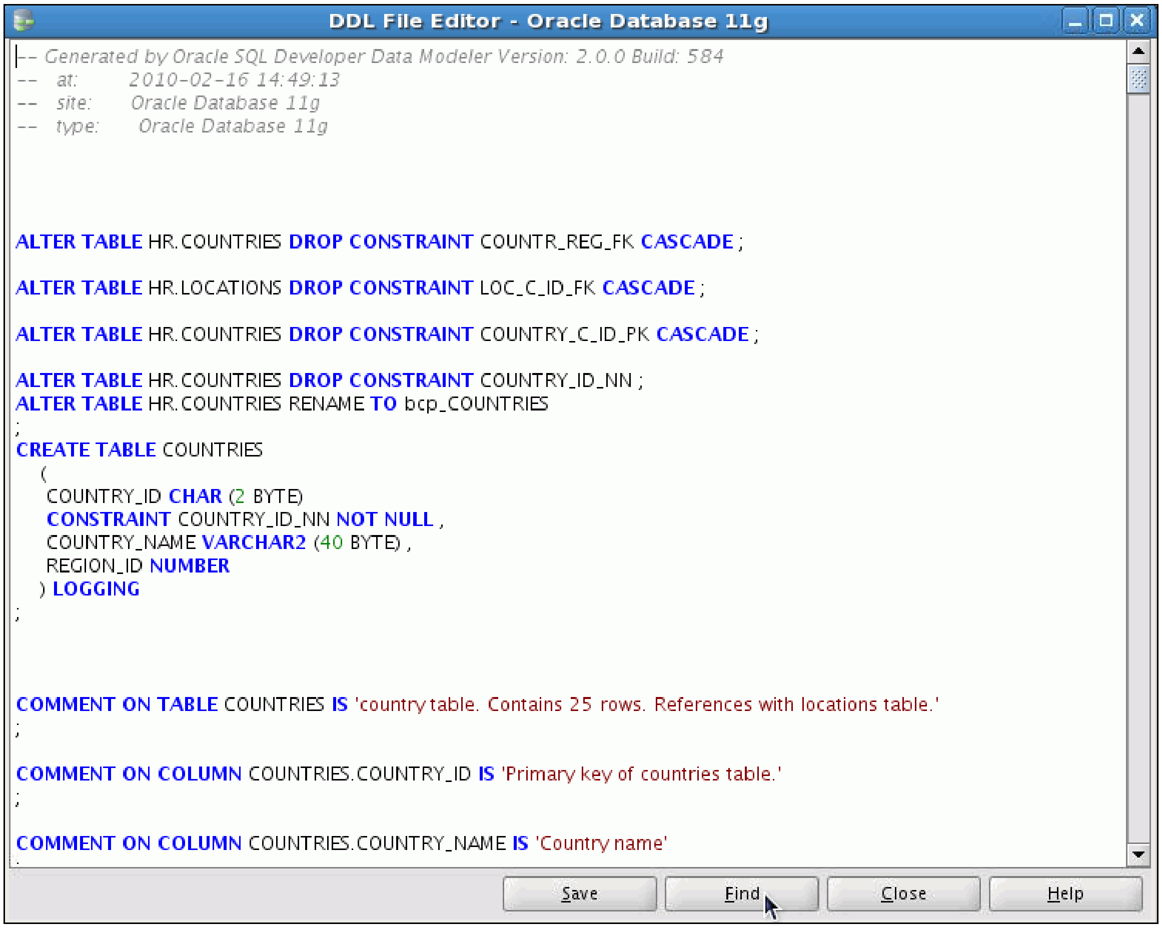
45) The Compare Models window appears. Notice that you are comparing Relational_2 on the left with orcl, the database on the right. Expand some of the nodes to see what will happen if you generated the DDL. Notice that a new column is added to the DEPARTMENTS table, the new DEPENDENTS table is added and the columns in the EMPLOYEES table are reordered. To see the DDL, click DDL Preview.

Solution 20-1: Re-Engineer the HR Schema (continued)



46) Notice that the DDL contains Alter statements instead of Create statements. Click Find.

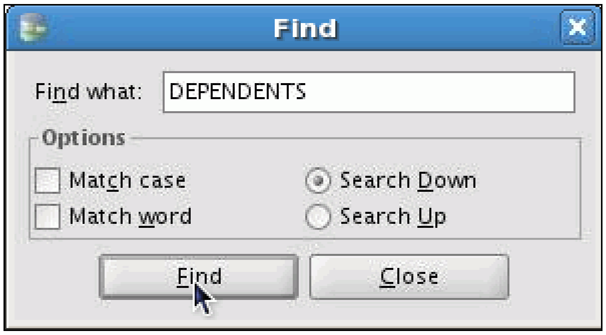
Solution 20-1: Re-Engineer the HR Schema (continued)



DDL File Editor - Oracle Database 11g

```
-- Generated by Oracle SQL Developer Data Modeler Version: 2.0.0 Build: 584
--   at:        2010-02-16 14:49:13
--   site:      Oracle Database 11g
--   type:      Oracle Database 11g



ALTER TABLE HR.COUNTRIES DROP CONSTRAINT COUNTR_REG_FK CASCADE ;

ALTER TABLE HR.LOCATIONS DROP CONSTRAINT LOC_C_ID_FK CASCADE ;

ALTER TABLE HR.COUNTRIES DROP CONSTRAINT COUNTRY_C_ID_PK CASCADE ;

ALTER TABLE HR.COUNTRIES DROP CONSTRAINT COUNTRY_ID_NN ;
ALTER TABLE HR.COUNTRIES RENAME TO bcp_COUNTRIES
;
CREATE TABLE COUNTRIES
    (
     COUNTRY_ID CHAR (2 BYTE)
     CONSTRAINT COUNTRY_ID_NN NOT NULL ,
     COUNTRY_NAME VARCHAR2 (40 BYTE) ,
     REGION_ID NUMBER
    ) LOGGING
;



COMMENT ON TABLE COUNTRIES IS 'country table. Contains 25 rows. References with locations table.'
;

COMMENT ON COLUMN COUNTRIES.COUNTRY_ID IS 'Primary key of countries table.'
;

COMMENT ON COLUMN COUNTRIES.COUNTRY_NAME IS 'Country name'
```

47) Enter DEPENDENTS in the Find What field, and click Find.



Find

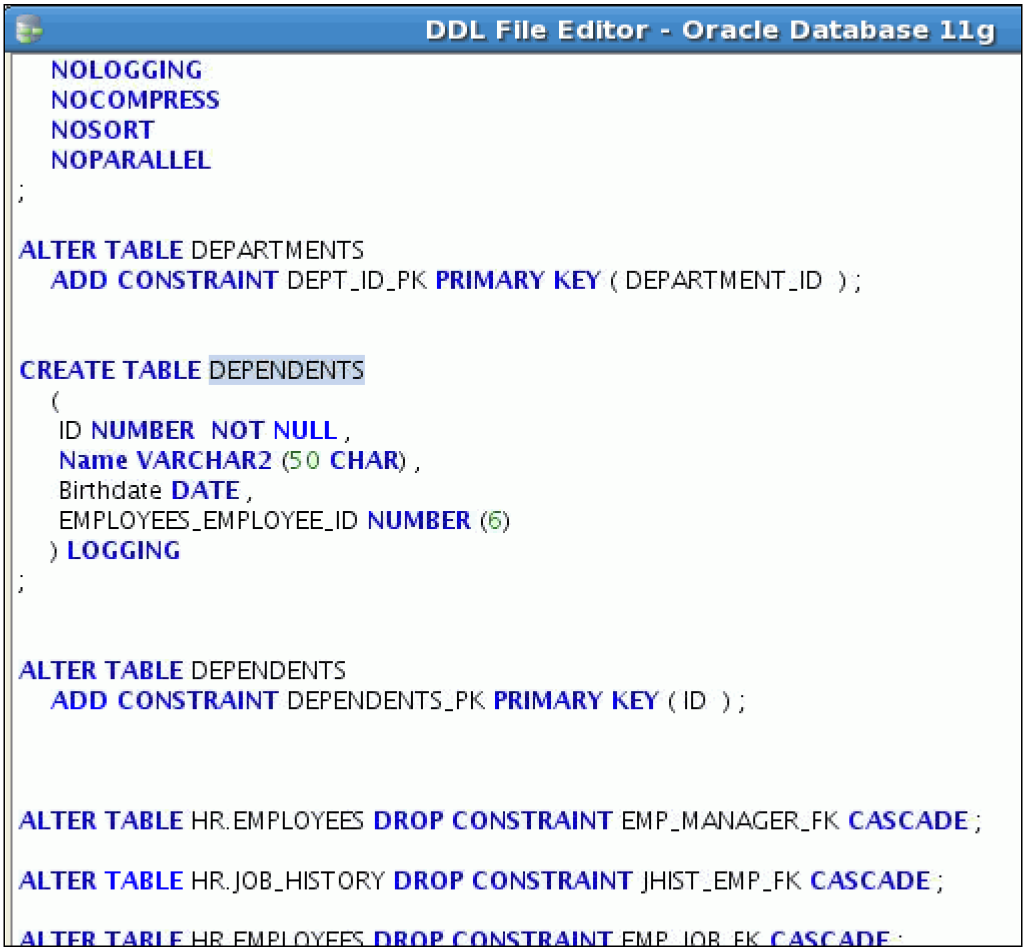
Find what: DEPENDENTS

Options

☐ Match case ● Search Down
☐ Match word ○ Search Up

48) The DDL for the new table is displayed. When done reviewing, click Close.

Solution 20-1: Re-Engineer the HR Schema (continued)

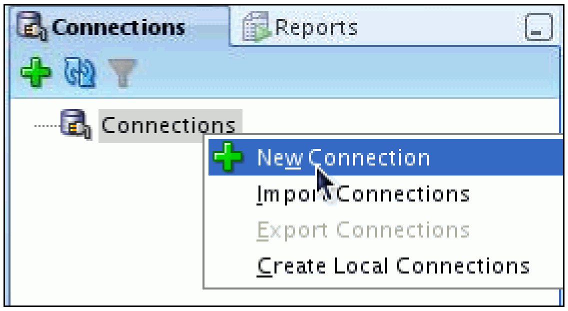


```
   NOLOGGING
   NOCOMPRESS
   NOSORT
   NOPARALLEL
;

ALTER TABLE DEPARTMENTS
   ADD CONSTRAINT DEPT_ID_PK PRIMARY KEY ( DEPARTMENT_ID ) ;


CREATE TABLE DEPENDENTS
   (
    ID NUMBER  NOT NULL ,
    Name VARCHAR2 (50 CHAR) ,
    Birthdate DATE ,
    EMPLOYEES_EMPLOYEE_ID NUMBER (6)
   ) LOGGING
;


ALTER TABLE DEPENDENTS
   ADD CONSTRAINT DEPENDENTS_PK PRIMARY KEY ( ID ) ;



ALTER TABLE HR.EMPLOYEES DROP CONSTRAINT EMP_MANAGER_FK CASCADE ;

ALTER TABLE HR.JOB_HISTORY DROP CONSTRAINT JHIST_EMP_FK CASCADE ;

ALTER TABLE HR.EMPLOYEES DROP CONSTRAINT EMP_JOB_FK CASCADE ;
```

49) Click Cancel to cancel the comparison.

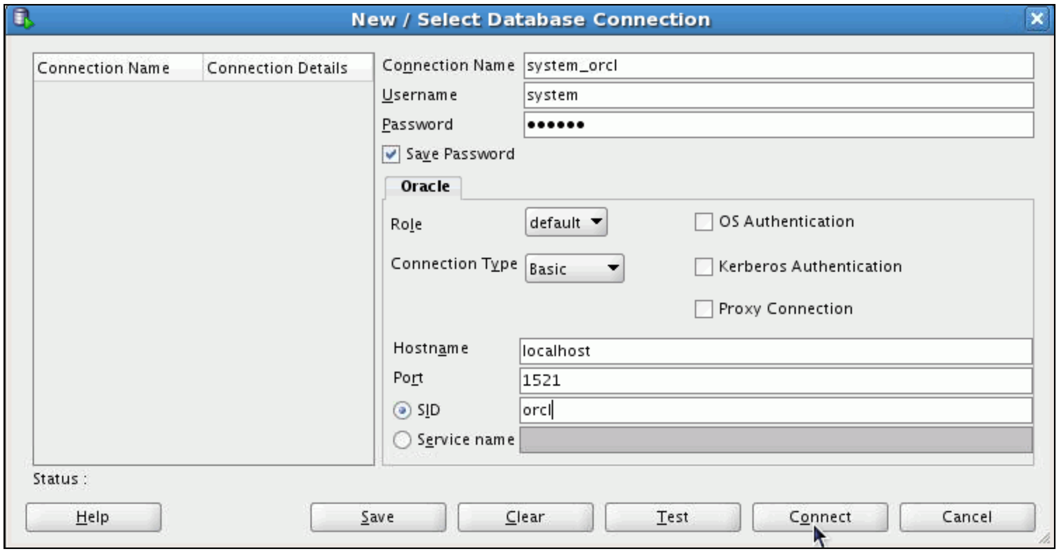
50) At this point, you want to run some data modeler reports in Oracle SQL Developer. In order to run the reports, create a user that will store the metadata. Open Oracle SQL Developer.

51) Right-click Connections, and select New Connection.



52) To create a new user, create a connection as `system`. Enter `system_orcl` for Connection Name, enter `system` for Username and `oracle` for Password, enter `orcl` for SID, and click Connect.

Solution 20-1: Re-Engineer the HR Schema (continued)



Connection Name: system_orcl

Username: system

Password: ••••••

☑ Save Password

Role: default

Connection Type: Basic

□ OS Authentication

□ Kerberos Authentication

□ Proxy Connection

Hostname: localhost

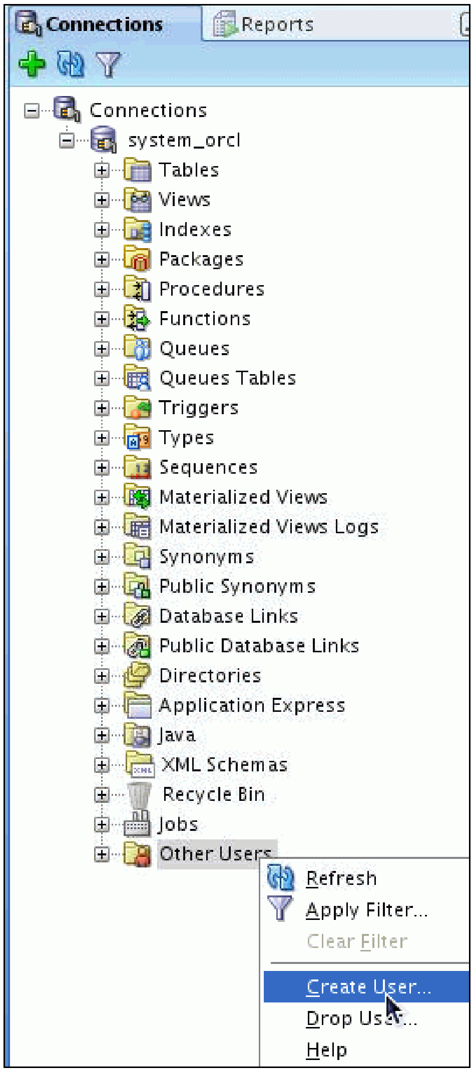
Port: 1521

● SID: orcl

○ Service name:

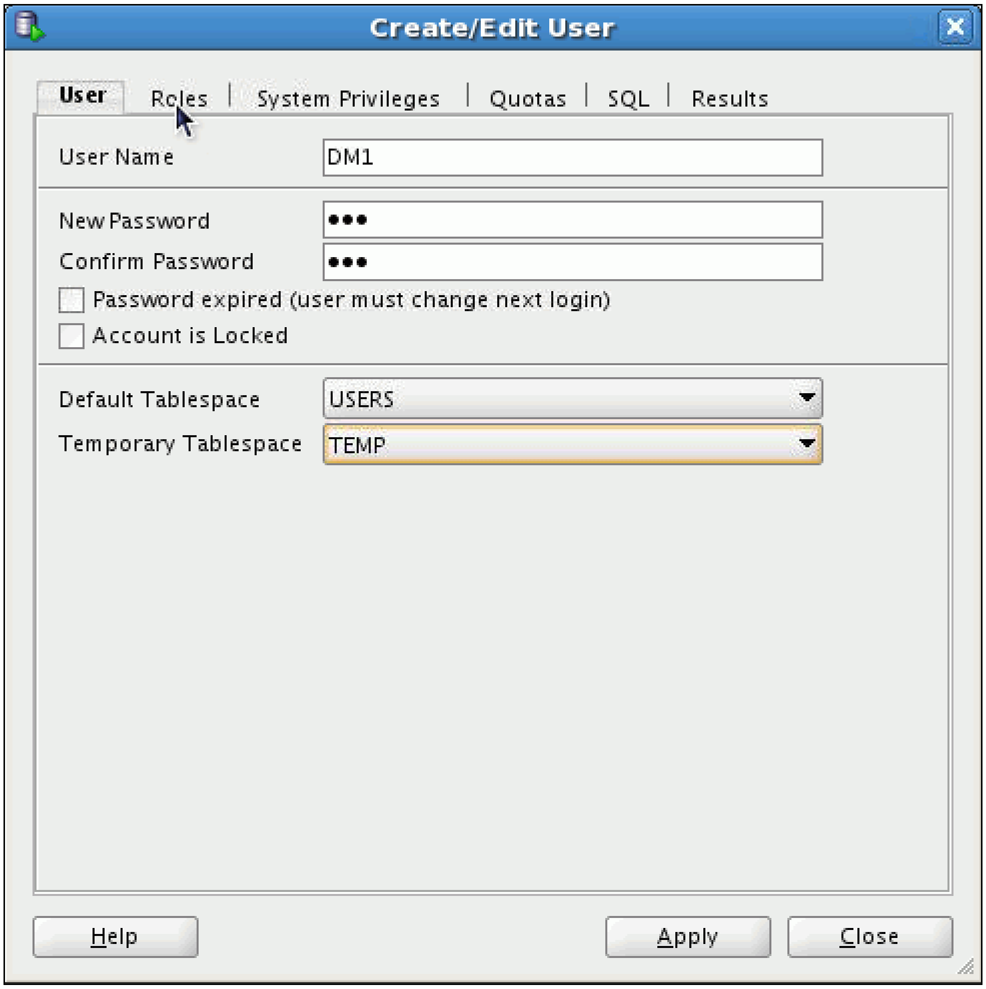
53) Expand system_orcl, right-click Other Users, and select Create User.

Solution 20-1: Re-Engineer the HR Schema (continued)



54) Enter `DM1` for User Name and `dm1` for New Password and Confirm Password. Select USERS for Default Tablespace and TEMP for Temporary Tablespace, and click the Roles tab.

Solution 20-1: Re-Engineer the HR Schema (continued)



User Name: DM1

New Password: ●●●

Confirm Password: ●●●

□ Password expired (user must change next login)

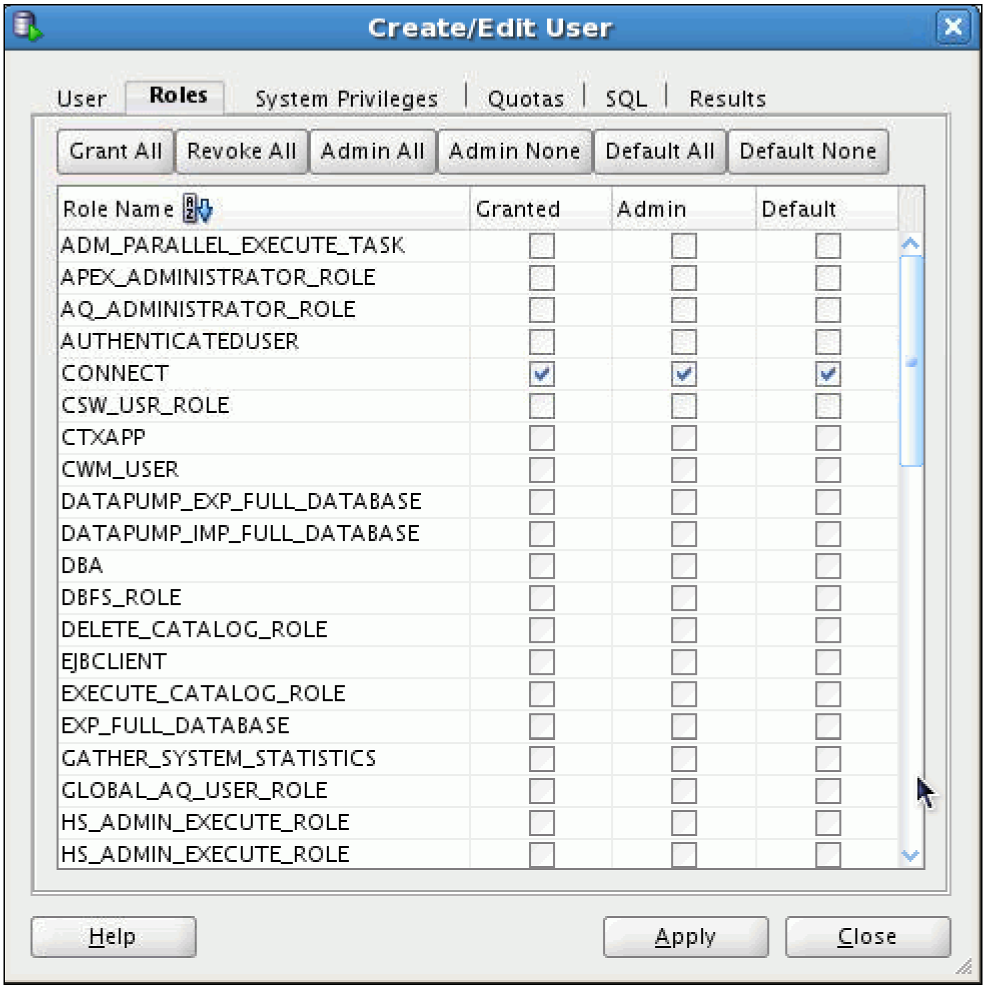
□ Account is Locked

Default Tablespace: USERS

Temporary Tablespace: TEMP

55) Select all check boxes for the CONNECT role, and scroll down the list to the Resource role.

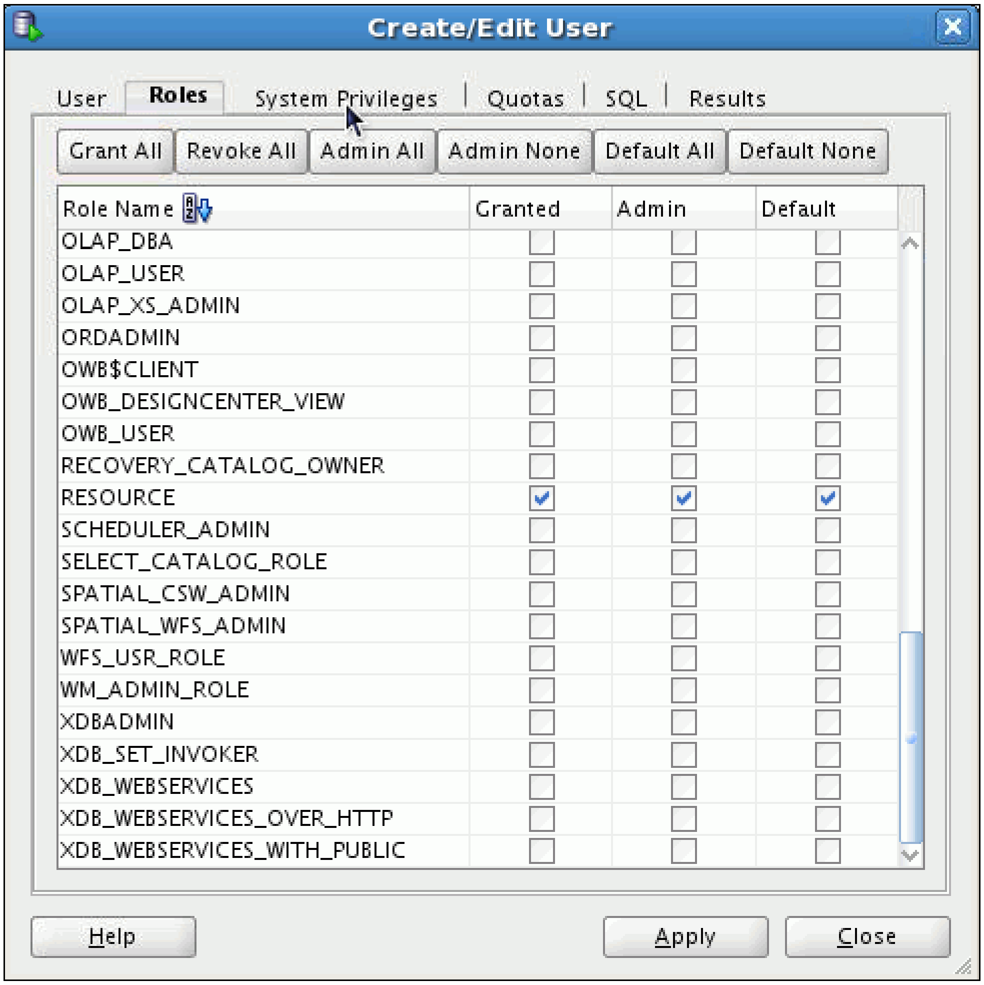
Solution 20-1: Re-Engineer the HR Schema (continued)



Role Name	Granted	Admin	Default
ADM_PARALLEL_EXECUTE_TASK	☐	☐	☐
APEX_ADMINISTRATOR_ROLE	☐	☐	☐
AQ_ADMINISTRATOR_ROLE	☐	☐	☐
AUTHENTICATEDUSER	☐	☐	☐
CONNECT	☑	☑	☑
CSW_USR_ROLE	☐	☐	☐
CTXAPP	☐	☐	☐
CWM_USER	☐	☐	☐
DATAPUMP_EXP_FULL_DATABASE	☐	☐	☐
DATAPUMP_IMP_FULL_DATABASE	☐	☐	☐
DBA	☐	☐	☐
DBFS_ROLE	☐	☐	☐
DELETE_CATALOG_ROLE	☐	☐	☐
EJBCLIENT	☐	☐	☐
EXECUTE_CATALOG_ROLE	☐	☐	☐
EXP_FULL_DATABASE	☐	☐	☐
GATHER_SYSTEM_STATISTICS	☐	☐	☐
GLOBAL_AQ_USER_ROLE	☐	☐	☐
HS_ADMIN_EXECUTE_ROLE	☐	☐	☐
HS_ADMIN_EXECUTE_ROLE	☐	☐	☐

56) Select all check boxes for the RESOURCE role, and click the System Privileges tab.

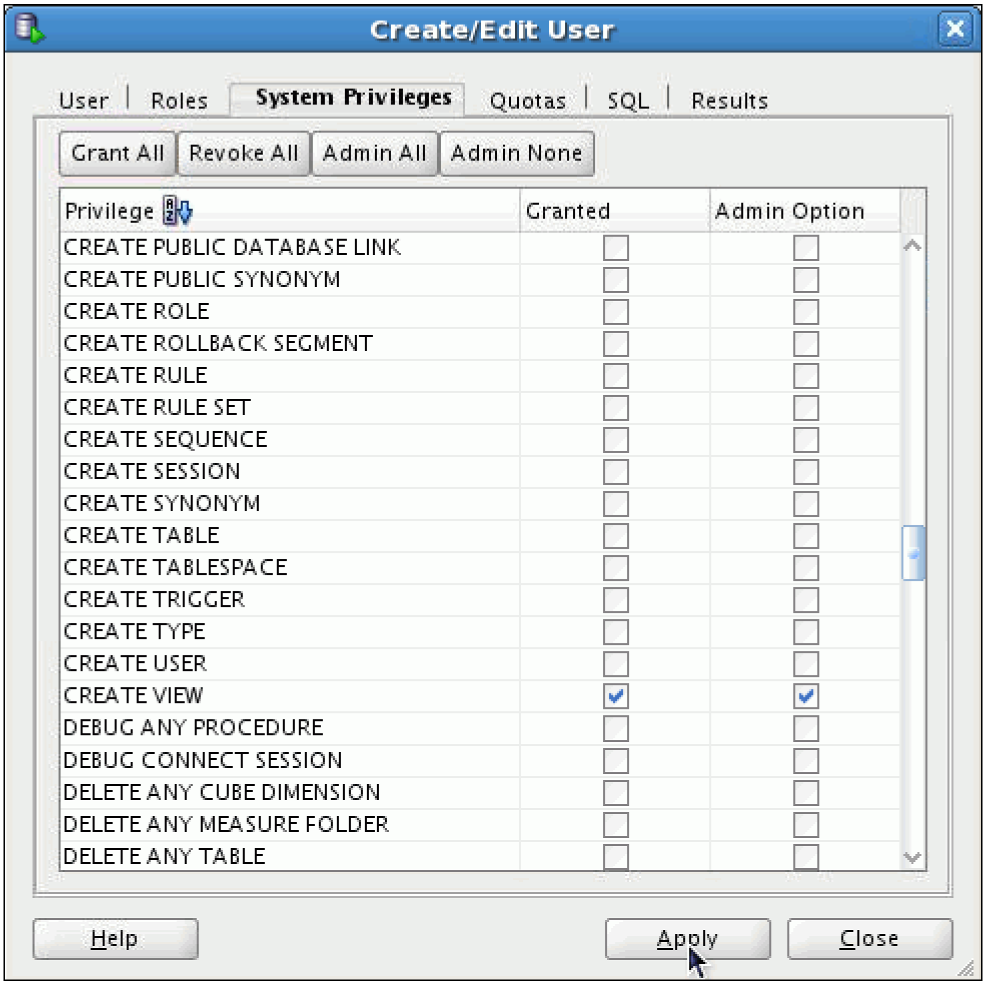
Solution 20-1: Re-Engineer the HR Schema (continued)



Role Name	Granted	Admin	Default
OLAP_DBA	☐	☐	☐
OLAP_USER	☐	☐	☐
OLAP_XS_ADMIN	☐	☐	☐
ORDADMIN	☐	☐	☐
OWB$CLIENT	☐	☐	☐
OWB_DESIGNCENTER_VIEW	☐	☐	☐
OWB_USER	☐	☐	☐
RECOVERY_CATALOG_OWNER	☐	☐	☐
RESOURCE	☑	☑	☑
SCHEDULER_ADMIN	☐	☐	☐
SELECT_CATALOG_ROLE	☐	☐	☐
SPATIAL_CSW_ADMIN	☐	☐	☐
SPATIAL_WFS_ADMIN	☐	☐	☐
WFS_USR_ROLE	☐	☐	☐
WM_ADMIN_ROLE	☐	☐	☐
XDBADMIN	☐	☐	☐
XDB_SET_INVOKER	☐	☐	☐
XDB_WEBSERVICES	☐	☐	☐
XDB_WEBSERVICES_OVER_HTTP	☐	☐	☐
XDB_WEBSERVICES_WITH_PUBLIC	☐	☐	☐

57) Scroll down to the CREATE VIEW privilege and select all check boxes for that privilege. Click Apply.

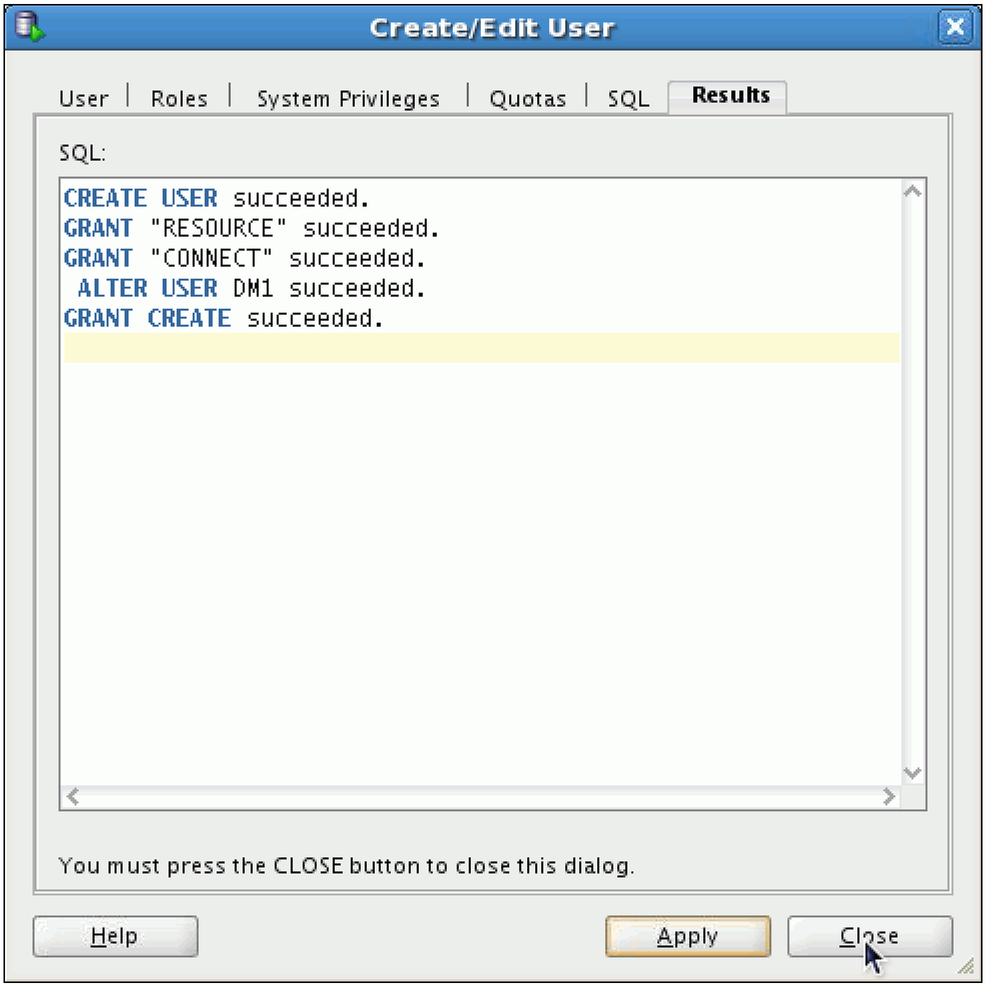
Solution 20-1: Re-Engineer the HR Schema (continued)



Privilege	Granted	Admin Option
CREATE PUBLIC DATABASE LINK	☐	☐
CREATE PUBLIC SYNONYM	☐	☐
CREATE ROLE	☐	☐
CREATE ROLLBACK SEGMENT	☐	☐
CREATE RULE	☐	☐
CREATE RULE SET	☐	☐
CREATE SEQUENCE	☐	☐
CREATE SESSION	☐	☐
CREATE SYNONYM	☐	☐
CREATE TABLE	☐	☐
CREATE TABLESPACE	☐	☐
CREATE TRIGGER	☐	☐
CREATE TYPE	☐	☐
CREATE USER	☐	☐
CREATE VIEW	☑	☑
DEBUG ANY PROCEDURE	☐	☐
DEBUG CONNECT SESSION	☐	☐
DELETE ANY CUBE DIMENSION	☐	☐
DELETE ANY MEASURE FOLDER	☐	☐
DELETE ANY TABLE	☐	☐

58) The DM1 user with the correct privileges was created successfully. Click Close.

Solution 20-1: Re-Engineer the HR Schema (continued)

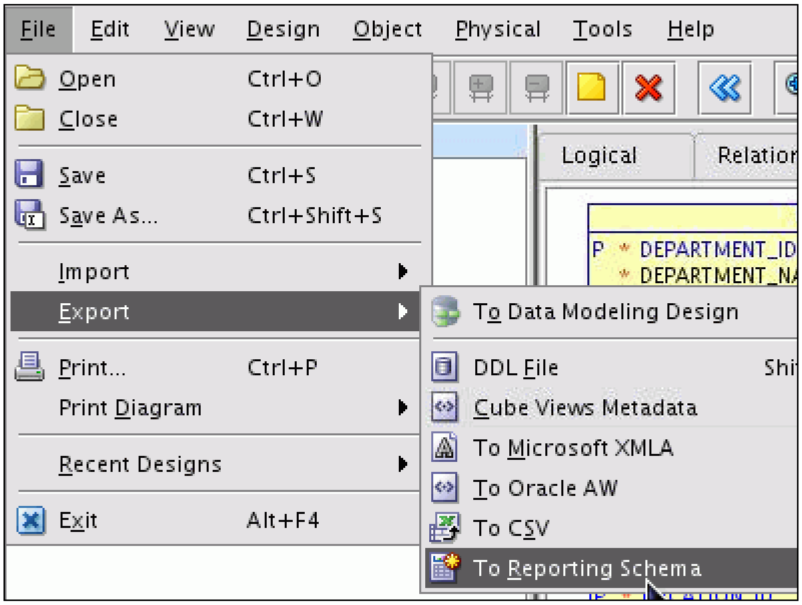


```
CREATE USER succeeded.
GRANT "RESOURCE" succeeded.
GRANT "CONNECT" succeeded.
 ALTER USER DM1 succeeded.
GRANT CREATE succeeded.
```

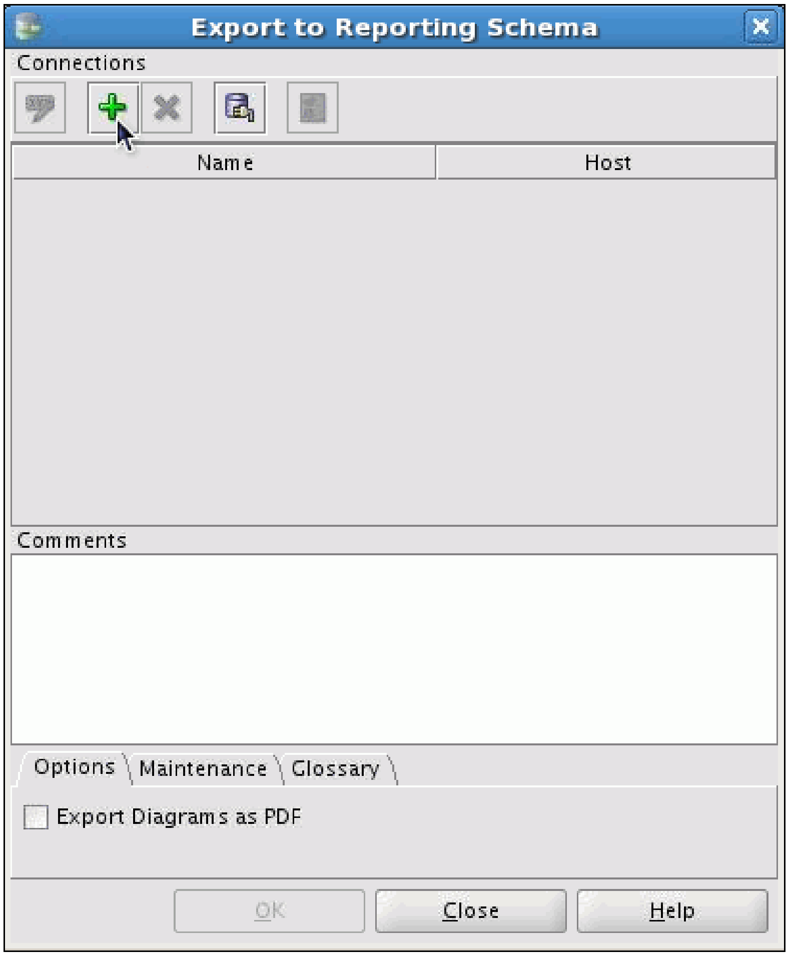
You must press the CLOSE button to close this dialog.

59) You can now switch to Oracle SQL Developer Data Modeler and export your model to the reporting schema. Select File > Export > To Reporting Schema.

Solution 20-1: Re-Engineer the HR Schema (continued)

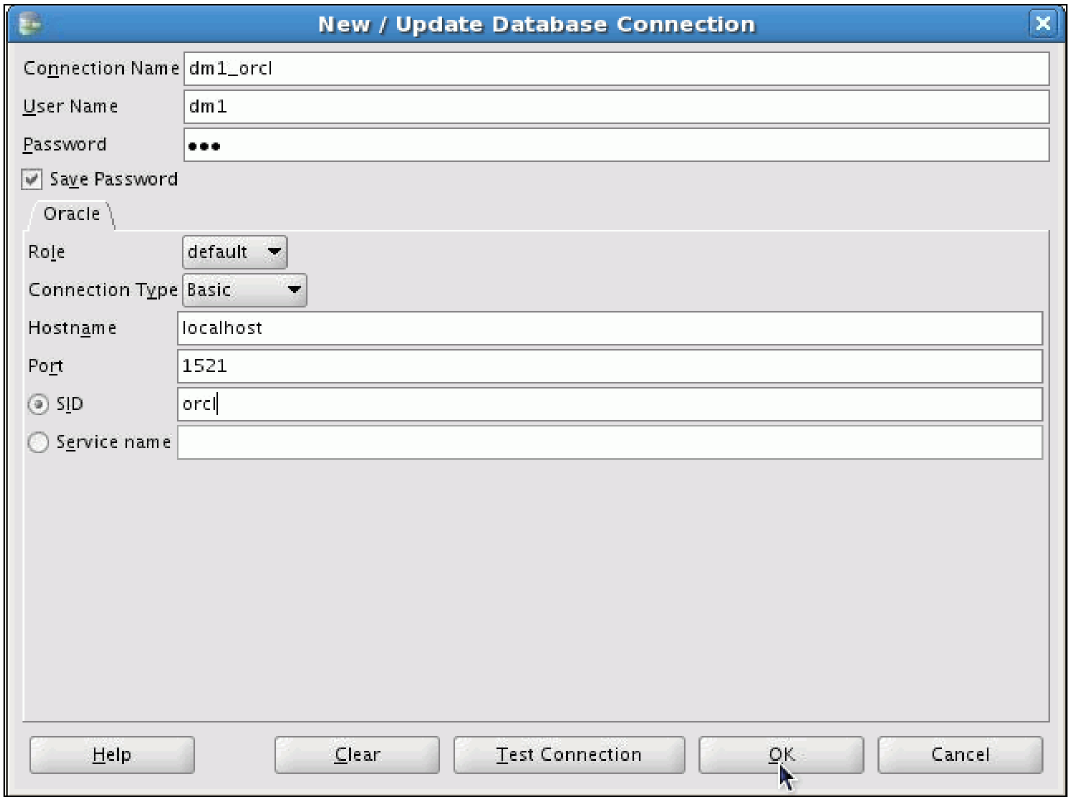


60) Click the Add ➕ icon.



61) Enter `dm1_orcl` for Connection Name, `dm1` for User Name and Password, and `orcl` for SID. Then click OK.

Solution 20-1: Re-Engineer the HR Schema (continued)



Connection Name dm1_orcl

User Name dm1

Password •••

☑ Save Password

Role default

Connection Type Basic

Hostname localhost

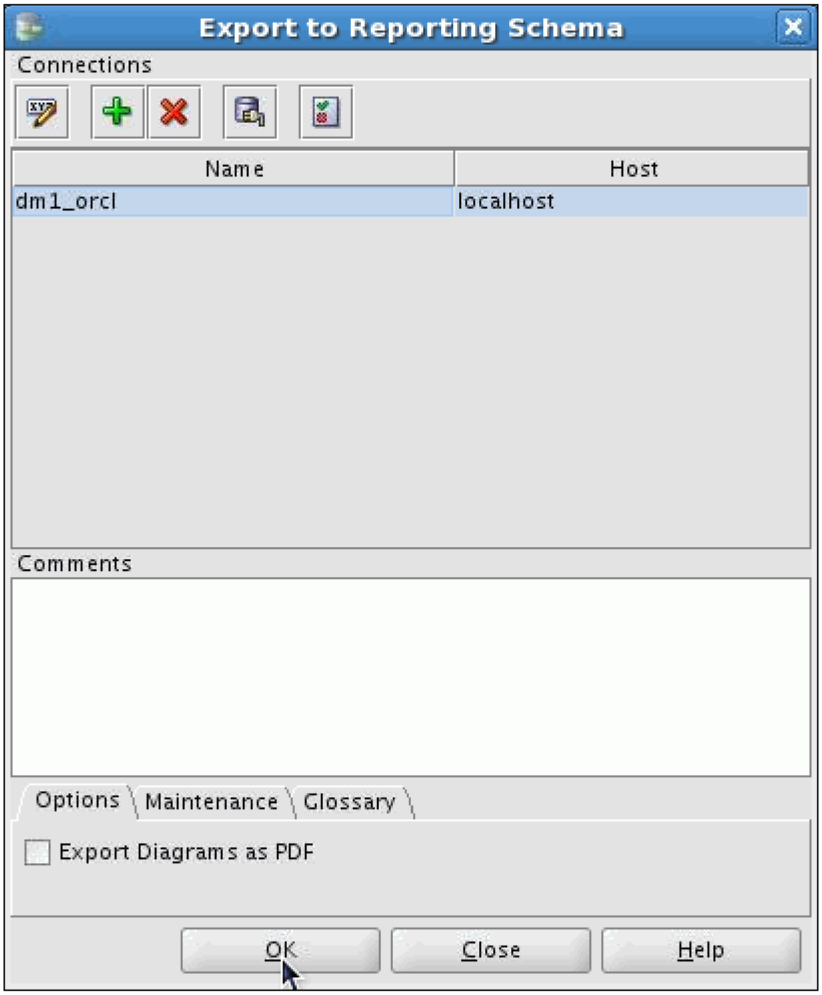
Port 1521

● SID orcl

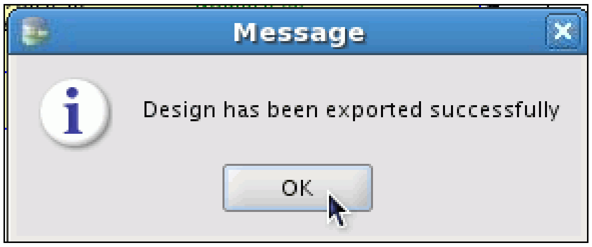
○ Service name

62) Select the dm1_orcl connection from the list, and click OK.

Solution 20-1: Re-Engineer the HR Schema (continued)



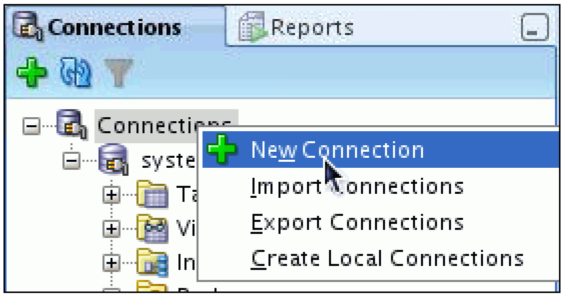
63) Your design has been exported successfully. Note that you may encounter the Message dialog box behind another window. Drag the Message header to another area in your window, and click OK.



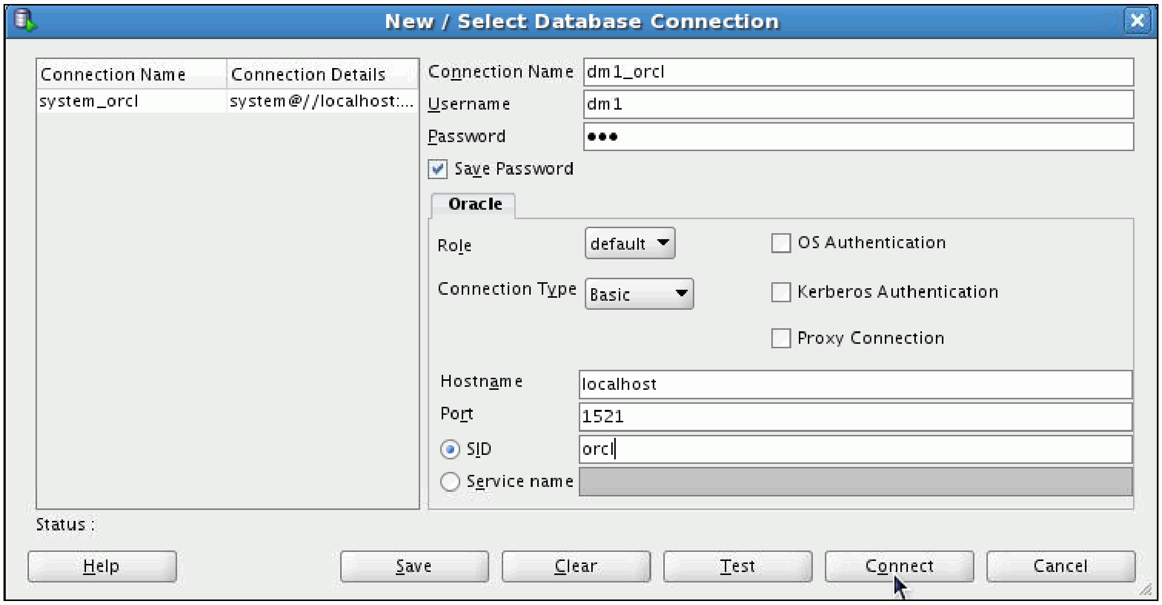
Design has been exported successfully

64) Now you can switch to Oracle SQL Developer and review the reports. You first need to create a connection to the DM1 user. Right-click Connections, and select New Connection.

Solution 20-1: Re-Engineer the HR Schema (continued)



65) Enter `dm1_orcl` for Connection Name, `dm1` for Username and Password, and `orcl` for SID, and then click Connect.


New / Select Database Connection

Connection Name	Connection Details
system_orcl	system@//localhost:...

Connection Name: dm1_orcl

Username: dm1

Password: •••

☑ Save Password

Oracle

Role: default

Connection Type: Basic

□ OS Authentication

□ Kerberos Authentication

□ Proxy Connection

Hostname: localhost

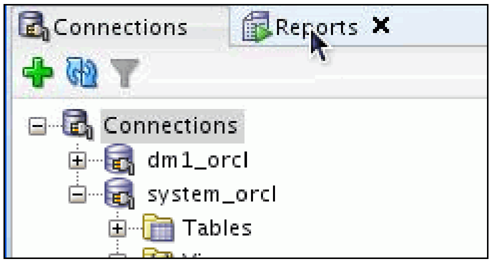
Port: 1521

● SID: orcl

○ Service name

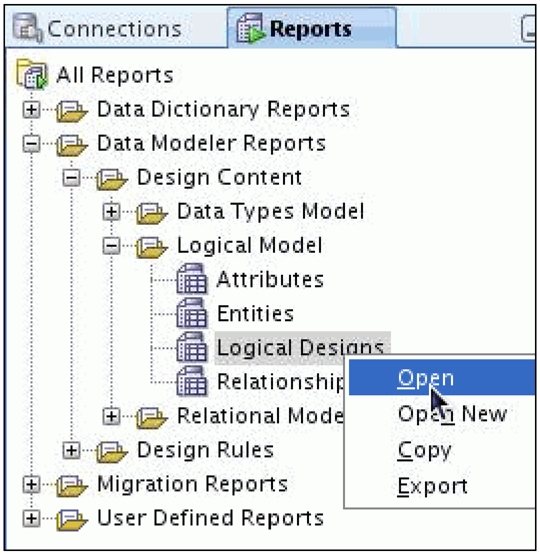
Status :

66) Click the Reports tab.

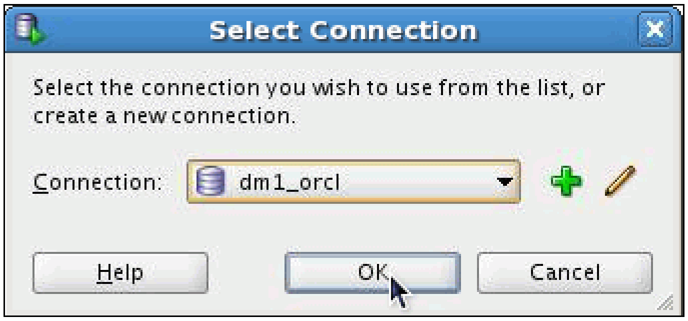


67) Expand Data Modeler Reports > Design Content > Logical Model, right-click Logical Designs, and select Open.

Solution 20-1: Re-Engineer the HR Schema (continued)



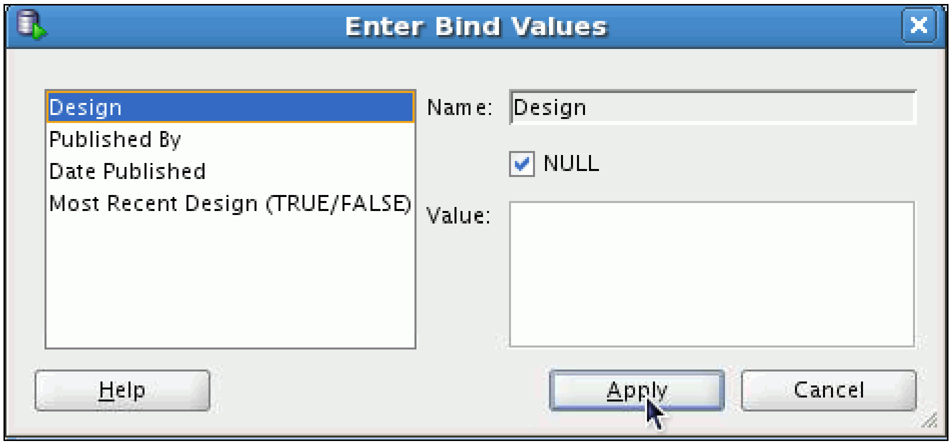
68) Make sure that the dm1_orcl connection is selected, and click OK.



Select the connection you wish to use from the list, or create a new connection.

Connection: dm1_orcl

69) You can specify some bind variables to narrow down what the report displays. In this case, click Apply.



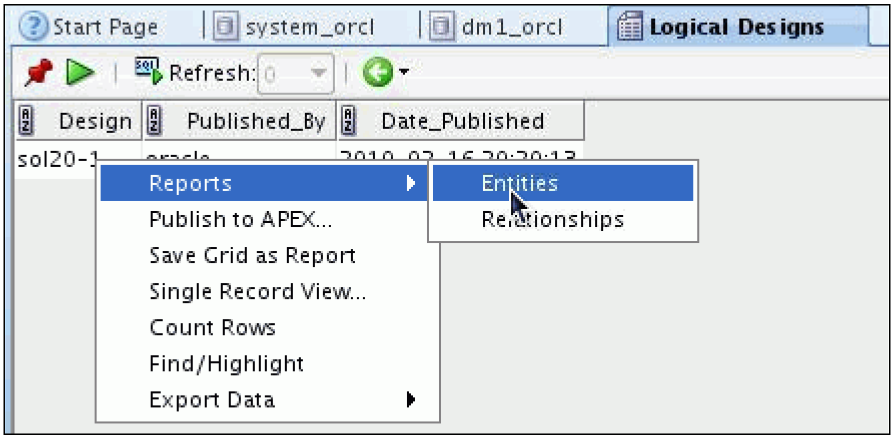
Name: Design

☑ NULL

Value:

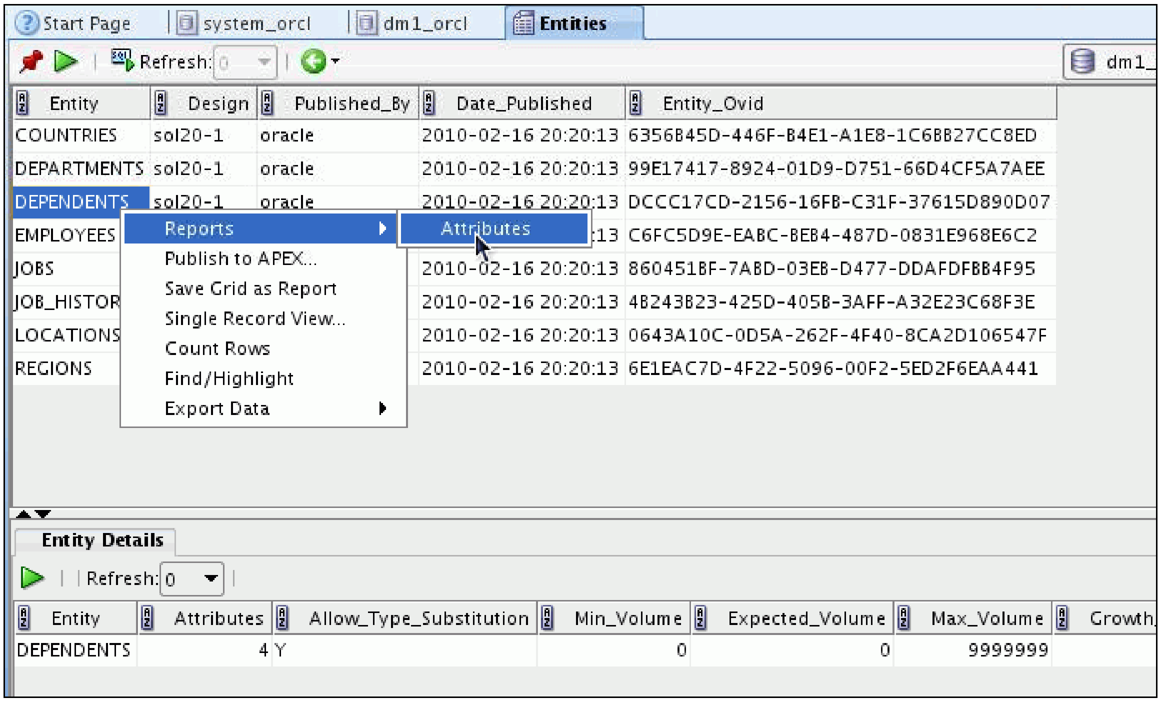
70) A list of your designs is displayed. Right-click your design name, and select Reports > Entities.

Solution 20-1: Re-Engineer the HR Schema (continued)



Design	Published_By	Date_Published
sol20-1	oracle	2010-02-16 20:20:13

71) A list of the entities is displayed. Select the DEPENDENTS entity to see the details. To drill into the entity to see the list of attributes, right-click DEPENDENTS, and select Reports and then Attributes.

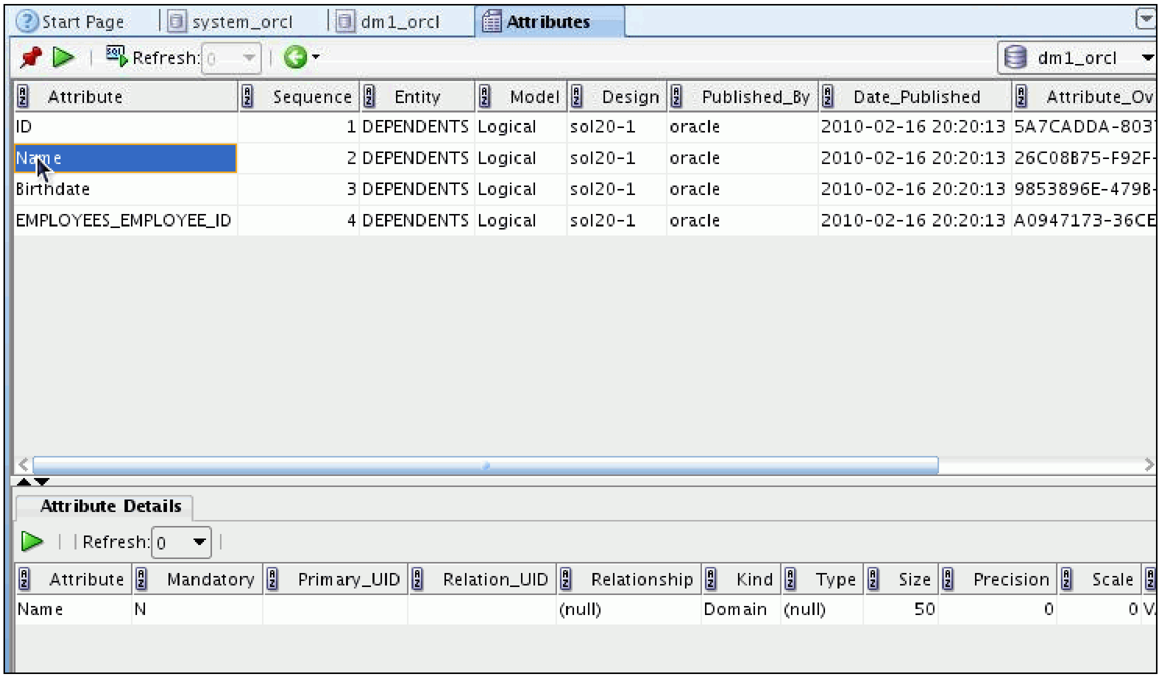


Entity	Design	Published_By	Date_Published	Entity_Ovid
COUNTRIES	sol20-1	oracle	2010-02-16 20:20:13	6356B45D-446F-B4E1-A1E8-1C6BB27CC8ED
DEPARTMENTS	sol20-1	oracle	2010-02-16 20:20:13	99E17417-8924-01D9-D751-66D4CF5A7AEE
DEPENDENTS	sol20-1	oracle	2010-02-16 20:20:13	DCCC17CD-2156-16FB-C31F-37615D890D07
EMPLOYEES			:13	C6FC5D9E-EABC-BEB4-487D-0831E968E6C2
JOBS			2010-02-16 20:20:13	860451BF-7ABD-03EB-D477-DDAFDFBB4F95
JOB_HISTOR			2010-02-16 20:20:13	4B243B23-425D-405B-3AFF-A32E23C68F3E
LOCATIONS			2010-02-16 20:20:13	0643A10C-0D5A-262F-4F40-8CA2D106547F
REGIONS			2010-02-16 20:20:13	6E1EAC7D-4F22-5096-00F2-5ED2F6EAA441

Entity	Attributes	Allow_Type_Substitution	Min_Volume	Expected_Volume	Max_Volume	Growth
DEPENDENTS	4	Y	0	0	9999999	

72) The list of attributes is displayed. Select the Name attribute to see the details of that attribute.

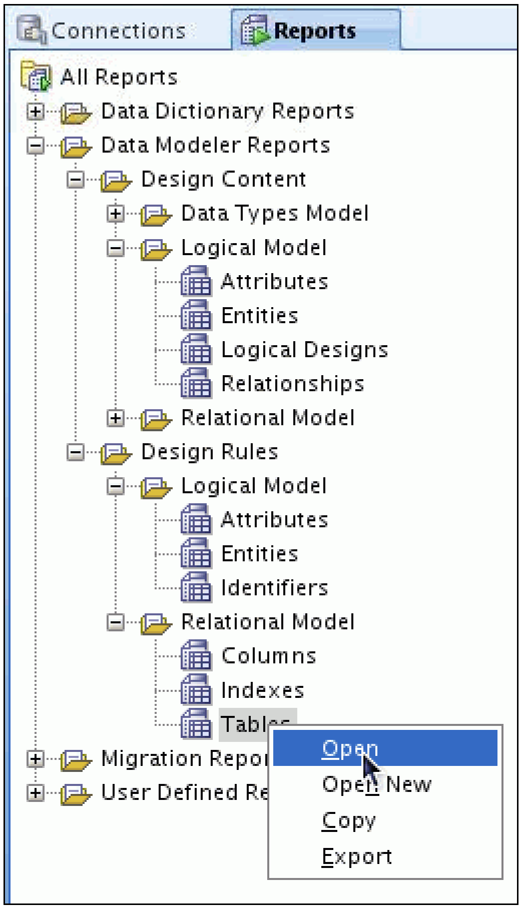
Solution 20-1: Re-Engineer the HR Schema (continued)



Attribute	Sequence	Entity	Model	Design	Published_By	Date_Published	Attribute_Ov
ID	1	DEPENDENTS	Logical	sol20-1	oracle	2010-02-16 20:20:13	5A7CADDA-803
Name	2	DEPENDENTS	Logical	sol20-1	oracle	2010-02-16 20:20:13	26C08B75-F92F
Birthdate	3	DEPENDENTS	Logical	sol20-1	oracle	2010-02-16 20:20:13	9853896E-479B
EMPLOYEES_EMPLOYEE_ID	4	DEPENDENTS	Logical	sol20-1	oracle	2010-02-16 20:20:13	A0947173-36CE

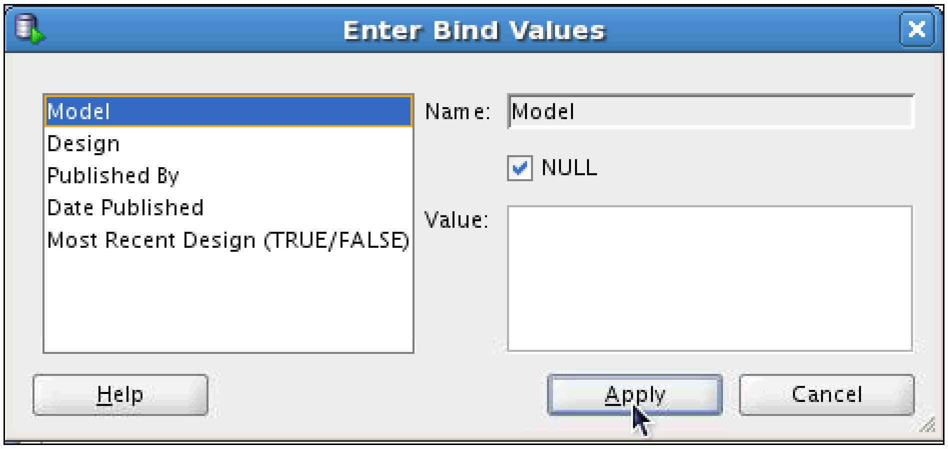
Attribute	Mandatory	Primary_UID	Relation_UID	Relationship	Kind	Type	Size	Precision	Scale
Name	N			(null)	Domain	(null)	50	0	0

73) You can also run some exception reports. Expand Design Rules > Relational Models, right-click Tables, and select Open.



74) Click Apply.

Solution 20-1: Re-Engineer the HR Schema (continued)



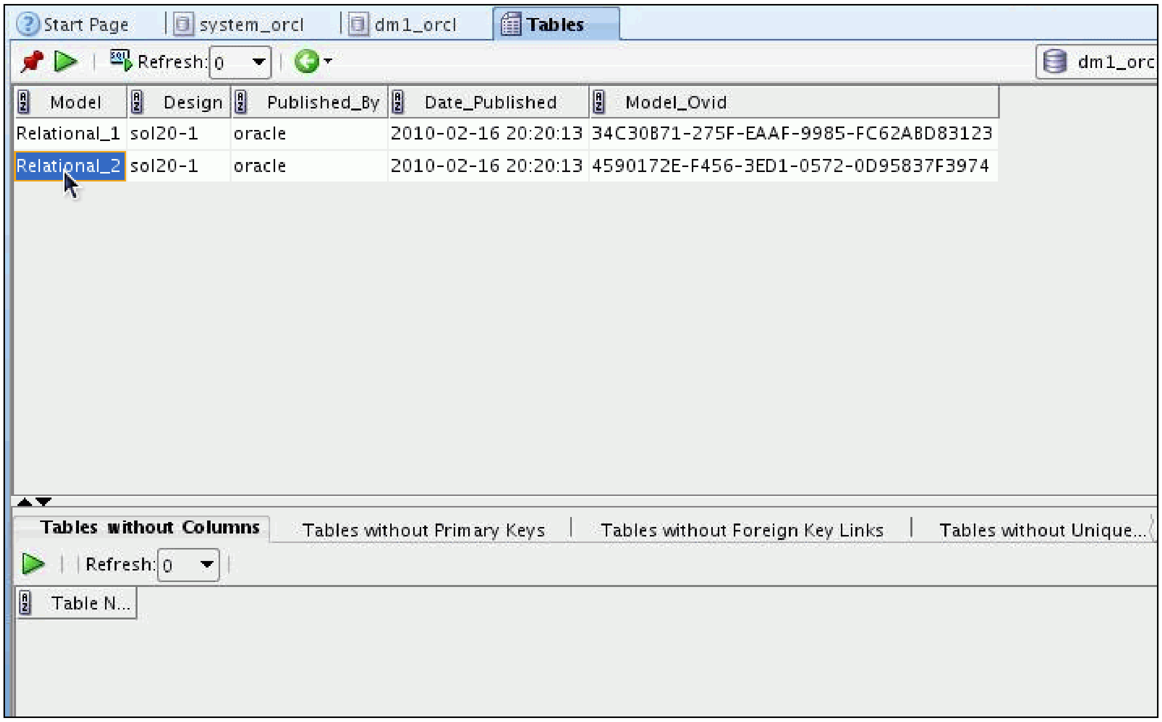
Model
Design
Published By
Date Published
Most Recent Design (TRUE/FALSE)

Name: Model

☑ NULL

Value:

75) All your relational models are listed. Select Relational_2 to see the “Tables without Columns” exceptions. In this case, all the tables have at least one column; otherwise, the table would be listed.



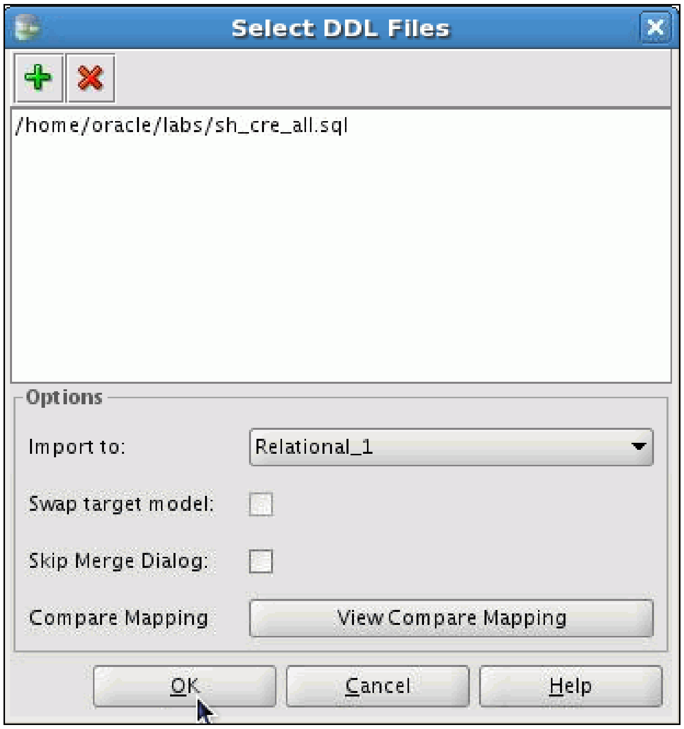
Model	Design	Published_By	Date_Published	Model_Ovid
Relational_1	sol20-1	oracle	2010-02-16 20:20:13	34C30B71-275F-EAAF-9985-FC62ABD83123
Relational_2	sol20-1	oracle	2010-02-16 20:20:13	4590172E-F456-3ED1-0572-0D95837F3974

76) Review the other supplied reports.

Solution 21-1: Build a Multidimensional Model

Below is one possible solution to this practice.

1) Close all open models.
2) Select File > Import and select DDL File.
3) Click the Add ✚ icon.
4) Select the `sh_cre_all.sql` file in the `/home/oracle/labs` directory, and click Open.
5) Click OK.



Select DDL Files

/home/oracle/labs/sh_cre_all.sql

Options

Import to: Relational_1

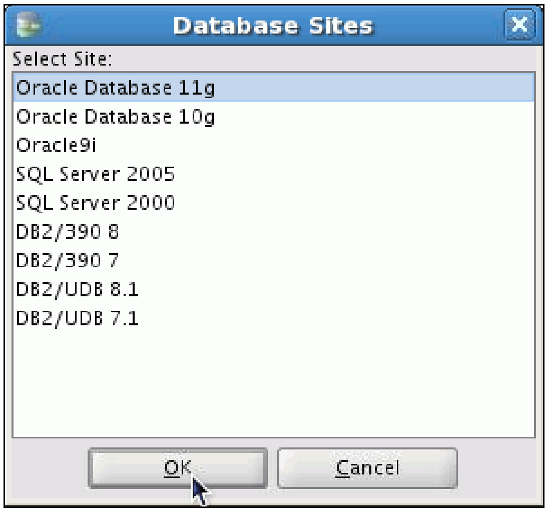
Swap target model: □

Skip Merge Dialog: □

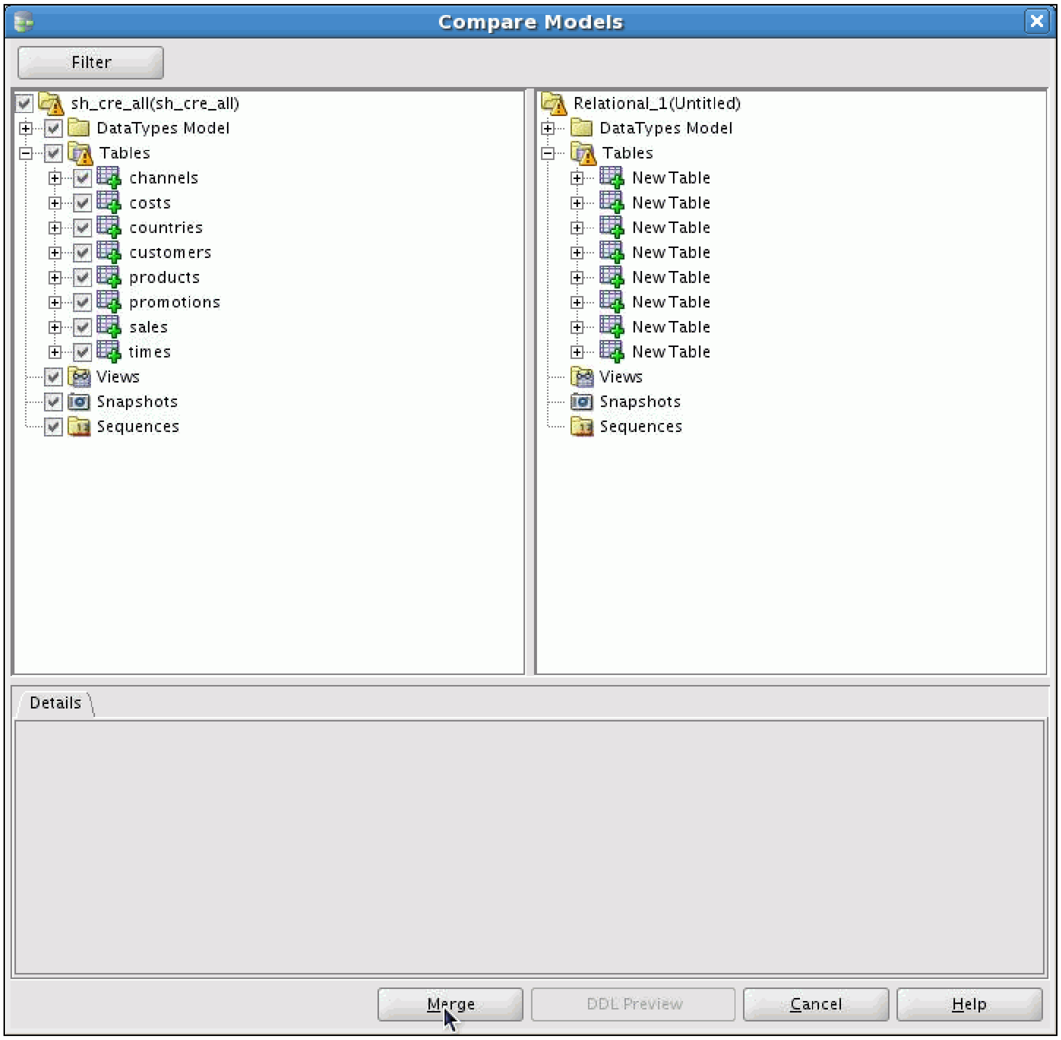
Compare Mapping: View Compare Mapping

6) Select Oracle Database 11g for Database Site, and click OK.

Solution 21-1: Build a Multidimensional Model (Continued)

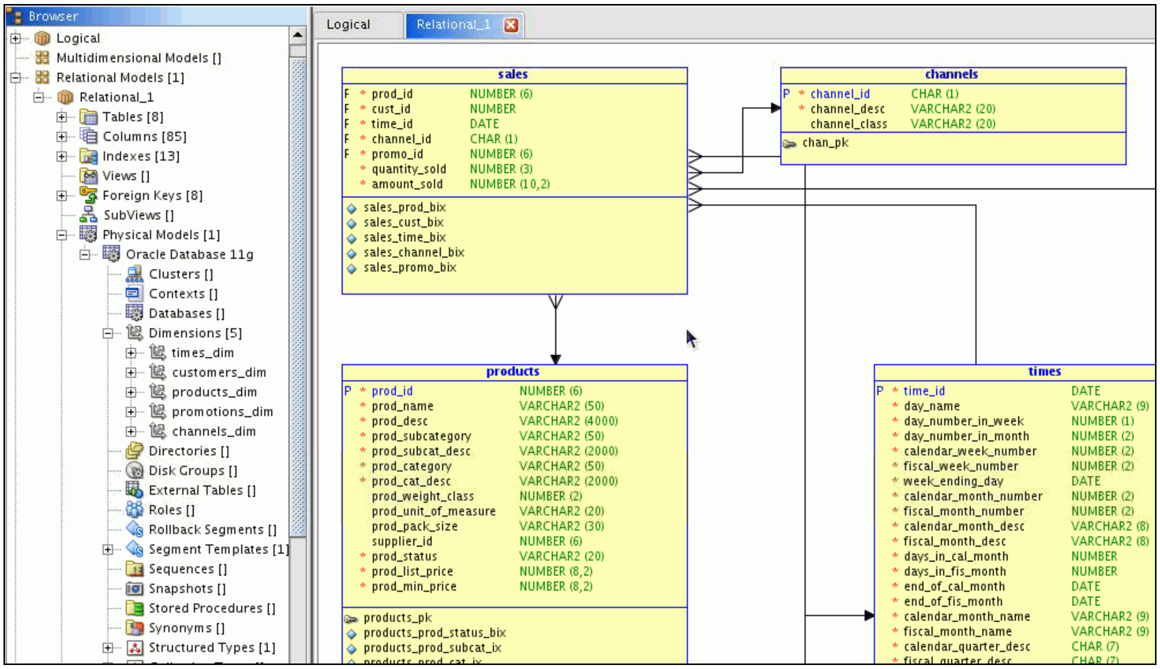


7) Expand the Tables node to see the tables that will be created, and click Merge.



8) Your relational model was created. To see the list of dimensions in your physical model, expand Relational Models > Relational_1 >Physical Model > Oracle Database 11g > Dimensions.

Solution 21-1: Build a Multidimensional Model (Continued)



sales

F	*	prod_id	NUMBER (6)
F	*	cust_id	NUMBER
F	*	time_id	DATE
F	*	channel_id	CHAR (1)
F	*	promo_id	NUMBER (6)
	*	quantity_sold	NUMBER (3)
	*	amount_sold	NUMBER (10,2)

◆ sales_prod_bix
◆ sales_cust_bix
◆ sales_time_bix
◆ sales_channel_bix
◆ sales_promo_bix

channels

P	*	channel_id	CHAR (1)
	*	channel_desc	VARCHAR2 (20)
		channel_class	VARCHAR2 (20)

chan_pk

products

P	*	prod_id	NUMBER (6)
	*	prod_name	VARCHAR2 (50)
	*	prod_desc	VARCHAR2 (4000)
	*	prod_subcategory	VARCHAR2 (50)
	*	prod_subcat_desc	VARCHAR2 (2000)
	*	prod_category	VARCHAR2 (50)
	*	prod_cat_desc	VARCHAR2 (2000)
		prod_weight_class	NUMBER (2)
		prod_unit_of_measure	VARCHAR2 (20)
		prod_pack_size	VARCHAR2 (30)
		supplier_id	NUMBER (6)
	*	prod_status	VARCHAR2 (20)
	*	prod_list_price	NUMBER (8,2)
	*	prod_min_price	NUMBER (8,2)

products_pk
◆ products_prod_status_bix
◆ products_prod_subcat_ix
◆ products_prod_cat_ix

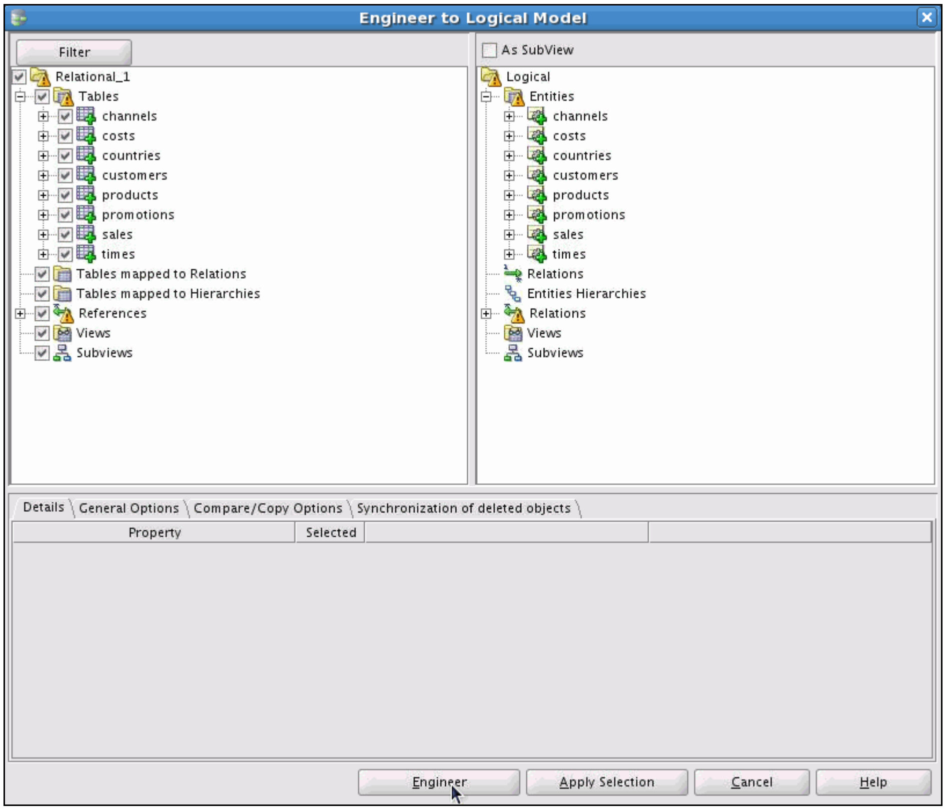
times

P	*	time_id	DATE
	*	day_name	VARCHAR2 (9)
	*	day_number_in_week	NUMBER (1)
	*	day_number_in_month	NUMBER (2)
	*	calendar_week_number	NUMBER (2)
	*	fiscal_week_number	NUMBER (2)
	*	week_ending_day	DATE
	*	calendar_month_number	NUMBER (2)
	*	fiscal_month_number	NUMBER (2)
	*	calendar_month_desc	VARCHAR2 (8)
	*	fiscal_month_desc	VARCHAR2 (8)
	*	days_in_cal_month	NUMBER
	*	days_in_fis_month	NUMBER
	*	end_of_cal_month	DATE
	*	end_of_fis_month	DATE
	*	calendar_month_name	VARCHAR2 (9)
	*	fiscal_month_name	VARCHAR2 (9)
	*	calendar_quarter_desc	CHAR (7)
	*	fiscal_quarter_desc	CHAR (7)

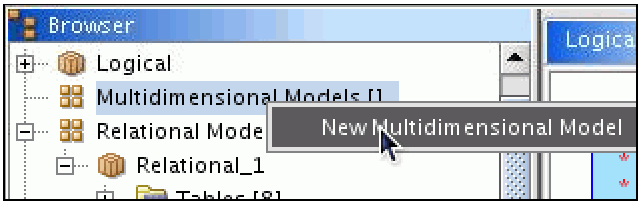
9) You can now create the Logical Model. Click the "Engineer to Logical Model" « reverse engineering icon.

10) Expand the Tables node to see the entities that will be created. Click Engineer.

Solution 21-1: Build a Multidimensional Model (Continued)



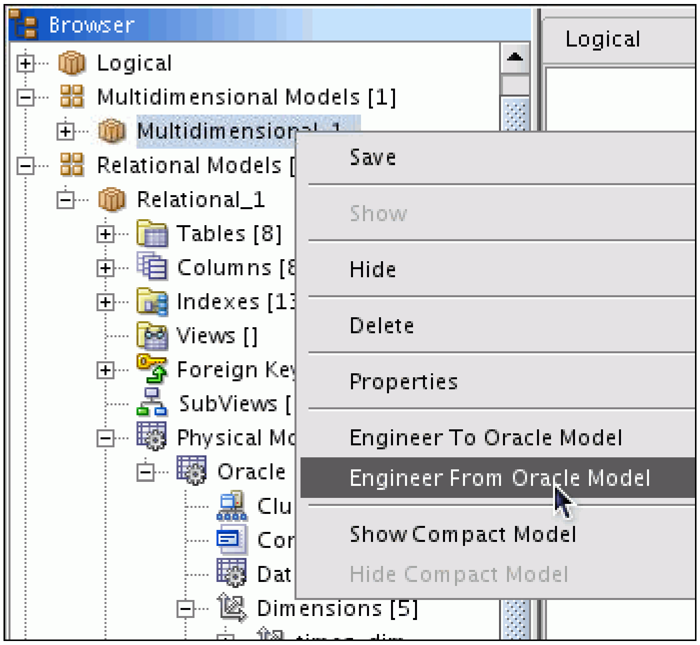
11) Your logical model was created. Now you are ready to create a multidimensional model. Right-click Multidimensional Models, and select New Multidimensional Model.



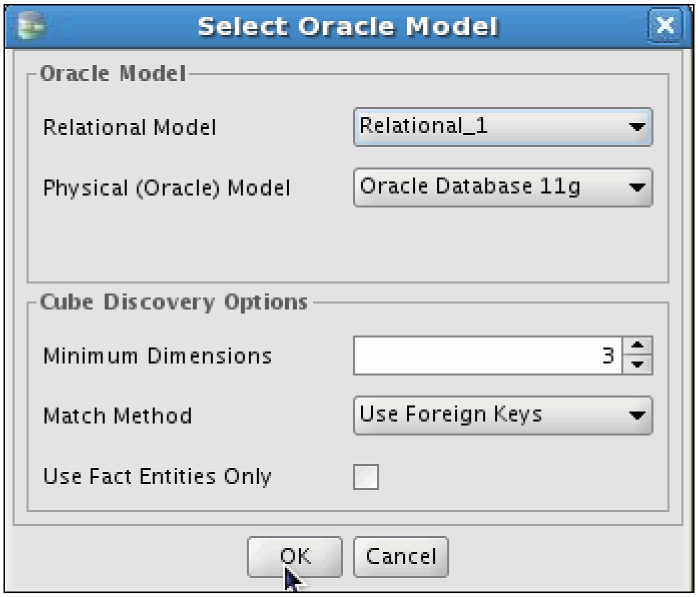
.

12) You can now populate the multidimensional model. Expand the Multidimensional Models node, right-click Multidimensional_1, and select Engineer From Oracle Model.

Solution 21-1: Build a Multidimensional Model (Continued)



13) Click OK to create the model.



Select Oracle Model

Oracle Model

Relational Model: Relational_1

Physical (Oracle) Model: Oracle Database 11g

Cube Discovery Options

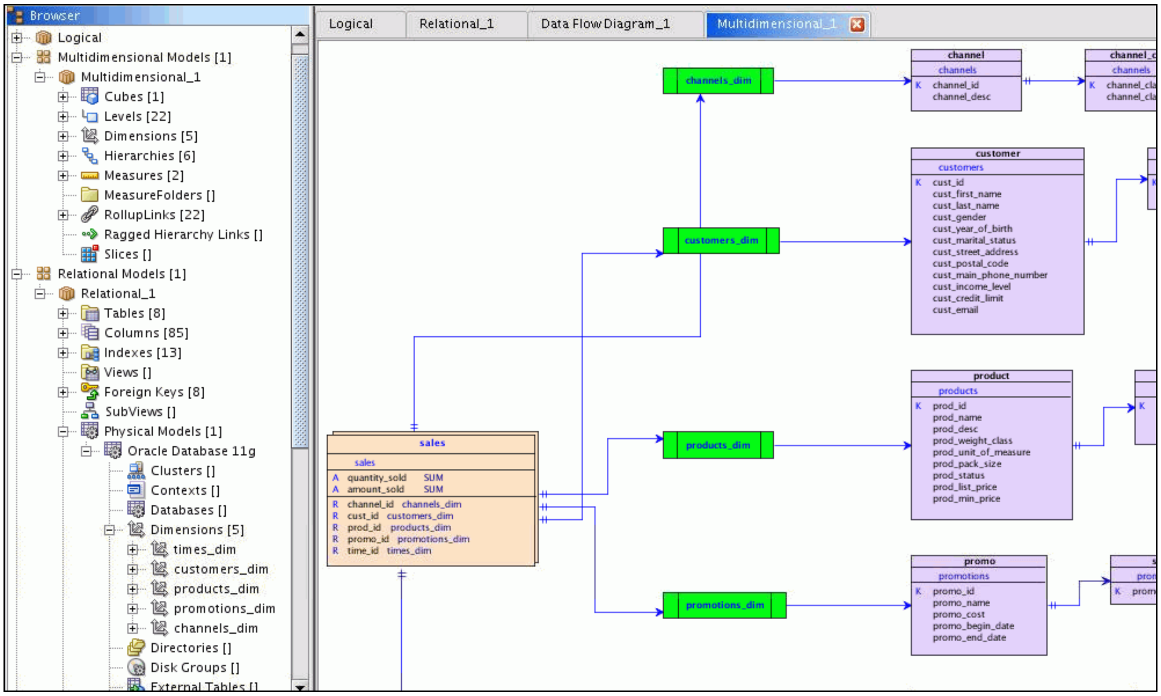
Minimum Dimensions: 3

Match Method: Use Foreign Keys

Use Fact Entities Only: □

14) The multidimensional model is displayed. Review the model.

Solution 21-1: Build a Multidimensional Model (Continued)



Browser
Logical
Multidimensional Models [1]
Multidimensional_1
Cubes [1]
Levels [22]
Dimensions [5]
Hierarchies [6]
Measures [2]
MeasureFolders []
RollupLinks [22]
Ragged Hierarchy Links []
Slices []
Relational Models [1]
Relational_1
Tables [8]
Columns [85]
Indexes [13]
Views []
Foreign Keys [8]
SubViews []
Physical Models [1]
Oracle Database 11g
Clusters []
Contexts []
Databases []
Dimensions [5]
times_dim
customers_dim
products_dim
promotions_dim
channels_dim
Directories []
Disk Groups []
External Tables []
Logical
Relational_1
Data Flow Diagram_1
Multidimensional_1
channels_dim
customers_dim
products_dim
promotions_dim
sales
sales
A quantity_sold SUM
A amount_sold SUM
R channel_id channels_dim
R cust_id customers_dim
R prod_id products_dim
R promo_id promotions_dim
R time_id times_dim
channel
channels
K channel_id
channel_desc
channel_c
channels
K channel_cla
channel_cla
customer
customers
K cust_id
cust_first_name
cust_last_name
cust_gender
cust_year_of_birth
cust_marital_status
cust_street_address
cust_postal_code
cust_main_phone_number
cust_income_level
cust_credit_limit
cust_email
product
products
K prod_id
prod_name
prod_desc
prod_weight_class
prod_unit_of_measure
prod_pack_size
prod_status
prod_list_price
prod_min_price
promo
promotions
K promo_id
promo_name
promo_cost
promo_begin_date
promo_end_date
K
K
prom
K prom