5-1
Mapping Entities and Attributes
Road Map

Mapping Entities and Attributes

Mapping Primary and Foreign Keys

You are here
Objectives

This lesson covers the following objectives:

• Describe why you need to create a relational model
• Explain the naming conventions used in a relational database
Objectives

This lesson covers the following objectives:

• Use Oracle SQL Developer Data Modeler to apply naming standards by creating:
  – Glossary
  – Name abbreviations
  – Design rules
  – Custom rules
  – Transformations

• Map simple entities to tables
• Map attributes to column names
Need for Creating a Relational Model

A relational model:

- Is the blue print for the actual database implementation
- Can be used as the basis for implementing any type of DBMS. The ideal model can be adapted to an RDBMS model.
Transformation Process: Logical to Relational Model

Mapping Entities and Attributes
Tables: A Recap

Table: EMPLOYEES

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Address</th>
<th>Birth_date</th>
<th>Department_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>Jones</td>
<td>12 Oxford Street</td>
<td>03-03-66</td>
<td>10</td>
</tr>
<tr>
<td>301</td>
<td>Smith</td>
<td>53 Hayes Drive</td>
<td>08-12-53</td>
<td>20</td>
</tr>
<tr>
<td>134</td>
<td>Gonzales</td>
<td>5609 Maple Court</td>
<td>10-02-87</td>
<td>40</td>
</tr>
</tbody>
</table>

Primary key column

Foreign key column to the DEPARTMENTS table

Table in the relational model
# Terminology Mapping: A Recap

<table>
<thead>
<tr>
<th>ANALYSIS</th>
<th>DESIGN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical model</td>
<td>Relational model</td>
</tr>
<tr>
<td>Entity</td>
<td>Table</td>
</tr>
<tr>
<td>Attribute</td>
<td>Column</td>
</tr>
<tr>
<td>Primary UID</td>
<td>Primary key</td>
</tr>
<tr>
<td>Secondary UID</td>
<td>Unique constraint</td>
</tr>
<tr>
<td>Relationship</td>
<td>Foreign key</td>
</tr>
<tr>
<td>Business constraints</td>
<td>Check constraints</td>
</tr>
</tbody>
</table>
Naming Conventions

Decide on conventions for:

• Table names
• Column names and special characters (%, *, #, -, space, …)
• Table short names (abbreviations)
Naming Conventions

Decide on conventions for:

• Primary and unique key constraint names
• Foreign key constraint names
• Foreign key column names
• Check constraints
Naming Restrictions with Oracle Database

• Table and column names:
  – Must start with a letter
  – May contain up to 30 alphanumeric characters
  – Must not contain spaces or some special characters
  – Must avoid reserved words

• Table names must be unique within a schema.

• Column names must be unique within a table.
Applying Naming Standards Using Oracle SQL Developer Data Modeler

- Glossary
- Name Translation
- Naming Templates
- Logical Model
- Relational Model
- Name Formatter
- Naming Restrictions
- Design Rule Validation

Mapping Entities and Attributes
Creating a Glossary from a Logical Model
Glossary Editor
Glossary Editor

Mapping Entities and Attributes
Name Abbreviations

Select a .CSV file with predefined names:
D:\Priya_D\Oracle DataModeler 4.0\datamodeler\datamodeler\templates\ABBREVS_SAMPLE.csv

Scope
- All Objects
- Abbreviations
- Tables
- Views
- FK Constraints
- PK & UK Constraints

Direction
- Name to abbreviation
- Abbreviation to name
- Keep letter case
- Separator:

Example of a .CSV file content:

CUST, CUSTOMER
CUST, CUS
Executing Design Rules

[Diagram showing menu options for executing design rules]

- Design Rules
- Compare/Merge Models
- Preferences...

[Diagram showing a window for executing design rules]

a. Design Rules dropdown
b. Design Rules window
Creating Rule Sets for Design Rules

1. Mapping Entities and Attributes

2. Design Rules

3. Design Rule Sets

4. Rule Sets

5. General

6. Logical

7. Relational

8. Process Model

9. Physical

Oracle Academy

DFo 5-1

Mapping Entities and Attributes

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.
Custom Design Rules and Transformation
Custom Design Rules and Transformation
Case Scenario: Design Rules

Let me see if the ERD for the Simplified Library Management System is complying with the database design rules.

If the ERD is not complying with the design rules, can I fix it?
Case Scenario: Applying Design Rules

The warning is fixed, and now the design is OK.
Mapping Simple Entities to Tables

- Entities

  - DEPARTMENT
  - EMPLOYEE

- Tables

  - DEPARTMENTS
  - EMPLOYEES
Naming Entities

The image shows a software interface for managing entity properties. The highlighted areas indicate:

- **a**: The name field is set to "DEPARTMENT".
- **b**: The preferred abbreviation field is set to "DEPARTMENTS".
- **c**: The long name field is set to "Entity_1".

These fields are part of the general properties of an entity, which can be used for mapping entities and attributes in a database or data modeling context.
Engineering Entities

[Image of a software interface for mapping entities and attributes]
Mapping Attributes to Columns

**Entities**

- **DEPARTMENT**
  - ID
  - Name

- **EMPLOYEE**
  - ID
  - First Name
  - Last Name
  - Email
  - Phone Number
  - Hire Date
  - Salary
  - Commission Percentage

**Tables**

- **DEPARTMENTS**
  - ID: NUMBER (6)
  - Name: VARCHAR2 (30)

- **EMPLOYEES**
  - ID: NUMBER (6)
  - First_Name: VARCHAR2 (20)
  - Last_Name: VARCHAR2 (25)
  - Email: VARCHAR2 (25)
  - Phone_Number: VARCHAR2 (20)
  - Hire_Date: DATE
  - SAL: NUMBER (10,2)
  - Commission_Percentage: NUMBER (2)
Mapping Attributes to Columns: Column Names

[Image of a screenshot showing the Attribute Properties window with the following details:

- **Name:** Salary
- **Preferred Abbreviation:** SAL
- **Datatype:** MONEY (10, 2)
- **Entity:** EMPLOYEE
- **Source Name:**
- **Source Type:**
- **Formula Description:**
- **Scope:**
- **Type Substitution:** ALL
- **Sensitive Type:**
- **Sensitive Data Description:**
- **Deprecated:**]
Engineering Attributes
Reviewing the Glossary

Mapping Entities and Attributes
Adding the Glossary as the Naming Standard

a. Browser
b. Naming Standard

c. Glossary
Mapping Attributes to Columns with the Glossary

- **Entities**

- **Tables After Glossary Is Applied**
Applying Name Abbreviations

- **.getTransaction()**
- **getName()**
- **getPublisher()**

**Diagram:**
- **Transaction, TRNS, Author, ATR, Publisher, PUBLR**
- **Name Abbreviations**
  - **Tools**
  - **Help**
  - **Domains Administration**
  - **Types Administration**
  - **RDBMS Site Administration**
  - **Mask Templates Administration**
  - **Table To View Wizard**
  - **View To Table Wizard**
  - **Types To Domains Wizard**
  - **Name Abbreviations**
  - **Glossary Editor**
  - **Object Names Administration**
  - **Design Rules**
  - **Compare/Merge Models**
  - **Preferences**

**Entity Relationship Diagram:**
- **BOOK_TRNS**
  - TRNS_ID VARCHAR2
  - TRNS_DATE DATE
  - TRNS_TYPE VARCHAR2
  - BOOK_ID VARCHAR2
  - MEMBER_ID VARCHAR2
  - BOOK_TRNS_PK(TRNS_ID)
- **PUBLR**
  - PUBLR_ID VARCHAR2
  - PUBLR_NAME VARCHAR2
  - PUBLR_PK(PUBLR_ID)

- **View Log**
- **Standardized Objects:**
  - Tables: 3
  - Columns: 10
  - Indexes: 0
  - Views: 0
Summary

In this lesson, you should have learned how to:

• Describe why you need to create a relational model
• Explain the naming conventions used in a relational database
Summary

In this lesson, you should have learned how to:

• Use Oracle SQL Developer Data Modeler to apply naming standards by creating:
  – Glossary
  – Name abbreviations
  – Design rules
  – Custom rules
  – Transformations

• Map simple entities to tables
• Map attributes to column names