Database Foundations

5-2
Mapping Primary and Foreign Keys
Road Map

Mapping Entities and Attributes

Mapping Primary and Foreign Keys

You are here
Objectives

This lesson covers the following objectives:

• Map UIDs to primary keys
• Engineer UIDs
• Map relationships to foreign keys
• Define naming templates
• Apply templates to the relational model
• Map exclusive relationships to foreign keys
• Map subtypes to tables
• Identify overlapping and folding keys
Basic Mapping: Unique Identifiers

Primary UID

Secondary UID
Mapping UIDs to Primary Keys

Entities

DEPARTMENT
- ID
- Name

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

Tables

DEPARTMENTS
- ID
- NAME

EMPLOYEES
- ID
- FIRST_NAME
- LAST_NAME
- EMAIL
- PH_NO
- HIRE_DATE
- SAL
- COMM_PCT
Engineering UIDs

Mapping Primary and Foreign Keys
Relationship Mapping

Barker Notation

Bachman Notation

<table>
<thead>
<tr>
<th>Barker Notation</th>
<th>Bachman Notation</th>
<th>Transformed To</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>P</td>
<td>Primary Key</td>
</tr>
<tr>
<td>U</td>
<td>U</td>
<td>Unique Key</td>
</tr>
<tr>
<td>Relationship</td>
<td>Relationship</td>
<td>Foreign Key</td>
</tr>
</tbody>
</table>
Mapping Relationships to Foreign Keys

Entities

DEPARTMENT
- ID
- Name

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

1:N relation

Tables

DEPARTMENTS
- ID
  - NUMBER (6)
  - Name
    - VARCHAR2 (50)

EMPLOYEES
- ID
  - NUMBER (6)
  - First_Name
    - VARCHAR2 (50)
  - Last_Name
    - VARCHAR2 (50)
  - Email
    - VARCHAR2 (50)
  - Phone_Number
    - VARCHAR2 (20)
  - Hire_Date
    - DATE
  - Salary
  - NUMBER (9,2)
  - Commission_Percentage
    - NUMBER (9,2)

- DEPARTMENTS_ID
  - NUMBER (6)
- EMPLOYEES_ID
  - NUMBER (6)

Foreign key
Mapping Relationships to Foreign Keys
Defining Naming Templates

Right-click Design > Properties > Naming Standard > Templates.
Example: Naming Templates

- **Table name:** ADMIN
- **Model name:** ORACLEDEMO

<table>
<thead>
<tr>
<th>Template</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{table}_PK</code></td>
<td>ADMIN_PK</td>
</tr>
<tr>
<td><code>SUBSTR(7,4,FRONT,{model})</code></td>
<td>DEMO</td>
</tr>
<tr>
<td><code>SUBSTR(1,3,FRONT,{table})</code></td>
<td>ADM</td>
</tr>
<tr>
<td><code>SUBSTR(1,3,FRONT,TABLE)</code></td>
<td>TAB (where &quot;TABLE&quot; is a constant rather than a variable)</td>
</tr>
<tr>
<td><code>IX_SUBSTR(7,4,FRONT,{model})_SUBSTR(1,3,FRONT,{table})_{seq nr}</code></td>
<td>IX_DEMO_ADM_1</td>
</tr>
</tbody>
</table>
Applying Templates to One Table

- **a** Mapping Primary and Foreign Keys
- **b** Generating table properties
- **c** Applying naming rules
Applying Templates to the Relational Model

Mapping Primary and Foreign Keys
Managing Prefixes

Change Object Names Prefix

Prefix replacement
- Add Classification prefix

Current Prefix:
New Prefix: HR_
- Case sensitive
- Add new prefix

Apply to
- Tables
- Columns
- Views
- Indexes
- FK Constraints
- PK & UK Constraints

Apply
Cancel Help

Message
2 names have been changed
OK

HR_DEPARTMENTS

ID | NUMBER (6)
Name | VARCHAR2 (30)
Department_PK (ID)

HR_EMPLOYEES

ID (PK)
First_Name | VARCHAR2 (20)
Last_Name | VARCHAR2 (25)
Email | VARCHAR2 (25)
Phone_Number | VARCHAR2 (20)
Hire_Date | DATE
Salary | NUMBER (1,0,2)
Commission_Percentage | NUMBER (2)
Department_ID | NUMBER (6)
Employees_PK (ID)
Case Scenario: Template and Prefix

How about applying the naming templates to the Simplified Library Management Relational Model?

Maybe it might be a good idea to prefix the objects with "SLM"?
Case Scenario: Applying Naming Templates

Mapping Primary and Foreign Keys
Case Scenario: Applying the Prefix

a. Change Object Names Prefix
b. Prefix replacement: SLM_
c. 9 names have been changed
d. SLM_BOOK_TRNS
   - TRNS_ID VARCHAR2
   - TRNS_DATE DATE
   - TRNS_TYPE VARCHAR2
   - BOOK_ID VARCHAR2
   - MEMBER_ID VARCHAR2
   - BOOK_TRNS_PK(BOOK_ID)
   - BOOK_TRNS_BOOK_ID_PK(BOOK_ID)

d. SLM_MEMBERS
   - MEMBER_ID VARCHAR2
   - FIRST_NAME VARCHAR2
   - LAST_NAME VARCHAR2
   - STREET_ADDRESS VARCHAR2
   - CITY VARCHAR2
   - STATE VARCHAR2
   - ZIP VARCHAR2
   - MEMBER_PK(MEMBER_ID)
Mapping Exclusive Relationships to Foreign Keys

Entities

Exclusive relationship

Tables

Foreign keys

EMPLOYEE

# ID
# First Name
# Last Name
# Email
# Phone Number
# Hire Date
# Salary
# Commission Percentage

COUNTRY
# Code
# Name

COUNTY
# Code
# Name

OTHER STATE
# ID
# Name

EMP PK (ID)

EMP
P = ID
F = FIRST_NM VARCHAR2 (20)
F = LAST_NM VARCHAR2 (25)
F = EMAIL VARCHAR2 (25)
F = PH_NO VARCHAR2 (20)
F = HIRE_DATE DATE
F = SAL NUMBER (10, 2)
F = CTRY_CODE NUMBER (8)
F = COUNTRY_CODE NUMBER (8)
F = OTHER_STATE_ID NUMBER (8)

CTRY
P = CODE
F = NM VARCHAR2 (50)
F = CTRY_PK ICODD

COUNTRY
P = CODE
F = NM VARCHAR2 (50)
F = COUNTRY_PK ICODD

OTHER_STATE
P = ID
F = NM VARCHAR2 (50)
F = OTHER_STATE_PK ICODD
Engineering Exclusive Relationships

Mapping Primary and Foreign Keys
Mapping Subtypes to Tables

[Diagram showing a database model with entities like Department, Employee, Exempt Employee, Non-Exempt Employee, Country, County, and Other State, with relationships and attributes.]
Applying General Options

In the 'General Options' tab, you can select options such as:
- Engineer Coordinates
- Engineer only objects created in "Logical" model
- Apply name translation
- Use preferred abbreviations
- Use Template Table

These options help in customizing the mapping process according to specific requirements.
Setting Compare/Copy Options
Viewing the Mapping Comparison
Synchronizing Deleted Objects
Identifying Overlapping and Folding Keys

• Two attributes in the same entity relate to the same UID attribute.

• You can fold the keys into one column in the relational model during engineering.
Identifying Overlapping and Folding Keys

[Diagram of a software interface showing a window labeled "Engineer to Relational Model" with categorized views such as Logical, Relations, Views, and Subviews on the left side and Relational_1 on the right side. The window is divided into two panels: one displaying "Logical" and "Entity Hierarchies" and the other showing "Tables", "Objects mapped to relations", "Views", and "Subviews". The highlighted section of the interface focuses on "Mapping Primary and Foreign Keys" with a table listing entities and attribute pairs such as "Class Enrollment" and "Course_Division_ID - Student_Division_ID".]

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.
Summary

In this lesson, you should have learned how to:

• Map UIDs to primary keys
• Engineer UIDs
• Map relationships to foreign keys
• Define naming templates
• Apply templates to the relational model
• Map exclusive relationships to foreign keys
• Map subtypes to tables
• Identify overlapping and folding keys