ORACLE

Academy

# Database Design

## 9-1

## Introduction to Relational Database Concepts

ORACLE
Academy

# Objectives

- This lesson covers the following objectives:
  - Define a primary key
  - Define a foreign key
  - Define a column-integrity rule
  - Identify row, column, primary key, unique key, and foreign key elements given a diagram of a table containing these elements
  - Identify violations of data-integrity rules

# Purpose

- The conceptual data model will be transformed into a relational database design
- This means that our entities, attributes, relationships, and unique identifiers will be translated into objects in a relational database
- Compare this to a clothing designer who is taking his design from paper and implementing it with fabric
- The designer needs to understand how to sew the designs just like you will need to understand the structure of relational database objects

# Relational Database Illustrated

- A relational database is a database that is seen by the user as a collection of two-dimensional tables, each containing rows and columns
- The table below contains employee data

**EMPLOYEES (table name)**

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | DEPARTMENT_ID |
|---|---|---|---|
| 100 | Steven | King | 90 |
| 101 | Neena | Kochhar | 90 |
| 102 | Lex | De Haan | 90 |
| 200 | Jennifer | Whalen | 10 |
| 205 | Shelley | Higgins | 110 |

Row

Column

# Language to Access Data

- Structured query language (SQL) allows us to access data in relational databases in an efficient way
- Instead of manually searching through each row to find the record for employee number 200, we use the following SQL statement:

```
SELECT last_name, department_id
FROM employees
WHERE employee_id = 200;
```

- You can see the result of this statement on the next slide

# SQL Query Illustrated

**EMPLOYEES (table name)**

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | DEPARTMENT_ID |
|---|---|---|---|
| 100 | Steven | King | 90 |
| 101 | Neena | Kochhar | 90 |
| 102 | Lex | De Haan | 90 |
| **200** | **Jennifer** | **Whalen** | **10** |
| 205 | Shelley | Higgins | 110 |

```
SELECT last_name, department_id
FROM employees
WHERE employee_id = 200;
```

| LAST_NAME | DEPARTMENT_ID |
|---|---|
| Whalen | 10 |

# Specific SQL Query

- To find all the employees in department number 90, we write a different SQL statement:

```
SELECT *
FROM employees
WHERE department_id = 90;
```

- Again, you can see the result on the next slide

# Specific SQL Query

EMPLOYEES (table name)

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | …. | DEPARTMENT_ID |
|---|---|---|---|---|
| 100 | Steven | King | …. | 90 |
| 101 | Neena | Kochhar | …. | 90 |
| 102 | Lex | De Haan | …. | 90 |
| 200 | Jennifer | Whalen | …. | 10 |
| 205 | Shelley | Higgins | …. | 110 |

```
SELECT  *
FROM employees
WHERE department_id = 90;
```

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | …. | DEPARTMENT_ID |
|---|---|---|---|---|
| 100 | Steven | King | …. | 90 |
| 101 | Neena | Kochhar | …. | 90 |
| 102 | Lex | De Haan | …. | 90 |

# Primary Key

- A primary key (PK) is a column or set of columns that uniquely identifies each row in a table

**ACCOUNTS**

| BANK_NO | ACCT_NO | BALANCE | DATE_OPENED |
|---|---|---|---|
| 104 | 75760 | 120,050.00 | 21-OKT-89 |
| 104 | 77956 | 100.10 | |
| 105 | 89570 | 55,775.00 | 15-JAN-85 |
| 103 | 55890 | 15,001.85 | 10-MAR-91 |
| 105 | 75760 | 5.00 | 22-SEP-03 |

Multiple Column Primary Key

**EMPLOYEES**

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | .... | DEPARTMENT_ID |
|---|---|---|---|---|
| 100 | Steven | King | .... | 90 |
| 101 | Neena | Kochhar | .... | 90 |
| 102 | Lex | De Haan | .... | 90 |
| 200 | Jennifer | Whalen | .... | 10 |
| 205 | Shelley | Higgins | .... | 110 |

Single Column Primary Key

ORACLE
Academy

# Primary Key

- Each table should have a primary key, and a primary key must be unique

**ACCOUNTS**

| BANK_NO | ACCT_NO | BALANCE | DATE_OPENED |
|---|---|---|---|
| 104 | 75760 | 120,050.00 | 21-OKT-89 |
| 104 | 77956 | 100.10 | |
| 105 | 89570 | 55,775.00 | 15-JAN-85 |
| 103 | 55890 | 15,001.85 | 10-MAR-91 |
| 105 | 75760 | 5.00 | 22-SEP-03 |

Multiple Column Primary Key

**EMPLOYEES**

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | …. | DEPARTMENT_ID |
|---|---|---|---|---|
| 100 | Steven | King | …. | 90 |
| 101 | Neena | Kochhar | …. | 90 |
| 102 | Lex | De Haan | …. | 90 |
| 200 | Jennifer | Whalen | …. | 10 |
| 205 | Shelley | Higgins | …. | 110 |

Single Column Primary Key

ORACLE
Academy

# Primary Key

- No part of the primary key can be null

**ACCOUNTS**

| BANK_NO | ACCT_NO | BALANCE | DATE_OPENED |
|---|---|---|---|
| 104 | 75760 | 120,050.00 | 21-OKT-89 |
| 104 | 77956 | 100.10 | |
| 105 | 89570 | 55,775.00 | 15-JAN-85 |
| 103 | 55890 | 15,001.85 | 10-MAR-91 |
| 105 | 75760 | 5.00 | 22-SEP-03 |

Multiple Column Primary Key

**EMPLOYEES**

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | …. | DEPARTMENT_ID |
|---|---|---|---|---|
| 100 | Steven | King | …. | 90 |
| 101 | Neena | Kochhar | …. | 90 |
| 102 | Lex | De Haan | …. | 90 |
| 200 | Jennifer | Whalen | …. | 10 |
| 205 | Shelley | Higgins | …. | 110 |

Single Column Primary Key

# Primary Key Candidates

- A table can have more than one column, or combinations of columns, that could serve as the table’s primary key
- Each column, or combination of columns, is called a "candidate" key because it could be selected for use as the primary key

**MEMBERS**

| MEMBER_ID | LAST_NAME | FIRST_NAME | PAYROLL_ID |
|---|---|---|---|
| 100 | SMITH | DANA | 21215 |
| 310 | ADAMS | TYLER | 59877 |
| 210 | CHEN | LAWRENCE | 1101 |
| 405 | GOMEZ | CARLOS | 52 |
| 378 | LOUNGANI | NEIL | 90386 |

↑ Candidate Key

Candidate Key ↑

# Choose a Candidate Key

- Select one candidate key to be the primary key for the table
- The other candidates become alternate keys (or unique keys)

**MEMBERS**

| MEMBER_ID | LAST_NAME | FIRST_NAME | PAYROLL_ID |
|---|---|---|---|
| 100 | SMITH | DANA | 21215 |
| 310 | ADAMS | TYLER | 59877 |
| 210 | CHEN | LAWRENCE | 1101 |
| 405 | GOMEZ | CARLOS | 52 |
| 378 | LOUNGANI | NEIL | 90386 |

Primary Key

Alternate or Unique Key (UK)

# Foreign Key

- A foreign key (FK) is a column, or combination of columns, in one table that contains values that match the primary key value in another table

Foreign Key

EMPLOYEES

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | .... | DEPARTMENT_ID |
|---|---|---|---|---|
| 100 | Steven | King | .... | 90 |
| 101 | Neena | Kochhar | .... | 90 |
| 102 | Lex | De Haan | .... | 90 |
| 200 | Jennifer | Whalen | .... | 10 |
| 205 | Shelley | Higgins | .... | 110 |

refers to

DEPARTMENTS

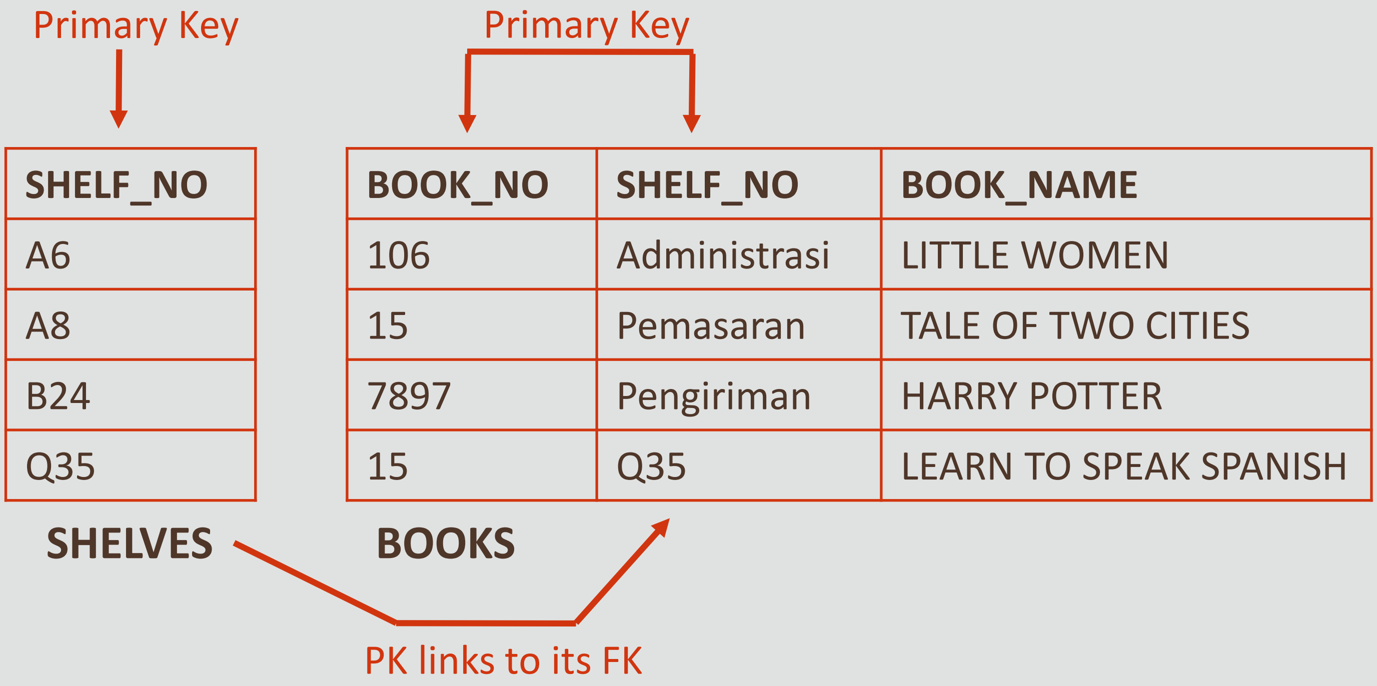
| DEPARTMENT_ID | DEPARTMENT_NAME |
|---|---|
| 10 | Administrasi |
| 20 | Pemasaran |
| 50 | Pengiriman |

Primary Key

ORACLE
Academy

# Foreign Key Rules

- If a primary key is composed of one or more foreign keys, the FK value cannot be NULL



| SHELF_NO |
|---|
| A6 |
| A8 |
| B24 |
| Q35 |

SHELVES

| BOOK_NO | SHELF_NO | BOOK_NAME |
|---|---|---|
| 106 | Administrasi | LITTLE WOMEN |
| 15 | Pemasaran | TALE OF TWO CITIES |
| 7897 | Pengiriman | HARRY POTTER |
| 15 | Q35 | LEARN TO SPEAK SPANISH |

BOOKS

ORACLE Academy

# Column Integrity

- A column must contain only values that are consistent with the defined data format of the column

**ACCOUNTS**

| BANK_NO | ACCT_NO | BALANCE | DATE_OPENED |
|---|---|---|---|
| 104 | 75760 | 120050.00 | 21-OKT-89 |
| 104 | 77956 | 100.10 | |
| 105 | 89570 | 55775.00 | 15-JAN-85 |
| 103 | 55890 | 15001.85 | 10-MAR-91 |
| 105 | 75760 | 5.00 | 22-SEP-03 |

**ACCOUNTS Table Definition**

| Column Name | Data Type | Optionality |
|---|---|---|
| BANK_NO | Number (5) | Not Null |
| ACCT_NO | Number (8) | Not Null |
| BALANCE | Number (12,2) | Not Null |
| DATE_OPENED | Date | |

ORACLE
Academy

# Summary of Data-Integrity Rules

- Data-integrity rules (also known as constraints) define the relationally correct state for a database
- Data-integrity rules ensure that users can perform only those operations that leave the database in a correct, consistent state

# Summary of Data-Integrity Rules

| Constraint Type | Explanation | Example |
|---|---|---|
| **Entity Integrity** | A primary key must be unique, and no part of the primary key can be null | The column emp_no in the EMPLOYEES table cannot be null |
| **Referential Integrity** | A foreign key must match an existing primary key value (or else be null if nulls are allowed) | The value in the dept_no column of the EMPLOYEES table must match a value in the dept_no column in the DEPARTMENTS table |
| **Column Integrity** | A column must contain only values consistent with the defined data format of the column | The value in the balance column of the ACCOUNTS table must be numeric |
| **User-Defined Integrity** | The data stored in a database must comply with the rules of the business | If the value in the balance column of the ACCOUNTS table is below 1.00, we must send a letter to the account owner ( this will need additional programming to enforce) |

# Terminology

- Key terms used in this lesson included:
  - Candidate key
  - Column
  - Foreign key
  - Primary key
  - Relational database
  - Row
  - Unique key

# Summary

- In this lesson, you should have learned how to:
  - Define a primary key
  - Define a foreign key
  - Define a column-integrity rule
  - Identify row, column, primary key, unique key, and foreign key elements given a diagram of a table containing these elements
  - Identify violations of data-integrity rules

ORACLE

Academy