# DATABASE FOUNDATIONS

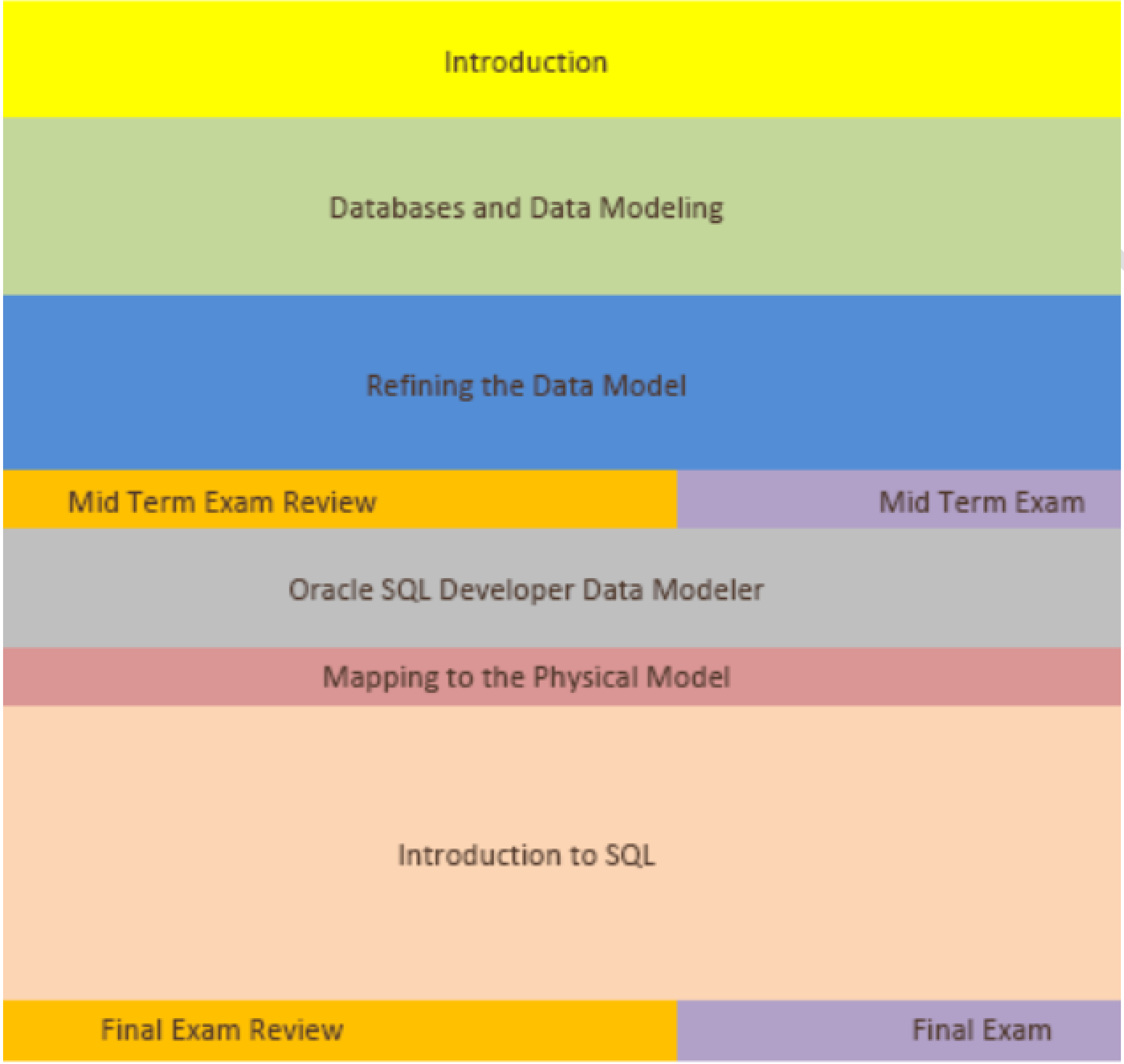
ORACLE ACADEMY

6 DE MAYO DE 2025

UNIVERSIDAD POLITECNICA DE AGUASCALIENTES
Juan Carlos Herrera Hernández

# Contenido

# 1. Introduction



Introduction

Databases and Data Modeling

Refining the Data Model

Mid Term Exam Review

Mid Term Exam

Oracle SQL Developer Data Modeler

Mapping to the Physical Model

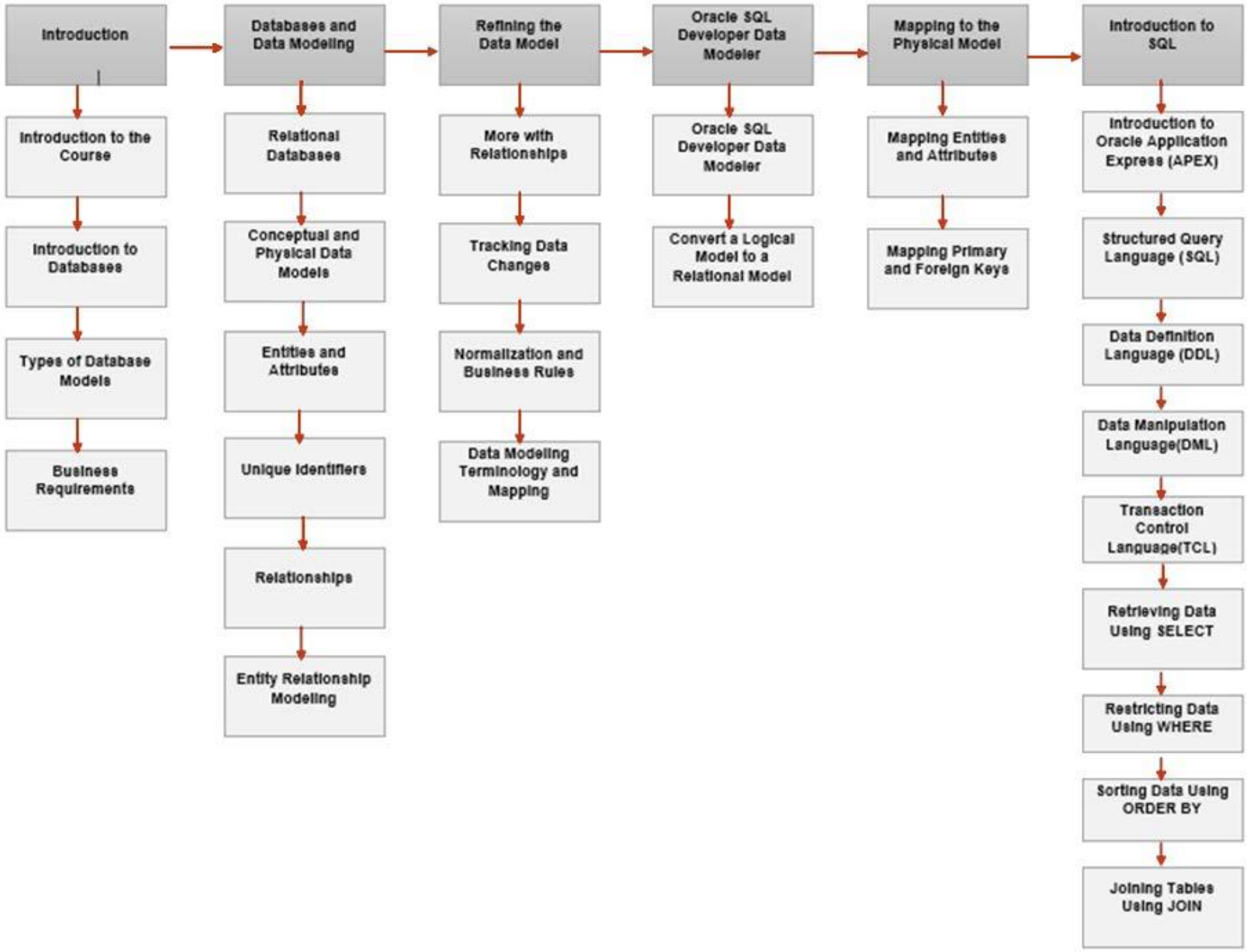
Introduction to SQL

Final Exam Review

Final Exam

→

## 1.1. Introduction



Introduction
Introduction to the Course
Introduction to Databases
Types of Database Models
Business Requirements
Databases and Data Modeling
Relational Databases
Conceptual and Physical Data Models
Entities and Attributes
Unique Identifiers
Relationships
Entity Relationship Modeling
Refining the Data Model
More with Relationships
Tracking Data Changes
Normalization and Business Rules
Data Modeling Terminology and Mapping
Oracle SQL Developer Data Modeler
Oracle SQL Developer Data Modeler
Convert a Logical Model to a Relational Model
Mapping to the Physical Model
Mapping Entities and Attributes
Mapping Primary and Foreign Keys
Introduction to SQL
Introduction to Oracle Application Express (APEX)
Structured Query Language (SQL)
Data Definition Language (DDL)
Data Manipulation Language(DML)
Transaction Control Language(TCL)
Retrieving Data Using SELECT
Restricting Data Using WHERE
Sorting Data Using ORDER BY
Joining Tables Using JOIN

Technological Requirements:

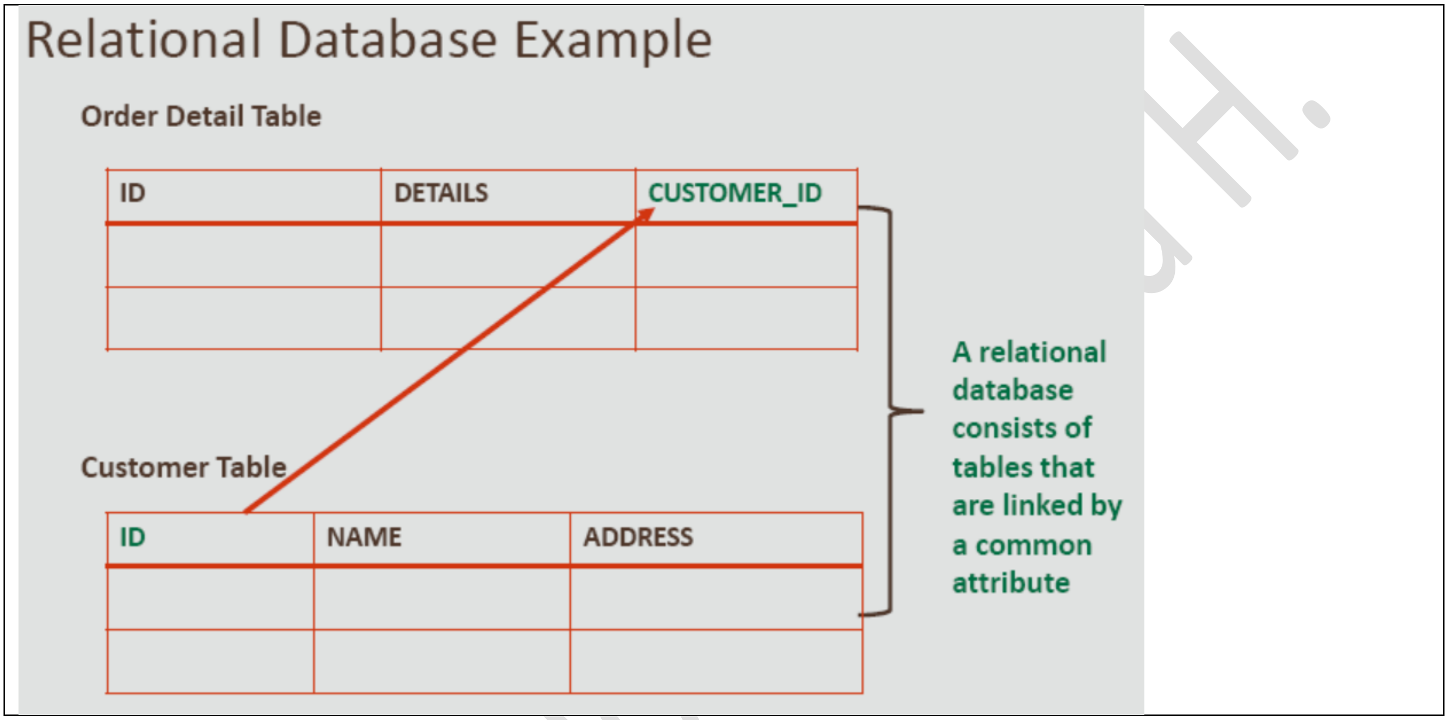
Oracle SQL Developer  or Oralce APEX application
Oracle Data Modeler

→

## 1.2. Introduction to Databases

Data vs Information.

- A relational database stores information in tables with rows and columns
- A table is a collection of records
- A row is called a record (or instance)
- A column is referred to as a field (or attribute)



# Relational Database Example

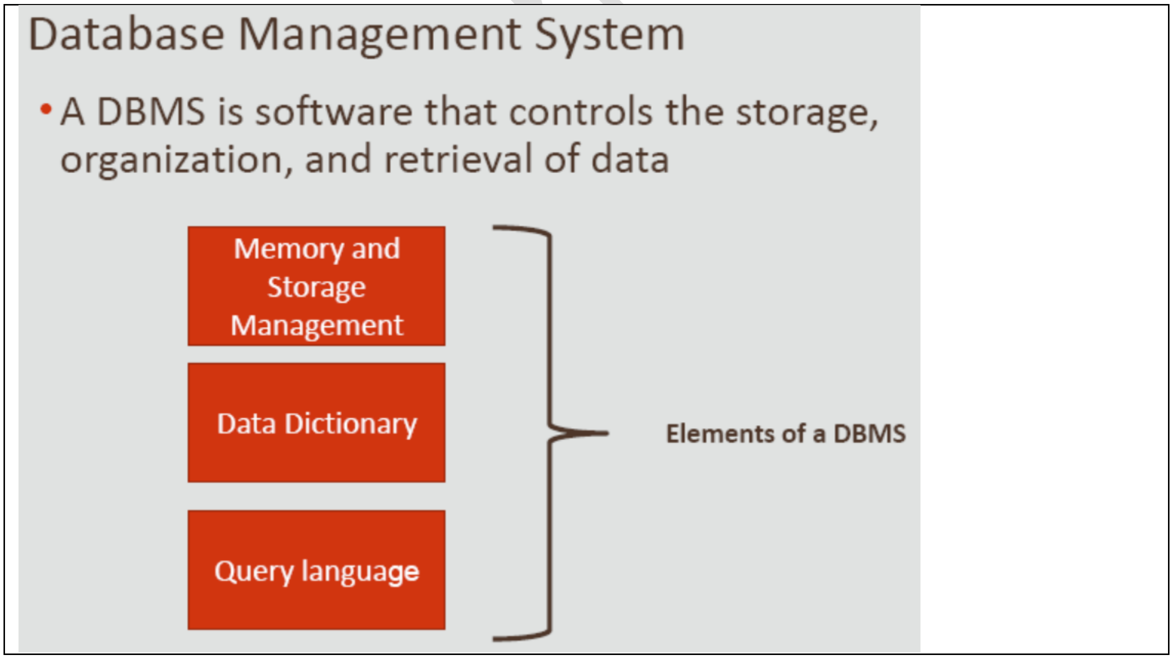
Order Detail Table

| ID | DETAILS | CUSTOMER_ID |
|---|---|---|
| | | |
| | | |

Customer Table

| ID | NAME | ADDRESS |
|---|---|---|
| | | |
| | | |

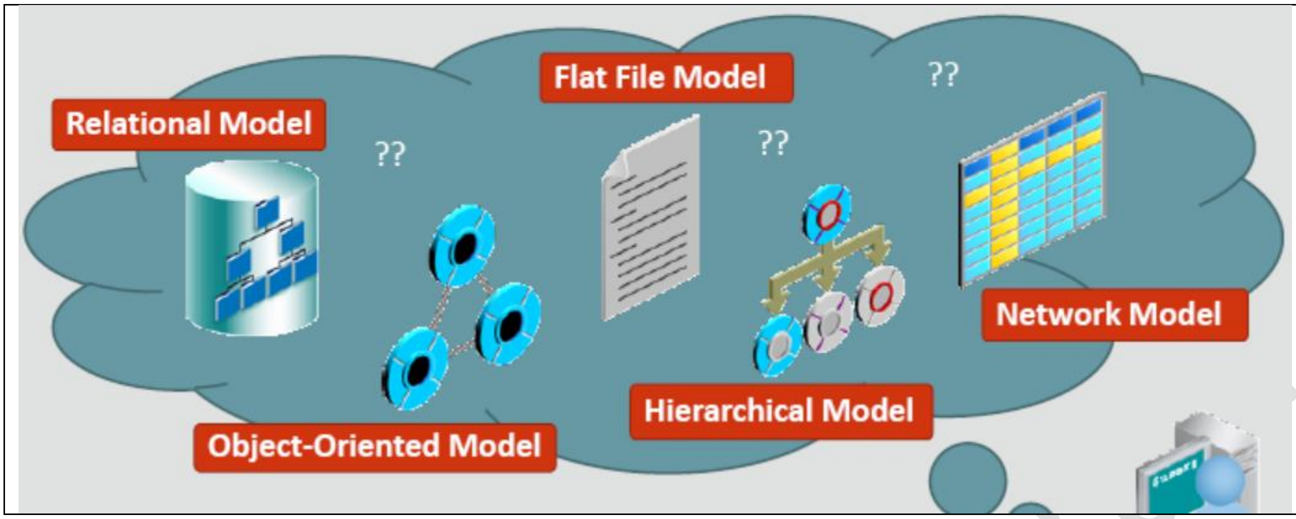
A relational database consists of tables that are linked by a common attribute

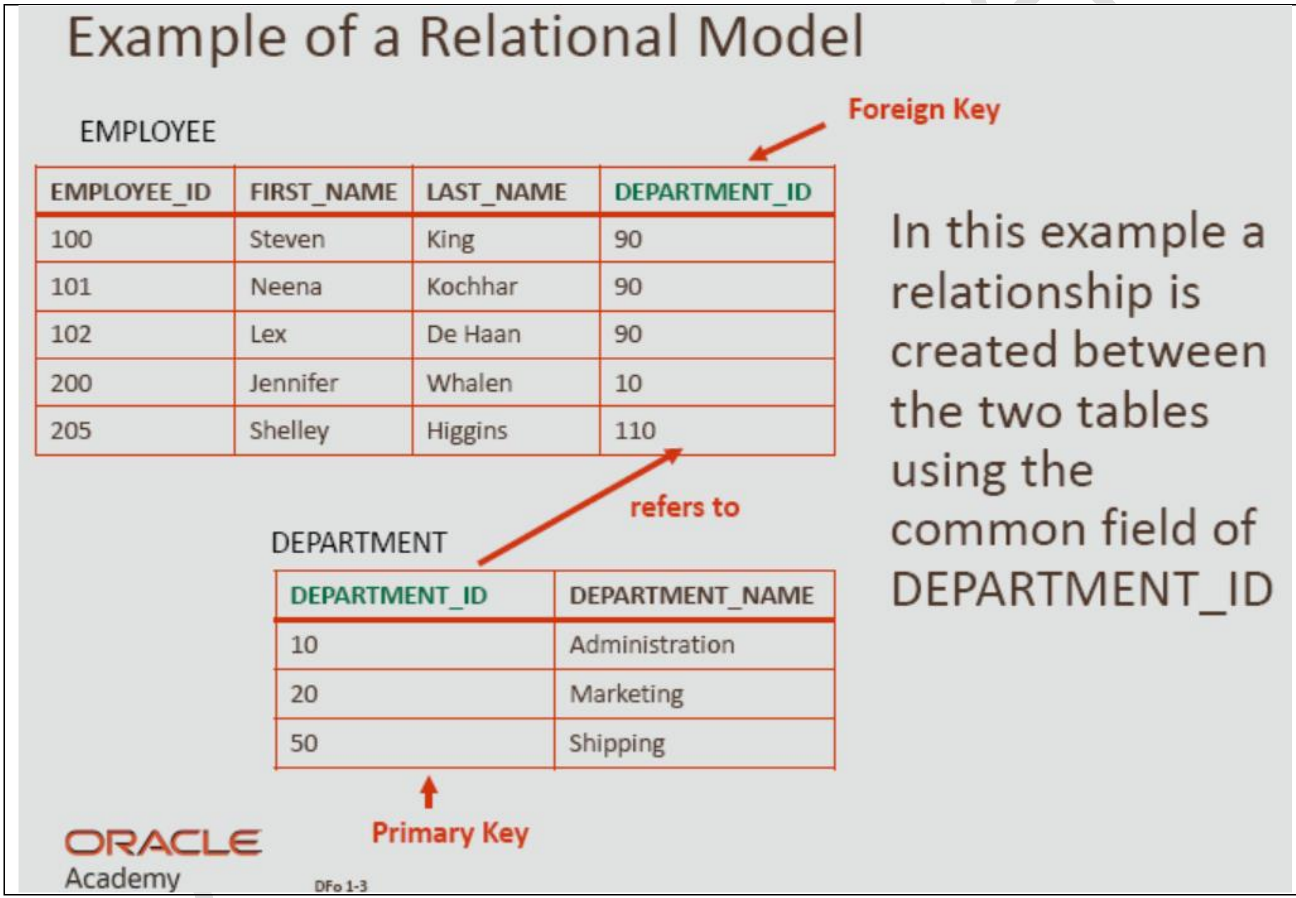


# Database Management System

- A DBMS is software that controls the storage, organization, and retrieval of data

## 1.3. Types of Database Models





Example of a Relational Model

EMPLOYEE

Foreign Key

| 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

DEPARTMENT

| DEPARTMENT_ID | DEPARTMENT_NAME |
|---|---|
| 10 | Administration |
| 20 | Marketing |
| 50 | Shipping |

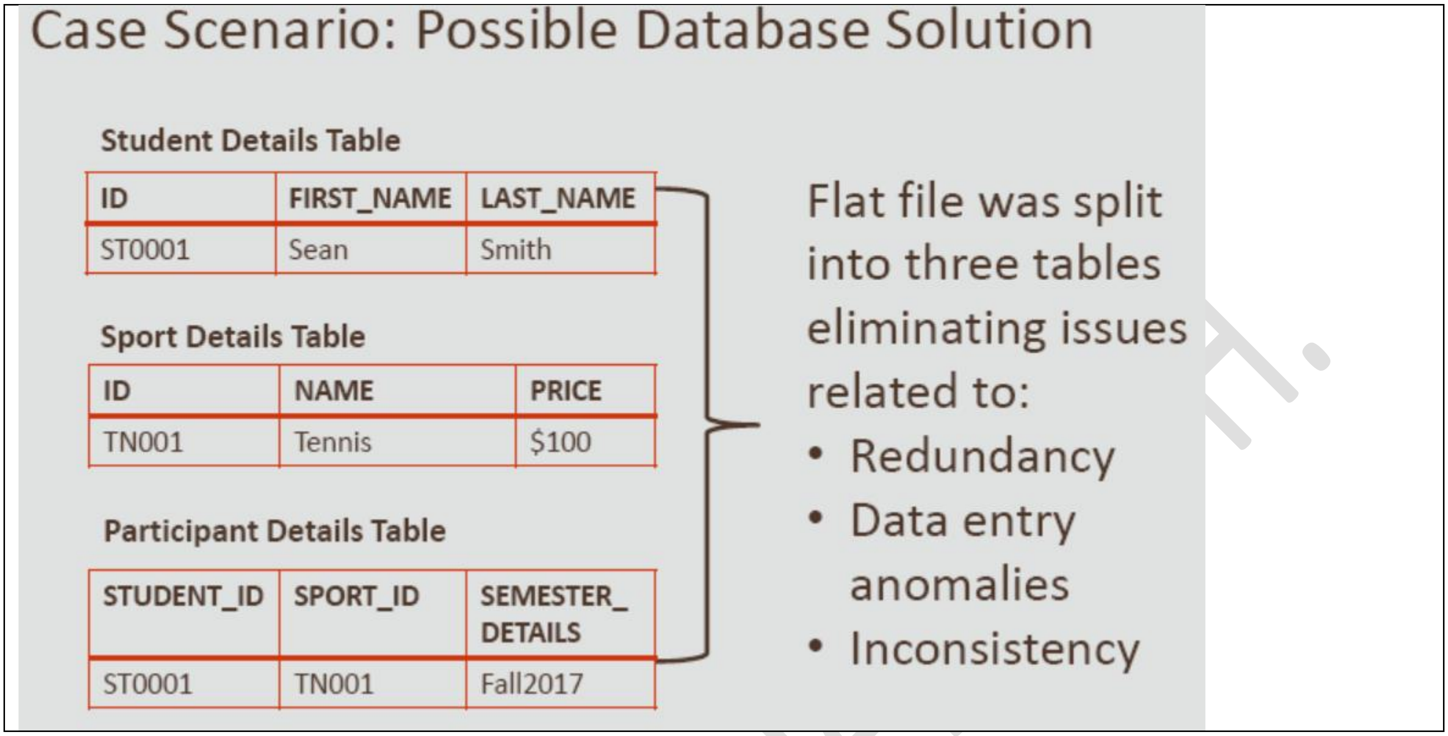
Primary Key

In this example a relationship is created between the two tables using the common field of DEPARTMENT_ID

ORACLE Academy

DFo 1-3

## 1.4. Business Requirements



Case Scenario: Possible Database Solution

Student Details Table

| ID | FIRST_NAME | LAST_NAME |
|---|---|---|
| ST0001 | Sean | Smith |

Sport Details Table

| ID | NAME | PRICE |
|---|---|---|
| TN001 | Tennis | $100 |

Participant Details Table

| STUDENT_ID | SPORT_ID | SEMESTER_ DETAILS |
|---|---|---|
| ST0001 | TN001 | Fall2017 |

Flat file was split into three tables eliminating issues related to:

- Redundancy
- Data entry anomalies
- Inconsistency

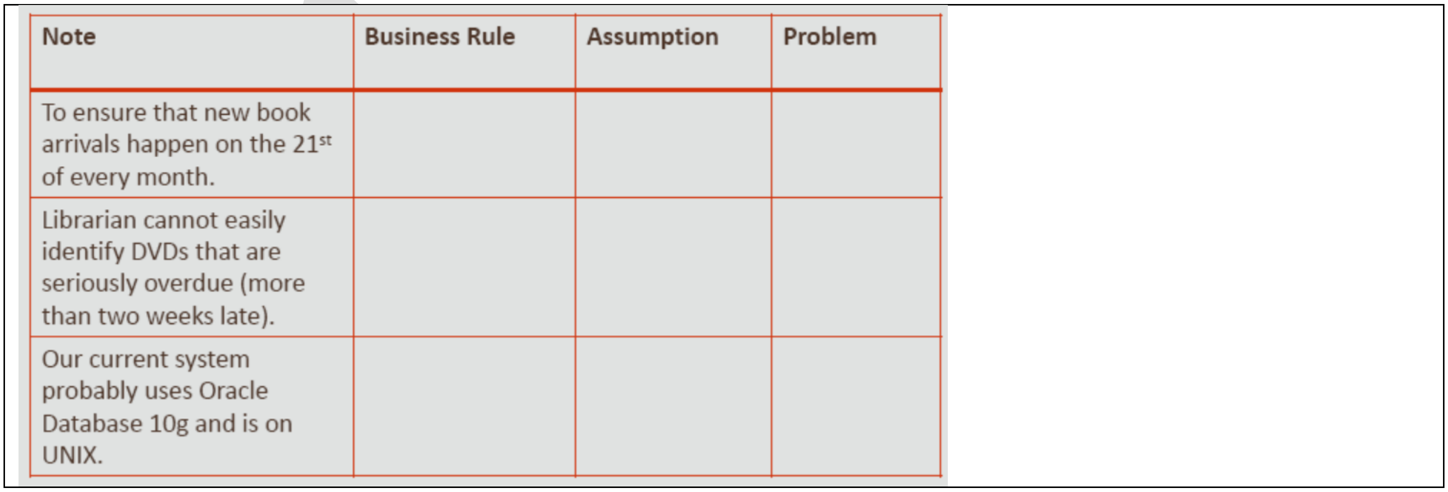
Importance of Business Rules

It is important to identify and document business rules when designing a database

Business rules:

- Allow the developer/architect to understand the relationship and constraints of the participating entities
- Help you understand the standardization procedure that an organization follows when handling huge data
- Should be simple and easy to understand
- Must be kept up-to-date

Example:

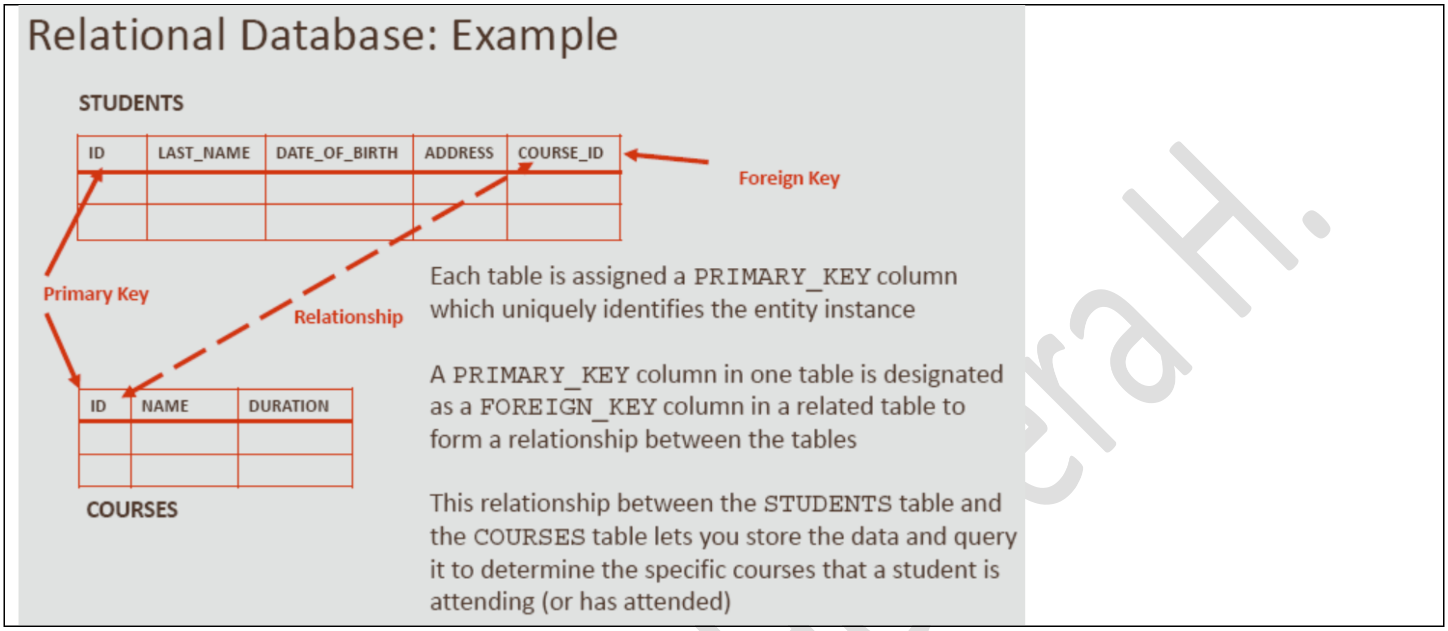


| Note | Business Rule | Assumption | Problem |
|---|---|---|---|
| To ensure that new book arrivals happen on the 21st of every month. | | | |
| Librarian cannot easily identify DVDs that are seriously overdue (more than two weeks late). | | | |
| Our current system probably uses Oracle Database 10g and is on UNIX. | | | |

→

# 2. Databases and Data Modeling

## 2.1. Relational Databases



### Relational Database: Example

STUDENTS

| ID | LAST_NAME | DATE_OF_BIRTH | ADDRESS | COURSE_ID |
|---|---|---|---|---|
|  |  |  |  |  |
|  |  |  |  |  |

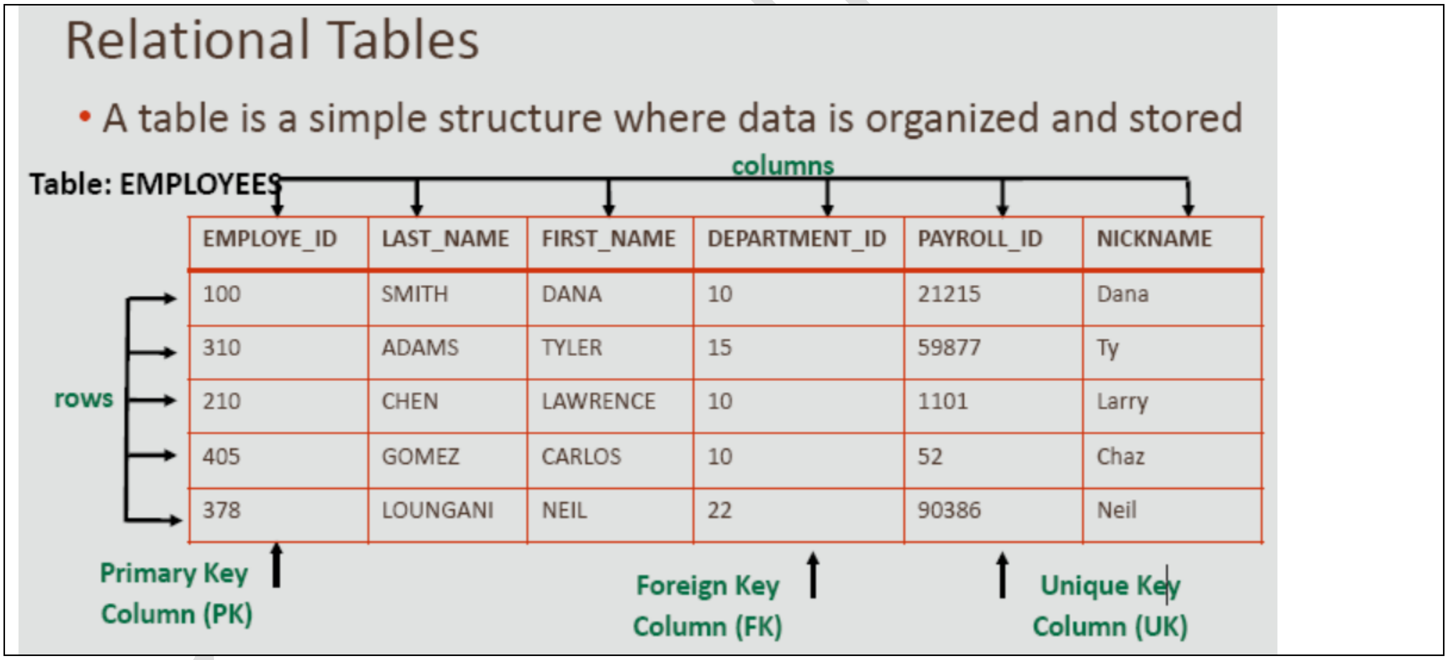
| ID | NAME | DURATION |
|---|---|---|
|  |  |  |
|  |  |  |

COURSES

Each table is assigned a PRIMARY_KEY column which uniquely identifies the entity instance

A PRIMARY_KEY column in one table is designated as a FOREIGN_KEY column in a related table to form a relationship between the tables

This relationship between the STUDENTS table and the COURSES table lets you store the data and query it to determine the specific courses that a student is attending (or has attended)



### Relational Tables

- A table is a simple structure where data is organized and stored

Table: EMPLOYEES

| EMPLOYE_ID | LAST_NAME | FIRST_NAME | DEPARTMENT_ID | PAYROLL_ID | NICKNAME |
|---|---|---|---|---|---|
| 100 | SMITH | DANA | 10 | 21215 | Dana |
| 310 | ADAMS | TYLER | 15 | 59877 | Ty |
| 210 | CHEN | LAWRENCE | 10 | 1101 | Larry |
| 405 | GOMEZ | CARLOS | 10 | 52 | Chaz |
| 378 | LOUNGANI | NEIL | 22 | 90386 | Neil |

## Rules for Relational Database Tables

- Each table has a distinct name
- Each table may contain multiple rows
- Each table has a value to uniquely identify the rows
- Each column in a table has a unique name
- Entries in columns are single values
- Entries in columns are of the same kind
- Order of rows and columns is insignificant

## Key Terms

**Table** –A basic storage structure
**Column**–attribute that describes the information in the table
**Primary Key** –the unique identifier for each row
**Foreign Key** –a column that refers to a primary key column in another table
**Row**–data for one table instance
**Field** –the one value found at the intersection of a row and column

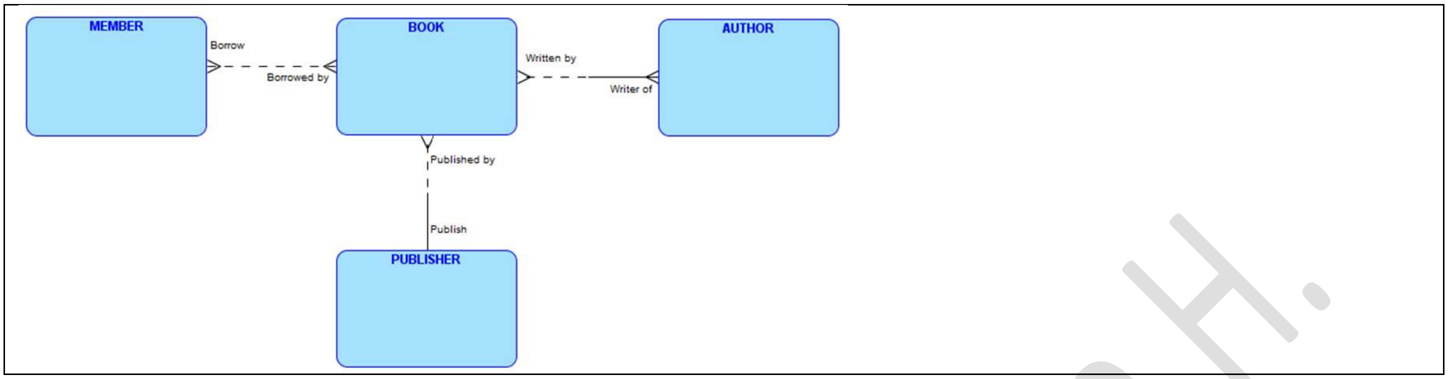
**Modeling Performed** :

| | | |
|---|---|---|
| Entities | -> | Tables |
| Attributes | -> | Columns |
| Relationships | -> | Foreign keys |

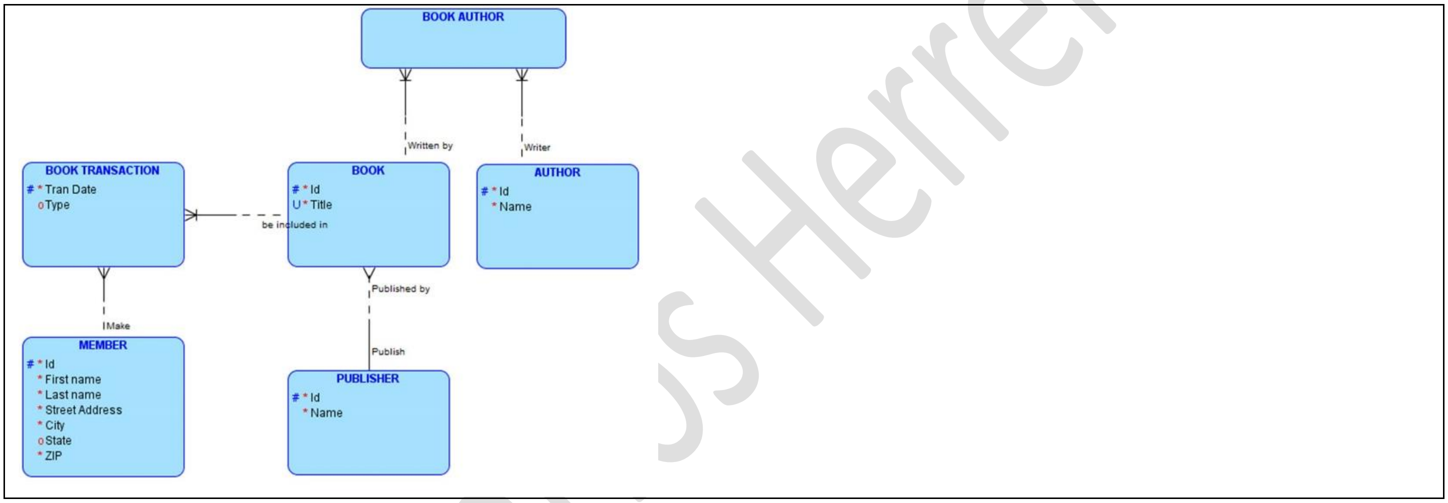
→

## 2.2. Conceptual and Physical Data Models

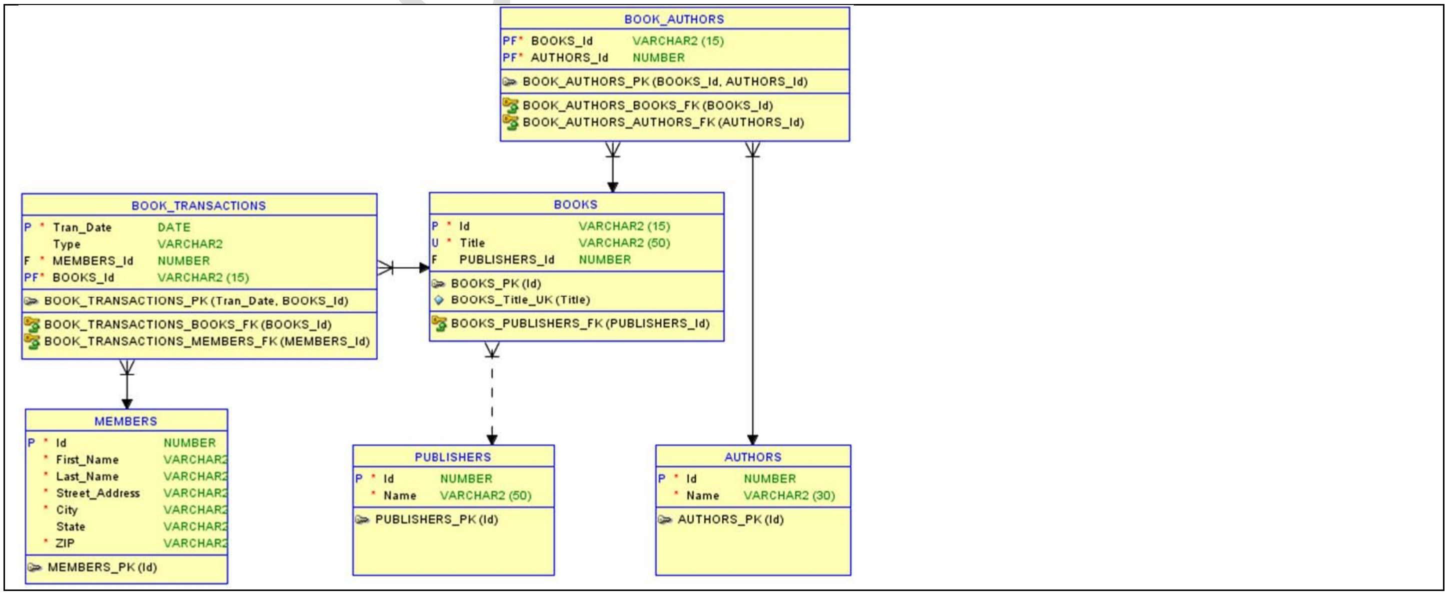
Conceptual Model



Logical Model



Physical Model: Entities -> Relationships -> Attributes -> Constrains

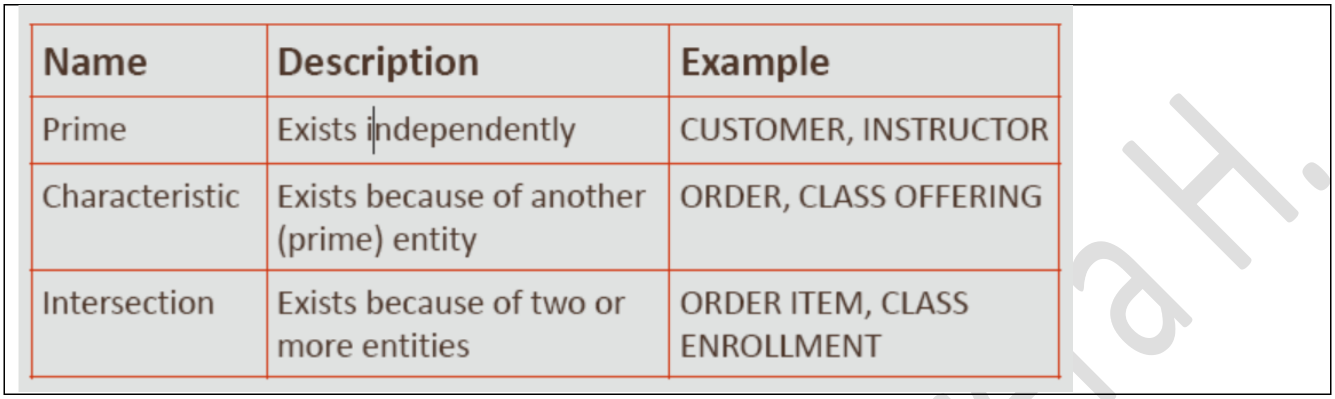


## 2.3. Entities and Attributes

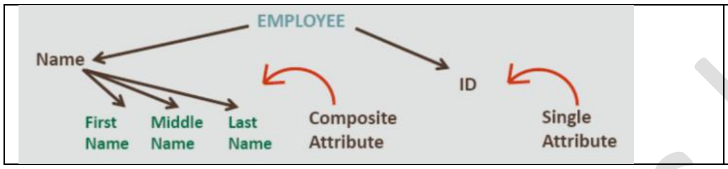
Identify mandatory(*), optional(o), volatile(age), and nonvolatile(birthDate) attributes

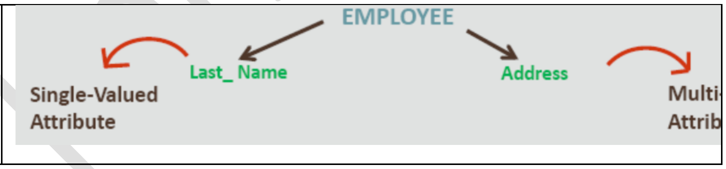
Entity Types

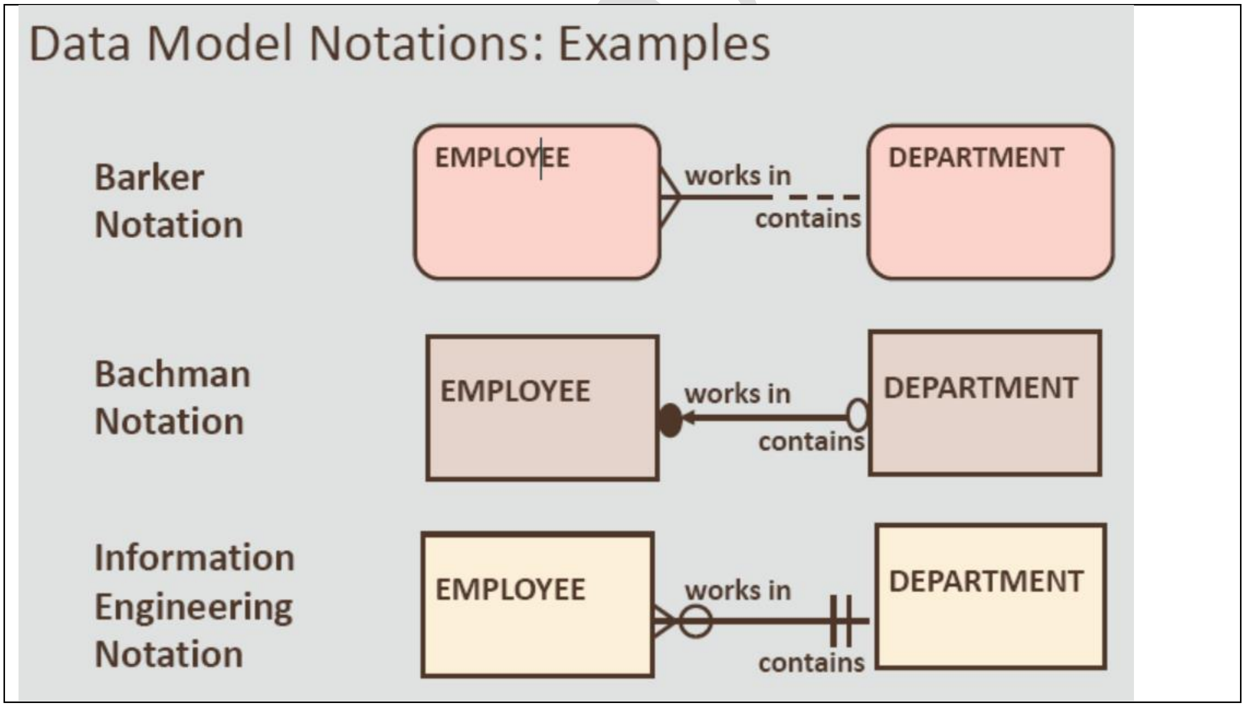
An entity can be classified as one of the following types:



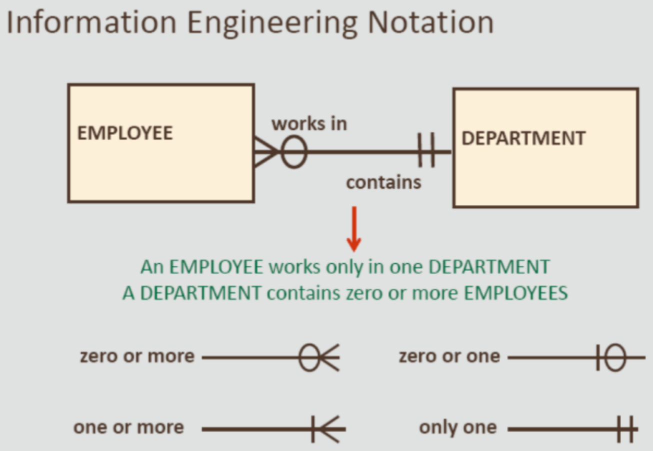
| Name | Description | Example |
|---|---|---|
| Prime | Exists independently | CUSTOMER, INSTRUCTOR |
| Characteristic | Exists because of another (prime) entity | ORDER, CLASS OFFERING |
| Intersection | Exists because of two or more entities | ORDER ITEM, CLASS ENROLLMENT |





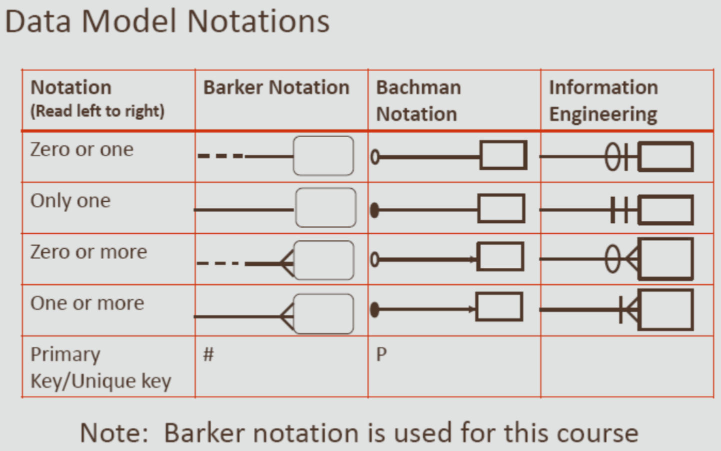


Data Model Notations: Examples



Information Engineering Notation

An EMPLOYEE works only in one DEPARTMENT
A DEPARTMENT contains zero or more EMPLOYEES

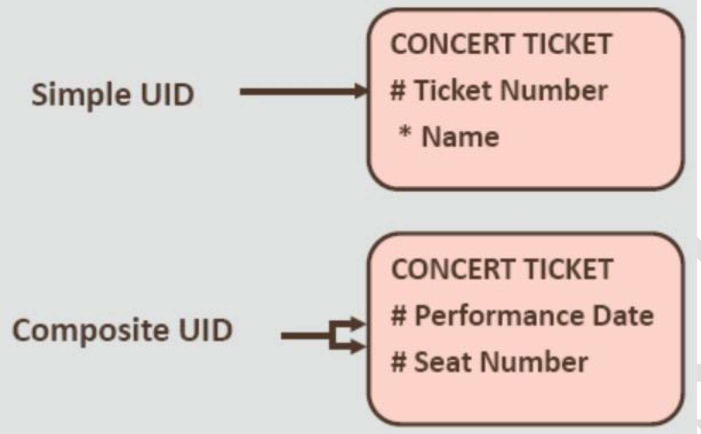


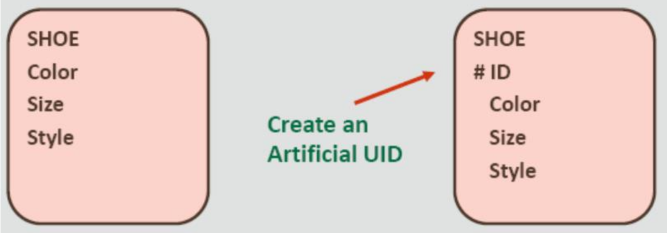
Data Model Notations

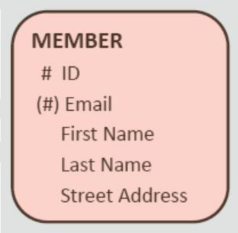
| Notation<br>(Read left to right) | Barker Notation | Bachman Notation | Information Engineering |
|---|---|---|---|
| Zero or one | | | |
| Only one | | | |
| Zero or more | | | |
| One or more | | | |
| Primary Key/Unique key | # | P | |

Note: Barker notation is used for this course

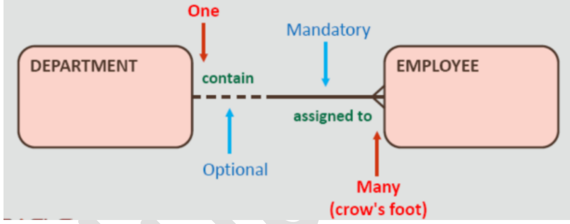
## 2.4. Unique Identifiers



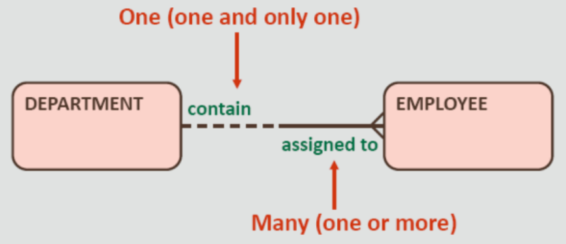


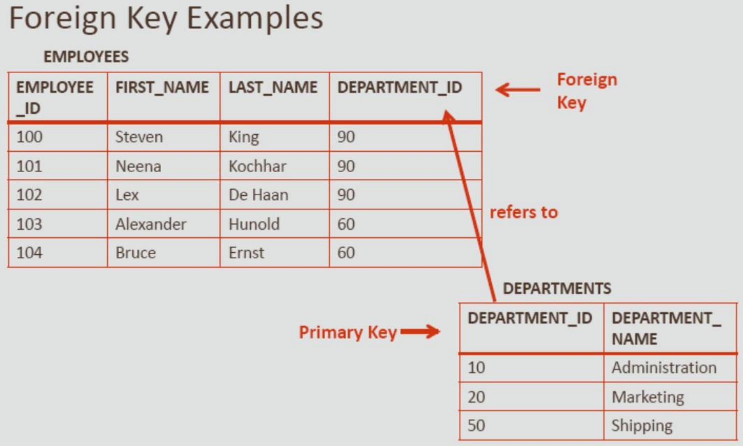


## 2.5. Relationships



Cardinality: 1 : M





Foreign Key Examples

EMPLOYEES

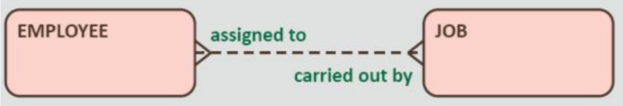
| EMPLOYEE _ID | FIRST_NAME | LAST_NAME | DEPARTMENT_ID |
|---|---|---|---|
| 100 | Steven | King | 90 |
| 101 | Neena | Kochhar | 90 |
| 102 | Lex | De Haan | 90 |
| 103 | Alexander | Hunold | 60 |
| 104 | Bruce | Ernst | 60 |

Foreign Key
refers to
Primary Key

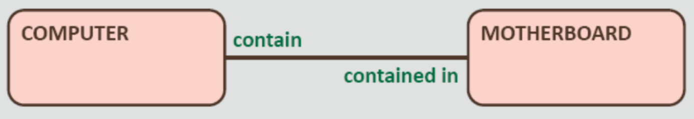
DEPARTMENTS

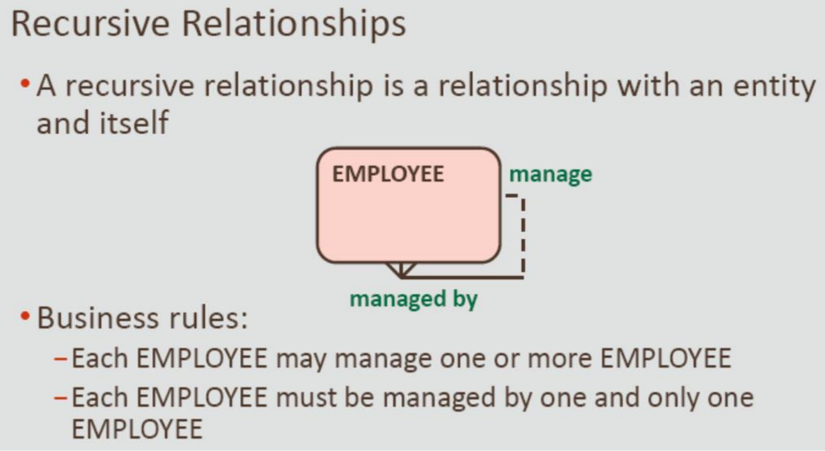
| DEPARTMENT_ID | DEPARTMENT_ NAME |
|---|---|
| 10 | Administration |
| 20 | Marketing |
| 50 | Shipping |

Cardinality  M : N



Cardinality  1 : 1



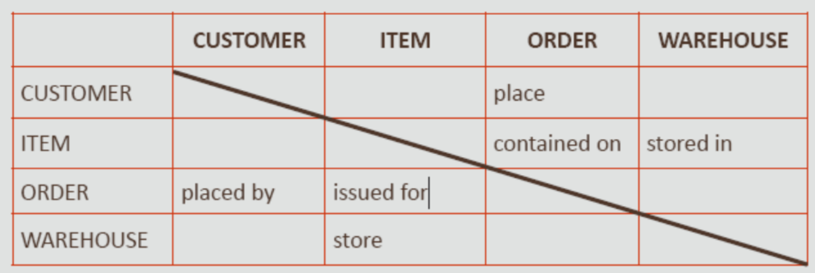


## Recursive Relationships

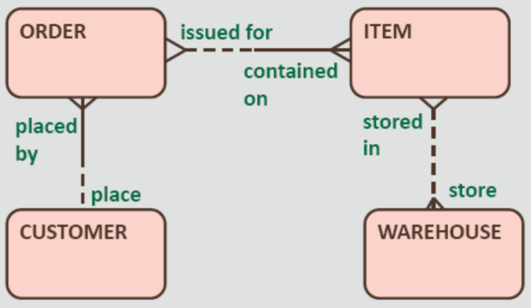
- A recursive relationship is a relationship with an entity and itself

- Business rules:
  - Each EMPLOYEE may manage one or more EMPLOYEE
  - Each EMPLOYEE must be managed by one and only one EMPLOYEE

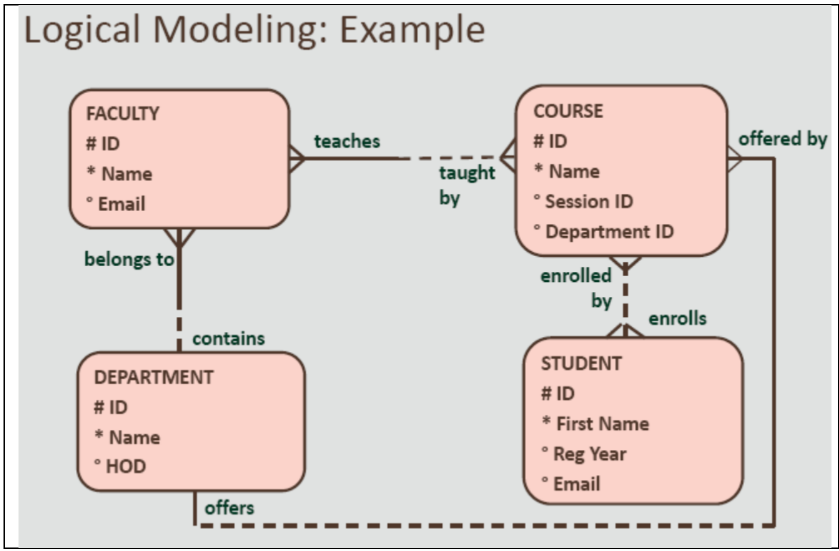
Relationship Matrix: Mapping the Contents



| | CUSTOMER | ITEM | ORDER | WAREHOUSE |
|---|---|---|---|---|
| CUSTOMER | | | place | |
| ITEM | | | contained on | stored in |
| ORDER | placed by | issued for | | |
| WAREHOUSE | | store | | |



## 2.6. Entity Relationship Modeling (ERDs)



Logical Modeling: Example

FACULTY
# ID
* Name
° Email

teaches

taught by

COURSE
# ID
* Name
° Session ID
° Department ID

offered by

belongs to

contains

DEPARTMENT
# ID
* Name
° HOD

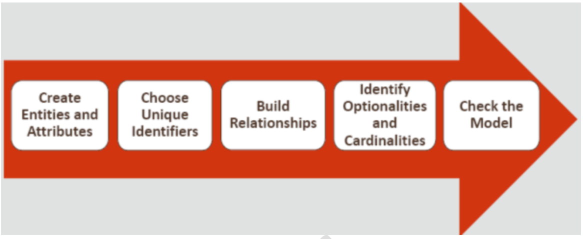
offers

enrolled by

enrolls

STUDENT
# ID
* First Name
° Reg Year
° Email

Steps to Build an ERD



Create Entities and Attributes

Choose Unique Identifiers

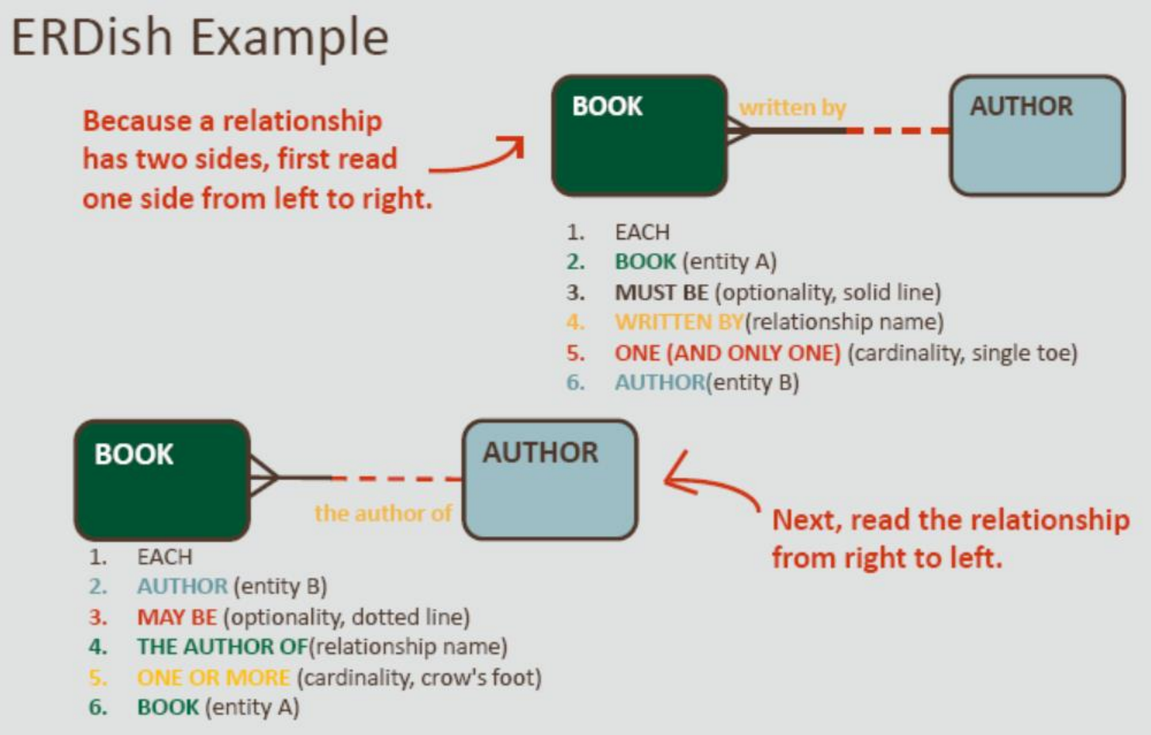
Build Relationships

Identify Optionalities and Cardinalities

Check the Model

**Components of ERDish**

- EACH
- Entity A
- OPTIONALITY (must be/may be)
- RELATIONSHIP NAME
- CARDINALITY (one and only one/ one or more)
- Entity B



ERDish Example

Because a relationship has two sides, first read one side from left to right.

BOOK

written by

AUTHOR

1. EACH
2. BOOK (entity A)
3. MUST BE (optionality, solid line)
4. WRITTEN BY(relationship name)
5. ONE (AND ONLY ONE) (cardinality, single toe)
6. AUTHOR(entity B)

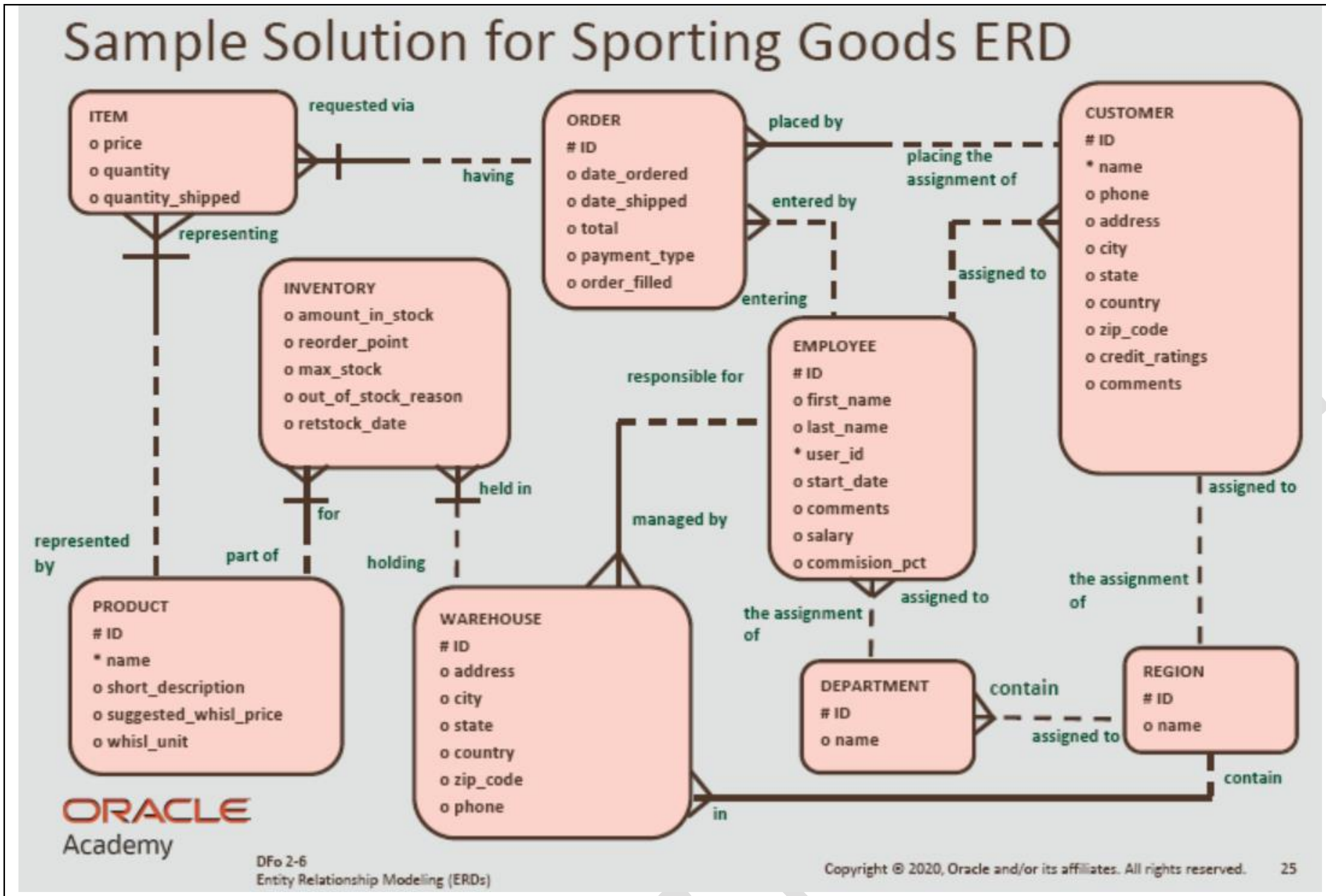
BOOK

the author of

AUTHOR

Next, read the relationship from right to left.

1. EACH
2. AUTHOR (entity B)
3. MAY BE (optionality, dotted line)
4. THE AUTHOR OF(relationship name)
5. ONE OR MORE (cardinality, crow's foot)
6. BOOK (entity A)



## Sample Solution for Sporting Goods ERD

ITEM
o price
o quantity
o quantity_shipped

ORDER
# ID
o date_ordered
o date_shipped
o total
o payment_type
o order_filled

CUSTOMER
# ID
* name
o phone
o address
o city
o state
o country
o zip_code
o credit_ratings
o comments

INVENTORY
o amount_in_stock
o reorder_point
o max_stock
o out_of_stock_reason
o retstock_date

EMPLOYEE
# ID
o first_name
o last_name
* user_id
o start_date
o comments
o salary
o commision_pct

PRODUCT
# ID
* name
o short_description
o suggested_whisl_price
o whisl_unit

WAREHOUSE
# ID
o address
o city
o state
o country
o zip_code
o phone

DEPARTMENT
# ID
o name

REGION
# ID
o name

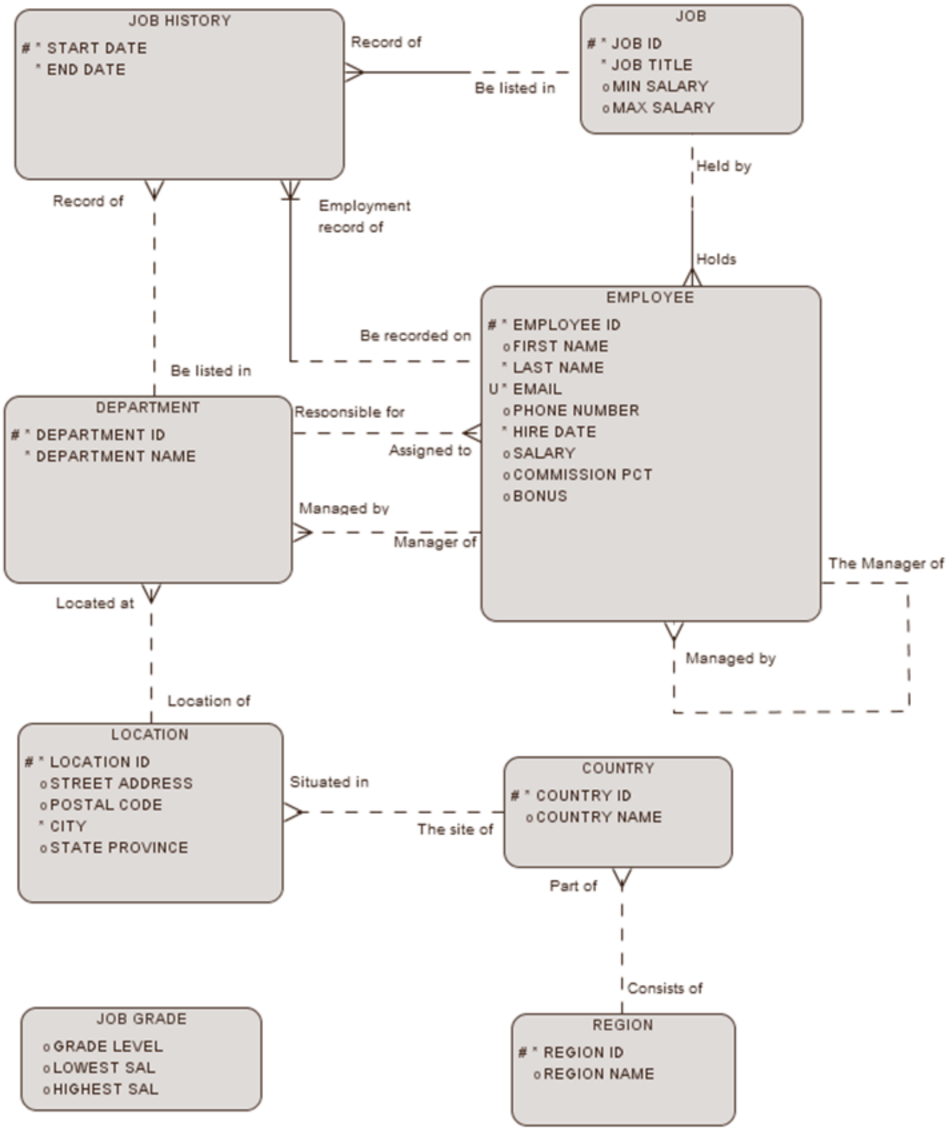
ORACLE Academy

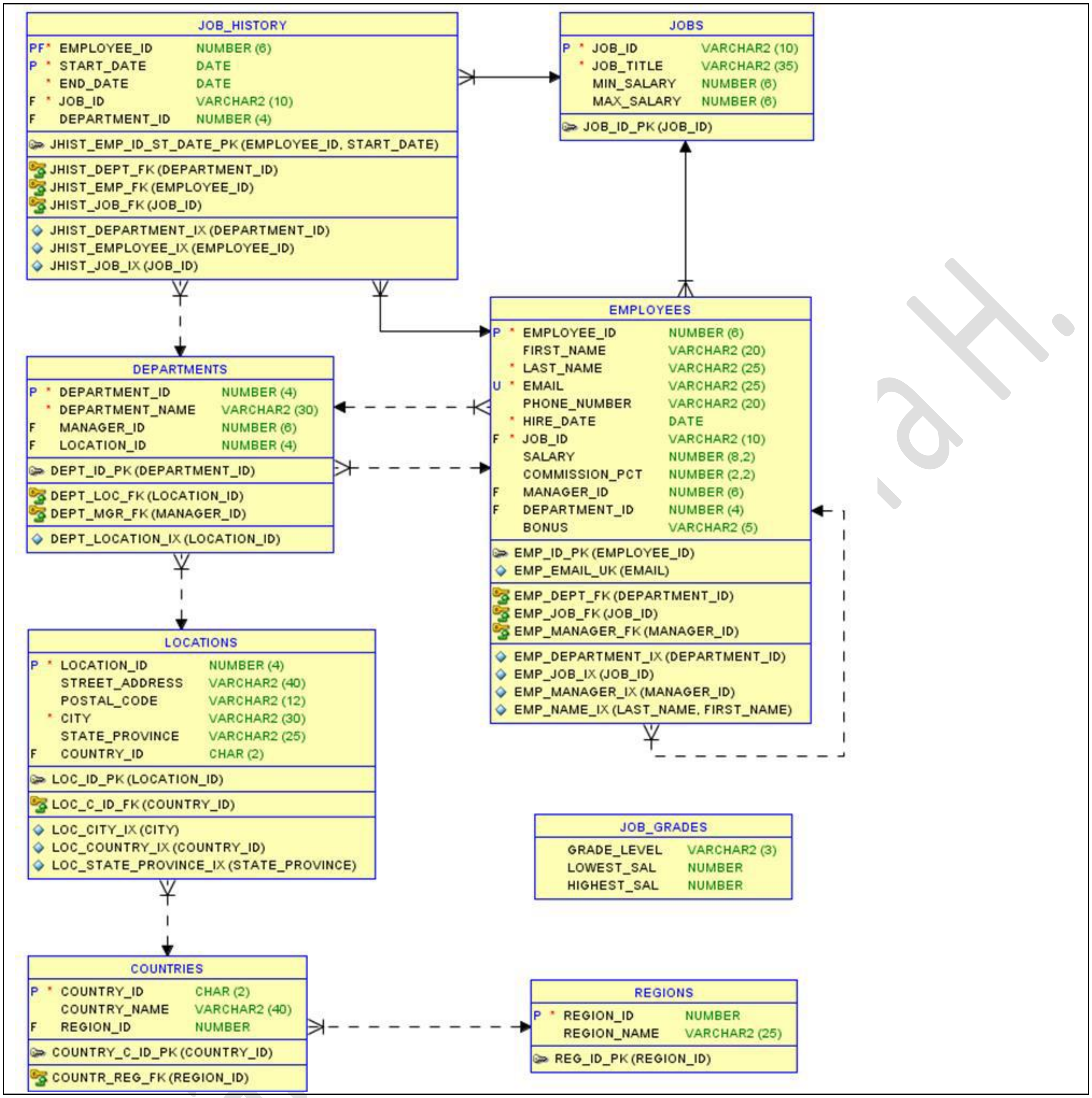
DFo 2-6
Entity Relationship Modeling (ERDs)

Copyright © 2020, Oracle and/or its affiliates. All rights reserved.

25

→


JOB HISTORY
# * START DATE
* END DATE
Record of
Be listed in
JOB
# * JOB ID
* JOB TITLE
o MIN SALARY
o MAX SALARY
Held by
Holds
Record of
Employment record of
Be recorded on
Be listed in
EMPLOYEE
# * EMPLOYEE ID
o FIRST NAME
* LAST NAME
U * EMAIL
o PHONE NUMBER
* HIRE DATE
o SALARY
o COMMISSION PCT
o BONUS
DEPARTMENT
# * DEPARTMENT ID
* DEPARTMENT NAME
Responsible for
Assigned to
Managed by
Manager of
The Manager of
Managed by
Located at
Location of
LOCATION
# * LOCATION ID
o STREET ADDRESS
o POSTAL CODE
* CITY
o STATE PROVINCE
Situated in
The site of
COUNTRY
# * COUNTRY ID
o COUNTRY NAME
Part of
Consists of
REGION
# * REGION ID
o REGION NAME
JOB GRADE
o GRADE LEVEL
o LOWEST SAL
o HIGHEST SAL


JOB_HISTORY
PF * EMPLOYEE_ID NUMBER (6)
P * START_DATE DATE
* END_DATE DATE
F * JOB_ID VARCHAR2 (10)
F DEPARTMENT_ID NUMBER (4)
JHIST_EMP_ID_ST_DATE_PK (EMPLOYEE_ID, START_DATE)
JHIST_DEPT_FK (DEPARTMENT_ID)
JHIST_EMP_FK (EMPLOYEE_ID)
JHIST_JOB_FK (JOB_ID)
JHIST_DEPARTMENT_IX (DEPARTMENT_ID)
JHIST_EMPLOYEE_IX (EMPLOYEE_ID)
JHIST_JOB_IX (JOB_ID)
JOBS
P * JOB_ID VARCHAR2 (10)
* JOB_TITLE VARCHAR2 (35)
MIN_SALARY NUMBER (6)
MAX_SALARY NUMBER (6)
JOB_ID_PK (JOB_ID)
EMPLOYEES
P * EMPLOYEE_ID NUMBER (6)
FIRST_NAME VARCHAR2 (20)
* LAST_NAME VARCHAR2 (25)
U * EMAIL VARCHAR2 (25)
PHONE_NUMBER VARCHAR2 (20)
* HIRE_DATE DATE
F * JOB_ID VARCHAR2 (10)
SALARY NUMBER (8,2)
COMMISSION_PCT NUMBER (2,2)
F MANAGER_ID NUMBER (6)
F DEPARTMENT_ID NUMBER (4)
BONUS VARCHAR2 (5)
EMP_ID_PK (EMPLOYEE_ID)
EMP_EMAIL_UK (EMAIL)
EMP_DEPT_FK (DEPARTMENT_ID)
EMP_JOB_FK (JOB_ID)
EMP_MANAGER_FK (MANAGER_ID)
EMP_DEPARTMENT_IX (DEPARTMENT_ID)
EMP_JOB_IX (JOB_ID)
EMP_MANAGER_IX (MANAGER_ID)
EMP_NAME_IX (LAST_NAME, FIRST_NAME)
DEPARTMENTS
P * DEPARTMENT_ID NUMBER (4)
* DEPARTMENT_NAME VARCHAR2 (30)
F MANAGER_ID NUMBER (6)
F LOCATION_ID NUMBER (4)
DEPT_ID_PK (DEPARTMENT_ID)
DEPT_LOC_FK (LOCATION_ID)
DEPT_MGR_FK (MANAGER_ID)
DEPT_LOCATION_IX (LOCATION_ID)
LOCATIONS
P * LOCATION_ID NUMBER (4)
STREET_ADDRESS VARCHAR2 (40)
POSTAL_CODE VARCHAR2 (12)
* CITY VARCHAR2 (30)
STATE_PROVINCE VARCHAR2 (25)
F COUNTRY_ID CHAR (2)
LOC_ID_PK (LOCATION_ID)
LOC_C_ID_FK (COUNTRY_ID)
LOC_CITY_IX (CITY)
LOC_COUNTRY_IX (COUNTRY_ID)
LOC_STATE_PROVINCE_IX (STATE_PROVINCE)
JOB_GRADES
GRADE_LEVEL VARCHAR2 (3)
LOWEST_SAL NUMBER
HIGHEST_SAL NUMBER
COUNTRIES
P * COUNTRY_ID CHAR (2)
COUNTRY_NAME VARCHAR2 (40)
F REGION_ID NUMBER
COUNTRY_C_ID_PK (COUNTRY_ID)
COUNTR_REG_FK (REGION_ID)
REGIONS
P * REGION_ID NUMBER
REGION_NAME VARCHAR2 (25)
REG_ID_PK (REGION_ID)

→

# 3. Refining the Data Model

## 3.1. More with Relationships

## 3.2. Tracking Data Changes

## 3.3. Normalization and Business Rules

## 3.4. Data Modeling Terminology and Mapping

→

# 4. Oracle SQL Developer Data Modeler

## 4.1. Oracle SQL Developer Data Modeler

## 4.2. Convert a Logical Model to a Relational Model

→

# 5. Mapping to the Physical Model

## 5.1. Mapping Entities and Attributes

## 5.2. Mapping Primary and Foreign Keys

→

# 6. Introduction to SQL

## 6.1. Introduction to Oracle Application Express

## 6.2. Structured Query Language (SQL)

## 6.3. Data Definition Language (DDL)

## 6.4. Data Manipulation Language (DML)

## 6.5. Transaction Control Language (TCL)

## 6.6. Retrieving Data Using SELECT

## 6.7. Restricting Data Using WHERE

## 6.8. Sorting Data Using ORDER BY

## 6.9. Joining Tables Using JOIN

→