



DATABASE FOUNDATIONS

ORACLE ACADEMY



6 DE MAYO DE 2025

<https://academy.oracle.com/>

[HTTPS://GITHUB.COM/ISC-UPA/2025-2-ISC05-DB](https://github.com/ISC-UPA/2025-2-ISC05-DB)

Contenido

1.	Introduction.....	2
1.1.	Introduction.....	3
1.2.	Introduction to Databases.....	4
1.3.	Types of Database Models	6
1.4.	Business Requirements	7
2.	Databases and Data Modeling	8
2.1.	Relational Databases	8
2.2.	Conceptual and Physical Data Models	10
2.3.	Entities and Attributes.....	11
2.4.	Unique Identifiers.....	12
2.5.	Relationships \$\$.....	12
2.6.	Entity Relationship Modeling (ERDs).....	14
3.	Refining the Data Model	18
3.1.	More with Relationships.....	18
3.2.	Tracking Data Changes	18
3.3.	Normalization and Business Rules.....	18
3.4.	Data Modeling Terminology and Mapping.....	18
4.	Oracle SQL Developer Data Modeler	19
4.1.	Oracle SQL Developer Data Modeler.....	19
4.2.	Convert a Logical Model to a Relational Model	19
5.	Mapping to the Physical Model.....	20
5.1.	Mapping Entities and Attributes	20
5.2.	Mapping Primary and Foreign Keys.....	20
6.	Introduction to SQL.....	21
6.1.	Introduction to Oracle Application Express.....	21
6.2.	Structured Query Language (SQL)	21
6.3.	Data Definition Language (DDL)	21
6.4.	Data Manipulation Language (DML).....	21
6.5.	Transaction Control Language (TCL).....	21
6.6.	Retrieving Data Using SELECT	21
6.7.	Restricting Data Using WHERE	21
6.8.	Sorting Data Using ORDER BY	21
6.9.	Joining Tables Using JOIN	21

1. Introduction

Directrices para Fundamentos de bases de datos							
Duración de la sesión: 45 minutos							
	Sesión 1	Sesión 2	Sesión 3	Sesión 4	Sesión 5		
Semana 1	Introducción						
Semana 2							
Semana 3	Bases de datos y modelado de datos						
Semana 4							
Semana 5							
Semana 6	Acotación del modelo de datos						
Semana 7							
Semana 8							
Semana 9	Revisión del examen de mitad de trimestre			Examen de mitad de trimestre			
Semana 10	Oracle SQL Developer Data Modeler						
Semana 11							
Semana 12	Asignación al modelo físico						
Semana 13	Introducción a SQL						
Semana 14							
Semana 15							
Semana 16							
Semana 17							
Semana 18	Revisión del examen final			Examen final			

Database Foundations – Español : Dfo 1-1: Introducción al curso (Diapositivas de la lección)

Home / My Classes / Taking a Class / Class Course Lesson

Course Progress Save and Continue

PowerPoint Presentation 3 / 13 67%

Hoja de ruta

Y es importante que los alumnos pulsen ese botón para registrar sus progresos.

Introducción al curso

Introducción a las bases de datos

Tipos de modelos de bases de datos

Requisitos de negocio

Introducción a SQL

Introducción a Oracle SQL Developer Data Modeler

Asignación al modelo físico

Revisión del examen de mitad de trimestre

Examen de mitad de trimestre

Revisión del examen final

Examen final

Database Foundations – Español

Home / My Classes / Taking a Class

Sections in Course

Difficulty

Status in Progress

En este curso, los estudiantes con poca experiencia en database aprenden técnicas de diseño de bases de datos, diseñar bases de datos con una herramienta de modelado y reciben una introducción a SQL para implementar y realizar consultas en bases de datos por medio de actividades prácticas y estimulantes.

Course Outline

Sección 0 - Recursos del curso

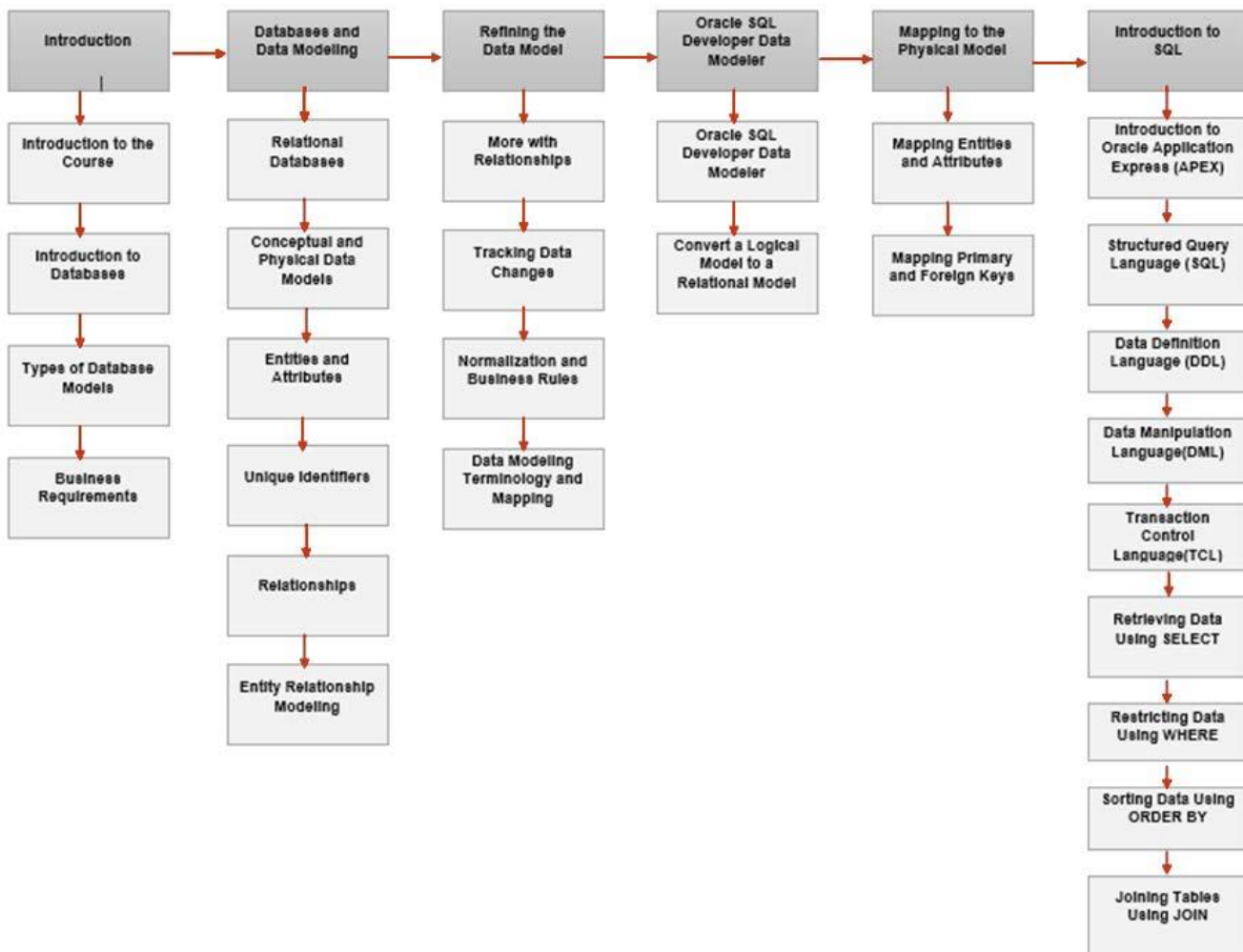
Sección 1 - Introducción

Sección 2 - Bases de datos y modelado de datos

Sección 3 - Acotación del modelo de datos

→

1.1. Introduction



Technological Requirements:

Oracle SQL Developer or Oracle APEX application
Oracle Data Modeler

→

1.2. Introduction to Databases

Data vs Information.

Data: Collected facts about a topic or item

Information: The result of combining, comparing, and performing calculations on data.

Introduction to Relational Databases

- 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 record is a collection of fields
- A column is referred to as a field (or attribute)

Relational Database Example

Order Detail Table

ID	DETAILS	CUSTOMER_ID

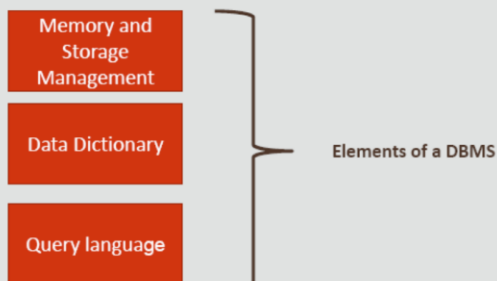
Customer Table

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



Key Computing Terms

Hardware: physical parts of a computer

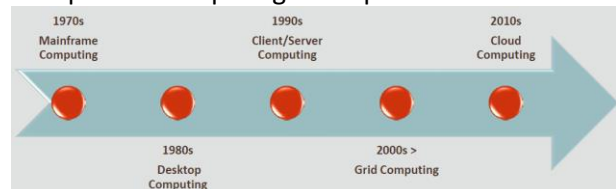
Software: instructions to tell hardware what to do

Operating system : software that directly controls the hardware

Application: performs specific task

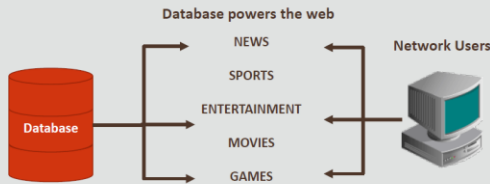
Client : workstation used by end users

Server : accepts work requiring more power from clients



2000s: Grid Computing (Shared Processing)

- In the grid-computing model, all of an organization's computers in different locations can be utilized just like a pool of computing resources
- Grid computing builds a software infrastructure that can run on a large number of networked servers
- A user makes a request for information or computation from his or her workstation and that request is processed somewhere in the grid as efficiently as possible

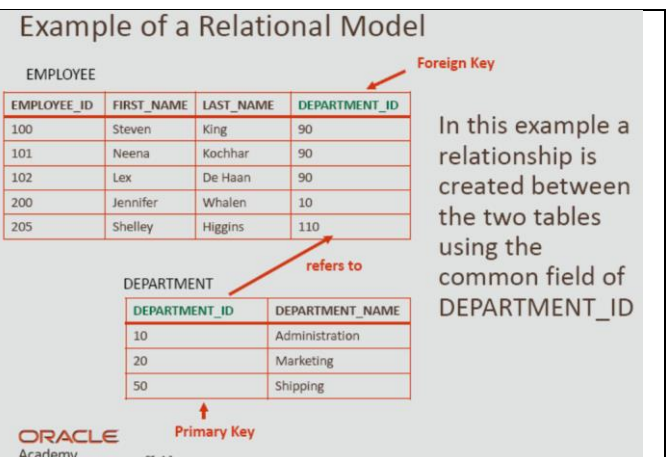
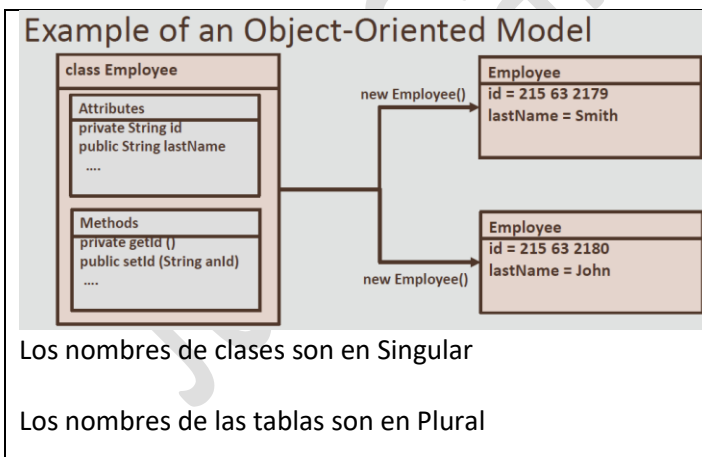
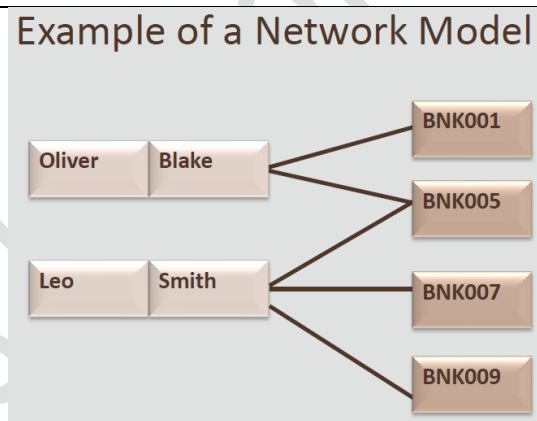
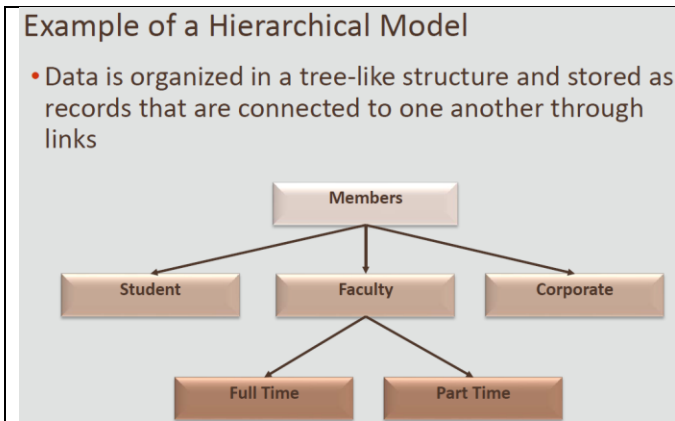
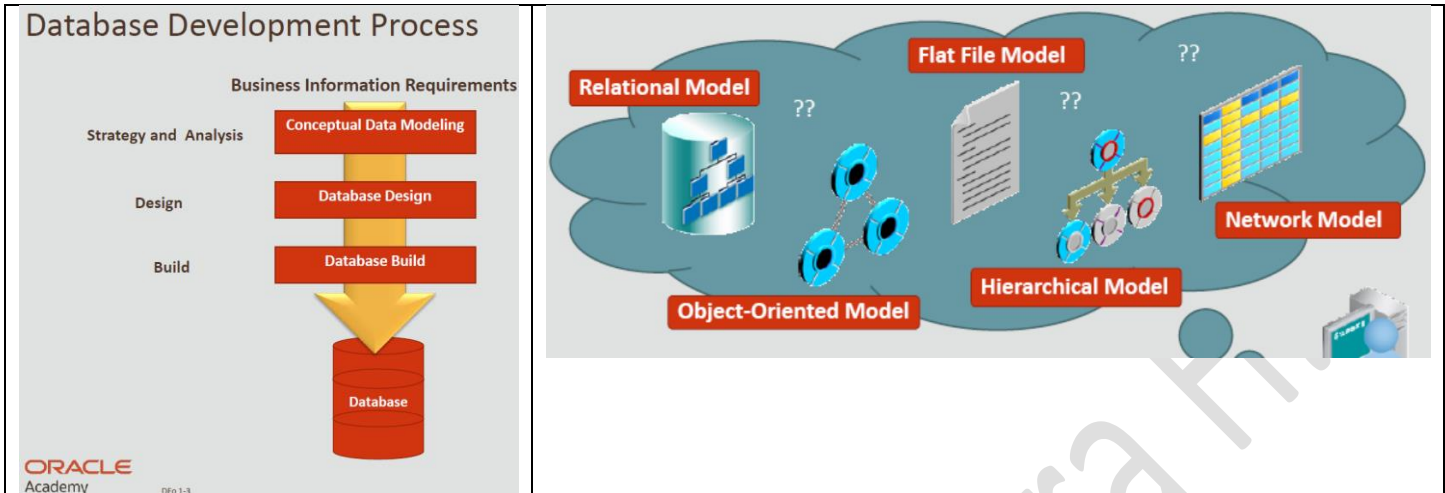


2010s: Cloud Computing (Internet Based Processing)

- Databases run from the web (as a service)
-
- The diagram illustrates the Cloud Computing model. On the left, a blue cloud contains three red cylinders, each labeled 'Database'. To the right of the cloud is a computer icon labeled 'Web Users'. Between the cloud and the user, three horizontal lines represent different service levels: 'INFRASTRUCTURE (IaaS)', 'PLATFORM (PaaS)', and 'SOFTWARE (SaaS)'. Arrows point from the cloud to each of these lines, and from each line to the 'Web Users' icon.
- Cloud computing allows the delivery of computing services over the Internet
 - The three main categories of cloud services are:
 - IaaS – Allows you to rent cloud based servers, storage, operating systems etc
 - PaaS – Gives access to an online environment for developing and testing software without any setup or management costs
 - SaaS – Delivers software direct from the Internet. Users normally access it through a web browser

→

1.3. Types of Database Models



→

1.4. Business Requirements

Case Scenario: Need a Database Solution

	STUDENT_ID	SPORT_1	PRICE_1	SPORT_2	PRICE_2
Record 1	ST0001	Tennis	\$100	Badminton	\$150
Record 2	ST0002	Soccer	\$175	Tennis	\$100
Record 3	ST0003	Cycling	\$200	Badminton	\$150
.....

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

Note: Not all business rules can be modeled in a database, but must be documented

Case Scenario: Identifying Key Business Rules, Problems, and Assumptions

- Business rule: Used to understand business processes and the nature, role, and scope of the data
- Assumption: Can be defined as a fact or a statement that has been taken for granted
- Problem: Can be defined as a situation or scenario that requires attention and a possible solution to alleviate the situation

Example:

Note	Business Rule	Assumption	Problem
To ensure that new book arrivals happen on the 21 st 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.			

Identify the statements as a business rule, a problem, or an assumption.



2. Databases and Data Modeling

2.1. Relational Databases

Relational Database: Example

STUDENTS

ID	LAST_NAME	DATE_OF_BIRTH	ADDRESS	COURSE_ID

Foreign Key

Primary Key

Relationship

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

ID	NAME	DURATION

COURSES

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

columns

EMPLOYEE_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

rows

Primary Key
Column (PK)

Foreign Key
Column (FK)

Unique Key
Column (UK)

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

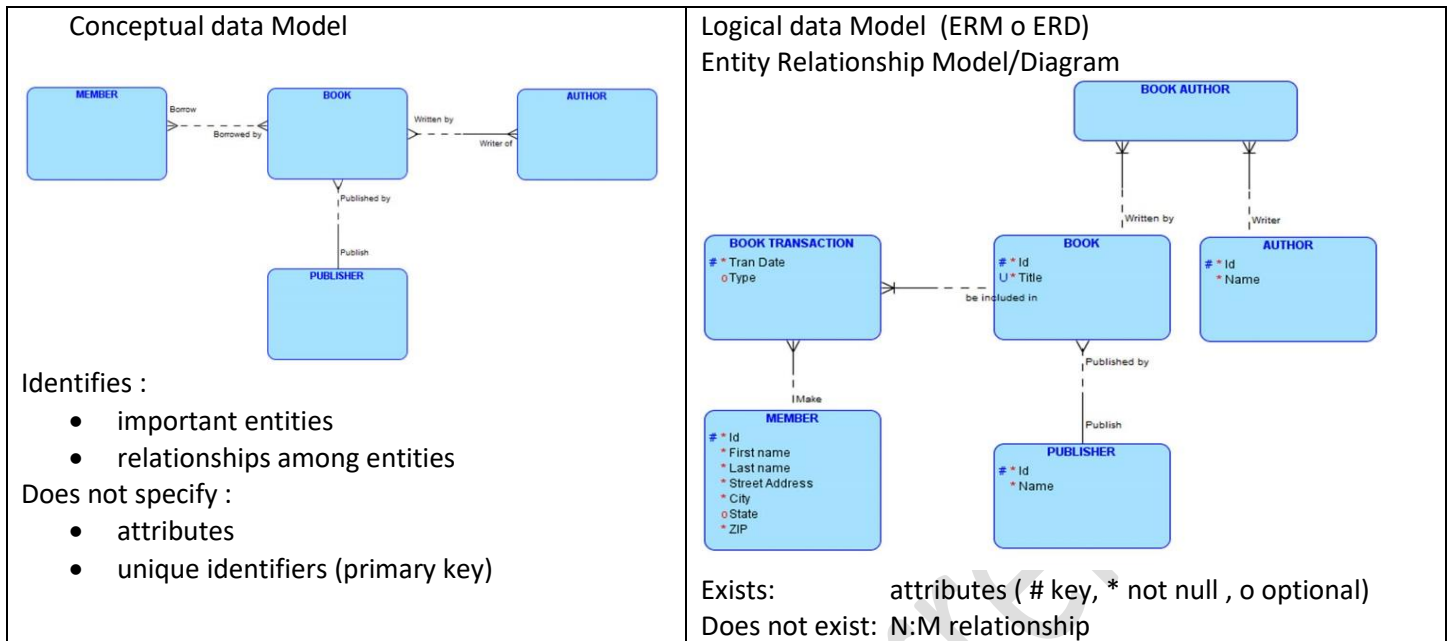
Foreign Key –a column that refers to a primary key column in another table

Row–data for one table instance

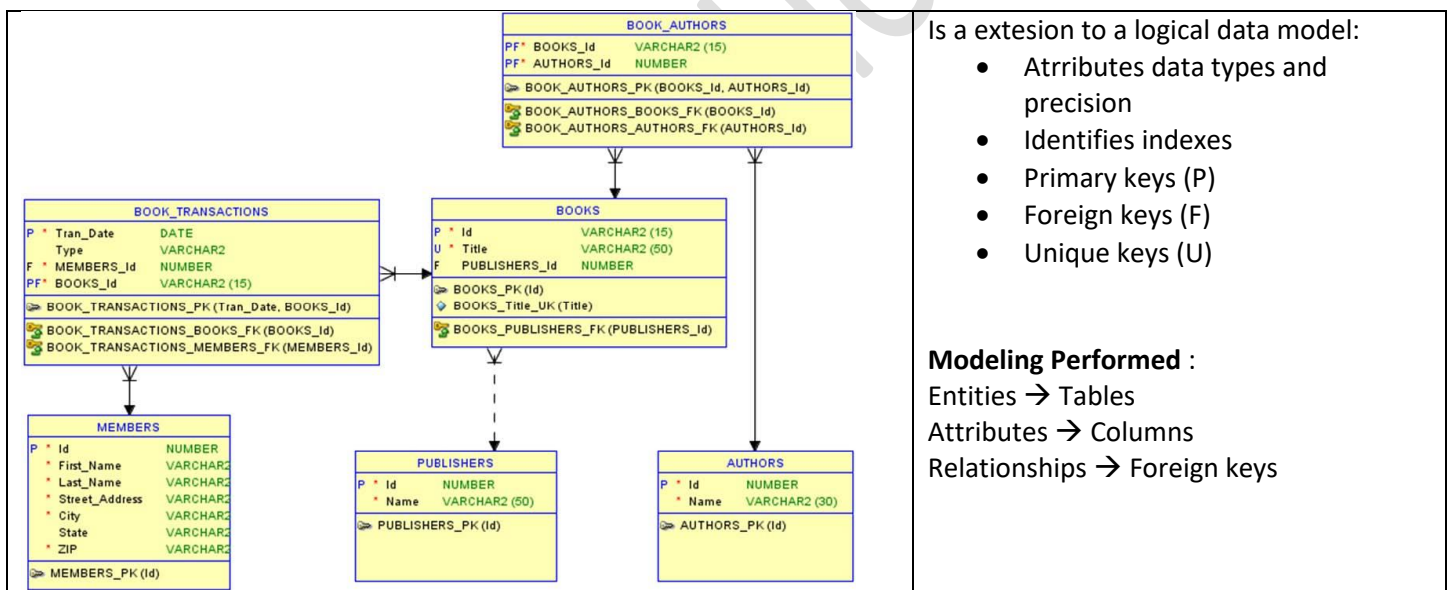
Field –the one value found at the intersection of a row and column

→

2.2. Conceptual and Physical Data Models



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



→

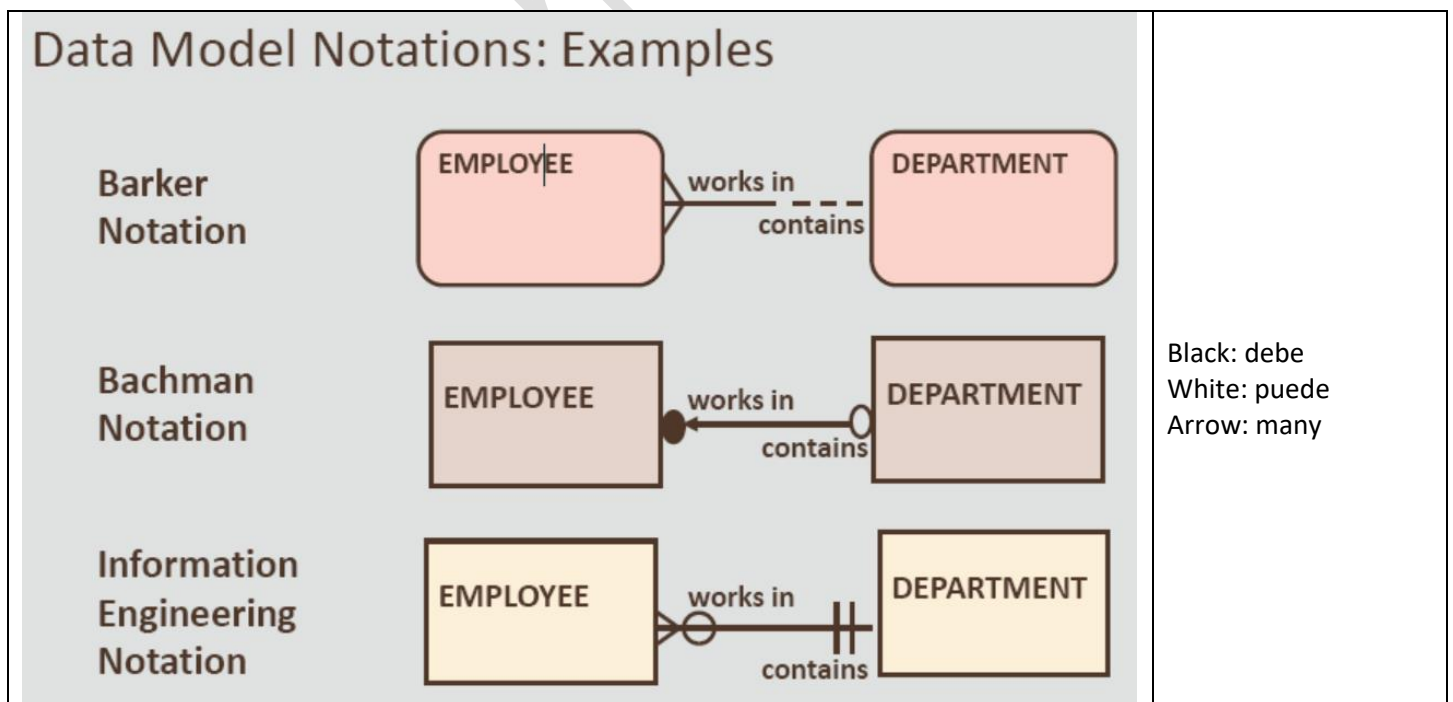
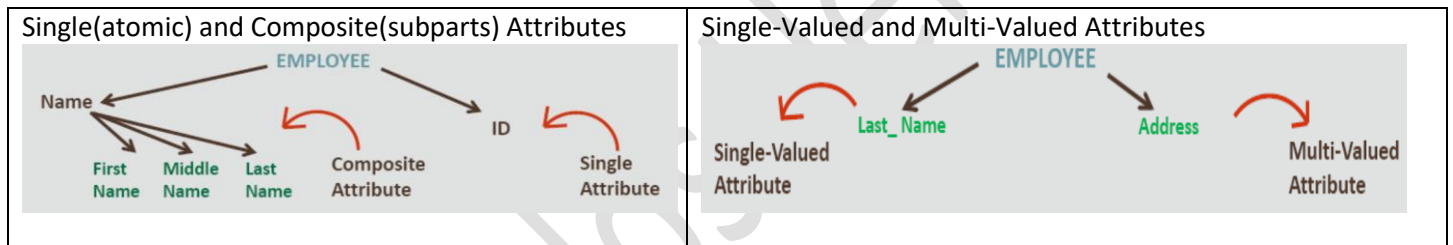
2.3. Entities and Attributes

Identify UID(#), mandatory(*), optional(o), volatile or derivate(age), and nonvolatile(birthDate) attributes

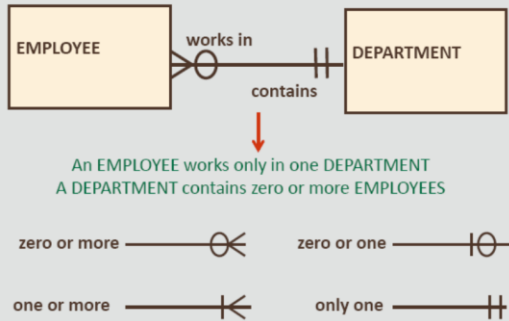
Entity Types

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

Name	Description	Example	Tipos de Entidad: Principal Característica Interseccion
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	Entidades: Fuertes Débiles



Information Engineering Notation



Data Model Notations

Notation (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

Simple UID

CONCERT TICKET
Ticket Number
* Name

Composite UID

CONCERT TICKET
Performance Date
Seat Number

Each entity must have a unique identifier
Otherwise, it is not an entity

SHOE
Color
Size
Style

Create an
Artificial UID

SHOE
ID
Color
Size
Style

MEMBER

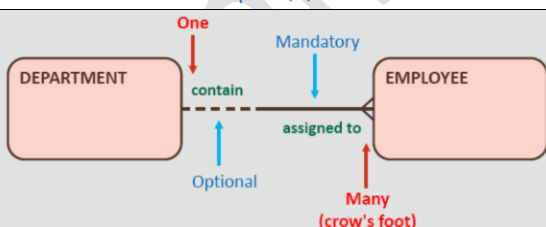
ID
(#) Email
First Name
Last Name
Street Address

Candidate unique identifiers or
secondary UIDs, so define primary
key

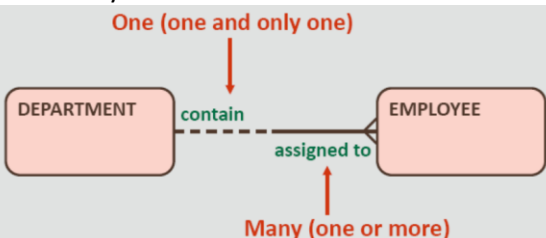
It has a non-Null
It has a value that never changes

Artificial UIDs do not occur in the natural world but are created for identification purposes in a system
Example Composite UID: Bank_No and Account_No.

2.5. Relationships \$\$



Cardinality: 1 : M



Foreign Key Examples

EMPLOYEES				
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID	Foreign Key
100	Steven	King	90	
101	Neena	Kochhar	90	
102	Lex	De Haan	90	
103	Alexander	Hunold	60	
104	Bruce	Ernst	60	

refers to

DEPARTMENTS	
DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
50	Shipping

Primary Key

Cardinality M : N

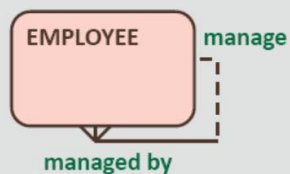


Cardinality 1 : 1



Recursive Relationships

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

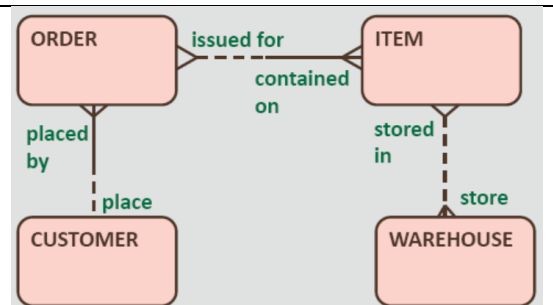


• Business rules:

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

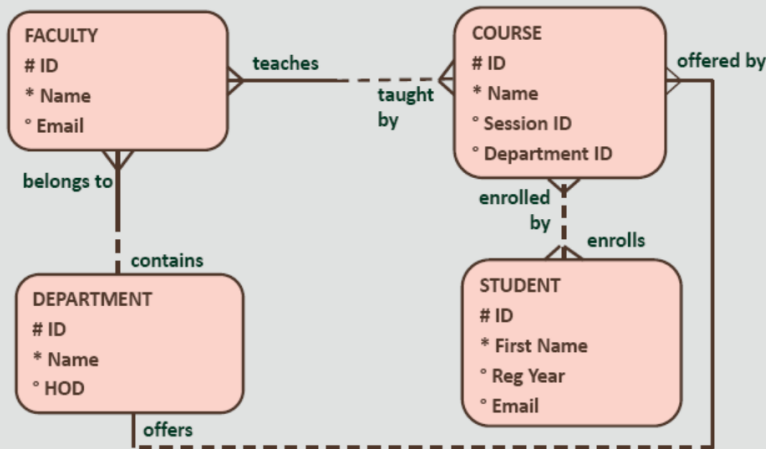
Relationship Matrix: Mapping the Contents

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

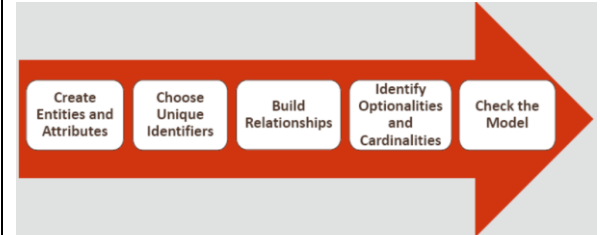


2.6. Entity Relationship Modeling (ERDs)

Logical Modeling: Example



Steps to Build an ERD

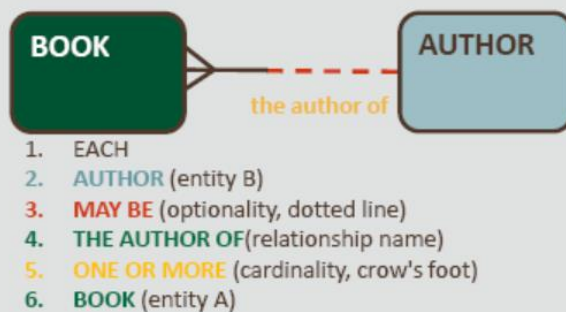
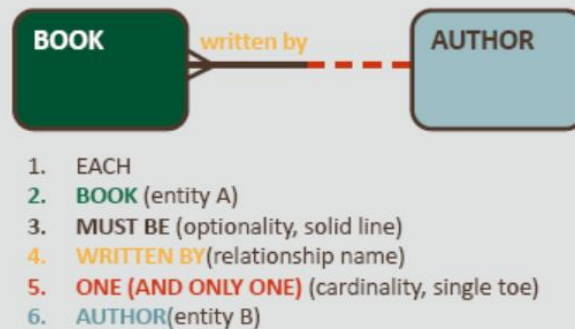


Components of ERDish

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

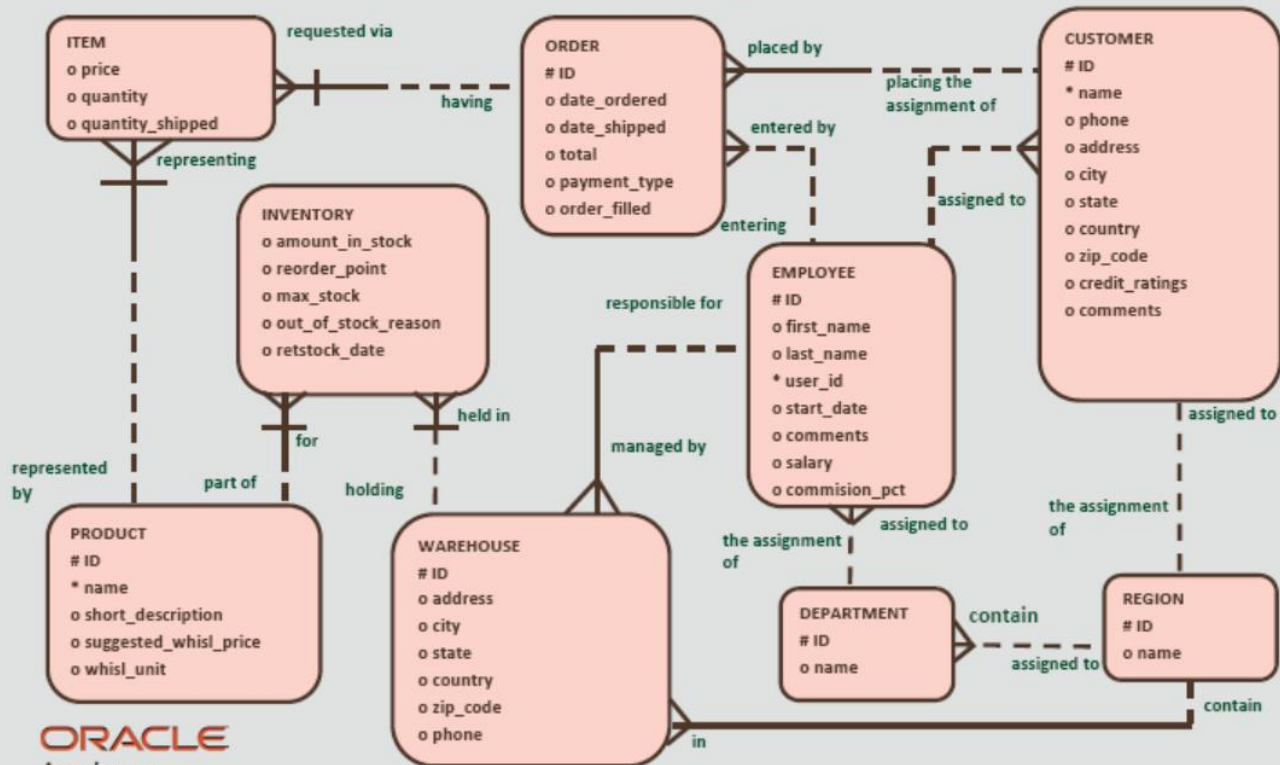
ERDish Example

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



Next, read the relationship from right to left.

Sample Solution for Sporting Goods ERD

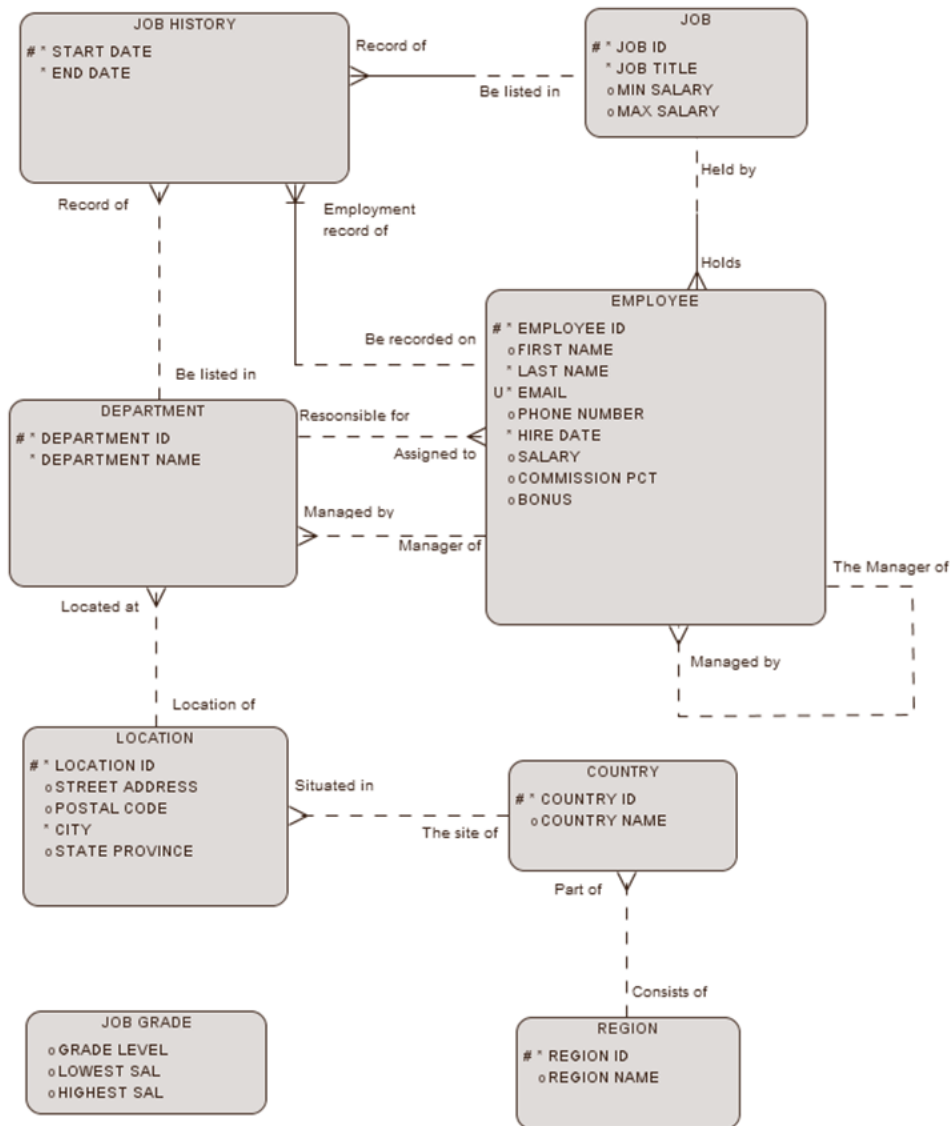


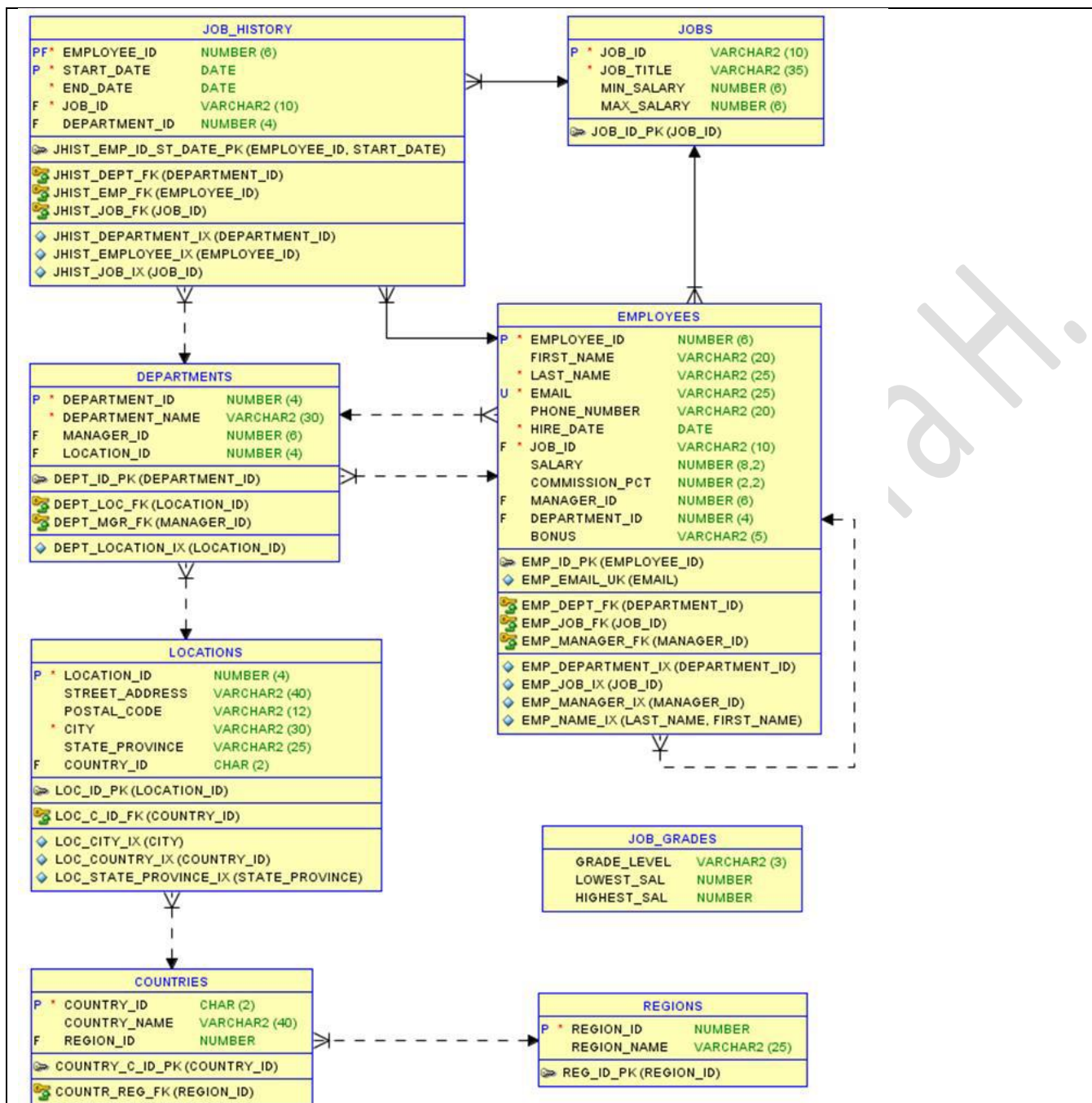
ORACLE
Academy

DFo 2-6
Entity Relationship Modelling (ERDs)

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. 25

25





→

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



Juan Carlos Herrera H.

4. Oracle SQL Developer Data Modeler

4.1. Oracle SQL Developer Data Modeler

4.2. Convert a Logical Model to a Relational Model



Juan Carlos Herrera H.

5. Mapping to the Physical Model

5.1. Mapping Entities and Attributes

5.2. Mapping Primary and Foreign Keys



Juan Carlos Herrera H.

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

→

Juan Carlos Herrera H.