DATABASE FOUNDATIONS

ORACLE ACADEMY

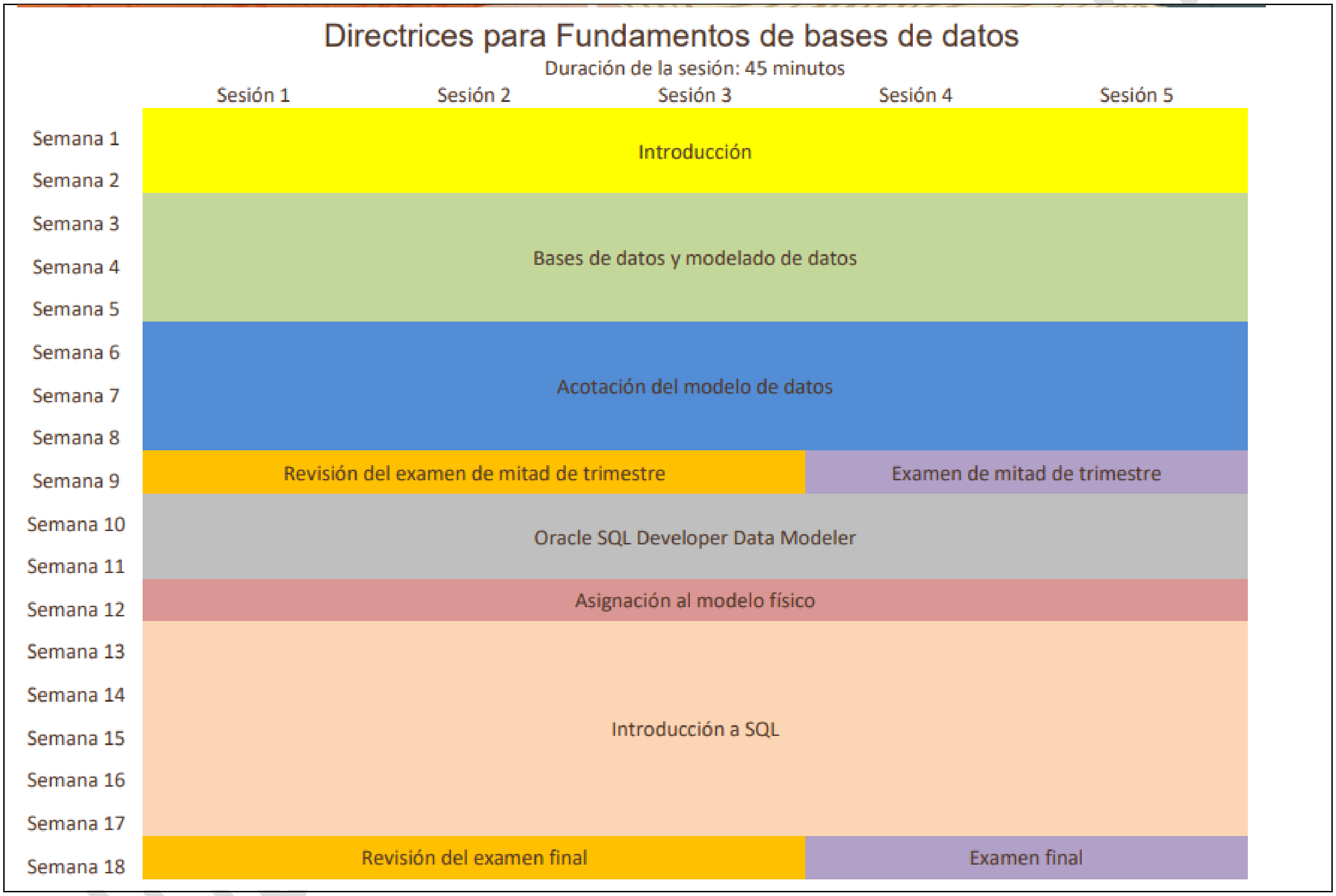
6 DE MAYO DE 2025

https://academy.oracle.com/

HTTPS://GITHUB.COM/ISC-UPA/2025-2-ISC05-DB

Contenido

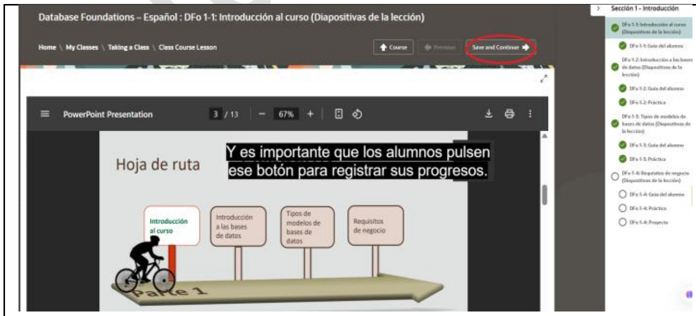
1. Introduction



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

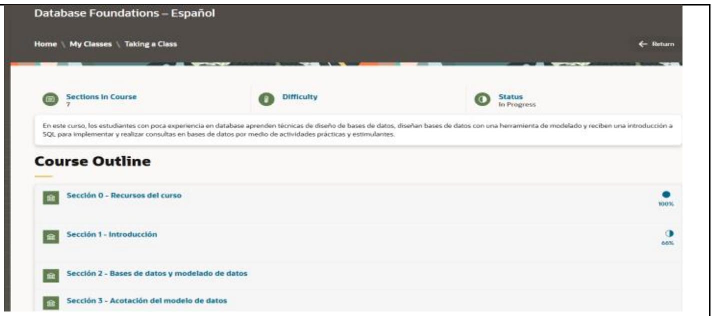
Hoja de ruta

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
Parte 1

Sección 1 - Introducción

- DFo 1-1: Introducción al curso (Diapositivas de la lección)
- DFo 1-1: Guía del alumno
- DFo 1-2: Introducción a las bases de datos (Diapositivas de la lección)
- DFo 1-2: Guía del alumno
- DFo 1-2: Práctica
- DFo 1-3: Tipos de modelos de bases de datos (Diapositivas de la lección)
- DFo 1-3: Guía del alumno
- DFo 1-3: Práctica
- DFo 1-4: Requisitos de negocio (Diapositivas de la lección)
- DFo 1-4: Guía del alumno
- DFo 1-4: Práctica
- DFo 1-4: Proyecto



Database Foundations – Español

Home \ My Classes \ Taking a Class

Sections In Course
7

Difficulty

Status
In Progress

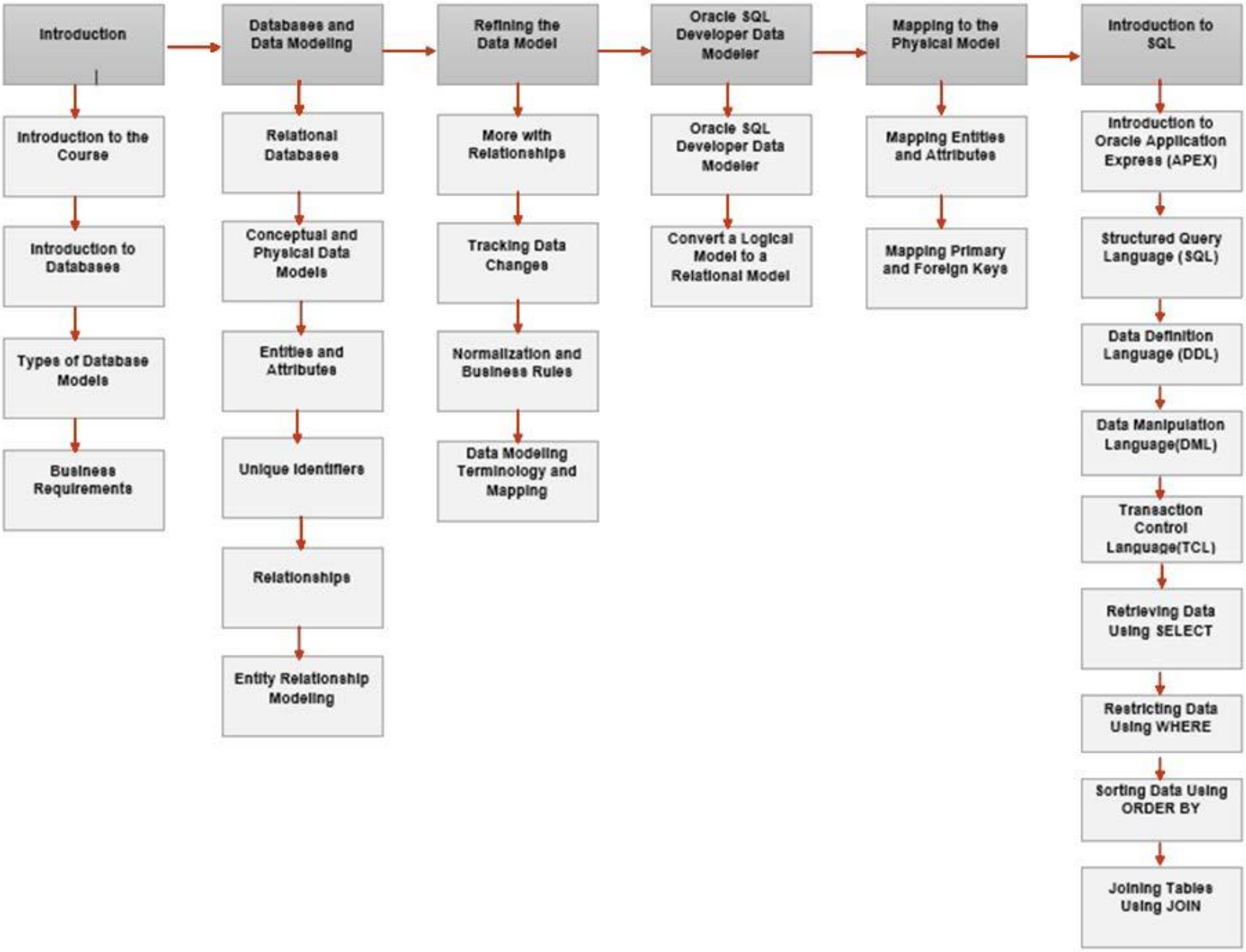
En este curso, los estudiantes con poca experiencia en database aprenden técnicas de diseño de bases de datos, diseñan bases de datos con una herramienta de modelado y reciben una introducción a SQL para implementar y realizar consultas en bases de datos por medio de actividades prácticas y estimulantes.

Course Outline

- Sección 0 – Recursos del curso — 100%
- Sección 1 - Introducción — 66%
- Sección 2 - Bases de datos y modelado de datos
- Sección 3 - Acotación del modelo de datos

→

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 Oracle APEX application
Oracle Data Modeler

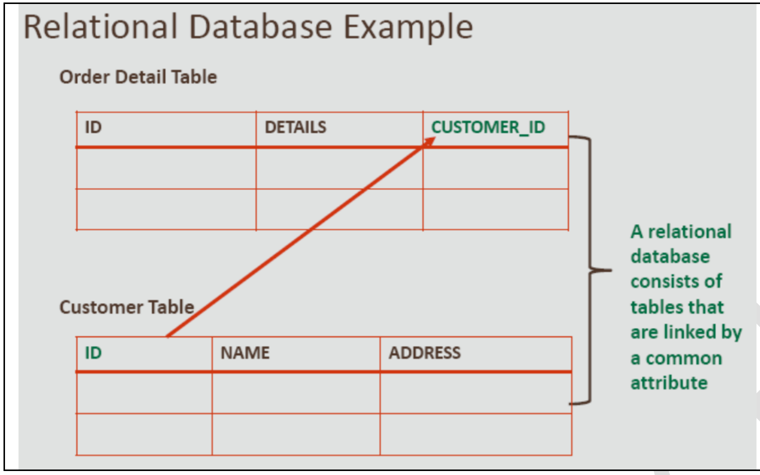
→

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.

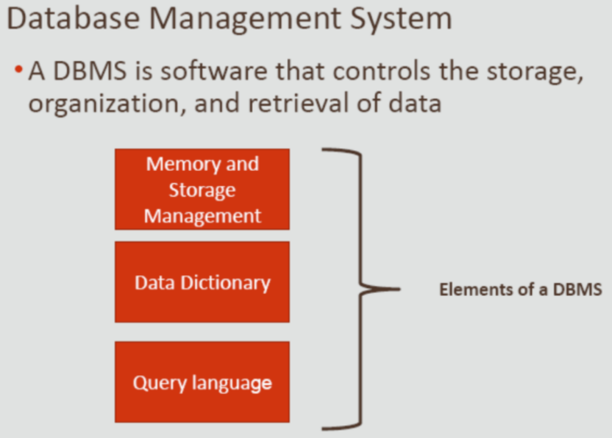
Introduction to Relational Databases

- 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

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

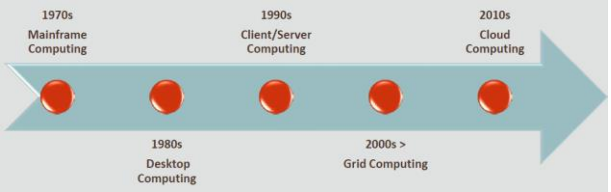


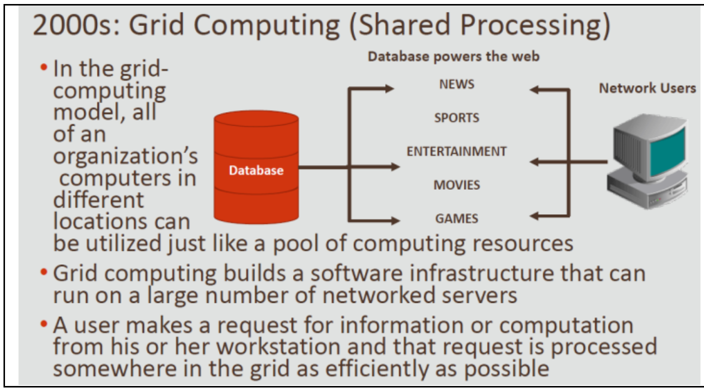
Database Management System

- 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)

Database powers the web

Database

NEWS

SPORTS

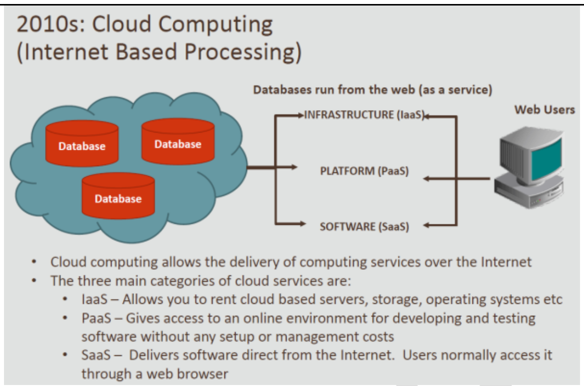
ENTERTAINMENT

MOVIES

GAMES

Network Users

- 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)

Database

Database

Database

INFRASTRUCTURE (IaaS)

PLATFORM (PaaS)

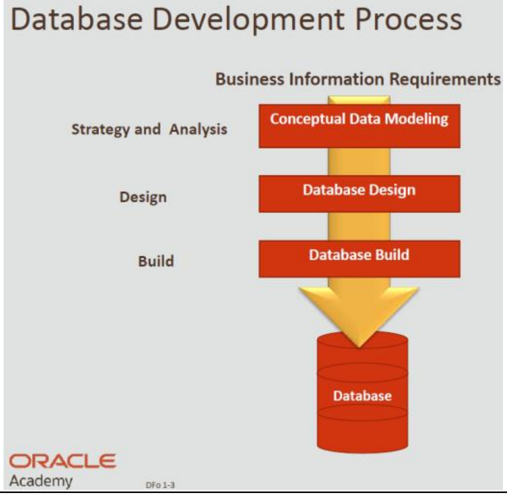
SOFTWARE (SaaS)

Web Users

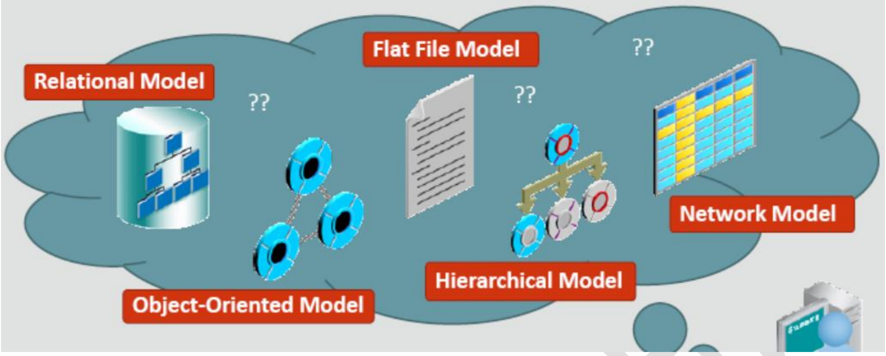
- Cloud computing allows the delivery of computing services over the Internet
- The three main categories of cloud services are:
  - IaaS – Allows you to rent cloud based servers, storage, operating systems etc
  - PaaS – Gives access to an online environment for developing and testing software without any setup or management costs
  - SaaS – Delivers software direct from the Internet. Users normally access it through a web browser

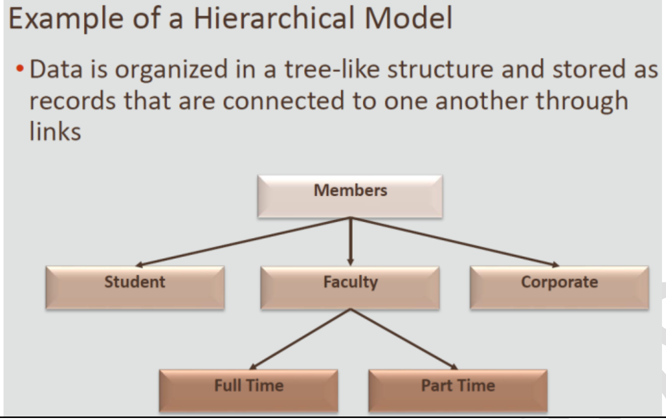
→

1.3. Types of Database Models



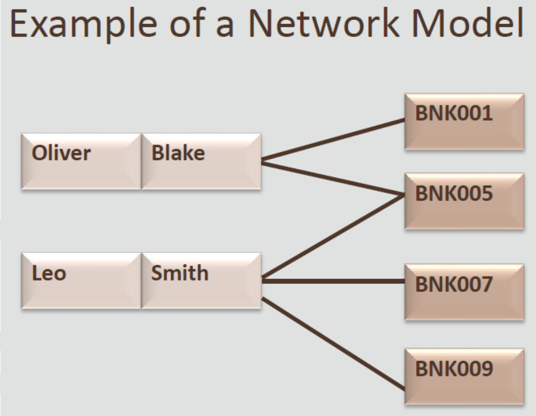
Database Development Process



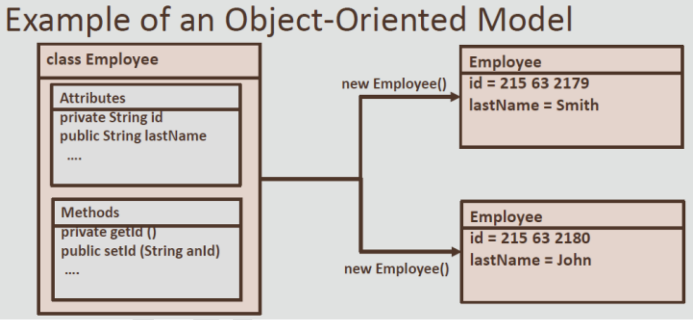


Example of a Hierarchical Model

- Data is organized in a tree-like structure and stored as records that are connected to one another through links



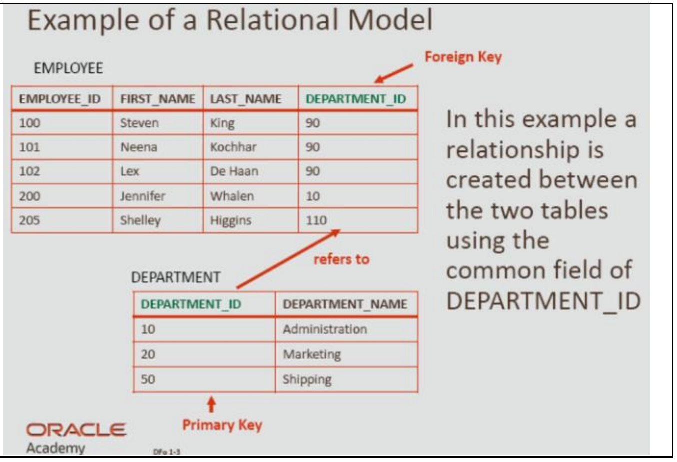
Example of a Network Model



Example of an Object-Oriented Model

Los nombres de clases son en Singular

Los nombres de las tablas son en Plural



Example of a Relational Model

EMPLOYEE

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

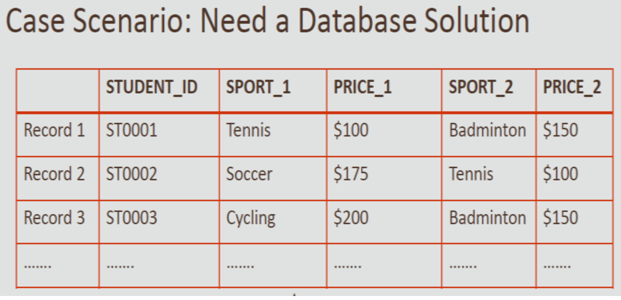
DEPARTMENT

DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
50	Shipping

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

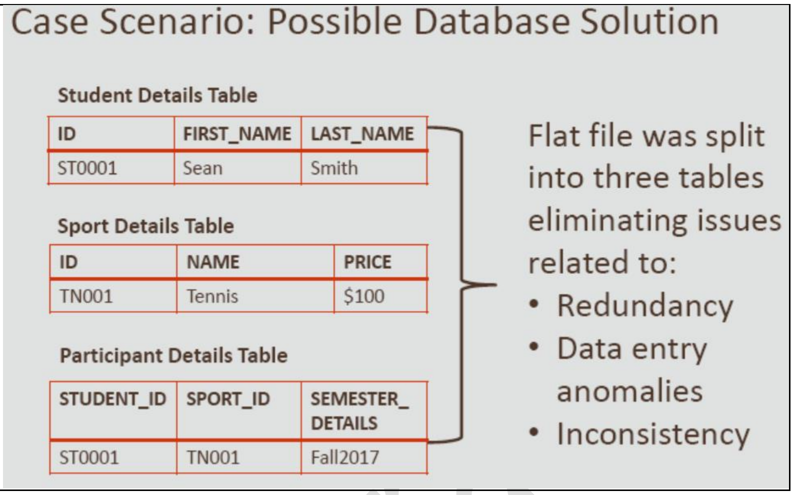
→

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:

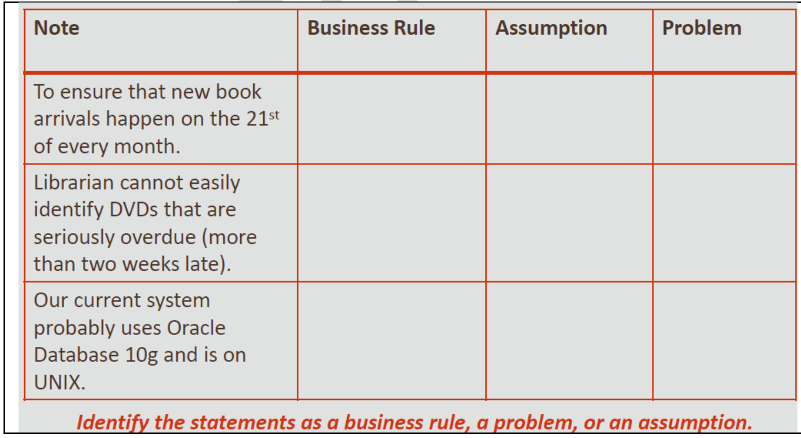
- Allow the developer/architect to understand the relationship and constraints of the participating entities
- Help you understand the standardization procedure that an organization follows when handling huge data
- Should be simple and easy to understand
- Must be kept up-to-date

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:



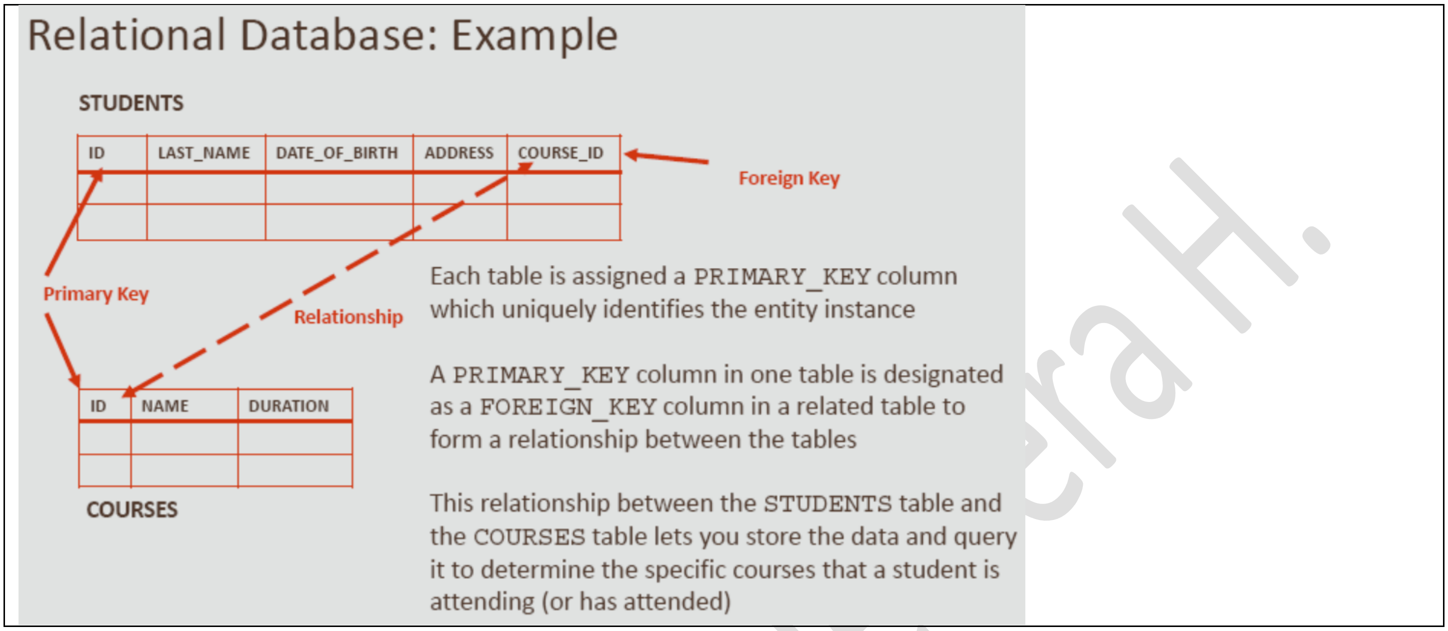
Note	Business Rule	Assumption	Problem
To ensure that new book arrivals happen on the 21st of every month.			
Librarian cannot easily identify DVDs that are seriously overdue (more than two weeks late).			
Our current system probably uses Oracle Database 10g and is on UNIX.			

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

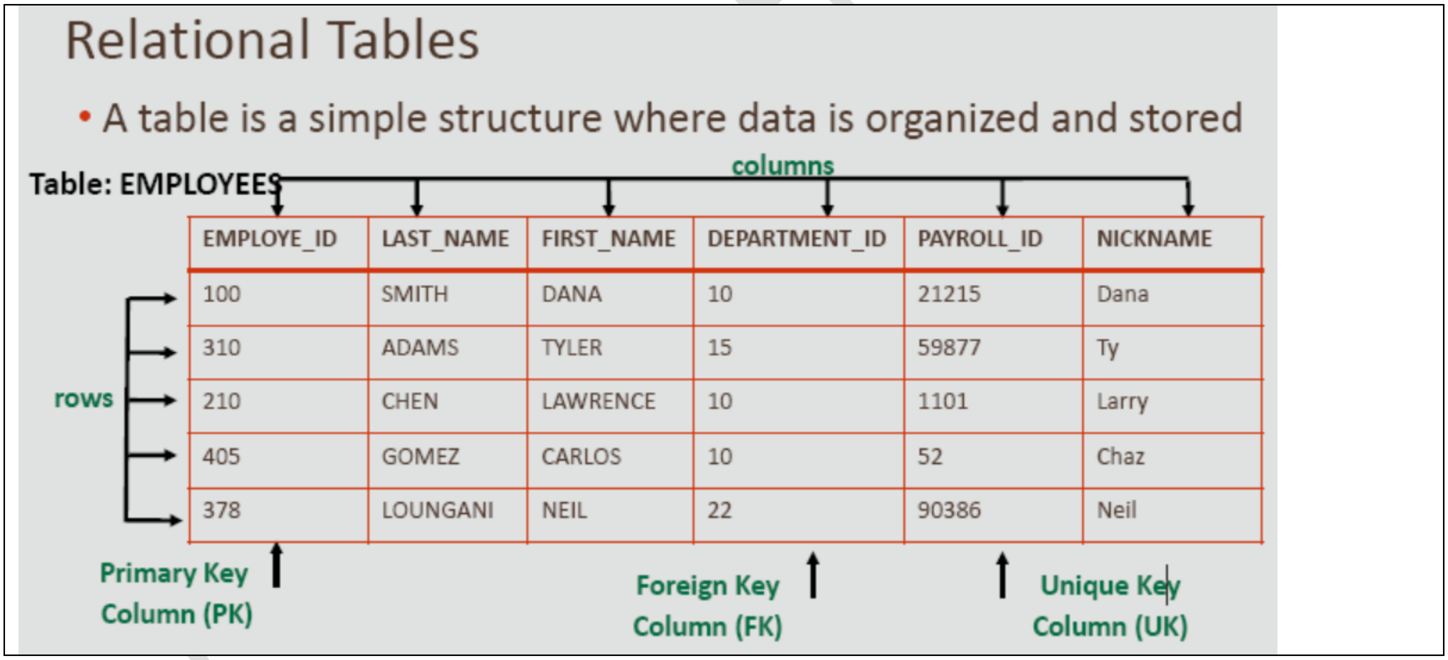
ID	NAME	DURATION

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

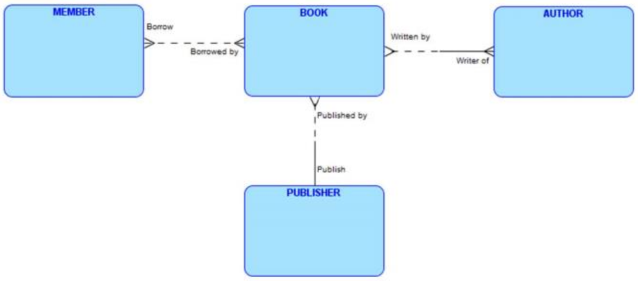
Key Terms

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

→

2.2. Conceptual and Physical Data Models

Conceptual data Model



Identifies :

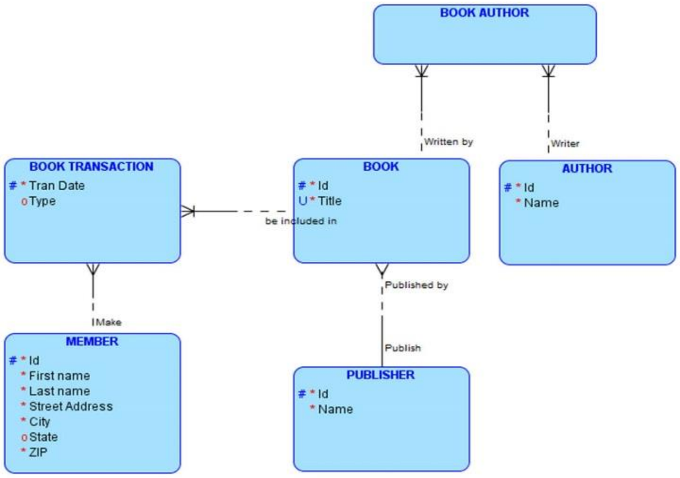
- important entities
- relationships among entities

Does not specify :

- attributes
- unique identifiers (primary key)

Logical data Model (ERM o ERD)

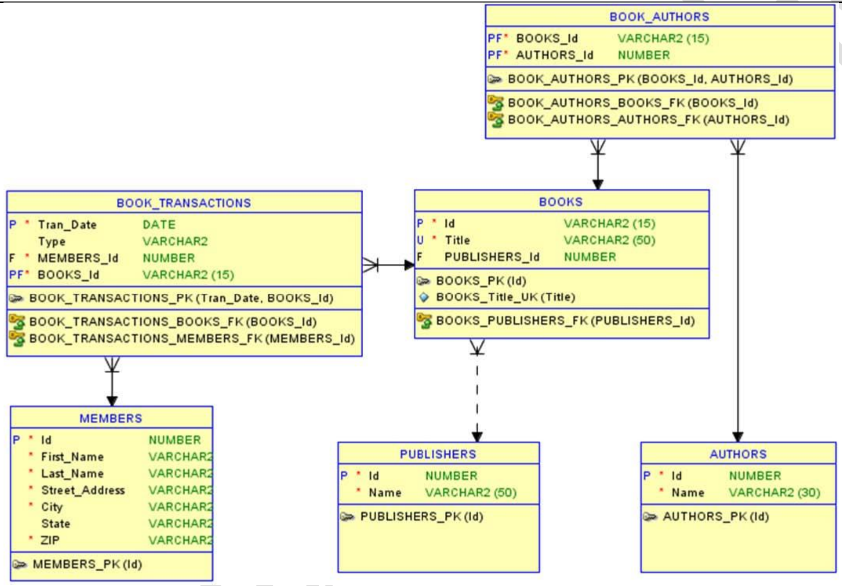
Entity Relationship Model/Diagram



Exists: attributes (# key, * not null , o optional)

Does not exist: N:M relationship

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



Is a extesion to a logical data model:

- Atrributes data types and precision
- Identifies indexes
- Primary keys (P)
- Foreign keys (F)
- Unique keys (U)

Modeling Performed :

Entities → Tables

Attributes → Columns

Relationships → Foreign keys

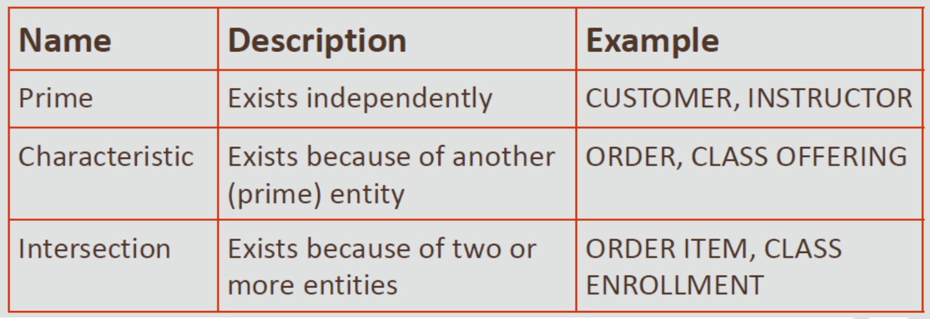
→

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:

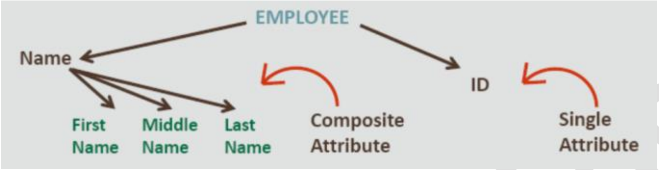


Name	Description	Example
Prime	Exists independently	CUSTOMER, INSTRUCTOR
Characteristic	Exists because of another (prime) entity	ORDER, CLASS OFFERING
Intersection	Exists because of two or more entities	ORDER ITEM, CLASS ENROLLMENT

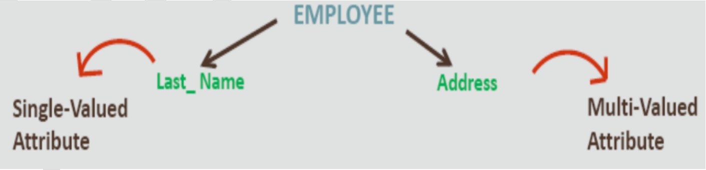
Tipos de Entidad:
Principal
Caracteristica
Interseccion

Entidades:
Fuertes
Débiles

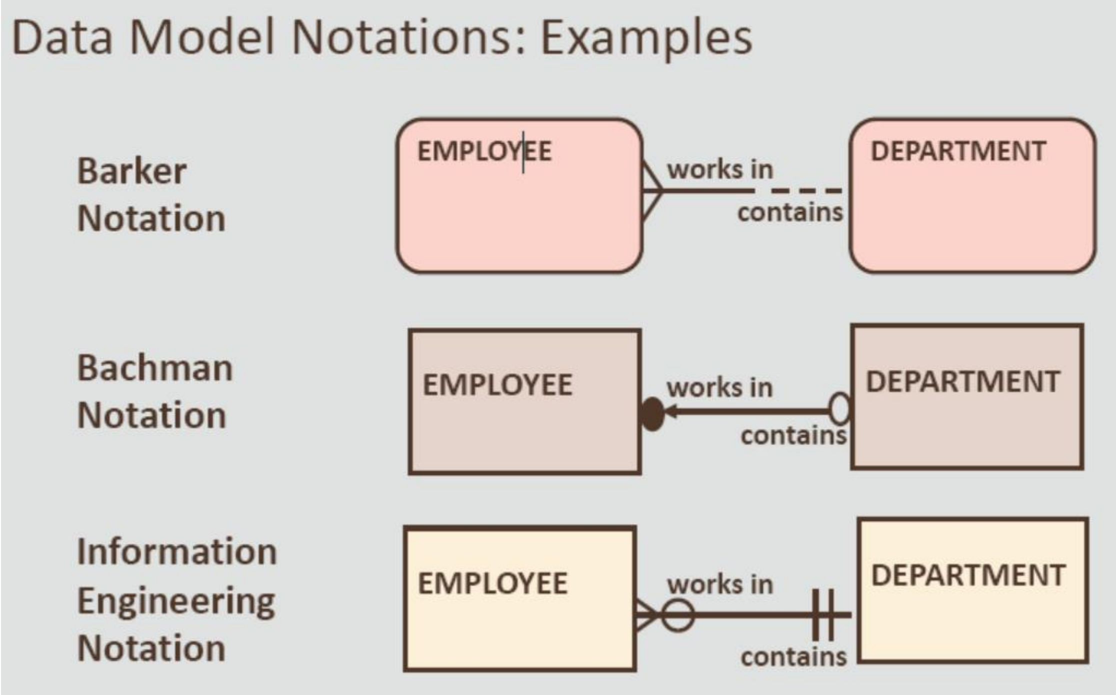
Single(atomic) and Composite(subparts) Attributes



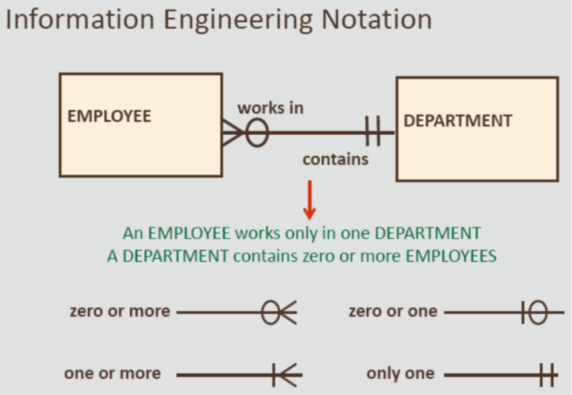
Single-Valued and Multi-Valued Attributes

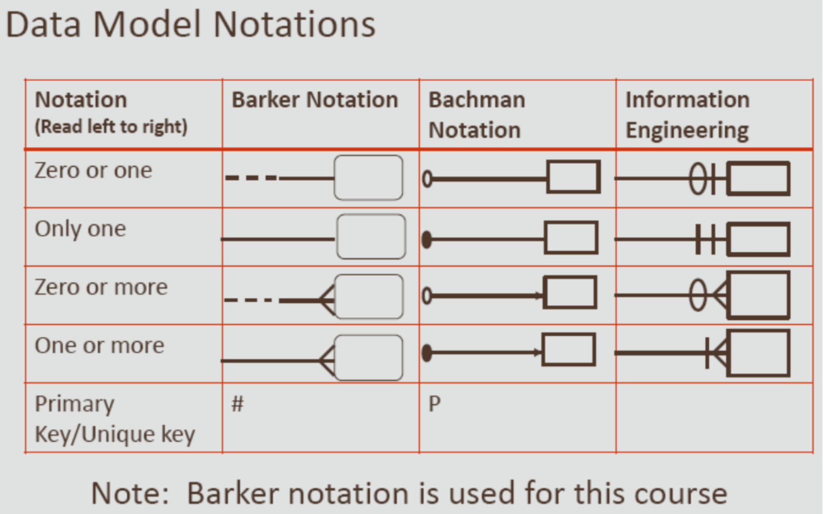


Data Model Notations: Examples



Black: debe
White: puede
Arrow: many

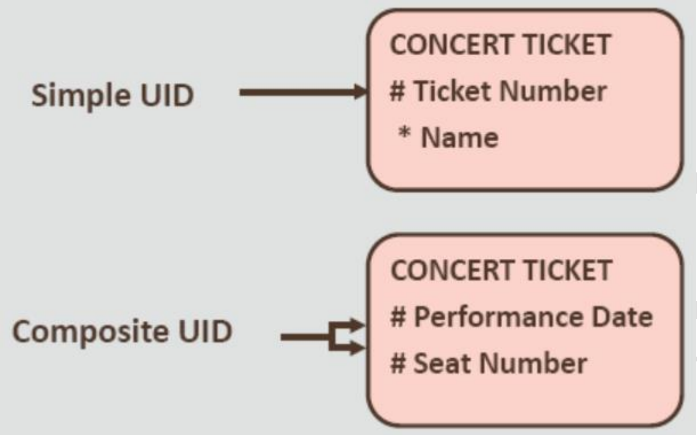

Information Engineering Notation
EMPLOYEE
works in
DEPARTMENT
contains
An EMPLOYEE works only in one DEPARTMENT
A DEPARTMENT contains zero or more EMPLOYEES
zero or more
zero or one
one or more
only one


Data Model Notations

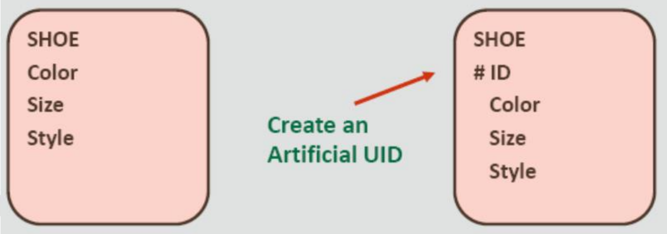
Notation (Read left to right)	Barker Notation	Bachman Notation	Information Engineering
Zero or one			
Only one			
Zero or more			
One or more			
Primary Key/Unique key	#	P	

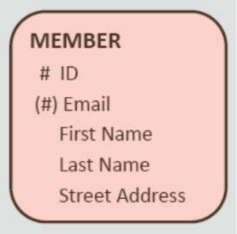
Note: Barker notation is used for this course

2.4. Unique Identifiers



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



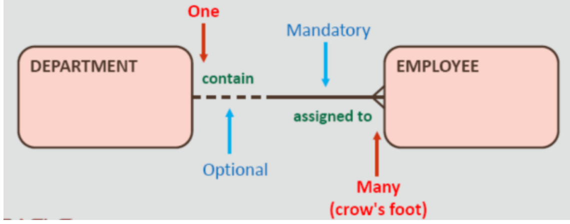


Candidate unique identifiers or secondary UIDs, so define primary key

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



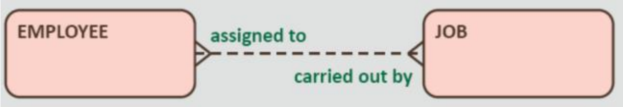
Type:	1 : M
Cardinality:	one and only one in one direction and one or more in other direction
Optional:	Use "may be" or "may."
Mandatory:	Use "must be" or "must."
Line:	Use "one and only one."
Crow's feet:	Use "one or more."

Relationship Types or Cardinality

Many-to-one (M : 1) or one-to-many (1 : M)
Many-to-many (M : M)
One-to-one (1 : 1)

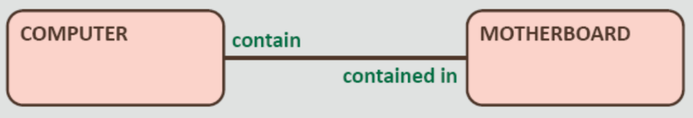
Type: Many to Many

Cardinality: one or more in both directions



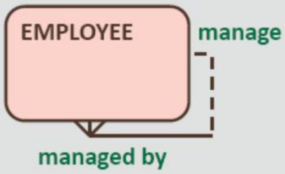
Type 1 : 1

Cardinality: one and only one in both directions

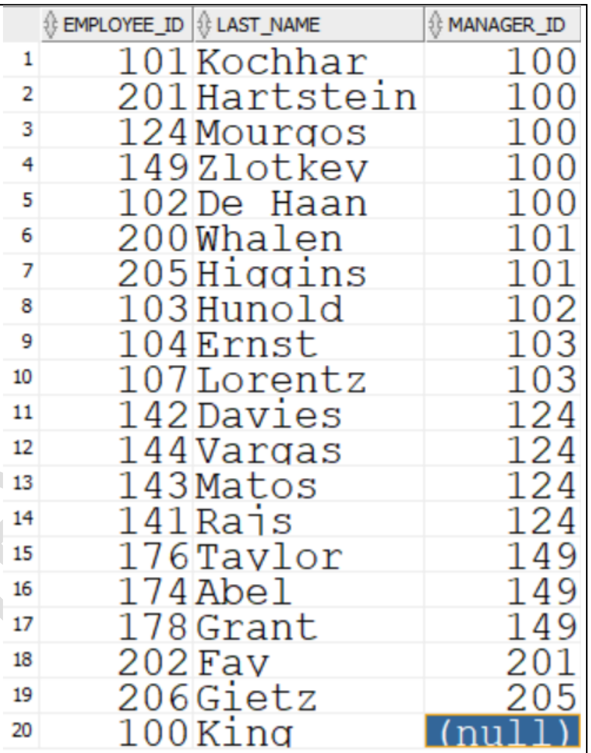


Recursive Relationships

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

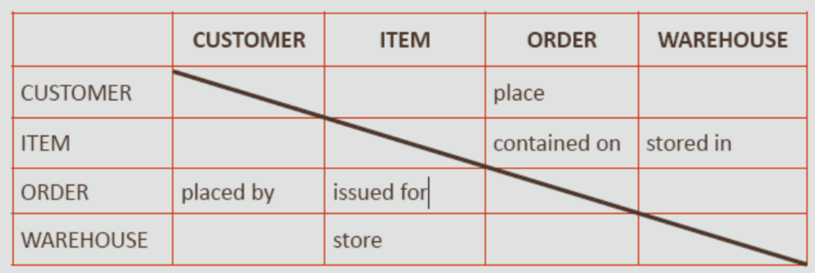


- Business rules:
  - Each EMPLOYEE may manage one or more EMPLOYEE
  - Each EMPLOYEE must be managed by one and only one EMPLOYEE



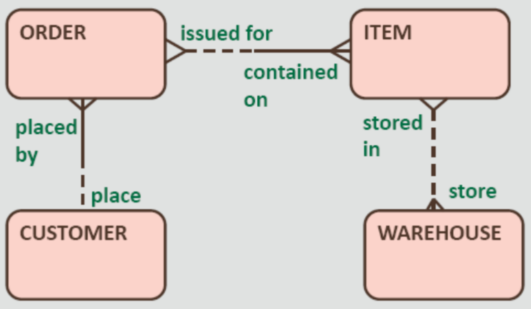
	EMPLOYEE_ID	LAST_NAME	MANAGER_ID
1	101	Kochhar	100
2	201	Hartstein	100
3	124	Mourgos	100
4	149	Zlotkey	100
5	102	De Haan	100
6	200	Whalen	101
7	205	Higgins	101
8	103	Hunold	102
9	104	Ernst	103
10	107	Lorentz	103
11	142	Davies	124
12	144	Vargas	124
13	143	Matos	124
14	141	Rajs	124
15	176	Taylor	149
16	174	Abel	149
17	178	Grant	149
18	202	Fay	201
19	206	Gietz	205
20	100	King	(null)

Relationship Matrix: Mapping the Contents



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

A relationship matrix can be used to collect initial information about the relationships among a set of entities



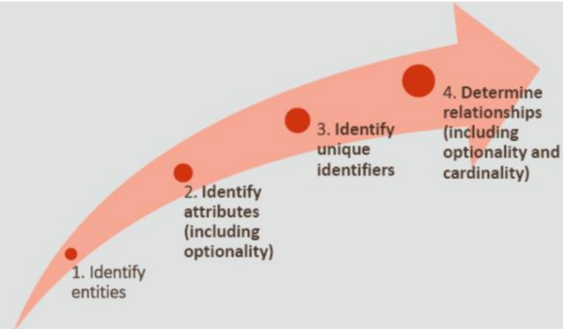
→

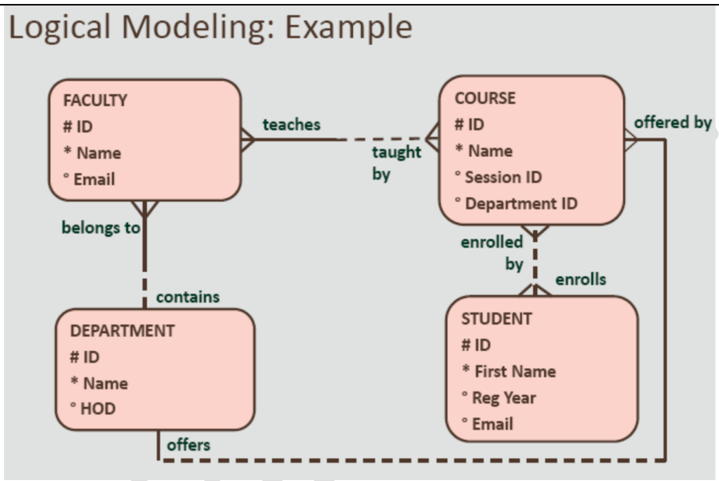
2.6. Entity Relationship Modeling (ERDs)

DB roles: designers, database administrators, and application developers

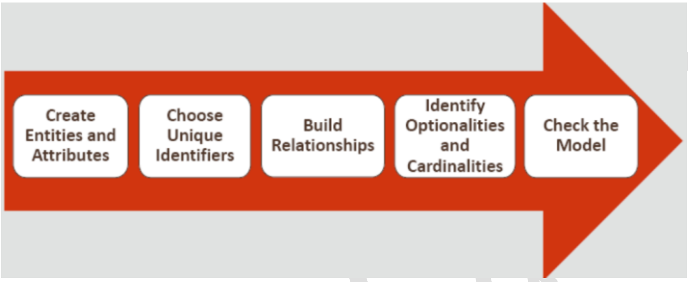
Logical Modeling:
Includes all entities, attributes, UIDs and relationships as well as optionality and cardinality of these ítems

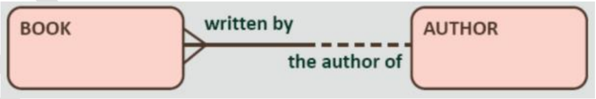
Steps to Create a Logical Model


1. Identify entities
2. Identify attributes (including optionality)
3. Identify unique identifiers
4. Determine relationships (including optionality and cardinality)



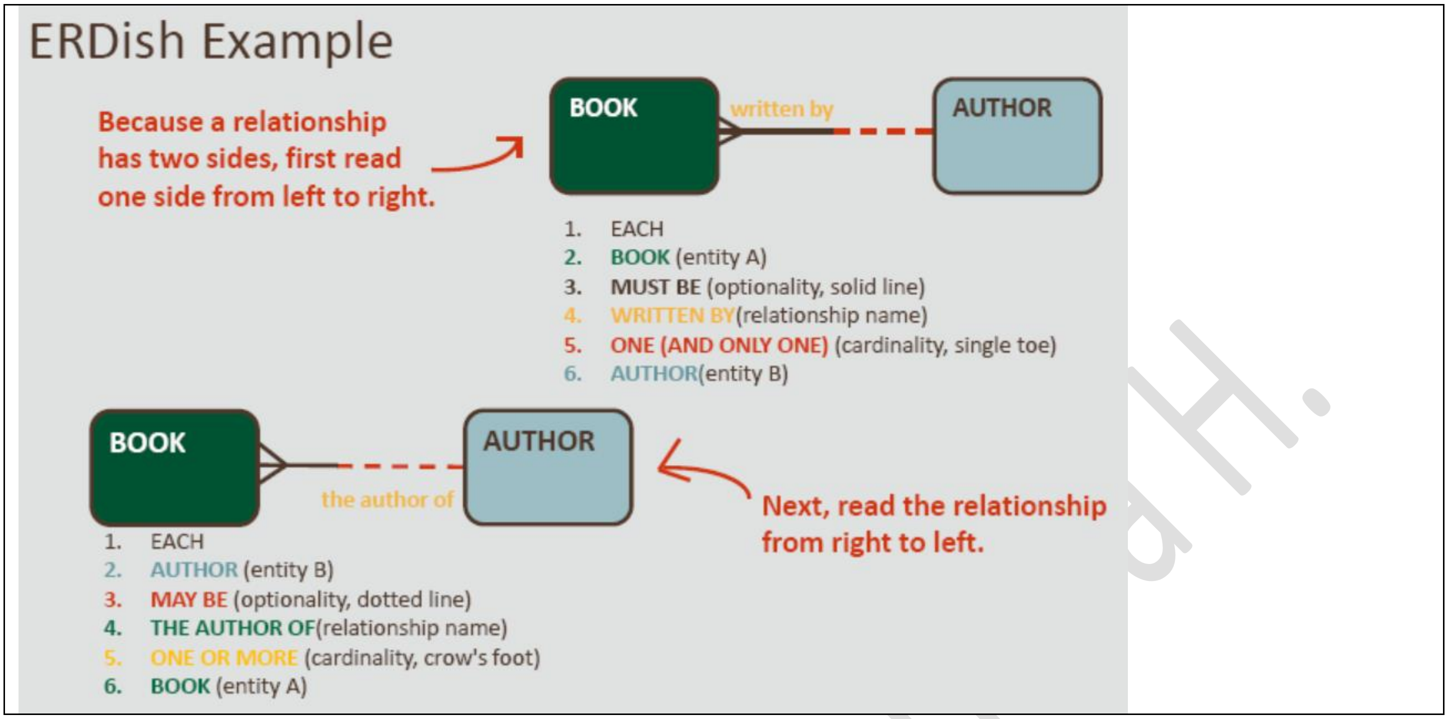
Steps to Build an ERD


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



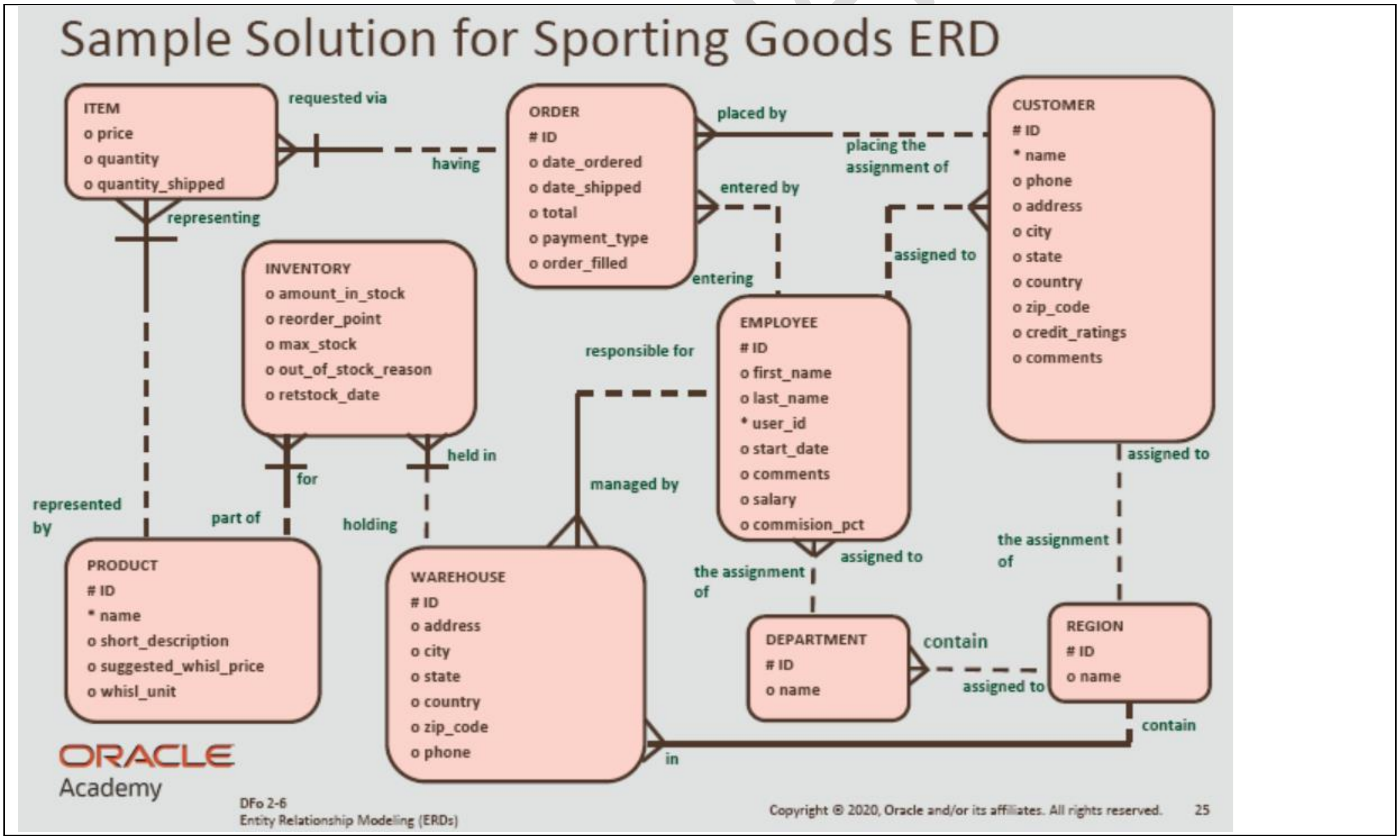
ERDish Example

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

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)

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

ORACLE Academy

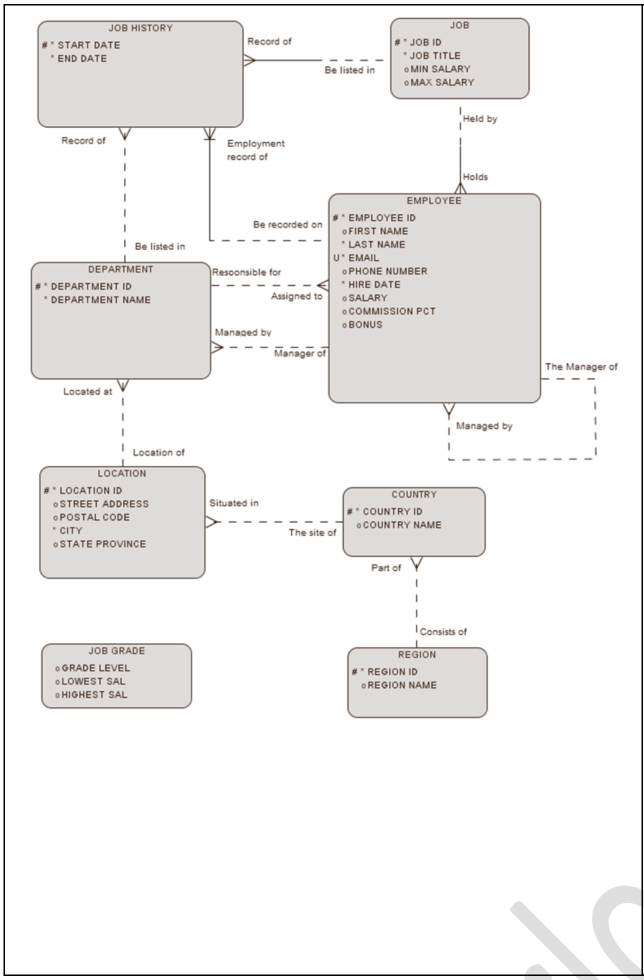
DFo 2-6
Entity Relationship Modeling (ERDs)

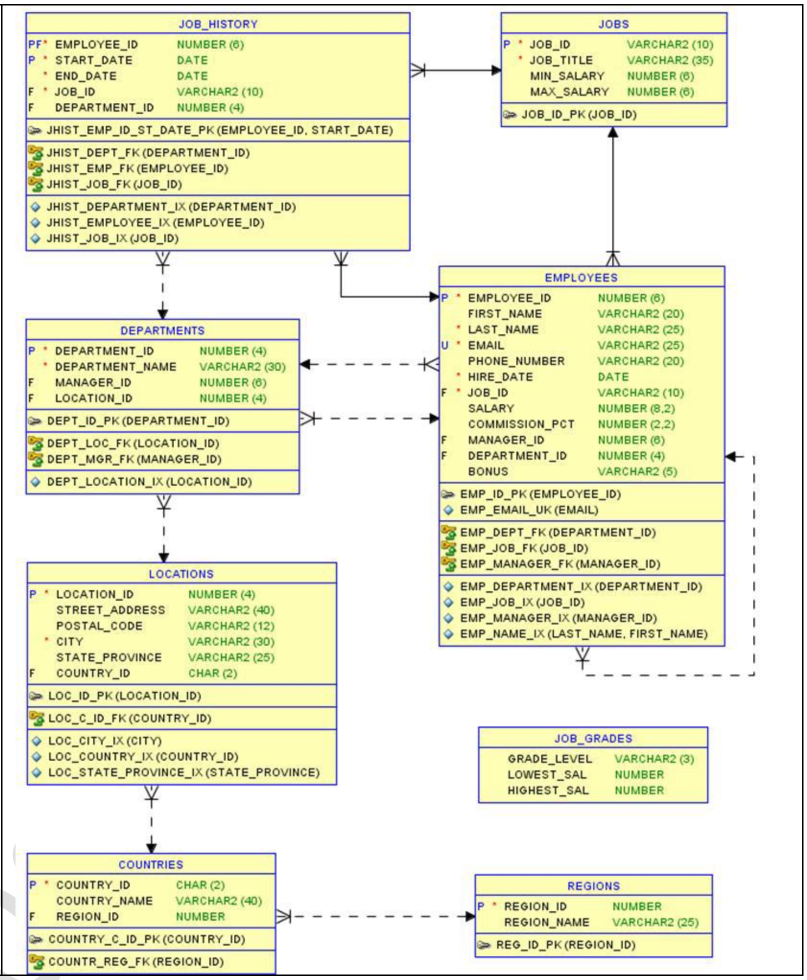
Copyright © 2020, Oracle and/or its affiliates. All rights reserved. 25

→

Logical Data Model

Physical data Model



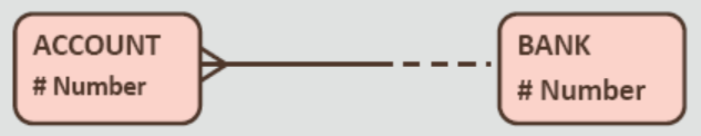

JOB_HISTORY
PF * EMPLOYEE_ID NUMBER (6)
P * START_DATE DATE
* END_DATE DATE
F * JOB_ID VARCHAR2 (10)
F DEPARTMENT_ID NUMBER (4)
JHIST_EMP_ID_ST_DATE_PK (EMPLOYEE_ID, START_DATE)
JHIST_DEPT_FK (DEPARTMENT_ID)
JHIST_EMP_FK (EMPLOYEE_ID)
JHIST_JOB_FK (JOB_ID)
JHIST_DEPARTMENT_IX (DEPARTMENT_ID)
JHIST_EMPLOYEE_IX (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)
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)
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)
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)

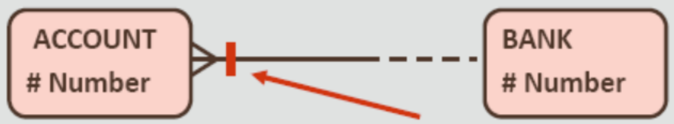
→

3. Refining the Data Model

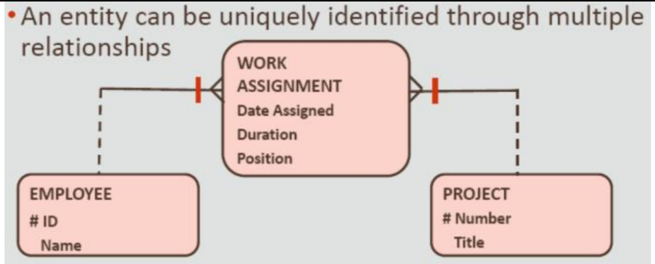
3.1. More with Relationships

3.1.1. Identifying (| Barred) Relationships





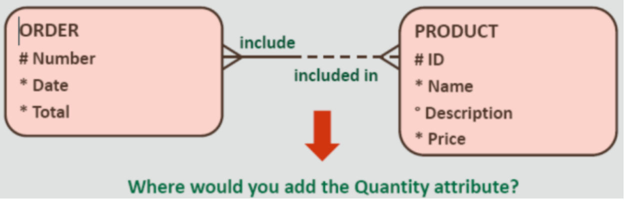
The UID for the ACCOUNT :
BANK Number , ACCOUNT Number



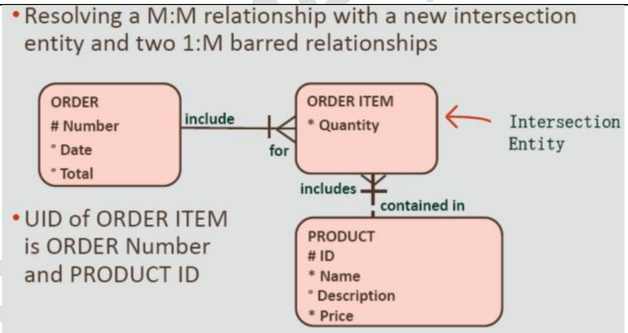
- An entity can be uniquely identified through multiple relationships

The UID for WORK ASSIGNMENT:
EMPLOYEE ID , PROJECT Number

3.1.2. M:M Relationships



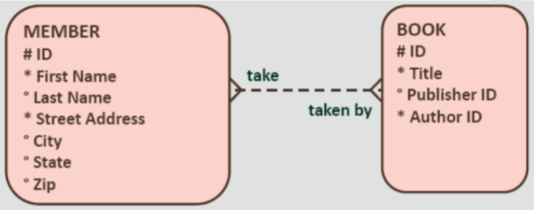
Where would you add the Quantity attribute?



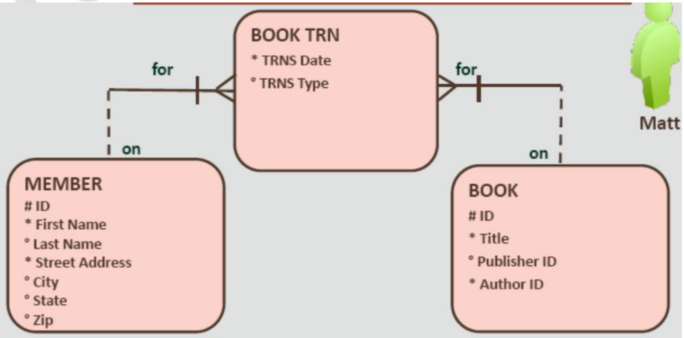
- Resolving a M:M relationship with a new intersection entity and two 1:M barred relationships

- UID of ORDER ITEM is ORDER Number and PRODUCT ID

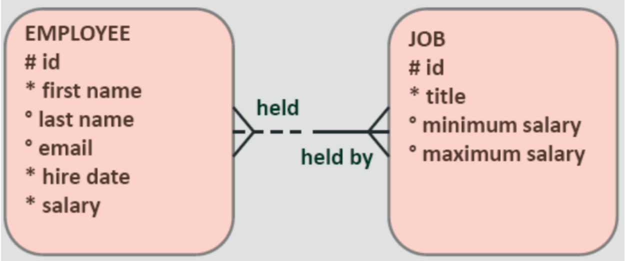
Case Scenario: Resolving M:M Relationships



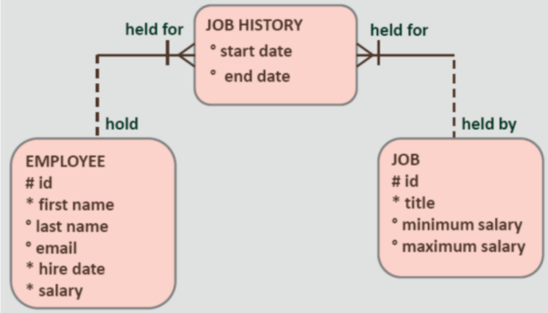
Case Scenario: Creating an Intersection Entity with Barred



Resolving M:M Relationships: Example 2

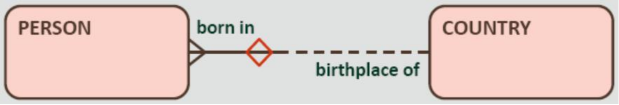


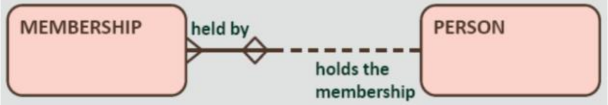
Resolving M:M Relationships Using Intersection



→

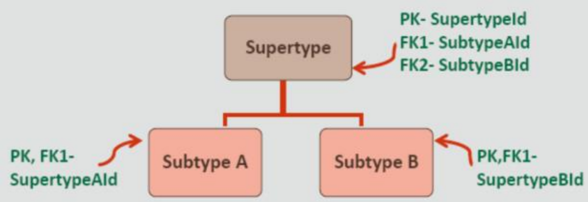
3.1.3. Non-Transferable Relationships ◇



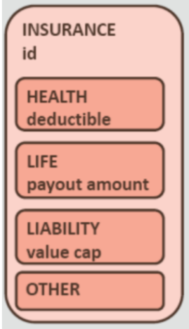


3.1.4. Supertype and Subtype Entities

- Supertype has a parent-child relationship with one or more subtypes
- Subtype is a subgrouping of the entity in an entity type which has attributes that are distinct from those in other subgroupings



Drawing a Subtype

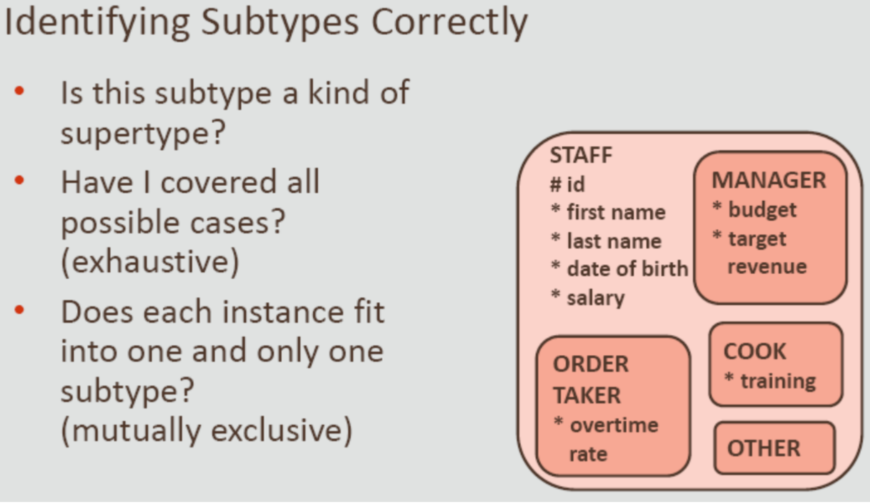


Characteristics of a Subtype

A subtype:

- Inherits all attributes of the supertype
- Inherits all relationships of the supertype
- Usually has its own attributes or relationships
- Is drawn within the supertype
- Never exists alone
- Has identical primary keys of the supertype and subtype

Example: An employee must be:
full time, part time, or other

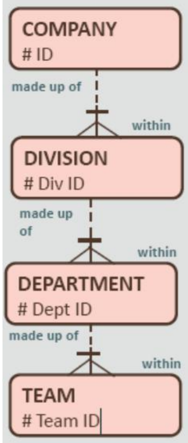


Identifying Subtypes Correctly

- Is this subtype a kind of supertype?
- Have I covered all possible cases? (exhaustive)
- Does each instance fit into one and only one subtype? (mutually exclusive)

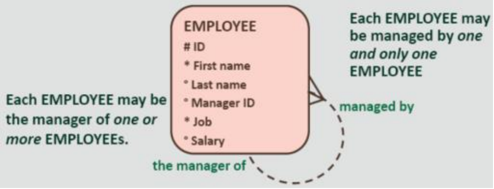
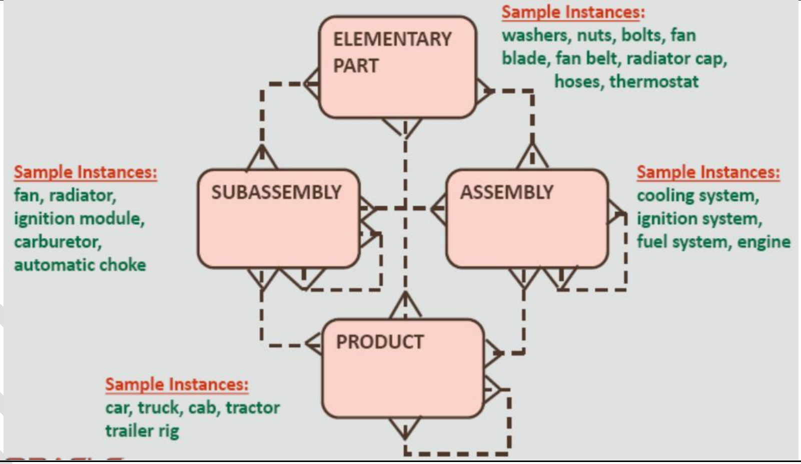
→

3.1.5. Modeling Hierarchical Data

3.1.6. Recursive Relationships

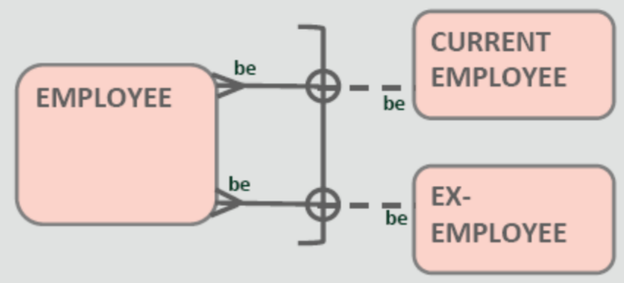
- A recursive relationship is always modeled with a loop.
- A recursive relationship is one where an entity instance is related to another instance in the same entity

EMPLOYEE
# ID
* First name
* Last name
* Manager ID
* Job
* Salary
Each EMPLOYEE may be managed by *one and only one* EMPLOYEE
managed by
Each EMPLOYEE may be the manager of *one or more* EMPLOYEEs.
the manager of

ELEMENTARY PART
Sample Instances:
washers, nuts, bolts, fan blade, fan belt, radiator cap, hoses, thermostat
SUBASSEMBLY
Sample Instances:
fan, radiator, ignition module, carburetor, automatic choke
ASSEMBLY
Sample Instances:
cooling system, ignition system, fuel system, engine
PRODUCT
Sample Instances:
car, truck, cab, tractor trailer rig

3.1.7. Arc Relationship ⌒

→

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

→