DATABASE FOUNDATIONS

ORACLE ACADEMY

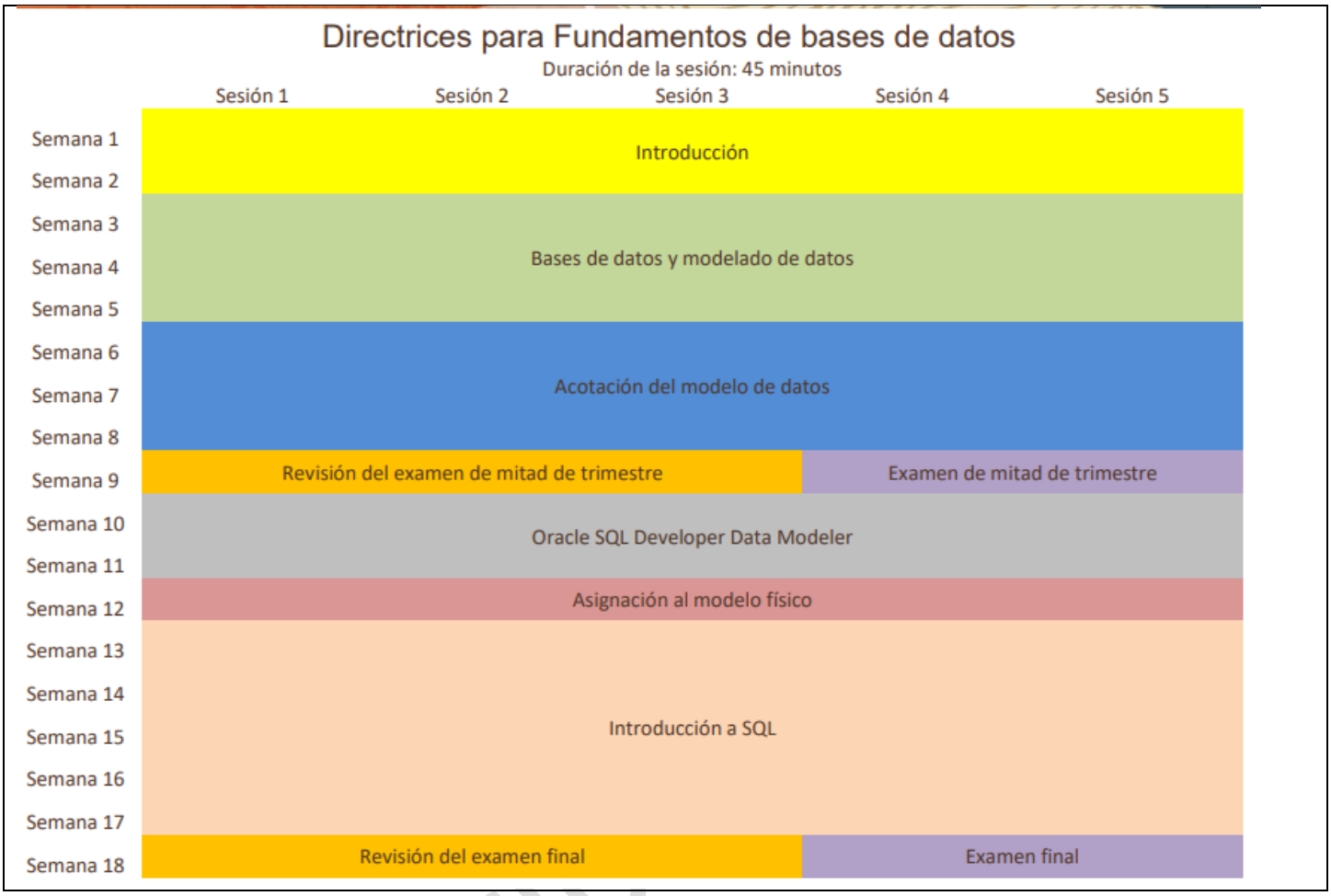
6 DE MAYO DE 2025

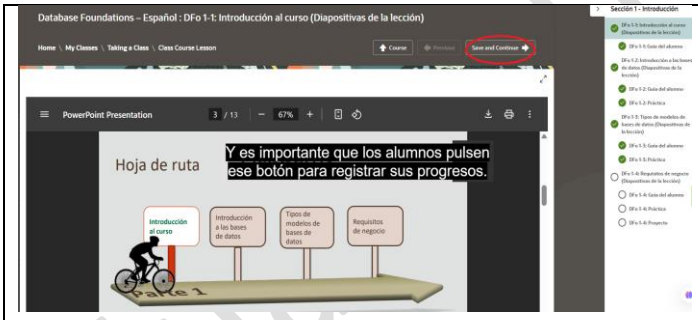
https://academy.oracle.com/

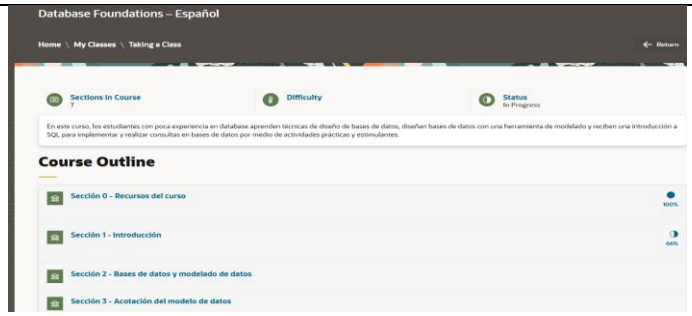
HTTPS://GITHUB.COM/ISC-UPA/2025-2-ISC05-DB

Contenido

1. Introduction

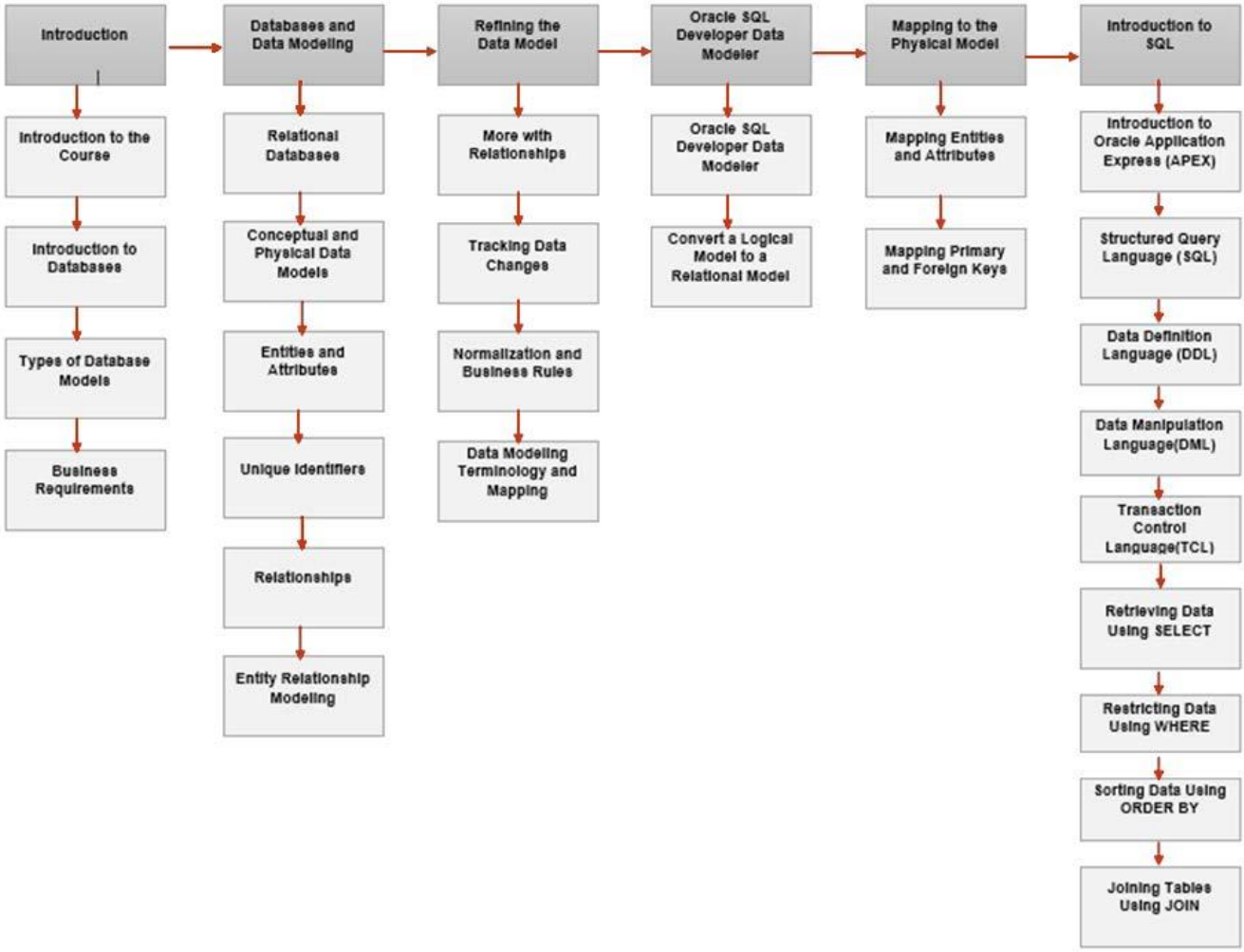






→

1.1. Introduction



Technological Requirements:

Oracle SQL Developer or Oracle APEX application
Oracle Data Modeler

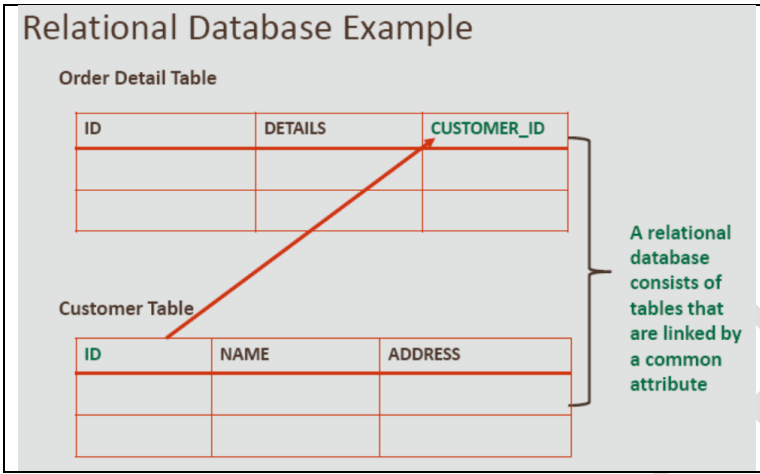
→

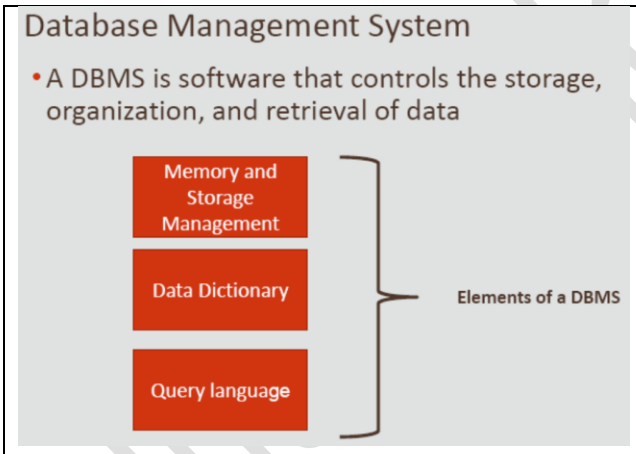
1.2. Introduction to Databases

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

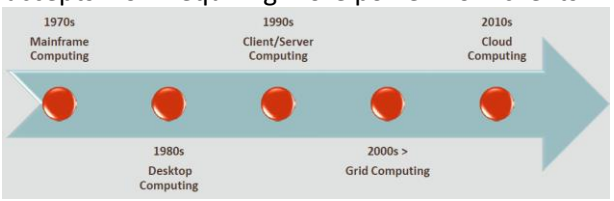
- A relational database stores information in tables with rows and columns
- A table is a collection of records
- A row is called a record (or instance)
- A record is a collection of fields
- A column is referred to as a field (or attribute)

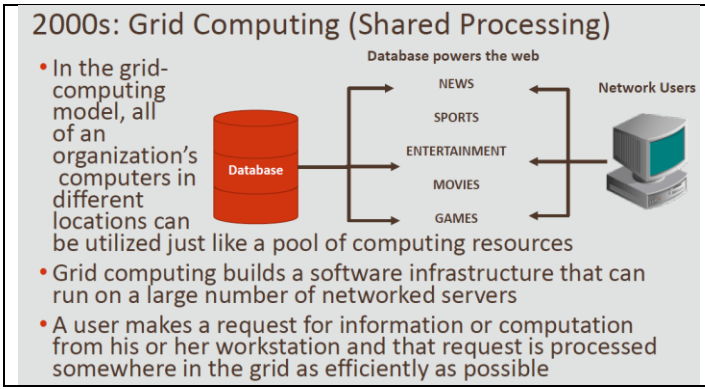


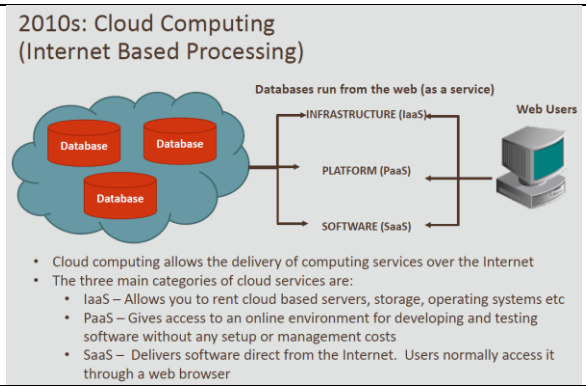


Key Computing Terms

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

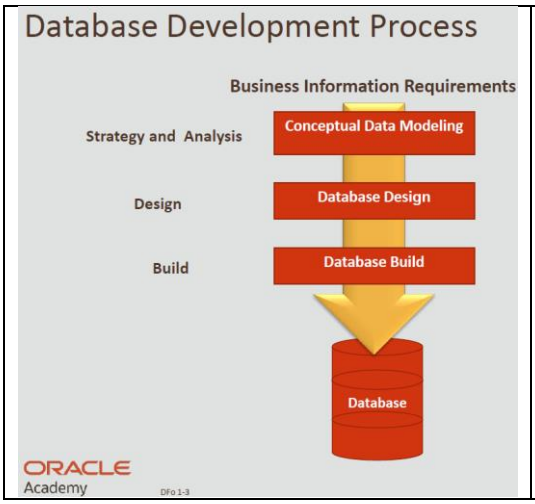


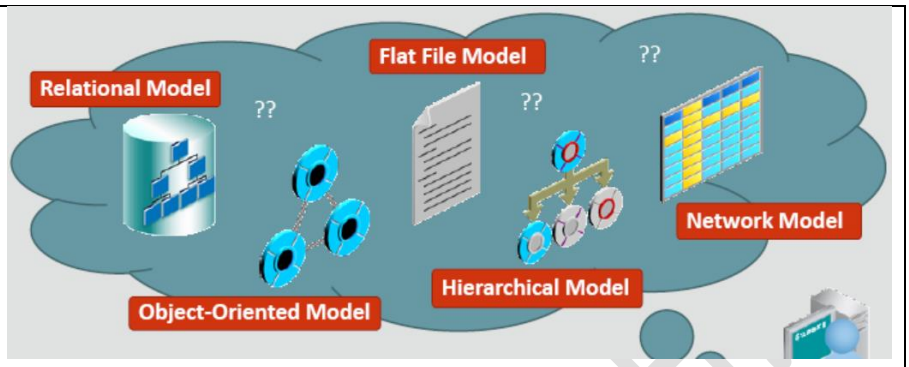


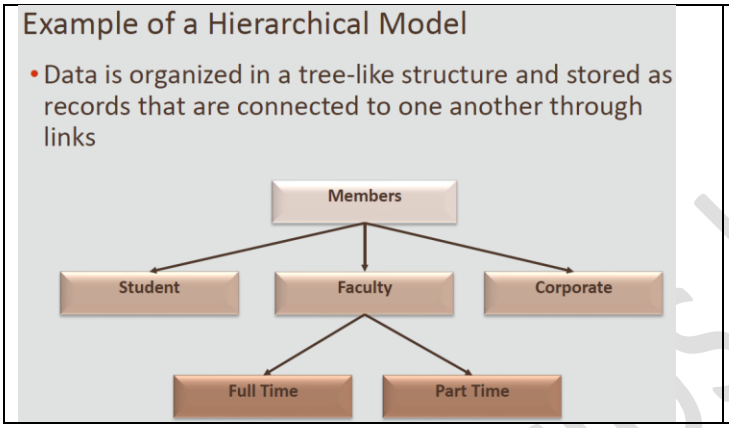


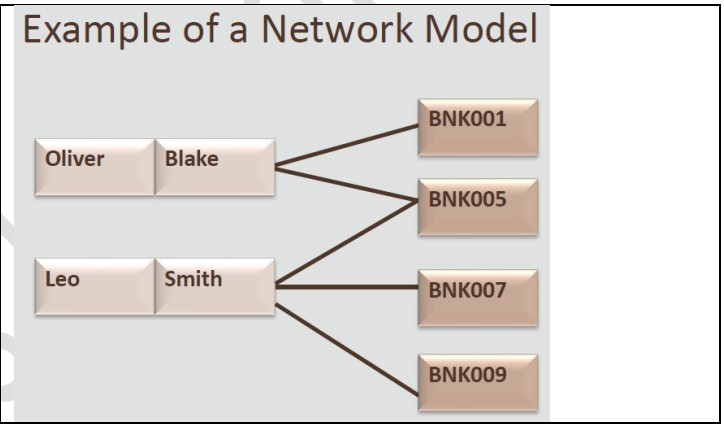
→

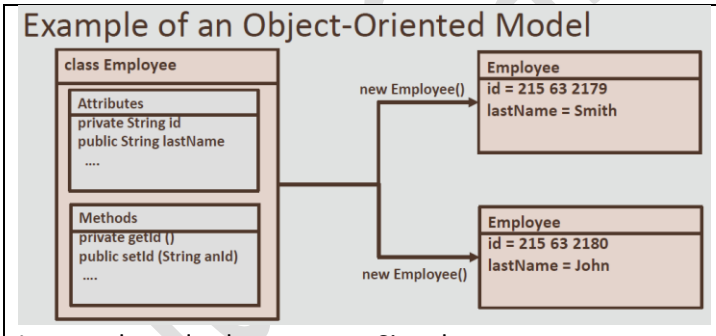
1.3. Types of Database Models





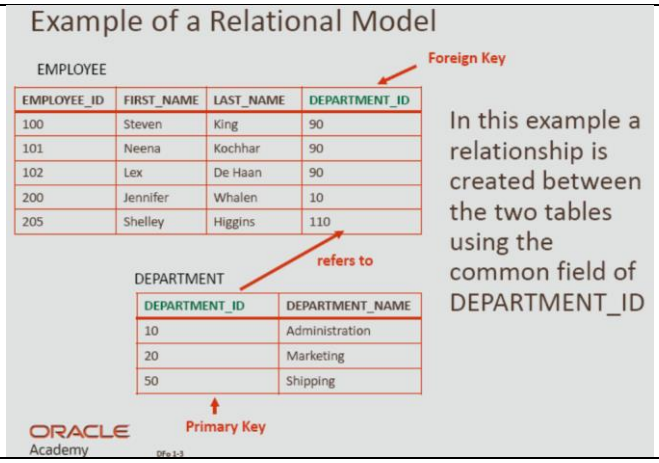






Los nombres de clases son en Singular

Los nombres de las tablas son en Plural



→

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:

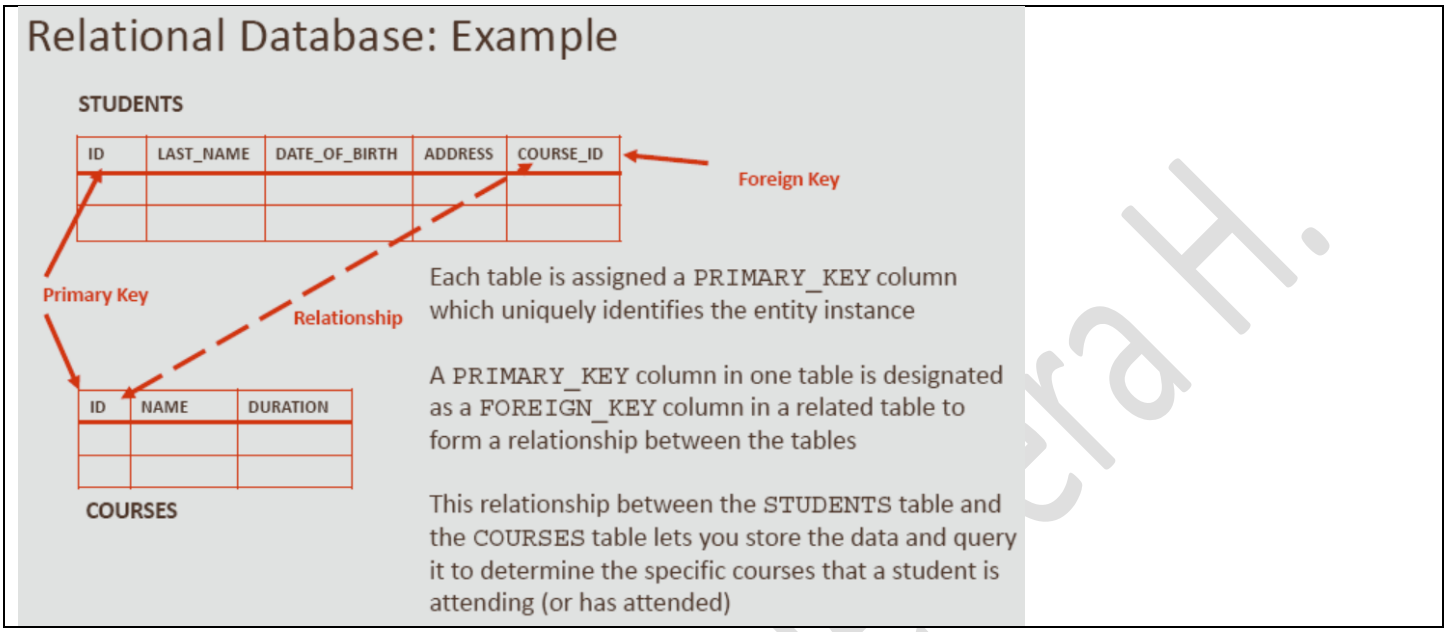
Note	Business Rule	Assumption	Problem
To ensure that new book arrivals happen on the 21st of every month.			
Librarian cannot easily identify DVDs that are seriously overdue (more than two weeks late).			
Our current system probably uses Oracle Database 10g and is on UNIX.			

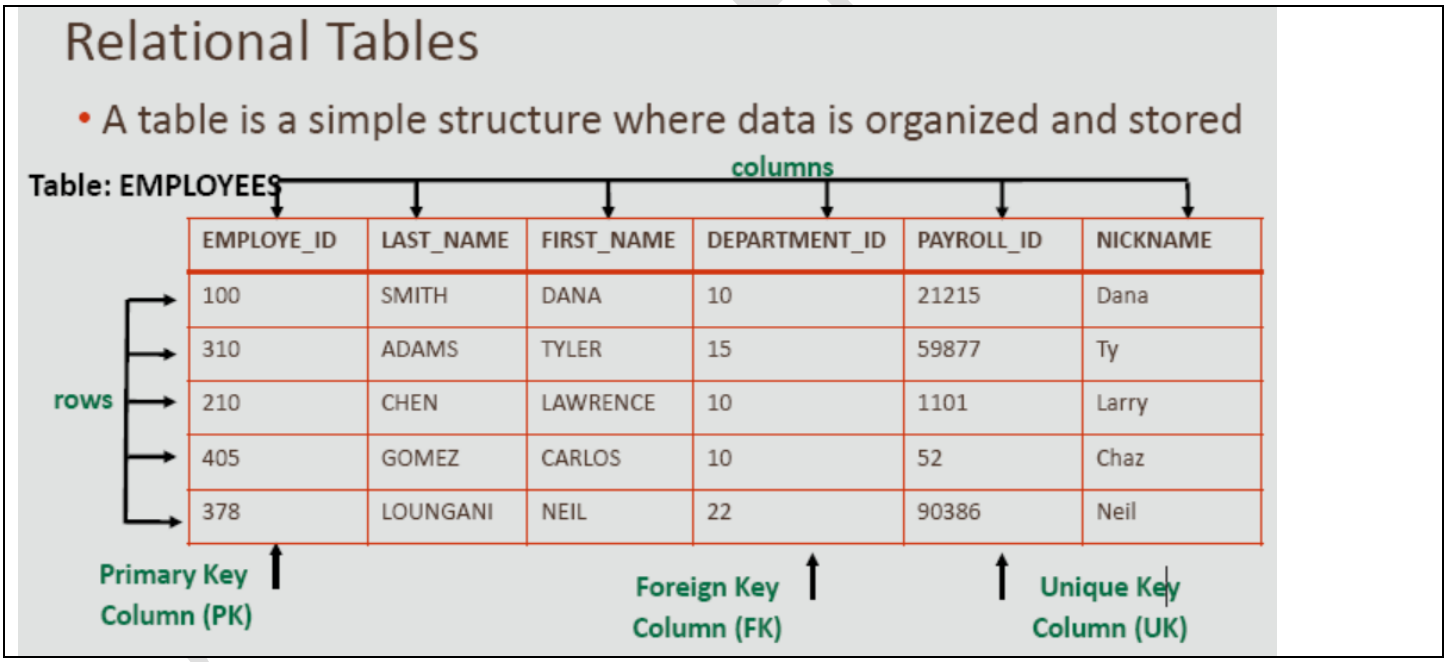
Identify the statements as a business rule, a problem, or an assumption.

→

2. Databases and Data Modeling

2.1. Relational Databases





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

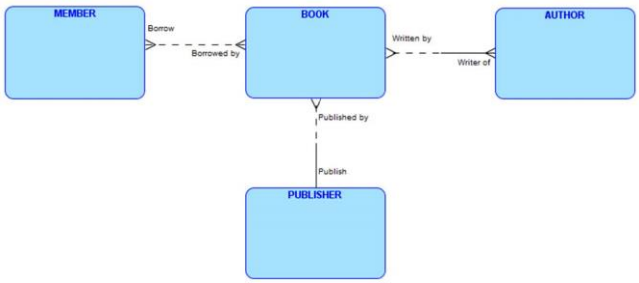
Key Terms

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



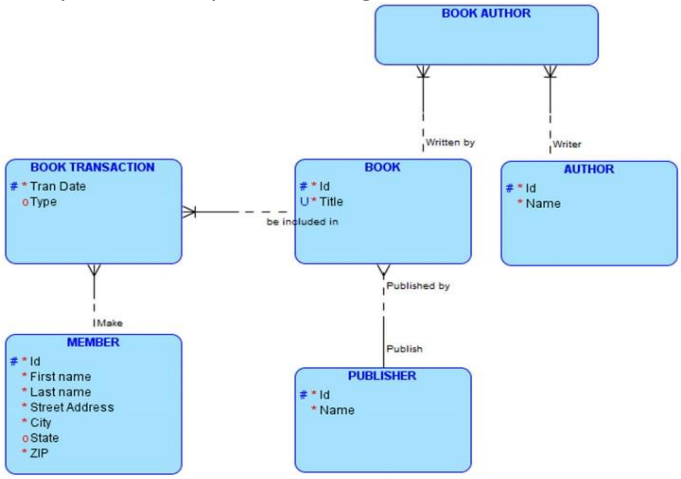
Identifies :

- important entities
- relationships among entities

Does not specify :

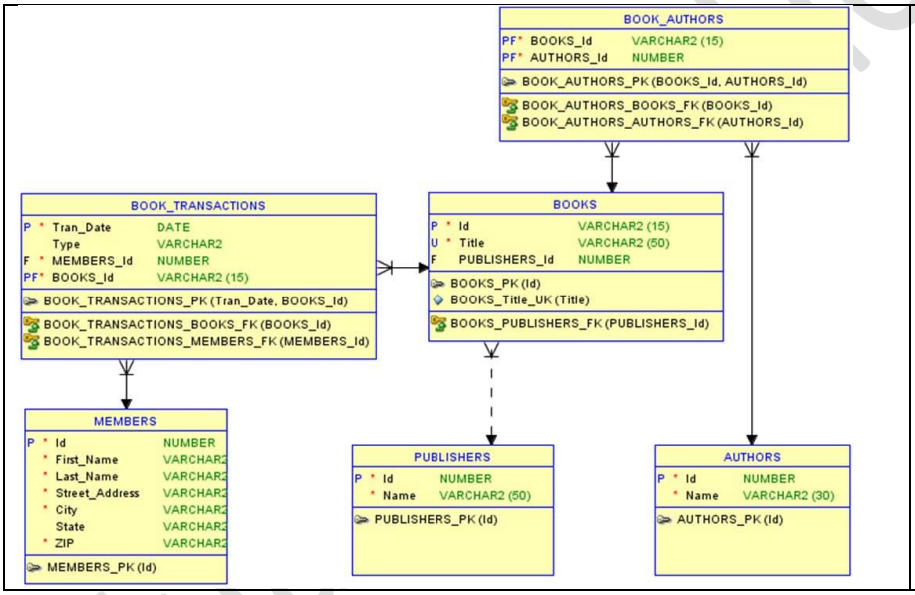
- attributes
- unique identifiers (primary key)

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

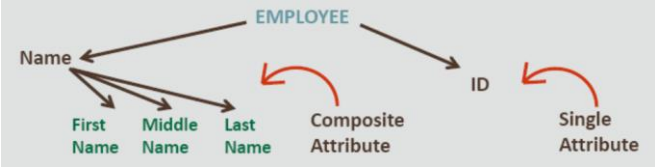
An entity can be classified as one of the following types:

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

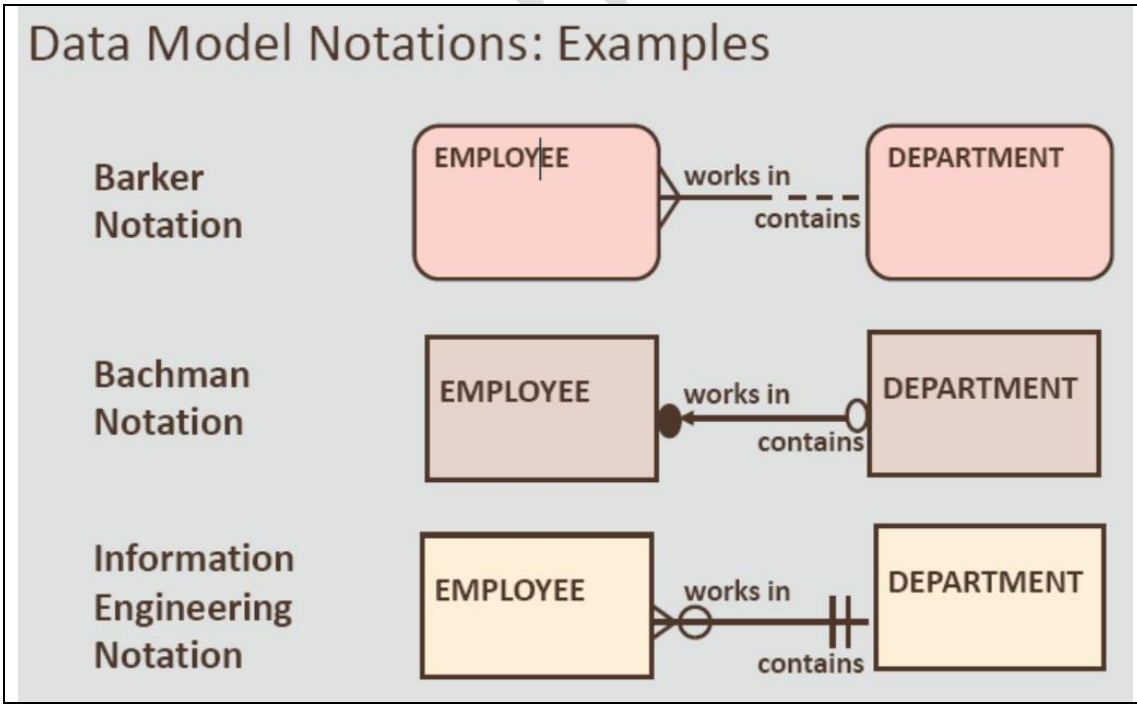
Tipos de Entidad:
- Principal
- Caracteristica
- Interseccion

Entidades:
- Fuertes
- Débiles

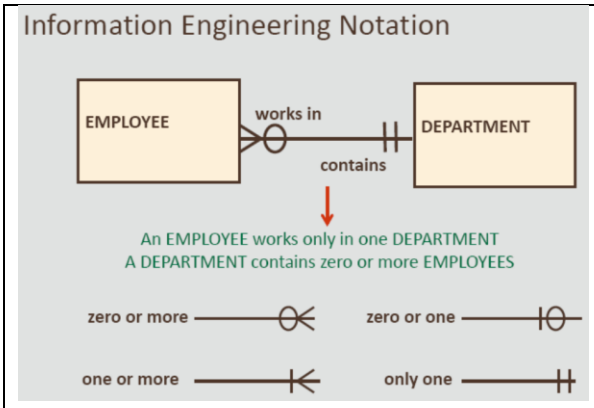
Single(atomic) and Composite(subparts) Attributes

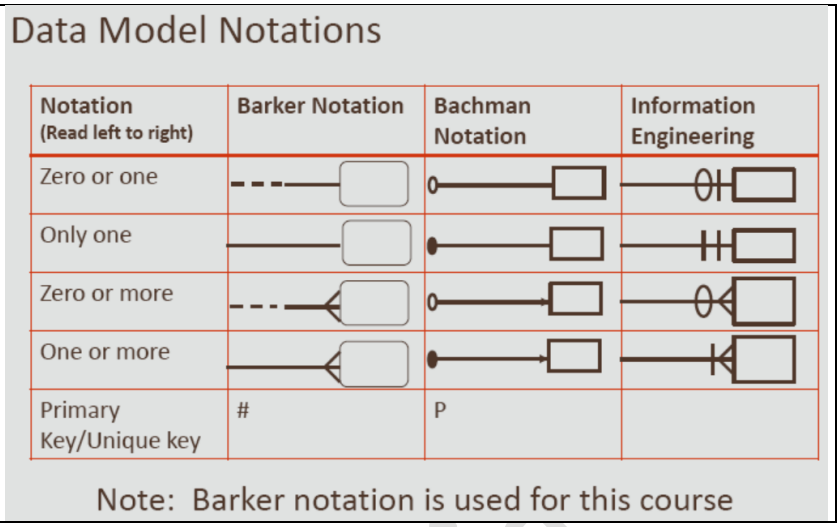


Single-Valued and Multi-Valued Attributes

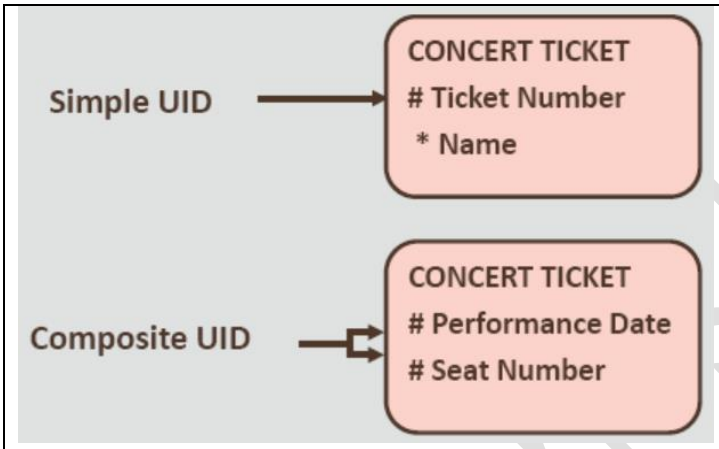


Black: debe
White: puede
Arrow: many





2.4. Unique Identifiers



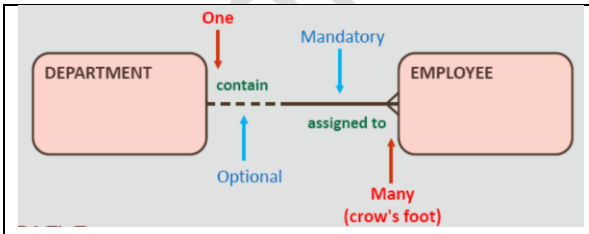
Each entity must have a unique identifier
Otherwise, it is not an entity

Candidate unique identifiers or secondary UIDs, so define primary key

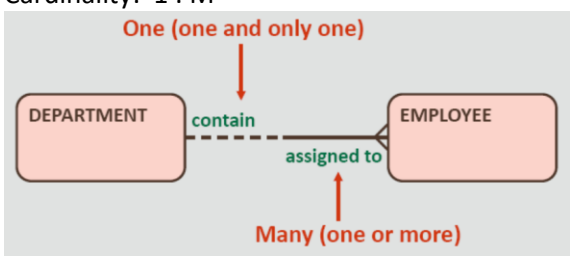
It has a non-Null
It has a value that never changes

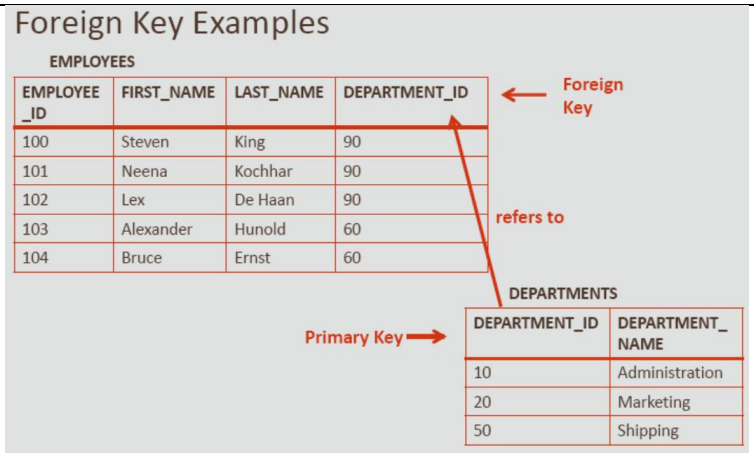
Artificial UIDs do not occur in the natural world but are created for identification purposes in a system
Example Composite UID: Bank_No and Account_No.

2.5. Relationships $$



Cardinality: 1 : M



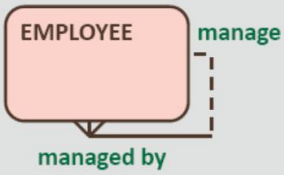


Cardinality M : N

Cardinality 1 : 1

Recursive Relationships

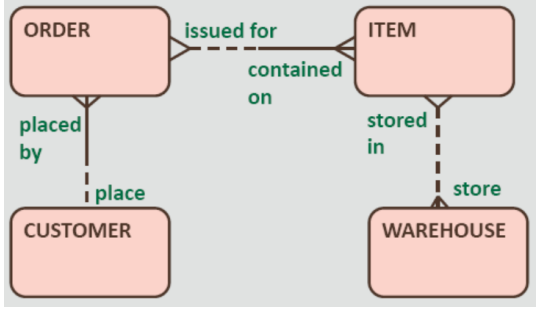
- A recursive relationship is a relationship with an entity and itself



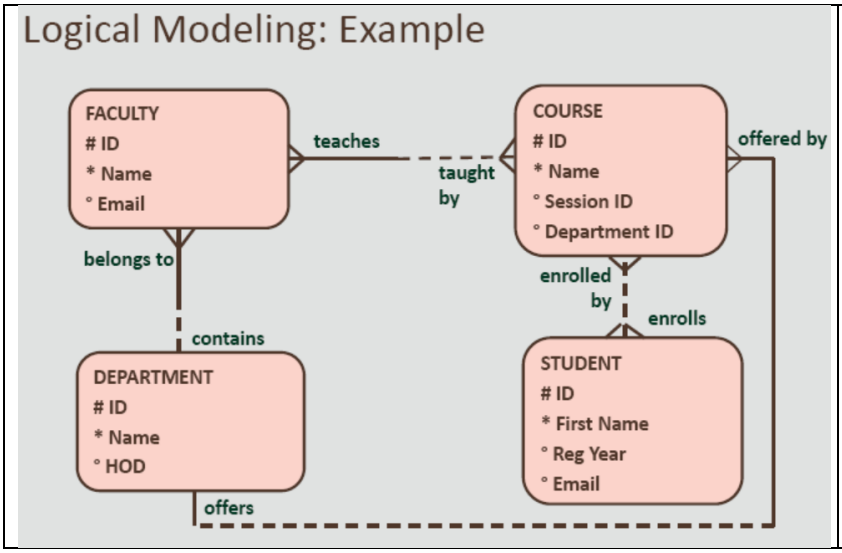
- Business rules:
  - Each EMPLOYEE may manage one or more EMPLOYEE
  - Each EMPLOYEE must be managed by one and only one EMPLOYEE

Relationship Matrix: Mapping the Contents

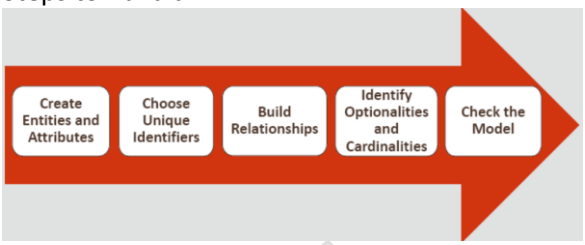
	CUSTOMER	ITEM	ORDER	WAREHOUSE
CUSTOMER			place	
ITEM			contained on	stored in
ORDER	placed by	issued for		
WAREHOUSE		store		



2.6. Entity Relationship Modeling (ERDs)

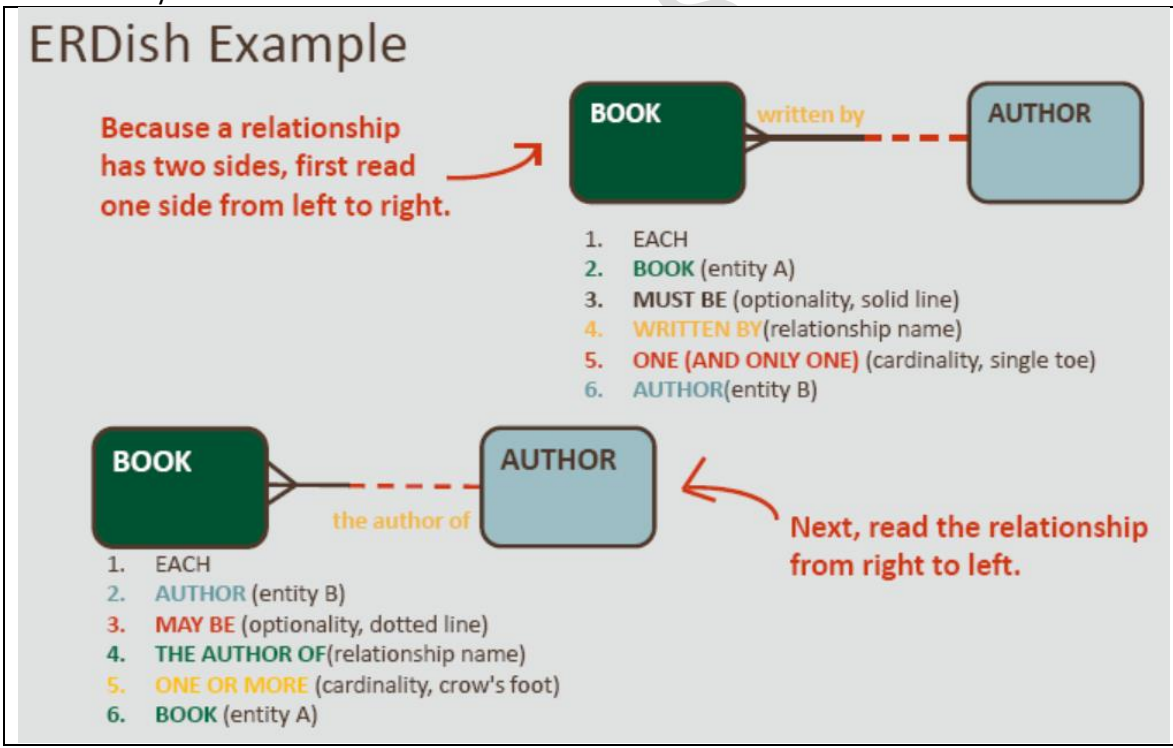


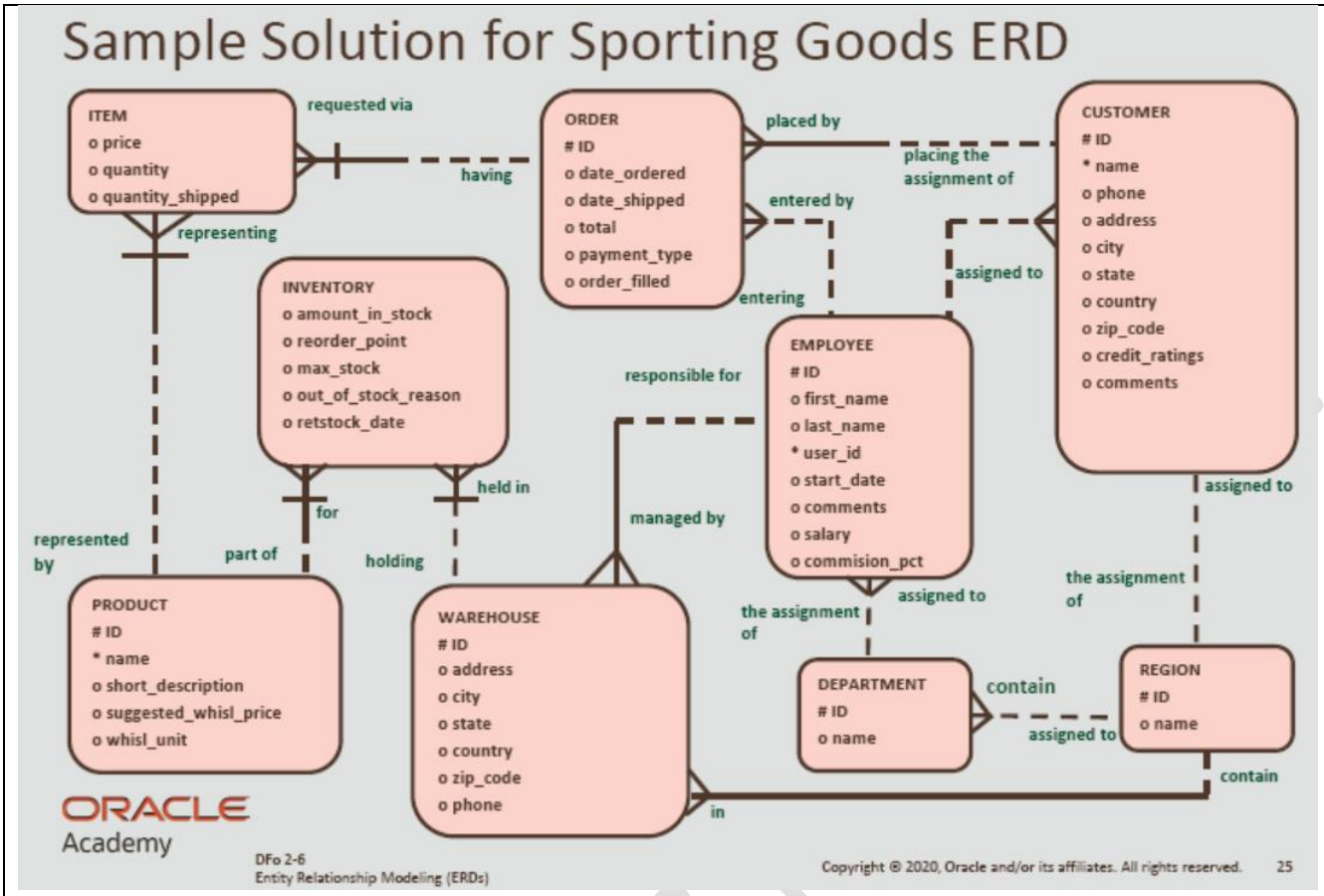
Steps to Build an ERD



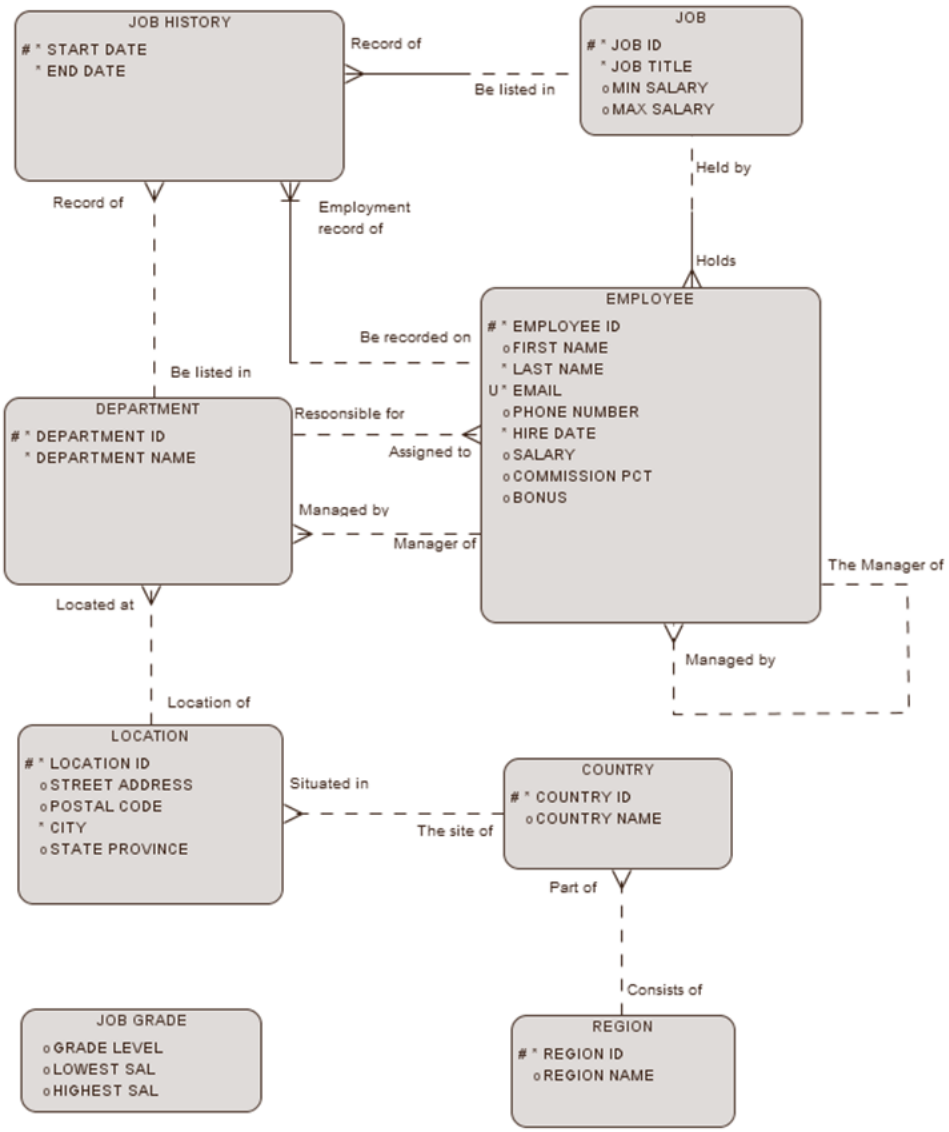
Components of ERDish

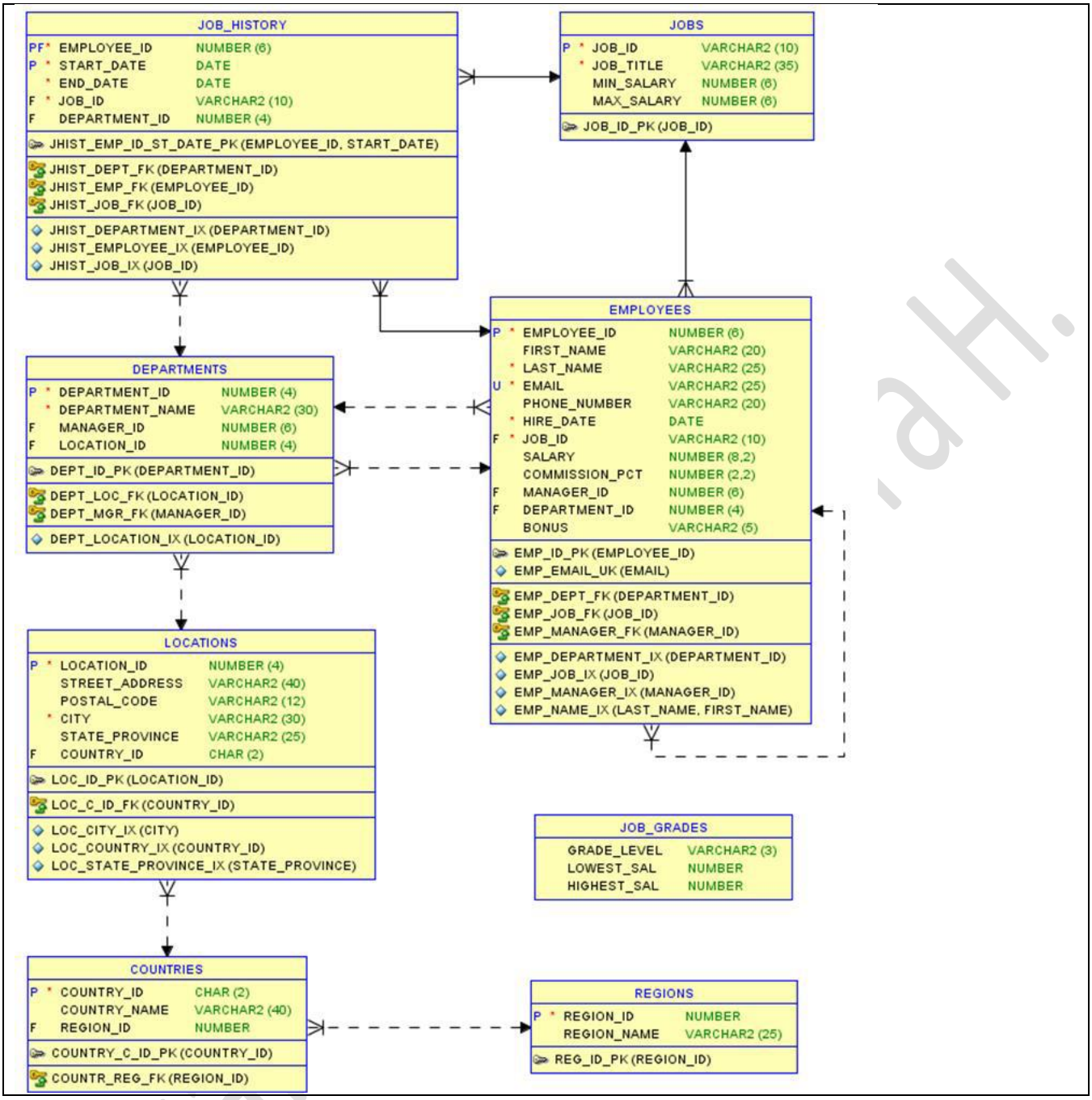
- EACH
- Entity A
- OPTIONALITY (must be/may be)
- RELATIONSHIP NAME
- CARDINALITY (one and only one/ one or more)
- Entity B




Sample Solution for Sporting Goods ERD
ITEM
o price
o quantity
o quantity_shipped
requested via
having
representing
ORDER
ID
o date_ordered
o date_shipped
o total
o payment_type
o order_filled
placed by
placing the assignment of
entered by
entering
CUSTOMER
ID
* name
o phone
o address
o city
o state
o country
o zip_code
o credit_ratings
o comments
assigned to
INVENTORY
o amount_in_stock
o reorder_point
o max_stock
o out_of_stock_reason
o retstock_date
for
held in
represented by
part of
holding
responsible for
managed by
EMPLOYEE
ID
o first_name
o last_name
* user_id
o start_date
o comments
o salary
o commision_pct
assigned to
the assignment of
assigned to
the assignment of
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
contain
assigned to
REGION
ID
o name
contain
in
ORACLE
Academy
DFo 2-6
Entity Relationship Modeling (ERDs)
Copyright © 2020, Oracle and/or its affiliates. All rights reserved.
25

→




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

→