DATABASE FOUNDATIONS

ORACLE ACADEMY

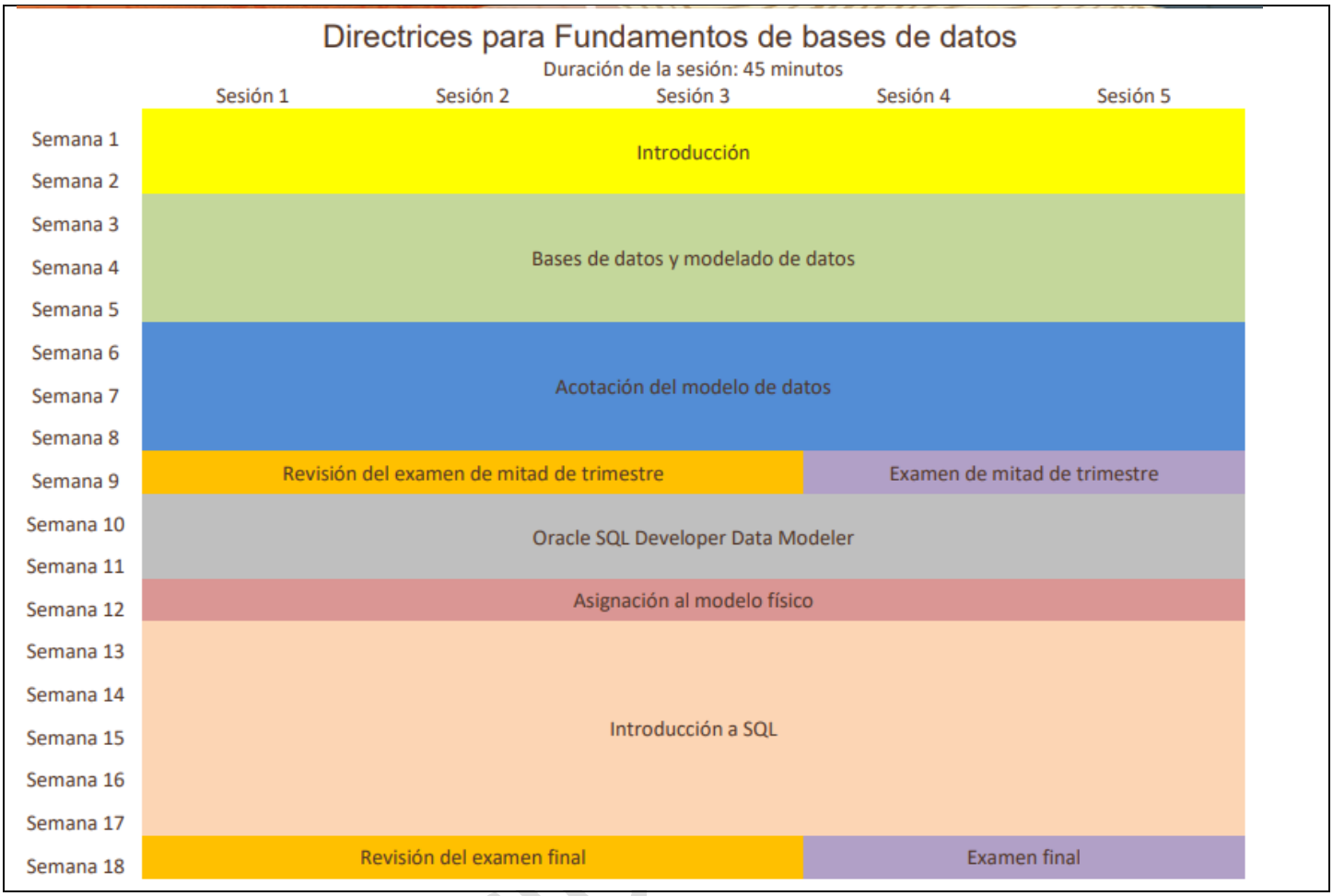
6 DE MAYO DE 2025

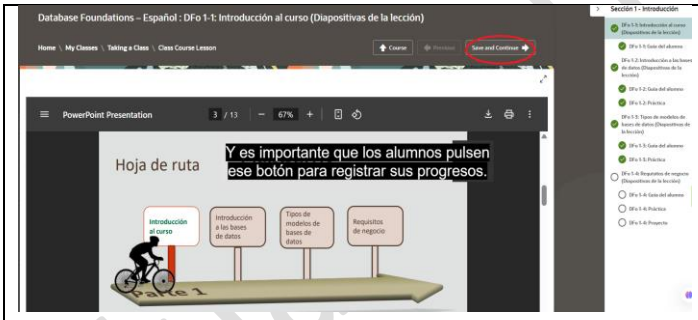
https://academy.oracle.com/

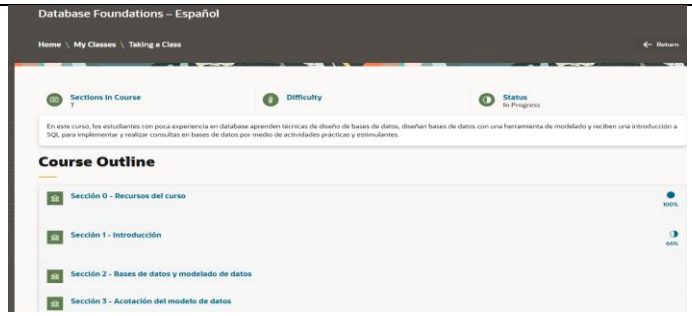
HTTPS://GITHUB.COM/ISC-UPA/2025-2-ISC05-DB

Contenido

1. Introduction

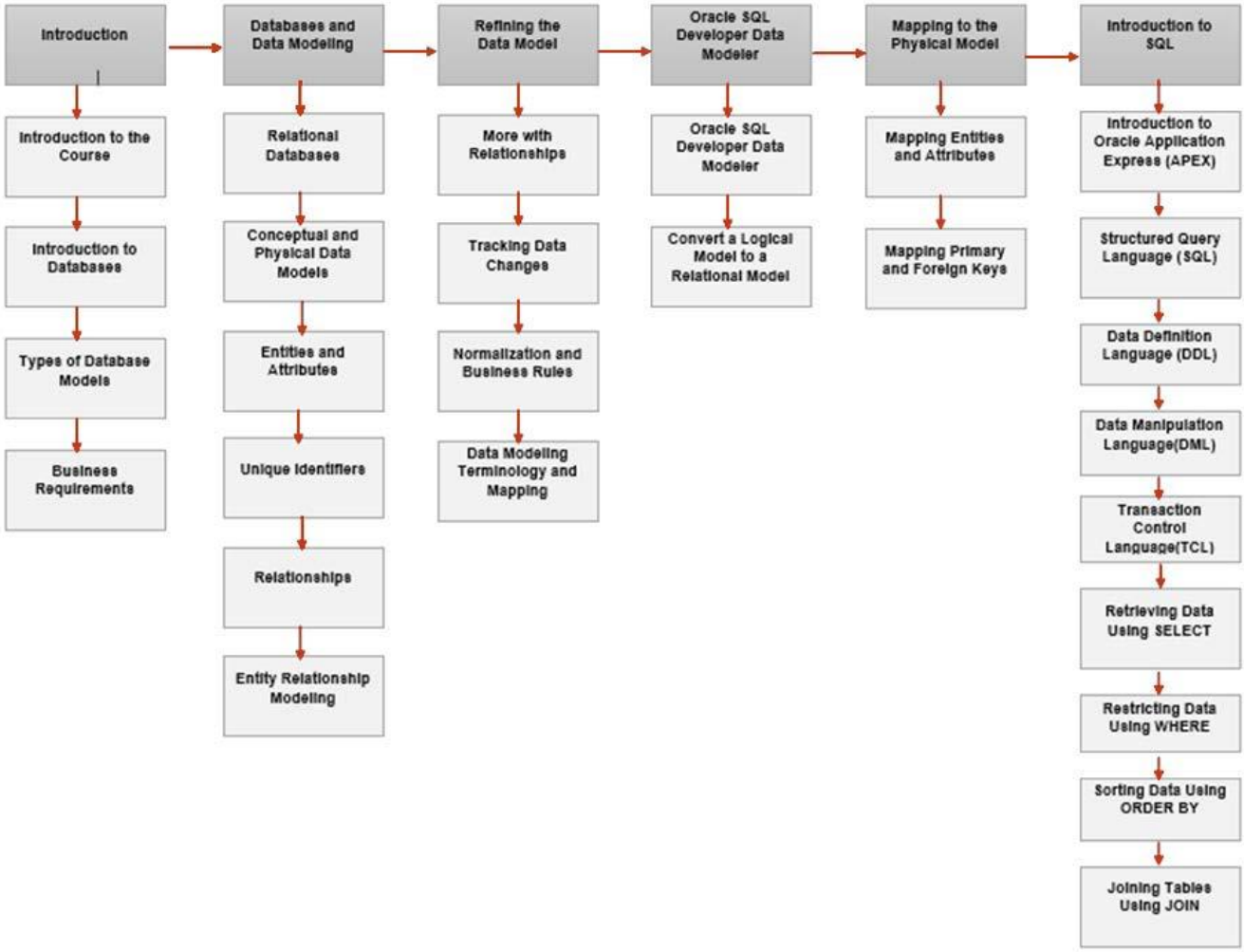






→

1.1. Introduction



Technological Requirements:

Oracle SQL Developer or Oracle APEX application
Oracle Data Modeler

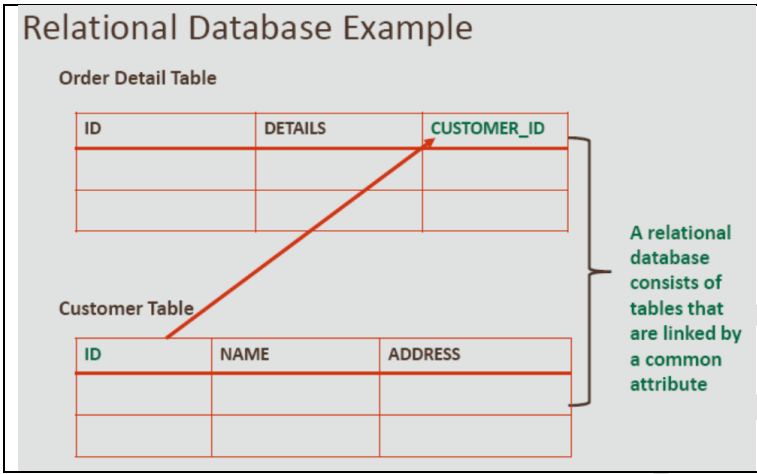
→

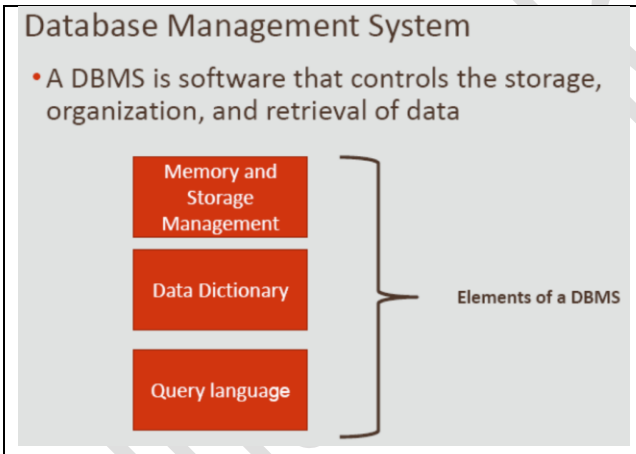
1.2. Introduction to Databases

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

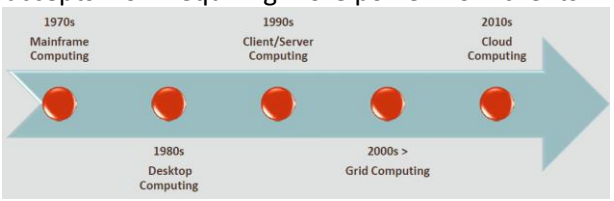
- A relational database stores information in tables with rows and columns
- A table is a collection of records
- A row is called a record (or instance)
- A record is a collection of fields
- A column is referred to as a field (or attribute)





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

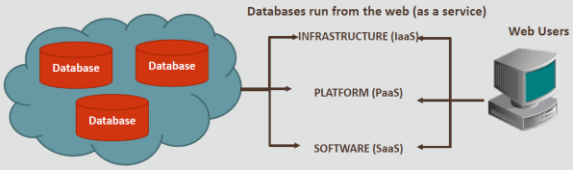


2000s: Grid Computing (Shared Processing)

- 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

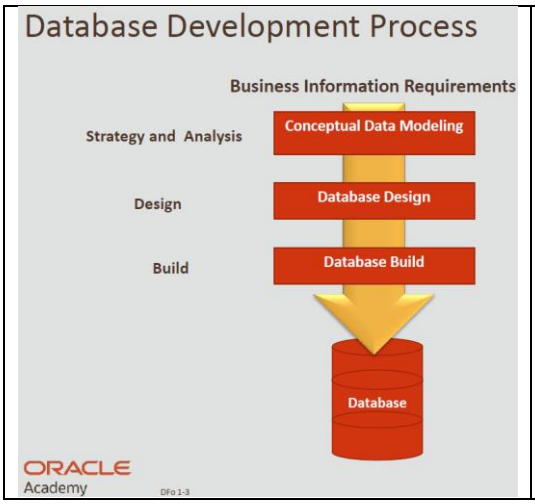
2010s: Cloud Computing (Internet Based Processing)

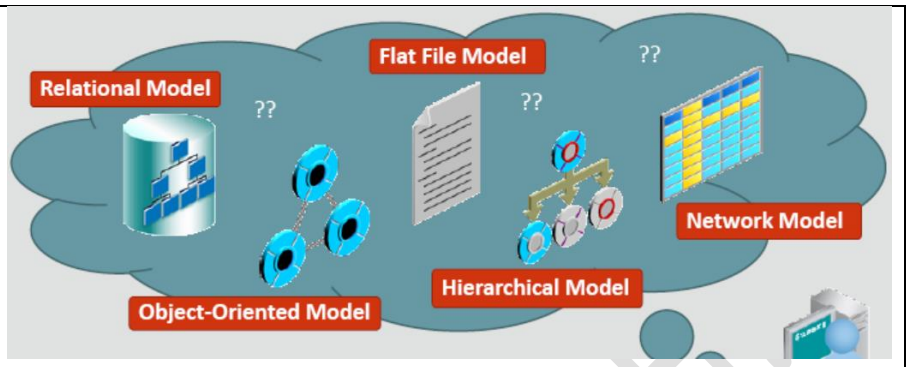


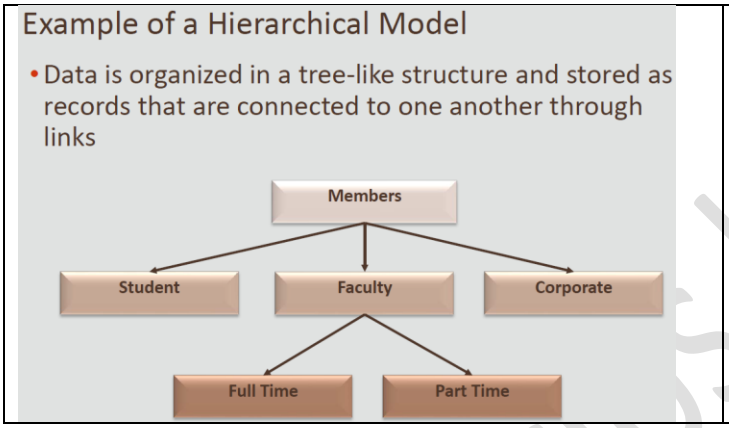
- Cloud computing allows the delivery of computing services over the Internet
- The three main categories of cloud services are:
  - IaaS – Allows you to rent cloud based servers, storage, operating systems etc
  - PaaS – Gives access to an online environment for developing and testing software without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

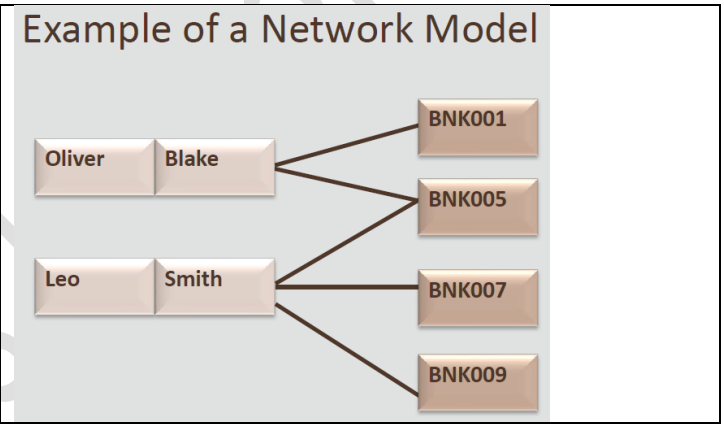
→

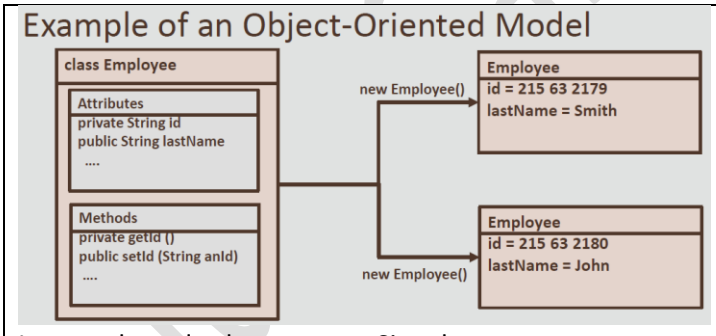
1.3. Types of Database Models





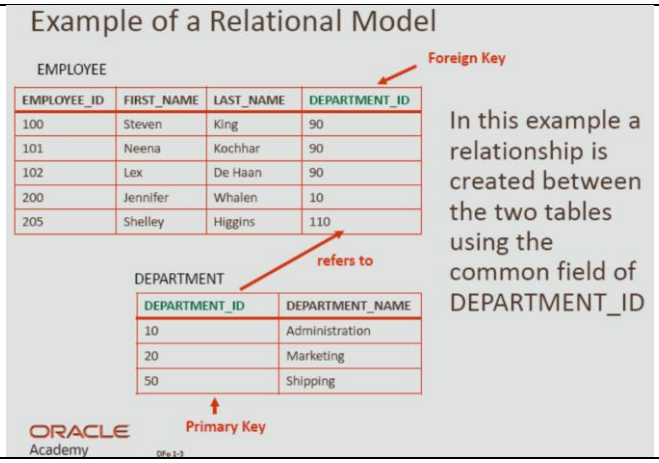






Los nombres de clases son en Singular

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:

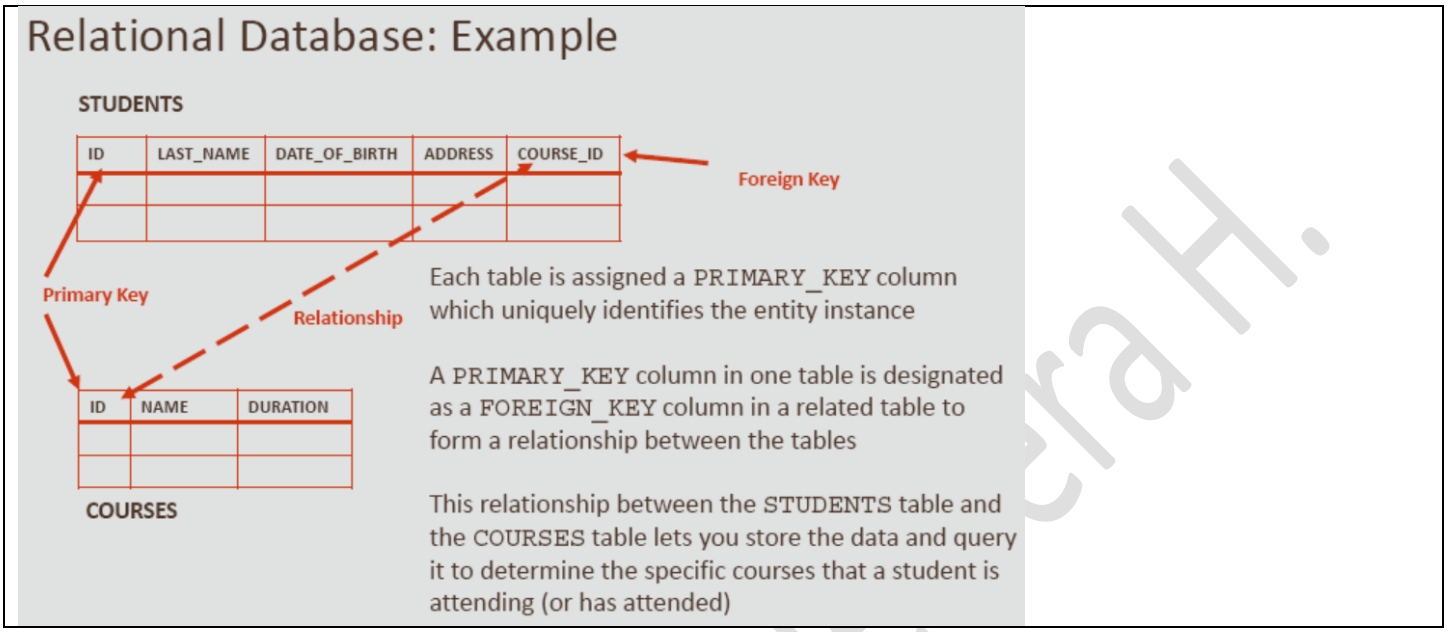
Note	Business Rule	Assumption	Problem
To ensure that new book arrivals happen on the 21st of every month.			
Librarian cannot easily identify DVDs that are seriously overdue (more than two weeks late).			
Our current system probably uses Oracle Database 10g and is on UNIX.			

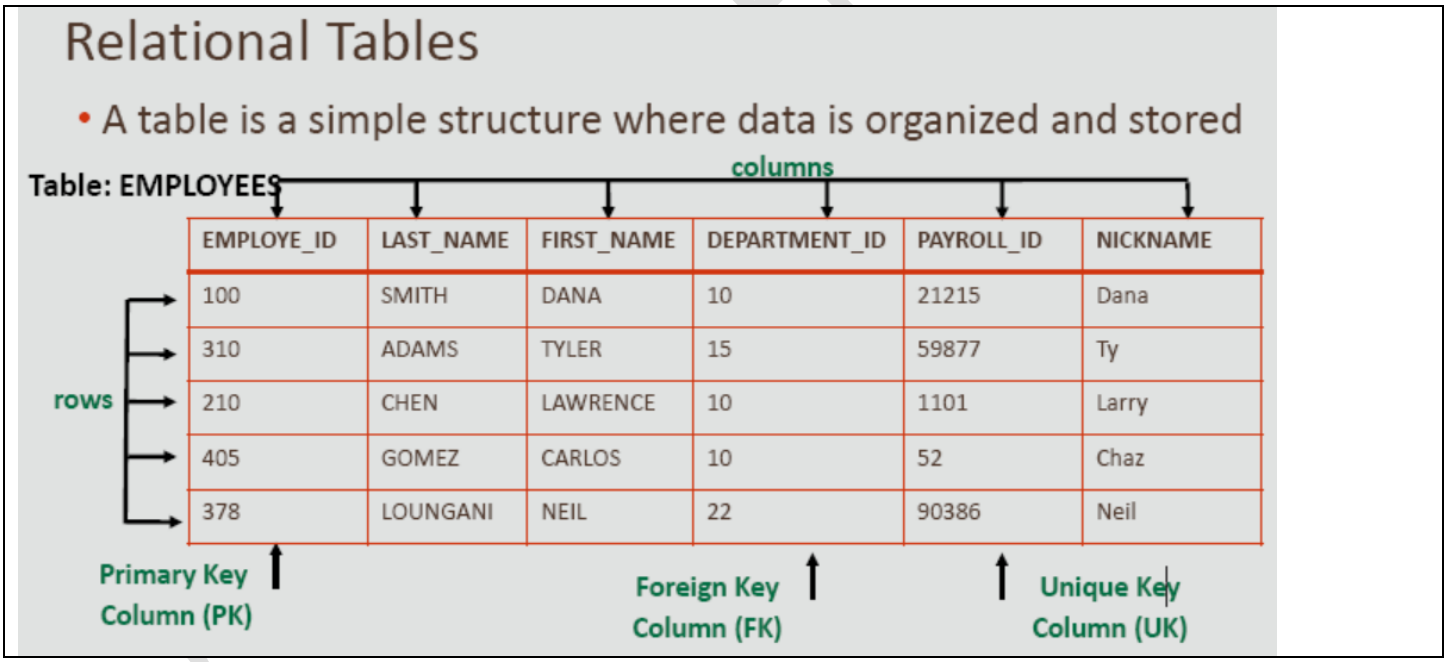
Identify the statements as a business rule, a problem, or an assumption.

→

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

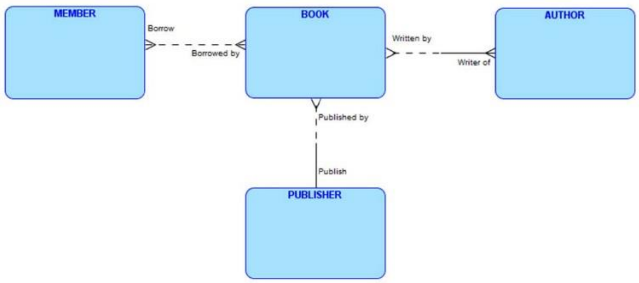
Key Terms

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



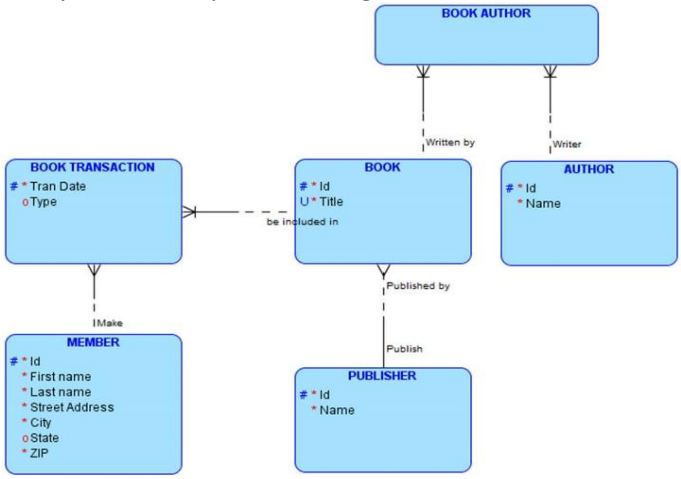
Identifies :

- important entities
- relationships among entities

Does not specify :

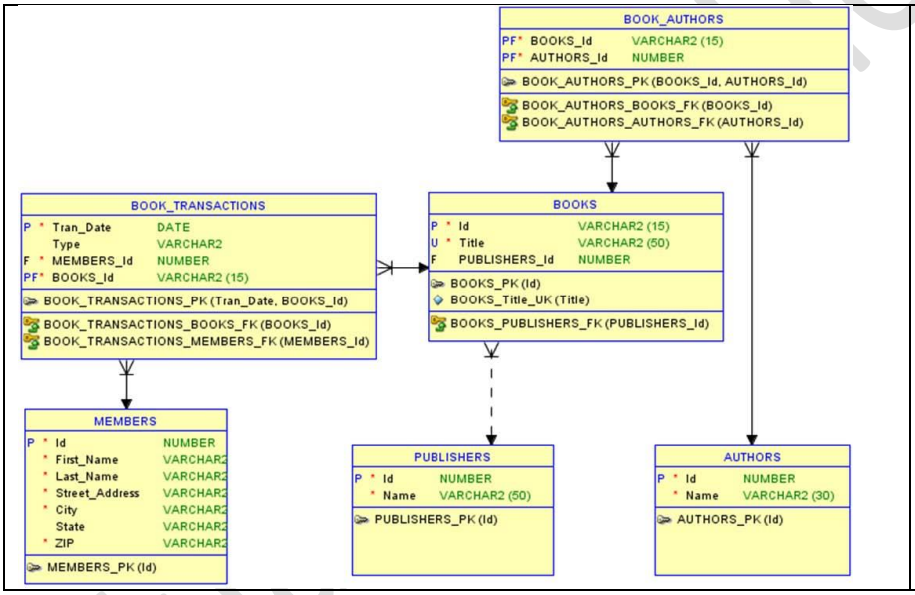
- attributes
- unique identifiers (primary key)

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

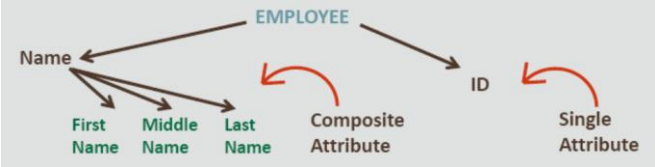
An entity can be classified as one of the following types:

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

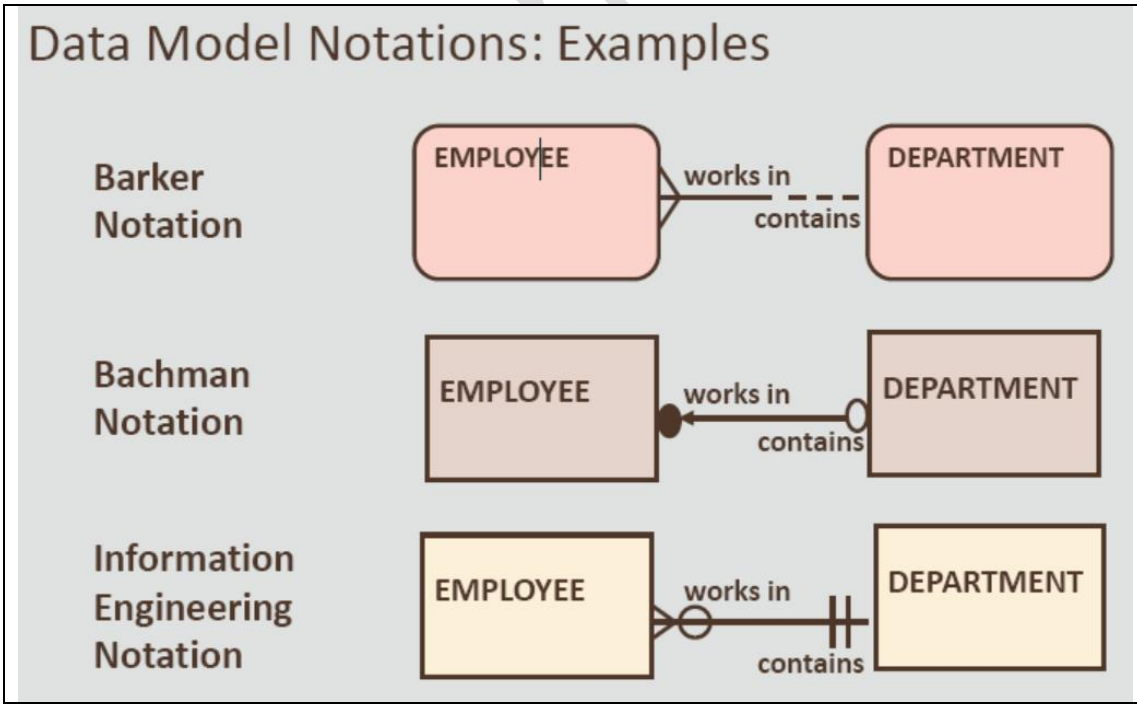
Tipos de Entidad:
- Principal
- Caracteristica
- Interseccion

Entidades:
- Fuertes
- Débiles

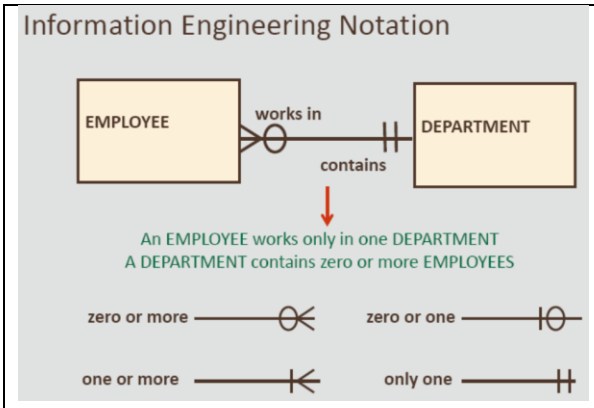
Single(atomic) and Composite(subparts) Attributes

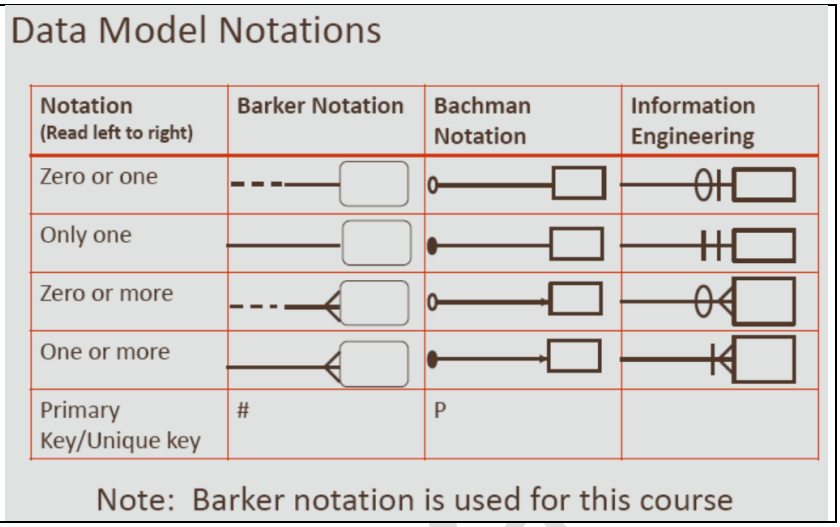


Single-Valued and Multi-Valued Attributes

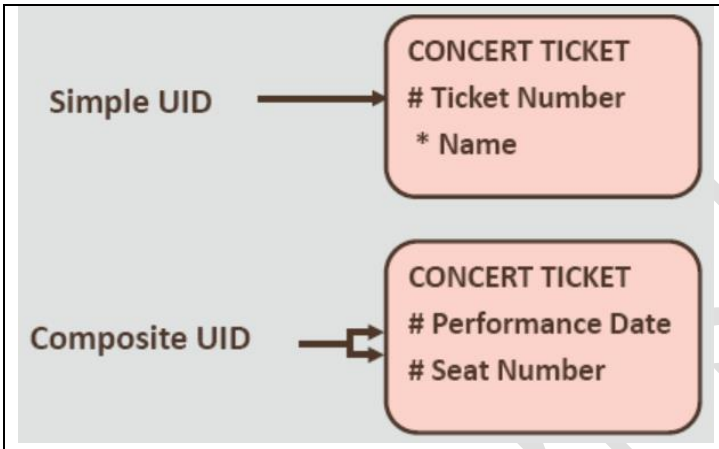


Black: debe
White: puede
Arrow: many

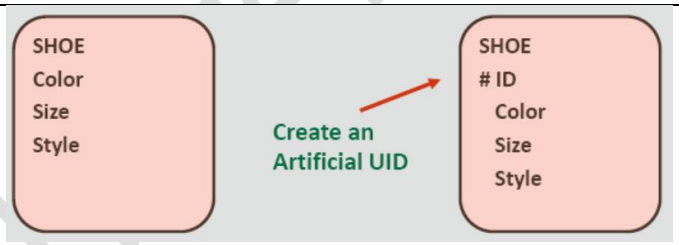




2.4. Unique Identifiers



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

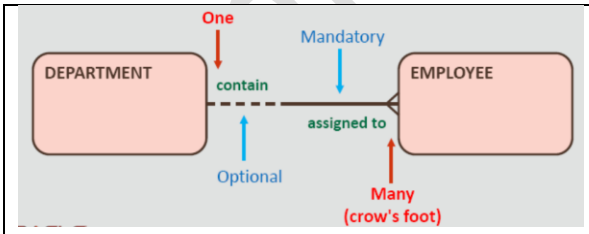


Candidate unique identifiers or secondary UIDs, so define primary key

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.

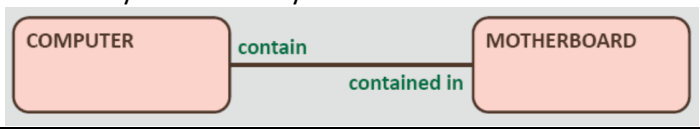
2.5. Relationships



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."

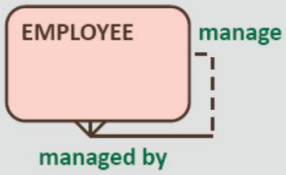
Relationship Types or Cardinality

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



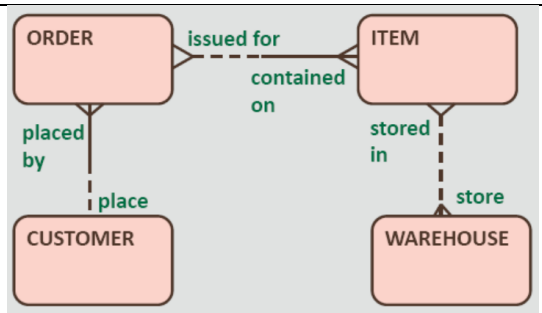
- Business rules:
  - Each EMPLOYEE may manage one or more EMPLOYEE
  - Each EMPLOYEE must be managed by one and only one EMPLOYEE

	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



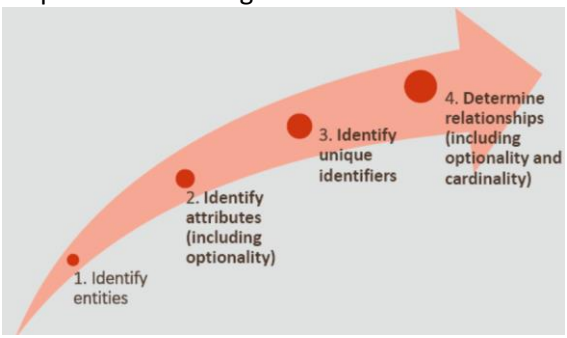
→

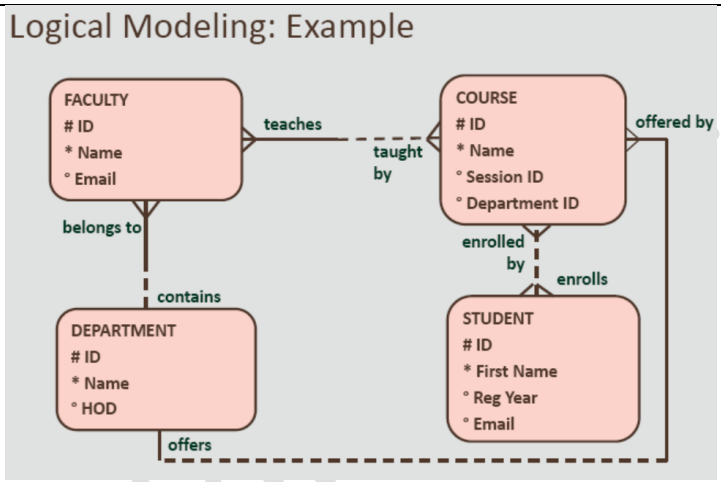
2.6. Entity Relationship Modeling (ERDs)

DB roles: designers, database administrators, and application developers

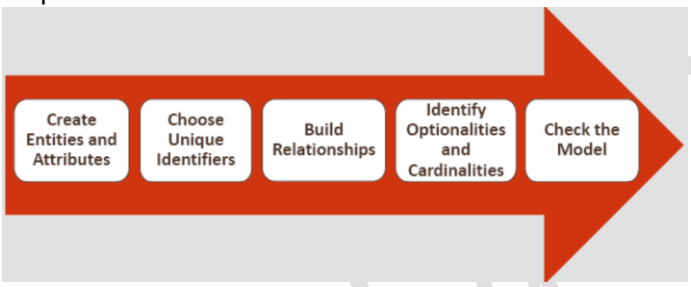
Logical Modeling:
Includes all entities, attributes, UIDs and relationships as well as optionality and cardinality of these ítems

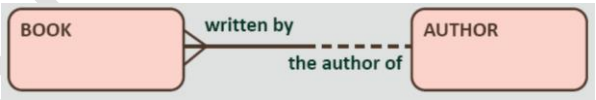
Steps to Create a Logical Model





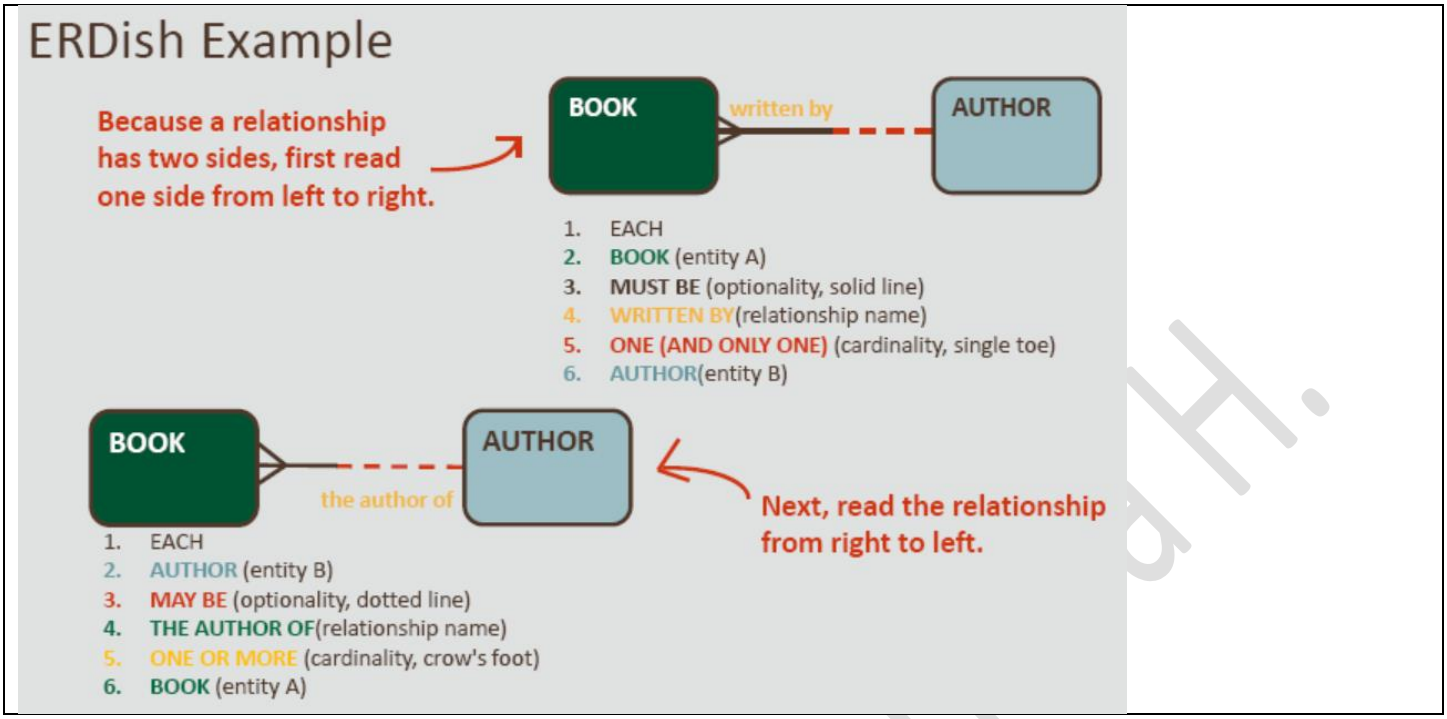
Steps to Build an ERD

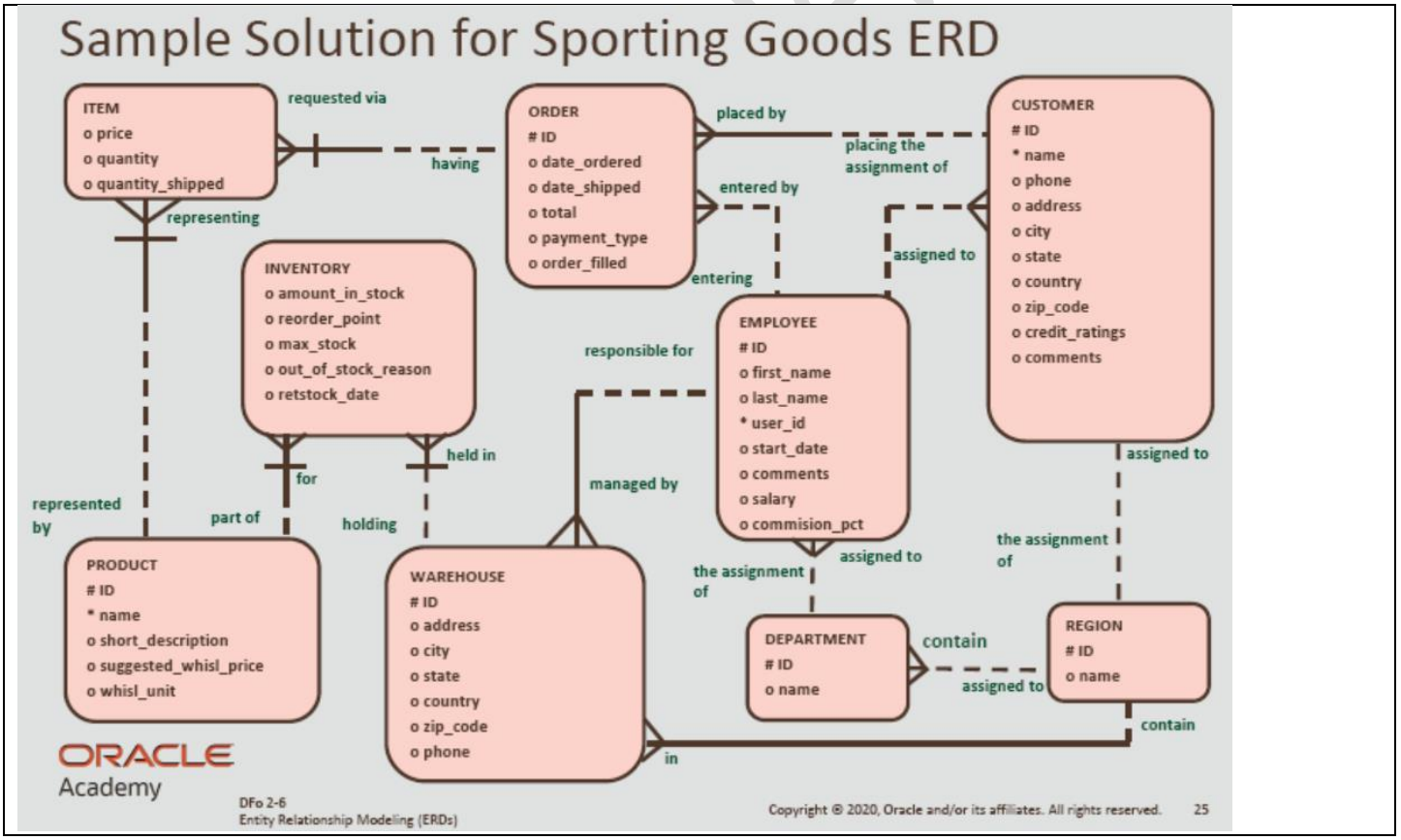




Components of ERDish

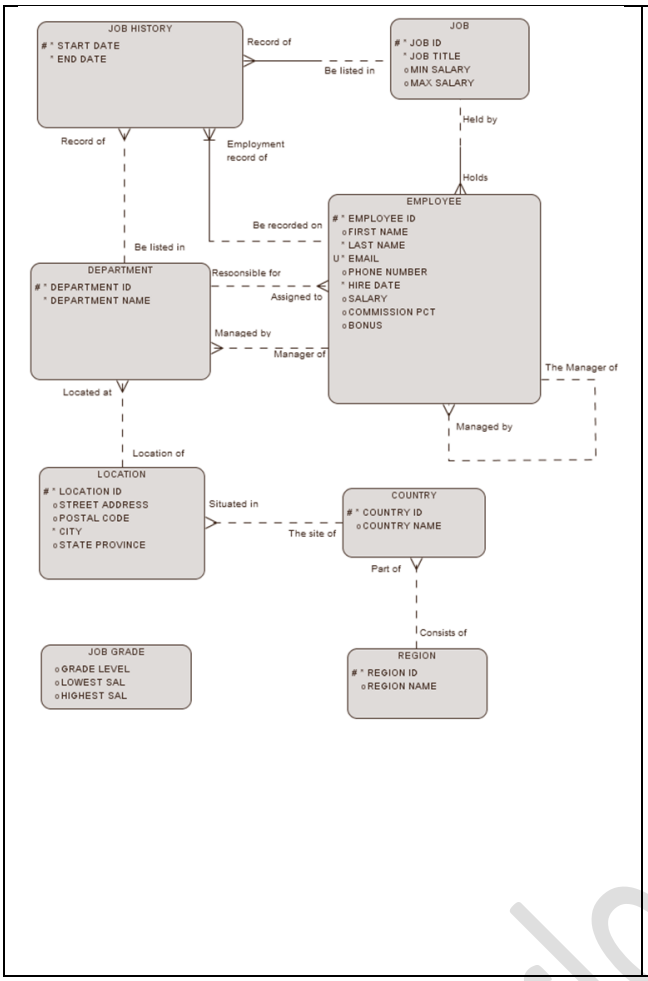
- EACH
- Entity A
- OPTIONALITY (must be/may be)
- RELATIONSHIP NAME
- CARDINALITY (one and only one/ one or more)
- Entity B


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

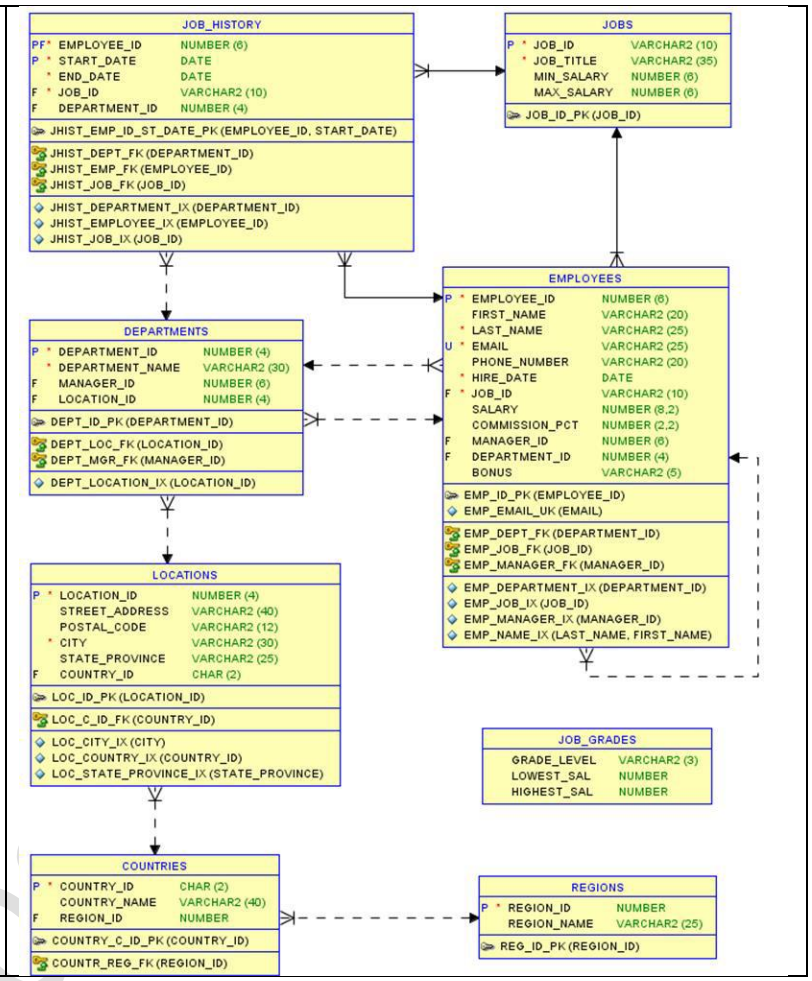

Sample Solution for Sporting Goods ERD
ITEM
o price
o quantity
o quantity_shipped
requested via
having
representing
ORDER
ID
o date_ordered
o date_shipped
o total
o payment_type
o order_filled
placed by
placing the assignment of
entered by
entering
CUSTOMER
ID
* name
o phone
o address
o city
o state
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
responsible for
EMPLOYEE
ID
o first_name
o last_name
* user_id
o start_date
o comments
o salary
o commision_pct
assigned to
represented by
part of
holding
managed by
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

→

Logical Data Model



Physical data Model

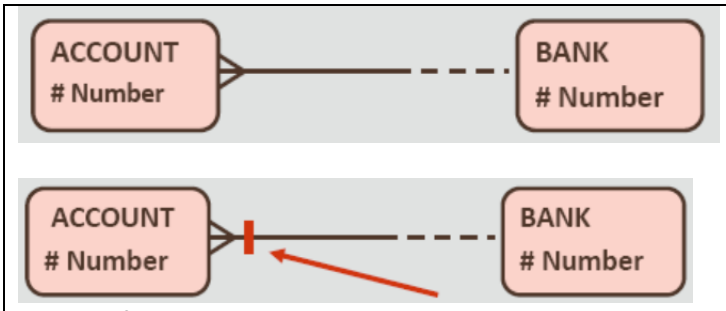


→

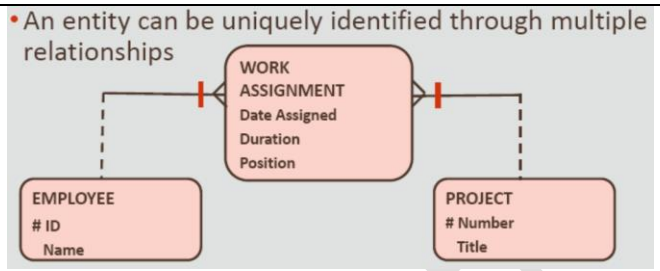
3. Refining the Data Model

3.1. More with Relationships

3.1.1. Identifying (| Barred) Relationships

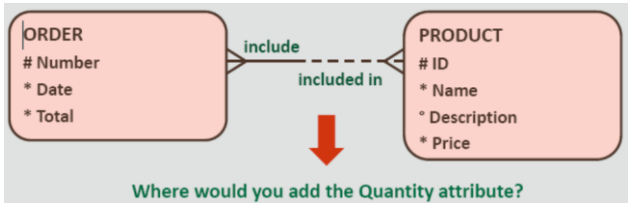


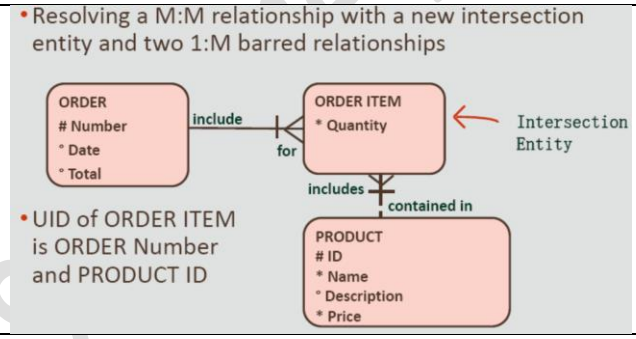
The UID for the ACCOUNT :
BANK Number , ACCOUNT Number



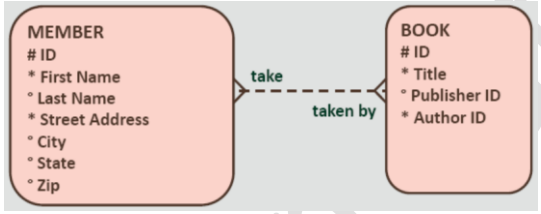
The UID for WORK ASSIGNMENT:
EMPLOYEE ID , PROJECT Number

3.1.2. M:M Relationships

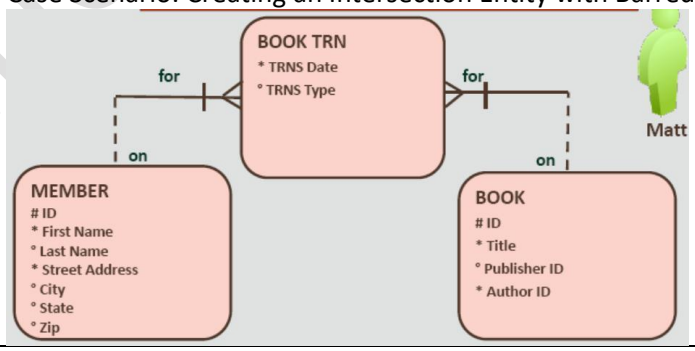




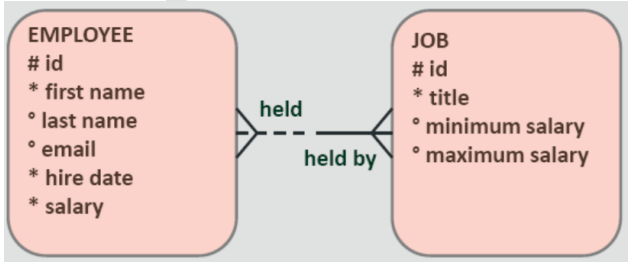
Case Scenario: Resolving M:M Relationships



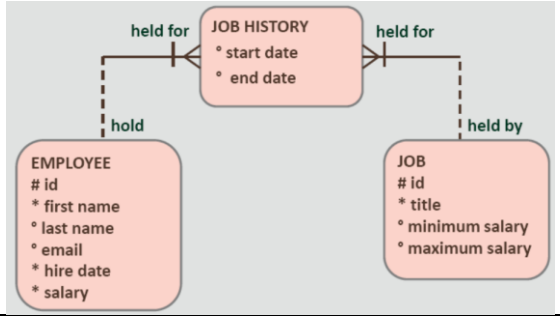
Case Scenario: Creating an Intersection Entity with Barred



Resolving M:M Relationships: Example 2

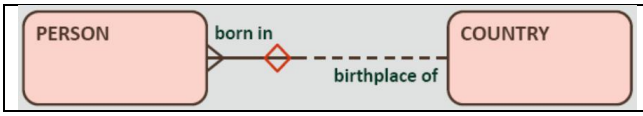


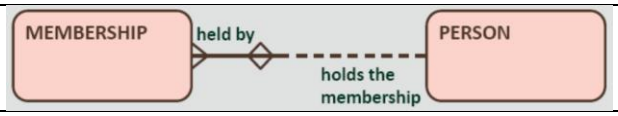
Resolving M:M Relationships Using Intersection



→

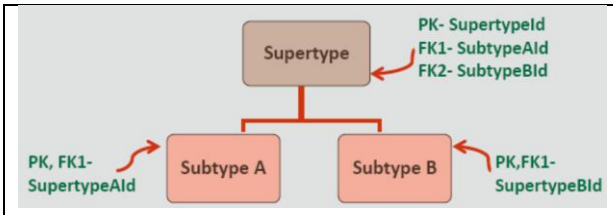
3.1.3. Non-Transferable Relationships ◇



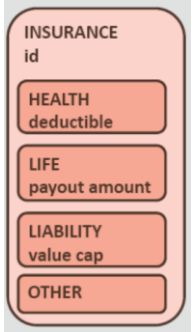


3.1.4. Supertype and Subtype Entities

- 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

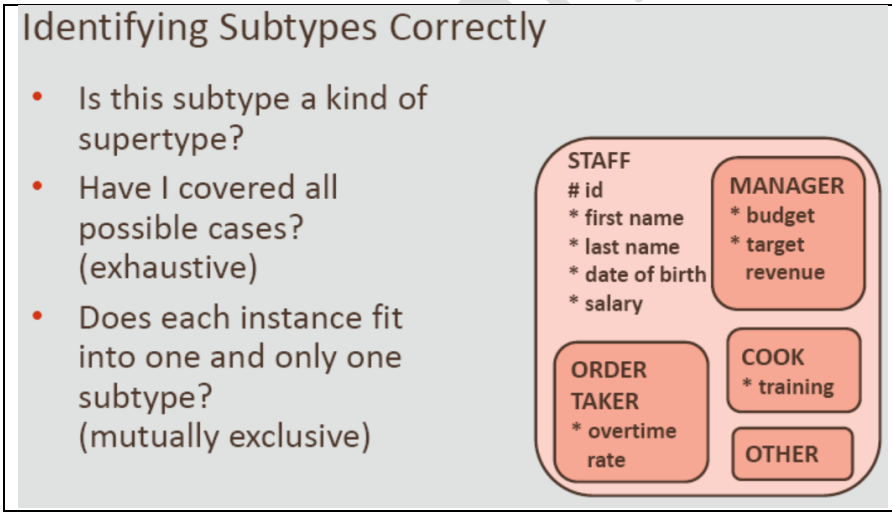


Characteristics of a Subtype

A subtype:

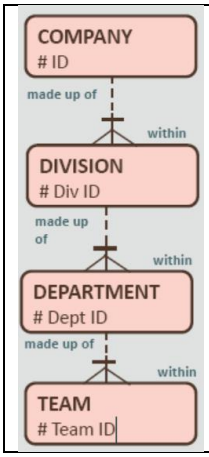
- Inherits all attributes of the supertype
- Inherits all relationships of the supertype
- Usually has its own attributes or relationships
- Is drawn within the supertype
- Never exists alone
- Has identical primary keys of the supertype and subtype

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



→

3.1.5. Modeling Hierarchical Data

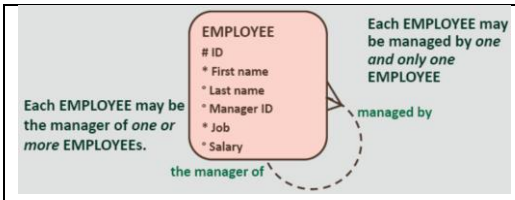


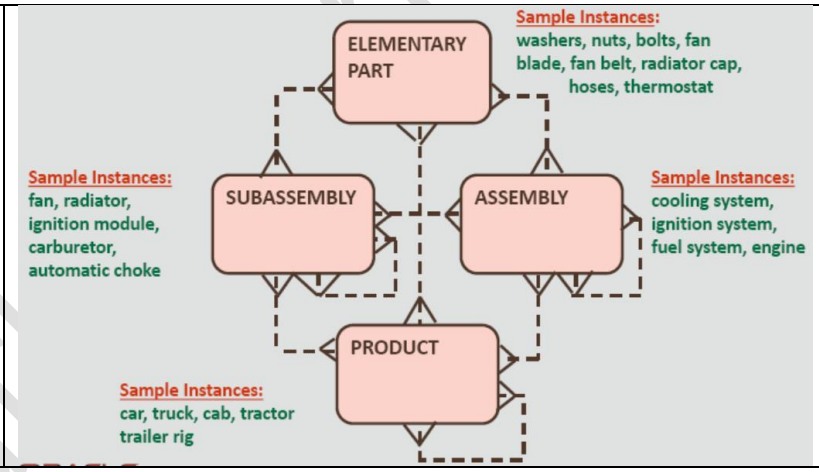
Resolver: La jerarquía del Hotel

En un hotel se tienen varias recamaras en cada departamento o suite(1,2 o 3), cuenta con 2 edificios, uno de 15 pisos y otro de 20 pisos.

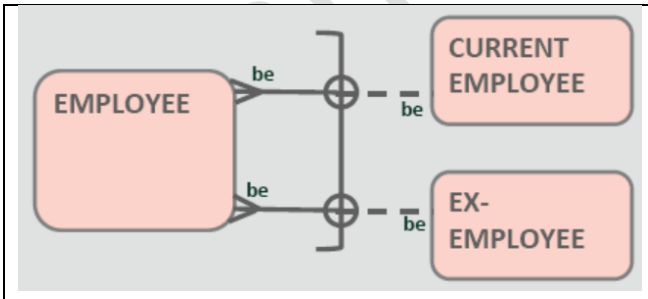
3.1.6. Recursive Relationships

- 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





3.1.7. Arc Relationship ⌒



An arc is an exclusive relationship group, which is defined such that only one of the relationships can exist for any instance of an entity
Exor

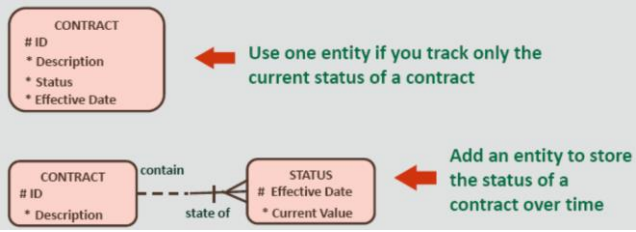
A supertype entity and its subtypes can be modeled as an arc relationship

→

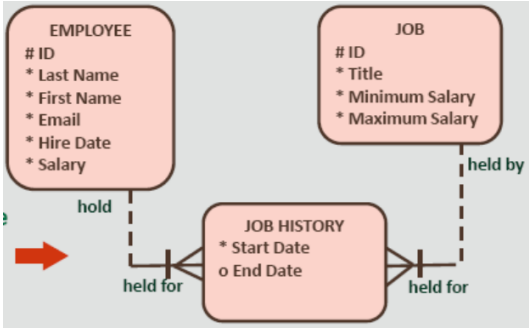
3.2. Tracking Data Changes

Objetive: Keep track of data that changes over time

Tracking Attribute Values

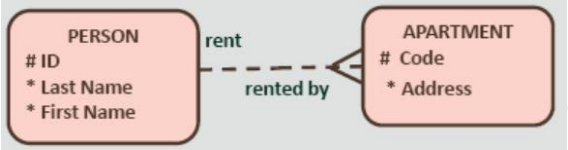


Creating Intersection Entities

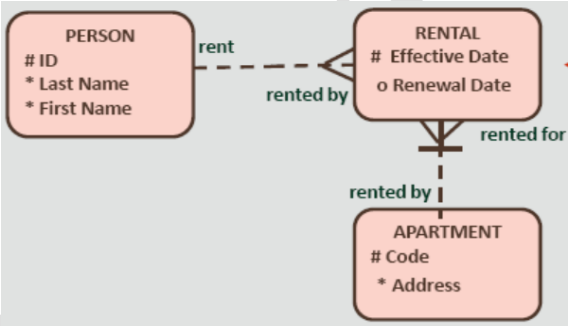


Accommodating a Changing Relationship

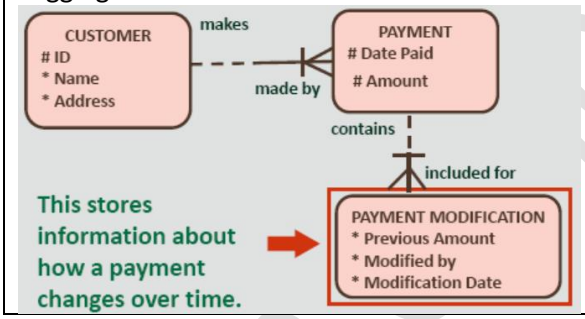
This does not allow tracking of the date that an apartment is rented.



Add an entity to capture the rental relationship over time

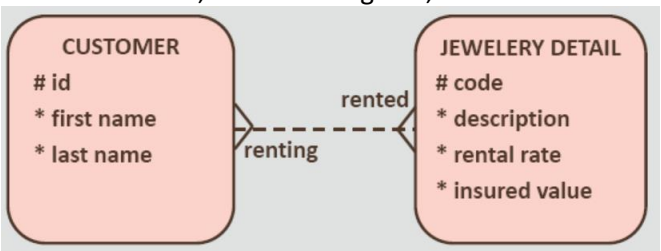


Logging Modified Information



Model Rental over Time: **Scenario**

Cuando se rentó, cuando lo regreso, cuanto le cobraron



→

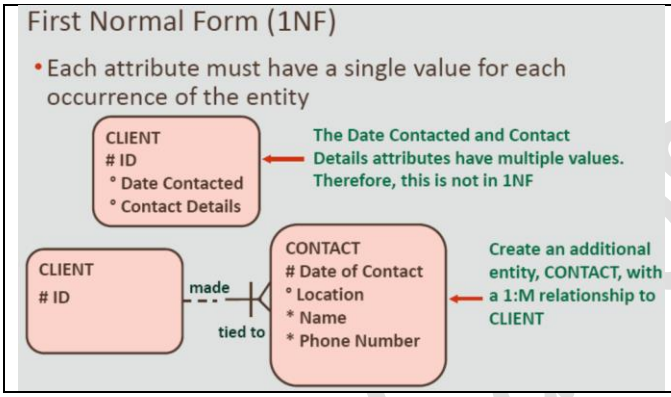
3.3. Normalization and Business Rules

Normalization: Is the process of organizing the attributes and tables of a relational database to minimize redundancy.

No duplicate content, Increase the integrity of data.

Rule	Description
First Normal Form (1NF)	All attributes must be single-valued. (no multi-valued = atomic)
Second Normal Form (2NF)	An attribute must be dependent on its entity's entire UID.
	No existen dependencias parciales
Third Normal Form (3NF)	No non-UID attributes can be dependent on another non-UID attribute.
	Ningún atributo no UID puede depender de otro atributo no UID.
	No existen dependencias transitivas.

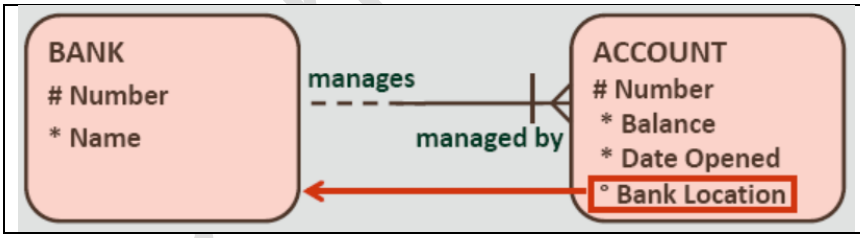
1NF : If an attribute is multi-valued, create an additional entity and relate it to the original entity with a 1:M relationship



Scenario:

Por cada estudiante, se desea registrar los lenguajes de programación que conoce.

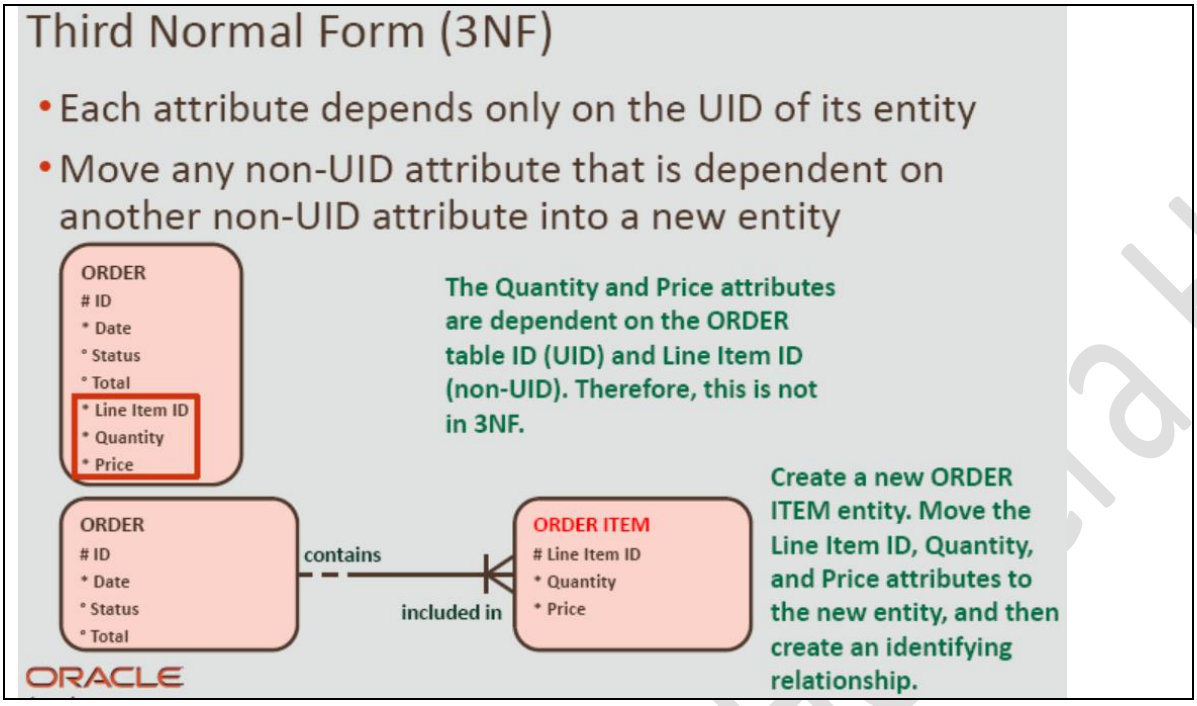
2NF : If an attribute is not dependent on the entire UID, create an additional entity with the partial UID



The Bank Location attribute is dependent on BANK,not on ACCOUNT. Therefore, this is not in 2NF.
Move the attribute to the BANK entity

3NF : You need to move any non-UID attribute that is dependent on another non-UID attribute into a new entity

A transitive dependency exists when any attribute in an entity is dependent on any other non-UID attribute in that entity



Business Rules

A business rule is a statement that defines or constrains some aspect of the business

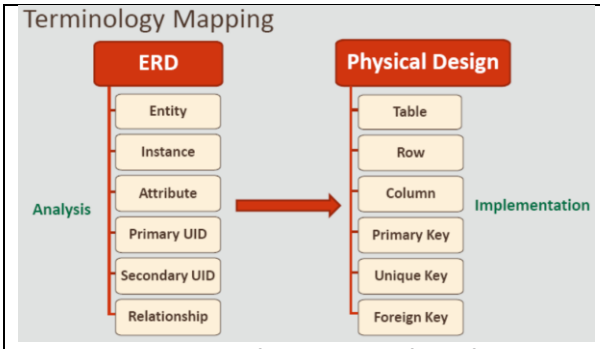
There are two types of business rules:

- Structural: These rules can always be diagrammed in the ERD
- Procedural: Some procedural business rules cannot be diagrammed. But must still be documented.
  Example: Event A must happen before event B
  La aprobación debe estar firmada por el director.
  La tienda no acepta entregar pedido al día siguiente, si se recibe después de las 3pm.
  El libro debe entregarse antes de las 4 dias.
  No se permite prestar otro libro, si debe libros con fecha expirada.
  Las horas extras deberán pagarse al 1.5 veces la tarifa.

Written documentation

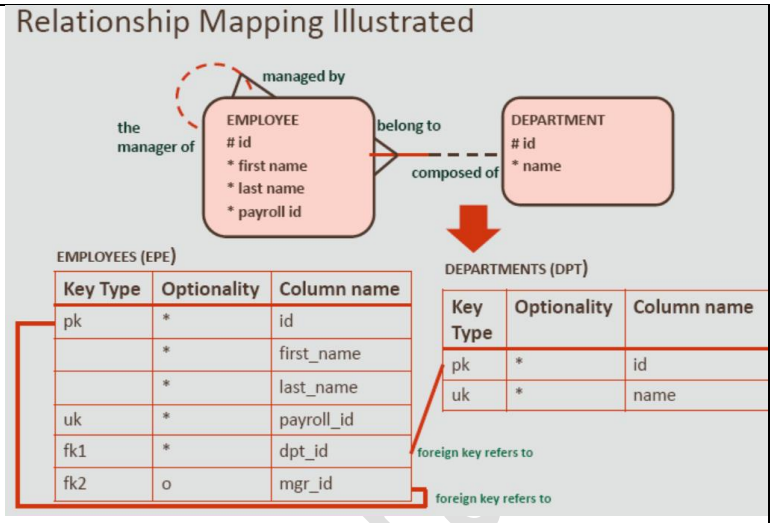
- Procedures
- Standards
- Operations manuals

3.4. Data Modeling Terminology and Mapping

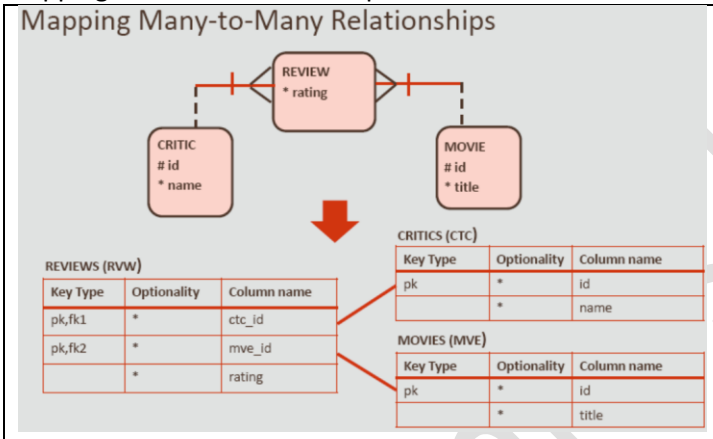


Attributes are classified as one of the following:
- Not null (mandatory): Indicated by the (*)
- Optional (nulls allowed): Indicated by the (o)

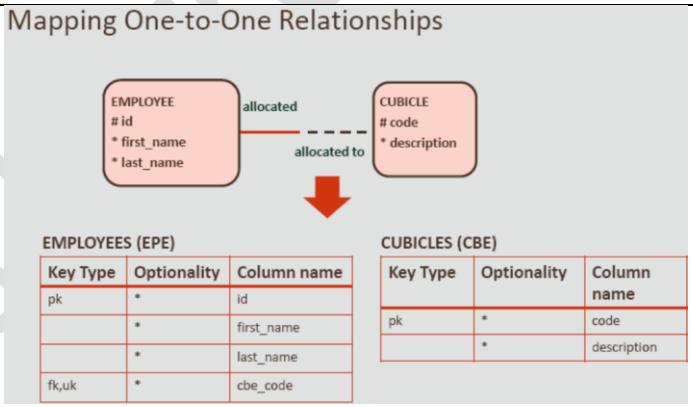
Key Type: pk, uk, fk



Mapping of Barred Relationships

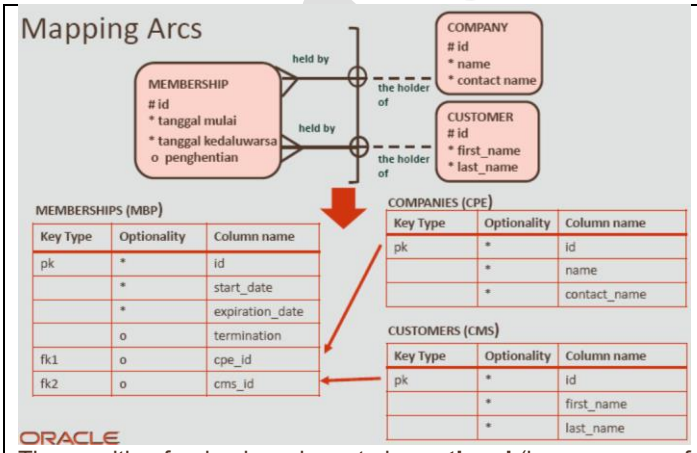


In this case, the foreign-key column plays a double role because it is also part of the primary key



When transforming a 1:1 relationship, you create a foreign key and a unique key

3.4.1. Mapping Arcs



The resulting foreign keys have to be **optional** (because one of them will always be blank).
- CHECK (cpe_id is not null AND cms_id is null)
- OR (cpe_id is null AND cms_id is not null)

For entity names that contain more than one word, take the:
- First character of the first word
- First character of the second word
- Last character of the last word

Example: PRIVATE HOME gets a short name of PHE.

For entity names that contain one word but more than one syllable, take the:
- First character of the first syllable
- First character of the second syllable
- Last character of the last syllable

Example: EMPLOYEE gets a short name of EPE and CLIENT gets a short name of CET.

For entity names that contain one syllable but more than one character, take the:
- First character
- Second character
- Last character

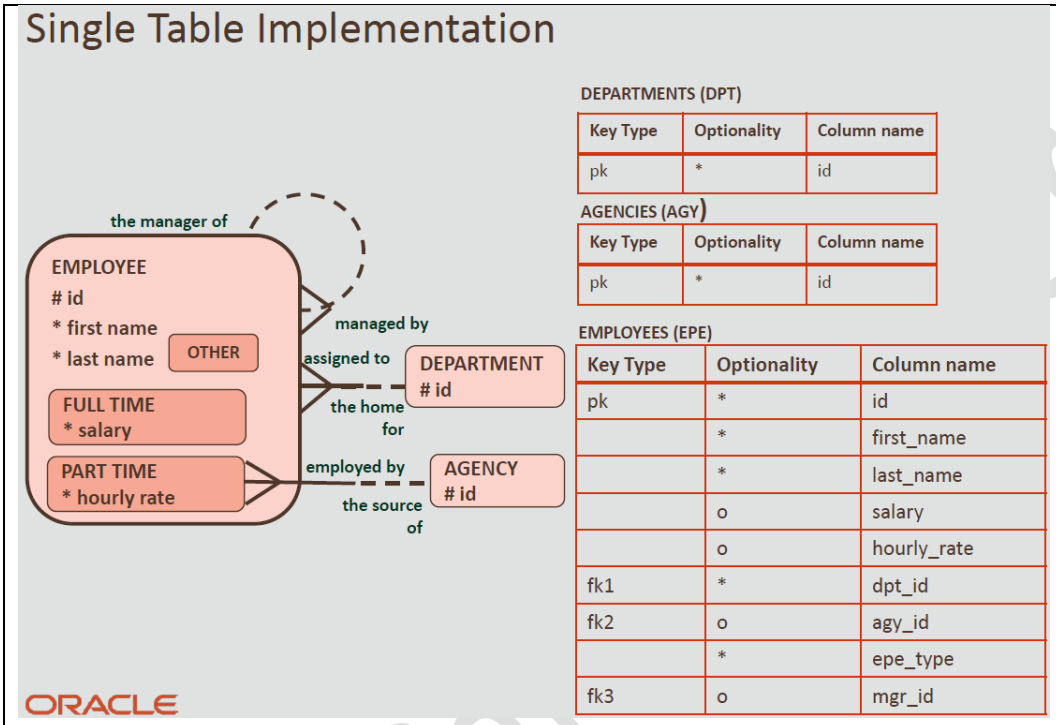
Example: FLIGHT gets a short name of FLT.

3.4.2. Mapping Supertype/Subtypes

Supertype/subtype entities can be mapped in multiple ways:

- Single table implementation : one table is created regardless of number of subtypes, used when most of the attributes and relationships are shared and therefore at the supertype level
- Two table implementation : a table is created for each of the subtypes (so there can be more than two tables), used when subtypes have little in common and few shared attributes and relationships

3.4.2.1. Single Table Implementation



DEPARTMENTS (DPT)

Key Type	Optionality	Column name
pk	*	id

AGENCIES (AGY)

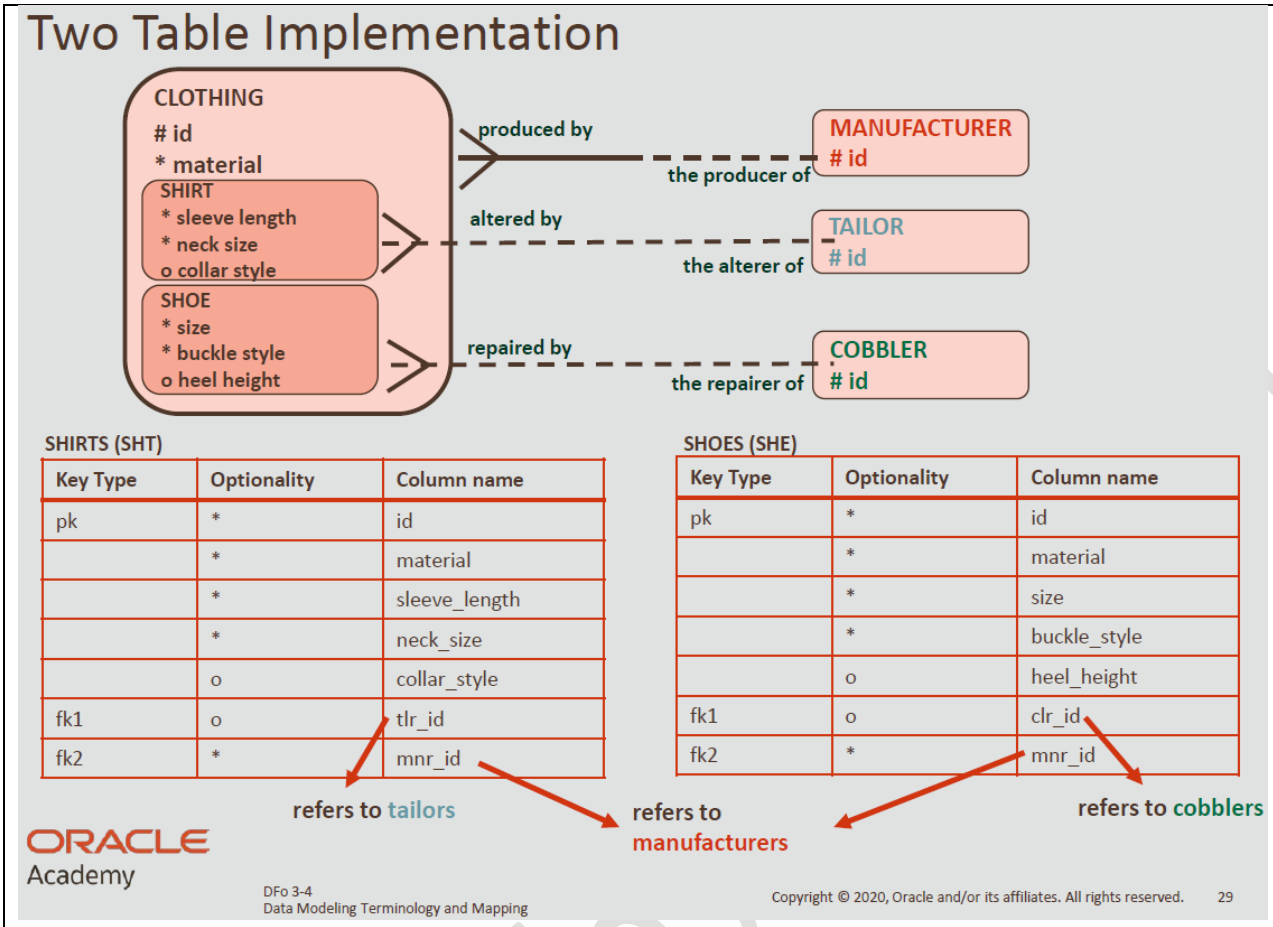
Key Type	Optionality	Column name
pk	*	id

EMPLOYEES (EPE)

Key Type	Optionality	Column name
pk	*	id
	*	first_name
	*	last_name
	o	salary
	o	hourly_rate
fk1	*	dpt_id
fk2	o	agy_id
	*	epe_type
fk3	o	mgr_id

A mandatory column should be created to act as a discriminator column to distinguish between the different subtypes of the entity. For this example: **epe_type** (full time or part time)

3.4.2.2. Two Table Implementation



For relationships at the subtype levels, the foreign key is implemented in the table it is mapped to, original optionality is retained. (tailors, cobblers)

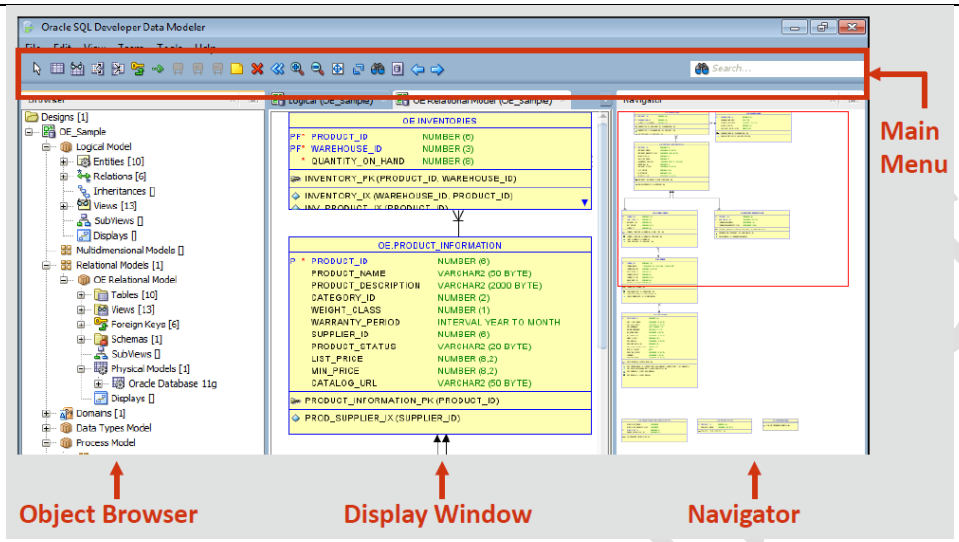
→

4. Oracle SQL Developer Data Modeler

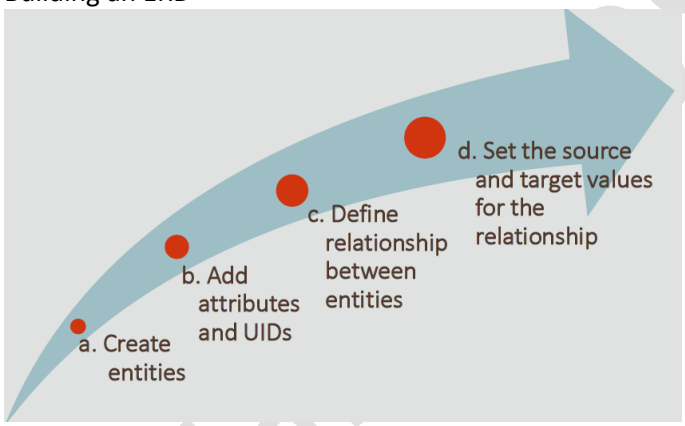
4.1. Oracle SQL Developer Data Modeler

Use Oracle SQL Developer Data Modeler to create:

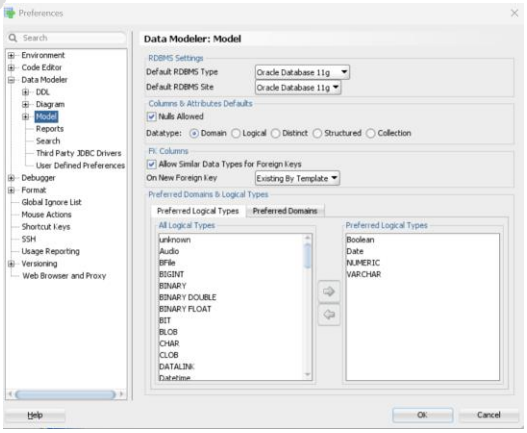
- Entities, attributes, and UIDs with correct optionality and cardinality
- Supertype and subtype entities
- Arc, hierarchical, barred, and recursive relationships



Building an ERD

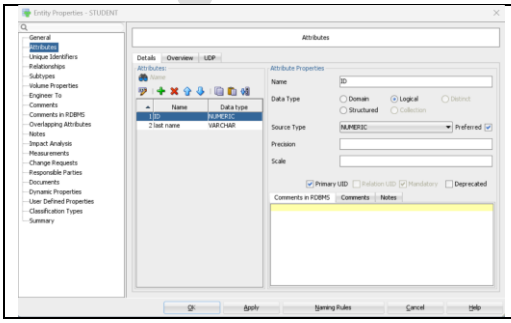


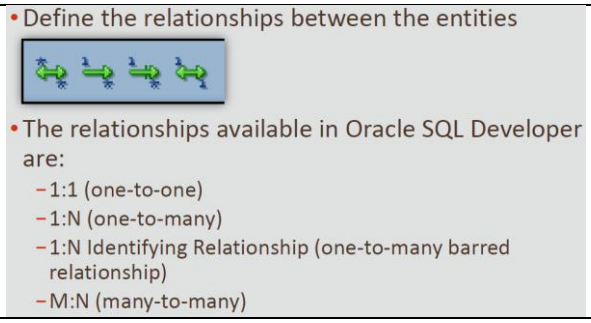
Tools -> Preferences -> Data Modeler -> Model
Boolean, Date, NUMERIC, VARCHAR



Entity -> Attributes -> Attributes Details -> Preferred

Set Primary and Secondary UIDs

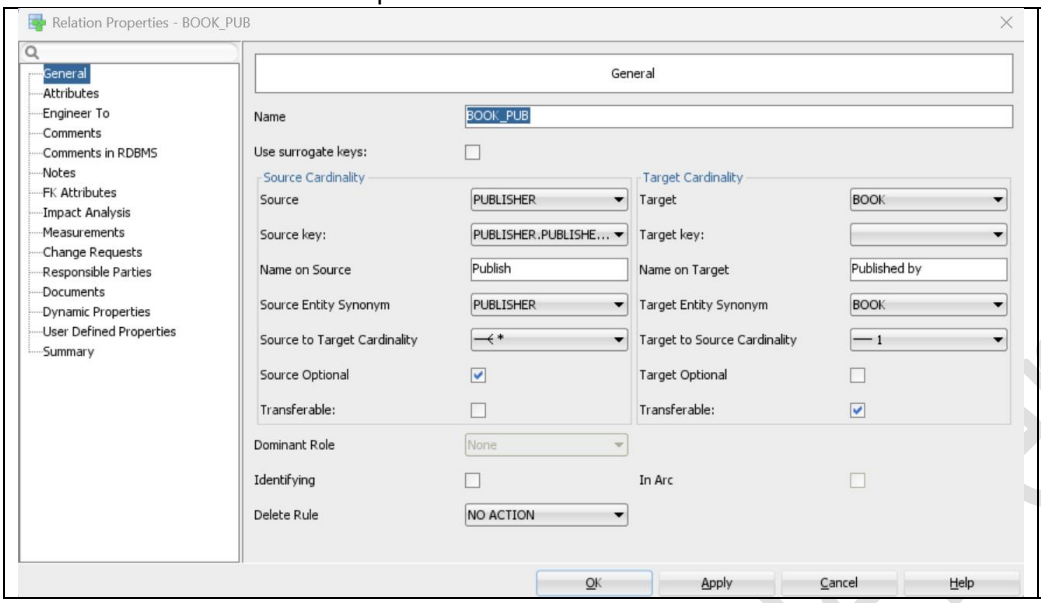




To define the relationships between entities in Oracle SQL Developer, perform the following steps:

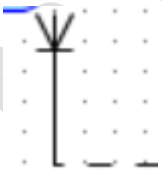
a. Click a relationship type on the toolbar
b. Click the source entity(PK) and then click the target entity(FK) . The relationship is created

Double-Click on the relationship established



Show labels:
Right click on the whitespace
Show -> Labels

Change the path
Right click on the relationship
Select Add elbow



Creating the Supertype Entity

Select the table that is subtype (Full Time) General properties -> Super Type (select Employee)	

Creating the Arc Relationship

Perform the following steps: a. Hold the ctrl key and select the intersecting entity and both relationships on which you want to create the Arc relationship b. Click the New Arc icon in the toolbar. The exclusive relationship is created with the arc	

Creating the Barred Relationship

To add a barred relationship select Identifying Relationship from the toolbar, and click the source and target entities to add the relationship between the entities	

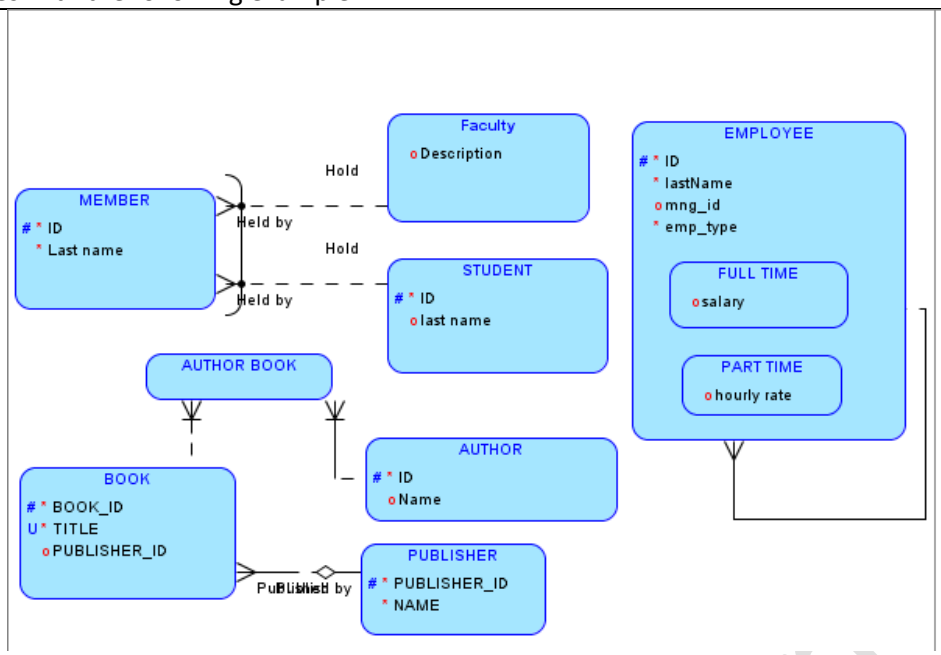
Creating the Hierarchical Relationship

The UIDs for a set of hierarchical entities can be propagated through multiple relationships by making the relationships Identifying

Creating the Recursive Relationship

To add a Recursive Relationship, select the required relationship from the toolbar as normal, then click on the entity to make it the source, and click on the same entity a second time to make it the target.

Practices with the following example:



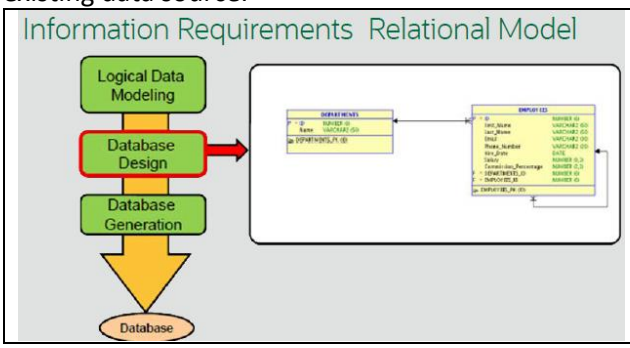
→

4.2. Convert a Logical Model to a Relational Model

Objectives:

- Describe how to convert a logical model to a relational model
- List the steps to convert a logical model to a relational model
- List the steps to convert a relational model to a logical model

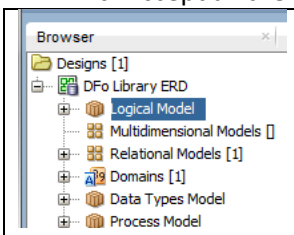
In Oracle SQL Developer Data Modeler a physical model is represented by a Relational Model.
Forward engineering is the process of transforming a logical data model to a relational model.
Reverse engineering is the process of creating a conceptual or logical model by extracting the information from an existing data source.

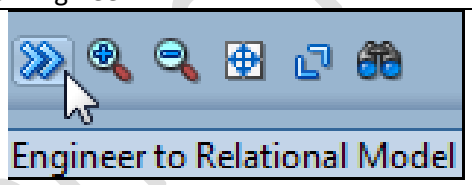


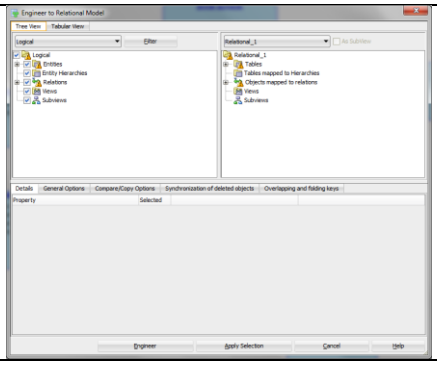
ANALYSIS	DESIGN
Logical Model	Relational Model
Entity	Table
Attribute	Column
Instance	Row
Primary UID	Primary Key
Secondary UID	Unique Constraint
Relationship	Foreign Key
Business Constraints	Check Constraints

Logical Model to a Relational Model:

a. Select the logical model.
b. Click the Engineer to Relational Model icon
c. Accept all the defaults and click Engineer



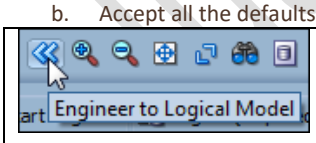


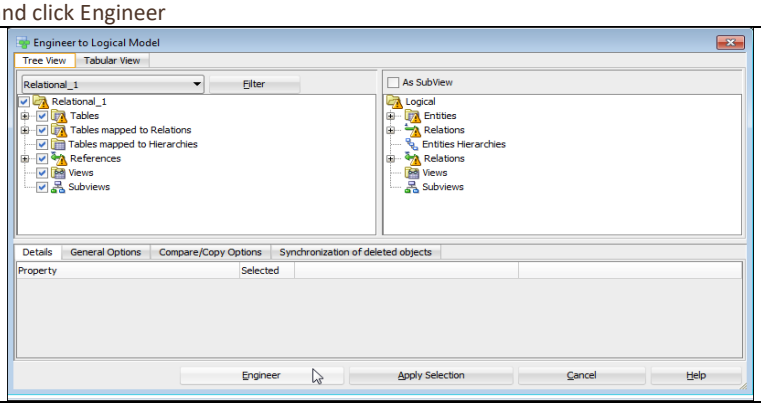


Reverse engineering from a Relational to a Logical Model

Allows an ERD to be created from an existing Physical design

a. Click the Engineer to Logical Model icon
b. Accept all the defaults and click Engineer





→

5. Mapping to the Physical Model

5.1. Mapping Entities and Attributes

In Oracle SQL Developer Data Modeler a Physical model is known as a Relational Model.

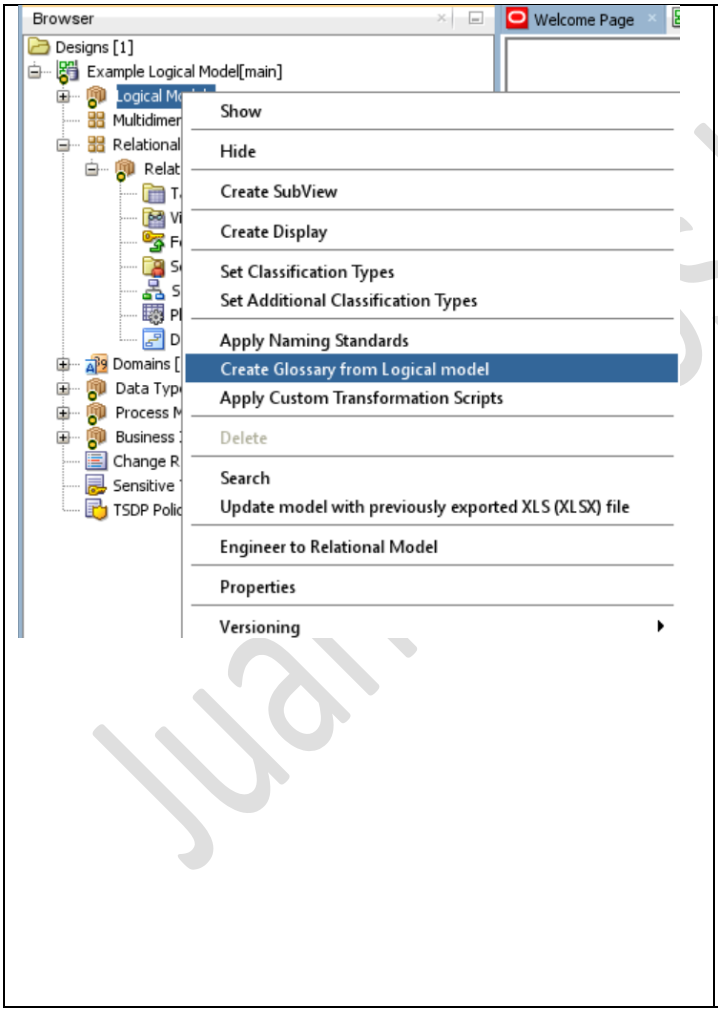
An Entity Relationship Model (ERM) does not highlight the physical and database constraints. It is essential to transform the ERM into a relational model that can serve as the basis for defining the physical implementation of the database.

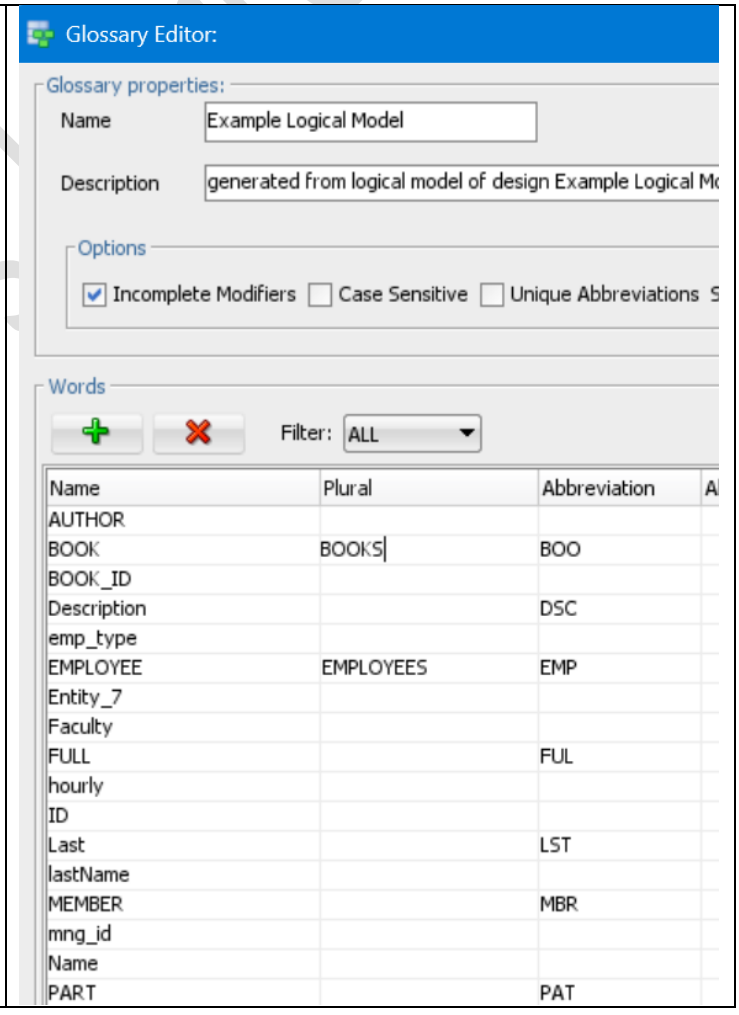
Table short names (abbreviations)
A unique abbreviation for every table is a very useful element for the naming of foreign key columns or foreign key constraints.

Applying Naming Standards from Logical to a Relational Model

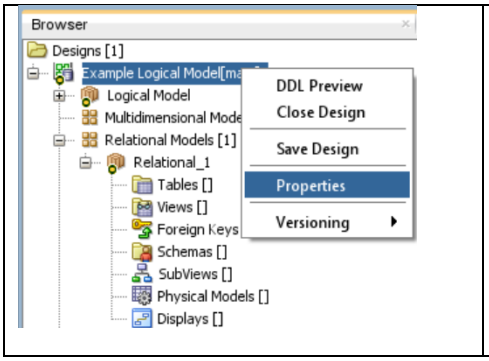
Logical Model (ERD)		Physical model (Relational Model)
Entity		Table
EMPLOYEE	to	EMPLOYEES
Last name	to	Last_name

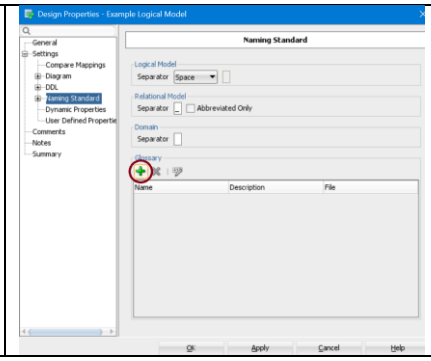
Create a Glossary from a Logical Model

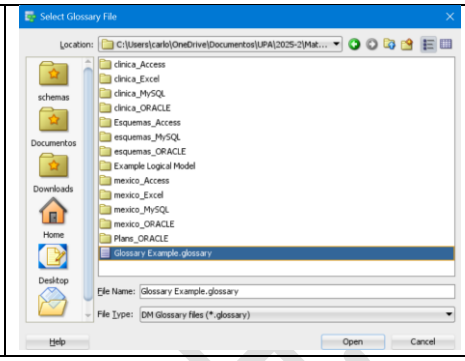




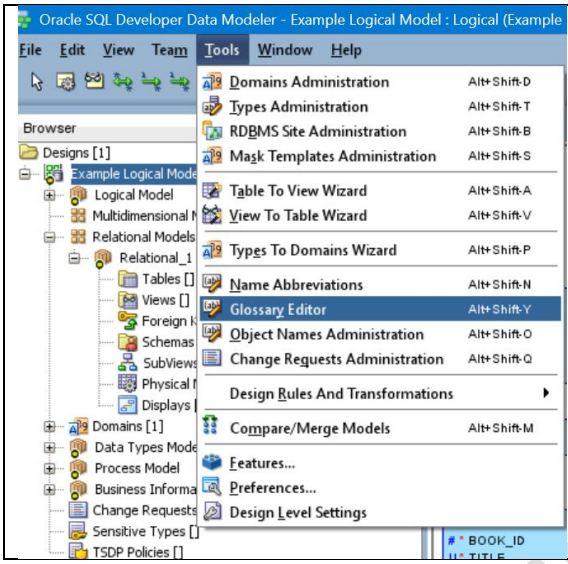
Adding the Glossary as the Naming Standard

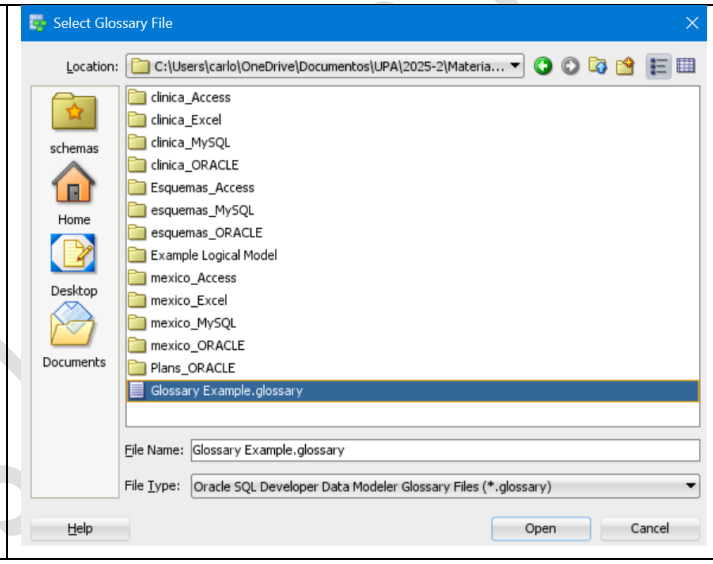






Or

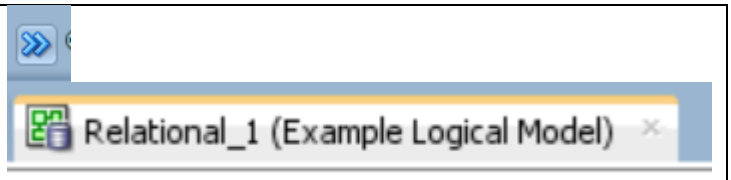


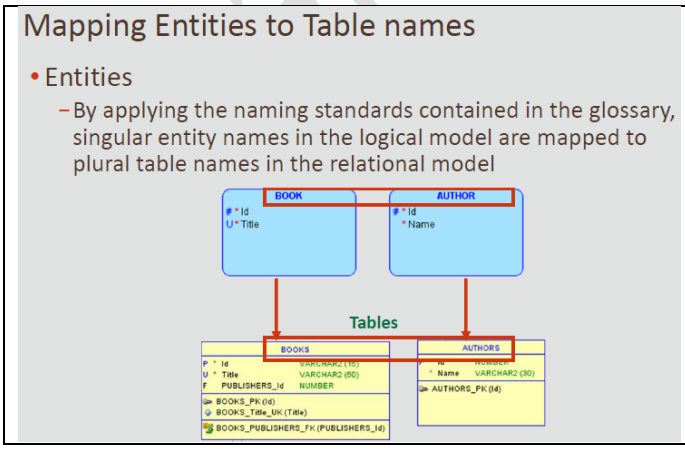


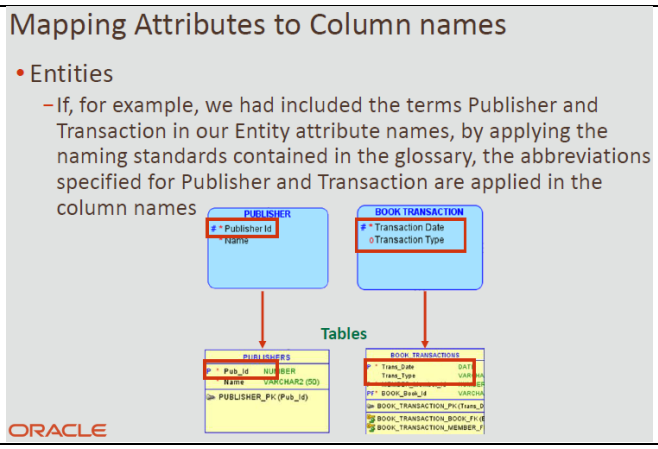
Applying the Naming Standard

To Apply the Glossary as the naming standard,

a) Engineer the Logical model once more
b) lick the relational tab to view the results







5.2. Mapping Primary and Foreign Keys $$

→

6. Introduction to SQL

→