Bases de datos I

Transform An Entity-relationship To Relational Model

Escuela de Ingeniería de sistemas y computación.
Universidad del Valle

GUIA Grupo de Univalle en Inteligencia Artificial

Universidad del Valle

Jefferson A.

1. E-R to REL

2. SQL / NoSQL / No-REL

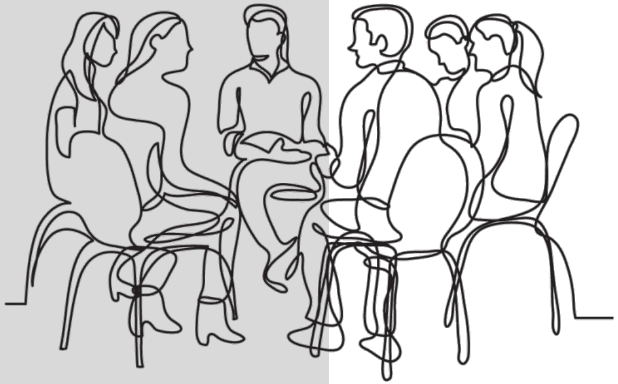
3. PostgreSQL

750030-M-Bases de datos I

Review: Spanish

En el modelo relacional los datos son representados como ~~tablas~~ o relaciones.

Cada relación tiene un nombre y dos dimensiones, una serie de columnas nombradas y un número arbitrario de filas. Las relaciones se caracterizan por que cada intersección fila- columna es simple, cada columna determina un dominio y cada fila es única.



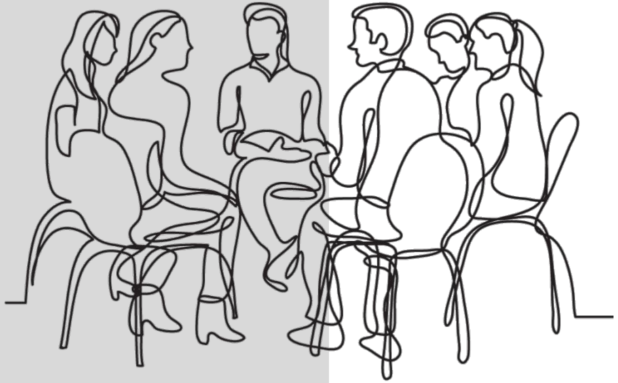
Las relaciones posee una cantidad mínima de redundancia y permite insertar, modificar y eliminar las filas sin errores o inconsistencias.

1. E-R to REL

2. SQL / NoSQL / No-REL

3. PostgreSQL

Outline



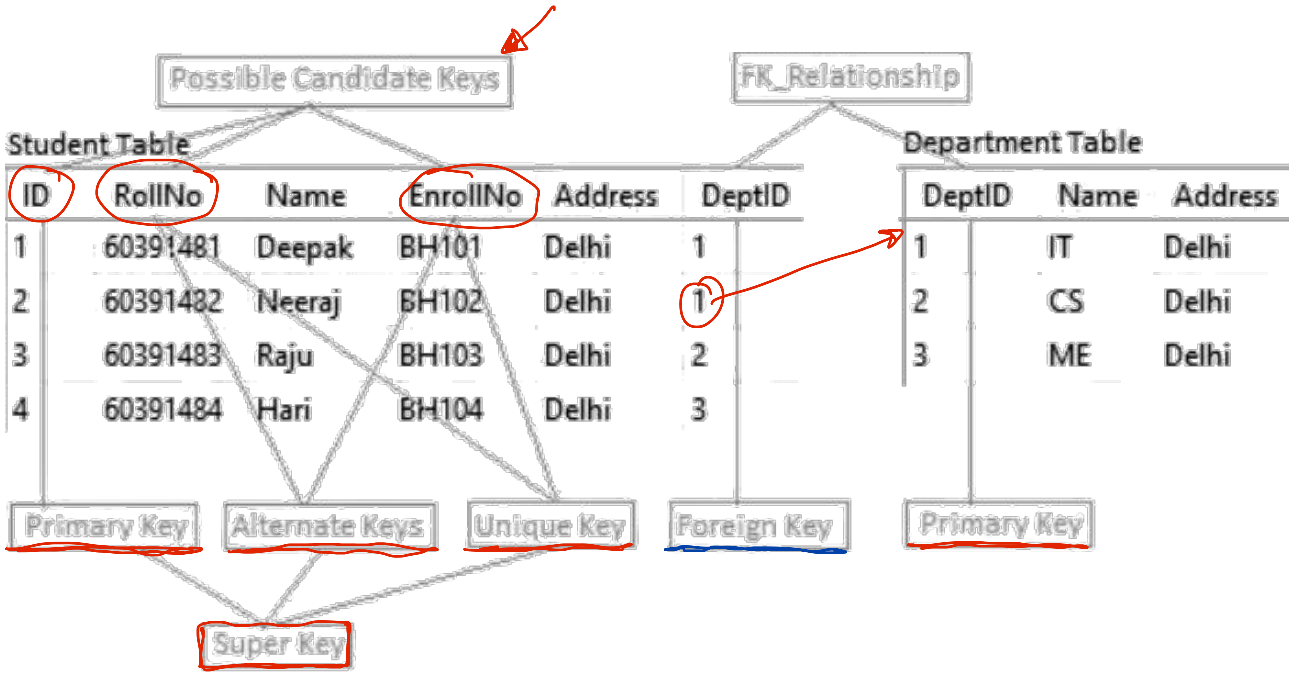
Preliminar definitions

1.
2.
3.

A set of attributes is a **key** for a relation if two distinct tuples can not have same values in all key fields,

If there's more than one keys for a relation, one of the keys is chosen (by DBA) to be the **primary key**. The primary key can not have null value

A **foreign key** is a set of fields in one relation that is used to `refer' to a tuple in another relation



Student Table

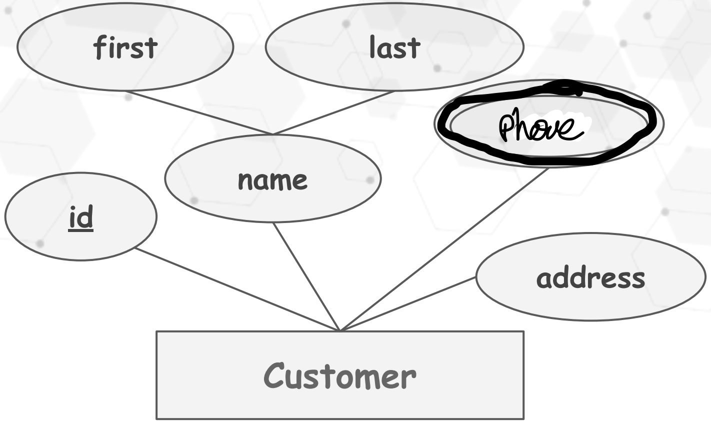
ID	RollNo	Name	EnrollNo	Address	DeptID
1	60391481	Deepak	BH101	Delhi	1
2	60391482	Neeraj	BH102	Delhi	1
3	60391483	Raju	BH103	Delhi	2
4	60391484	Hari	BH104	Delhi	3

Department Table

DeptID	Name	Address
1	IT	Delhi
2	CS	Delhi
3	ME	Delhi

Transforming E-R to REL: Entities

An entity is a real-world object with some attributes.



Transforming E-R to REL: Entities

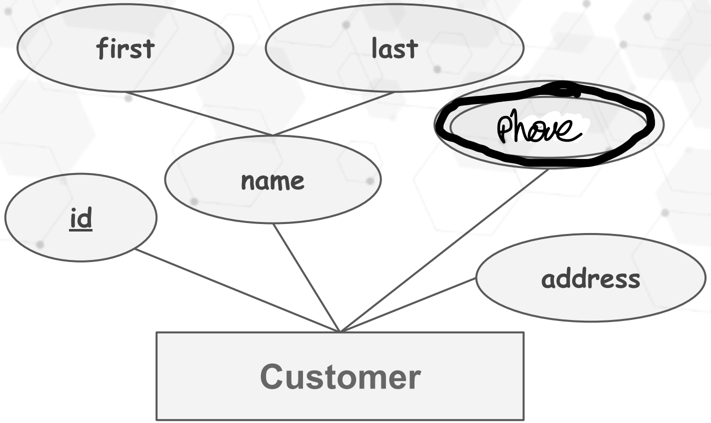
1.

2.

3.

An entity is a real-world object with some attributes.

- Create table for each entity.
- Entity's attributes should become fields of tables with their respective data types.
- Declare primary key.



Customer
id: String PK
first_name: String
last_name: String
Phone: String
address:String

1.

2.

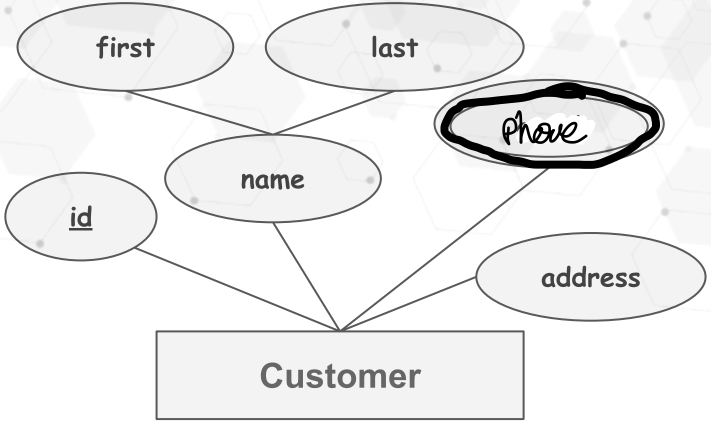
3.

750030-M-Bases de datos I

Transforming E-R to REL: Entities

An entity is a real-world object with some attributes.

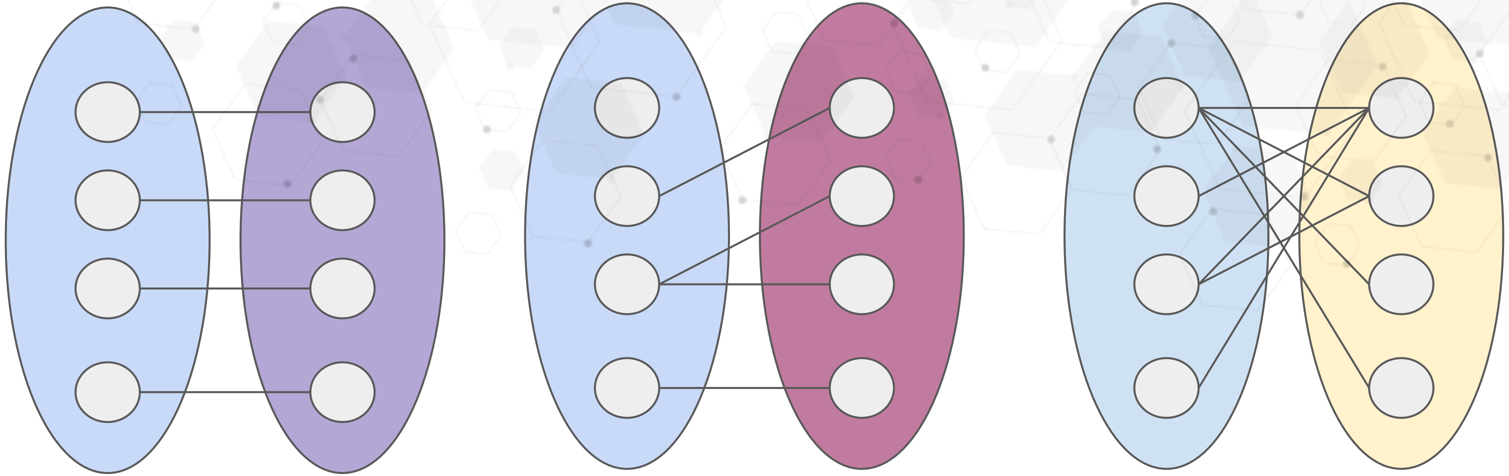
- Create table for each entity.
- Entity's attributes should become fields of tables with their respective data types.
- Declare primary key.



Customer
id: String
first_name: String
last_name: String
~~email~~: String
address:String

id	first_name	last_name	phone	dirección
Ax928	Anakin	Skywalker	038 722772	Calle siempre viva 123

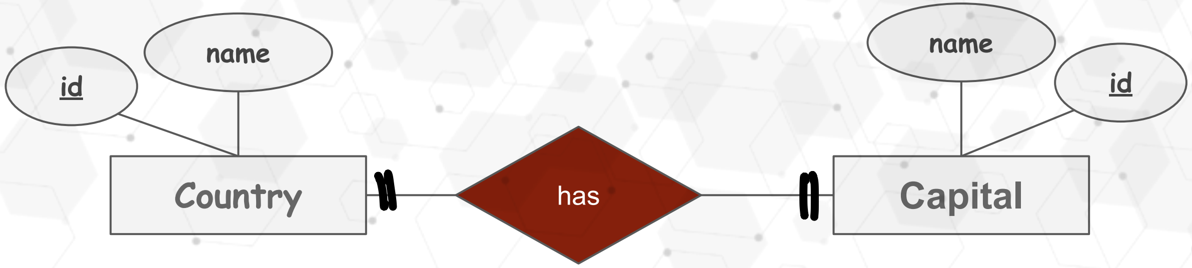
Transforming E-R to REL: Relationships



Transforming E-R to REL: Relationships

A relationship is an association among entities. If cardinality is one-to one among A and B

- Add the primary key of A as a foreign key of B
- Add the primary key of B as a foreign key of A
- Both



1.

2.

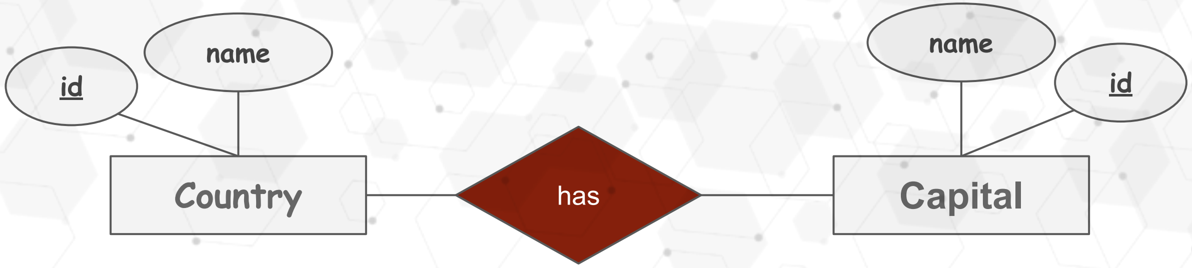
3.

750030-M-Bases de datos I

Transforming E-R to REL: Relationships

A relationship is an association among entities. If cardinality is one-to one among A and B

- Add the primary key of A as a foreign key of B
- Add the primary key of B as a foreign key of A
- Both



Country	
id: String	PK
name:String	

[1, 'Colombia']
[2, 'Brasil']

Capital	
id: String	PK
name:String	
countryId:String	FK

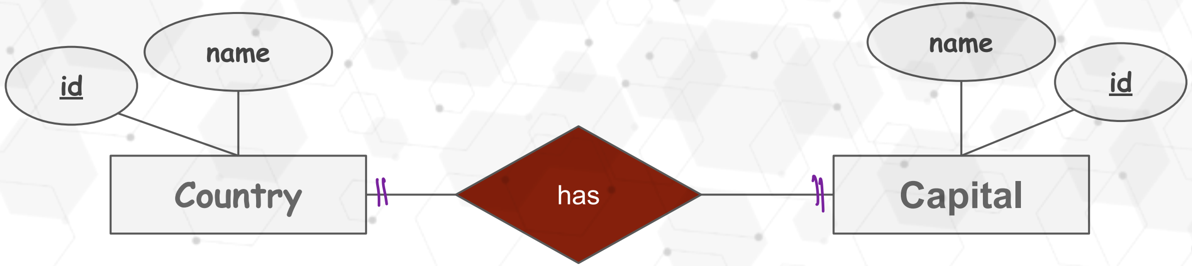
[95, 'Bogota', 1]
[93, 'Brasilia', 2]

Transforming E-R to REL: Relationships

1.
2.
3.

A relationship is an association among entities. If cardinality is one-to one among A and B

- Add the primary key of A as a foreign key of B
- Add the primary key of B as a foreign key of A
- [✓ Both]



Country
id: String
name:String
capitalID:String

[203, 'Colombia', 105]

Capital
id: String
name:String

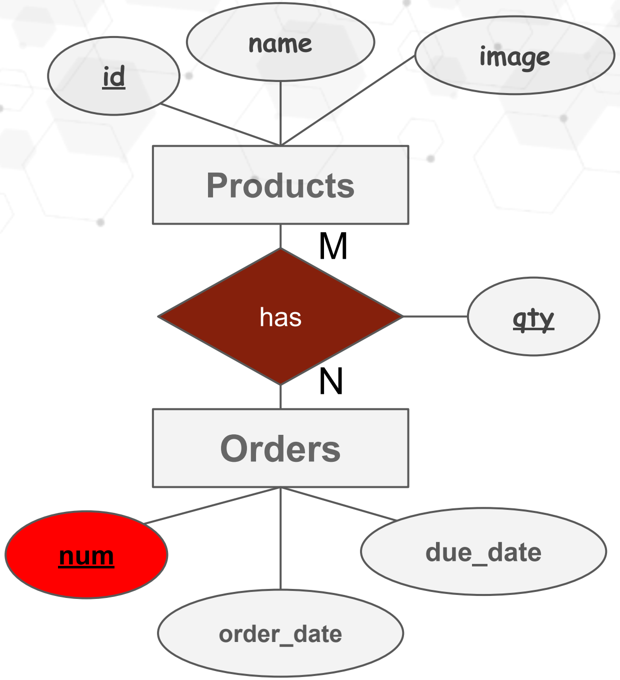
[105, Bogota]

Transforming E-R to REL: Relationships

1.
2.
3.

A relationship is an association among entities. If cardinality is many-to-many among A and B

- Create table for a relationship.
- Add the primary keys of all participating Entities as fields of table with their respective data types.
- If relationship has any attribute, add each attribute as field of table.
- Declare a primary key composing all the primary keys of participating entities.
- Declare all foreign key constraints.

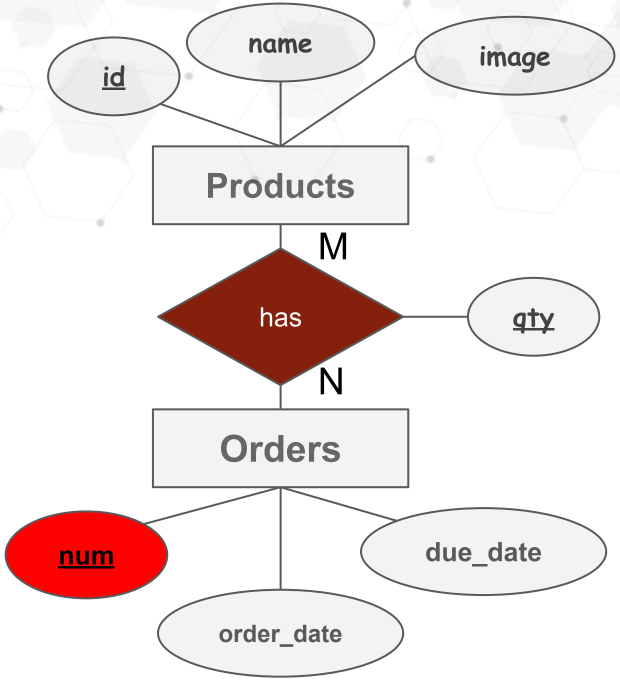


Transforming E-R to REL: Relationships

1.
2.
3.

A relationship is an association among entities. If cardinality is many-to-many among A and B

- Create table for a relationship.
- Add the primary keys of all participating Entities as fields of table with their respective data types.
- If relationship has any attribute, add each attribute as field of table.
- Declare a primary key composing all the primary keys of participating entities.
- Declare all foreign key constraints.



Products
id: String PK
name:String
image:String

Orders
num: String PK
due_date: Timestamp
order_date: Timestamp

List
num: String PK FK
id: String PK FK
qty: int

750030-M-Bases de datos I

Transforming E-R to REL: Relationships

1.

2.

3.

A relationship is an association among entities. If cardinality is many-to-many among A and B

- Create table for a relationship.
- Add the primary keys of all participating Entities as fields of table with their respective data types.
- If relationship has any attribute, add each attribute as field of table.
- Declare a primary key composing all the primary keys of participating entities.
- Declare all foreign key constraints.

Order

num	due_date	order_date
31	4/12/2021	8/12/2021
28	4/12/2021	6/12/2021

Productos

id	desc	image
prd1	Uncharted: Drake's Fortune	UCDF.jpg
prd2	Uncharted: Fortune Hunter	UCFH.png

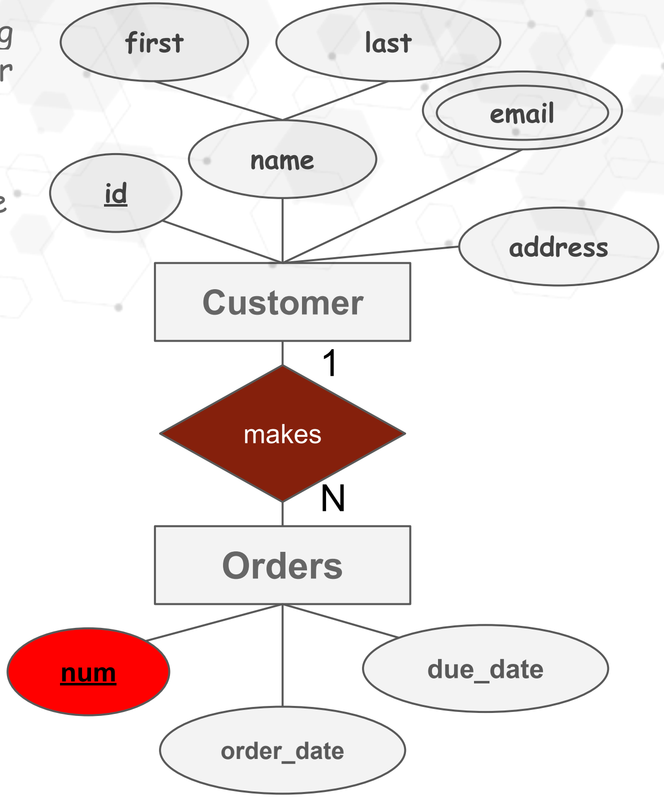
Product Order

num	id	qty
~~Ped123~~ 31	prd1	5
~~Ped456~~ 28	prd1	8
~~Ped123~~ 31	prd2	23

Transforming E-R to REL: Relationships

A relationship is an association among entities. If cardinality is one-to-many or many-to-one among A and B

- Declare a primary key composing all the primary keys of participating entities.
- Declare all foreign key constraints.

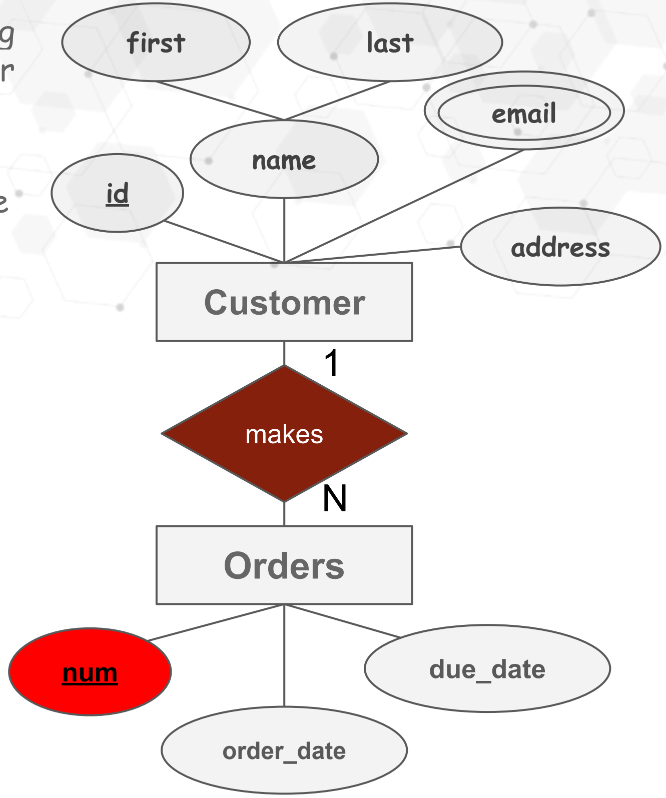


Transforming E-R to REL: Relationships

1.
2.
3.

A relationship is an association among entities. If cardinality is one-to-many or many-to-one among A and B

- Declare a primary key composing all the primary keys of participating entities.
- Declare all foreign key constraints.



Customer	
id: String	PK
first_name: String	
last_name: String	
email: String	
address:String	

Orders	
num: Integer	PK
order_date: String	
due_Date: String	
customer_id: String	FK

Transforming E-R to REL: Relationships

A relationship is an association among entities. If cardinality is one-to-many or many-to-one among A and B

- Declare a primary key composing all the primary keys of participating entities.
- Declare all foreign key constraints.

id	fname	...	address
Ac827	Homero	3121	Cll . . .
Ac456	Tom	23212	Carrera

Customer
id: String
first_name: String
last_name: String
email: String
address:String

num	fpedido	...	c_id
34	4/12/2021	4/12/2021	**Ac827**
89	4/12/2021	4/12/2021	Ac456
97	4/12/2021	4/12/2021	**Ac827**
98	4/12/2021	4/12/2021	**Ac827**

Orders
num: Integer
order_date: String
due_Date: String
customer_id: String

1.

2.

3.

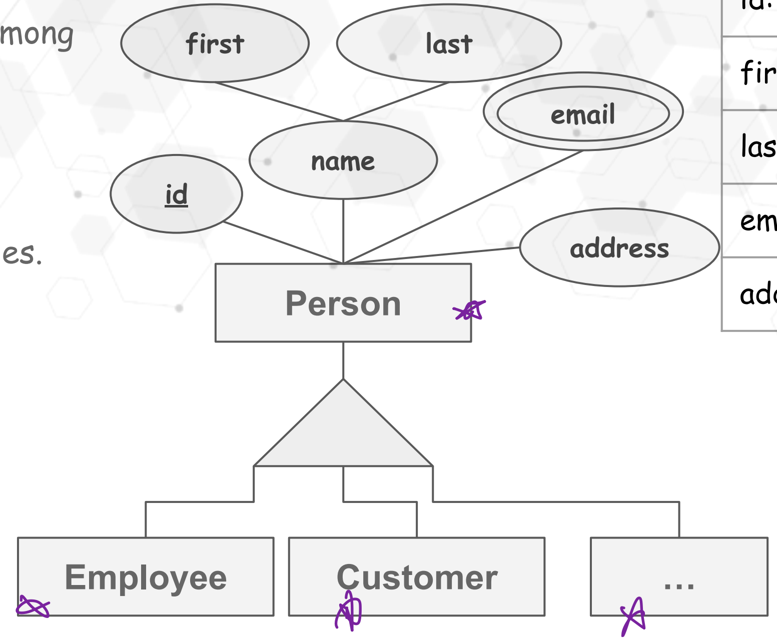
750030-M-Bases de datos I

Transforming E-R to REL: Generalization

A relationship is an association among entities.

.

- Create tables for all higher-level entities.
- Create tables for lower-level entities.
- Add primary keys of higher-level entities in the table of lower-level entities.
- In lower-level tables, add all other attributes of lower-level entities.



Person
id: String
first_name: String
last_name: String
email: String
address:String

. . .
id: String
first_name: String
last_name: String
email: String
address:String

Customer
id: String
first_name: String
last_name: String
email: String
address:String

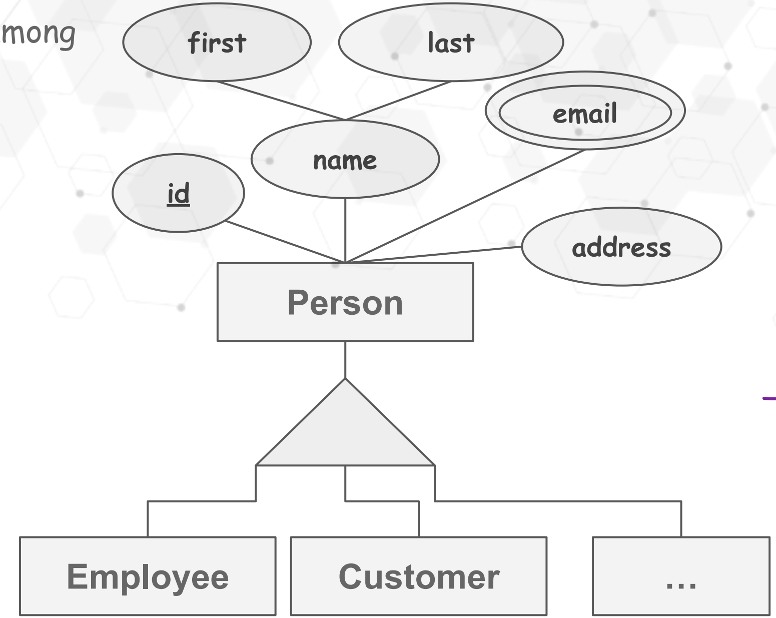
Transforming E-R to REL: Generalization

1.
2.
3.

A relationship is an association among entities.

.

- Create tables for all higher-level entities.
- Add a attribute to know subclass



Person
id: String
first_name: String
last_name: String
email: String
address:String
→ person_type: String

1.

2.

3.

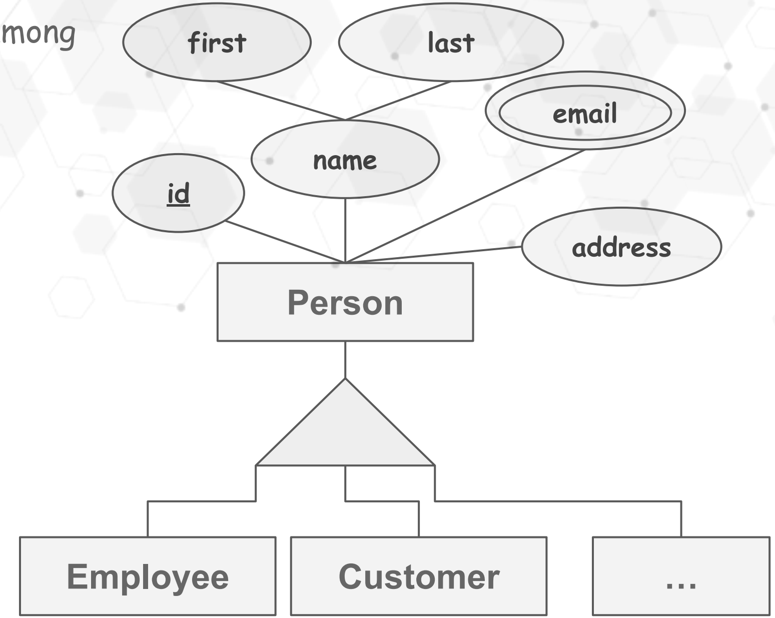
750030-M-Bases de datos I

Transforming E-R to REL: Generalization

A relationship is an association among entities.

.

- Create tables for all higher-level entities.
- Create tables for know subclass
- Add primary keys of higher-level entities in the table of lower-level entities.
- In lower-level tables, add all other attributes of lower-level entities.



PersonType
id: String PK
name: String

Person
id: String PK
first_name: String
last_name: String
email: String
address:String
persontype:Integer FK

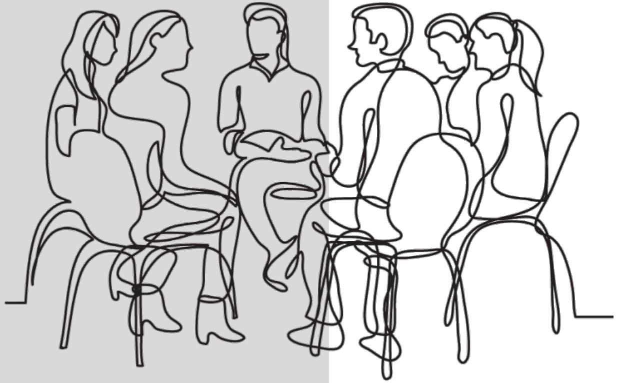
1. E-R to REL

2. SQL / NoSQL / No-REL

3. PostgreSQL

750030-M-Bases de datos I

Outline



750030-M-Bases de datos I

SGBD

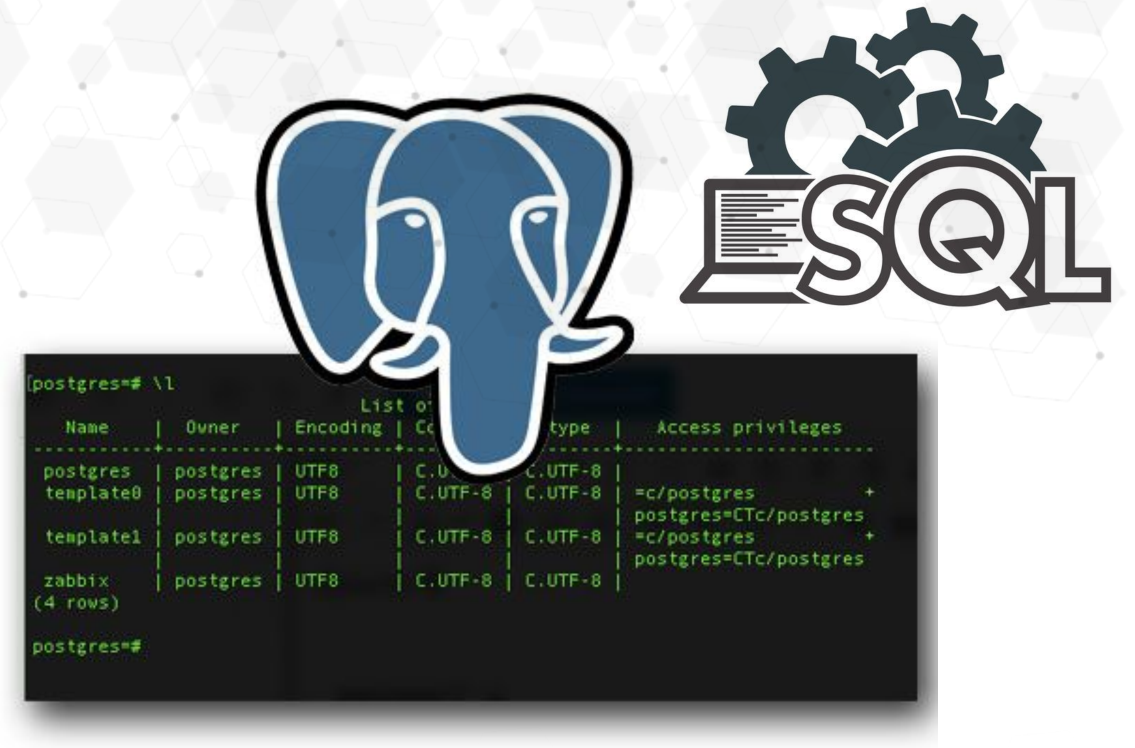
1.
2.
3.
4.

SQL

SQL is a structured query language. specifically designed to manage information in relational-type database management systems.

SQL can be considered a programming language. SQL has variables, data types, conditional and logical structures. SQL is the de facto standard for data management and allows:

- Query, update and reorganize data
- Create and modify the data structure
- Control access to data



```
[postgres=# \l
                                List o
   Name    |  Owner   | Encoding | Cc              type   |   Access privileges
-----------+----------+----------+---              -------+-----------------------
 postgres  | postgres | UTF8     | C.U            C.UTF-8 |
 template0 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres          +
           |          |          |         |         | postgres=CTc/postgres
 template1 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres          +
           |          |          |         |         | postgres=CTc/postgres
 zabbix    | postgres | UTF8     | C.UTF-8 | C.UTF-8 |
(4 rows)

postgres=#
```

SQL

Supported data types are supported by most database system that support SQL, with a few exceptions that are particular to specific system *
In general the data types include:

- Numeric
- Character
- Binary
- Date/Hour
- Logical (boolean)
- List
- Geometric
- Networks

- Bit string
- Text
- UUID
- XML
- JSON
- Arrangements
- Compounds (Compound)
- object identifiers

◦

* PostgreSQL also uses geometric types and network addresses

```
[postgres=# \l
                               List o
   Name    |  Owner   | Encoding | Co              type   |   Access privileges
-----------+----------+----------+-----------------------+-----------------------
 postgres  | postgres | UTF8     | C.U         C.UTF-8 |
 template0 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres          +
           |          |          |         |         | postgres=CTc/postgres
 template1 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres          +
           |          |          |         |         | postgres=CTc/postgres
 zabbix    | postgres | UTF8     | C.UTF-8 | C.UTF-8 |
(4 rows)

postgres=#
```

SQL es un DDL

1.
2.
3.
4.

It is used to define the database structure or schema. SQL is a language provided by the database management system that allows define tables, procedures and functions.

A Data Definition Language or Data Description Language (DDL) is a programming language for defining data structures.

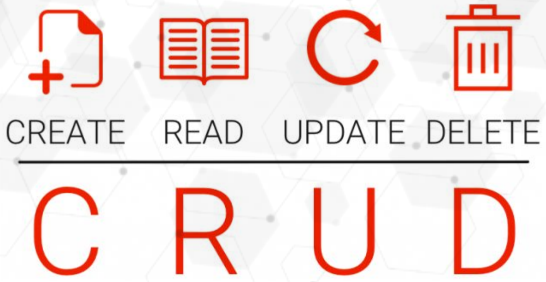
- ***CREATE*** *- PARA CREAR OBJETOS EN LA BASE DE DATOS*
- ***ALTER*** *- ALTERA LA ESTRUCTURA DE LA BASE DE DATOS*
- ***DROP*** *- ELIMINA LOS OBJETOS DE LA BASE DE DATOS*
- ***TRUNCATE*** *- ELIMINAR TODOS LOS REGISTROS DE UNA TABLA , INCLUYENDO TODOS LOS ESPACIOS ASIGNADOS A LOS REGISTROS SE ELIMINAN*
- ***COMMENT*** *- AGREGAR COMENTARIOS AL DICCIONARIO DE DATOS*
- ***RENAME*** *- CAMBIAR EL NOMBRE DE UN OBJETO*

1.
2.
3.
4.

SQL es un DML

It is used for data management within schema objects

SQL is a language provided by the database management systems that allows retrieve and manipulate data in a relational database.



- ***SELECT*** *- RECUPERAR DATOS DE LA BASE DE DATOS.*
- ***INSERT*** *- INSERTAR DATOS EN UNA TABLA.*
- ***UPDATE*** *- ACTUALIZACIONES DE DATOS EXISTENTES EN UNA TABLA.*
- ***DELETE*** *- ELIMINA TODOS LOS REGISTROS DE UNA TABLA.*
- ***MERGE*** *- OPERACIÓN UPSERT (INSERCIÓN O ACTUALIZACIÓN).*
- ***CALL*** *- LLAMA A UN PL / SQL O SUBPROGRAMA JAVA.*
- ***EXPLAIN PLAN*** *- EXPLICAR LA RUTA DE ACCESO A LOS DATOS.*
- ***LOCK TABLE*** *- CONCURRENCIAS DE CONTROL.*

1.
2.
3.
4.

SQL es un DCL

A Data Control Language is a language adapted by the Database Management System that includes a series of SQL commands that allow the administrator to control access to the data contained in the database.

- **GRANT**: Permite dar permisos a uno o varios usuarios o roles para realizar tareas determinadas.
- **REVOKE**: Permite eliminar permisos que previamente se han concedido con GRANT.

SQL es un TCL

SQL is a programming language used to control transaction processing in a database.

A transaction is a logical unit of work that comprises one or more SQL statements, typically a Data Manipulation Language (DML) group.

- **COMMIT** - GUARDA EL TRABAJO REALIZADO
- **SAVEPOINT** - IDENTIFICA UN PUNTO EN UNA TRANSACCIÓN A LA QUE MÁS TARDE SE PUEDE VOLVER.
- **ROLLBACK** - RESTAURAR LA BASE DE DATOS A LA ORIGINAL, HASTA EL ÚLTIMO COMMIT.
- **SET TRANSACTION** - CAMBIA LAS OPCIONES DE TRANSACCIÓN COMO NIVEL DE AISLAMIENTO Y QUÉ SEGMENTO DE CANCELACIÓN UTILIZA.

Other models

1.
2.
3.
4.

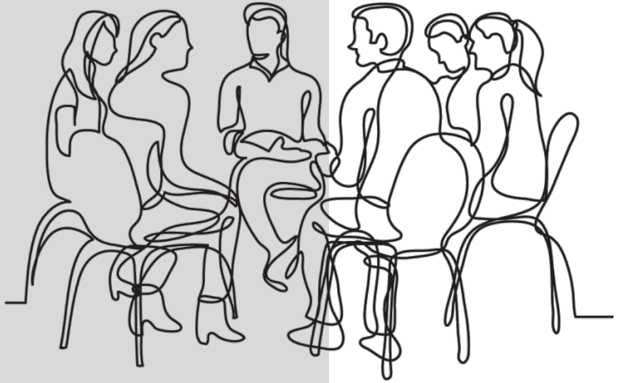
Apache **Cassandra** is an open source NoSQL distributed **database** trusted by thousands of companies for scalability and high availability

Other models

MongoDB is a cross-platform document-oriented **database** program. Although classified as a **schema-less database** program

Outline

1. E-R to REL
2. SQL / NoSQL / No-REL
3. PostgreSQL



Bases de datos I

Transform An Entity-relationship To Relational Model

Escuela de Ingeniería de sistemas y computación.
Universidad del Valle
2022