



El futuro digital
es de todos

MinTIC



UNIVERSIDAD
DE ANTIOQUIA

Facultad de Ingeniería

«Misión
TIC2022»

«Misión
TIC2022»

SEMANA 6

INICIAMOS 8:05PM



UNIVERSIDAD
DE ANTIOQUIA

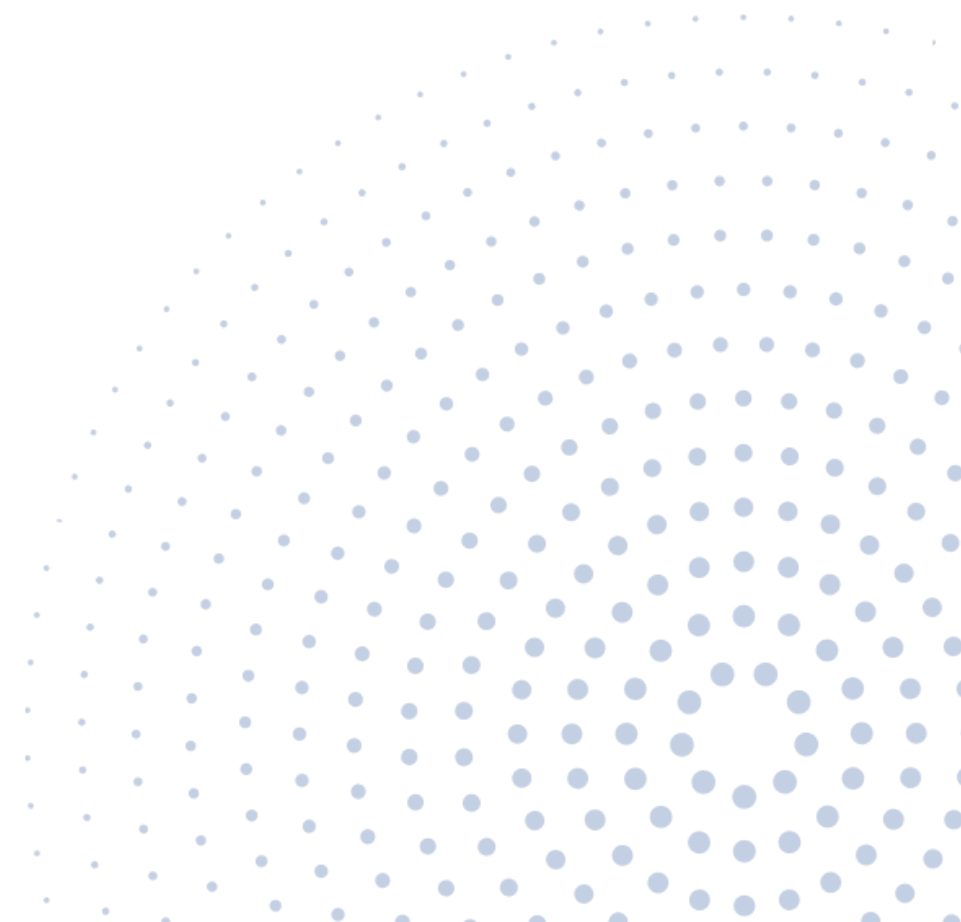
Facultad de Ingeniería

Luisa Fernanda Restrepo.



Agenda

- Diagramas entidad-relación
- Introducción SQL
- Sentencias SQL





El futuro digital
es de todos

MinTIC

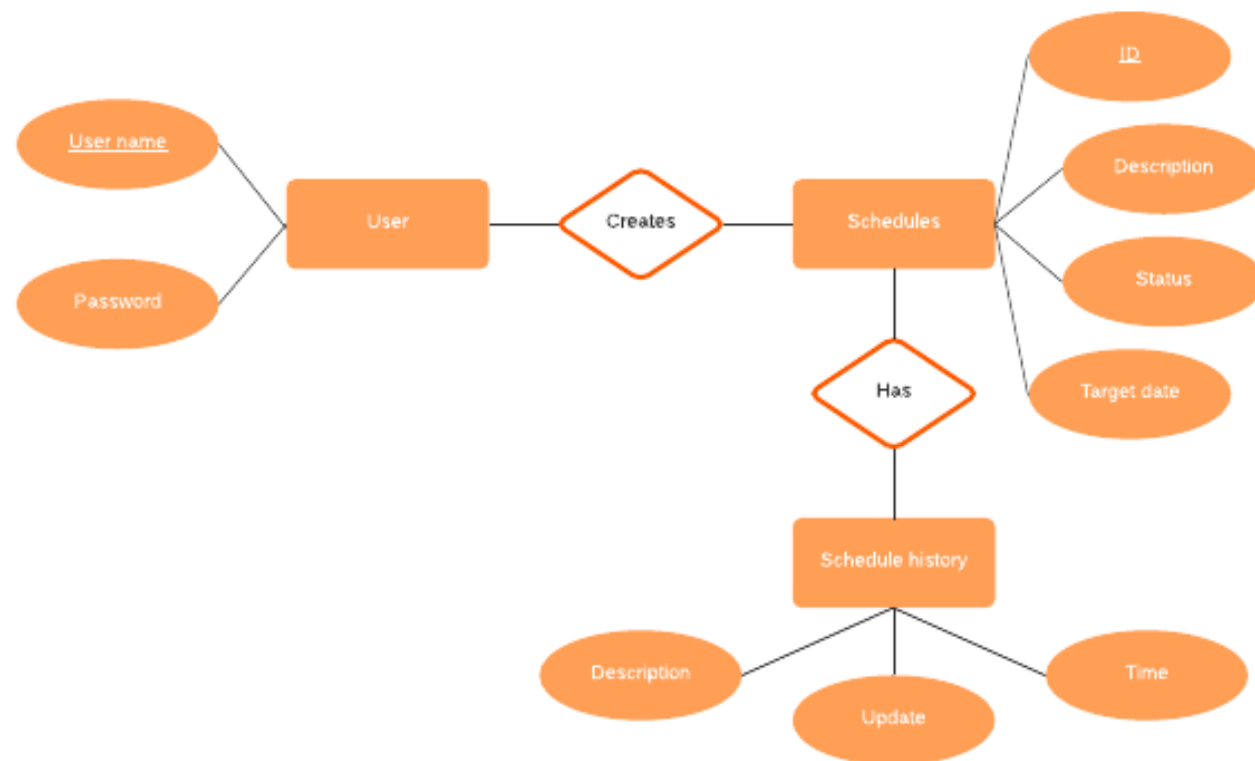


Diagrama entidad-relación



Diagramas entidad-relación

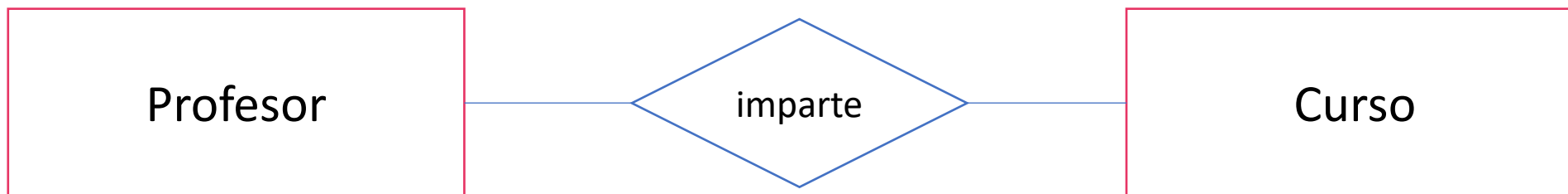
- Se utilizan para modelar la perspectiva de los datos.
- Se basa en una percepción de un mundo real que consiste:
 - Objetos básicos (entidades)
 - Relaciones entre estos objetos
- Muestran las estructuras de un objeto de un dominio.





Elementos: Tipo relación

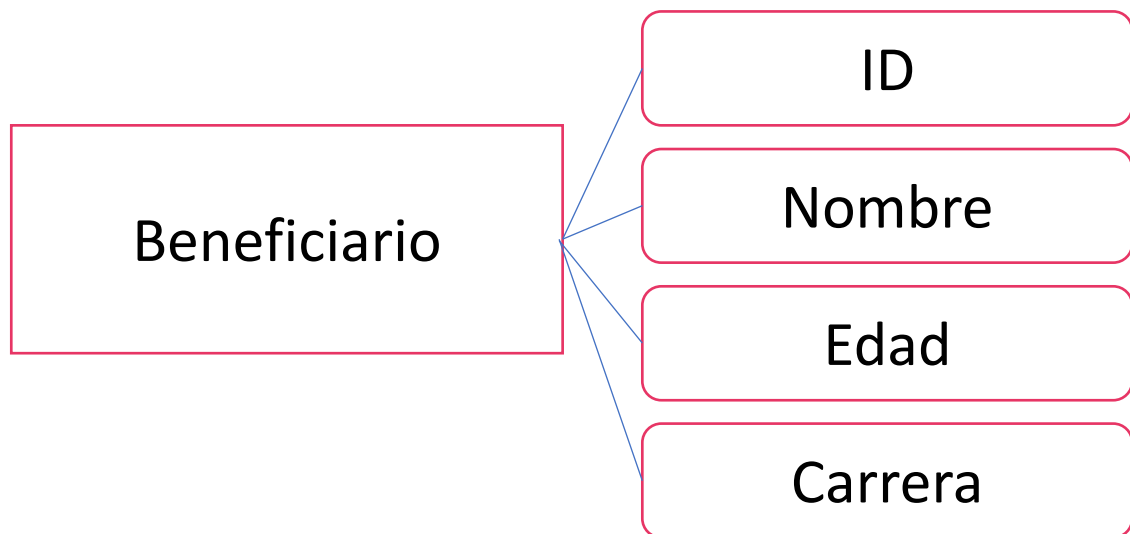
- Es una asociación, vinculación o correspondencia significativa entre dos tipos de entidades.
- La relación debe tener un nombre que sea capaz de identificar el tipo de correspondencia entre ambas entidades.
 - Generalmente estos nombres son verbos.





Elementos: Atributos

- Representa cualquier propiedad o característica significativa de un tipo entidad o tipo relación.
- Cada tipo entidad posee los mismos atributos, diferenciándose cada una por los valores que toman dichos atributos.

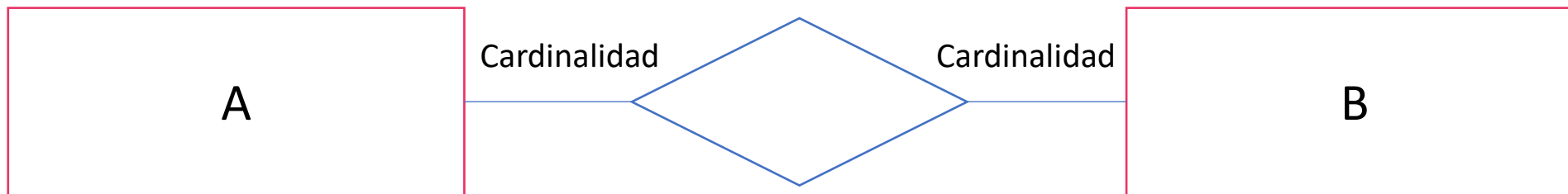


ID	Nombre	Edad	Carrera
1	Sofía	18	Derecho
2	Jorge	19	Contaduría
3	Wilbert	20	Ingeniería
4	Juan	19	Veterinaria



Cardinalidad

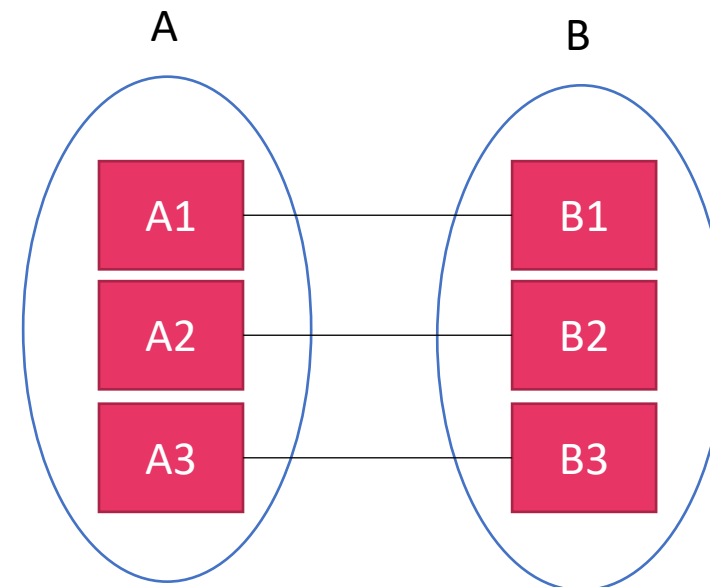
- Es el número de instancias con las cuales está relacionado un tipo entidad con otra.
- Con cuantas instancias de B se puede relacionar A y con cuantas instancias de A se puede relacionar B.





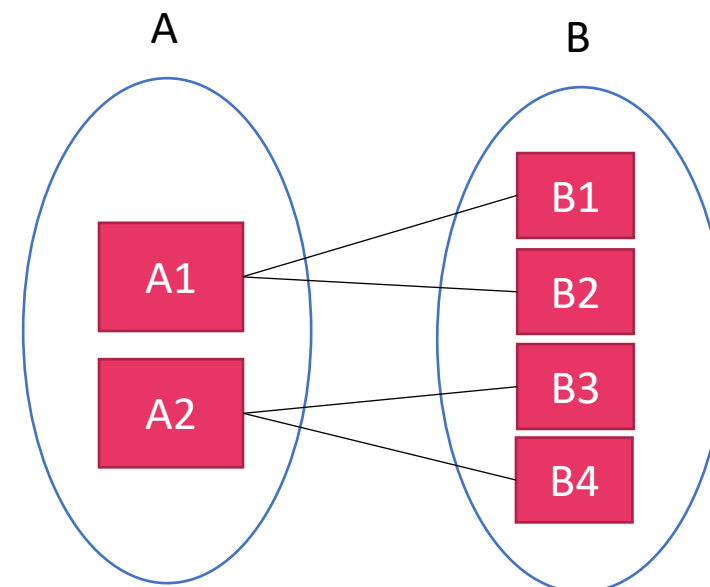
- # Uno a Uno (1 - 1)

- 1 entidad en A está asociado a 1 entidad de B.
- 1 entidad en B está asociada a 1 entidad de A.



- # Uno a Varios (1 - *)

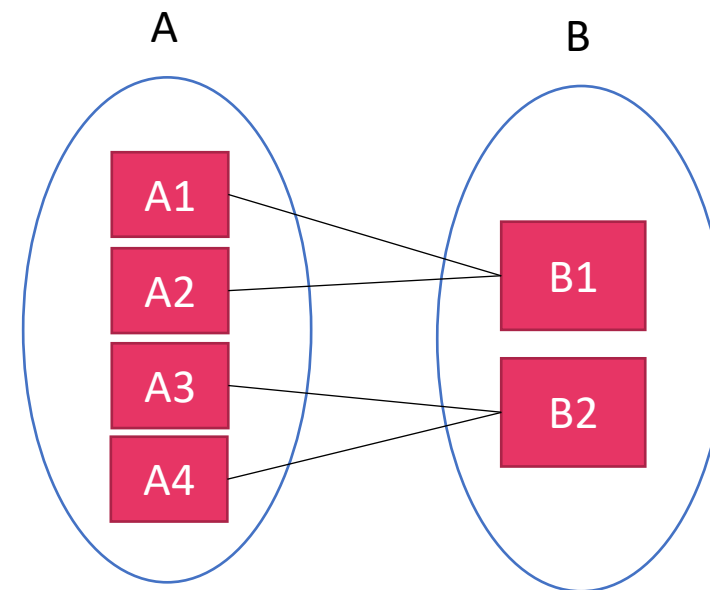
- 1 entidad en A está asociado a N entidades de B.
- 1 entidad en B está asociada a 1 entidad de A.





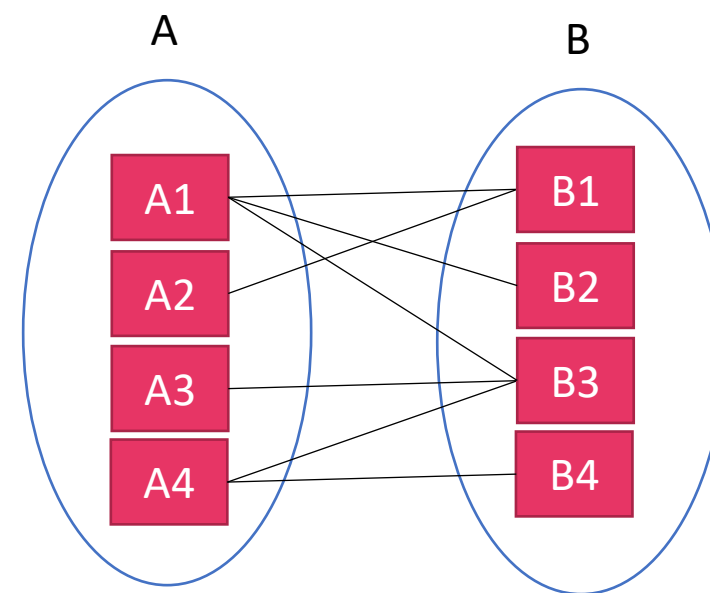
- **Varios a Uno (* - 1)**

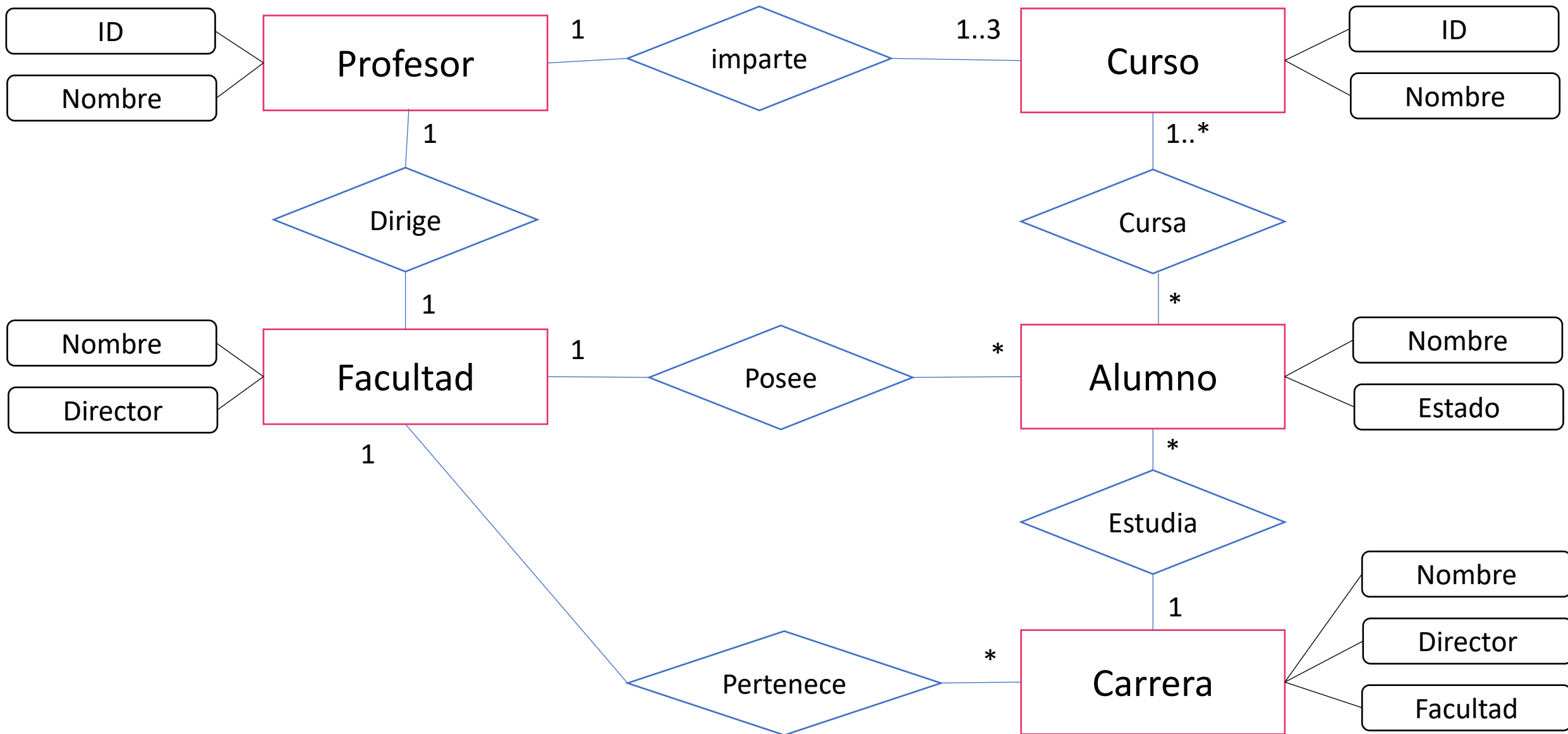
- 1 entidad en A está asociado a 1 entidad de B.
- 1 entidad en B está asociada a **N** entidad de A.



- **Varios a Varios (* - *)**

- 1 entidad en A está asociado a **N** entidades de B.
- 1 entidad en B está asociada a **N** entidad de A.







Actividad

- Tiempo 20 minutos
- Construir el diagrama entidad - relación.

Tipos de relaciones a considerar

1. Una aerolínea posee aviones, como mínimo uno, de lo contrario no podría prestar los servicios de transporte aéreo.
2. Cada avión efectúa una cantidad ilimitada de vuelos, pero cada vuelo es único y se caracteriza por tener un número o identificador único.
3. Un vuelo posee una cantidad de asientos (dependiendo de la cantidad de asientos que posee un avión), considerando que un asiento de un vuelo es aquel que ha sido vendido, por lo tanto un vuelo puede contener múltiples asientos o ninguno.
4. Un pasajero puede realizar una cantidad ilimitada de reservas y cada reserva debe contener como mínimo un asiento, pudiendo también contener varios asientos.
5. Cada reserva es única y se caracteriza por tener un código.
6. Una reserva está asociada a un pago por el valor total del número de asientos vendidos y cada pago está asociado a una única reserva.



El futuro digital
es de todos

MinTIC

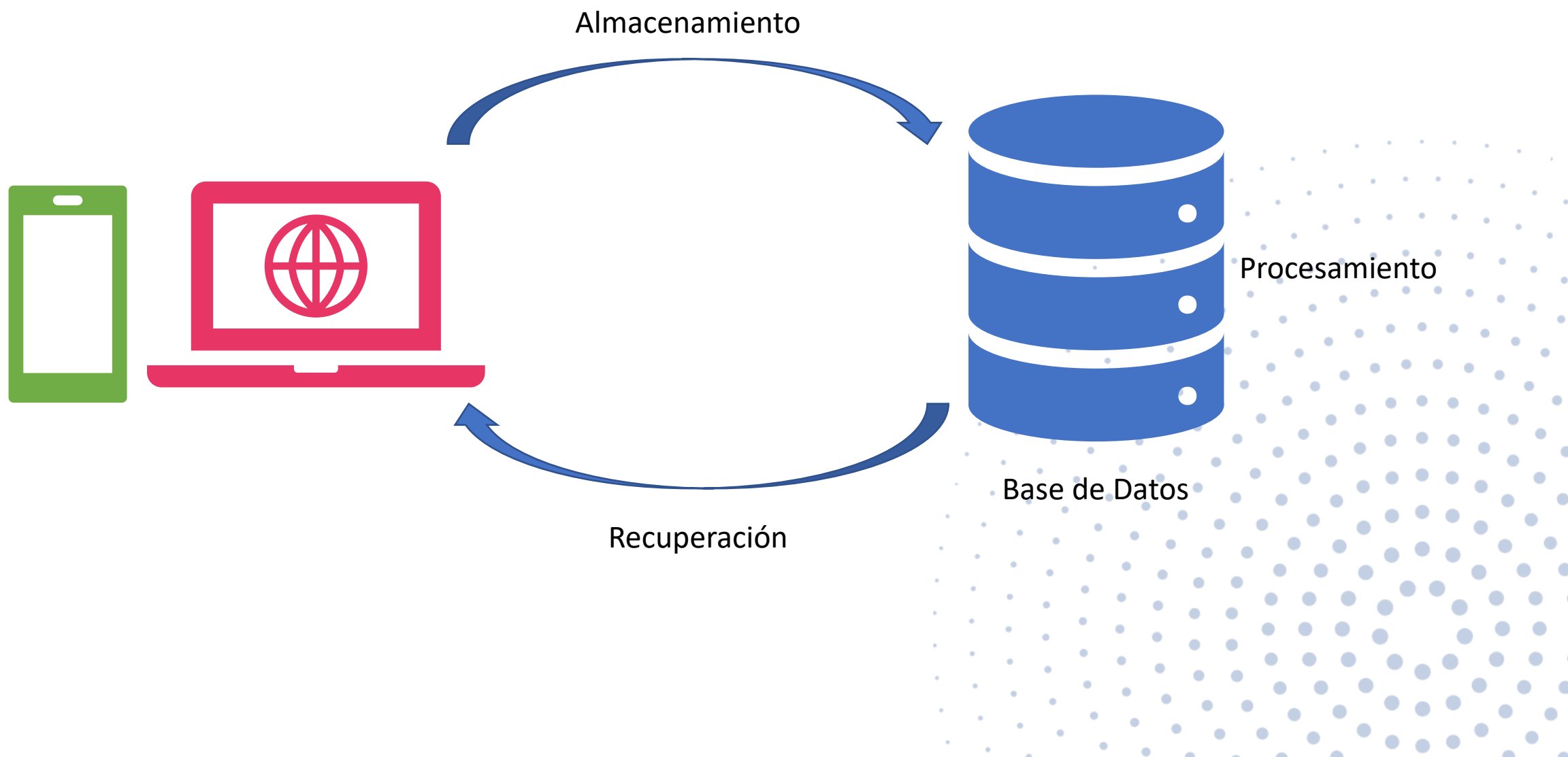


Introducción SQL



El futuro digital
es de todos

MinTIC





¿Qué es SQL?

- SQL son las siglas de Structured Query Language
- SQL le permite acceder y manipular bases de datos
 - SQL puede ejecutar consultas en una base de datos
 - SQL puede recuperar datos de una base de datos
 - SQL puede insertar registros en una base de datos
 - SQL puede actualizar registros en una base de datos
 - SQL puede eliminar registros de una base de datos
 - SQL puede crear nuevas bases de datos
 - SQL puede crear nuevas tablas en una base de datos
 - SQL puede crear procedimientos almacenados en una base de datos
 - SQL puede crear vistas en una base de datos
 - SQL puede establecer permisos en tablas, procedimientos y vistas



Database Management System (DBMS)

El sistema de gestión de bases de datos (DBMS) es utilizado por los usuarios para acceder a los datos almacenados en los archivos de base de datos.

Un DBMS también se utiliza para realizar tareas administrativas en las bases de datos y los objetos contenidos en la base de datos.

DBMS es una colección de aplicaciones que permite a los usuarios y otros programas capturar y analizar datos proporcionando funcionalidad adicional como servicios de informes para ayudarle a crear, implementar y administrar informes para su organización.

Un RDBMS es un sistema de software diseñado para permitir la definición, creación, consulta y actualización de datos almacenados en bases de datos relacionales. Algunos ejemplos de RDBMS incluyen; Microsoft SQL Server, Microsoft Access y MySQL



Declaraciones SQL

La mayoría de las acciones que necesita realizar en una base de datos se realizan con sentencias SQL.

La siguiente instrucción SQL selecciona todos los registros de la tabla “usuarios”

```
SELECT * FROM Usuarios;
```



Algunos de los comandos SQL más importantes

- **SELECT** - extrae datos de una base de datos
- **UPDATE** - actualiza datos en una base de datos
- **DELETE** - elimina datos de una base de datos
- **INSERT INTO** - inserta nuevos datos en una base de datos
- **CREATE DATABASE** - crea una nueva base de datos
- **ALTER DATABASE** - modifica una base de datos
- **CREATE TABLE** - crea una nueva tabla
- **ALTER TABLE** - modifica una tabla
- **DROP TABLE** - elimina una tabla
- **CREATE INDEX** - crea un índice (clave de búsqueda)
- **DROP INDEX** - elimina un índice



Database Management System (DBMS)

El sistema de gestión de bases de datos (DBMS) es utilizado por los usuarios para acceder a los datos almacenados en los archivos de base de datos.

Un DBMS también se utiliza para realizar tareas administrativas en las bases de datos y los objetos contenidos en la base de datos.

DBMS es una colección de aplicaciones que permite a los usuarios y otros programas capturar y analizar datos proporcionando funcionalidad adicional como servicios de informes para ayudarle a crear, implementar y administrar informes para su organización.

Un RDBMS es un sistema de software diseñado para permitir la definición, creación, consulta y actualización de datos almacenados en bases de datos relacionales. Algunos ejemplos de RDBMS incluyen; Microsoft SQL Server, Microsoft Access y MySQL



El futuro digital
es de todos

MinTIC



Sentencias SQL



Sakila database

Descargar y ejecutar los archivos (schema y data) en MySQL Workbench.

<https://dev.mysql.com/doc/index-other.html>

Example Databases

Title	DB Download	HTML Setup Guide	PDF Setup Guide
employee data (large dataset, includes data and test/verification suite)	GitHub	View	US Ltr A4
world database	TGZ Zip	View	US Ltr A4
world_x database	TGZ Zip	View	US Ltr A4
sakila database	TGZ Zip	View	US Ltr A4
airportdb database (large dataset, intended for MySQL on OCI and HeatWave)	TGZ Zip	View	US Ltr A4
menagerie database	TGZ Zip		



SELECT

La instrucción SELECT se utiliza para recuperar filas y columnas de una tabla.

La instrucción SELECT requiere el nombre de la tabla y los nombres de columna * (recupera todas las columnas) o específicos.

Para limitar el número de filas devueltas, puede incluir la cláusula WHERE en la instrucción SELECT

```
SELECT * FROM sakila.customer;
```

```
SELECT first_name,email FROM sakila.customer;
```

```
SELECT * FROM sakila.customer WHERE active = 0;
```

Se pueden
utilizar
operadores
lógicos



SELECT

- Recuperación de filas dentro de un intervalo de fechas mediante la cláusula BETWEEN
- Ordenar el conjunto de resultados mediante ORDER BY para especificar por qué campo se va a ordenar.
- Puede ordenar en orden descendente mediante la cláusula DESC.

```
SELECT * FROM sakila.customer WHERE active BETWEEN 0 AND 1;
```

```
SELECT * FROM sakila.customer WHERE active = 0 ORDER BY  
first_name;
```

```
SELECT * FROM sakila.customer WHERE active = 0 ORDER BY  
first_name DESC;
```




SELECT

- Escribir una consulta para devolver datos que especifiquen lo que no desea que se devuelva.

```
SELECT * FROM sakila.customer WHERE NOT store_id = 2 ORDER  
BY first_name;
```



Funciones agregadas

SQL Server proporciona funciones de agregado para ayudar con el resumen de grandes volúmenes de datos:

```
SELECT count(payment_id) as pagos,  
sum(amount) as suma,  
avg(amount) as promedio,  
min(amount) as minimo,  
max(amount) as maximo  
FROM sakila.payment;
```

	pagos	suma	promedio	minimo	maximo
▶	16049	67416.51	4.200667	0.00	11.99



Inserción de datos

- Puede agregar una nueva fila a una tabla mediante la instrucción INSERT

```
INSERT INTO sakila.actor  
(`first_name`, `last_name`)  
VALUES ("LUISA", "RESTREPO");
```

Puede agregar varias filas a una tabla mediante la siguiente instrucción
INSERT

```
INSERT INTO sakila.actor  
(`first_name`, `last_name`)  
VALUES ("LUISA", "RESTREPO"),  
VALUES ("LAURA", "MESA");
```



Instrucción update

- La instrucción UPDATE se utiliza para modificar los datos que ya están almacenados en una tabla

```
UPDATE sakila.actor  
SET  
first_name="ANDRES",  
last_name = "VELEZ"  
WHERE actor_id = 202;
```



Instrucción DELETE

- La instrucción DELETE se utiliza para eliminar filas de una table

```
DELETE FROM sakila.actor  
WHERE actor_id = 202;
```

Una instrucción DELETE sin una cláusula WHERE hará que se eliminen todas las filas



Cláusula JOIN

La cláusula JOIN permite combinar datos relacionados de varias tablas en un conjunto de resultados

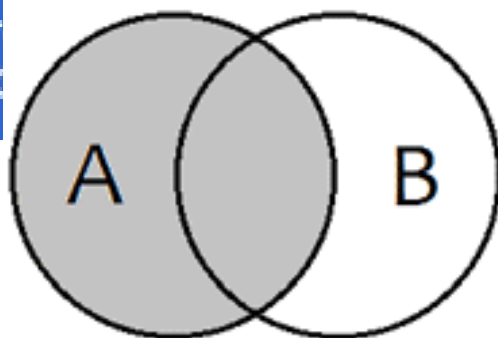
- INNER JOINS utiliza un operador de comparación para hacer coincidir las filas de dos tablas en función de los valores de una columna común que existe en ambas tablas
- OUTER JOINS (izquierda, derecha o completa) incluye filas de una o ambas tablas, incluso si no tienen valores coincidentes
- CROSS JOINS devuelve todas las filas de la tabla izquierda con todas las filas de la tabla derecha. Siempre se deben incluir las condiciones WHERE.



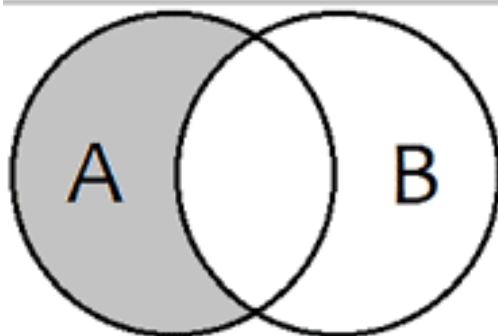
SQL JOINS



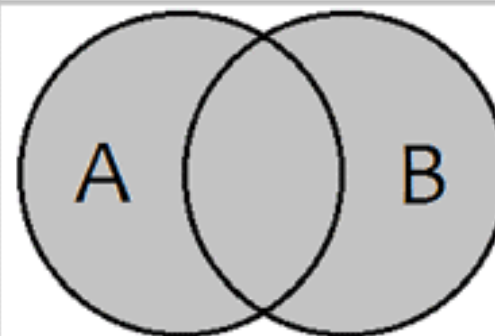
RAÚL PRIETO
FERNÁNDEZ



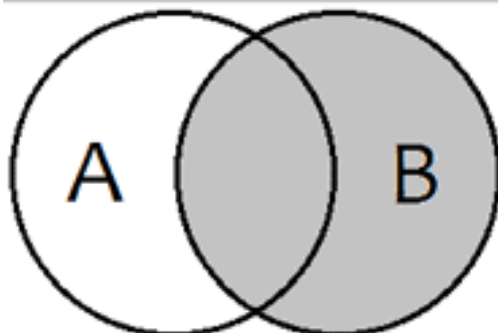
```
SELECT *  
FROM TableA a  
LEFT JOIN TableB b  
ON a.Key = b.Key
```



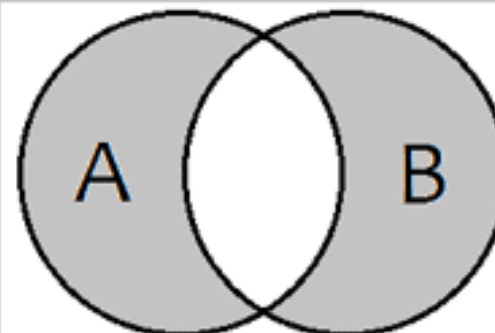
```
SELECT *  
FROM TableA a  
LEFT JOIN TableB b  
ON a.Key = b.Key  
WHERE b.Key IS NULL
```



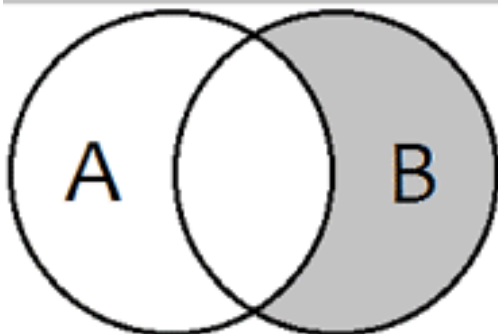
```
SELECT *  
FROM TableA a  
FULL OUTER JOIN TableB b  
ON a.Key = b.Key
```



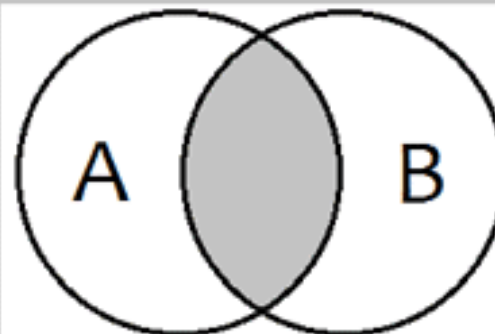
```
SELECT *  
FROM TableA a  
RIGHT JOIN TableB b  
ON a.Key = b.Key
```



```
SELECT *  
FROM TableA a  
FULL OUTER JOIN TableB b  
ON a.Key = b.Key  
WHERE a.Key IS NULL  
OR b.Key IS NULL
```



```
SELECT *  
FROM TableA a  
RIGHT JOIN TableB b  
ON a.Key = b.Key  
WHERE a.Key IS NULL
```



```
SELECT *  
FROM TableA a  
INNER JOIN TableB b  
ON a.Key = b.Key
```




Instrucción JOIN

- La cláusula JOIN permite combinar datos relacionados de varias tablas en un conjunto de resultados

```
SELECT * FROM sakila.actor as actor  
INNER JOIN sakila.film_actor as peli  
ON actor.actor_id = peli.actor_id  
INNER JOIN sakila.film as p  
ON peli.film_id = p.film_id;
```



Common DDL statements

- CREATE – define new entities
- ALTER – modify existing entities
- DROP – remove existing entities



CREATE

Se usa para crear nuevas entidades en SQL.

```
CREATE TABLE city (  
  city_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
  city VARCHAR(50) NOT NULL,  
  country_id SMALLINT UNSIGNED NOT NULL,  
  last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
  PRIMARY KEY (city_id),  
  KEY idx_fk_country_id (country_id),  
  CONSTRAINT `fk_city_country` FOREIGN KEY (country_id) REFERENCES country (country_id) ON DELETE RESTRICT ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



Instrucción ALTER

- Se usa para modificar entidades existentes en SQL.

ALTER DATABASE Sales

Modify Name = SalesForecast ;

Referencias

Brian Alderman, Pete Harris. Database Fundamentals. Microsoft Virtual Academy.

Pohl, K., & Rupp, C. (2011). Requirements Engineering Fundamentals : A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB Compliant. Santa Barbara, CA: Rocky Nook.

SQL, https://www.w3schools.com/sql/sql_syntax.asp