



## CICLO 2

[FORMACIÓN POR CICLOS]

# Programación Básica JAVA

Semana 6



# Programa

<b>Semana 6</b>	Diseño básico de bases de datos	2	5	7
	Bases de datos relacionales (con SQL)	5	12	19

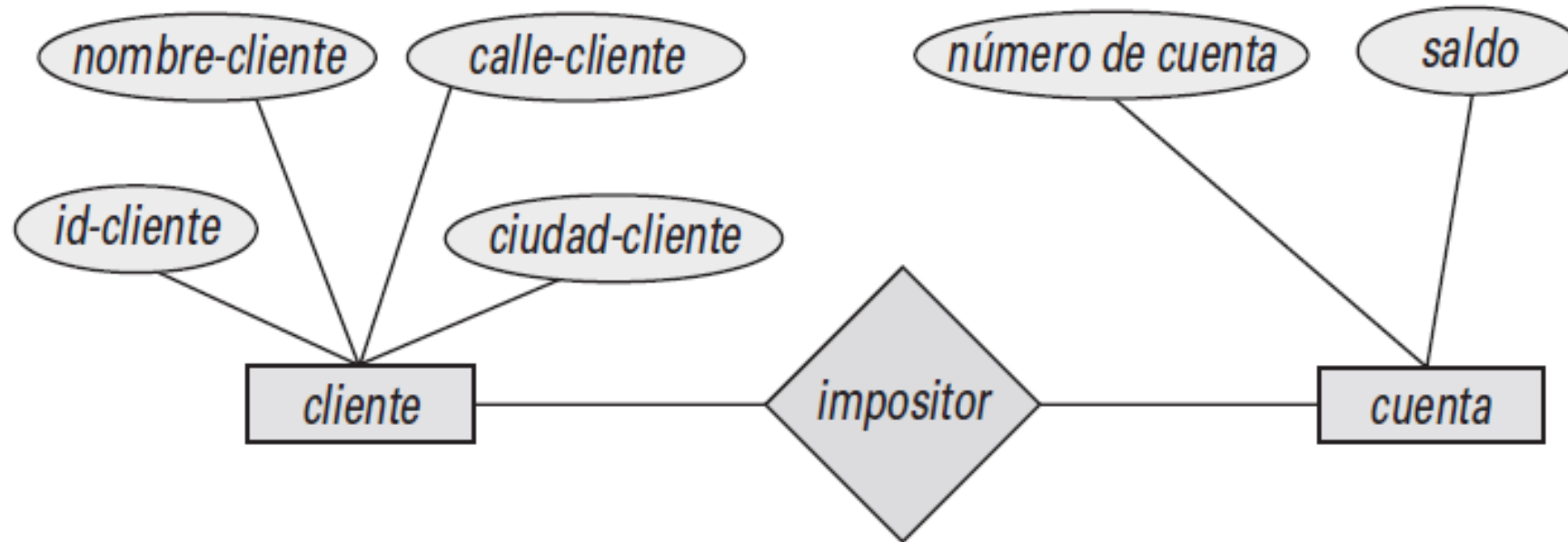
# Temas semana 6

- Junit, entrada consola
  - Reto 3 Socializar
  - Repaso bases de datos
  - Quizizz
  - Descripción entidad-relación
  - Workbench
  - Comandos típicos
- 
- Tienda: Ejemplo Excel → Normalización y Workbench\*
  - Quizizz
  - Aplicaciones

# Modelos Entidad-Relación componentes

- **Rectángulos**, que representan conjuntos de entidades.
- **Elipses**, que representan atributos.
- **Rombos**, que representan relaciones entre conjuntos de entidades.
- **Líneas**, que unen los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones.

# Ejemplo



# Modelo Entidad-Relación

- Si la relación termina en “pata de gallina”, como observamos antes, se trata de una relación de “muchos”.
- Si la relación termina con una o dos líneas perpendiculares, se trata de una relación **obligatoria**.
- Si el final de la relación incluye un círculo, se trata de una relación **condicional**.

# SGBD

Consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada **base de datos**, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto *práctica* como *eficiente*.

# Niveles de normalización



1FN: Eliminar grupos repetidos

2FN: Eliminar datos redundantes

3FN: Eliminar columnas que no dependan de la clave

4FN: Aislar relaciones múltiples independientes

5FN: Aislar relaciones múltiples semánticamente relacionadas



# Normalización

Tener almacenado en el menor espacio posible

Eliminar datos repetidos

Eliminar errores lógicos

Datos ordenados

<https://www.youtube.com/watch?v=bO18omSzeR4>

# Normalización

## Primera Forma Normal

Para que una base de datos esté en **primera forma normal (1FN)** cada columna de una tabla debe ser atómica, es decir, que cada atributo debe contener un único valor del dominio.

### **Un atributo compuesto,**

Por ejemplo: El nombre del cliente: Juan Martínez Arce. Podemos considerar el nombre como un dato atómico o, si nos interesa, separarlo en Apellido1, Apellido2, Nombre.

### **Un atributo multivaluado,**

Por ejemplo: CodCliente, ....., teléfono. En el caso de que quisiéramos recoger distintos números de teléfono para cada cliente tendríamos que crear una nueva fila para cada valor de números de clientes.

# Normalización

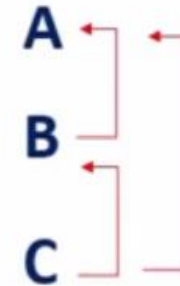
## Segunda Forma Normal

Una tabla está en segunda forma normal (2FN) cuando está en 1FN y además todos los atributos que no forman parte de la clave principal tienen dependencia funcional de la clave completa y no de parte de ella.

### Dependencia Funcional



### Dependencia Transitiva



Hay que eliminar las dependencias parciales, es decir, dependencias basadas en una parte de la clave primaria que producirán redundancias al insertar nuevas filas en la tabla. Para eliminar esas dependencias crearemos nuevas tablas que recojan las dependencias.

# Normalización

## Tercera Forma

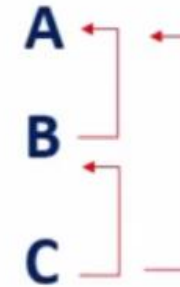
Una base de datos está en **tercera forma normal(3FN)** si está en 2FN y no existen atributos que no pertenezcan a la clave primaria que puedan ser conocidos mediante otro atributo que no forme parte de la clave primaria. Es decir, que no existan dependencias **funcionales transitivas**.

Para eliminar esta dependencia guardamos esos atributos en una nueva tabla.

### Dependencia Funcional



### Dependencia Transitiva



# QUIZIZZ

<https://quizizz.com/join?gc=10987926>

# Comandos más importantes

- **SELECT** - Extrae datos de la base de datos
- **UPDATE** - Actualiza datos en la base de datos
- **DELETE** - Borra datos de la base de datos
- **INSERT INTO** - Agrega nuevos datos a la 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** - Borra una tabla
- **CREATE INDEX** - Crea un índice (busca una llave)
- **DROP INDEX** - Borra un índice

# Comando CREATE DATABASE

```
CREATE DATABASE nombre_database;
```

## Comando CREATE TABLE

```
CREATE TABLE nombre_tabla(  
    columna1 tipodato,  
    columna2 tipodato,  
    columna3 tipodato,  
    . . .  
);
```





# Seleccionar de varias tablas

/\*

Seleccionamos los campos de cada tabla  
creamos unas instancias de cada una  
comparamos las claves primaria y foranea

\*/

```
SELECT c.nombre_1, f.num_compra
```

```
FROM factura f, cliente c
```

```
WHERE f.Cliente_cc = c.cc;
```

```
SELECT cliente.nombre_1, factura.num_compra
```

```
FROM factura , cliente
```

```
WHERE cliente.cc = factura.Cliente_cc ;
```

# Comando ALTER TABLE

```
ALTER TABLE nombre_tabla  
ADD nombre_columna tipodato;
```

## Comando SELECT

```
SELECT columna1, columna2, ...  
FROM nombre_tabla;
```

```
SELECT * FROM nombre_tabla;
```

```
SELECT DISTINCT columna1, columna2, ...  
FROM nombre_tabla;
```

***Extrae solo los datos diferentes***

# Comando WHERE

```
SELECT columna1, columna2, ...  
FROM nombre_tabla  
WHERE condicion;
```

# Operadores clausula WHERE

Operador	Descripción
=	Igual
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
<>	No es igual. <b>Nota:</b> En algunas versiones de SQL este operador puede ser escrito !=
BETWEEN	Entre un cierto rango
LIKE	Buscar por un patrón
IN	Para especificar multiples valores posibles para una columna

# Operadores AND, OR, y NOT

```
SELECT columna1, columna2, ...  
FROM nombre_tabla  
WHERE condicion1 AND condicion2 AND condicion3  
...;
```

## Comando ORDER BY clave

```
SELECT columna1, columna2, ...  
FROM nombre_tabla  
ORDER BY columna1, columna2, ... ASC|DESC;
```



# Comando INSERT INTO

```
INSERT INTO nombre_tabla(columna1, columna2, c  
olumna3, ...)  
VALUES (valor1, valor2, valor3, ...);
```

# Comando UPDATE

```
UPDATE  nombre_tabla  
SET columna1 = valor1, columna2 = valor2, ...  
WHERE condicion;
```

# Comando DELETE

```
DELETE FROM nombre_tabla WHERE condicion;
```

# EJERCICIO

Crear una base de datos llamada grupo 10, con una tabla llamada estudiantes y las columnas nombre1, nombre2, apellido1, apellido2, fecha de Nacimiento, teléfono.

Llenar con datos de del salón de clase. Use el motor de mecanismo de almacenamiento INNODB.

**ENGINE= INNODB;**

# Clausula **SELECT TOP**

```
SELECT nombre_columna(s)  
FROM nombre_tabla  
WHERE condicion  
LIMIT numero;
```

# Funciones SQL COUNT(), AVG(), SUM(), MIN(),MAX()

```
SELECT COUNT(nombre_columna)  
FROM nombre_tabla  
WHERE condicion;
```

```
SELECT AVG(nombre_columna)  
FROM nombre_tabla  
WHERE condicion;
```

## Crear una vista

```
CREATE VIEW mivista as  
SELECT first_name, last_name  
FROM actor  
WHERE first_name='Julia';
```

```
SELECT * FROM mivista;
```

# Clausula SQL JOIN

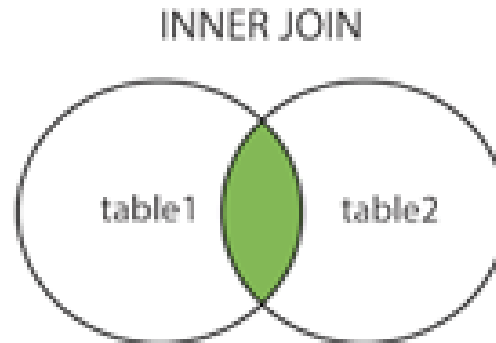
Combina filas de dos o más tablas

- **(INNER) JOIN**: Retorna todos los registros que tenga valores correspondientes en ambas tablas
- **LEFT (OUTER) JOIN**: Retorna todos los registros de la tabla de la izquierda y los valores emparejados en la tabla de la derecha
- **RIGHT (OUTER) JOIN**: Retorna todos los registros de la tabla de la derecha y los valores emparejados en la tabla de la izquierda
- **FULL (OUTER) JOIN**: Retorna todos los registros cuando hay una coincidencia en la tabla izquierda o derecha



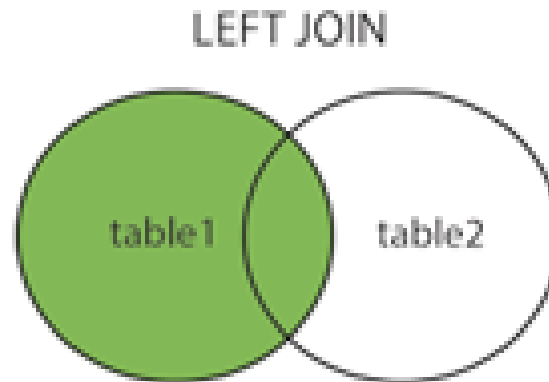
# INNER JOIN

```
SELECT column_name(s)  
FROM table1  
INNER JOIN table2  
ON table1.column_name = table2.column_name;
```



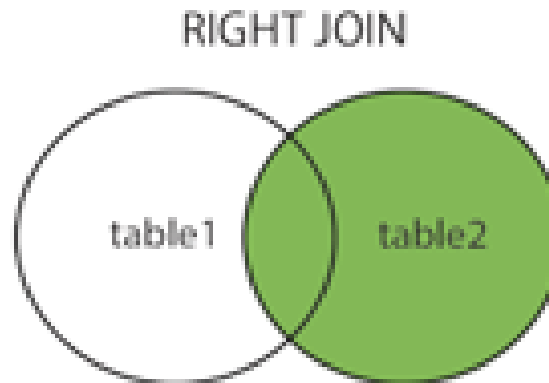
# LEFT JOIN

```
SELECT column_name(s)  
FROM table1  
LEFT JOIN table2  
ON table1.column_name = table2.column_name;
```



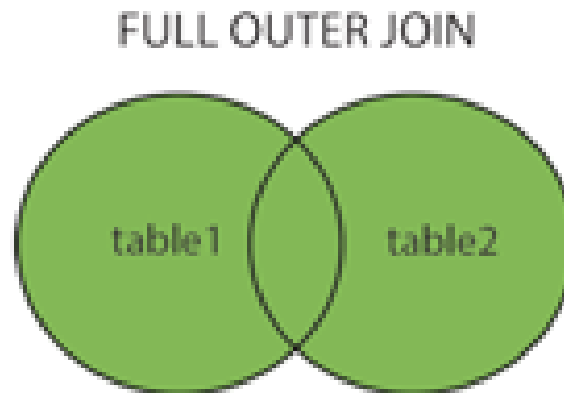
# RIGHT JOIN

```
SELECT column_name(s)  
FROM table1  
RIGHT JOIN table2  
ON table1.column_name = table2.column_name;
```



# FULL JOIN

```
SELECT column_name(s)  
FROM table1  
FULL OUTER JOIN table2  
ON table1.column_name = table2.column_name  
WHERE condition;
```



# QUIZIZZ

<https://quizizz.com/join?gc=18046614>

# EJERCICIO

Trabajar con la base de datos sakila disponible en workbench y encontrar:

# EJERCICIO

1. Clientes que estén activos y el address\_id > 300
2. Actores con el primer nombre 'Julia'
3. Actores con apellido 'Kilmer'
4. De logística indique el nombre del que tiene el password Null
5. Encuentre los lenguajes de las películas
6. Muestre las primeras 5 categorías de películas
7. Películas que no tengan un rating de NC-17
8. Cuente el número de actores
9. Encuentre las Ciudades del country 'Colombia'.
10. ¿Cuántas ciudades hay en country Colombia?

# EJERCICIO

11. De la tabla payment encuentre la cantidad de pagos, el promedio, la suma, el máximo y el mínimo de amount.
12. Cuente el número de películas en la tabla film\_text
13. Muestre las películas de la tabla film\_text que están entre las letras P y la S.
14. Encontrar la mayor duración de una película
15. Encontrar la película de mayor duración (si varias duran lo mismo mostrar las primeras 5)
16. Número de películas que contengan 'k' en su título
17. Películas que contengan 'k' en su título
18. Ciudades de Argentina que tienen en la tercera letra una e



# EJERCICIO

19. Muestre todos los países
20. Liste las ciudades de Brazil
21. Liste las ciudades con nombres compuestos
22. Películas con una duración entre 40 y 60
23. Actores con primeros nombres mayor de 8 caracteres
24. Películas con menor duración
25. Encontrar la película de menor duración (si varias duran lo mismo mostrar las primeras 5)
26. Numero de películas de cada rating
27. Mostrar las ciudades del country correspondiente (multitabla).
28. Mostrar el nombre de la película y el de sus categorías
29. Mostrar la cantidad de datos de payment

# EJERCICIO

- 30. Mostrar la tabla que comprende los pagos desde el 1 de junio de 2005 y el 1 de julio de 2005 y decir cuantos son los registros
- 31. Mostrar los últimos 10 registros
- 32. Mostrar las filas de country donde están los países 'Colombia', 'Argentina', 'Ecuador' y 'Venezuela'
- 33. Mostrar el country, la ciudad y dirección de cada miembro del staff
- 34. Mostrar el nombre de la película y el nombre de los actores
- 35. Mostrar el nombre del actor y las películas en las que trabajo
- 36. Mostrar el id, nombre, apellido, email de cliente, además la suma de la cantidad de la tabla payment y mostrar la suma de manera descendente.

# Base de datos Tienda

<https://www.develou.com/sistema-facturacion-base-datos/>