

AYUDANTÍA N°4 FULL STACK JAVA 0002

PROFESOR AYUDANTE: GERMÁN PINTO RAMÍREZ.
EDUTECNO.



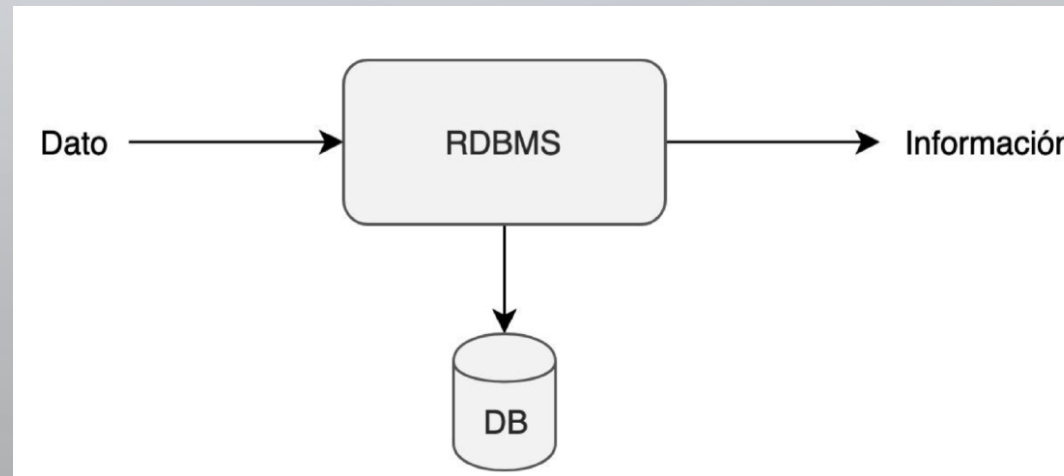
OBJETIVOS

- Aplanar la curva de aprendizaje del curso.
- Mejorar la base de todos los alumnos.
- Adquirir nuevas herramientas para resolver los desafíos del boot camp.
- Aprender a manipular datos mediante sentencias de base de datos

INTRODUCCIÓN A LAS BASES DE DATOS

¿Qué es una base de datos?

Una base de datos es una colección organizada de información estructurada, o datos, típicamente almacenados electrónicamente en un sistema de computadora. Una base de datos es usualmente controlada por un **sistema de gestión de base de datos (DBMS)**. En conjunto, los datos y el DBMS, junto con las aplicaciones que están asociados con ellos, se conocen como un sistema de base de datos.



DATABASE MANAGEMENT SYSTEM (DBMS)

¿Qué es un sistema de administración de base de datos (DBMS)?

Una base de datos generalmente requiere un programa completo de **software** de base de datos, que se conoce como **sistema de administración de bases de datos** (DBMS). Un DBMS sirve como una interfaz entre la base de datos y sus usuarios o programas finales, lo que permite a los usuarios recuperar, actualizar y administrar cómo se organiza y optimiza la información.

Algunos ejemplos de software de bases de datos o DBMS populares incluyen MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, Oracle Database, SQLite y dBASE.



STRUCTURED QUERY LANGUAGE (SQL)

¿ Qué es SQL?

Lenguaje estructurado de consultas: es un lenguaje creado para la definición y la manipulación de bases de datos **relacionales**. El beneficio de este lenguaje es que facilita la administración de datos almacenados.

Palabras clave: `select`, `from`, `where` son identificadores con un significado especial para sql.

```
select pokedex, nombre from pokemones  
where tipo1='fuego';
```

Operadores: `=`, `and` combinación de caracteres para realizar asignaciones o comparar datos

STRUCTURED QUERY LANGUAGE (SQL)

Creando conexión en sql developer

Nueva / Seleccionar Conexión a Base de Datos

Nombre de Cone...	Detalles de Cone...
desafio_ejemplo01	HR@//localhost:1...
ejemploJstl	admin@//localhos...
ensayoPrueba	HR@//localhost:1...
HR	HR@//localhost:1...
MICONEXION	MIUSUARIO@//lo...
MICONEXION2	MIUSUARIO2@//l...
MICONEXION3	USUARIO3@//loc...
MICONEXION4	USUARIO4@//loc...
prueba_horoscopo	ADMIN_HOROSC...
SYS	SYS@//localhost:...
usuario_prueba	alumno@//localh...

Name: MICONEXION4

Tipo de Base de Datos: Oracle

Información de usuario: Usuario de Proxy

Tipo de autenticación: Por defecto

Usuario: USUARIO4

Contraseña:

Rol: valor por defecto

Guardar Contraseña: ☐

Tipo de Conexión: Básico

Detalles: Avanzado

Nombre del Host: localhost

Puerto: 1521

SID: xe

Nombre del Servicio:

Nombre de la conexión: MICONEXION4

Nombre y contraseña del usuario previamente creado

Nombre de mi host es el localhost, porque es mi servidor de bd que se encuentra en mi computadora

Puerto por defecto que reserva Oracle para su conexión (es configurable).

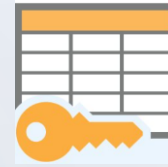
TABLAS

Una base de datos se compone de múltiples tablas. Cada una de éstas presentarán dos dimensiones:

- Filas, que representan a los registros en la tabla.
- Columnas, que van a representar los atributos ingresados en cada registro, definiendo el tipo de dato a ingresar.

desafioBlog.sql * HR * EMPLOYEES *							
Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL							
Ordenar... Filtrar:							
EMPLOYEE...	LASTNAME	FIRSTNAME	EXTENSION	EMAIL	OFFICECODE	REPORTSTO	JOBTITLE
1	1002 Murphy	Diane	x5800	dmurphy@classicmodelcars.com	1	(null)	President
2	1056 Patterson	Mary	x4611	mpatterso@classicmodelcars.com	1	1002	VP Sales
3	1076 Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com	1	1002	VP Marketing
4	1088 Patterson	William	x4871	wpatterson@classicmodelcars.com	6	1056	Sales Manager (APAC)
5	1102 Bondur	Gerard	x5408	gbondur@classicmodelcars.com	4	1056	Sale Manager (EMEA)
6	1143 Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)
7	1165 Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep
8	1166 Thompson	Leslie	x4065	lthompson@classicmodelcars.com	1	1143	Sales Rep
9	1188 Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	2	1143	Sales Rep
10	1216 Patterson	Steve	x4334	spatterson@classicmodelcars.com	2	1143	Sales Rep
11	1286 Tseng	Foon Yue	x2248	ftseng@classicmodelcars.com	3	1143	Sales Rep
12	1323 Vanauf	George	x4102	gvanauf@classicmodelcars.com	3	1143	Sales Rep
13	1337 Bondur	Loui	x6493	lbondur@classicmodelcars.com	4	1102	Sales Rep
14	1370 Hernandez	Gerard	x2028	ghernande@classicmodelcars.com	4	1102	Sales Rep
15	1401 Castillo	Pamela	x2759	pcastillo@classicmodelcars.com	4	1102	Sales Rep
16	1501 Bott	Larry	x2311	lbott@classicmodelcars.com	7	1102	Sales Rep
17	1504 Jones	Barry	x102	bjones@classicmodelcars.com	7	1102	Sales Rep
18	1611 Fixter	Andy	x101	afixter@classicmodelcars.com	6	1088	Sales Rep
19	1612 Marsh	Peter	x102	pmarsh@classicmodelcars.com	6	1088	Sales Rep
20	1619 King	Tom	x103	tking@classicmodelcars.com	6	1088	Sales Rep
21	1621 Nishi	Mami	x101	mnishi@classicmodelcars.com	5	1056	Sales Rep
22	1625 Kato	Yoshimi	x102	ykato@classicmodelcars.com	5	1621	Sales Rep
23	1702 Gerard	Martin	x2312	mgerard@classicmodelcars.com	4	1102	Sales Rep

CLAVES

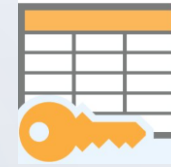


Buscan generar identificadores para cada registro de estas tablas mediante algún valor específico de una columna o atributo.

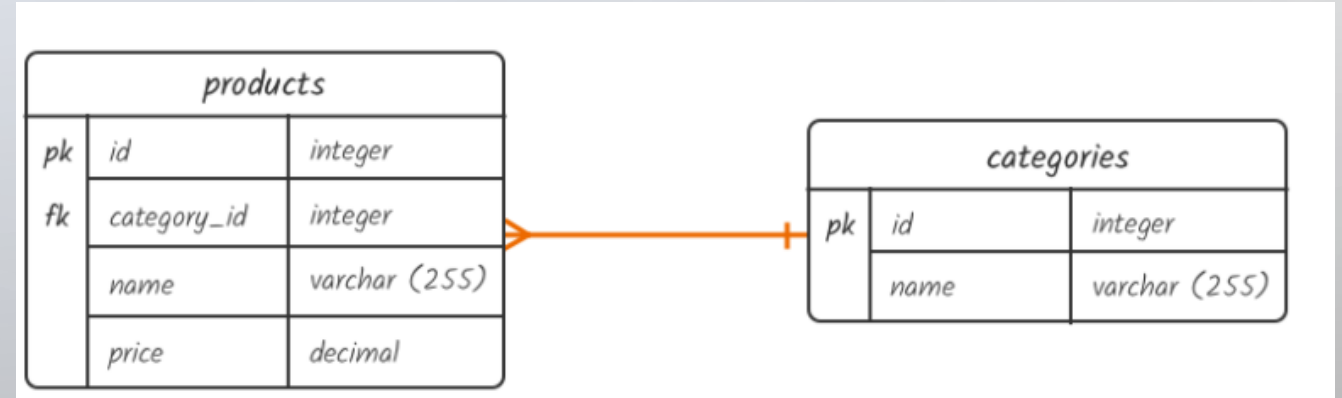
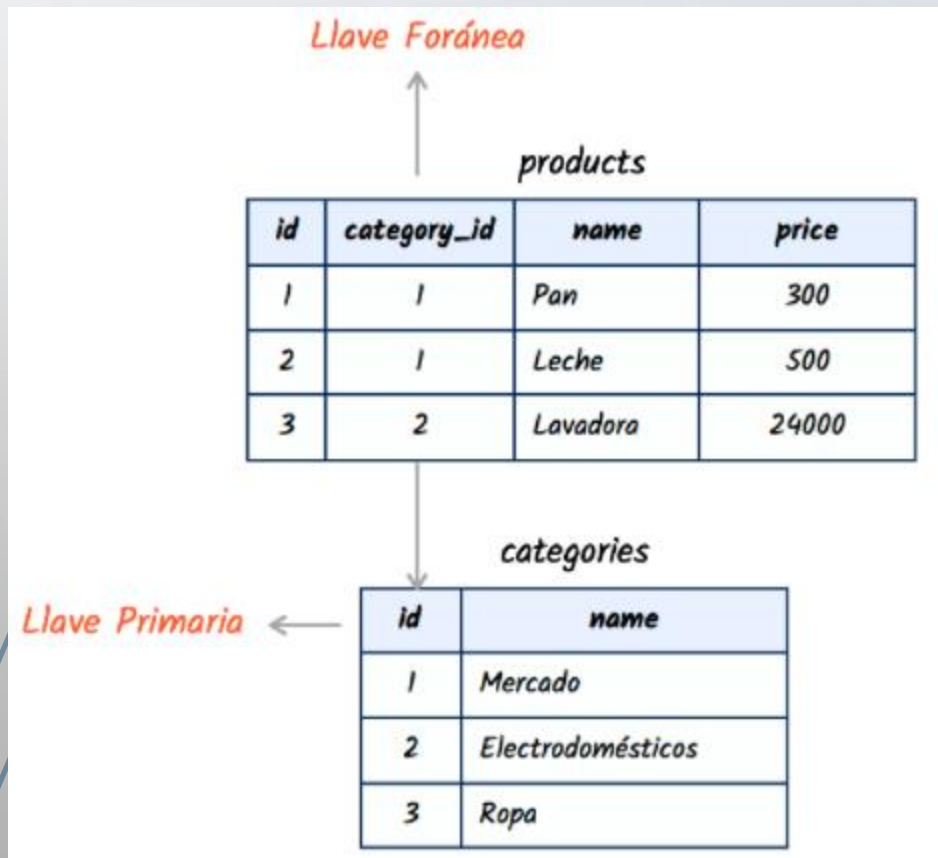
Clave primaria (primary key): Cuando hacemos referencia a una columna dentro de su tabla de origen, hablaremos de una clave primaria. Esta clave siempre será de carácter único.

Clave foránea (foreign key): Cuando hacemos referencia a una columna identificadora en otra tabla a la cual hacemos referencia, hablamos de una clave foránea.

CLAVES



En el siguiente ejemplo tenemos una tabla con productos y otra con categorías. Cada producto pertenece a una categoría, por eso productos tiene como atributo (llave foránea) la llave primaria de la categoría.



CREACIÓN DE TABLAS

La creación de la tabla permitirá almacenar los datos en filas y columnas.

```
CREATE TABLE nombre_tabla(  
  columna1 tipo_de_dato1,  
  columna2 tipo_de_dato2,  
  columna3 tipo_de_dato3,  
  PRIMARY KEY (columnaN)  
)
```

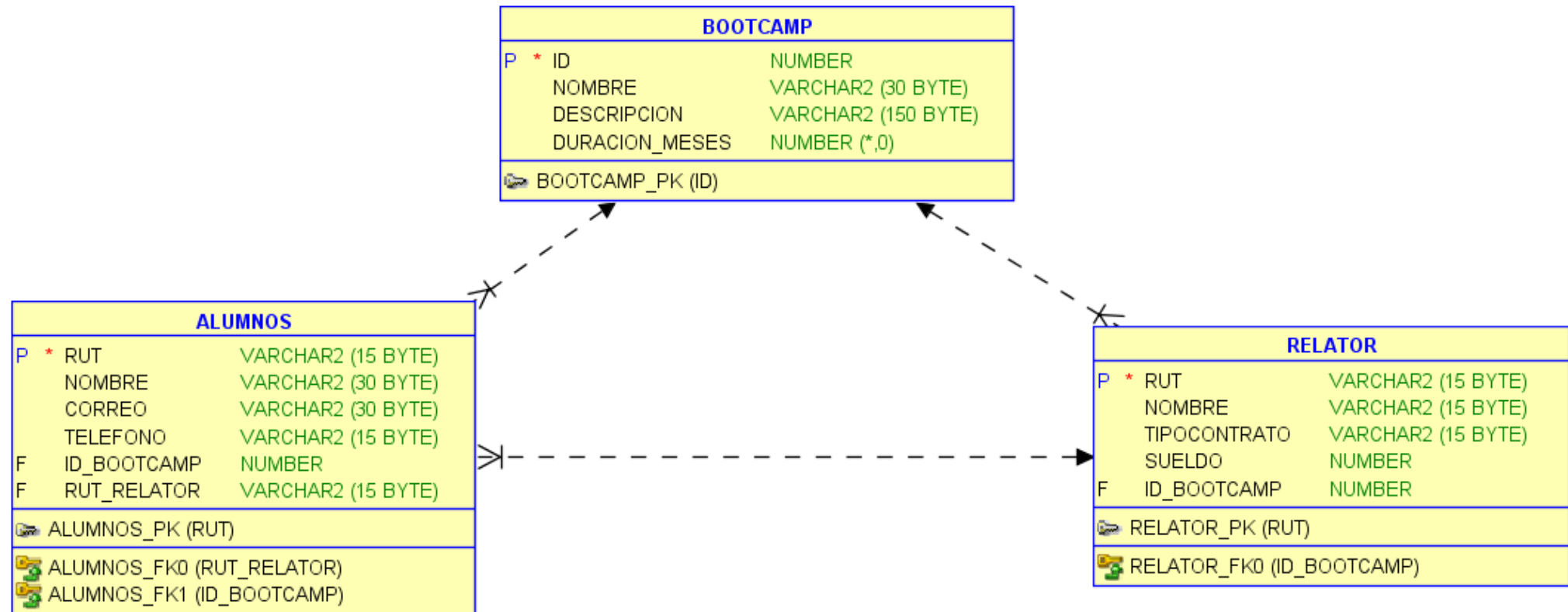
```
CREATE TABLE post  
(  
  id NUMBER,  
  usuario VARCHAR2(20),  
  fecha_creacion DATE,  
  contenido VARCHAR2(200),  
  descripcion VARCHAR2(255),  
  PRIMARY KEY (id)  
);
```

```
CREATE TABLE ventas  
(  
  fecha      VARCHAR2(20),  
  id_auto    NUMBER,  
  cliente    VARCHAR2(25),  
  referencia NUMBER,  
  cantidad   NUMBER(10, 2),  
  FOREIGN KEY (id_auto) REFERENCES autos(id)  
);
```

```
CREATE TABLE post  
(  
  id NUMBER PRIMARY KEY ,  
  usuario_id NUMBER NOT NULL REFERENCES usuario(id) ,  
  titulo VARCHAR2( 100 ) NOT NULL UNIQUE,  
  fecha DATE  
);
```

EJEMPLO PRÁCTICO

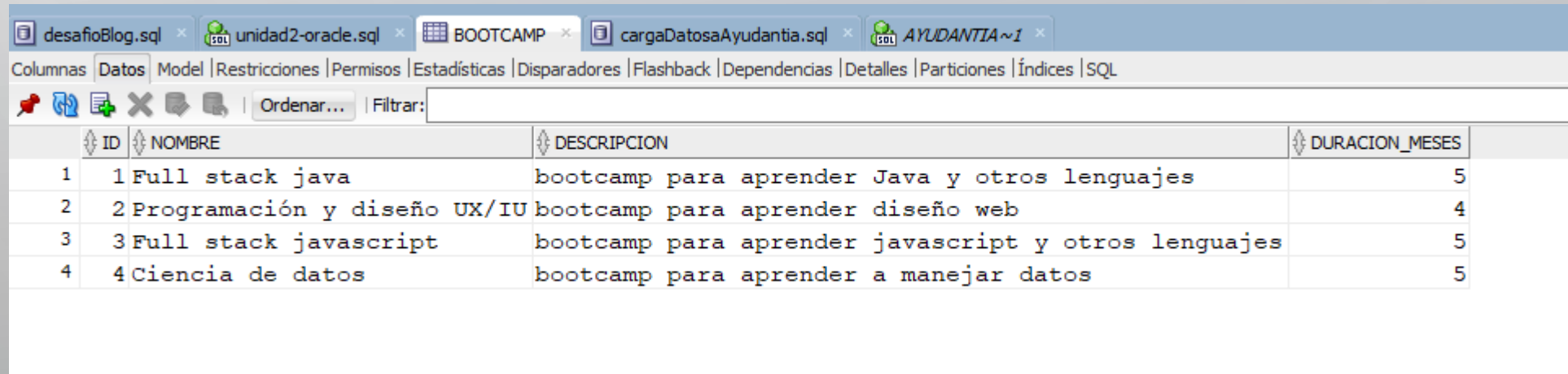
Para este ejemplo usemos un esquema que conozcamos todos. Haremos un BD que refleje de forma compacta como se relacionan los distintos bootcamp de talentodigital con los alumnos y profesores:



EJEMPLO PRÁCTICO

Empezaremos a crear las tablas y poblar nuestra BD:

```
/*CREACION DE LAS TABLAS*/  
CREATE TABLE BOOTCAMP  
(  
  ID NUMBER,  
  NOMBRE VARCHAR(30),  
  DESCRIPCION VARCHAR(150),  
  DURACION_MESES INT,  
  PRIMARY KEY (ID)  
)
```



The screenshot shows a database management interface with several tabs at the top: desafioBlog.sql, unidad2-oracle.sql, BOOTCAMP, cargaDatosAyudantia.sql, and AYUDANTIA~1. The 'BOOTCAMP' tab is active, displaying the table structure and data. The interface includes a menu bar with options like Columnas, Datos, Model, Restricciones, Permisos, Estadísticas, Disparadores, Flashback, Dependencias, Detalles, Particiones, Índices, and SQL. Below the menu, there are icons for various database actions and a search bar labeled 'Filtrar:'. The table data is as follows:

ID	NOMBRE	DESCRIPCION	DURACION_MESES
1	Full stack java	bootcamp para aprender Java y otros lenguajes	5
2	Programación y diseño UX/IU	bootcamp para aprender diseño web	4
3	Full stack javascript	bootcamp para aprender javascript y otros lenguajes	5
4	Ciencia de datos	bootcamp para aprender a manejar datos	5


INSTRUCCIÓN SELECT

Nos sirve para obtener información de las columnas que cumplan las condiciones indicadas. Podemos seleccionar columnas de una tabla o todas las columnas con el carácter '*'.

58 | select * from alumnos; 59 |

Salida de Script x

Resultado de la Consulta x

 SQL

Todas las Filas Recuperadas: 7 en 0,003 segundos

	RUT	NOMBRE	CORREO	TELEFONO	ID_BOOTCAMP	RUT_RELATOR
1	1111111-1	Juan Perez	juanP@mimail.com	5555555	1	16018433-3
2	2222222-2	Alan Brito	alanB@mimail.com	7777777	1	16018433-3
3	3333333-3	Jesse Pinkman	jpinkman@mailly.com	8888888	1	16018433-3
4	4444444-3	Camila Flores	cami@mimail.com	44444444	2	16018444-3
5	5555555-1	Alvaro Soto	alvaro@mimail.com	99999	2	16018444-3
6	7777777-2	Felipe Sureda	felipS@mimail.com	7777777	4	16018777-3
7	8888888-3	Skayler White	skayler@mailly.com	8888888	4	16018777-3

OPERACIONES CONDICIONALES

Operador	Descripción
=	Igual
>	Mayor que
<	Menor que
<>, !=	Distinto. El operador != es solo válido en algunas versiones de SQL
>=	Mayor o igual
<=	Menor o igual
BETWEEN	Entre cierto rango
LIKE	Por patrón
IN	Para especificar múltiples valores posibles para una columna

SELECT CON WHERE

Podemos consultar algunas columnas, hacer filtrado de la información de la query con la clausula where, también podemos usar los conectores 'and' y 'or'.

```
6 select rut, nombre, correo from alumnos
7 where rut like '%-2' or correo='alvaro@mimail.com';
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 3 en 0,095 segundos

	RUT	NOMBRE	CORREO
1	2222222-2	Alan Brito	alanB@mimail.com
2	5555555-1	Alvaro Soto	alvaro@mimail.com
3	7777777-2	Felipe Sureda	felipS@mimail.com

```
8 select * from relator where tipocontrato='contrato' and sueldo>1500000;
9
```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,05 segundos

	RUT	NOMBRE	TIPOCONTRATO	SUELDO	ID_BOOTCAMP
1	16018555-3	Vicente McMAN	contrato	2000000	3

FUNCIONES EN CONSULTAS

Para tener más opciones de filtrar la información de nuestras consultas, podemos ocupar las operaciones o funciones:

- **MIN:** Entrega el mínimo de los datos de una columna.
- **MAX:** Entrega el máximo de los datos de una columna.
- **LENGTH:** Calcula el largo de los datos en una columna.
- **COUNT:** Cuenta la cantidad de ocurrencias de las filas.
- **SUM:** Suma los valores de una columna ignorando los valores null.

EJEMPLOS

Cuantos registros tendremos en la tabla bootcamp:

```
9
10 select count(*) as cant_bootcap from bootcamp;
11
```

Resultado de la Consulta x	
SQL Todas las Filas Recuperadas: 1 en 0,002 segundos	
CANT_BOOTCAP	
1	4

Cuál o cuales es o son los relatores que gana más:

```
12 select * from relator
13 where sueldo= (select max(sueldo) from relator);
```

Resultado de la Consulta x				
SQL Todas las Filas Recuperadas: 2 en 0,002 segundos				
RUT	NOMBRE	TIPOCONTRATO	SUELDO	ID_BOOTCAMP
1 16018433-3	Walter White	honorario	2000000	1
2 16018555-3	Vicente McMAN	contrato	2000000	3

Cuanto es lo que dura el bootcamp con menor duración:

```
16 select min(duracion_meses)menor_duracion from bootcamp;
```

Resultado de la Consulta x	
SQL Todas las Filas Recuperadas: 1 en 0,002 segundos	
MENOR_DURACION	
1	4

Cuanto gasta la institución en los pagos a docentes:

```
19 select sum(sueldo)monto_sueldos from relator;
```

Resultado de la Consulta x	
SQL Todas las Filas Recuperadas: 1 en 0,002 segundos	
MONTO_SUELDOS	
1	6700000

EJERCICIO CONSULTAS SOBRE UNA TABLA

Cree consultas a la bd que muestre:

1. Cantidad de bootcamp que son full stack.
2. LA suma de los sueldos de los docentes que trabajan con contrato.
3. El largo del correo más extenso de los alumnos. (muestre solo el número).
4. El nombre del bootcamp que dura menos.

AGRUPACIÓN CON GROUP BY

Agrupar filas que tengan datos iguales. Dejará las filas juntas según los valores de la columna indicada. La instrucción GROUP BY a menudo se usa con funciones agregadas (COUNT, MAX, MIN, SUM) para agrupar el conjunto de resultados por una o más columnas.

```
36 select count(tipocontrato), tipocontrato
37 from relator
38 group by tipocontrato;
```

	COUNT(TIPOCONTRATO)	TIPOCONTRATO
1	2	contrato
2	2	honorario

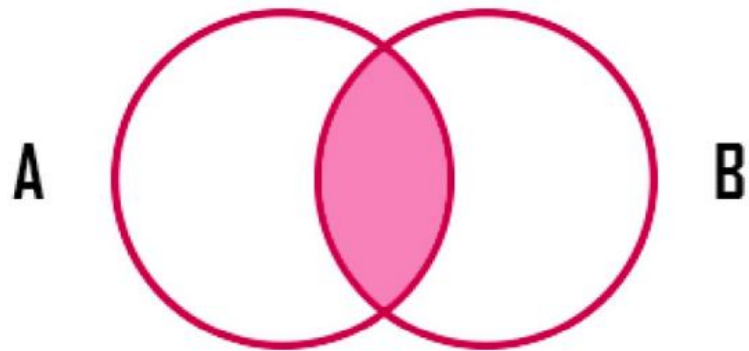
```
41 select sum(sueldo), tipocontrato from relator
42 group by tipocontrato;
```

	SUM(SUELDO)	TIPOCONTRATO
1	3200000	contrato
2	3500000	honorario

OPERACIONES DE UNIÓN ENTRE TABLAS

- INNER JOIN: Devuelve **todas las filas** cuando hay al menos **una coincidencia** en **ambas** tablas.

INNER JOIN



```
SELECT columnas  
FROM A  
INNER JOIN B  
ON A.columna=B.columna
```

```
45 select rel.nombre nom_prof, bot.nombre nom_curso  
46 from relator rel  
47 inner join bootcamp bot  
48 on rel.id_bootcamp = bot.id;  
49
```

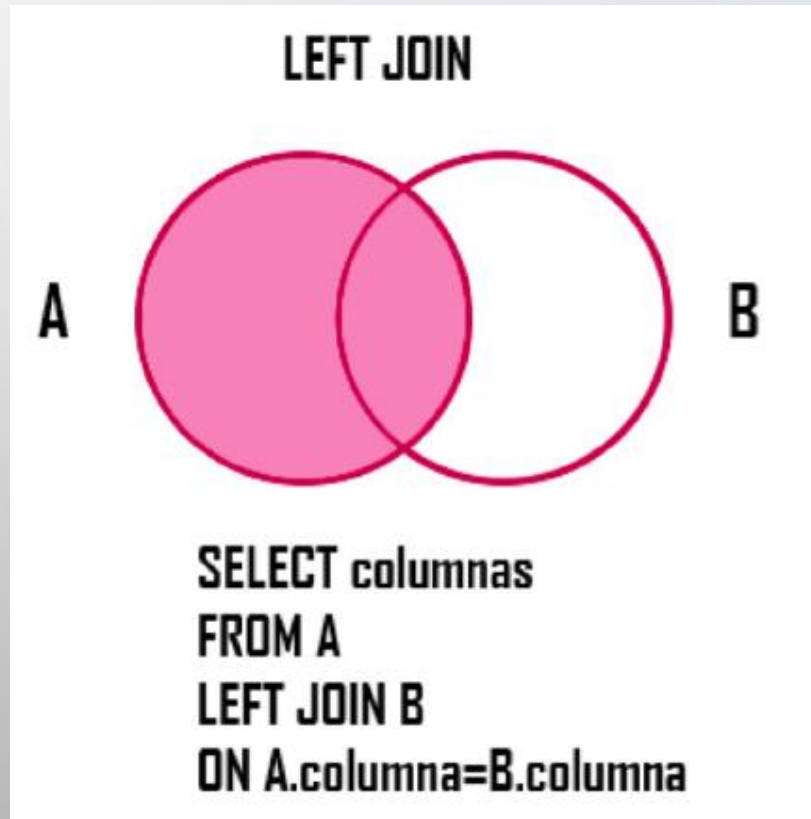
Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 4 en 0,003 segundos

	NOM_PROF	NOM_CURSO
1	Walter White	Full stack java
2	Steve Austin	Programación y diseño UX/IU
3	Vicente McMan	Full stack javascript
4	Juana Snow	Ciencia de datos





OPERACIONES DE UNIÓN ENTRE TABLAS

- LEFT JOIN: Devuelve todas las filas de la tabla de la **izquierda**, y las filas coincidentes de la tabla de la **derecha**.



```
50 select bot.nombre, al.nombre nom_alu, al.correo
51 from bootcamp bot
52 left join alumnos al
53 on bot.id= al.id_bootcamp;
54
```

Salida de Script x Resultado de la Consulta x

    SQL | Todas las Filas Recuperadas: 8 en 0,004 segundos

	NOMBRE	NOM_ALU	CORREO
1	Full stack java	Juan Perez	juanP@mimail.com
2	Full stack java	Alan Brito	alanB@mimail.com
3	Full stack java	Jesse Pinkman	jpinkman@mailly.com
4	Programación y diseño UX/IU	Camila Flores	cami@mimail.com
5	Programación y diseño UX/IU	Alvaro Soto	alvaro@mimail.com
6	Ciencia de datos	Felipe Sureda	felipS@mimail.com
7	Ciencia de datos	Skayler White	skayler@mailly.com
8	Full stack javascript	(null)	(null)

OPERACIONES DE UNIÓN ENTRE TABLAS

Antes de pasar al right join, hagamos una modificación a nuestra BD y agreguemos un nuevo curso:

```
insert into bootcamp  
(ID,NOMBRE,DESCRIPCION,DURACION_MESES)  
values(5,'Front-End Trainee','bootcamp para aprender  
JavaScript y otros lenguajes',5);
```

De esta forma podemos tener un nuevo valor para ejemplificar.

OPERACIONES DE UNIÓN ENTRE TABLAS

- RIGHT JOIN: Devuelve todas las filas de la tabla de la **derecha**, y las filas coincidentes de la tabla de la **izquierda**.

```
7 select rel.nombre nom_relator, boot.nombre curso
3 from relator rel
9 right join bootcamp boot
0 on rel.id_bootcamp=boot.id;
```

Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 5 en 0,003 segundos

	NOM_RELATOR	CURSO
1	Walter White	Full stack java
2	Steve Austin	Programación y diseño UX/IU
3	Vicente McMan	Full stack javascript
4	Juana Snow	Ciencia de datos
5	(null)	Front-End Trainee

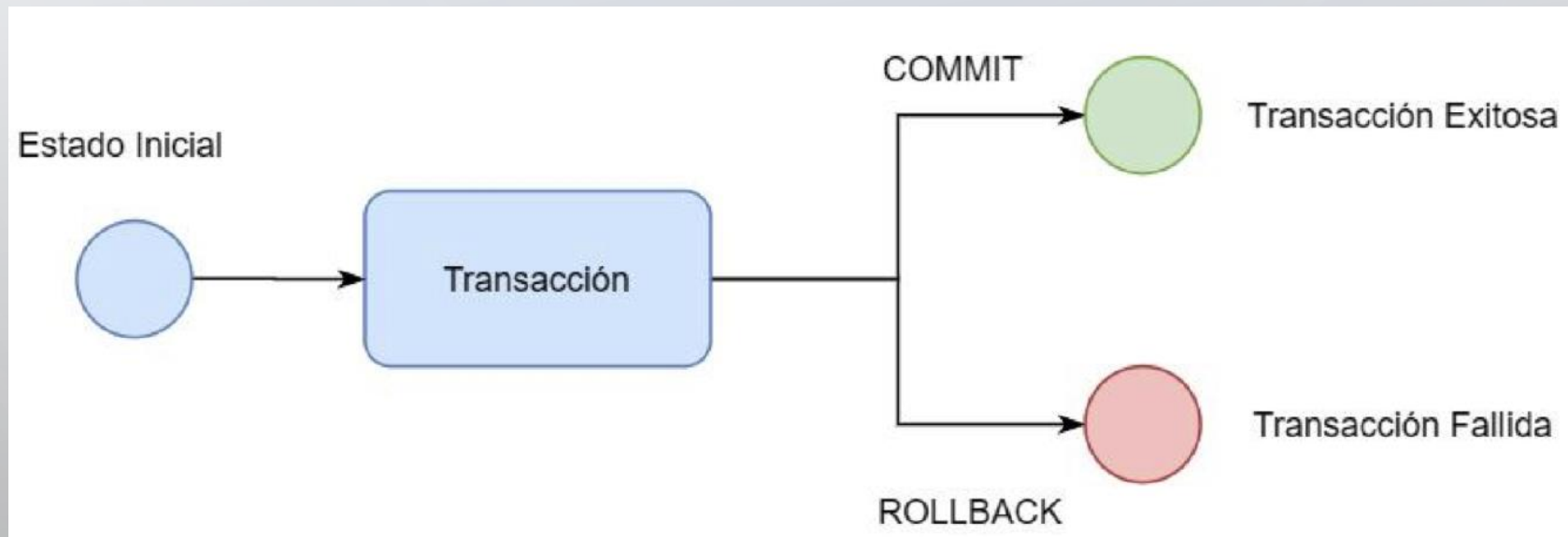
EJERCICIO CONSULTAS AGRUPADAS

Cree consultas a la bd que muestre:

1. Nombre del relator y alumno, donde el relator tiene el rut 16018433-3.
2. Nombre y duración de los bootcamp en donde los relatores ganan 2 millones.
3. Muestre el nombre y cantidad de alumnos de los profesores. (pista: use group by).
4. Muestre el nombre y la descripción de los bootcamp que no tienen alumnos.

TRANSACCIONES

Una **transacción** es una o varias sentencias SQL (SELECT, INSERT, UPDATE y/o DELETE) que se ejecutan en una base de datos como una única operación, confirmándose (commit) o deshaciéndose(rollback) en grupo.



COMANDOS USADOS EN TRANSACCIONES

Comando	Descripción
BEGIN	El sistema permite que se ejecuten todas las sentencias SQL que necesitemos.
COMMIT	Guarda los cambios de la transacción.
ROLLBACK	Retrocede los cambios realizados.
SAVEPOINT	Guarda el punto de partida al cual volver a la hora de aplicar ROLLBACK.
SET TRANSACTION	Le asigna nombre a la transacción.

La sintaxis de estos comandos es la siguiente:

- COMMIT;
- SAVEPOINT nombre_savepoint;
- ROLLBACK [TO nombre_savepoint];

Lo que está entre corchetes es de carácter opcional, por lo que podemos decirle con ROLLBACK a qué punto volver. Este volverá al último punto guardado o por defecto al estado inicial de la transacción.