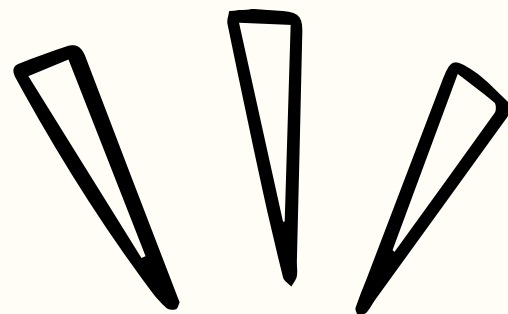




COMANDOS MYSQL



Lo primero que debemos hacer es crear
la base de datos

CREATE DATABASE HOSPITAL;

Luego debemos indicar que queremos
usar esa base de datos

USE HOSPITAL;

Ahora crearemos las tablas

Consejo: Empezar por las que no tienen foreign key

```
CREATE TABLE CONSULTORIO (ID_CONS INT PRIMARY KEY,  
NOMBRE VARCHAR(30),  
DIRECCION VARCHAR(50));
```

*Indico que este atributo
será la primary key*

*Indico el tipo de dato
y su longitud*

Si debo crear una tabla que su **foreign key** apunta a la creada anteriormente lo hago de la siguiente manera:

```
CREATE TABLE MEDICO (ID_MEDICO INT PRIMARY KEY,  
NOMBRE VARCHAR(30),  
ESPECIALIDAD VARCHAR(50),  
ID_CONSULTORIO_FK INT,  
FOREIGN KEY(ID_CONSULTORIO_FK) REFERENCES CONSULTORIO(ID_CONS));
```

Indico que es FK

*Señalo el atributo
que es FK*

*Indico con qué
tabla se relaciona*

*a qué atributo
de esa tabla*

! EL TIPO DE DATO DE ID_CONSULTORIO_FK Y EL DE ID_CONS DEBEN SER IGUALES

Cree una tabla y me olvide de indicar cuál es su Primary Key ¿Qué hago?

```
CREATE TABLE PACIENTE (DNI_PACIENTE BIGINT,  
NOMBRE VARCHAR(30),  
EDAD INT,  
DOMICILIO VARCHAR(30));
```

Uso el comando para modificar una tabla
alter table

```
ALTER TABLE PACIENTE ADD PRIMARY KEY(DNI_PACIENTE);
```

Si debo crear una tabla que posee **primary key** compuesta y dos **foreign key** lo hago de la siguiente manera:

```
CREATE TABLE CITA ( DNI_PACIENTE BIGINT,  
                    ID_MEDICO INT,  
                    FECHA DATE,  
                    PRIMARY KEY (DNI_PACIENTE, ID_MEDICO),  
                    FOREIGN KEY (DNI_PACIENTE) REFERENCES PACIENTE(DNI_PACIENTE),  
                    FOREIGN KEY (ID_MEDICO) REFERENCES MEDICO(ID_MEDICO));
```

¿Qué más puedo hacer con el *alter table*?

→ Modificar el tipo de dato de una columna

```
ALTER TABLE PACIENTE MODIFY COLUMN EDAD FLOAT;
```

→ Modificar el nombre de una columna

```
ALTER TABLE CITA RENAME COLUMN FECHA TO FECHA_CITA;
```

→ Agregar nueva columna

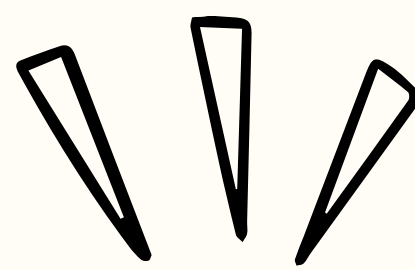
```
ALTER TABLE MEDICO ADD MAIL VARCHAR(50);
```

→ Eliminar una columna

```
ALTER TABLE PACIENTE DROP COLUMN DOMICILIO;
```



INSERCIÓN DE DATOS EN TABLAS



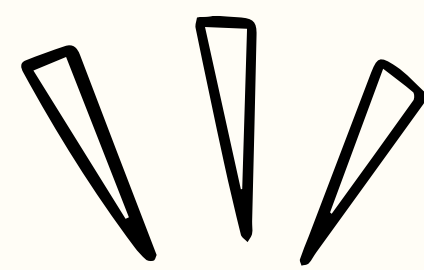
Para insertar datos en una tabla debemos usar el comando *insert into*

```
INSERT INTO CONSULTORIO (NOMBRE, DIRECCION, ID_CONS)  
VALUES ("ARCOIRIS", "CORRIENTES 700", 1);
```

Debo colocar los datos que ingresaré en el orden que detalle las columnas entre paréntesis

Si no, debo respetar el orden de cómo cree la tabla

```
INSERT INTO CONSULTORIO VALUES (2, "NUBES", "CORRIENTES 702");
```

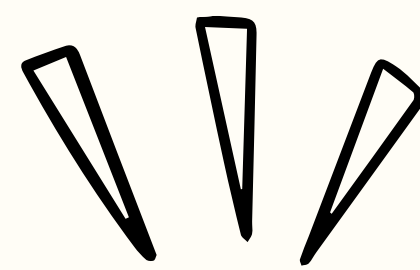


De esta manera ingresamos varios registros en la misma query

```
INSERT INTO CONSULTORIO VALUES (3, "ARBOL", "CORRIENTES 704"),  
(4, "CACTUS", "CORRIENTES 706"),  
(5, "AGUA", "CORRIENTES 708");
```

Los registros se ven así en la tabla

id_cons	nombre	direccion
1	Arcoiris	Corrientes 700
2	Nubes	Corrientes 702
3	Arbol	Corrientes 704
4	Cactus	Corrientes 706



Para ingresar datos en una tabla que posee FK, ese dato ya debe existir en la otra tabla

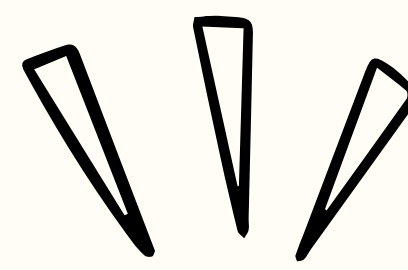
INSERT INTO MEDICO VALUES (256, "RENÉ FAVALORO", "CARDIÓLOGO", "4", "RENE@MAIL.COM");

id_cons	nombre	direccion
1	Arcoiris	Corrientes 700
2	Nubes	Corrientes 702
3	Arbol	Corrientes 704
4	Cactus	Corrientes 706

id_medico	nombre	especialidad	id_consultorio_fk	mail
256	René Favalaro	Cardiólogo	4	rene@mail.com

PRIMERAS QUERYS





Para ver todos los registros de una tabla uso

select * from tabla

SELECT * FROM CONSULTORIO;

	id_cons	nombre	direccion
▶	1	Arcoiris	Corrientes 700
	2	Nubes	Corrientes 702
	3	Arbol	Corrientes 704
	4	Cactus	Corrientes 706

Si quiero ver sólo alguna columna en específico lo debo aclarar luego del select

SELECT NOMBRE, ESPECIALIDAD, MAIL FROM MEDICO;

	nombre	especialidad	mail
▶	René Favalaro	Cardiólogo	rene@mail.com

Dentro de la query puedo poner condiciones con el **where**

```
SELECT * FROM CONSULTORIO WHERE ID_CONS = 3;
```

	id_cons	nombre	direccion
▶	3	Arbol	Corrientes 704

Los operadores que puedo usar son **=** , **<** , **>** , **<=** , **>=**

```
SELECT * FROM CONSULTORIO WHERE ID_CONS <= 3;
```

	id_cons	nombre	direccion
	1	Arcoiris	Corrientes 700
	2	Nubes	Corrientes 702
	3	Arbol	Corrientes 704

Dentro de la query puedo poner VARIAS condiciones en el where

```
SELECT * FROM CONSULTORIO WHERE ID_CONS >= 2 AND ID_CONS <=4;
```

id_cons	nombre	direccion
2	Nubes	Corrientes 702
3	Arbol	Corrientes 704
4	Cactus	Corrientes 706

```
SELECT * FROM CONSULTORIO WHERE ID_CONS >= 2 OR ID_CONS <=4;
```

id_cons	nombre	direccion
1	Arcoiris	Corrientes 700
2	Nubes	Corrientes 702
3	Arbol	Corrientes 704
4	Cactus	Corrientes 706

SELECT * FROM CONSULTORIO WHERE NOMBRE = 'ARCOIRIS' AND ID_CONS = 1;

id_cons	nombre	direccion
1	Arcoiris	Corrientes 700

SELECT * FROM CONSULTORIO WHERE NOMBRE = 'ARCOIRIS' OR ID_CONS = 4;

id_cons	nombre	direccion
1	Arcoiris	Corrientes 700
4	Cactus	Corrientes 706

al poner = lo que esta a su derecha debe coincidir exacto

Si quiero que el filtro coincida con solo una parte uso el comando **like**
me ayudo con el símbolo **%**

→ Datos que el nombre empiece con 'A'

```
SELECT * FROM CONSULTORIO WHERE NOMBRE LIKE 'A%';
```

id_cons	nombre	direccion
1	Arcoiris	Corrientes 700
3	Arbol	Corrientes 704

→ Datos que el nombre contenga 'u'

```
SELECT * FROM CONSULTORIO WHERE NOMBRE LIKE '%U%';
```

id_cons	nombre	direccion
2	Nubes	Corrientes 702
4	Cactus	Corrientes 706

→ Datos que el nombre finalice con 's'

```
SELECT * FROM CONSULTORIO WHERE NOMBRE LIKE '%S';
```

id_cons	nombre	direccion
1	Arcoiris	Corrientes 700
2	Nubes	Corrientes 702
4	Cactus	Corrientes 706

También puedo indicar en función de la cantidad de caracteres que tiene un dato para cada caracter usamos un `_`

→ Datos el nombre tenga 5 caracteres

```
SELECT * FROM CONSULTORIO WHERE NOMBRE LIKE '_____';
```

id_cons	nombre	direccion
2	Nubes	Corrientes 702
3	Arbol	Corrientes 704

Puedo indicar en qué posición tiene que estar determinado caracter

→ Datos que la tercera letra de su nombre sea 'c'

```
SELECT * FROM CONSULTORIO WHERE NOMBRE LIKE '__c%';
```

id_cons	nombre	direccion
1	Arcoiris	Corrientes 700
4	Cactus	Corrientes 706

Negaciones dentro de la query usando <> y NOT LIKE

→Mostrar los consultorios cuyo id sea *distinto a 3*

```
SELECT * FROM CONSULTORIO WHERE ID_CONS <> 3;
```

	id_cons	nombre	direccion
▶	1	Arcoiris	Corrientes 700
	2	Nubes	Corrientes 702
	4	Cactus	Corrientes 706

→Mostrar los consultorios cuyo nombre NO contenga 'a'

```
SELECT * FROM CONSULTORIO WHERE NOMBRE NOT LIKE '%A%';
```

id_cons	nombre	direccion
2	Nubes	Corrientes 702

Ordenar los dato con **ORDER BY**

→ASC (De menor a mayor o de la A a la Z)

```
SELECT * FROM CONSULTORIO ORDER BY ID_CONS ASC;
```

id_cons	nombre	direccion
1	Arcoiris	Corrientes 700
2	Nubes	Corrientes 702
3	Arbol	Corrientes 704
4	Cactus	Corrientes 706

→DESC (De mayor a menor o de la Z a la A)

```
SELECT * FROM CONSULTORIO ORDER BY NOMBRE DESC;
```

id_cons	nombre	direccion
2	Nubes	Corrientes 702
4	Cactus	Corrientes 706
1	Arcoiris	Corrientes 700
3	Arbol	Corrientes 704