SHOW DATABASES; //MOSTRAR DATABASES

CREATE DATABASE appsalon;

Tipos de datos

INT numero

FLOAT decimales

DOUBLE decimal grande

DECIMAL valores donde no se acepta redondeos

VARCHAR texto corto y texto hasta 255 caracteres

CHAR corto, como un pasword encriptado

TEXT Texto largo

BLOB imágenes, sonidos, y archivos comprimidos (no recomentadble)

DATE AAAA-MM-DD

TIME hh:mm:ss

DATETIME fechas

**CREAR TABLA**

CREATE TABLE servicios (

id INT(11) NOT NULL AUTO\_INCREMENT,

nombre VARCHAR(60 NOT NULL),

precio DECIMAL(5,2) NOT NULL,

PRIMARY KEY(id) );

DESCRIBE servicios; //ver estructura

CRUD(Create read update delete)

**INSERTAR**

INSERT INTO servicios (nombre, precio) VALUES (“Corte de cabello adulto”, 804);

INSERT INTO servicios (nombre, precio) VALUES (“Corte de cabello niño”, 804);

INSERT INTO servicios(nombre, precio) VALUES (“CORte mujer”,349),(“peinado”,50443);

**SELECCIONAR**

SELECT \* FROM servicios;

SELECT nombre FROM servicios;//solo una tabla

SELECT nombre, precio FROM servicios

SELECT id, nombre, precio FROM servicios ORDER BY precio ASC;(o DESC)//ascendente o descendente

SELECT id, nombre, precio FROM servicios ORDER BY id DESC LIMIT 2;//ordena descendientemente por id 2 valores

SELECT id, nombre, precio FROM servicios WHERE id= 4;

**ACTUALIZAR**

UPDATE servicios SET precio = 70 WHERE id=2;//where obligatorio

UPDATE servicios SET nombre = “CORTE peinado” WHERE id=2;

UPDATE servicios SET nombre =”actualizado”, precio= 494 WHERE id = 23;

**DELETE**

DELETE FROM servicios WHERE id = 1;

Nota: SQL continua el número de id con los que fueron agregados, no remplaza 1 id eliminado

MODIFICAR BASE DE DATOS

Agregar campo a una table

ALTER TABLE servicios ADD descripción VARCHAR(100) NOT NULL;

Modificar tabla existente//no se puede modificar el tipo

ALTER TABLE servicios CHANGE descripción nuevonombre ;

ELIMINAR TABLA

DROP TABLE servicios;

////////////////////////////////////crear otra tabla

CREATE TABLE servicios (

Id INT(11) NOT NULL AUTO\_INCREMENT,

Nombre VARCHAR(60) NOT NULL,

Precio DECIMAL(6,2) NOT NULL,

PRIMARY KEY(Id));

CREATE TABLE reservaciones(

Id INT(11) NOT NULL AUTO\_INCREMENT,

Nombre VARCHAR(60) NOT NULL,

Apellido VARCHAR(60) NOT NULL,

Hora TIME DEFAULT NULL,

Fecha DATE DEFAULT NULL,

Servicios VARCHAR(255) NOT NULL,

PRIMARY KEY(ID)

);

Seleccionar según parametro

SELECT \* FROM servicios WHERE precio > 90;

Rango

SELECT \*FROM servicios WHERE precio BETWEEN 100 AND 200;

**AGREGADORAS**

SELECT COUNT(id), fecha

FROM reservaciones

GROUP BY fecha

ORDER BY COUNT(id) DESC;

Suma de todos los servicios

SELECT SUM(precio) AS totalServicios FROM servicios;

EL menor valor

SELECT MIN(precio) AS precioMenor FROM servicios;

EL mayor valor

SELECT MAX(precio) AS precioMenor FROM servicios;

**BUSQUEDAS SEGÚN PARAMETRO**

SELECT \* FROM servicios WHERE nombre LIKE ‘Corte%’//Inicia con la palabra corte y %Corte% en algún lugar de la cadena de texto

**CONCATENACION**

SELECT CONCAT(nombre, ‘ ’, apellido) AS nombreCOmpleto FROM reservaciones;

SELECT \* FROM reservaciones WHERE CONCAT(nombre, “ ”, apellido) LIKE ‘%Ana Preciado%’;

SELECT hora, fecha, CONCAT(nombre, “ ”, apellido) as ‘Nombre completo’ FROM reservaciones WHERE CONCAT(nombre, “ ”, apellido) LIKE ‘%Ana preciado%’

**MULTIPLES CONDICIONES**

SELECT \* FROM reservaciones WHERE id IN(1,3) //muestra las r con id 1 y 3

SELECT \* FROM reservaciones WHERE fecha=”2021-06-5”;

SELECT \* FROM reservaciones WHERE fecha=”2021-06-5 AND id=1 AND nombre= “nombre”;

NORMALIZACIÓN

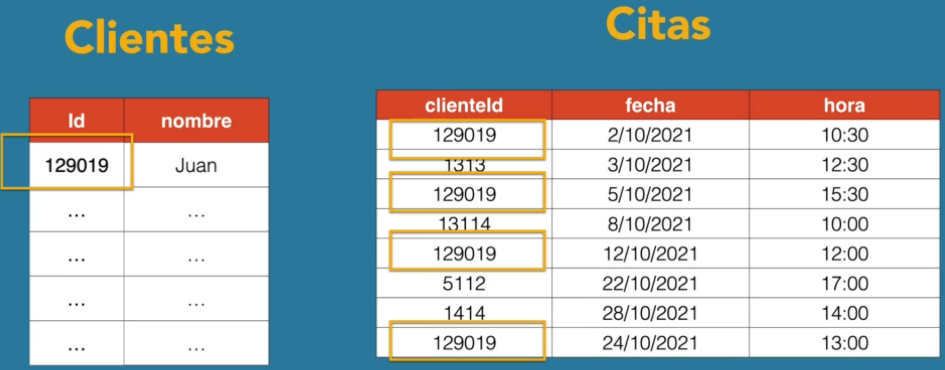
1NF,2NF,3NF

1NF

Cada columna debe tener solo1 valor y no debe haber grupos repetidos.

2NF

Llaves primarias compuestas



3NF

Relación de no llaves compuestas



DENORMALIZACION

Algunas veces se encuentran bases de datos que rompen la normalización



**DIAGRAMA ENTIDAD RELACION**

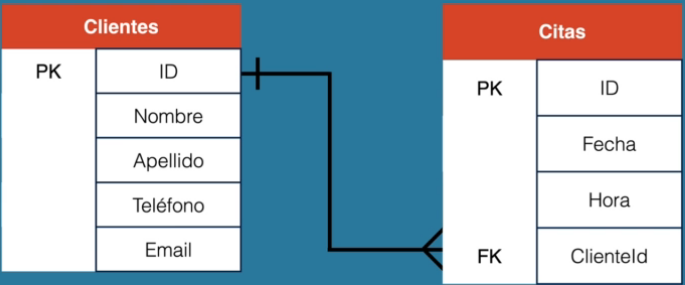
**Tabla:**



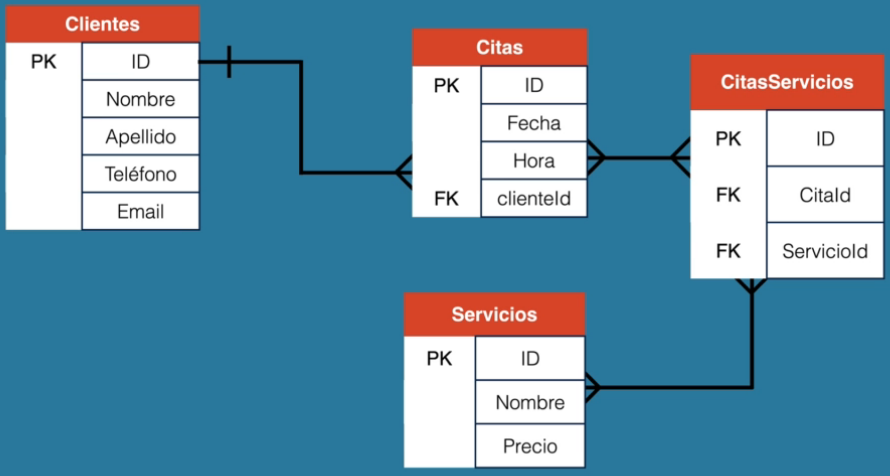
La cardinalidad es el número de veces que una instancia se relaciona con otra



Relación



Finalmente



CREATE TABLE clientes (

Id INT(11) NOT NULL AUTO\_INCREMENT,

Nombre VARCHAR(60) NOT NULL,

Apellido VARCHAR(60) NOT NULL,

Teléfono VARCHAR(60) NOT NULL,

Email VARCHAR(30) NOT NULL UNIQUE

PRIMARY KEY(Id)

);

INSERT INTO clientes (nombre, apellido, teléfono,email) VALUES (“Juan”,”Alvarez”,”1827634”,”peroo@as.com”);

SELECT \* FROM clientes;

CREATE TABLE citas (

Id INT(11) NOT NULL AUTO\_INCREMENT,

Fecha DATE NOT NULL,

Hora TIME NOT NULL,

ClienteID INT(11) NOT NULL,

PRIMARY KEY (Id),

KEY clienteID (clienteID),

CONSTRAINT cliente\_FK

FOREING KEY (clienteID)

REFERENCES clientes(Id)

);

INSERT INTO citas (fecha, hora, clienteID) VALUES ( ‘2021-23-3’, 10:30:00, 1);

UNIR 2 TABLAS

SELECT \* FROM citas

INNER JOIN clientes ON clientes.id = citas.clienteID;

**TABLA PIVOTE muchos a muchos**

CREATE TABLE citasServicio (

Id INT(11) NOT NULL AUTO\_INCREMENT,

citaID INT(11) NOT NULL,

servicioID INT(11) NOT NULL,

PRIMARY KEY (Id),

KEY citaID (citaID),

CONSTRAINT citas\_FK FOREING KEY (citasID)

REFERENCES citas(Id)

KEY servicioID (servicioID),

CONSTRAINT servicios\_FK FOREING KEY(servicioID),

REFERENCES servicios (id) );

**CONSULTAS TABLA PIVOTE**

SELECT \* FROM citasServicios

INNER JOIM citas ON citas.id = citasServicio.citaID

INNER JOIN servicios ON servicios.id = citasServicio.servicioID;

SELECT \* FROM citasServicios

INNER JOIM citas ON citas.id = citasServicio.citaID

INNER JOIN servicios ON servicios.id = citasServicio.servicioID;

SELECT \* FROM citasServicios

INNER JOIN citas ON citas.id = citasServicios.citaId

INNER JOIN clientes ON citas.clienteId = clientes.id

INNER JOIN servicios ON servicios.id = citasServicios.serviciosID;