1. Ejecutar el siguiente Script para crear y cargar la tabla

CREATE TABLE PACIENTE

(ID\_PACIENTE INT PRIMARY KEY,

APELLIDO VARCHAR(30) NOT NULL,

NOMBRE VARCHAR(30) NOT NULL,

DOB DATE NOT NULL,

GENERO CHAR(1) NOT NULL,

PESO float NOT NULL,

ALTURA float NOT NULL,

ESTA\_VACUNADO CHAR(1) NOT NULL);

INSERT INTO PACIENTE (ID\_PACIENTE, APELLIDO, NOMBRE, DOB,

GENERO, PESO, ALTURA, ESTA\_VACUNADO)

VALUES

(15223, 'Smith', 'Deniz', '2018/12/31', 'F', 21.4, 29.2, 'Y'),

(15224, 'Agarwal', 'Arjun', '2017/08/29', 'M', 28.1, 34.2, 'Y'),

(15225, 'Adams', 'Poppy', '2015/02/14', 'F', 34.0, 39.2, 'N'),

(15226, 'Johnson', 'Tierra', '2019/08/15', 'F', 14.6, 24.5, 'Y'),

(15227, 'Khouri', 'Mohammed', '2014/03/30', 'M', 41.5, 44.1, 'Y'),

(15228, 'Jones', 'Ben', '2011/04/04', 'M', 70.1, 52.2, 'Y'),

(15229, 'Kowalczyk', 'Alexandra', '2019/08/27', 'F', 15.2, 23.9, 'Y');

**2. Insertar adicionalmente dos pacientes dentro de la tabla.**

insert into paciente (id\_paciente,apellido,nombre,dob,genero,peso,altura,esta\_vacunado)

values(14100, 'viltez', 'hernan','1984/06/15', 'M', 80.4,40.5, 'Y'),

(14101, 'viltez', 'javier','1999/06/15', 'M', 89.4,45.5, 'N');

**3. Actualizar el estado de Vacunado en YES para aquellos pacientes que son de Genero F y el DOB está entre el 2015 y 2019.**

update paciente set esta\_vacunado='Y'

where genero='F' and dob between '2015/1/1' and '2019/1/1';

**4. Actualizar la columna Genero de F a Femenino de las pacientes 15223, 15225 y 15229.**

alter table paciente alter column genero type varchar(20);

update paciente set genero='Femenino'

where id\_paciente in (15223, 15225, 15229);

**5. Incrementar en 2kg aquellos pacientes que su altura este por arriba de los 39.**

update paciente set peso= peso+2

where altura>39;

**6. Eliminar de la tabla aquellos pacientes donde en el campo Nombre contengan la secuencia de letras “am”.**

delete from paciente

where nombre like '%am%';

**7. Eliminar de la tabla aquellos pacientes donde el Genero sea M, no esté vacunado y el DOB este entre el 2014 y 2015.**

insert into paciente (id\_paciente,apellido,nombre,dob,genero,peso,altura,esta\_vacunado)

values(14108, 'viltez', 'hernan','2014/06/15', 'M', 80.4,40.5, 'N');

delete from paciente

where genero= 'M'

and esta\_vacunado='N'

and dob between '2014/1/1' and '2015/1/1';

**8. Borrar el contendido de toda la tabla de manera más ágil y con el menor costo.**

delete from paciente;

**9. Eliminar la Tabla.**

drop table paciente;

select \* from almacen;

select \* from articulo;

select \* from material;

select \* from ciudad;

select \* from proveedor;

drop table proveedor;

create table almacen

(nro integer primary key,

responsable varchar(50) not null);

create table articulo

(codArt int primary key,

descripcion varchar(50) not null,

precio float);

create table material

(codMat int primary key,

descripcion VARCHAR(50) not null);

create table ciudad

(codCiudad int primary key,

nombre varchar(30), not null);

create table proveedor

(codProv int primary key,

nombre varchar(30) not null,

domicilio varchar(50) not null,

codCiudad int not null,

constraint FK\_codCiudad

foreign key (codCiudad)

references ciudad(codCiudad));

create table contiene

(cod\_contiene serial primary key ,

nro int not null,

codArt int not null,

constraint FK\_nro\_almacen

foreign key (nro)

references almacen(nro),

constraint FK\_articulo

foreign key (codArt)

references articulo(codArt));

drop table contiene;

create table compuesto\_por

(cod\_composicion serial primary key,

codArt int not null,

codMat int not null,

constraint FK\_articulo foreign key(codArt) references articulo(codArt),

constraint FK\_material foreign key(codMat) references material(codMat));

drop table compuesto\_por;

create table provisto\_por

(cod\_provisto serial primary key,

codMat int not null,

codProv int not null,

constraint FK\_material foreign key(codMat) references material(codMat),

constraint FK\_proveedor foreign key(codProv) references proveedor(codprov));

drop table provisto\_por;

INSERT INTO ALMACEN (Nro, Responsable)

VALUES (001, 'Alfredo'),

(002, 'Esteban'),

(003, 'Juan'),

(004, 'Roberto');

INSERT INTO ARTICULO (CodArt, Descripcion, Precio)

VALUES (001, 'Pan', 130.70),

(002, 'Facturas', 300.00),

(003, 'Cheese Cake', 450.87),

(004, 'Pasta Frola', 278.90);

INSERT INTO MATERIAL (CodMat, Descripcion)

VALUES (001, 'Aceite'),

(002, 'Harina'),

(003, 'Levadura'),

(004, 'Huevo'),

(005, 'Azucar'),

(006, 'Sal'),

(007, 'Agua');

INSERT INTO CIUDAD(CodCiudad, Nombre)

VALUES (1, 'La Plata'),

(2, 'Capital Federal'),

(3, 'Ramos Mejia'),

(4, 'La Matanza');

INSERT INTO PROVEEDOR (CodProv, Nombre, Domicilio,

CodCiudad)

VALUES(1, 'Arcor', 'Ayacucho 1234', 1),

(2, 'Molinos', 'Yatay 456', 4),

(3, 'Ledesma', 'Mario Bravo 987', 1),

(4, 'Marolio', 'Potosi 098', 2),

(5, 'Glaciar', 'Sarmiento 555', 3);

INSERT INTO CONTIENE (Nro, CodArt)

VALUES (001, 001),

(001, 002),

(001, 003),

(001, 004),

(002, 003),

(002, 004),

(003, 001),

(004, 002);

INSERT INTO COMPUESTO\_POR (CodArt, CodMat)

VALUES(001, 001),

(001, 002),

(001, 003),

(002, 002),

(002, 005),

(002, 007),

(003, 001),

(003, 002),

(003, 006),

(004, 007);

INSERT INTO PROVISTO\_POR(CodMat, CodProv)

VALUES (001, 1),

(002, 3),

(003, 5),

(004, 4),

(005, 2),

(006, 2),

(007, 5);

1) Listar nombre de todos los proveedores y de su ciudad

select p.nombre proveedor, c.nombre ciudad

from proveedor p

join ciudad c on p.codciudad=c.codciudad;

2)Listar los nombres de los proveedores de la ciudad de La Plata

select p.nombre

from proveedor p

join ciudad c on p.codciudad=c.codciudad

where c.nombre='La Plata';

3) Listar los números de almacenes que almacenan el

artículo de descripción que empiece con P

select distinct c.nro numero\_de\_almacen

from contiene c

join articulo a on c.codArt = a.codArt

where a.descripcion like 'P%'

order by c.nro;

4) Listar los números de almacenes y su responsable que

almacenan el artículo de descripción que empiece con P

select distinct a.nro numero\_almacen, a.responsable

from almacen a

join contiene c on a.nro=c.nro

join articulo ar on c.codart=ar.codart

where ar.descripcion like 'P%';

5) Listar los materiales (código y descripción)

provistos por proveedores de la ciudad de Ramos Mejia

select m.codmat, m.descripcion

from material m

join provisto\_por pp on m.codmat=pp.codmat

join proveedor p on pp.codprov=p.codprov

join ciudad c on c.codciudad=p.codciudad

where c.nombre='Ramos Mejia';

6) Listar los nombres de los proveedores que proveen materiales

para artículos ubicados en almacenes que Roberto tiene a su cargo

select pr.nombre

from proveedor pr

join provisto\_por pp on pr.codprov=pp.codprov

join material m on pp.codmat=m.codmat

join compuesto\_por c on m.codmat=c.codmat

join articulo a on c.codart=a.codart

join contiene co on a.codart=co.codart

join almacen al on co.nro=al.nro

where al.responsable='Roberto';