**EJERCICIO 1**

Dada la siguiente base de datos:

Almacén (**Nro**, Responsable)

Artículo (**CodArt**, descripción, Precio)

Material (**CodMat**, Descripción)

Proveedor (**CodProv**, Nombre, Domicilio, Ciudad)

Tiene (**Nro, CodArt**)

Compuesto\_por (**CodArt, CodMat**)

Provisto\_por (**CodMat, CodProv**)

Creacion de tablas:

CREATE TABLE Almacen

(

Nro int primary key,

Responsable varchar(50)

)

CREATE TABLE Articulo

(

CodArt int primary key,

Descripcion varchar(50),

Precio decimal(12, 3)

)

CREATE TABLE Material

(

CodMat int primary key,

Descripcion varchar(50)

)

CREATE TABLE Proveedor

(

CodProv int primary key,

Nombre varchar(50),

Domicilio varchar(50),

Ciudad varchar(50)

)

CREATE TABLE Tiene

(

Nro int,

CodArt int

)

CREATE TABLE Compuesto\_Por

(

CodArt int,

CodMat int

)

CREATE TABLE Provisto\_Por

(

CodMat int,

CodProv int

)

Datos:

INSERT INTO Almacen VALUES

(1, 'Juan Perez'),

(2, 'Jose Basualdo'),

(3, 'Rogelio Rodriguez')

INSERT INTO Articulo VALUES

(1, 'Sandwich JyQ', 5),

(2, 'Pancho', 6),

(3, 'Hamburguesa', 10),

(4, 'Hamburguesa completa', 15)

INSERT INTO Material VALUES

(1, 'Pan'),

(2, 'Jamon'),

(3, 'Queso'),

(4, 'Salchicha'),

(5, 'Pan Pancho'),

(6, 'Paty'),

(7, 'Lechuga'),

(8, 'Tomate')

INSERT INTO Proveedor VALUES

(1, 'Panadería Carlitos', 'Carlos Calvo 1212', 'CABA'),

(2, 'Fiambres Perez', 'San Martin 121', 'Pergamino'),

(3, 'Almacen San Pedrito', 'San Pedrito 1244', 'CABA'),

(4, 'Carnicería Boedo', 'Av. Boedo 3232', 'CABA'),

(5, 'Verdulería Platense', '5 3232', 'La Plata')

INSERT INTO Tiene VALUES

--Juan Perez

(1, 1),

--Jose Basualdo

(2, 1),(2, 2),(2, 3),(2, 4),

--Rogelio Rodriguez

(3, 3),(3, 4)

INSERT INTO Compuesto\_Por VALUES

--Sandwich JyQ

(1, 1), (1, 2), (1, 3),

--Pancho

(2, 4), (2, 5),

--Hamburguesa

(3, 1), (3, 6),

--Hamburguesa completa

(4, 1), (4, 6), (4, 7), (4, 8)

INSERT INTO Provisto\_Por VALUES

--Pan

(1, 1), (1, 3),

--Jamon

(2, 2), (2, 3), (2, 4),

--Queso

(3, 2), (3, 3),

--Salchicha

(4, 3), (4, 4),

--Pan Pancho

(5, 1), (5, 3),

--Paty

(6, 3), (6, 4),

--Lechuga

(7, 3), (7, 5),

--Tomate

(8, 3), (8, 5)

1. Listar los nombres de los proveedores de la ciudad de La Plata.

SELECT Nombre

FROM Proveedor

WHERE ciudad = 'La Plata'

2. Listar los números de artículos cuyo precio sea inferior a $10.

SELECT CodArt

FROM Articulo

WHERE precio < 10

3. Listar los responsables de los almacenes.

SELECT Responsable

FROM Almacen

4. Listar los códigos de los materiales que provea el proveedor ~~10~~ 3 y no los provea el proveedor ~~15~~ 5.

SELECT CodMat

FROM Provisto\_Por

WHERE CodProv = 3

EXCEPT

SELECT CodMat

FROM Provisto\_Por

WHERE CodProv = 5

-- Otra forma de resolverlo

SELECT CodMat

FROM Provisto\_por

WHERE CodProv = 3

AND CodMat NOT IN ( SELECT CodMat

FROM Provisto\_por

WHERE CodProv = 5)

5. Listar los números de almacenes que almacenan el artículo ~~A~~ Pancho.

SELECT \*

FROM Tiene

WHERE CodArt = ( SELECT CodArt

FROM Articulo

WHERE descripcion = 'Pancho' )

-- Otra forma de resolverlo

SELECT t.\*

FROM Tiene t, Articulo a

WHERE t.CodArt = a.CodArt

AND a.descripcion = 'Pancho'

6. Listar los proveedores de Pergamino que se llamen Pérez.

7. Listar los almacenes que contienen los artículos Pancho y los artículos Hamburguesa completa (ambos).

SELECT t.Nro

FROM Tiene t, Articulo a

WHERE t.CodArt = a.CodArt

AND a.descripcion = 'Pancho'

INTERSECT

SELECT t.Nro

FROM Tiene t, Articulo a

WHERE t.CodArt = a.CodArt

AND a.descripcion = 'Hamburguesa completa'

8. Listar los artículos que cuesten más de 8 o que estén compuestos por el

Material Lechuga.

SELECT CodArt

FROM Articulo

WHERE Precio > 8

UNION

SELECT c.CodArt

FROM Compuesto\_por c, Material m

WHERE c.CodMat = m.CodMat

AND m.Descripcion = 'Lechuga'

9. Listar los materiales, código y descripción, provistos por proveedores de la ciudad

de Rosario.

10. Listar el código, descripción y precio de los artículos que se almacenan en A1.

11. Listar la descripción de los materiales que componen el artículo B.

12. Listar los nombres de los proveedores que proveen los materiales al almacén que

Martín Gómez tiene a su cargo.

SELECT a.Nro, q.Nombre

FROM Almacen a, Tiene t, Compuesto\_Por c, Provisto\_Por p, Proveedor q

WHERE a.Responsable = 'Jose Basualdo'

AND a.Nro = t.Nro

AND t.CodArt = c.CodArt

AND c.CodMat = p.CodMat

AND p.CodProv = q.CodProv

SELECT a.Nro, q.Nombre

FROM Almacen a INNER JOIN Tiene t ON (a.Nro = t.Nro)

INNER JOIN Compuesto\_Por c ON (t.CodArt = c.CodArt)

INNER JOIN Provisto\_Por p ON (c.CodMat = p.CodMat)

INNER JOIN Proveedor q ON (p.CodProv = q.CodProv)

WHERE a.Responsable = 'Jose Basualdo'

13. Listar códigos y descripciones de los artículos compuestos por al menos un

material provisto por el proveedor López.

14. Hallar los códigos y nombres de los proveedores que proveen al menos un

material que se usa en algún artículo cuyo precio es mayor a $100.

15. Listar los números de almacenes que tienen todos los artículos que incluyen el

material con código 6.

Almacenes tales que NO EXISTE un Articulo compuesto por el Material 6 que NO tengan

SELECT \*

FROM Almacen a

WHERE not exists ( SELECT \*

FROM Compuesto\_por c

WHERE CodMat = 6

AND NOT EXISTS ( SELECT \*

FROM Tiene t

WHERE t.CodArt = c.CodArt

AND t.Nro = a.Nro ))

COCIENTE

Listar los Alumnos que aprobaron TODAS las Materias

Traduccion:

Listar los Alumnos tales que NO EXISTE una Materia que NO hayan aprobado

Listar los Medicos que trabajan en TODOS los Hospitales

Traduccion:

Listar los Medicos tales que NO EXISTE un Hospital en el que NO trabaje

Listar los Clientes que compraron TODOS los Productos

Traduccion:

Listar los Clientes tales que NO EXISTE un Producto que NO haya comprado

16. Listar los proveedores de Capital Federal que sean únicos proveedores de algún

material.

SELECT \*

FROM Proveedor p, Provisto\_por x

WHERE p.Ciudad = 'CABA'

AND p.CodProv = x.CodProv

AND NOT EXISTS ( SELECT \*

FROM Provisto\_Por x2, Proveedor p2

WHERE x2.CodMat = x.CodMat

AND x2.CodProv <> x.CodProv

AND x2.CodProv = p2.CodProv

AND p2.Ciudad = 'CABA')

17. Listar el/los artículo/s de mayor precio.

SELECT \*

FROM Articulo

WHERE precio = ( SELECT MAX(Precio)

FROM Articulo )

SELECT \*

FROM Articulo

WHERE precio >= ALL ( SELECT Precio

FROM Articulo )

18. Listar el/los artículo/s de menor precio.

19. Listar el promedio de precios de los artículos en cada almacén.

SELECT t.Nro, AVG(a.Precio) PRECIO\_PROMEDIO

FROM Tiene t, Articulo a

WHERE t.CodArt = a.CodArt

GROUP BY t.Nro

20. Listar los almacenes que almacenan la mayor cantidad de artículos.

CREATE VIEW Cant\_Articulos AS

SELECT Nro, count(\*) CANT\_ART

FROM Tiene

GROUP BY Nro

SELECT \*

FROM Cant\_Articulos

WHERE CANT\_ART = ( SELECT MAX(CANT\_ART)

FROM Cant\_Articulos )

21. Listar los artículos compuestos por al menos 2 materiales.

22. Listar los artículos compuestos por exactamente 2 materiales.

SELECT CodArt, count(\*) CANT\_MATERIALES

FROM Compuesto\_Por

GROUP BY CodArt

HAVING count(\*) = 2

23. Listar los artículos que estén compuestos con hasta 2 materiales.

24. Listar los artículos compuestos por todos los materiales.

25. Listar las ciudades donde existan proveedores que provean todos los materiales.

**EJERCICIO 7**

Dada la siguiente base de datos:

Auto (**matrícula**, modelo, año)

Chofer(**nroLicencia**, nombre, apellido, fecha\_ingreso, teléfono)

Viaje(F**echaHoraInicio**, FechaHoraFin, **chofer**, cliente, auto,

kmTotales, esperaTotal, costoEspera, costoKms )

Cliente (**nroCliente**, calle, nro, localidad )

Creacion de tablas:

CREATE TABLE Auto

(

matricula char(7) PRIMARY KEY,

modelo varchar(20),

anio integer

)

CREATE TABLE Chofer

(

NroLicencia integer PRIMARY KEY,

nombre varchar(30),

apellido varchar(30),

fecha\_ingreso date,

telefono varchar(15),

)

CREATE TABLE Cliente

(

nroCliente integer PRIMARY KEY,

calle varchar(30),

nro integer,

localidad varchar(30),

)

CREATE TABLE Viaje

(

FechaHoraInicio datetime,

FechaHoraFin datetime,

chofer integer,

cliente integer,

auto integer,

kmTotales integer,

esperaTotal integer,

costoEspera integer,

costoKms integer,

)

Datos:

INSERT INTO Auto VALUES (1000,'208',2020)

INSERT INTO Auto VALUES (1001,'Golf',2018)

INSERT INTO Auto VALUES (1002,'Focus',2022)

INSERT INTO Chofer VALUES (10,'Juan','Perez', CONVERT(DATE,'20/04/2019',103), null)

INSERT INTO Chofer VALUES (11,'Jorge','Gonzalez', CONVERT(DATE,'15/10/2020',103), null)

INSERT INTO Chofer VALUES (12,'Mario','Tala', '2021-03-22', null)

INSERT INTO Cliente VALUES (100,'Rivadavia',123,'CABA')

INSERT INTO Cliente VALUES (101,'Medrano',456,'CABA')

INSERT INTO Cliente VALUES (102,'Quintana',520,'Castelar')

INSERT INTO Cliente VALUES (103,'Directorio',856,'Haedo')

INSERT INTO Viaje VALUES ('2020-01-18 14:30:00','2020-01-18 17:20:00',10,100,1000,230,55,1200,9900)

INSERT INTO Viaje VALUES ('2020-01-22 09:15:00','2020-01-22 11:40:00',10,101,1001,88,0,0,7500)

INSERT INTO Viaje VALUES ('2020-01-23 09:15:00','2020-01-23 11:40:00',10,101,1002,88,0,0,7500)

INSERT INTO Viaje VALUES ('2020-01-24 09:15:00','2020-01-24 11:40:00',10,101,1003,88,0,0,7500)

INSERT INTO Viaje VALUES ('2023-05-18 09:15:00','2023-05-18 11:40:00',10,102,1001,30,0,0,6500)

INSERT INTO Viaje VALUES ('2023-05-18 19:00:00','2023-05-18 19:40:00',10,101,1000,30,0,0,4000)

INSERT INTO Viaje VALUES ('2023-05-19 09:15:00','2020-01-22 11:40:00',11,102,1002,20,0,0,3500)

INSERT INTO Viaje VALUES ('2023-05-20 09:15:00','2020-01-22 11:40:00',12,103,1002,40,0,0,4000)

1. Indique cuales son los autos con mayor cantidad de kilómetros realizados en el

último mes.

CREATE VIEW Vista1 AS

SELECT auto, SUM(kmTotales) SUMA\_KM

FROM Viaje

WHERE FechaHoraInicio >= getdate()-30

--WHERE FechaHoraInicio BETWEEN '2023-05-01' AND '2023-06-01'

--WHERE FechaHoraInicio >= '2023-05-01'

--WHERE year(FechaHoraInicio) = 2023 AND month(FechaHoraInicio) = 5

GROUP BY auto

SELECT \*

FROM Vista1

WHERE suma\_km = ( SELECT MAX(suma\_km)

FROM Vista1 )

2. Indique los clientes que más viajes hayan realizado con el mismo chofer.

CREATE VIEW Vista2 AS

SELECT cliente, chofer, count(\*) CANT\_VIAJES

FROM Viaje

GROUP BY cliente, chofer

SELECT \*

FROM Vista2

WHERE cant\_viajes = ( SELECT MAX(cant\_viajes)

FROM Vista2 )

3. Indique el o los clientes con mayor cantidad de viajes en este año.

CREATE VIEW Vista3 AS

SELECT Cliente, count(\*) CANT\_VIAJES

FROM Viaje

WHERE year(FechaHoraInicio) = 2023

GROUP BY Cliente

SELECT \*

FROM Vista3

WHERE cant\_viajes = ( SELECT MAX(cant\_viajes)

FROM Vista3 )

4. Obtenga nombre y apellido de los choferes que no manejaron todos los vehículos

que disponemos.

--Choferes tales que Existe un Auto que no manejaron

SELECT \*

FROM Chofer c

WHERE EXISTS ( SELECT \*

FROM Auto a

WHERE NOT EXISTS ( SELECT \*

FROM Viaje v

WHERE v.chofer = c.NroLicencia

AND v.auto = a.matricula ))

--Otra forma

SELECT a.chofer, count(\*)

FROM

( SELECT distinct chofer, auto

FROM Viaje

) a

GROUP BY a.chofer

HAVING count(\*) = ( SELECT count(\*)

FROM Auto )

5. Obtenga el nombre y apellido de los clientes que hayan viajado en todos nuestros

autos.

--Clientes tales que No Existe un Auto en el que no Viajaron

SELECT \*

FROM Cliente c

WHERE NOT EXISTS ( SELECT \*

FROM Auto a

WHERE NOT EXISTS ( SELECT \*

FROM Viaje v

WHERE v.cliente = c.NroCliente

AND v.auto = a.matricula ))

6. Queremos conocer el tiempo de espera promedio de los viajes de los últimos 2

meses

7. Indique los kilómetros realizados en viajes por cada auto.

8. Indique el costo promedio de los viajes realizados por cada auto.

9. Indique el costo total de los viajes realizados por cada chofer en el último mes.

10. Indique la fecha inicial, el chofer y el cliente que hayan realizado el viaje más largo

de este año.

**EJERCICIO 9**

Dada la siguiente base de datos:

Persona (**TipoDoc, NroDoc**, Nombre, Direccion, FechaNac, Sexo)

Progenitor (**TipoDoc, NroDoc, TipoDocHijo, NroDocHijo**)

1. Hallar para una persona dada, por ejemplo José Pérez, los tipos y números de

documentos, nombre, dirección y fecha de nacimiento de todos sus hijos.

2. Hallar para cada persona los tipos y números de documento, nombre, domicilio y

fecha de nacimiento de:

a. Todos sus hermanos, incluyendo medios hermanos.

b. Su madre

c. Su abuelo materno

d. Todos sus nietos

Creacion de Tablas:

CREATE TABLE Persona

(

DNI numeric(8,0) PRIMARY KEY,

Nombre varchar(30),

Direccion varchar(80),

FechaNac date,

Sexo char(1),

)

CREATE TABLE Progenitor

(

DNI numeric(8,0),

DNI\_Hijo numeric(8,0),

PRIMARY KEY (DNI, DNI\_Hijo),

FOREIGN KEY (DNI) REFERENCES Persona(DNI),

FOREIGN KEY (DNI\_Hijo) REFERENCES Persona(DNI),

)

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (10100100,'Hector','M');

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (10100200,'Maria','F');

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (10200100,'Francisco','M');

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (10200200,'Juana','F');

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (20100100,'Jorge','M');

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (20100200,'Claudio','M');

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (20100300,'Susana','F');

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (20200100,'Oscar','M');

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (20200200,'Alicia','F');

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (30100100,'Diego','M');

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (30100200,'Natalia','F');

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (30200100,'Pablo','M');

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (30200200,'Laura','F');

INSERT INTO Persona (DNI,NOMBRE,SEXO) VALUES (30200300,'Sabrina','F');

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (10100100,20100100);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (10100100,20100200);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (10100100,20100300);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (10100200,20100100);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (10100200,20100200);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (10100200,20100300);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (10200100,20200100);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (10200100,20200200);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (10200200,20200100);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (10200200,20200200);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (20100100,30100100);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (20100100,30100200);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (20200200,30100100);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (20200200,30100200);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (20100300,30200100);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (20100300,30200200);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (20100300,30200300);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (20200100,30200100);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (20200100,30200200);

INSERT INTO Progenitor (DNI,DNI\_HIJO) VALUES (20200100,30200300);

Arbol Genealógico

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hector | Maria |  | Francisco | Juana |
| Jorge |  |  | Oscar |  |
| Claudio |  |  | Alicia |  |
| Susana |  |  |  |  |
|  |  |  |  |  |
|  | Jorge | Alicia |  |  |
|  | Diego |  |  |  |
|  | Natalia |  |  |  |
|  |  |  |  |  |
|  | Susana | Oscar |  |  |
|  | Pablo |  |  |  |
|  | Laura |  |  |  |
|  | Sabrina |  |  |  |

Soluciones:

1. Hallar para una persona dada, por ejemplo José Pérez, los tipos y números de

documentos, nombre, dirección y fecha de nacimiento de todos sus hijos.

SELECT per.dni, per.nombre,

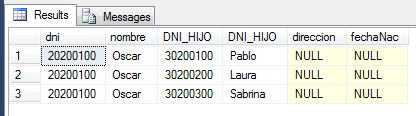
hij.dni DNI\_HIJO, hij.nombre DNI\_HIJO, hij.direccion, hij.fechaNac

FROM Persona per, Progenitor pro, Persona hij

WHERE per.dni=pro.dni

AND pro.dni\_hijo=hij.dni

AND per.nombre='Oscar'



2. Hallar para cada persona:

a. Todos sus hermanos, incluyendo medios hermanos.

SELECT DISTINCT per.dni, per.nombre,

her.dni, her.nombre

FROM Persona per, Progenitor pro1, Progenitor pro2, Persona her

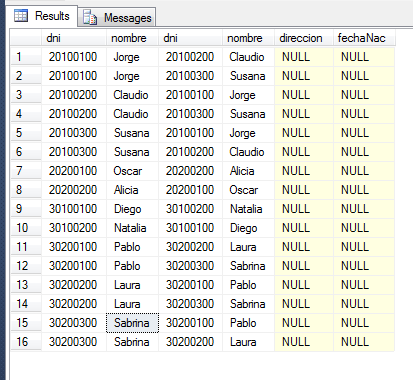
WHERE per.dni=pro1.dni\_hijo

AND pro1.dni=pro2.dni

AND pro2.dni\_hijo=her.dni

AND per.dni<>her.dni

ORDER BY 1,3



b. Su madre

SELECT per.dni, per.nombre,

mad.dni DNI\_MADRE, mad.nombre NOMBRE\_MADRE

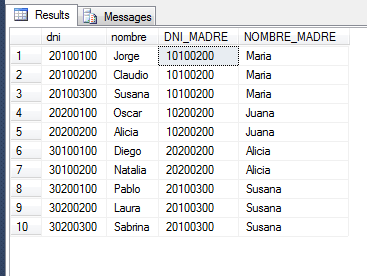
FROM Persona per, Progenitor pro, Persona mad

WHERE per.dni=pro.dni\_hijo

AND pro.dni=mad.dni

AND mad.sexo='F'

ORDER BY 1



c. Su abuelo materno

SELECT per.dni, per.nombre,

abu.dni DNI\_ABUELO, abu.nombre NOMBRE\_ABUELO

FROM Persona per, Progenitor pro1, Persona mad, Progenitor pro2, Persona abu

WHERE per.dni=pro1.dni\_hijo

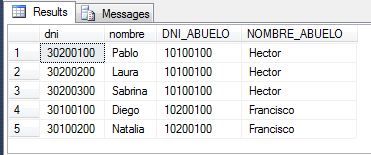
AND pro1.dni=mad.dni

AND mad.sexo='F'

AND mad.dni=pro2.dni\_hijo

AND pro2.dni=abu.dni

AND abu.sexo='M'



d. Todos sus nietos

SELECT per.dni, per.nombre,

nie.dni DNI\_NIETO, nie.nombre NOMBRE\_NIETO

FROM Persona per, Progenitor pro1, Progenitor pro2, Persona nie

WHERE per.dni=pro1.dni

AND pro1.dni\_hijo=pro2.dni

AND pro2.dni\_hijo=nie.dni

