

UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

DIVISIÓN DE INGENIERÍAS DEPARTAMENTO DE INFORMÁTICA



Seminario de solución de problemas de Bases de Datos

Lunes y Miercoles

Actividad 9

Actividad de Aprendizaje

ALUMNO: LOPEZ RODRIGUEZ CHRISTIAN ADRIAN

Código: 218022125

Ciclo escolar: 2023-A

Profesor: Murillo Leño Maria Magdalena. Sección: D01

PROBLEMA 1.-

1- Cree la tabla agenda con la siguiente estructura:(apellido varchar(30), nombre varchar(20) not null, domicilio varchar(30), telefono varchar(11), mail varchar(30),)

2- Ingrese los siguientes registros:

```
('Perez','Juan','Sarmiento 345','4334455','juancito@gmail.com'),
('Garcia','Ana','Urquiza 367','4226677','anamariagarcia@hotmail.com'),
('Lopez','Juan','Avellaneda 900',null,'juancitoLopez@gmail.com'),
('Juarez','Mariana','Sucre 123','0525657687','marianaJuarez2@gmail.com'),
('Molinari','Lucia','Peru 1254','4590987','molinarilucia@hotmail.com'),
('Ferreya','Patricia','Colon 1534','4585858',null),
('Perez','Susana','San Martin 333',null,null),
('Perez','Luis','Urquiza 444','0354545256','perezluisalberto@hotmail.com')
('Lopez','Maria','Salta 314',null,'lopezmariayo@gmail.com')
```

CREATE TABLE estructura

```
(
    apellido varchar(30),
    nombre varchar(20) not null,
    domicilio varchar(30),
    telefono varchar(11),
    mail varchar(30)
)
```

SELECT * FROM estructura

INSERT INTO estructura

VALUES

```
('Perez','Juan','Sarmiento 345','4334455','juancito@gmail.com'),
```

```

('Garcia','Ana','Urquiza 367','4226677','anamariagarcia@hotmail.com'),
('Lopez','Juan','Avellaneda 900',null,'juancitoLopez@gmail.com'),
('Juarez','Mariana','Sucre 123','0525657687','marianaJuarez2@gmail.com'),
('Molinari','Lucia','Peru 1254','4590987','molinarilucia@hotmail.com'),
('Ferreyra','Patricia','Colon 1534','4585858',null),
('Perez','Susana','San Martin 333',null,null),
('Perez','Luis','Urquiza 444','0354545256','perezluisalberto@hotmail.com'),
('Lopez','Maria','Salta 314',null,'lopezmariayo@gmail.com')

```

The screenshot shows a PostgreSQL database interface. On the left, the 'Schemas' tree is expanded to 'public', and 'Tables (12)' is selected. The main query editor displays the following SQL code:

```

IS_TEMPLATE = False;

CREATE TABLE estructura
(
    apellido varchar(30),
    nombre varchar(20) not null,
    domicilio varchar(30),
    telefono varchar(11),
    mail varchar(30)
)

SELECT * FROM estructura

INSERT INTO estructura
VALUES
('Perez','Juan','Sarmiento 345','4334455','juancito@gmail.com'),
('Garcia','Ana','Urquiza 367','4226677','anamariagarcia@hotmail.com'),
('Lopez','Juan','Avellaneda 900',null,'juancitoLopez@gmail.com'),
('Juarez','Mariana','Sucre 123','0525657687','marianaJuarez2@gmail.com'),
('Molinari','Lucia','Peru 1254','4590987','molinarilucia@hotmail.com'),
('Ferreyra','Patricia','Colon 1534','4585858',null),
('Perez','Susana','San Martin 333',null,null),
('Perez','Luis','Urquiza 444','0354545256','perezluisalberto@hotmail.com'),
('Lopez','Maria','Salta 314',null,'lopezmariayo@gmail.com')

```

Below the query editor, the 'Data Output' tab shows the results of the query. The table has 4 columns: 'apellido', 'nombre', 'domicilio', and 'telefono'. The results are as follows:

	apellido character varying (30)	nombre character varying (20)	domicilio character varying (30)	telefono character varying (11)
1	Perez	Juan	Sarmiento 345	4334455
2	Garcia	Ana	Urquiza 367	4226677
3	Lopez	Juan	Avellaneda 900	[null]
4	Juarez	Mariana	Sucre 123	0525657687
5	Molinari	Lucia	Peru 1254	4590987

At the bottom, it indicates 'Total rows: 9 of 9' and 'Query complete 00:00:00.880'.

IMAGEN 1.1

3- Cree un índice común por el campo apellido.

create index INDICE on estructura(apellido);

4- Cree un índice único por el mail.

create unique index INDICE_unico on

estructura(mail);

The screenshot shows the PostgreSQL Query Editor interface. On the left, the 'Schemas (1)' tree is expanded to 'public', and the 'Tables (12)' list is visible. The 'estructura' table is selected. The main query window contains the following SQL code:

```
17  apellido varchar(30),
18  nombre varchar(20) not null,
19  domicilio varchar(30),
20  telefono varchar(11),
21  mail varchar(30)
22  )
23
24  SELECT * FROM estructura
25
26  INSERT INTO estructura
27  VALUES
28  ('Perez','Juan','Sarmiento 345','4334455','juancito@gmail.com'),
29  ('Garcia','Ana','Urquiza 367','4226677','anamariagarcia@hotmail.com'),
30  ('Lopez','Juan','Avellaneda 900',null,'juancitoLopez@gmail.com'),
31  ('Juarez','Mariana','Sucre 123','0525657687','marianaJuarez2@gmail.com'),
32  ('Molinari','Lucia','Peru 1254','4590987','molinariLucia@hotmail.com'),
33  ('Ferreyra','Patricia','Colon 1534','4585858',null),
34  ('Perez','Susana','San Martin 333',null,null),
35  ('Perez','Luis','Urquiza 444','0354545256','perezluisalberto@hotmail.com'),
36  ('Lopez','Maria','Salta 314',null,'lopezmarfayo@gmail.com')
37
38
39  create index INDICE on estructura(apellido);
40
41  create unique index INDICE_unico on
42  estructura(mail);
43
```

The 'Messages' tab at the bottom shows the output: 'CREATE INDEX' and 'Query returned successfully in 265 msec.' A green status bar at the bottom right indicates 'Query returned successfully in 265 msec.'

5- Borre los dos índices.

drop index INDICE;

drop index INDICE_unico;

The screenshot shows the PostgreSQL Query Editor interface. The 'Schemas (1)' tree is expanded to 'public', and the 'Tables (12)' list is visible. The 'estructura' table is selected. The main query window contains the following SQL code:

```
22  )
23
24  SELECT * FROM estructura
25
26  INSERT INTO estructura
27  VALUES
28  ('Perez','Juan','Sarmiento 345','4334455','juancito@gmail.com'),
29  ('Garcia','Ana','Urquiza 367','4226677','anamariagarcia@hotmail.com'),
30  ('Lopez','Juan','Avellaneda 900',null,'juancitoLopez@gmail.com'),
31  ('Juarez','Mariana','Sucre 123','0525657687','marianaJuarez2@gmail.com'),
32  ('Molinari','Lucia','Peru 1254','4590987','molinariLucia@hotmail.com'),
33  ('Ferreyra','Patricia','Colon 1534','4585858',null),
34  ('Perez','Susana','San Martin 333',null,null),
35  ('Perez','Luis','Urquiza 444','0354545256','perezluisalberto@hotmail.com'),
36  ('Lopez','Maria','Salta 314',null,'lopezmarfayo@gmail.com')
37
38
39  create index INDICE on estructura(apellido);
40
41  create unique index INDICE_unico on
42  estructura(mail);
43
44  drop index INDICE;
45  drop index INDICE_unico;
46
47
48
49
```

The 'Messages' tab at the bottom shows the output: 'drop INDEX' and 'Query returned successfully in 241 msec.' A green status bar at the bottom right indicates 'Query returned successfully in 241 msec.'

PROBLEMA 2:

Una empresa de remises tiene registrada la información de sus vehículos en una tabla llamada "remis".

1- Cree la tabla con la siguiente estructura: remis(numero serial, patente char(6),
marca varchar(15), modelo char(4))

2- Ingrese algunos registros sin repetir patente:

```
insert into remis (patente,marca,modelo)values('ABC123','Renault 12','1990');
```

```
insert into remis (patente,marca,modelo)values('DEF456','Fiat Duna','1995');
```

CREATE TABLE remis

```
(  
    numero serial,  
    patente char(6),  
    marca varchar(15),  
    modelo char(4)  
)
```

SELECT * FROM remis

The screenshot shows a PostgreSQL database interface. On the left, the 'Schemas (1)' tree is expanded to show the 'public' schema, with 'Tables (13)' listed. The main query editor displays the following SQL code:

```
41 create unique index INDICE_unico on  
42 estructura(ma1);  
43  
44 drop index INDICE;  
45 drop index INDICE_unico;  
46  
47  
48 drop table remis;  
49  
50 CREATE TABLE remis  
51 (  
52     numero serial,  
53     patente char(6),  
54     marca varchar(15),  
55     modelo char(4)  
56 )  
57  
58 SELECT * FROM remis  
59  
60 insert into remis (patente,marca,modelo)values('ABC123','Renault 12','1990');  
61  
62 insert into remis (patente,marca,modelo)values('DEF456','Fiat Duna','1995');
```

Below the query editor, the 'Data Output' tab shows the results of the SELECT query:

numero	patente	marca	modelo
1	ABC123	Renault 12	1990
2	DEF456	Fiat Duna	1995

The status bar at the bottom indicates 'Total rows: 2 of 2' and 'Query complete 00:00:00.085'.

```
insert into remis (patente,marca,modelo)values('ABC123','Renault 12','1990');
```

```
insert into remis (patente,marca,modelo)values('DEF456','Fiat Duna','1995');
```

3- Definir una restricción "primary key" para el campo "patente".

```
alter table remis
```

```
add constraint PK_patente
```

```
primary key(patente);
```

4- Establezca una restricción "primary key" para el campo "numero". Y anote lo que sucede:

No pasa nada, este manda error porque ya existe una Primary key para esta tabla.

5- Vea la información de las restricciones

PROBLEMA A RESOLVER:

Una empresa tiene registrados sus clientes en una tabla llamada "clientes", también tiene una tabla "provincias" donde registra los nombres de las provincias.

1- Cree las tablas "clientes" y "provincias":

```
clientes ( codigo serial, nombre varchar(30), domicilio varchar(30), ciudad  
varchar(20), codigoprovincia smallint, primary key(codigo) )
```

```
provincias( codigo serial, nombre varchar(20), primary key(codigo) )
```

2- Ingrese algunos registros para ambas tablas:

Provincias:

```
(1,'Cordoba');
```

```
(2,'Santa Fe');
```

```
(3,'Misiones');
```

```
(4,'Rio Negro');
```

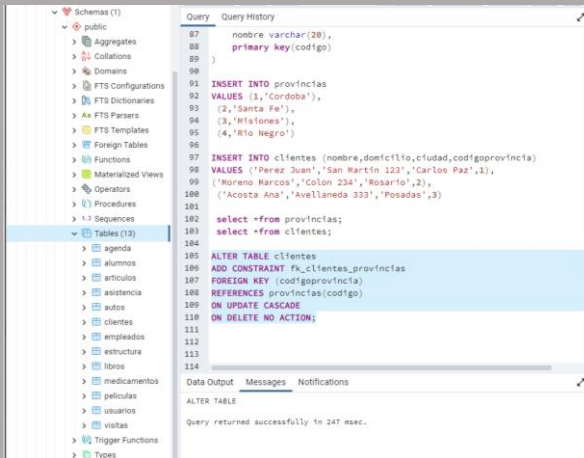
Clientes:

```
('Perez Juan','San Martin 123','Carlos Paz',1)
```

```
('Moreno Marcos','Colon 234','Rosario',2)
```

```
('Acosta Ana','Avellaneda 333','Posadas',3)
```

3- Establezca una restricción "foreign key" especificando la acción "en cascade" para actualizaciones y "no_action" para eliminaciones.



The screenshot shows a database management tool interface. On the left, a tree view displays the database schema, including tables like 'provincias' and 'clientes'. The main window shows a SQL query with the following content:

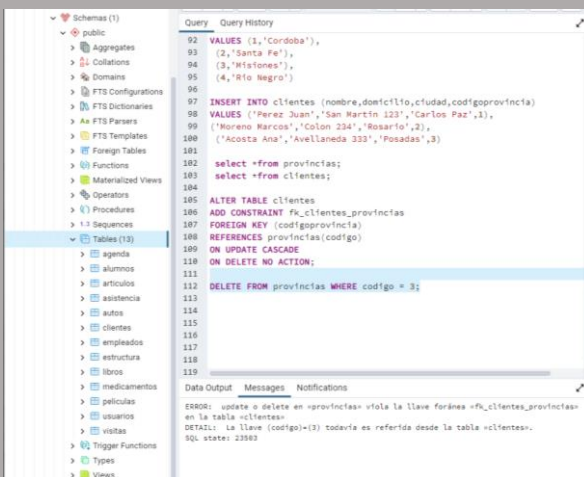
```
87 nombre varchar(20),
88 primary key(codigo)
89 )
90
91 INSERT INTO provincias
92 VALUES (1,'Cordoba'),
93 (2,'Santa Fe'),
94 (3,'Misiones'),
95 (4,'Rio Negro')
96
97 INSERT INTO clientes (nombre,domicilio,ciudad,codigoprovincia)
98 VALUES ('Perez Juan','San Martin 123','Carlos Paz',1),
99 ('Moreno Marcos','Colon 234','Rosario',2),
100 ('Acosta Ana','Avellaneda 333','Posadas',3)
101
102 select *from provincias;
103 select *from clientes;
104
105 ALTER TABLE clientes
106 ADD CONSTRAINT fk_clientes_provincias
107 FOREIGN KEY (codigoprovincia)
108 REFERENCES provincias(codigo)
109 ON UPDATE CASCADE
110 ON DELETE NO ACTION;
111
112
113
114
```

Below the query, the 'Data Output' tab shows the message: 'Query returned successfully in 247 msec.'

ALTER TABLE clientes
ADD CONSTRAINT fk_clientes_provincias
FOREIGN KEY (codigoprovincia)
REFERENCES provincias(codigo)
ON UPDATE CASCADE
ON DELETE NO ACTION;

4- Intente eliminar el registro con código 3, de "provincias".

No se puede porque hay registros en "clientes" al cual hace referencia y la opción para eliminaciones se estableció como "no action".



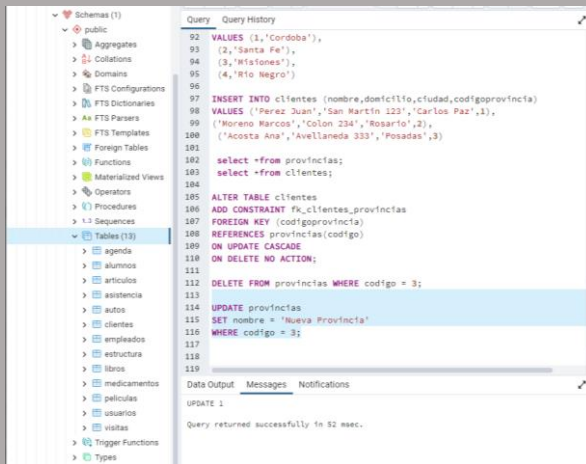
The screenshot shows the same database management tool interface. The SQL query window now contains the following content:

```
92 VALUES (1,'Cordoba'),
93 (2,'Santa Fe'),
94 (3,'Misiones'),
95 (4,'Rio Negro')
96
97 INSERT INTO clientes (nombre,domicilio,ciudad,codigoprovincia)
98 VALUES ('Perez Juan','San Martin 123','Carlos Paz',1),
99 ('Moreno Marcos','Colon 234','Rosario',2),
100 ('Acosta Ana','Avellaneda 333','Posadas',3)
101
102 select *from provincias;
103 select *from clientes;
104
105 ALTER TABLE clientes
106 ADD CONSTRAINT fk_clientes_provincias
107 FOREIGN KEY (codigoprovincia)
108 REFERENCES provincias(codigo)
109 ON UPDATE CASCADE
110 ON DELETE NO ACTION;
111
112 DELETE FROM provincias WHERE codigo = 3;
113
114
115
116
117
118
119
```

Below the query, the 'Data Output' tab shows an error message:

```
ERROR: update «delete en «provincias» viola la llave foránea «fk_clientes_provincias» en la tabla «clientes»
DETAIL: la llave (codigo)=(3) todavía es referida desde la tabla «clientes».
SQL state: 23503
```

5- Modifique el registro con código 3, de "provincias".



UPDATE provincias

SET nombre = 'Nueva Provincia'

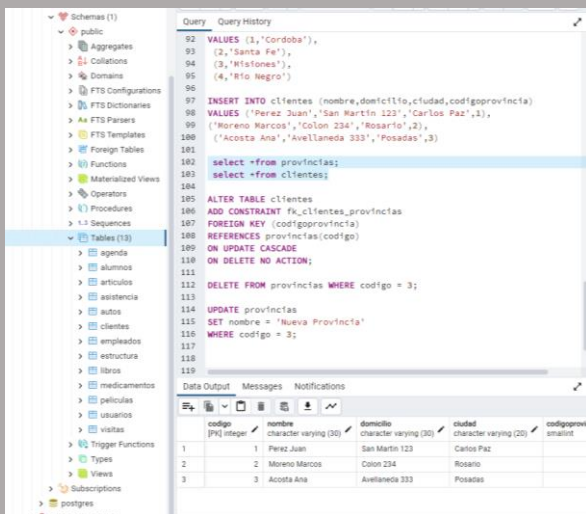
WHERE codigo = 3;

6- Verifique que el cambio se realizó en cascada, es decir, que se modificó en la tabla

"provincias" y en "clientes":

select *from provincias;

select *from clientes;



7- Intente modificar la restricción "foreign key" para que permita eliminación en cascada.

Mensaje de error, no se pueden modificar las restricciones.

8- Intente eliminar la tabla "provincias".

No se puede eliminar porque una restricción "foreign key" hace referencia a ella.