

**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 Miércoles

**Actividad 6**

**Actividad de Aprendizaje**

**ALUMNO: LOPEZ RODRIGUEZ CHRISTIAN ADRIAN**

**Código:** 218022125

**Ciclo escolar:** 2023-A

**Profesor: Murillo Leaño María Magdalena. Sección: D01**

**PROBLEMA 1.-**

Una empresa tiene registrados sus clientes en una tabla llamada "clientes".

1- Crée la tabla

clientes (código serial, nombre varchar(30) not null, domicilio varchar (30), ciudad varchar (20), provincia varchar (20), crédito decimal(9,2), primary key(código))

2- Ingrese 5 registros:

'Lopez Marcos','Colon 111','Cordoba','Cordoba',1900.56

'Perez Ana','San Martin 222','Cruz del Eje','Cordoba',450.33

'Garcia Juan','Rivadavia 333','Villa del Rosario','Cordoba',190

'Olmos Luis','Sarmiento 444','Rosario','Santa Fe',670.22

'Pereyra Lucas','San Martin 555','Cruz del Eje','Cordoba',500.55

2.- Muestre todos los registros.

The screenshot shows the pgAdmin 4 interface with the following details:

- Servers:** PostgreSQL 15
- Databases:** Actividad SSPBD
- Tables:** Actividad SSPBD
- Query Editor:** Contains the SQL code for creating the 'clientes' table and inserting five records. The table has columns: código (serial), nombre (varchar(30)), domicilio (varchar(30)), ciudad (varchar(20)), provincia (varchar(20)), and crédito (decimal(9,2)). Primary key is set to 'código'. The inserted data includes five rows with names like Lopez Marcos, Perez Ana, Garcia Juan, Olmos Luis, and Pereyra Lucas, along with their respective addresses, cities, provinces, and credit values.
- Data Output:** Shows the five inserted rows in a grid format.

```

CREATE TABLE clientes (
    código serial,
    nombre varchar(30) not null,
    domicilio varchar(30),
    ciudad varchar(20),
    provincia varchar(20),
    crédito decimal(9,2),
    primary key(código)
);

INSERT INTO clientes (nombre,domicilio,ciudad,provincia,crédito)
VALUES ('Lopez Marcos','Colon 111','Cordoba','Cordoba',1900.56),
('Perez Ana','San Martin 222','Cruz del Eje','Cordoba',450.33),
('Garcia Juan','Rivadavia 333','Villa del Rosario','Cordoba',190),
('Olmos Luis','Sarmiento 444','Rosario','Santa Fe',670.22),
('Pereyra Lucas','San Martin 555','Cruz del Eje','Cordoba',500.55);

SELECT * FROM clientes

```

código	nombre	domicilio	ciudad	provincia	crédito
1	Lopez Marcos	Colon 111	Cordoba	Cordoba	1900.56
2	Perez Ana	San Martin 222	Cruz del Eje	Cordoba	450.33
3	Garcia Juan	Rivadavia 333	Villa del Rosario	Cordoba	190.00
4	Olmos Luis	Sarmiento 444	Rosario	Santa Fe	670.22
5	Pereyra Lucas	San Martin 555	Cruz del Eje	Cordoba	500.55

#### 4- Mostrar el campo crédito redondeado hacia arriba y hacia abajo.

##### REDONDEADO HACIA ARRIBA

The screenshot shows the pgAdmin 4 interface with a query editor and a results grid. The query uses the `ceiling` function to round credit values upwards. The results show the rounded values for each client.

```
10    LC_CTYPE = 'Spanish_Mexico.1252'
11    TABLESPACE = pg_default
12    CONNECTION LIMIT = -1
13    IS_TEMPLATE = False;
14
15    CREATE TABLE clientes (
16        codigo serial,
17        nombre varchar(30) not null,
18        domicilio varchar(30),
19        ciudad character varying(20),
20        provincia character varying(20),
21        crédito decimal(9,2),
22        primary key(codigo)
23    );
24
25    INSERT INTO clientes (nombre,domicilio,ciudad,provincia,crédito)
26    VALUES ('Lopez Marcos','Colon 111','Cordoba','Cordoba',1900.50),
27    ('Perez Ana','San Martin 222','Cruz del Eje','Cordoba',450.33),
28    ('Garcia Juan','Rivadavia 333','Villa del Rosario','Cordoba',190),
29    ('Olmos Luis','Sarmiento 444','Rosario','Santa Fe',678.22),
30    ('Pereyra Lucas','San Martin 555','Cruz del Eje','Cordoba',500.55);
31
32    SELECT * FROM clientes
33
34    SELECT nombre,domicilio,ciudad,provincia,ceiling(crédito) FROM clientes
35
36
37
```

	nombre	domicilio	ciudad	provincia	ceiling
1	Lopez Marcos	Colon 111	Cordoba	Cordoba	1901
2	Perez Ana	San Martin 222	Cruz del Eje	Cordoba	451
3	Garcia Juan	Rivadavia 333	Villa del Rosario	Cordoba	190
4	Olmos Luis	Sarmiento 444	Rosario	Santa Fe	671
5	Pereyra Lucas	San Martin 555	Cruz del Eje	Cordoba	501

##### REDONDEADO HACIA ABAJO

The screenshot shows the pgAdmin 4 interface with a query editor and a results grid. The query uses the `floor` function to round credit values downwards. The results show the rounded values for each client.

```
10    LC_CTYPE = 'Spanish_Mexico.1252'
11    TABLESPACE = pg_default
12    CONNECTION LIMIT = -1
13    IS_TEMPLATE = False;
14
15    CREATE TABLE clientes (
16        codigo serial,
17        nombre varchar(30) not null,
18        domicilio varchar(30),
19        ciudad character varying(20),
20        provincia character varying(20),
21        crédito decimal(9,2),
22        primary key(codigo)
23    );
24
25    INSERT INTO clientes (nombre,domicilio,ciudad,provincia,crédito)
26    VALUES ('Lopez Marcos','Colon 111','Cordoba','Cordoba',1900.50),
27    ('Perez Ana','San Martin 222','Cruz del Eje','Cordoba',450.33),
28    ('Garcia Juan','Rivadavia 333','Villa del Rosario','Cordoba',190),
29    ('Olmos Luis','Sarmiento 444','Rosario','Santa Fe',678.22),
30    ('Pereyra Lucas','San Martin 555','Cruz del Eje','Cordoba',500.55);
31
32    SELECT * FROM clientes
33
34    SELECT nombre,domicilio,ciudad,provincia,floor(crédito) FROM clientes
35
36
37
```

	nombre	domicilio	ciudad	provincia	floor
1	Lopez Marcos	Colon 111	Cordoba	Cordoba	1900
2	Perez Ana	San Martin 222	Cruz del Eje	Cordoba	450
3	Garcia Juan	Rivadavia 333	Villa del Rosario	Cordoba	190
4	Olmos Luis	Sarmiento 444	Rosario	Santa Fe	670
5	Pereyra Lucas	San Martin 555	Cruz del Eje	Cordoba	500

## PROBLEMA 2.-

En una página web se guardan los siguientes datos de las visitas: número de visita, nombre, mail, país, fecha.

### 1- Crear la siguiente tabla:

visitas ( numero serial, nombre varchar(30) default 'Anonimo', mail varchar(50), pais varchar (20), fecha timestamp, primary key(numero))

### 2- Ingrese lo siguientes registros:

```
'Ana Maria Lopez','AnaMaria@hotmail.com','Argentina','2006-10-10 10:10',
'Gustavo Gonzalez','GustavoGGonzalez@hotmail.com','Chile','2006-10-10 21:30',
'Juancito','JuanJosePerez@hotmail.com','Argentina','2006-10-11 15:45',
'Fabiola Martinez','MartinezFabiola@hotmail.com','Mexico','2006-10-12 08:15',
'Fabiola Martinez','MartinezFabiola@hotmail.com','Mexico','2006-09-12 20:45',
'Juancito','JuanJosePerez@hotmail.com','Argentina','2006-09-12 16:20',
'Juancito','JuanJosePerez@hotmail.com','Argentina','2006-09-15 16:25'
```

### 2- Ordene los registros por fecha, en orden descendente.

The screenshot shows the pgAdmin 4 interface. On the left, the tree view displays the database structure under 'Actividad SSPBD'. The 'Tables' node is expanded, showing the 'visitas' table. The 'Data Output' tab at the bottom shows the results of a query that inserts seven rows of data into the 'visitas' table and then selects all columns from the table ordered by 'fecha' in descending order. The results table has columns: nombre, mail, pais, and fecha. The data is as follows:

	nombre	mail	pais	fecha
1	Fabiola Martinez	MartinezFabiola@hotmail.com	Mexico	2006-10-12 08:15:00
2	Juancito	JuanJosePerez@hotmail.com	Argentina	2006-10-11 15:45:00
3	Gustavo Gonzalez	GustavoGGonzalez@hotmail.com	Chile	2006-10-10 21:30:00
4	Ana Maria Lopez	AnaMaria@hotmail.com	Argentina	2006-09-15 16:25:00
5	Juancito	JuanJosePerez@hotmail.com	Argentina	2006-09-15 16:25:00
6	Fabiola Martinez	MartinezFabiola@hotmail.com	Mexico	2006-09-15 16:25:00

- 3- Muestre el nombre del usuario, pais y el número de mes, ordenado por país (ascendente) y número de mes (descendente)

```

CREATE TABLE vistitas(
    numero serial,
    nombre varchar(30) default 'Anonimo',
    mail varchar(50),
    pais varchar(20),
    fecha timestamp,
    primary key(numero)
);

INSERT INTO vistitas (nombre,mail,pais,fecha)
VALUES ('Ana Maria Lopez','Analoria@hotmail.com','Argentina','2006-10-10 10:10'),
('Gustavo Gonzalez','GustavoGonzalez@hotmail.com','Chile','2006-10-10 21:30'),
('Juanito','JuanJosePerez@hotmail.com','Argentina','2006-10-11 19:45'),
('Fabiola Martinez','MartinezFabiola@hotmail.com','Mexico','2006-10-12 08:15'),
('Fabiola Martinez','MartinezFabiola@hotmail.com','Mexico','2006-09-12 20:45'),
('Juanito','JuanJosePerez@hotmail.com','Argentina','2006-09-12 16:20'),
('Juanito','JuanJosePerez@hotmail.com','Argentina','2006-09-15 16:25');

SELECT * FROM vistitas
SELECT nombre,mail,pais, EXTRACT(MONTH FROM fecha) as mes FROM vistitas
ORDER BY pais ASC, mes DESC;

```

nombre	mail	pais	mes
Juanito	JuanJosePerez@hotmail.com	Argentina	10
Ana Maria Lopez	Analoria@hotmail.com	Argentina	10
Juanito	JuanJosePerez@hotmail.com	Argentina	9
Juanito	JuanJosePerez@hotmail.com	Argentina	9
Gustavo Gonzalez	GustavoGonzalez@hotmail.com	Chile	10
Fabiola Martinez	MartinezFabiola@hotmail.com	Mexico	10

- 4- Muestre el pais, el mes, el día y la hora y ordene las visitas por nombre del mes, del día y la hora.

```

SELECT nombre,mail,pais, EXTRACT(year FROM fecha) as anio,
EXTRACT(MONTH FROM fecha) as mes,
EXTRACT(day FROM fecha) as dia,
EXTRACT(hour FROM fecha) as hora
FROM vistitas
ORDER BY mes,dia,hora;

```

nombre	mail	pais	anio	mes	dia	hora
Juanito	JuanJosePerez@hotmail.com	Argentina	2006	9	12	16
Fabiola Martinez	MartinezFabiola@hotmail.com	Mexico	2006	9	12	20
Juanito	JuanJosePerez@hotmail.com	Argentina	2006	9	15	16
Ana Maria Lopez	Analoria@hotmail.com	Argentina	2006	10	10	10
Gustavo Gonzalez	GustavoGonzalez@hotmail.com	Chile	2006	10	10	21
Juanito	JuanJosePerez@hotmail.com	Argentina	2006	10	11	15

**SELECT nombre,mail,pais, EXTRACT(year FROM fecha) as anio,  
EXTRACT(MONTH FROM fecha) as mes,  
EXTRACT(day FROM fecha) as dia,  
EXTRACT(hour FROM fecha) as hora  
FROM visitas  
ORDER BY mes,dia,hora;**

6- Muestre los mail, país, ordenado por país, de todos los que visitaron la página en octubre (4 registros)

```

pgAdmin 4
File Object Tools Help
Servers (1)
  PostgreSQL 15
    Databases (2)
      Actividad SSPBD
        Activities (1)
          Casts
          Catalogs
          Event Triggers
          Extensions
          Foreign Data Wrappers
          Languages
          Publications
          Schemas (1)
            public
              Aggregates
              Collations
              Domains
              FTS Configurations
              FTS Dictionaries
              FTS Parsers
              FTS Templates
              Foreign Tables
              Functions
              Materialized Views
              Operators
              Procedures
              Sequences
              Tables
              Trigger Functions
              Types
              Views
            Subscriptions
            postgres
          Login/Group Roles
          Tablespaces
Dashboard Properties SQL Statistics Dependencies Dependents Processes Actividad SSPBD/postgres@PostgreSQL 15*
Query History
You are currently running version 6.15 of pgAdmin 4, however the current version is 6.20.
Please click here for more information.
Query Query History
47
48 INSERT INTO visitas (nombre,mail,pais,fecha)
49 VALUES ('Ana Maria Lopez','AnaMaria@hotmail.com','Argentina','2006-10-10 10:10'),
50 ('Gustavo Gonzalez','GustavoGonzalez@hotmail.com','Chile','2006-10-10 21:30'),
51 ('Juan Jose Perez','JuanJosePerez@hotmail.com','Argentina','2006-09-12 12:20'),
52 ('Fabiola Martinez','FabiolaMartinez@hotmail.com','Mexico','2006-09-12 06:15'),
53 ('Fabiola Martinez','FabiolaMartinez@hotmail.com','Mexico','2006-09-12 20:45'),
54 ('Juancito','JuanJosePerez@hotmail.com','Argentina','2006-09-12 16:28'),
55 ('Juancito','JuanJosePerez@hotmail.com','Argentina','2006-09-15 16:25');
56
57 SELECT * FROM visitas
58
59 SELECT nombre,mail,pais, EXTRACT(MONTH FROM fecha) as mes FROM visitas
60 ORDER BY pais ASC, mes DESC;
61
62 4- Muestre el país, el mes, el día y la hora y ordene las visitas por nombre del mes, del día y la hora.
63
64 SELECT nombre,mail,pais, EXTRACT(year FROM fecha) as año,
65 EXTRACT(MONTH FROM fecha) as mes
66 FROM visitas
67 ORDER BY pais;
68
69
70
71
72
73
74
Data Output Messages Notifications
Ln 67, Col 14
Vínculos 22°C Soleado 12:01 p. m. 20/03/2023

```

### PROBLEMA 3.-

Trabaje con la tabla "libros" de una librería.

1.-Crear tabla: libros( codigo serial, titulo varchar(40) not null, autor varchar(20) default 'Desconocido',

editorial varchar(20), precio decimal(6,2), primary key(codigo) )

2.-Ingrese siguientes registros:

'El aleph','Borges','Emece',15.90

'Antología poética','Borges','Planeta',39.50

'Java en 10 minutos','Mario Molina','Planeta',50.50

'Alicia en el pais de las maravillas','Lewis Carroll','Emece',19.90

'Martin Fierro','Jose Hernandez','Emece',25.90

'Martin Fierro','Jose Hernandez','Paidos',16.80

'Aprenda PHP','Mario Molina','Emece',19.50

'Cervantes y el quijote','Borges','Paidos',18.40

3.-Recupere los libros cuyo editorial sea igual a " Emece " y cuyo precio no supere los 20 pesos:

```

74     codigo serial,
75     titulo varchar(40) not null,
76     autor varchar(20) default 'Desconocido',
77     editorial varchar(20),
78     precio decimal(6,2),
79     primary key(codigo)
80   );
81
82   SELECT * FROM libros
83
84   INSERT INTO libros (titulo,autor,editorial,precio)
85   VALUES ('El aleph','Borges','Emece',15.99),
86   ('Antología poética','Borges','Planeta',39.50),
87   ('Java en 10 minutos','Mario Molina','Planeta',50.50),
88   ('Alicia en el país de las maravillas','Lewis Carroll','Emece',19.90),
89   ('Martin Fierro','José Hernández','Paidos',25.00),
90   ('Martin Fierro','José Hernández','Paidos',16.80),
91   ('Aprenda PHP','Mario Molina','Emece',19.50),
92   ('Cervantes y el Quijote','Borges','Paidos',18.40);
93
94 3.-Recupere los libros cuyo editorial sea igual a " Emece " y cuyo precio no supere los 20 pesos:
95
96   SELECT titulo,autor,editorial,precio FROM libros
97   WHERE (editorial = 'Emece') AND (precio < 20);
98
99
100
101

```

Total rows: 3 of 3    Query complete 00:00:00.118

Successfully run. Total query runtime: 118 msec. 3 rows affected.

4.-Seleccione los libros cuyo autor es "Jose Hernandez" y/o cuya editorial es " Emece ":

```

74     codigo serial,
75     titulo varchar(40) not null,
76     autor varchar(20) default 'Desconocido',
77     editorial varchar(20),
78     precio decimal(6,2),
79     primary key(codigo)
80   );
81
82   SELECT * FROM libros
83
84   INSERT INTO libros (titulo,autor,editorial,precio)
85   VALUES ('El aleph','Borges','Emece',15.99),
86   ('Antología poética','Borges','Planeta',39.50),
87   ('Java en 10 minutos','Mario Molina','Planeta',50.50),
88   ('Alicia en el país de las maravillas','Lewis Carroll','Emece',19.90),
89   ('Martin Fierro','José Hernández','Emece',25.00),
90   ('Martin Fierro','José Hernández','Paidos',16.80),
91   ('Aprenda PHP','Mario Molina','Emece',19.50),
92   ('Cervantes y el Quijote','Borges','Paidos',18.40);
93
94 3.-Recupere los libros cuyo editorial sea igual a " Emece " y cuyo precio no supere los 20 pesos:
95
96   SELECT titulo,autor,editorial,precio FROM libros
97   WHERE autor = 'Jose Hernandez';
98
99
100
101

```

Total rows: 2 of 2    Query complete 00:00:00.117

Successfully run. Total query runtime: 117 msec. 2 rows affected.

`SELECT titulo,autor,editorial,precio FROM libros`

`WHERE autor = 'Jose Hernandez'`

## 5.-Recupere los libros cuya editorial NO es "Planeta":

```

74     codigo serial,
75     titulo varchar(40) not null,
76     autor varchar(20) default 'Desconocido',
77     editorial varchar(20),
78     precio decimal(6,2),
79     primary key(codigo)
80   );
81
82   SELECT * FROM libros
83
84   INSERT INTO libros (titulo,autor,editorial,precio)
85   VALUES ('El aleph','Borges','Emece',15.99),
86   ('Antología poética','Borges','Planeta',39.50),
87   ('Java en 10 minutos','Mario Molina','Planeta',50.50),
88   ('Alice en el país de las maravillas','Lewis Carroll','Emece',19.90),
89   ('Martin Fierro','José Hernández','Paidos',25.90),
90   ('Martin Fierro','José Hernández','Paidos',16.80),
91   ('Aprende PHP','Mario Molina','Emece',19.50),
92   ('Cervantes y el Quijote','Borges','Paidos',18.40);
93
94 3.-Recupere los libros cuyo editorial sea igual a " Emece " y cuyo precio no supere los 20 pesos:
95
96   SELECT titulo,autor,editorial,precio FROM libros
97   WHERE editorial=>'Planeta'
98
99
100
101

```

Total rows: 6 of 6 Query complete 00:00:00.108

`SELECT titulo,autor,editorial,precio FROM libros`

`WHERE editorial <> 'Planeta'`

## 6.-Muestre el resultado de cada una de los select siguientes, e indique las diferencias:

`select * from libros`

`where (autor='Borges') or`

`(editorial='Paidos' and precio<20);`

```

81
82   SELECT * FROM libros
83
84   INSERT INTO libros (titulo,autor,editorial,precio)
85   VALUES ('El aleph','Borges','Emece',15.99),
86   ('Antología poética','Borges','Planeta',39.50),
87   ('Java en 10 minutos','Mario Molina','Planeta',50.50),
88   ('Alice en el país de las maravillas','Lewis Carroll','Emece',19.90),
89   ('Martin Fierro','José Hernández','Paidos',25.90),
90   ('Martin Fierro','José Hernández','Paidos',16.80),
91   ('Aprende PHP','Mario Molina','Emece',19.50),
92   ('Cervantes y el Quijote','Borges','Paidos',18.40);
93
94 3.-Recupere los libros cuyo editorial sea igual a " Emece " y cuyo precio no supere los 20 pesos:
95
96   SELECT titulo,autor,editorial,precio FROM libros
97   WHERE editorial=>'Planeta'
98
99
100
101

```

```

81
82   SELECT * FROM libros
83
84   INSERT INTO libros (titulo,autor,editorial,precio)
85   VALUES ('El aleph','Borges','Emece',15.99),
86   ('Antología poética','Borges','Planeta',39.50),
87   ('Java en 10 minutos','Mario Molina','Planeta',50.50),
88   ('Alice en el país de las maravillas','Lewis Carroll','Emece',19.90),
89   ('Martin Fierro','José Hernández','Paidos',25.90),
90   ('Martin Fierro','José Hernández','Paidos',16.80),
91   ('Aprende PHP','Mario Molina','Emece',19.50),
92   ('Cervantes y el Quijote','Borges','Paidos',18.40);
93
94 3.-Recupere los libros cuyo editorial sea igual a " Emece " y cuyo precio no supere los 20 pesos:
95
96   SELECT titulo,autor,editorial,precio FROM libros
97   WHERE editorial=>'Planeta'
98
99
100
101

```

Total rows: 4 of 4 Query complete 00:00:00.070

Successfully run. Total query runtime: 70 msec. 4 rows affected.

```
select * from libros
where (autor='Borges' or editorial='Paidos') and
(precio<20);
```

```

81 SELECT * FROM libros
82
83 INSERT INTO libros (titulo,autor,editorial,precio)
84     ('El Aleph','Borges','Emece',15.90),
85     ('Aventuras de un Loco','Borges','Planeta',19.50),
86     ('Cervantes y el Quijote','Borges','Planeta',18.40),
87     ('Alice en el pais de las maravillas','Lewis Carroll','Emece',19.90),
88     ('Martin Fierro','Jose Hernandez','Emece',25.90),
89     ('Martin Fierro','Jose Hernandez','Paidos',16.80),
90     ('Aprenda PHP','Mario Molina','Emece',19.50),
91     ('Cervantes y el Quijote','Borges','Paidos',18.40);
92
93 3.- Recupere los libros cuyo editorial sea igual a "Emece" y cuyo precio no supere los 20 pesos:
94
95 SELECT titulo,autor,editorial,precio FROM libros
96 WHERE editorial => 'Planeta'
97
98 select * from libros
99 where (autor='Borges') or
100 (editorial='Paidos' and precio<20);
101
102 select * from libros
103 where (autor='Borges' or editorial='Paidos') and
104 (precio<20);
105
106
107
108

```

	codigo [PK] integer	titulo character varying(40)	autor character varying(20)	editorial character varying(20)	precio numeric(6,2)
1	1	El Aleph	Borges	Emece	15.90
2	6	Martin Fierro	Jose Hernandez	Paidos	16.80
3	8	Cervantes y el Quijote	Borges	Paidos	18.40

Total rows: 3 of 3    Query complete 00:00:00.072    Successfully run. Total query runtime: 72 msec. 3 rows affected.

En el primer select sabemos que estamos seleccionando una columna y lo que queremos que muestre en este caso a Borges y si no lo está mostramos otro condicional en este caso Borges no cuenta con otro parámetro más que mostrar los que contengan ese nombre sin necesidad de verificar el precio.

En el segundo select estás diciendo que te muestre los dos parámetros si los tiene pero con el cambio que los parámetros se condicionan es decir que el precio sea menor que 20 afecta los parámetros anteriores y viceversa.

#### PROBLEMA 4.-

Una concesionaria de autos vende autos usados y almacena la información en una tabla

llamada "autos"(patente char(6), marca varchar(20), modelo char(4), precio decimal(8,2), primary key(patente) )

Inserte los siguientes registros:

```
('ACD123','Fiat 128','1970',15000);
('ACG234','Renault 11','1980',40000);
('BCD333','Peugeot 505','1990',80000);
('GCD123','Renault Clio','1995',70000);
('BCC333','Renault Megane','1998',95000);
('BVF543','Fiat 128','1975',20000);
```

1- Seleccione todos los autos cuyo modelo se encuentre entre '1970' y '1990' usando el Operador "between" y ordénelos por dicho campo(4 registros)

pgAdmin 4

File ▾ Object ▾ Tools ▾ Help ▾

Browser

Servers (1)  
PostgreSQL 15  
Databases (2)  
Actividad SSPBD

Query Query History

```
10 patient char(6),
11 name varchar(20),
12 modelo char(4),
13 precio decimal(10,2),
14 primary key(patient)
15 );
16
17 INSERT INTO autos
18 VALUES ('ACD123','Fiat 128','1970',15000),
19 ('AGC224','Renault 11','1980',40000),
20 ('BCD333','Peugeot 505','1990',80000),
21 ('BCC223','Renault Clio','1995',70000),
22 ('BCC223','Renault Megane','1998',95000),
23 ('BVF543','Fiat 128','1975',20000);
24
25 SELECT * FROM autos
26
27 1- Seleccionne todos los autos cuyo modelo se encuentre entre '1970' y '1990' usando el
28 Operador "between" y ordenelos por dicho campo(4 registros)
29
30
31
32
33
34
35 2- Seleccionne todos los autos cuyo precio esté entre 50000 y 100000.
36
37
```

Data Output Messages Notifications

patient	name	modelo	precio
ACD123	Fiat 128	1970	15000.00
BVF543	Fiat 128	1975	20000.00
AGC224	Renault 11	1980	40000.00
BCD333	Peugeot 505	1990	80000.00

Total rows: 4 of 4    Query complete 00:00:00.070    ✓ Successfully run. Total query runtime: 70 msec. 4 rows affected. [x]

Buscar Vinculos 22°C Soleado 12:00 p. m. 20/03/2023

**SELECT \* FROM autos**

WHERE modelo BETWEEN '1970' AND '1990'

ORDER BY modelo

2- Seleccione todos los autos cuyo precio esté entre 50000 y 100000.

pgAdmin 4

File Object Tools Help

Browser

Servers (1)  
PostgreSQL 15  
Databases (2)  
Activated SSPBD

Casts  
Catalogs  
Event Triggers  
Extensions  
Foreign Data Wrappers  
Languages  
Publications  
Schemas (1)  
public  
Aggregates  
Collations  
Domains  
FTB Configurations  
FTB Dictionaries  
FTB Parsers  
FTB Templates  
Foreign Tables  
Functions  
Materialized Views  
Operators  
Procedures  
Sequences  
Tables  
Trigger Functions  
Types  
Views  
Subscriptions

postgres  
Login/Group Roles  
Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Processes Actividad SSPBD/postgres@PostgreSQL 15\*

Query Query History

```
109     patente char(6),
110     marca varchar(20),
111     modelo char(4),
112     precio decimal(8,2),
113     primary key(patente)
114   );
115
116
117 INSERT INTO autos
118   VALUES ('ACD123','Fiat 128','1970',15000),
119   ('AGC224','Renault 11','1980',40000),
120   ('BCD333','Peugeot 505','1990',80000),
121   ('GCC123','Renault Clio','1995',70000),
122   ('BCC333','Renault Megane','1998',95000),
123   ('BVF542','Fiat 128','1975',20000);
124
125 SELECT * FROM autos
126
127 1- Seleccionne todos los autos cuyo modelo se encuentre entre '1970' y '1990' usando el
128 Operador "between" y ordénelos por dicho campo(4 registros)
129
130
131 SELECT * FROM autos
132 WHERE precio BETWEEN 50000 AND 100000
133 ORDER BY precio
134
135 2- Seleccionne todos los autos cuyo precio esté entre 50000 y 100000.
136
137
```

Data Output Messages Notifications

	patente	marca	modelo	precio
1	ACD123	Renault	Clio	15000.00
2	BCD333	Peugeot	505	80000.00
3	BCC333	Renault	Megane	95000.00

Total rows: 3 of 3    Query complete 00:00:00.072    ✓ Successfully run. Total query runtime: 72 msec. 3 rows affected.

Buscar Vinculos 22°C Soleado 12:21 p.m. 20/01/2023

```
SELECT * FROM autos  
WHERE precio BETWEEN 50000 AND 100000  
ORDER BY precio
```

**PROBLEMA 5.-**

Trabaje con la tabla medicamentos(

codigo serial,  
nombre varchar(20),  
laboratorio varchar(20),  
precio decimal(6,2),  
cantidad smallint,  
fechavencimiento date not null,  
primary key(codigo)

)

1- Ingrese algunos registros:

```
('Sertal','Roche',5.2,1,'2005-02-01');  
('Buscapina','Roche',4.10,3,'2006-03-01');  
('Amoxidal 500','Bayer',15.60,100,'2007-05-01');  
('Paracetamol 500','Bago',1.90,20,'2008-02-01');  
('Bayaspirina','Bayer',2.10,150,'2009-12-01');  
('Amoxidal jarabe','Bayer',5.10,250,'2010-10-01');
```

2- Recupere los nombres y precios de los medicamentos cuyo laboratorio sea "Bayer" o "Bago" empleando el operador "in" (4 registros)

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of database objects under 'Actividad SSPBD'. The main area contains a query editor with the following SQL code:

```
142    laboratorio varchar(20),
143    precio decimal(6,2),
144    cantidad smallint,
145    fechavencimiento date not null,
146    primary key(codigo)
147  );
148
149
150  INSERT INTO medicamentos (nombre,laboratorio,precio,cantidad,fechavencimiento)
151  VALUES ('Sertal','Roche',5.2,1,'2005-02-01'),
152  ('Buscapina','Roche',4.19,3,'2005-03-01'),
153  ('Amoxicida 500','Bayer',15.60,100,'2007-05-01'),
154  ('Paracetamol 500','Bago',1.99,20,'2008-02-01'),
155  ('Bayaspirina','Bayer',2.18,150,'2009-12-01'),
156  ('Amoxicida jarabe','Bayer',5.10,250,'2010-10-01');
157
158  SELECT * FROM medicamentos
159
160  2- Recupere los nombres y precios de los medicamentos
161  cuyo laboratorio sea "Bayer" o "Bago" empleando el operador "in" (4 registros)
162
163  SELECT nombre,precio,laboratorio FROM medicamentos
164  WHERE laboratorio IN ('Bayer', 'Bago')
165
166
167
168
169
170
```

The results grid shows four rows of data:

	nombre	precio	laboratorio
1	Amoxicida 500	15.60	Bayer
2	Paracetamol 500	1.90	Bago
3	Bayaspirina	2.10	Bayer
4	Amoxicida jarabe	5.10	Bayer

Total rows: 4 of 4    Query complete 00:00:00.108

Successfully run. Total query runtime: 108 msec. 4 rows affected.

**SELECT nombre,precio,laboratorio FROM medicamentos**

**WHERE laboratorio IN ('Bayer', 'Bago')**

3- Seleccione los remedios cuya cantidad se encuentre entre 1 y 5 empleando el

operador "between" y luego el operador "in" (2 registros)

Note que es más conveniente

emplear, en este caso, el operador "between".

BETWEEN

The screenshot shows the pgAdmin 4 interface. In the left sidebar, under 'Servers (1)', 'PostgreSQL 15' is selected, and 'Databases (2)' contains 'Actividad SSPBD'. The main area displays a query in the SQL tab:

```
152 ("Buscapina", 'Roche', 4,10,3,'2006-03-01'),
153 ("Anoxidal 500", 'Bayer', 15,60,100,'2007-05-01'),
154 ("Paracetamol 500", 'Bago', 1,19,20,'2008-02-01'),
155 ("Bayaspirina", 'Bayer', 2,18,150,'2009-12-01'),
156 ("Anoxidal jarabe", 'Bayer', 5,10,250,'2010-10-01');

157
158 SELECT * FROM medicamentos
159
160 2- Recupere los nombres y precios de los medicamentos
161 cuyo laboratorio sea "Bayer" o "Bago" empleando el operador "in" (4 registros)
162
163 SELECT nombre,precio,laboratorio FROM medicamentos
164 WHERE laboratorio IN ('Bayer', 'Bago')
165
166 3- Seleccione los remedios cuya cantidad se encuentre entre 1 y 5 empleando el
167 operador "between" y luego el operador "in" (2 registros)
168 Note que es más conveniente emplear, en este caso, el operador "between".
169
170 SELECT nombre,precio,laboratorio,cantidad FROM medicamentos
171 WHERE cantidad BETWEEN 1 AND 5
172
173
174
175
176
177
178
179
```

The Data Output tab shows the results of the query:

nombre	precio	laboratorio	cantidad
Serial	5.20	Roche	1
Buscapina	4.10	Roche	3

Total rows: 2 of 2    Query complete 00:00:00.072    Successfully run. Total query runtime: 72 msec. 2 rows affected.

SELECT nombre,precio,laboratorio,cantidad FROM medicamentos

WHERE cantidad BETWEEN 1 AND 5

IN

The screenshot shows the pgAdmin 4 interface. In the left sidebar, under 'Servers (1)', 'PostgreSQL 15' is selected, and 'Databases (2)' contains 'Actividad SSPBD'. The main area displays a query in the SQL tab:

```
152 ("Buscapina", 'Roche', 4,10,3,'2006-03-01'),
153 ("Anoxidal 500", 'Bayer', 15,60,100,'2007-05-01'),
154 ("Paracetamol 500", 'Bago', 1,19,20,'2008-02-01'),
155 ("Bayaspirina", 'Bayer', 2,18,150,'2009-12-01'),
156 ("Anoxidal jarabe", 'Bayer', 5,10,250,'2010-10-01');

157
158 SELECT * FROM medicamentos
159
160 2- Recupere los nombres y precios de los medicamentos
161 cuyo laboratorio sea "Bayer" o "Bago" empleando el operador "in" (4 registros)
162
163 SELECT nombre,precio,laboratorio FROM medicamentos
164 WHERE laboratorio IN ('Bayer', 'Bago')
165
166 3- Seleccione los remedios cuya cantidad se encuentre entre 1 y 5 empleando el
167 operador "between" y luego el operador "in" (2 registros)
168 Note que es más conveniente emplear, en este caso, el operador "between".
169
170 SELECT nombre,precio,laboratorio,cantidad FROM medicamentos
171 WHERE cantidad IN (1,5)
172
173
174
175
176
177
178
179
```

The Data Output tab shows the results of the query:

nombre	precio	laboratorio	cantidad
Serial	5.20	Roche	1

SELECT nombre,precio,laboratorio FROM medicamentos WHERE laboratorio IN ('Bayer', 'Bago')

## PROBLEMA 6.-

Una empresa almacena los datos de sus empleados en una tabla "empleados".

1- Cree la tabla empleados( nombre varchar(30), documento char(8), domicilio varchar(30), fechalingreso date, seccion varchar(20), sueldo decimal(6,2), primary key(documento) )

2- Ingrese algunos registros:

```
('Juan Perez','22333444','Colon 123','1990-10-08','Gerencia',900.50);
('Ana Acosta','23444555','Caseros 987','1995-12-18','Secretaria',590.30);
('Lucas Duarte','25666777','Sucre 235','2005-05-15','Sistemas',790);
('Pamela Gonzalez','26777888','Sarmiento 873','1999-02-12','Secretaria',550);
('Marcos Juarez','30000111','Rivadavia 801','2002-09-22','Contaduria',630.70);
('Yolanda Perez','35111222','Colon 180','1990-10-08','Administracion',400);
('Rodolfo Perez','35555888','Coronel Olmedo 588','1990-05-28','Sistemas',800);
```

3- Muestre todos los empleados con apellido "Perez" empleando el operador "like" (3 registros)

```
3- Muestre todos los empleados con apellido "Perez" empleando el operador "like" (3 registros)
select * from empleados
where nombre like '%Perez%';
```

nombre	documento	domicilio	fechalingreso	seccion	sueldo
1 Juan Perez	22333444	Colon 123	1990-10-08	Gerencia	900.50
2 Yolanda Perez	35111222	Colon 180	1990-10-08	Administracion	400.00
3 Rodolfo Perez	35555888	Coronel Olmedo 588	1990-05-28	Sistemas	800.00

select \* from empleados

where nombre like '%Perez%';

4.-Muestre todos los empleados cuyo domicilio comience con "Co" y tengan un "8" (2 registros)

```

VALUES ('Juan Perez', '22333444', 'Colon 123', '1990-10-08', 'Gerencia', 900.50),
       ('Ana Acosta', '23444555', 'Caseros 987', '1995-12-18', 'Secretaria', 590.30),
       ('Lucas Duarte', '25666777', 'Sucre 235', '2005-05-15', 'Sistemas', 790.00),
       ('Pamela Gonzalez', '26777888', 'Sarmiento 873', '1999-02-12', 'Secretaria', 550.00),
       ('Marcos Juarez', '30000111', 'Rivadavia 801', '2002-09-22', 'Contaduria', 630.70),
       ('Yolanda Perez', '35111222', 'Colon 180', '1990-10-08', 'Administracion', 400),
       ('Rodolfo Perez', '35555888', 'Coronel Olmedo 588', '1990-05-28', 'Sistemas', 800.00);

3- Muestre todos los empleados con apellido "Perez" empleando el operador "like" (3 registros)

select * from empleados
where (domicilio like 'Co%') AND (domicilio like '%8%');

4.-Muestre todos los empleados cuyo domicilio comience con "Co" y tengan un "8" (2 registros)

Total rows: 2 of 2 | Query complete 00:00:00.115

```

`select * from empleados  
where (domicilio like 'Co%') AND (domicilio like '%8%');`

5- Muestre todos los nombres y sueldos de los empleados cuyos sueldos incluyen centavos (3 registros)

```

CREATE TABLE empleados (
    nombre character varying(30),
    documento integer,
    domicilio character varying(30),
    fechalingreso date,
    seccion character varying(20),
    sueldo numeric(6,2)
);

SELECT * FROM empleados;

INSERT INTO empleados
VALUES ('Juan Perez', '22333444', 'Colon 123', '1990-10-08', 'Gerencia', 900.50),
       ('Ana Acosta', '23444555', 'Caseros 987', '1995-12-18', 'Secretaria', 590.30),
       ('Lucas Duarte', '25666777', 'Sucre 235', '2005-05-15', 'Sistemas', 790.00),
       ('Pamela Gonzalez', '26777888', 'Sarmiento 873', '1999-02-12', 'Secretaria', 550.00),
       ('Marcos Juarez', '30000111', 'Rivadavia 801', '2002-09-22', 'Contaduria', 630.70),
       ('Yolanda Perez', '35111222', 'Colon 180', '1990-10-08', 'Administracion', 400),
       ('Rodolfo Perez', '35555888', 'Coronel Olmedo 588', '1990-05-28', 'Sistemas', 800);

3- Muestre todos los empleados con apellido "Perez" empleando el operador "like" (3 registros)

select * from empleados
where cast(sueldo as varchar(80)) like '%.%';

4.-Muestre todos los empleados cuyo domicilio comience con "Co" y tengan un "8" (2 registros)

Total rows: 7 of 7 | Query complete 00:00:00.111 | Successfully run. Total query runtime: 111 msec. 7 rows affected.

```

`select * from empleados  
where cast(sueldo as varchar(80)) like '%.%'`

## 6.- Muestre los empleados que hayan ingresado en "1990" (3 registros)

The screenshot shows the pgAdmin 4 interface. In the left sidebar, under 'Servers (1)', there is one entry: 'PostgreSQL 15'. Under 'PostgreSQL 15', there is a 'Databases (2)' section with 'Actividad SSPBD' selected. The 'Tables' section of 'Actividad SSPBD' contains the 'empleados' table. The 'Query' tab at the top has the following SQL code:

```
179  alter table empleados
180    add column fechasingreso date;
181    section varchar(20);
182    sueldo decimal(6,2);
183    primary key(documento);
184  );
185
186  SELECT * FROM empleados
187
188  INSERT INTO empleados
189  VALUES ('Juan Perez', '22331444', 'Colon 123', '1990-10-08', 'Gerencia', 900.50),
190  ('Ana Acosta', '23444555', 'Caseros 887', '1995-12-18', 'Secretaria', 500.30),
191  ('Lucas Duarte', '25666777', 'Sucre 2351', '2005-05-15', 'Sistemas', 700),
192  ('Pamela Gonzalez', '26777888', 'Sarmiento 873', '1999-02-12', 'Secretaria', 650),
193  ('Marcos Juarez', '30000111', 'Rivadavia 801', '2002-09-22', 'Contaduria', 630.70),
194  ('Yolanda Perez', '35111222', 'Colon 180', '1990-10-08', 'Administracion', 400),
195  ('Rodolfo Perez', '35555888', 'Coronel Olmedo 588', '1999-05-28', 'Sistemas', 800);
196
197  3- Muestre todos los empleados con apellido "Perez" empleando el operador "like" (3
198  registros)
199
200
201  select * from empleados
202  where cast(fechasingreso as varchar(80)) like '%1990%';
203
204
205  4.-Muestre todos los empleados cuyo domicilio comience con "Co" y tengan un "a" (2 registros)
206
207
208
```

The 'Data Output' tab shows the results of the query:

nombre	documento	domicilio	fechasingreso	sección	sueldo
Juan Perez	22331444	Colon 123	1990-10-08	Gerencia	900.50
Yolanda Perez	35111222	Colon 180	1990-10-08	Administracion	400.00
Rodolfo Perez	35555888	Coronel Olmedo 588	1990-05-28	Sistemas	800.00

Total rows: 3 of 3    Query complete 00:00:00.183

Successfully run. Total query runtime: 183 msec. 3 rows affected.

select \* from empleados

where cast(fechasingreso as varchar(80)) like '%1990%'