

Juan Felipe Cucalon 201820969

Gabriela Caballero 201812917

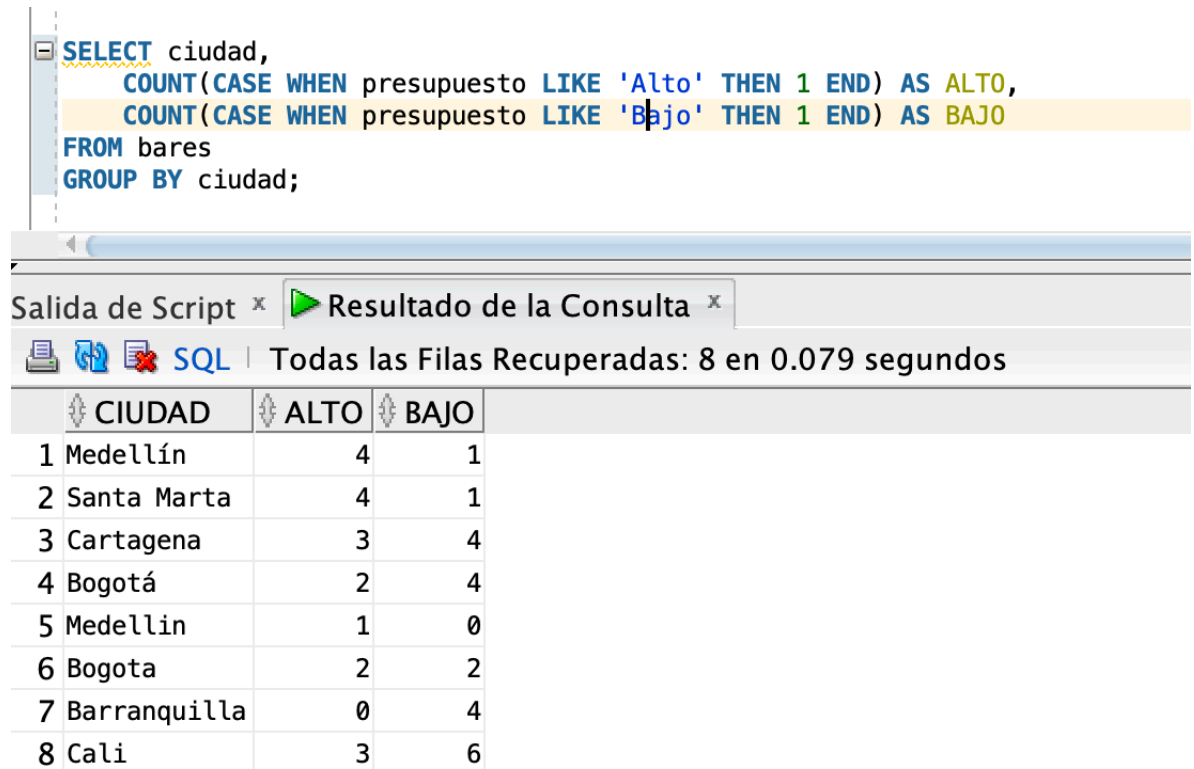
Taller 2

1. **(192*) Calentamiento.** Se quiere saber, para cada ciudad, su nombre y el número de bares de presupuesto 'Alto' y el número de bares de presupuesto 'Bajo'

Texto plano de SQL:

```
ALTER SESSION SET CURRENT_SCHEMA = PARRANDEROS;

SELECT ciudad,
       COUNT(CASE WHEN presupuesto LIKE 'Alto' THEN 1 END) AS ALTO,
       COUNT(CASE WHEN presupuesto LIKE 'Bajo' THEN 1 END) AS BAJO
FROM bares
GROUP BY ciudad;
```



The screenshot shows a SQL IDE interface. At the top, the SQL query is entered in a text area. Below the text area, there are two tabs: 'Salida de Script' and 'Resultado de la Consulta'. The 'Resultado de la Consulta' tab is active, displaying the results of the query in a table format. The table has three columns: 'CIUDAD', 'ALTO', and 'BAJO'. The results are as follows:

	CIUDAD	ALTO	BAJO
1	Medellín	4	1
2	Santa Marta	4	1
3	Cartagena	3	4
4	Bogotá	2	4
5	Medellin	1	0
6	Bogota	2	2
7	Barranquilla	0	4
8	Cali	3	6

2. **(201) Información de las bares y horarios:** Para cada bar, se quiere saber su mejor horario de venta a foráneos a quienes les gusta un cierto tipo de bebida, dado el nombre del tipo de bebida (por ejemplo, gaseosa). El mejor horario de venta a foráneos es aquel que es más frecuentado por los bebedores que **no** provienen de su misma ciudad. La respuesta debe contener el nombre del bar, el

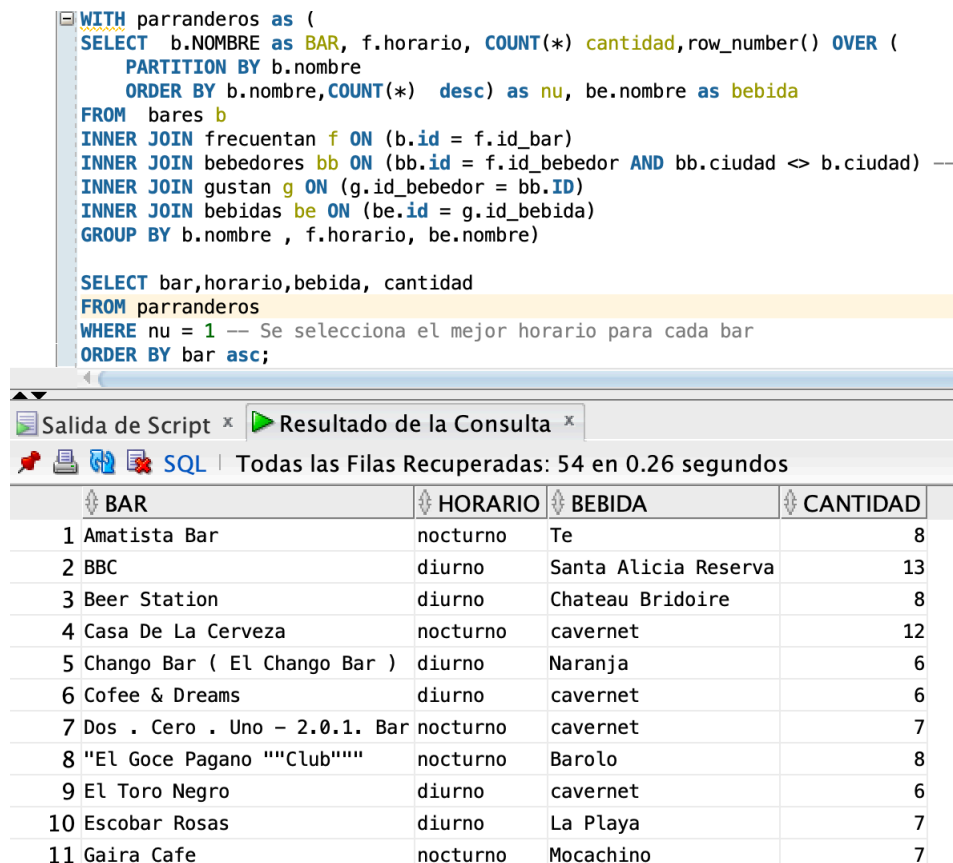
mejor horario y el número de bebedores foráneos que lo frecuentan en ese horario y les gusta el tipo de bebida indicada. El resultado debe estar ordenado de forma ascendente por el nombre del bar

Texto plano de SQL:

```
ALTER SESSION SET CURRENT_SCHEMA = PARRANDEROS;

WITH parranderos as (
SELECT b.NOMBRE as BAR, f.horario, COUNT(*) cantidad,row_number() OVER (
PARTITION BY b.nombre
ORDER BY b.nombre,COUNT(*) desc) as nu, be.nombre as bebida
FROM bares b
INNER JOIN frecuentan f ON (b.id = f.id_bar)
INNER JOIN bebedores bb ON (bb.id = f.id_bebedor AND bb.ciudad <> b.ciudad)
-- <> ciudad de los bebedores diferente a la ciudad de los bares
INNER JOIN gustan g ON (g.id_bebedor = bb.ID)
INNER JOIN bebidas be ON (be.id = g.id_bebida)
GROUP BY b.nombre , f.horario, be.nombre)

SELECT bar,horario,bebida, cantidad
FROM parranderos
WHERE nu = 1 -- Se selecciona el mejor horario para cada bar
ORDER BY bar asc;
```



The screenshot shows a SQL IDE with a query editor and a results window. The query is the same as the one provided in the text. The results window shows 54 rows of data, but only the first 11 rows are visible in the table below.

BAR	HORARIO	BEBIDA	CANTIDAD
1 Amatista Bar	nocturno	Te	8
2 BBC	diurno	Santa Alicia Reserva	13
3 Beer Station	diurno	Chateau Bridoire	8
4 Casa De La Cerveza	nocturno	cavernet	12
5 Chango Bar (El Chango Bar)	diurno	Naranja	6
6 Cofee & Dreams	diurno	cavernet	6
7 Dos . Cero . Uno - 2.0.1. Bar	nocturno	cavernet	7
8 "El Goce Pagano ""Club"""	nocturno	Barolo	8
9 El Toro Negro	diurno	cavernet	6
10 Escobar Rosas	diurno	La Playa	7
11 Gaira Cafe	nocturno	Mocachino	7

- (192) Los diez (10) bebedores con más apariciones en Parranderos: Se desea obtener el Id del bebedor, el nombre, cuántas veces aparece en la base de datos y su ciudad, ordenadas por su

número de apariciones descendientemente. Si varios bebedores aparecen el mismo número de veces, deben ordenarse por su ciudad

Texto plano de SQL:

```
ALTER SESSION SET CURRENT_SCHEMA = PARRANDEROS;

WITH parranderos as(
SELECT nombre nombre, COUNT(*) as numero
FROM bebedores
GROUP BY nombre
ORDER BY numero desc
)
SELECT p.nombre,b.id,b.ciudad, p.numero
FROM parranderos p
INNER JOIN bebedores b on b.nombre like p.nombre
WHERE ROWNUM <= 10
ORDER BY p.numero desc, ciudad;
```




```

WITH parranderos as(
SELECT nombre nombre, COUNT(*) as numero
FROM bebedores
GROUP BY nombre
ORDER BY numero desc
)
SELECT p.nombre,b.id,b.ciudad, p.numero
FROM parranderos p
INNER JOIN bebedores b on b.nombre like p.nombre
WHERE ROWNUM <= 10
ORDER BY p.numero desc, ciudad;

```

Salida de Script x

Resultado de la Consulta x

   SQL | Todas las Filas Recuperadas: 10 en 0.113 segundos

	NOMBRE	ID	CIUDAD	NUMERO
1	Krishna Bernal	2040	Barranquilla	7
2	Krishna Bernal	950	Bogotá	7
3	Krishna Bernal	69	Bogotá	7
4	Krishna Bernal	1147	Cali	7
5	Krishna Bernal	1053	Cali	7
6	Krishna Bernal	230	Cartagena	7
7	Krishna Bernal	1055	Medellín	7
8	Joaquin Diaz	375	Barranquilla	6
9	Joaquin Diaz	175	Cartagena	6
10	Joaquin Diaz	2046	Cartagena	6

- (192*) **La ciudad para lanzar una nueva bebida.** Se quiere sacar al mercado un nuevo vino, de 10 grados de alcohol, y entonces se quiere saber en qué ciudad sería mejor lanzar el piloto. La ciudad apropiada es aquella en la que se sirven más bebidas de ese tipo y entre 8 y 12 grados de alcohol. La respuesta esperada debe ser una tabla con el nombre de la ciudad y el número de bares, que debe ser el máximo, que sirven bebidas con las características descritas

SQL Texto Plano:

```

SELECT bares.ciudad, COUNT (bares.id) as numero_bares
FROM bebidas
INNER JOIN sirven s on s.id_bebida = bebidas.id
INNER JOIN bares on s.id_bar = bares.id
INNER JOIN tipo_bebida on bebidas.tipo = tipo_bebida.id
WHERE tipo_bebida.nombre = 'vino tinto'
AND parranderos.bebidas.grado_alcohol between 8 and 12
GROUP BY bares.ciudad
ORDER BY COUNT(bebidas.nombre) DESC;

```

```

SELECT bares.ciudad, COUNT (bares.id) as numero_bares
FROM bebidas
INNER JOIN sirven s on s.id_bebida = bebidas.id
INNER JOIN bares on s.id_bar = bares.id
INNER JOIN tipo_bebida on bebidas.tipo = tipo_bebida.id
WHERE tipo_bebida.nombre = 'vino tinto'
AND parranderos.bebidas.grado_alcohol between 8 and 12
GROUP BY bares.ciudad
ORDER BY COUNT(bebidas.nombre) DESC;

```

Salida de Script x Resultado de la Consulta x	
SQL Todas las Filas Recuperadas: 8 en 0.123 segundos	
CIUDAD	NUMERO_BARES
1 Cali	156
2 Bogotá	109
3 Cartagena	98
4 Santa Marta	92
5 Barranquilla	91
6 Medellín	81
7 Bogota	75
8 Medellin	8

5. (192*) Las diez (10) bebidas alcohólicas que más gustan a los bebedores “Viajeros”. Encontrar el identificador y el nombre y el tipo de las diez (10) bebidas que más gustan a los viajeros. Los bebedores viajeros son aquellos bebedores que frecuentan más bares de otras ciudades que bares en su misma ciudad de residencia

SQL Texto Plano:

```

ALTER SESSION SET CURRENT_SCHEMA = PARRANDEROS;
with bebedor as(
SELECT bebedores.id as id
from bebedores
inner join frecuentan f on f.id_bebedor Like bebedores.id

```

```

inner join bares b on f.id_bar like b.id and b.ciudad <> bebedores.ciudad )

SELECT be.id, be.nombre, be.tipo
From gustan g
inner join bebedor b on g.id_bebedor like b.id
inner join bebidas be on g.id_bebida like be.id
WHERE ROWNUM <= 10;

```

```

ALTER SESSION SET CURRENT_SCHEMA = PARRANDEROS;
with bebedor as(
SELECT bebedores.id as id
from bebedores
inner join frecuentan f on f.id_bebedor like bebedores.id
inner join bares b on f.id_bar like b.id and b.ciudad <> bebedores.ciudad )

SELECT be.id, be.nombre, be.tipo
From gustan g
inner join bebedor b on g.id_bebedor like b.id
inner join bebidas be on g.id_bebida like be.id
WHERE ROWNUM <= 10;

```

Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 10 en 0.135 segundos

ID	NOMBRE	TIPO
1	83 Edmeades	1
2	93 Guanabana	10
3	30 7Up	9
4	36 Colombiana Light	9
5	14 Costena	2
6	3 aguila	2
7	53 Café escoces	7
8	153 bebida153	9
9	468 bebida468	6
10	38 Coca Cola Zero	9

6. (202*) Información de bares: Obtener los nombres de los bares de cada ciudad que sirven entre 5 y 10 bebidas de más de 10% de alcohol y son de presupuesto alto. El resultado debe estar ordenado por ciudad, nombre del bar y número de bebidas

SQL Texto Plano:

```

with sirv as(
select count(*) as numero_bebidas, bares.nombre as nombre
from bebidas
inner join sirven on bebidas.id = sirven.id_bebida
inner join bares on sirven.id_bar = bares.id
where grado_alcohol > 10 and bares.presupuesto = 'Alto'
group by bares.nombre)

select numero_bebidas, nombre

```

```
from sirv
where numero_bebidas between 5 and 10;
```

Primero se hizo una consulta de las bebidas de más de 10° de alcohol y que son de presupuesto alto:

	NUMERO_BEBIDAS	NOMBRE
1	131	Irish Pub
2	42	Dos . Cero . Uno - 2.0.1. Bar
3	45	Nick Havana - Bar Karaoke
4	41	Palos De Moguer
5	73	BBC
6	58	Kukaramakara
7	42	The Beer Lounge
8	43	Amatista Bar
9	39	Mai Lirol Darlin
10	44	Sanalejo Café
11	46	Tabü Bar Restaurante
12	49	La Tea
13	43	The Full Pint
14	45	The Red Lion English Pub
15	52	Spin
16	53	El Toro Negro

Por lo que cuando se hace la consulta de ciudades que sirven entre 5 y 10 bebidas de más de 10% de alcohol y son de presupuesto alto, la tabla sale vacía, ya que no existen.

```
with sirv as(
select count(*) as numero_bebidas, bares.nombre as nombre
from bebidas
inner join sirven on bebidas.id = sirven.id_bebida
inner join bares on sirven.id_bar = bares.id
where grado_alcohol > 10 and bares.presupuesto = 'Alto'
group by bares.nombre)

select numero_bebidas, nombre
from sirv
where numero_bebidas between 5 and 10;
```

Salida de Script x

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 0 en 0.107 segundos

NUMER...	NOMBRE
----------	--------