

Taller 2: SQL parte 1

Carlos E. Figueredo Triana, Juan M. Rivera López
Sistemas transaccionales ISIS2304 2021-1
Profesor: German E. Bravo Córdoba
{ce.figueredo, [jm.rivera](mailto:jm.rivera@uniandes.edu.co)}@uniandes.edu.co
Fecha de presentación: 17 de marzo de 2021

Tabla de contenido

1. Introducción	1
2. Consultas sobre la base de datos “Parranderos”	1
2.1. Calentamiento.....	1
2.2. Información de los bares y horarios.....	2
2.3. Los diez (10) bebedores con más apariciones en Parranderos.....	4
2.4. La ciudad para lanzar una nueva bebida.....	5
2.5. Las diez (10) bebidas alcohólicas que más gustan a los bebedores “Viajeros”	6
2.6. Información de bares	9

1. Introducción

Con el fin de aprender a interactuar con una base de datos relacional, se realizarán una serie de búsquedas en la base de datos “Parranderos”.

2. Consultas sobre la base de datos “Parranderos”

2.1. 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’

Para contestar a esta pregunta se realizó la siguiente búsqueda

```
SELECT CIUDAD_NO, presupuesto, COUNT(presupuesto) AS CANTIDAD
FROM (SELECT bares.id, bares.nombre, bares.presupuesto,
bares.cant_sedes,
CASE WHEN bares.ciudad = 'Bogotá' THEN 'Bogota'
      WHEN bares.ciudad = 'Medellín' THEN 'Medellin'
      ELSE bares.ciudad
END AS CIUDAD_NO
FROM bares)BAR_NO
WHERE NOT presupuesto = 'Medio'
GROUP BY CIUDAD_NO, presupuesto
ORDER BY CIUDAD_NO;
```

Al ejecutar este código se obtuvo el resultado que se ve en la figura 1.

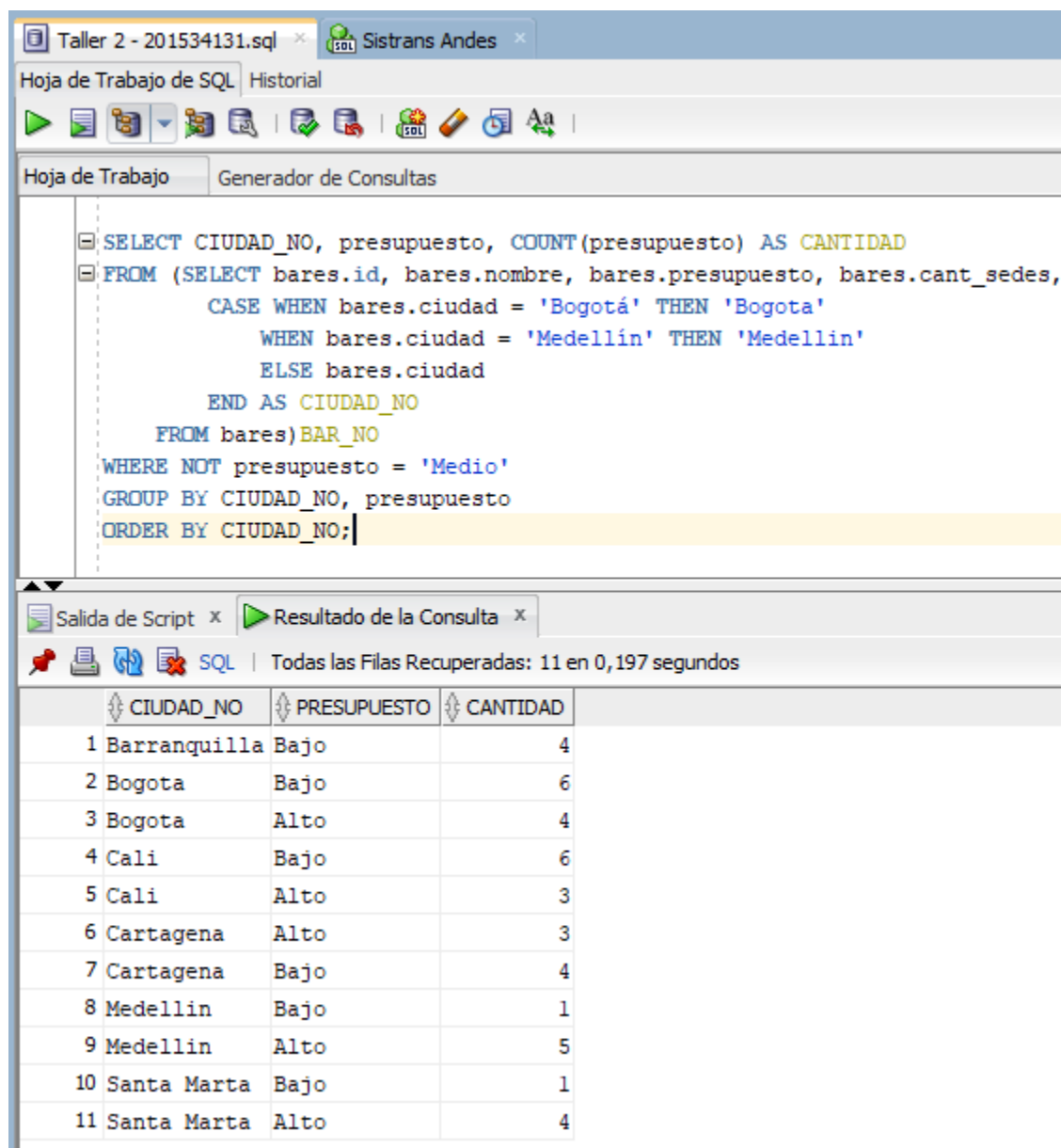


Figura 1. Búsqueda de bares con presupuesto alto y bajo para cada ciudad

2.2. Información de los 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.

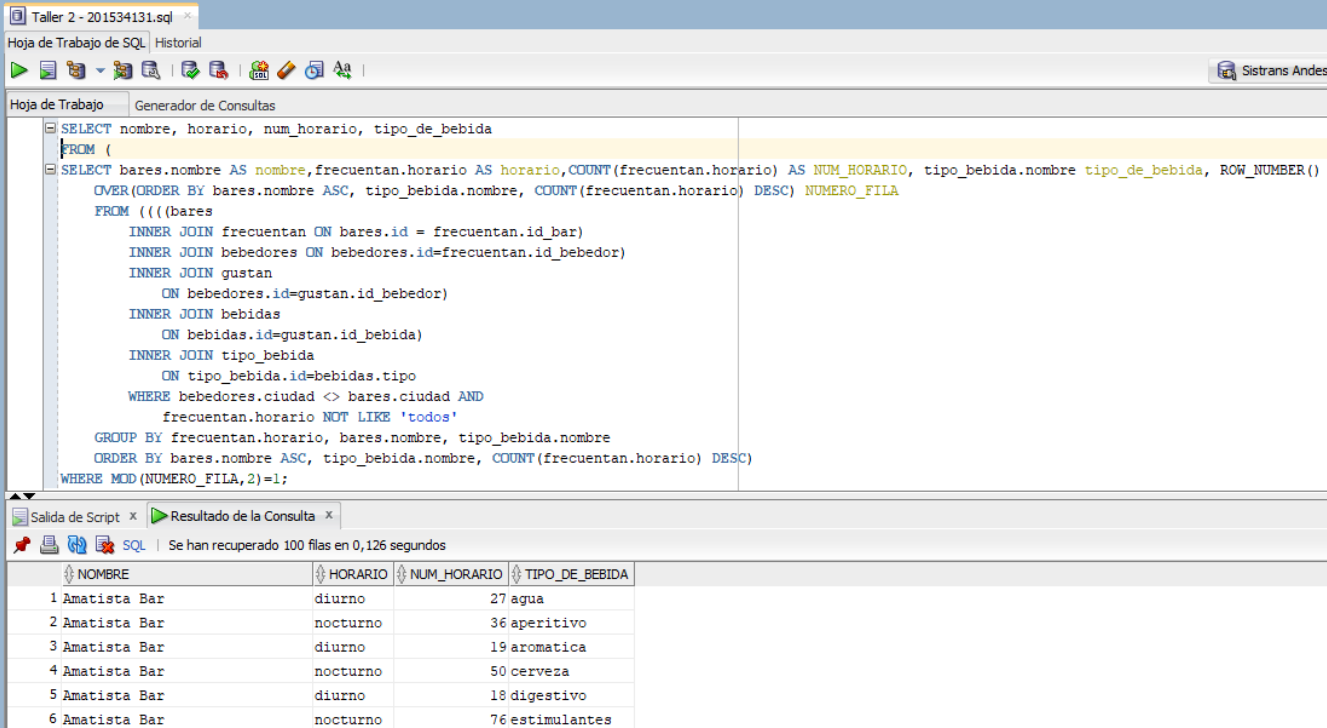
Para realizar a esta búsqueda se unieron las tablas que hacen referencia a los bares, los bebedores y las bebidas que les gustan. El resultado se ordenó de forma que se mostrara, para cada bar, el número de bebedores foráneos a los que les gusta un tipo de bebida y el horario en el que van al bar. Nótese que sólo se tuvieron en cuenta los horarios 'diurno' y 'nocturno', pues si un bebedor va en 'todos' se le contaría dos veces. El siguiente código realiza esta operación:

```

SELECT nombre, horario, num_horario, tipo_de_bebida
FROM (
SELECT bares.nombre AS nombre,frecuentan.horario AS
horario,COUNT(frecuentan.horario) AS NUM_HORARIO, tipo_bebida.nombre
tipo_de_bebida, ROW_NUMBER()
OVER(ORDER BY bares.nombre ASC, tipo_bebida.nombre,
COUNT(frecuentan.horario) DESC) NUMERO_FILA
FROM (((bares
INNER JOIN frecuentan ON bares.id = frecuentan.id_bar)
INNER JOIN bebedores ON bebedores.id=frecuentan.id_bebedor)
INNER JOIN gustan
ON bebedores.id=gustan.id_bebedor)
INNER JOIN bebidas
ON bebidas.id=gustan.id_bebida)
INNER JOIN tipo_bebida
ON tipo_bebida.id=bebidas.tipo
WHERE bebedores.ciudad <> bares.ciudad AND
frecuentan.horario NOT LIKE 'todos'
GROUP BY frecuentan.horario, bares.nombre, tipo_bebida.nombre
ORDER BY bares.nombre ASC, tipo_bebida.nombre,
COUNT(frecuentan.horario) DESC)
WHERE MOD(NUMERO_FILA,2)=1;

```

Al ejecutar este código se obtuvo el resultado que se ve en la figura 2.



Hoja de Trabajo de SQL | Historial

Hoja de Trabajo | Generador de Consultas

```

SELECT nombre, horario, num_horario, tipo_de_bebida
FROM (
SELECT bares.nombre AS nombre,frecuentan.horario AS
horario,COUNT(frecuentan.horario) AS NUM_HORARIO, tipo_bebida.nombre
tipo_de_bebida, ROW_NUMBER()
OVER(ORDER BY bares.nombre ASC, tipo_bebida.nombre,
COUNT(frecuentan.horario) DESC) NUMERO_FILA
FROM (((bares
INNER JOIN frecuentan ON bares.id = frecuentan.id_bar)
INNER JOIN bebedores ON bebedores.id=frecuentan.id_bebedor)
INNER JOIN gustan
ON bebedores.id=gustan.id_bebedor)
INNER JOIN bebidas
ON bebidas.id=gustan.id_bebida)
INNER JOIN tipo_bebida
ON tipo_bebida.id=bebidas.tipo
WHERE bebedores.ciudad <> bares.ciudad AND
frecuentan.horario NOT LIKE 'todos'
GROUP BY frecuentan.horario, bares.nombre, tipo_bebida.nombre
ORDER BY bares.nombre ASC, tipo_bebida.nombre,
COUNT(frecuentan.horario) DESC)
WHERE MOD(NUMERO_FILA,2)=1;

```

Salida de Script x | Resultado de la Consulta x

SQL | Se han recuperado 100 filas en 0,126 segundos

NOMBRE	HORARIO	NUM_HORARIO	TIPO_DE_BEBIDA
1 Amatista Bar	diurno	27	agua
2 Amatista Bar	nocturno	36	aperitivo
3 Amatista Bar	diurno	19	aromatica
4 Amatista Bar	nocturno	50	cerveza
5 Amatista Bar	diurno	18	digestivo
6 Amatista Bar	nocturno	76	estimulantes

Figura 2. Búsqueda del número de usuarios que frecuentan un bar, organizados de acuerdo con las bebidas que les gustan y al horario en el suelen ir al bar

2.3. 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

Se consideró como una aparición que un usuario frecuentara un bar (se registra una vez por bar) o que a un usuario le gustara una bebida (se registra una aparición por bebida), además de la una aparición por estar en la base de datos general. Con esto, se contó el número de apariciones y se eligieron a los 10 primeros usando el siguiente código:

```
SELECT bebedores.id, bebedores.nombre, APARICIONES, bebedores.ciudad
FROM bebedores
    INNER JOIN
        (SELECT id_bebedor, count(id_bebedor) AS apariciones
         FROM (SELECT frecuentan.id_bebedor
              FROM frecuentan
              UNION ALL
              SELECT bebedores.id
              FROM bebedores
              UNION ALL
              SELECT gustan.id_bebedor
              FROM gustan)
         GROUP BY id_bebedor)
    ON bebedores.id=id_bebedor
ORDER BY APARICIONES DESC, bebedores.ciudad
FETCH FIRST 10 ROW ONLY;
```

Al ejecutar este código se obtuvo el resultado que se ve en la figura 3.

```

27 SELECT bebedores.id, bebedores.nombre, APARICIONES, bebedores.ciudad
28 FROM bebedores
29     INNER JOIN
30     (SELECT id_bebedor, count(id_bebedor) AS apariciones
31     FROM (SELECT frecuentan.id_bebedor
32           FROM frecuentan
33           UNION ALL
34           SELECT bebedores.id
35           FROM bebedores
36           UNION ALL
37           SELECT gustan.id_bebedor
38           FROM gustan)
39     GROUP BY id_bebedor)
40 ON bebedores.id=id_bebedor
41 ORDER BY APARICIONES DESC, bebedores.ciudad
42 FETCH FIRST 10 ROW ONLY;

```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 10 en 0,021 segundos

ID	NOMBRE	APARICIONES	CIUDAD
1	Sergio Delgado	34	Medellín
2	751 Sofia Jimenez	31	Cartagena
3	803 Catalina Urrutia	30	Cali
4	362 Angelica Beltran	29	Bogotá
5	39 Catalina Romero	29	Medellín
6	550 Monserrat Diaz	28	Barranquilla
7	961 Benjamin Urrutia	28	Medellín

Figura 3. Búsqueda de los 10 bebedores que más apariciones tienen dentro del sistema

2.4. 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

Se realizó una búsqueda de los bares que sirvieran vino tinto (Cavernet, Merlot, Rosemount State...) y que tuvieran una graduación alcohólica entre 8° y 12°. Luego se agruparon y contaron el número de bares que cumplían con estas características usando el siguiente código:

```

SELECT CIUDAD_NO, count(CIUDAD_NO)
FROM (SELECT bares.id, bares.nombre, bares.presupuesto,
bares.cant_sedes,
CASE WHEN bares.ciudad = 'Bogotá' THEN 'Bogota'
      WHEN bares.ciudad = 'Medellín' THEN 'Medellin'
      ELSE bares.ciudad

```

```

        END AS CIUDAD_NO
    FROM bares) BAR_NO
    INNER JOIN sirven ON BAR_NO.id=sirven.id_bar
    INNER JOIN bebidas ON bebidas.id=sirven.id_bebida
WHERE bebidas.tipo=1 AND bebidas.grado_alcohol BETWEEN 8 AND 12
GROUP BY CIUDAD_NO
ORDER BY count(CIUDAD_NO) DESC
FETCH FIRST 1 ROWS ONLY;

```

Al ejecutar este código se obtuvo el resultado que se ve en la figura 4. La ciudad más apropiada para esto sería Bogotá, puesto que en total tiene 184 bares.

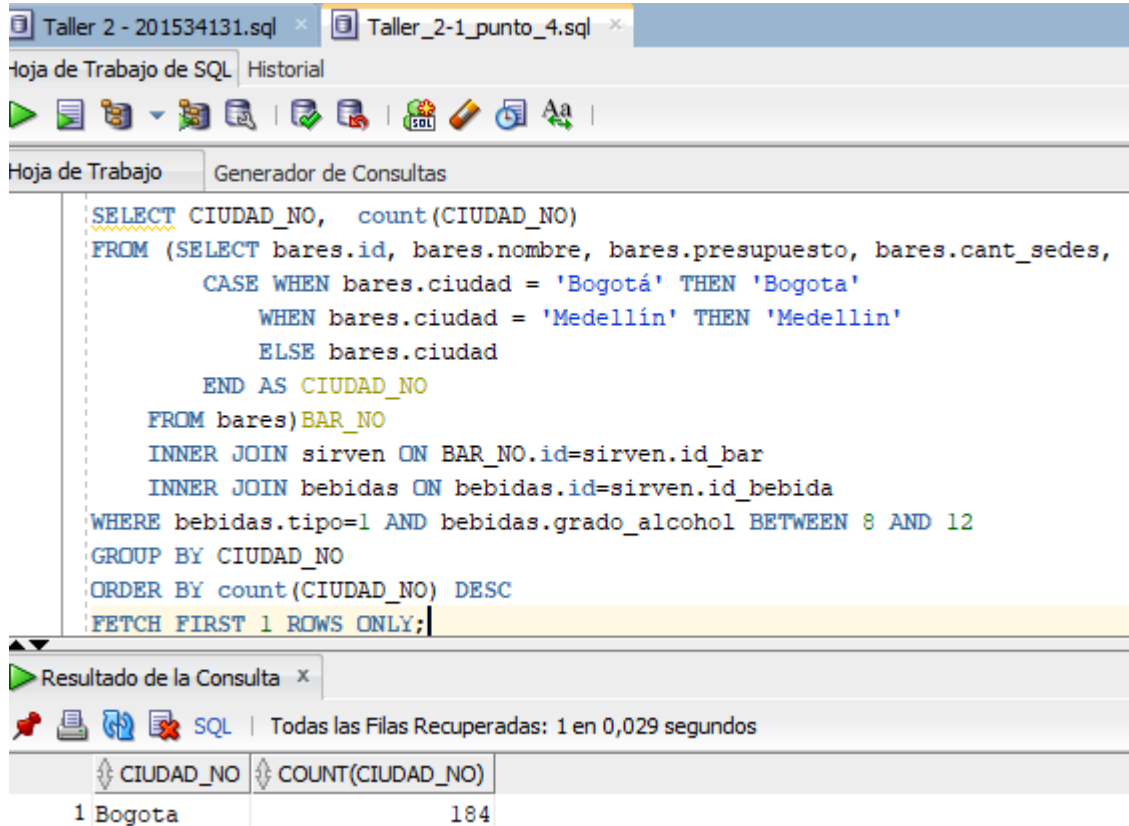


Figura 4. Búsqueda de la ciudad donde más bares sirven vino tinto

2.5. 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.

Lo primero que se hizo fue identificar a los bebedores que fueran viajeros. Para esto se contaron a cuántos bares frecuentaba un usuario en su misma ciudad y cuántos bares frecuentaba en otras ciudades. Nótese que si un viajero de Medellín iba a un bar que estaba localizado en Medellín la búsqueda lo tomará como de dos ciudades diferentes.

Luego se seleccionaron a aquellos que frecuentaran más bares en otras ciudades (VIAJERO>LOCAL). Luego se miraron cuántas bebidas había con un mismo id entre los bebedores viajeros. Para esto se implementó el siguiente código:

```

SELECT bebidas.id, COUNT(bebidas.id), bebidas.nombre,
tipo_bebida.nombre
FROM ((bebedores
      INNER JOIN gustan ON bebedores.id=gustan.id_bebedor)
      INNER JOIN bebidas ON bebidas.id=gustan.id_bebida)
      INNER JOIN tipo_bebida ON bebidas.tipo=tipo_bebida.id
WHERE bebedores.id IN (SELECT bebedores.id
      FROM ((bebedores
            INNER JOIN frecuentan ON
frecuentan.id_bebedor=bebedores.id)
            INNER JOIN bares ON frecuentan.id_bar=bares.id)
            INNER JOIN (SELECT bebedores.id AS beb_id,
COUNT(bebedores.id)AS VIAJERO
                        FROM (bebedores
                              INNER JOIN frecuentan ON
frecuentan.id_bebedor=bebedores.id)
                              INNER JOIN bares ON frecuentan.id_bar=bares.id
                              WHERE bares.ciudad<>bebedores.ciudad
                              GROUP BY bebedores.id
                              ORDER BY bebedores.id) ON beb_id=bebedores.id)
            INNER JOIN (SELECT bebedores.id AS beb_id2,
COUNT(bebedores.id)AS LOCAL
                        FROM (bebedores
                              INNER JOIN frecuentan ON
frecuentan.id_bebedor=bebedores.id)
                              INNER JOIN bares ON frecuentan.id_bar=bares.id
                              WHERE bares.ciudad=bebedores.ciudad
                              GROUP BY bebedores.id
                              ORDER BY bebedores.id) ON beb_id2=bebedores.id)
WHERE VIAJERO>LOCAL
      GROUP BY bebedores.id)
GROUP BY bebidas.id, bebidas.nombre, tipo_bebida.nombre
ORDER BY COUNT(bebidas.id)DESC
FETCH FIRST 10 ROWS ONLY;

```

Los resultados se pueden observar en la figura 5.

Taller 2 - 201534131.sql x

Hoja de Trabajo de SQL Historial

Hoja de Trabajo Generador de Consultas

```

57 --Punto 5
58 SELECT bebidas.id, COUNT(bebidas.id), bebidas.nombre, tipo_bebida.nombre
59 FROM ((bebedores
60     INNER JOIN gustan ON bebedores.id=gustan.id_bebedor)
61     INNER JOIN bebidas ON bebidas.id=gustan.id_bebida)
62     INNER JOIN tipo_bebida ON bebidas.tipo=tipo_bebida.id
63 WHERE bebedores.id IN (SELECT bebedores.id
64     FROM ((bebedores
65         INNER JOIN frecuentan ON frecuentan.id_bebedor=bebedores.id)
66         INNER JOIN bares ON frecuentan.id_bar=bares.id)
67     INNER JOIN (SELECT bebedores.id AS beb_id, COUNT(bebedores.id)AS VIAJERO
68         FROM (bebedores
69             INNER JOIN frecuentan ON frecuentan.id_bebedor=bebedores.id)
70             INNER JOIN bares ON frecuentan.id_bar=bares.id
71             WHERE bares.ciudad<>bebedores.ciudad
72             GROUP BY bebedores.id
73             ORDER BY bebedores.id) ON beb_id=bebedores.id)
74     INNER JOIN (SELECT bebedores.id AS beb_id2, COUNT(bebedores.id)AS LOCAL
75         FROM (bebedores
76             INNER JOIN frecuentan ON frecuentan.id_bebedor=bebedores.id)
77             INNER JOIN bares ON frecuentan.id_bar=bares.id
78             WHERE bares.ciudad=bebedores.ciudad
79             GROUP BY bebedores.id
80             ORDER BY bebedores.id) ON beb_id2=bebedores.id
81     WHERE VIAJERO>LOCAL
82     GROUP BY bebedores.id)
83 GROUP BY bebidas.id, bebidas.nombre, tipo_bebida.nombre
84 ORDER BY COUNT(bebidas.id)DESC
85 FETCH FIRST 10 ROWS ONLY;

```

Salida de Script x Resultado de la Consulta x Resultado de la Consulta 1 x

Todas las Filas Recuperadas: 10 en 0,034 segundos

ID	COUNT(BEBIDAS.ID)	NOMBRE	NOMBRE_1
1	1	75 cavernet	vino tinto
2	84	54 Chateau Bridoire	vino tinto
3	98	52 Mandarina	jugo
4	38	50 Coca Cola Zero	gaseosa
5	7	50 Tio pepe	jerez
6	19	49 Bud Light	cerveza
7	59	49 Moca frio	estimulantes
8	57	48 Batido Divino	estimulantes
9	74	48 Santa Alicia Reserva	vino tinto
10	28	48 Colombiana	gaseosa

Figura 5. Lista de bebidas preferidas por los bebedores viajeros

2.6. 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.

Para responder esto se filtraron los bares que fueran de presupuesto Alto. Luego se miró cuántas bebidas alcohólicas con más de 10° de alcohol servían estos bares. Para esto se implementó el siguiente código:

```
SELECT *
FROM (SELECT bares.id, COUNT(bebidas.id) AS NUM_BEBIDAS, bares.ciudad,
bares.nombre
      FROM (bares
            INNER JOIN sirven ON bares.id=sirven.id_bar)
            INNER JOIN bebidas ON sirven.id_bebida=bebidas.id
      WHERE bares.presupuesto='Alto' AND
            bebidas.grado_alcohol > 10
      GROUP BY bares.id, bares.ciudad, bares.nombre
      ORDER BY NUM_BEBIDAS)
WHERE num_bebidas BETWEEN 5 AND 10;
```

Los resultados se pueden observar en la figura 6.

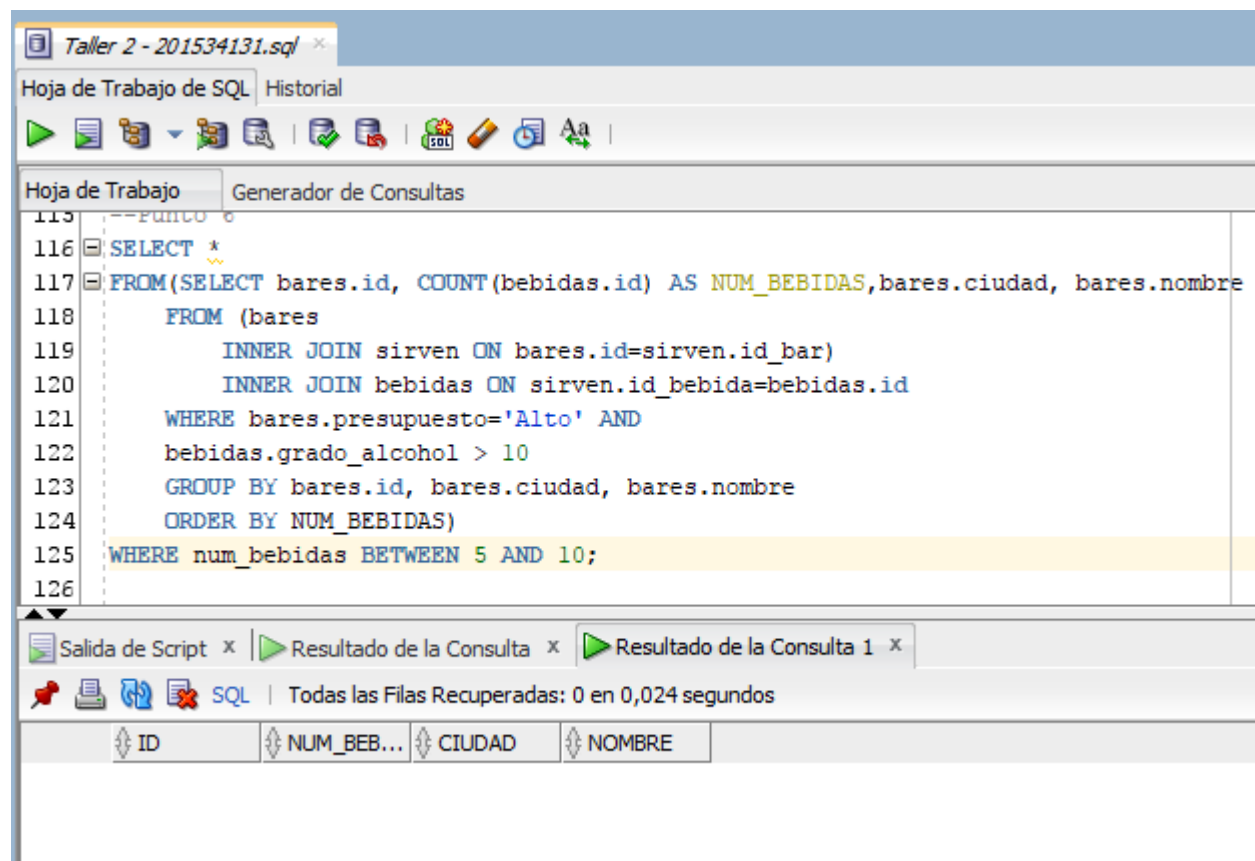


Figura 6. Bares de presupuesto alto que sirven entre 5 y 10 bebidas con más de 10° de alcohol

Para revisar que efectivamente no hubiera ningún bar con las características pedidas se revisó el número de bebidas con más de 10 grados de alcohol que servían los bares de presupuesto Alto. Efectivamente, el bar que menos bebidas vendía con más de 10 grados de alcohol es el Irish Pub de Medellín, que vende 39 bebidas con más de 10°. La lista se puede ver en la figura 7.

Hoja de Trabajo de SQL Historial

Hoja de Trabajo Generador de Consultas

```

135
136 SELECT bares.id, bares.ciudad, bares.nombre, COUNT(bebidas.id) AS NUM_BEBIDAS
137 FROM (bares
138       INNER JOIN sirven ON bares.id=sirven.id_bar)
139       INNER JOIN bebidas ON sirven.id_bebida=bebidas.id
140 WHERE bares.presupuesto='Alto' AND
141        bebidas.grado_alcohol > 10
142 GROUP BY bares.id, bares.ciudad, bares.nombre
143 ORDER BY NUM_BEBIDAS;

```

Salida de Script x Resultado de la Consulta x Resultado de la Consulta 1 x

Todas las Filas Recuperadas: 18 en 0,031 segundos

	ID	CIUDAD	NOMBRE	NUM_BEBIDAS
1	3	Medellin	Irish Pub	39
2	28	Santa Marta	Mai Lirol Darlin	39
3	36	Cali	Palos De Moguer	41
4	13	Bogotá	Dos . Cero . Uno - 2.0.1. Bar	42
5	50	Medellin	The Beer Lounge	42
6	8	Medellin	Amatista Bar	43
7	52	Cartagena	The Full Pint	43
8	45	Cali	Sanalejo Café	44
9	32	Santa Marta	Nick Havana - Bar Karaoke	45
10	54	Bogotá	The Red Lion English Pub	45
11	5	Bogota	Irish Pub	46
12	49	Cartagena	Tabü Bar Restaurante	46
13	4	Cali	Irish Pub	46
14	26	Medellin	La Tea	49

Figura 7. Bares de presupuesto alto que sirven bebidas con más de 10° de alcohol y el número de bebidas de este tipo que sirve cada uno