

Parte 2 del Taller 2 sobre SQL - Caso PARRANDEROS

Alonso Hernández Tavera, Cristian Bernardo Acuña Silva
Documento de reporte del Taller 2, parte 2– Caso PARRANDEROS
Universidad de los Andes, Bogotá, Colombia
{f.hernandezt, cb.acuna}@uniandes.edu.co
Fecha de presentación: marzo 21 de 2023

Tabla de contenido

1	Objetivo del taller	1
2	Desarrollo de los Requerimientos de consulta	1
2.1	Requerimiento 1:	1
2.2	Requerimiento 2:	2
2.3	Requerimiento 3:	4
2.4	Requerimiento 4:	5
2.5	Requerimiento 5:	7
3	Comentario final	10

1 Objetivo del taller

El objetivo principal que se puede identificar del taller es el de poner en práctica lo aprendido de forma autónoma y en clase sobre el lenguaje SQL. En este caso, el enfoque principal es realizar consultas sobre los **metadatos** de la base de datos donde se encuentra el esquema de ‘Parranderos’.

2 Desarrollo de los Requerimientos de consulta

2.1 Requerimiento 1:

Información general estadística de tablas: Obtener un listado con información de las tablas existentes en la base de datos de Parranderos. El listado debe incluir: El nombre de la tabla, el número de columnas que posee, el número de columnas que pueden tener NULL como valor y la fecha del último análisis hecho por Oracle sobre la tabla, ordenado por el nombre de la tabla.

Cláusula SQL utilizada para resolver el requerimiento

```
SELECT TABLE_NAME as NOMBRETABLA, count(COLUMN_NAME) as
NUMCOLUMNAS, count(case when NULLABLE='Y' then 1 end) as
NUMCOLUMNASNULL, LAST_ANALYZED as ULTIMOANALISIS
FROM ALL_TAB_COLUMNS
WHERE table_name IN ('BARES', 'BEBEDORES', 'BEBIDAS', 'FRECUENTAN',
'GUSTAN', 'SIRVEN', 'TIPO_BEBIDA')
group by TABLE_NAME, LAST_ANALYZED
order by TABLE_NAME;
```

Captura de pantalla de la cláusula y el resultado generado

```
4 SELECT TABLE_NAME as NOMBRETABLA, count(COLUMN_NAME) as NUMCOLUMNAS, count(case when NULLABLE='Y' then 1 end) as NUMCOLUMNASNULL, LAST_ANALYZED as ULTIMOANALISIS
5 FROM ALL_TAB_COLUMNS
6 WHERE table_name IN ('BARES', 'BEBEDORES', 'BEBIDAS', 'FRECUENTAN', 'GUSTAN', 'SIRVEN', 'TIPO_BEBIDA')
7 group by TABLE_NAME, LAST_ANALYZED
8 order by TABLE_NAME;
9
```

	NOMBRETABLA	NUMCOLUMNAS	NUMCOLUMNASNULL	ULTIMOANALISIS
1	BARES	5	4	10/03/23
2	BEBEDORES	4	3	10/03/23
3	BEBIDAS	4	3	24/09/20
4	FRECUENTAN	4	1	23/09/20
5	GUSTAN	2	0	28/02/21
6	GUSTAN	2	2	12/03/23
7	SIRVEN	3	0	21/08/20
8	SIRVEN	5	3	28/02/21
9	TIPO BEBIDA	2	1	24/09/20

2.2 Requerimiento 2:

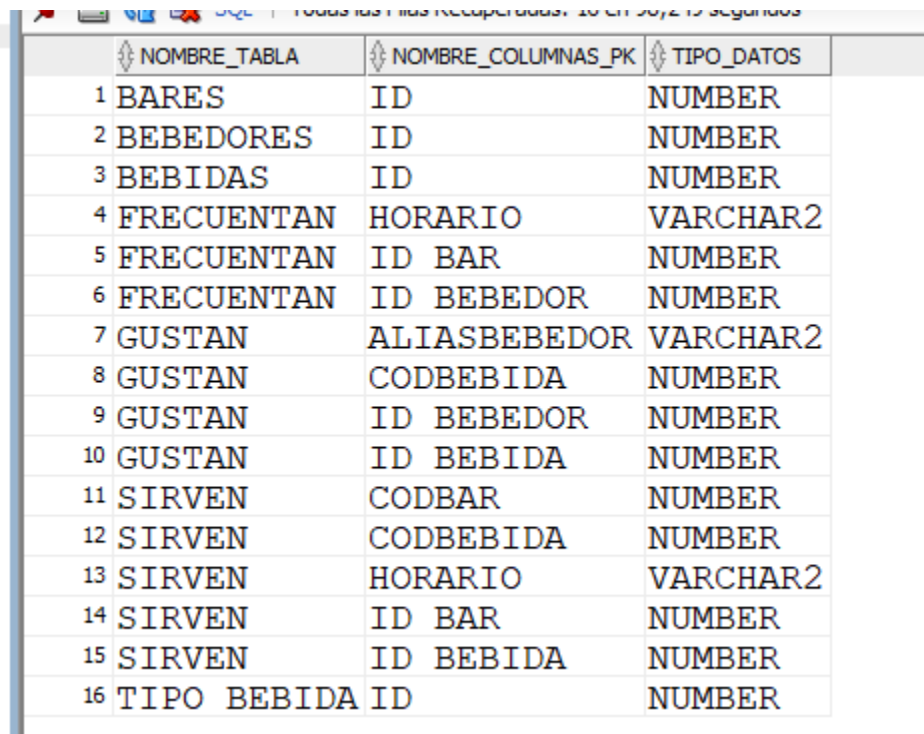
Información de las llaves primarias: Para todas las tablas de Parranderos se quiere mostrar el nombre de la tabla y la información de su llave primaria: cuáles son las columnas que la conforman y el tipo de datos de cada una. El resultado debe estar ordenado ascendentemente por nombre de la tabla y el nombre de la columna. Note que el nombre de la tabla se repite para cada una de las columnas de su llave primaria.

Cláusula SQL utilizada para resolver el requerimiento

```
SELECT distinct c.table_name as NOMBRE_TABLA,t.column_name as
NOMBRE_COLUMNAS_PK, tt.data_type as TIPO_DATOS
FROM ALL_constraints c inner join ALL_cons_COLUMNS t ON
t.constraint_NAME=c.constraint_NAME inner join all_tab_columns tt on
tt.column_name=t.column_name
WHERE tt.table_name IN ('BARES', 'BEBEDORES', 'BEBIDAS', 'FRECUENTAN',
'GUSTAN', 'SIRVEN', 'TIPO_BEBIDA')
and t.table_name IN ('BARES', 'BEBEDORES', 'BEBIDAS', 'FRECUENTAN', 'GUSTAN',
'SIRVEN', 'TIPO_BEBIDA')
and c.table_name IN ('BARES', 'BEBEDORES', 'BEBIDAS', 'FRECUENTAN', 'GUSTAN',
'SIRVEN', 'TIPO_BEBIDA')and c.constraint_type='P'
order by c.table_name ,t.column_name asc;
```

Captura de pantalla de la cláusula y el resultado generado

```
3 SELECT distinct c.table_name as NOMBRE_TABLA,t.column_name as NOMBRE_COLUMNAS_PK, tt.data_type as TIPO_DATOS
4 FROM ALL_constraints c inner join ALL_cons_COLUMNS t ON t.constraint_NAME=c.constraint_NAME inner join all_tab_columns tt on tt.column_name=t.column_name
5 WHERE tt.table_name IN ('BARES', 'BEBEDORES', 'BEBIDAS', 'FRECUENTAN', 'GUSTAN', 'SIRVEN', 'TIPO_BEBIDA')
6 and t.table_name IN ('BARES', 'BEBEDORES', 'BEBIDAS', 'FRECUENTAN', 'GUSTAN', 'SIRVEN', 'TIPO_BEBIDA')
7 and c.table_name IN ('BARES', 'BEBEDORES', 'BEBIDAS', 'FRECUENTAN', 'GUSTAN', 'SIRVEN', 'TIPO_BEBIDA')and c.constraint_type='P'
8 order by c.table_name ,t.column_name asc;
```



	NOMBRE_TABLA	NOMBRE_COLUMNAS_PK	TIPO_DATOS
1	BARES	ID	NUMBER
2	BEBEDORES	ID	NUMBER
3	BEBIDAS	ID	NUMBER
4	FRECUENTAN	HORARIO	VARCHAR2
5	FRECUENTAN	ID BAR	NUMBER
6	FRECUENTAN	ID BEBEDOR	NUMBER
7	GUSTAN	ALIASBEBEDOR	VARCHAR2
8	GUSTAN	CODBEBIDA	NUMBER
9	GUSTAN	ID BEBEDOR	NUMBER
10	GUSTAN	ID BEBIDA	NUMBER
11	SIRVEN	CODBAR	NUMBER
12	SIRVEN	CODBEBIDA	NUMBER
13	SIRVEN	HORARIO	VARCHAR2
14	SIRVEN	ID BAR	NUMBER
15	SIRVEN	ID BEBIDA	NUMBER
16	TIPO BEBIDA	ID	NUMBER

2.3 Requerimiento 3:

Caracterización de los tipos de datos de las tablas: Para todas las tablas de la aplicación (Parranderos) y para cada tipo de dato que utilice, se quiere mostrar cuántas columnas hay en la tabla de ese tipo de datos y el promedio de las longitudes de dichas columnas. El resultado debe estar ordenado de forma ascendente por el nombre de la tabla, el nombre, tipo de dato y el número de columnas con ese tipo de dato. Note que el nombre de la tabla se repite para cada tipo de dato que utilice.

Cláusula SQL utilizada para resolver el requerimiento

```
ALTER SESSION SET CURRENT_SCHEMA = PARRANDEROS;
SELECT
    table_name AS NombreTabla,
    data_type AS TipoDeDato,
    COUNT(column_name) AS NumColsTipoDato,
    ROUND(AVG(data_length), 2) AS PromedioLongitudCol
FROM
    ALL_TAB_COLUMNS
WHERE
    table_name IN ('BARES', 'BEBEDORES', 'BEBIDAS', 'FRECUENTAN', 'GUSTAN',
'SIRVEN', 'TIPO_BEBIDA')
GROUP BY
    table_name, data_type
ORDER BY
    table_name ASC,
    data_type ASC,
    COUNT(column_name) ASC;
```

Captura de pantalla de la cláusula y el resultado generado

(Puede verse en la siguiente página)

33	ALTER SESSION SET CURRENT_SCHEMA = PARRANDEROS;	
34	SELECT	
35	table_name AS NombreTabla,	
36	data_type AS TipoDeDato,	
37	COUNT(column_name) AS NumColsTipoDato,	
38	ROUND(AVG(data_length), 2) AS PromedioLongitudCol	
39	FROM	
40	ALL_TAB_COLUMNS	
41	WHERE	
42	table_name IN ('BARES', 'BEBEDORES', 'BEBIDAS', 'FRECUELTAN', 'GUSTAN', 'SIRVEN', 'TIPO_BEBIDA')	
43	GROUP BY	
44	table_name, data_type	
45	ORDER BY	
46	table_name ASC,	
47	data_type ASC,	
48	COUNT(column_name) ASC;	

Salida de Script x	Resultado de la Consulta x	Resultado de la Consulta 1 x	Resultado de la Consulta 2 x	Resultado de la Consulta 3 x
SQL Todas las Filas Recuperadas: 15 en 0,39 segundos				
NOMBRETABLA	TIPODEDATO	NUMCOLSTIPODATO	PROMEDIOLONGITUDCOL	
1 BARES	NUMBER	2	22	
2 BARES	VARCHAR2	3	23,33	
3 BEBEDORES	NUMBER	1	22	
4 BEBEDORES	VARCHAR2	3	21,67	
5 BEBIDAS	NUMBER	3	22	
6 BEBIDAS	VARCHAR2	1	25	
7 FRECUENTAN	DATE	1	7	
8 FRECUENTAN	NUMBER	2	22	
9 FRECUENTAN	VARCHAR2	1	20	
10 GUSTAN	NUMBER	3	22	
11 GUSTAN	VARCHAR2	1	15	
12 SIRVEN	NUMBER	6	22	
13 SIRVEN	VARCHAR2	2	11,5	
14 TIPO_BEBIDA	NUMBER	1	22	

2.4 Requerimiento 4:

Listado general de las columnas y sus restricciones: Para todas las columnas de las tablas de Parranderos se debe mostrar el nombre de la tabla, el nombre de la columna, el tipo de dato de la columna, el nombre de la restricción y si permite nulos o no. El resultado debe estar ordenado de forma ascendente por el nombre de la tabla, el nombre de la columna y el nombre de la restricción. Para las columnas que no tienen restricciones, en el nombre de la restricción debe aparecer “NO TIENE”. Note que para las columnas que tienen más de una restricción, la información de la columna se repite.

Cláusula SQL utilizada para resolver el requerimiento

SELECT

tc.table_name AS NombreTabla,

```

tc.column_name AS NombreColumna,
tc.data_type AS TipoDeDato,
COALESCE(ucc.constraint_name,'NO TIENE') AS NombreRestriccion,
CASE
    WHEN tc.nullable = 'Y' THEN 'Sí'
    ELSE 'No'
END AS PermiteNulos
FROM
    ALL_TAB_COLUMNS tc
LEFT JOIN
    ALL_CONS_COLUMNS ucc ON tc.table_name = ucc.table_name AND tc.column_name
= ucc.column_name

WHERE
    tc.table_name IN ('BARES', 'BEBEDORES', 'BEBIDAS', 'FRECUENTAN', 'GUSTAN',
'SIRVEN', 'TIPO_BEBIDA')
ORDER BY
    tc.table_name ASC,
    tc.column_name ASC,
    NombreRestriccion ASC;

```


Captura de pantalla de la cláusula y el resultado generado

```

51  -- REQUERIMIENTO 4
52
53  SELECT
54      tc.table_name AS NombreTabla,
55      tc.column_name AS NombreColumna,
56      tc.data_type AS TipoDeDato,
57      COALESCE(ucc.constraint_name,'NO TIENE') AS NombreRestriccion,
58      CASE
59          WHEN tc.nullable = 'Y' THEN 'Sí'
60          ELSE 'No'
61      END AS PermiteNulos
62  FROM
63      ALL_TAB_COLUMNS tc
64  LEFT JOIN
65      ALL_CONS_COLUMNS ucc ON tc.table_name = ucc.table_name AND tc.column_name = ucc.column_name
66
67  WHERE
68      tc.table_name IN ('BARES', 'BEBEDORES', 'BEBIDAS', 'FRECUENTAN', 'GUSTAN', 'SIRVEN', 'TIPO_BEBIDA')
69  ORDER BY
70      tc.table_name ASC,
71      tc.column_name ASC,
72      NombreRestriccion ASC;
73

```

Salida de Script x Resultado de la Consulta x

 SQL | Todas las Filas Recuperadas: 46 en 0,063 segundos

	NOMBRETABLA	NOMBRECOLUMNA	TIPODEDATO	NOMBRERESTRICCION	PERMITENULOS
1	BARES	CANT_SEDES	NUMBER	CK_CANTSEDES	Si
2	BARES	CIUDAD	VARCHAR2	NO TIENE	Si
3	BARES	ID	NUMBER	PK_BARES	No
4	BARES	NOMBRE	VARCHAR2	NO TIENE	Si
5	BARES	PRESUPUESTO	VARCHAR2	CK_PRESUPUESTO	Si
6	BEBEDORES	CIUDAD	VARCHAR2	NO TIENE	Si
7	BEBEDORES	ID	NUMBER	PK_BEBEDORES	No
8	BEBEDORES	NOMBRE	VARCHAR2	NO TIENE	Si
9	BEBEDORES	PRESUPUESTO	VARCHAR2	CK_BEBEDORES_PRESUPUESTO	Si
10	BEBIDAS	GRADO_ALCOHOL	NUMBER	CK_GRADOALCHL	Si
11	BEBIDAS	ID	NUMBER	PK_BEBIDAS	No
12	BEBIDAS	NOMBRE	VARCHAR2	NO TIENE	Si
13	BEBIDAS	TIPO	NUMBER	FK_B_TIPO_BEBIDA	Si
14	FRECUELTAN	FECHA_ULTIMA_VISITA	DATE	NO TIENE	Si
15	FRECUELTAN	HORARIO	VARCHAR2	CK_HORARIO	No
16	FRECUELTAN	HORARIO	VARCHAR2	PK_FRECUELTAN	No
17	FRECUELTAN	ID_BAR	NUMBER	FK_F_BARES	No
18	FRECUELTAN	ID_BAR	NUMBER	PK_FRECUELTAN	No
19	FRECUELTAN	ID BEBEDOR	NUMBER	FK F BEBEDOR	No

2.5 Requerimiento 5:

Se quiere tener un listado del subconjunto de columnas de la base de datos de Parranderos, cuyo nombre no contiene la letra “R” y que tiene máximo 2 vocales, indicando para cada columna su nombre, tabla a la que pertenece y el número de restricciones que la afectan (por ejemplo, si una columna es parte de la llave primaria de la tabla y tiene una restricción de llave foránea, el resultado en esta columna debe ser mayor o igual a dos (2))

Este listado puede ser solicitado en cualquier momento y por lo tanto debe generarse de forma semiautomática, mediante el siguiente proceso:

a. Mediante UNA consulta al catálogo, obtener un listado donde, para cada columna de Parranderos que debe ser incluida en el resultado, hay una fila de respuesta.

Cada fila de respuesta de la consulta anterior contiene una sentencia SQL que permite encontrar la información solicitada

para la columna correspondiente, terminado con punto y coma (;).

Este listado contiene las sentencias SQL que se requieren para la segunda etapa.

b. Utilizar el listado de la etapa a) como un script, cuya ejecución devuelve un nuevo listado donde cada fila contiene la información solicitada de las columnas solicitadas de Parranderos.

Parte a)

Cláusula SQL utilizada para resolver el requerimiento

```
SELECT 'SELECT ''' || c.column_name || ''' AS Columna, ''' || c.table_name || ''' AS Tabla, COUNT(*)
```

```
AS Num_Restricciones FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name
```

```
AND acc.owner = ac.owner WHERE acc.table_name = ''' || c.table_name || ''' AND acc.column_name = '''
```

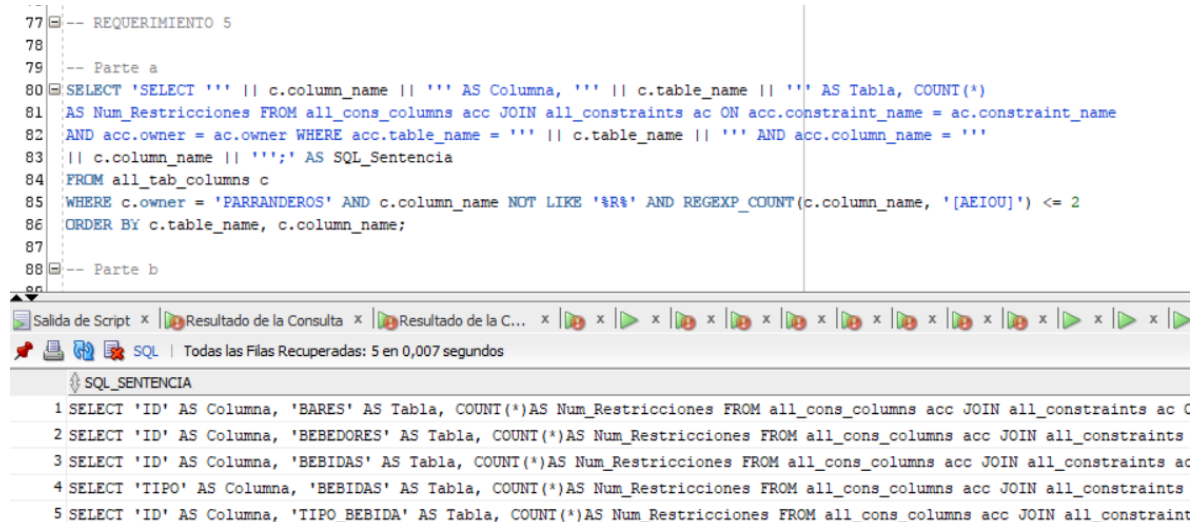
```
|| c.column_name || ''';' AS SQL_Sentencia
```

```
FROM all_tab_columns c
```

```
WHERE c.owner = 'PARRANDEROS' AND c.column_name NOT LIKE '%R%' AND REGEXP_COUNT(c.column_name, '[AEIOU]') <= 2
```

```
ORDER BY c.table_name, c.column_name;
```

Captura de pantalla de la cláusula y el resultado generado



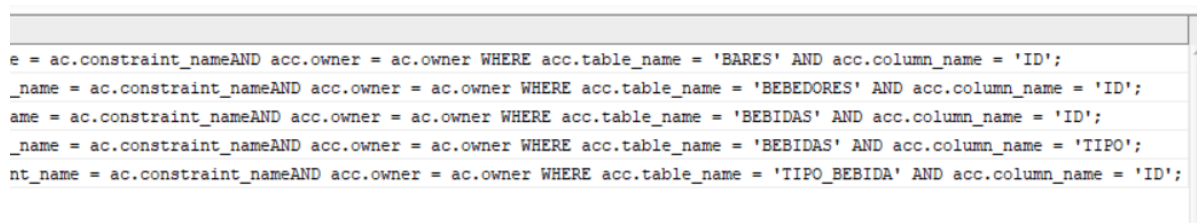
The screenshot shows a SQL IDE interface. The top pane displays a SQL query with line numbers 77 to 88. The query is a dynamic SQL statement that constructs a query to count the number of constraints for each column in the 'PARRANDEROS' schema, excluding columns containing the letter 'R' and having 3 or more vowels. The bottom pane shows the results of the query, which are 5 rows of SQL sentences. The first row is: 'SELECT 'ID' AS Columna, 'BARES' AS Tabla, COUNT(*)AS Num_Restricciones FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name AND acc.owner = ac.owner WHERE acc.table_name = 'BARES' AND acc.column_name = 'ID';'. The other rows follow a similar pattern for 'BEBEDORES', 'BEBIDAS', 'TIPO', and 'TIPO_BEBIDA'.

```
77 -- REQUERIMIENTO 5
78
79 -- Parte a
80 SELECT 'SELECT ''' || c.column_name || ''' AS Columna, ''' || c.table_name || ''' AS Tabla, COUNT(*)
81 AS Num_Restricciones FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name
82 AND acc.owner = ac.owner WHERE acc.table_name = ''' || c.table_name || ''' AND acc.column_name = '''
83 || c.column_name || ''';' AS SQL_Sentencia
84 FROM all_tab_columns c
85 WHERE c.owner = 'PARRANDEROS' AND c.column_name NOT LIKE '%R%' AND REGEXP_COUNT(c.column_name, '[AEIOU]') <= 2
86 ORDER BY c.table_name, c.column_name;
87
88 -- Parte b
```

SQL_SENTENCIA

```
1 SELECT 'ID' AS Columna, 'BARES' AS Tabla, COUNT(*)AS Num_Restricciones FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name AND acc.owner = ac.owner WHERE acc.table_name = 'BARES' AND acc.column_name = 'ID';
2 SELECT 'ID' AS Columna, 'BEBEDORES' AS Tabla, COUNT(*)AS Num_Restricciones FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name AND acc.owner = ac.owner WHERE acc.table_name = 'BEBEDORES' AND acc.column_name = 'ID';
3 SELECT 'ID' AS Columna, 'BEBIDAS' AS Tabla, COUNT(*)AS Num_Restricciones FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name AND acc.owner = ac.owner WHERE acc.table_name = 'BEBIDAS' AND acc.column_name = 'ID';
4 SELECT 'TIPO' AS Columna, 'BEBIDAS' AS Tabla, COUNT(*)AS Num_Restricciones FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name AND acc.owner = ac.owner WHERE acc.table_name = 'BEBIDAS' AND acc.column_name = 'TIPO';
5 SELECT 'ID' AS Columna, 'TIPO_BEBIDA' AS Tabla, COUNT(*)AS Num_Restricciones FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name AND acc.owner = ac.owner WHERE acc.table_name = 'TIPO_BEBIDA' AND acc.column_name = 'ID';
```

Terminación de las respuestas



The screenshot shows a SQL IDE interface with a single line of SQL code. The code is a dynamic SQL statement that constructs a query to count the number of constraints for each column in the 'PARRANDEROS' schema, excluding columns containing the letter 'R' and having 3 or more vowels. The code is: 'SELECT 'ID' AS Columna, 'BARES' AS Tabla, COUNT(*)AS Num_Restricciones FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name AND acc.owner = ac.owner WHERE acc.table_name = 'BARES' AND acc.column_name = 'ID';'. The code is highlighted in blue.

```
e = ac.constraint_nameAND acc.owner = ac.owner WHERE acc.table_name = 'BARES' AND acc.column_name = 'ID';
```


Parte b)

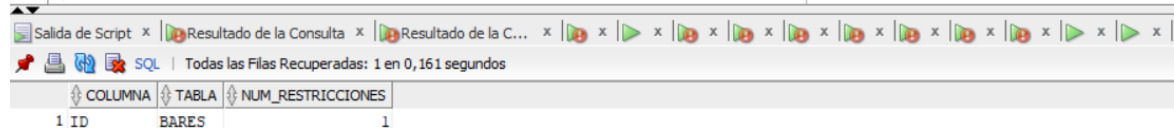
Para la parte b, de forma manual se copió y pegó cada uno de los resultados obtenidos en cada fila de la anterior consulta en espacios aparte para ejecutarlos de forma individual cuando se requiera.

```
88 -- Parte b
89
90 -- Primera fila
91 SELECT 'ID' AS Columna, 'BARES' AS Tabla, COUNT(*) AS Num_Restricciones
92 FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name AND acc.owner = ac.owner
93 WHERE acc.table_name = 'BARES' AND acc.column_name = 'ID';
94 -- Segunda fila
95 SELECT 'ID' AS Columna, 'BEBEDORES' AS Tabla, COUNT(*) AS Num_Restricciones
96 FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name AND acc.owner = ac.owner
97 WHERE acc.table_name = 'BEBEDORES' AND acc.column_name = 'ID';
98 -- Tercera fila
99 SELECT 'ID' AS Columna, 'BEBIDAS' AS Tabla, COUNT(*) AS Num_Restricciones
100 FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name
101 WHERE acc.table_name = 'BEBIDAS' AND acc.column_name = 'ID';
102 -- Cuarta fila
103 SELECT 'TIPO' AS Columna, 'BEBIDAS' AS Tabla, COUNT(*) AS Num_Restricciones
104 FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name
105 WHERE acc.table_name = 'BEBIDAS' AND acc.column_name = 'TIPO';
106 -- Quinta fila
107 SELECT 'ID' AS Columna, 'TIPO_BEBIDA' AS Tabla, COUNT(*) AS Num_Restricciones
108 FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name AND acc.owner = ac.owner
109 WHERE acc.table_name = 'TIPO_BEBIDA' AND acc.column_name = 'ID';
```

Por ejemplo, el resultado de ejecutar las primera y cuarta filas son los siguientes:

Resultado ejecución primera fila

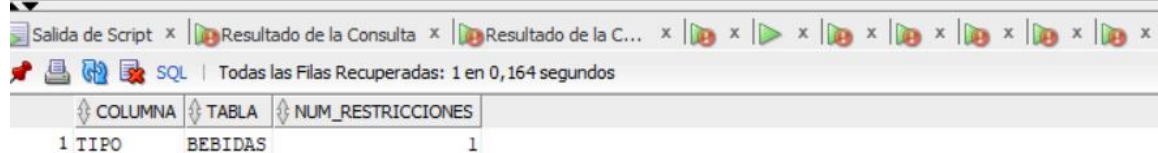
```
-- Primera fila
91 SELECT 'ID' AS Columna, 'BARES' AS Tabla, COUNT(*) AS Num_Restricciones
92 FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name AND acc.owner = ac.owner
93 WHERE acc.table_name = 'BARES' AND acc.column_name = 'ID';
94 -- Segunda fila
```



COLUMNA	TABLA	NUM_RESTRICCIONES
1 ID	BARES	1

Resultado ejecución cuarta fila

```
102 -- Cuarta fila
103 SELECT 'TIPO' AS Columna, 'BEBIDAS' AS Tabla, COUNT(*) AS Num_Restricciones
104 FROM all_cons_columns acc JOIN all_constraints ac ON acc.constraint_name = ac.constraint_name
105 WHERE acc.table_name = 'BEBIDAS' AND acc.column_name = 'TIPO';
106 -- Quinta fila
```



COLUMNA	TABLA	NUM_RESTRICCIONES
1 TIPO	BEBIDAS	1

3 Comentario final

Los archivos correspondientes a los scripts contruidos para cada requerimiento fueron adjuntados al repositorio junto al presente documento en la carpeta del Taller 2, parte 2. Ante cualquier duda o dificultad para visualizar el contenido, por favor comunicarse con el equipo por medio de los correos suministrados.