

“Documento de reporte | Taller #2 - SQL Parte 1”

José D. Flórez Ruiz, Carlos M. Muñoz Almeida
Universidad de los Andes, Bogotá, Colombia
{jd.florezr1, c.munoza}@uniandes.edu.co
Fecha de presentación: Marzo 19 de 2023

Tabla de contenido:

1	Consultas al Catálogo de la Base de Datos	1
1.1	Información general estadística de tablas	1
1.2	Información de las llaves primarias	2
1.3	Caracterización de los tipos de las tablas	3
1.4	Listado general de las columnas y sus restricciones	4
2	Consultas Sobre Catalogo y Datos	5

1 Consultas al Catálogo de la Base de Datos

1.1 Información general estadística de tablas

Descripción: 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 el número de columnas con restricción de llave foránea, ordenado por el nombre de la tabla. En caso de que la tabla no tenga llaves foráneas y/o columnas con restricciones de nulidad se debe marcar 0 en el campo respectivo.

Clausula SQL en texto plano:

```
SELECT NOM_TCOLUM, COALESCE(NumColumnas,0) AS NUMCOLUMNS,  
COALESCE(NumColsNull,0) AS NUMCOLSNULL, COALESCE(NumColsFks,0) AS  
NUMCOLSFKS  
  
FROM((SELECT TABLE_NAME AS NOM_TCOLUM, COUNT(TABLE_NAME) AS NumColumnas  
  
FROM ALL_TAB_COLUMNS  
  
WHERE owner = 'PARRANDEROS'  
  
GROUP BY TABLE_NAME) LEFT OUTER JOIN  
  
(SELECT TABLE_NAME AS NOM_TNULL, COUNT(NULLABLE) AS NumColsNull  
  
FROM ALL_TAB_COLUMNS  
  
WHERE owner = 'PARRANDEROS' AND NULLABLE = 'Y'  
  
GROUP BY TABLE_NAME) ON NOM_TCOLUM = NOM_TNULL) LEFT OUTER JOIN  
  
(SELECT TABLE_NAME AS NOM_TFKS, COUNT(CONSTRAINT_TYPE) AS  
NumColsFks
```

```

FROM ALL_CONSTRAINTS

WHERE owner = 'PARRANDEROS' AND CONSTRAINT_TYPE = 'R'

GROUP BY TABLE_NAME) ON NOM_TCOLUMN = NOM_TFKS

GROUP BY NOM_TCOLUMN, NumColumnas, NumColsNull, NumColsFks

ORDER BY NOM_TCOLUMN ASC;

```

Clausula SQL en foto:

```

SELECT NOM_TCOLUMN, COALESCE(NumColumnas,0) AS NUMCOLUMNS, COALESCE(NumColsNull,0) AS NUMCOLSNULL, COALESCE(NumColsFks,0) AS NUMCOLSFKS
FROM((SELECT TABLE_NAME AS NOM_TCOLUMN, COUNT(TABLE_NAME) AS NumColumnas
FROM ALL_TAB_COLUMNS
WHERE owner = 'PARRANDEROS'
GROUP BY TABLE_NAME) LEFT OUTER JOIN
(SELECT TABLE_NAME AS NOM_TNULL, COUNT(NULLABLE) AS NumColsNull
FROM ALL_TAB_COLUMNS
WHERE owner = 'PARRANDEROS' AND NULLABLE = 'Y'
GROUP BY TABLE_NAME) ON NOM_TCOLUMN = NOM_TNULL) LEFT OUTER JOIN
(SELECT TABLE_NAME AS NOM_TFKS, COUNT(CONSTRAINT_TYPE) AS NumColsFks
FROM ALL_CONSTRAINTS
WHERE owner = 'PARRANDEROS' AND CONSTRAINT_TYPE = 'R'
GROUP BY TABLE_NAME) ON NOM_TCOLUMN = NOM_TFKS
GROUP BY NOM_TCOLUMN, NumColumnas, NumColsNull, NumColsFks
ORDER BY NOM_TCOLUMN ASC;

```

Resultados obtenidos de clausula SQL:

	NOM_TCOLUMN	NUMCOLUMNS	NUMCOLSNULL	NUMCOLSFKS
1	BARES	5	4	0
2	BEBEDORES	4	3	0
3	BEBIDAS	4	3	1
4	FRECUELTAN	4	1	2
5	GUSTAN	2	2	2
6	SIRVEN	3	0	2
7	TIPO_BEBIDA	2	1	0

1.2 Información de las llaves primarias

Descripción: 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. El formato esperado se presenta a continuación.

Clausula SQL en texto plano:

```

SELECT c.table_name AS NombreTabla, cc.column_name AS NombreColsPK,
tc.data_type AS TipoDeDato

FROM ALL_CONSTRAINTS c

JOIN ALL_CONS_COLUMNS cc ON c.owner = cc.owner AND c.constraint_name =
cc.constraint_name

JOIN ALL_TAB_COLUMNS tc ON cc.owner = tc.owner AND cc.table_name =
tc.table_name AND cc.column_name = tc.column_name

```

```
WHERE c.owner = 'PARRANDEROS'

AND c.constraint_type = 'P'

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

Clausula SQL en foto:

```
SELECT c.table_name AS NombreTabla, cc.column_name AS NombreColsPK, tc.data_type AS TipoDeDato
FROM ALL_CONSTRAINTS c
JOIN ALL_CONS_COLUMNS cc ON c.owner = cc.owner AND c.constraint_name = cc.constraint_name
JOIN ALL_TAB_COLUMNS tc ON cc.owner = tc.owner AND cc.table_name = tc.table_name AND cc.column_name = tc.column_name
WHERE c.owner = 'PARRANDEROS'
AND c.constraint_type = 'P'
ORDER BY c.table_name ASC, cc.column_name ASC;
```

Resultados obtenidos de clausula SQL:

	◊ NOMBRETABLA	◊ NOMBRECOSP	◊ TIPODEDATO
1	BARES	ID	NUMBER
2	BEBEDORES	ID	NUMBER
3	BEBIDAS	ID	NUMBER
4	FRECUELTAN	HORARIO	VARCHAR2
5	FRECUELTAN	ID_BAR	NUMBER
6	FRECUELTAN	ID_BEBEDOR	NUMBER
7	GUSTAN	ID_BEBEDOR	NUMBER
8	GUSTAN	ID_BEBIDA	NUMBER
9	SIRVEN	HORARIO	VARCHAR2
10	SIRVEN	ID_BAR	NUMBER
11	SIRVEN	ID_BEBIDA	NUMBER
12	TIPO_BEBIDA	ID	NUMBER

1.3 Caracterización de los tipos de las tablas

Descripción: Caracterización de los tipos de datos de las tablas: Para todas las tablas de la aplicación 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.

Clausula SQL en texto plano:

```
SELECT tc.table_name AS NombreDeTabla, tc.data_type AS TipoDeDato,
COUNT(tc.column_name) AS NumColsTipoDato, ROUND(AVG(tc.avg_col_len), 2) AS
PromedioLongitudCol

FROM ALL_TAB_COLUMNS tc

WHERE tc.owner = 'PARRANDEROS'

GROUP BY tc.table_name, tc.data_type

ORDER BY tc.table_name ASC, tc.data_type ASC, num_columns ASC;
```

Clausula SQL en foto:

```
SELECT tc.table_name AS NombreDeTabla, tc.data_type AS TipoDeDato, COUNT(tc.column_name) AS NumColsTipoDato, ROUND(AVG(tc.avg_col_len), 2) AS PromedioLongitudCol
FROM ALL_TAB_COLUMNS tc
WHERE tc.owner = 'PARRANDEROS'
GROUP BY tc.table_name, tc.data_type
ORDER BY tc.table_name ASC, tc.data_type ASC, num_columns ASC;
```

Resultados obtenidos de clausula SQL:

	NOMBREDETABLA	TIPODEDATO	NUMCOLSTIPODATO	PROMEDIOLONGITUDCOL
1	BARES	NUMBER	2	3
2	BARES	VARCHAR2	3	10.33
3	BEBEDORES	NUMBER	1	4
4	BEBEDORES	VARCHAR2	3	10.67
5	BEBIDAS	NUMBER	3	3.33
6	BEBIDAS	VARCHAR2	1	11
7	FRECUELTAN	DATE	1	8
8	FRECUELTAN	NUMBER	2	3.5
9	FRECUELTAN	VARCHAR2	1	8
10	GUSTAN	NUMBER	2	4
11	SIRVEN	NUMBER	2	3.5
12	SIRVEN	VARCHAR2	1	8
13	TIPO_BEBIDA	NUMBER	1	3
14	TIPO_BEBIDA	VARCHAR2	1	9

1.4 Listado general de las columnas y sus restricciones

Descripción: 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.

Clausula SQL en texto plano:

```
SELECT tc.table_name AS NombreTabla, tc.column_name AS NombreColumna,
tc.data_type AS TipoDeDato, COALESCE (c.constraint_name, 'NO TIENE') AS
NombreRestriccion, NULLABLE AS PermiteNulos
```

```
FROM ALL_TAB_COLUMNS tc
```

```
LEFT JOIN ALL_CONS_COLUMNS cc ON tc.owner = cc.owner AND tc.table_name =
cc.table_name AND tc.column_name = cc.column_name
```

```
LEFT JOIN ALL_CONSTRAINTS c ON cc.owner = c.owner AND cc.constraint_name =
c.constraint_name
```

```
WHERE tc.owner = 'PARRANDEROS'
```

```
ORDER BY tc.table_name ASC, tc.column_name ASC, NombreRestriccion ASC;
```

Clausula SQL en foto:

```

SELECT tc.table_name AS NombreTabla, tc.column_name AS NombreColumna, tc.data_type AS TipoDeDato, COALESCE (c.constraint_name, 'NO TIENE') AS NombreRestriccion, NULLABLE AS PermiteNulos
FROM ALL_TAB_COLUMNS tc
LEFT JOIN ALL_CONS_COLUMNS cc ON tc.owner = cc.owner AND tc.table_name = cc.table_name AND tc.column_name = cc.column_name
LEFT JOIN ALL_CONSTRAINTS c ON cc.owner = c.owner AND cc.constraint_name = c.constraint_name
WHERE tc.owner = 'PARRANDEROS'
ORDER BY tc.table_name ASC, tc.column_name ASC, NombreRestriccion ASC;

```

Resultados obtenidos de clausula SQL:

NOMBRE TABLA	NOMBRE COLUMNA	TIPO DE DATO	NOMBRE RESTRICCIÓN	PERMITE NULOS
1 BARES	CANT_SEDES	NUMBER	CK_CANTSEDES	Y
2 BARES	CIUDAD	VARCHAR2	NO TIENE	Y
3 BARES	ID	NUMBER	PK_BARES	N
4 BARES	NOMBRE	VARCHAR2	NO TIENE	Y
5 BARES	PRESUPUESTO	VARCHAR2	CK_PRESUPUESTO	Y
6 BEBEDORES	CIUDAD	VARCHAR2	NO TIENE	Y
7 BEBEDORES	ID	NUMBER	PK_BEBEDORES	N
8 BEBEDORES	NOMBRE	VARCHAR2	NO TIENE	Y
9 BEBEDORES	PRESUPUESTO	VARCHAR2	CK_BEBEDORES_PRESUPUESTO	Y
10 BEBIDAS	GRADO_ALCOHOL	NUMBER	CK_GRADOALCHL	Y
11 BEBIDAS	ID	NUMBER	PK_BEBIDAS	N
12 BEBIDAS	NOMBRE	VARCHAR2	NO TIENE	Y
13 BEBIDAS	TIPO	NUMBER	FK B TIPO BEBIDA	Y

2 Consultas Sobre Catalogo y Datos

La siguiente consulta genera un listado donde cada fila de respuesta contiene una sentencia SQL que permite encontrar la información solicitada para la columna buscada:

Clausula SQL en texto plano:

```

SELECT DISTINCT 'SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNA, TC.TABLE_NAME AS
TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONES
FROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
ALL_CONS_COLUMNS.TABLE_NAME, COUNT(CONSTRAINT_NAME) AS CUENTA_REST
FROM ALL_CONS_COLUMNS
WHERE owner = ''PARRANDEROS''
GROUP BY COLUMN_NAME, TABLE_NAME ) CC ON
TC.COLUMN_NAME = CC.COLUMN_NAME
WHERE TC.OWNER = ''PARRANDEROS'' AND TC.COLUMN_NAME = ''' || COLUMN_NAME ||
''' GROUP BY TC.COLUMN_NAME, TC.TABLE_NAME , CC.CUENTA_REST;' AS
SENTENCIA
FROM ALL_TAB_COLUMNS
WHERE OWNER = 'PARRANDEROS' AND LENGTH(ALL_TAB_COLUMNS.COLUMN_NAME) > 6;

```

```

SELECT DISTINCT 'SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONES
FROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME, ALL_CONS_COLUMNS.TABLE_NAME, COUNT(CONSTRAINT_NAME) AS CUENTA_REST
FROM ALL_CONS_COLUMNS
WHERE owner = 'PARRANDEROS'
GROUP BY COLUMN_NAME, TABLE_NAME ) CC ON TC.COLUMN_NAME = CC.COLUMN_NAME
WHERE TC.OWNER = 'PARRANDEROS' AND TC.COLUMN_NAME = '' || COLUMN_NAME ||
''' GROUP BY TC.COLUMN_NAME, TC.TABLE_NAME , CC.CUENTA_REST;' AS SENTENCIA
FROM ALL_TAB_COLUMNS
WHERE OWNER = 'PARRANDEROS' AND LENGTH(ALL_TAB_COLUMNS.COLUMN_NAME) > 6;

```

Esta es la respuesta generada:

```

-- SENTENCIA
1 SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONESFROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
2 SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONESFROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
3 SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONESFROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
4 SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONESFROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
5 SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONESFROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
6 SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONESFROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
7 SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONESFROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,

```

Estas son todas las sentencias generadas:

```

SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS
TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONES
FROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
ALL_CONS_COLUMNS.TABLE_NAME, COUNT(CONSTRAINT_NAME) AS CUENTA_REST
FROM ALL_CONS_COLUMNS
WHERE owner = 'PARRANDEROS'
GROUP BY COLUMN_NAME, TABLE_NAME ) CC ON
TC.COLUMN_NAME = CC.COLUMN_NAME
WHERE TC.OWNER = 'PARRANDEROS' AND TC.COLUMN_NAME = 'ID_BEBEDOR' GROUP BY
TC.COLUMN_NAME, TC.TABLE_NAME , CC.CUENTA_REST;

```

```

SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS
TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONES
FROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
ALL_CONS_COLUMNS.TABLE_NAME, COUNT(CONSTRAINT_NAME) AS CUENTA_REST
FROM ALL_CONS_COLUMNS
WHERE owner = 'PARRANDEROS'
GROUP BY COLUMN_NAME, TABLE_NAME ) CC ON
TC.COLUMN_NAME = CC.COLUMN_NAME
WHERE TC.OWNER = 'PARRANDEROS' AND TC.COLUMN_NAME = 'PRESUPUESTO' GROUP BY
TC.COLUMN_NAME, TC.TABLE_NAME , CC.CUENTA_REST;

```

```

SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS
TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONES
FROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
ALL_CONS_COLUMNS.TABLE_NAME, COUNT(CONSTRAINT_NAME) AS CUENTA_REST
FROM ALL_CONS_COLUMNS
WHERE owner = 'PARRANDEROS'
GROUP BY COLUMN_NAME, TABLE_NAME ) CC ON
TC.COLUMN_NAME = CC.COLUMN_NAME
WHERE TC.OWNER = 'PARRANDEROS' AND TC.COLUMN_NAME = 'ID_BEBIDA' GROUP BY
TC.COLUMN_NAME, TC.TABLE_NAME , CC.CUENTA_REST;

```

```

SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS
TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONES
FROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
ALL_CONS_COLUMNS.TABLE_NAME, COUNT(CONSTRAINT_NAME) AS CUENTA_REST
FROM ALL_CONS_COLUMNS
WHERE owner = 'PARRANDEROS'
GROUP BY COLUMN_NAME, TABLE_NAME ) CC ON
TC.COLUMN_NAME = CC.COLUMN_NAME
WHERE TC.OWNER = 'PARRANDEROS' AND TC.COLUMN_NAME = 'FECHA_ULTIMA_VISITA'
GROUP BY TC.COLUMN_NAME, TC.TABLE_NAME , CC.CUENTA_REST;

```

```

SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS
TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONES
FROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
ALL_CONS_COLUMNS.TABLE_NAME, COUNT(CONSTRAINT_NAME) AS CUENTA_REST
FROM ALL_CONS_COLUMNS
WHERE owner = 'PARRANDEROS'
GROUP BY COLUMN_NAME, TABLE_NAME ) CC ON
TC.COLUMN_NAME = CC.COLUMN_NAME
WHERE TC.OWNER = 'PARRANDEROS' AND TC.COLUMN_NAME = 'CANT_SEDES' GROUP BY
TC.COLUMN_NAME, TC.TABLE_NAME , CC.CUENTA_REST;

```

```

SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS
TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONES
FROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
ALL_CONS_COLUMNS.TABLE_NAME, COUNT(CONSTRAINT_NAME) AS CUENTA_REST
FROM ALL_CONS_COLUMNS
WHERE owner = 'PARRANDEROS'
GROUP BY COLUMN_NAME, TABLE_NAME ) CC ON
TC.COLUMN_NAME = CC.COLUMN_NAME
WHERE TC.OWNER = 'PARRANDEROS' AND TC.COLUMN_NAME = 'GRADO_ALCOHOL' GROUP
BY TC.COLUMN_NAME, TC.TABLE_NAME , CC.CUENTA_REST;

```

```

SELECT TC.COLUMN_NAME AS NOMBRE_COLUMNNA, TC.TABLE_NAME AS
TABLA_PERTENECIENTE, COALESCE(CC.CUENTA_REST,0) AS NUM_RESTRICCIONES
FROM ALL_TAB_COLUMNS TC LEFT OUTER JOIN (SELECT COLUMN_NAME,
ALL_CONS_COLUMNS.TABLE_NAME, COUNT(CONSTRAINT_NAME) AS CUENTA_REST
FROM ALL_CONS_COLUMNS
WHERE owner = 'PARRANDEROS'
GROUP BY COLUMN_NAME, TABLE_NAME ) CC ON
TC.COLUMN_NAME = CC.COLUMN_NAME
WHERE TC.OWNER = 'PARRANDEROS' AND TC.COLUMN_NAME = 'HORARIO' GROUP BY
TC.COLUMN_NAME, TC.TABLE_NAME , CC.CUENTA_REST;

```

Estas son las respuestas generadas:

	NOMBRE_COLUMNNA	TABLA_PERTENECIENTE	NUM_RESTRICCIONES
1	ID_BEBEDOR	FRECUELTAN	2
2	ID_BEBEDOR	GUSTAN	2

	NOMBRE_COLUMNNA	TABLA_PERTENECIENTE	NUM_RESTRICCIONES
1	PRESUPUESTO	BARES	1
2	PRESUPUESTO	BEDEDORES	1

	NOMBRE_COLUMNNA	TABLA_PERTENECIENTE	NUM_RESTRICCIONES
1	ID_BEBIDA	SIRVEN	2
2	ID_BEBIDA	GUSTAN	2

	NOMBRE_COLUMNNA	TABLA_PERTENECIENTE	NUM_RESTRICCIONES
1	FECHA_ULTIMA_VISITA	FRECUELTAN	0

	NOMBRE_COLUMNNA	TABLA_PERTENECIENTE	NUM_RESTRICCIONES
1	CANT_SEDES	BARES	1

	NOMBRE_COLUMNNA	TABLA_PERTENECIENTE	NUM_RESTRICCIONES
1	GRADO_ALCOHOL	BEEDIDAS	1

	NOMBRE_COLUMNNA	TABLA_PERTENECIENTE	NUM_RESTRICCIONES
1	HORARIO	FRECUELTAN	2
2	HORARIO	SIRVEN	2