

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
LABORATORIO DE BASE DE DATOS  
SEMESTRE 2020 - 2

PRÁCTICA 11  
Y  
PRÁCTICA 11 COMPLEMENTARIA:  
FUNCIONES DE AGREGACIÓN Y SUBCONSULTAS

*ALUMNOS:*

Murrieta Villegas Alfonso  
Valdespino Mendieta Joaquín

*PROFESOR:*

Ing. Jorge Alberto Rodríguez Campos

*GRUPO TEORÍA: 3*

*GRUPO DE LABORATORIO: 5*

## Objetivo

- El alumno pondrá en práctica el uso de las funciones de agregación empleando las instrucciones group by, having, realizará consultas SQL que incluyan los distintos tipos de subconsultas.
- Entender y reafirmar los conceptos aprendidos en clase, asociados con el diseño de sentencias SQL tipo select aplicadas en la explotación de datos para un caso de estudio. El enfoque de esta práctica comprende 2 principales conceptos: funciones de agregación y uso de subconsultas..

## Introducción

Las funciones de agregación que son funciones que realizan cálculos o estadísticas sobre algún conjunto de datos. El SQL estándar define una gran variedad de estas funciones. Las más comunes se describen a continuación:

Función	Descripción
count	Cuenta el número de registros
min	Obtiene el valor menor de un atributo
max	Obtiene el valor mayor de un atributo
sum	Realiza la suma aritmética de un conjunto de valores
avg	Calcula el promedio de un conjunto de valores.

### Group by

Se utiliza normalmente cuando en la lista de columnas en la instrucción select existen funciones de agregación, se emplean para obtener información como la siguiente:

- Determinar el precio mínimo para cada tipo de producto
- Generar el precio promedio de los productos de un determinado código.

Como su nombre lo indica, GROUP BY se aplica para realizar cálculos sobre grupos de datos. Estos cálculos los realiza una función de agregación.

## Desarrollo Práctica

### Práctica - Complementaria

Para este apartado se incluye capturas de pantalla de los códigos asociados a cada una de las consultas

#### 1) Consulta

```
--Profesor
Prompt Tabla1
create table consulta_1 as
select count(*) as num_articulos, sum(precio_venta) as ingresos
from subasta_venta sv
  right join articulo a
    on sv.articulo_id = a.articulo_id
  join subasta s
    on a.subasta_id = s.subasta_id
where to_char(fecha_inicio, 'yyyy') = '2010';
```

#### 2) Consulta

```
Prompt Tabla2
create table consulta_2 as
select count(*) as articulos_no_vendidos
from subasta s, articulo a, subasta_venta sv
where s.subasta_id = a.subasta_id
      and a.articulo_id = sv.articulo_id (+)
      and sv.articulo_id is null
      and to_char(fecha_inicio, 'yyyy') = '2010';
```

### 3) Consulta

```
Prompt Tabla3
create table consulta_3 as
select min(a.precio_inicial) pi_mas_barato, max(precio_inicial) pi_mas_caro,
       min(precio_venta) as pv_mas_barato, max(precio_venta) pv_mas_caro
from subasta s
     join articulo a
       on s.subasta_id = a.subasta_id
     join subasta_venta sv
       on a.articulo_id = sv.articulo_id
where s.nombre = 'EXPO-MAZATLAN';
```

### 4) Consulta

```
Prompt Tabla4
create table consulta_4 as
select c.cliente_id, c.email, tc.numero_tarjeta
from tarjeta_cliente tc, cliente c, subasta_venta sv
where tc.cliente_id = c.cliente_id
      and c.cliente_id = sv.cliente_id (+)
      and sv.subasta_venta_id is null
      and to_number(to_char(c.fecha_nacimiento, 'yyyy')) between 1970 and 1975;
```

### 5) Consulta

```
--Profesor
Prompt Tabla5
create table consulta_5 as
select count(*) as num_articulos, a.tipo_articulo, s.clave
from articulo a, status_articulo s
where a.status_articulo_id = s.status_articulo_id
      and s.clave in ('ENTREGADO', 'VENDIDO')
group by a.tipo_articulo, s.clave;
```

### 6) Consulta

```

Prompt Tabla6
create table consulta_6 as
select s.nombre, s.fecha_inicio, s.lugar, a.tipo_articulo,
       sum(sv.precio_venta) as total_recaudado
from subasta_venta sv
join articulo a
  on sv.articulo_id = a.articulo_id
join subasta s
  on a.subasta_id = s.subasta_id
where to_char(s.fecha_inicio, 'yyyy') = '2009'
group by s.nombre, s.fecha_inicio, s.lugar, a.tipo_articulo;
|

```

## 7) Consulta

```

--Profesor
Prompt Tabla7
create table consulta_7 as
select c.cliente_id, c.nombre, c.apellido_paterno, c.apellido_materno,
       count(*) as articulos_comprados, sum(sv.precio_venta) as monto_total
from cliente c
join subasta_venta sv
  on c.cliente_id = sv.cliente_id
group by c.cliente_id, c.nombre, c.apellido_paterno, c.apellido_materno
having count(*) > 5
union
select c.cliente_id, c.nombre, c.apellido_paterno, c.apellido_materno,
       count(*) as articulos_comprados, sum(sv.precio_venta) as monto_total
from cliente c
join subasta_venta sv
  on c.cliente_id = sv.cliente_id
group by c.cliente_id, c.nombre, c.apellido_paterno, c.apellido_materno
having sum(sv.precio_venta) > 3000000;

```

## 8) Consulta

```

Prompt Tabla8
create table consulta_8 as
select q.subasta_id, q.nombre, q.fecha_inicio, a.nombre as nombre_articulo,
       a.clave_articulo, q.mas_caro
from subasta_venta sv, articulo a, (
  select s.subasta_id, s.nombre, s.fecha_inicio,
         max(sv.precio_venta) as mas_caro
  from subasta_venta sv, articulo a, subasta s
  where sv.articulo_id = a.articulo_id
        and a.subasta_id = s.subasta_id
        and to_char(s.fecha_inicio, 'mm') in ('01', '03', '06')
        and to_char(s.fecha_inicio, 'yyyy') = '2010'
  group by s.subasta_id, s.nombre, s.fecha_inicio
) q
where sv.articulo_id = a.articulo_id
      and a.subasta_id = q.subasta_id
      and sv.precio_venta = q.mas_caro;

```

## 9) Consulta

```

Prompt Tabla9
create table consulta_9 as
select sum(sv.precio_venta) as total
from factura_cliente fc, subasta_venta sv,
(
select c.cliente_id, max(fecha_factura) as fecha_factura
from cliente c, subasta_venta sv, factura_cliente fc
where c.cliente_id = sv.cliente_id
and sv.factura_cliente_id = fc.factura_cliente_id
and nombre = 'GALILEA'
and apellido_paterno = 'GOMEZ'
and apellido_materno = 'GONZALEZ'
group by c.cliente_id
) q
where fc.factura_cliente_id = sv.factura_cliente_id
and sv.cliente_id = q.cliente_id
and fc.fecha_factura = q.fecha_factura;

```

#### 10) Consulta

```

Prompt Tabla10
create table consulta_10 as
select s.subasta_id, s.nombre, count(*) as num_articulos_vendidos
from subasta s, articulo a, subasta_venta sv
where s.subasta_id = a.subasta_id
and a.articulo_id = sv.articulo_id
and to_char(s.fecha_fin, 'yyyy') = '2010'
group by s.subasta_id, s.nombre
having count(*) > 3
order by s.subasta_id;

```

#### 11) Consulta

```

--Alfonso
Prompt Tabla11
create table consulta_11 as
select s.subasta_id, fecha_inicio, a.articulo_id, a.nombre, precio_inicial,
(
select avg(precio_inicial) as promedio
from articulo a
join subasta s
on a.subasta_id = s.subasta_id
where a.nombre like '%Motocicleta%'
and to_char(fecha_inicio, 'YYYY')='2010'
) as promedio
from subasta s
join articulo a
on s.subasta_id = a.subasta_id
where fecha_inicio between to_date('01/07/2010', 'dd/mm/yyyy')
and to_date('31/07/2010', 'dd/mm/yyyy')
and a.nombre like '%Motocicleta%'
and ( status_articulo_id=3
or status_articulo_id=4
)
order by a.articulo_id desc;

```

## 12) Consulta

```
Prompt Tabla12
create table consulta_12 as
select p.pais_id, p.clave, p.descripcion
from pais p, articulo_donado ad, articulo a
where p.pais_id = ad.pais_id
      and ad.articulo_id = a.articulo_id
      and a.precio_inicial > 300000
group by p.pais_id, p.clave, p.descripcion
having count(*) >= 3;
```

## 13) Consulta

```
Prompt Tabla13
create table consulta_13 as
select s.subasta_id, s.nombre, s.fecha_inicio,
       sum(sv.precio_venta) as total_ventas
from subasta s, articulo a, subasta_venta sv
where s.subasta_id = a.subasta_id
      and a.articulo_id = sv.articulo_id
      and to_char(s.fecha_inicio, 'yyyy') = '2010'
group by s.subasta_id, s.nombre, s.fecha_inicio
having sum(sv.precio_venta) >= 3000000;
```

## 14) Consulta

```
Prompt Tabla14
create table consulta_14 as
select c.nombre, c.apellido_paterno, c.apellido_materno,
       sum(sv.precio_venta) as total_venta
from cliente c, subasta_venta sv
where c.cliente_id = sv.cliente_id
      and sv.factura_cliente_id is null
group by c.nombre, c.apellido_paterno, c.apellido_materno
having sum(sv.precio_venta) >= 1000000;
```

## 15) Consulta

```
Prompt Tabla15
create table consulta_15 as
select s.*
from subasta s,(
  select s.subasta_id, count(*) as maximo
  from subasta s, articulo a, subasta_venta sv
  where s.subasta_id = a.subasta_id
    and a.articulo_id = sv.articulo_id
  group by s.subasta_id
  having count(*) = (
    select max(count(*)) as maximo
    from subasta s, articulo a, subasta_venta sv
    where s.subasta_id = a.subasta_id
      and a.articulo_id = sv.articulo_id
    group by s.subasta_id
  )
) q
where s.subasta_id = q.subasta_id;
```

A continuación se muestran los resultados finales al evaluar las consultas anteriores con el validador de esta práctica:

### Validador - Alfonso Murrieta Villegas

```
Complementaria: rlwrap
=====
Iniciando proceso de validación de respuestas
Incluir en el reporte a partir de este punto
=====
SESSION_TIME: 18/05/2020 19:02:44
USR_COD HOST: 252776
OS_USER:      murry
BD_USER:      MUVA_P1101_SUBASTAS
CON_NAME:     cursobd
c5761c335366b64c0af17c0523a9ee5d04c0f4ad0ab414e5bef76b7391cf7367 s-03p-validador-consultas.plb

2020-05-18 19:02:45.033-252776-MF01CF01MF01 ==> OK Estructura del archivo de respuestas correcto
2020-05-18 19:02:45.051-252776-UJ02UJ02UJ02 ==> OK Consulta 1 correcta.
2020-05-18 19:02:45.068-252776-RZ03RZ03VZ03 ==> OK Consulta 2 correcta.
2020-05-18 19:02:45.071-252776-RM04SM04AM04 ==> OK Consulta 3 correcta.
2020-05-18 19:02:45.084-252776-YL05OL05AL05 ==> OK Consulta 4 correcta.
2020-05-18 19:02:45.096-252776-MV06BV06PV06 ==> OK Consulta 5 correcta.
2020-05-18 19:02:45.108-252776-U007D007I007 ==> OK Consulta 6 correcta.
2020-05-18 19:02:45.120-252776-RA08CA08IA08 ==> OK Consulta 7 correcta.
2020-05-18 19:02:45.136-252776-RT09UT090T09 ==> OK Consulta 8 correcta.
2020-05-18 19:02:45.147-252776-MZ10RZ10I210 ==> OK Consulta 9 correcta.
2020-05-18 19:02:45.155-252776-MW11SW110W11 ==> OK Consulta 10 correcta.
2020-05-18 19:02:45.170-252776-U0120012S012 ==> OK Consulta 11 correcta.
2020-05-18 19:02:45.183-252776-RQ13BQ13UQ13 ==> OK Consulta 12 correcta.
2020-05-18 19:02:45.196-252776-RM14CM14BM14 ==> OK Consulta 13 correcta.
2020-05-18 19:02:45.208-252776-MW15CW15AW15 ==> OK Consulta 14 correcta.
2020-05-18 19:02:45.220-252776-ME16UE16SE16 ==> OK Consulta 15 correcta.
2020-05-18 19:02:45.221-252776-UH17RH17TH17 ==> OK Validacion concluida.

Disconnected from Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - Production
```

### Validador - Joaquín Valdepino Mendieta

```
=====
Iniciando proceso de validación de respuestas
Incluir en el reporte a partir de este punto
=====
SESSION_TIME: 18/05/2020 19:36:26
USR_COD_HOST: 694761
OS_USER: joaquin-valdespino
BD_USER: MUVA_P1101_SUBASTAS
CON_NAME: cursobd
c5761c335366b64c0af17c0523a9ee5d04c0f4ad0ab414e5bef76b7391cf7367 s-03p-validador-consultas.plb

2020-05-18 19:36:26.916-694761-PK01CK01MK01 ==> OK Estructura del archivo de respuestas correcto
2020-05-18 19:36:26.980-694761-C002U002U002 ==> OK Consulta 1 correcta.
2020-05-18 19:36:27.038-694761-0W03RW03VW03 ==> OK Consulta 2 correcta.
2020-05-18 19:36:27.096-694761-JI04SI04AI04 ==> OK Consulta 3 correcta.
2020-05-18 19:36:27.149-694761-VY05OY05OY05 ==> OK Consulta 4 correcta.
2020-05-18 19:36:27.204-694761-MP06BP06PP06 ==> OK Consulta 5 correcta.
2020-05-18 19:36:27.260-694761-PC07DC071C07 ==> OK Consulta 6 correcta.
2020-05-18 19:36:27.315-694761-CD08CD081D08 ==> OK Consulta 7 correcta.
2020-05-18 19:36:27.376-694761-0I09UI090I09 ==> OK Consulta 8 correcta.
2020-05-18 19:36:27.402-694761-JY10RY101Y10 ==> OK Consulta 9 correcta.
2020-05-18 19:36:27.406-694761-VI11SI110I11 ==> OK Consulta 10 correcta.
2020-05-18 19:36:27.411-694761-PR12OR12SR12 ==> OK Consulta 11 correcta.
2020-05-18 19:36:27.415-694761-PF13BF13UF13 ==> OK Consulta 12 correcta.
2020-05-18 19:36:27.421-694761-CF14CF14BF14 ==> OK Consulta 13 correcta.
2020-05-18 19:36:27.426-694761-0V15CV15AV15 ==> OK Consulta 14 correcta.
2020-05-18 19:36:27.431-694761-JY16UY16SY16 ==> OK Consulta 15 correcta.
2020-05-18 19:36:27.432-694761-VE17RE17TE17 ==> OK Validacion concluida.

Disconnected from Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - Production
Version 18.3.0.0.0
```

## Conclusiones

En la presente práctica a pesar de que realmente el objetivo era práctica con funciones de agregación y con subconsultas, observamos que el análisis de datos también tiene un campo amplio ya que como se menciona en la parte de funciones de agregación, acompañadas de subconsultas, podemos hacer cálculos estadísticos con base en los datos proporcionados de la base, ampliando el análisis de los mismos. Con esta practica podemos reafirmar el conocimiento en diseño de consultas dada por estas dos partes mencionadas.

## Comentario

Realmente agradecemos muchísimo el que haya dado una clase más para poder hacer ejercicios de esta práctica la realidad es que para ambos nos resultó realmente complicada y sobre todo laboriosa, sin embargo, creemos que es necesaria sobre todo para reafirmar los conocimientos de tanto subconsultas como de funciones de agregación.

Por otro lado, tuvimos muchísimos problemas con el validador en las consulta 8,9 y 12, a pesar de ello, realmente sólo la consulta 9 fue la única que no pudimos ver que error tenía

## Referencias

- 1) SQL\*Plus Quick Start. Recuperado el 26 de abril de 2020, de [https://docs.oracle.com/cd/B19306\\_01/server.102/b14357/qstart.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14357/qstart.htm)
- 2) Tema 11. Recuperado el 9 de mayo de 2020, de <https://drive.google.com/drive/folders/0B4FYUEdc5Wx6bS1VQzI2OU55NVU>