

# DML - Consultas agrupadas y anidadas

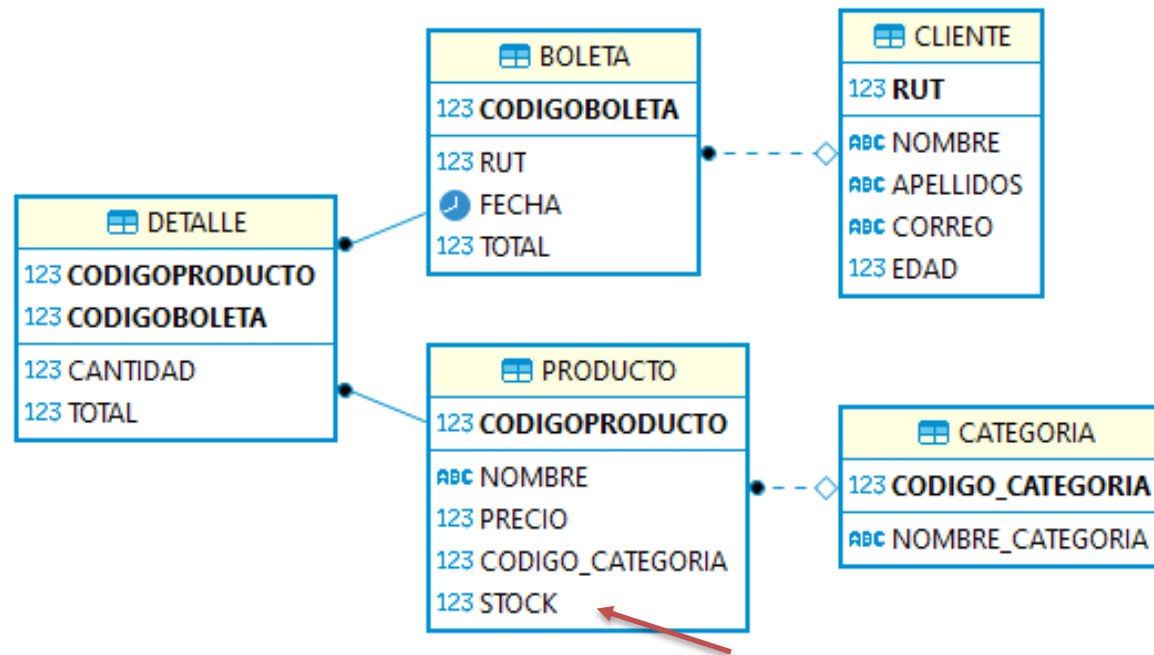
INF 324 Bases de datos – Ing. Civil Informática 2020

Laboratorio N2 | 01-10-2020

Daniel Morán Faúndez

# Modelo a trabajar

- Venta de productos a clientes (un atributo y tabla agregados)



- Puede ser que algunas fotos no coincidan con los resultados si prueban las consultas. Esto es por la boleta 6 que fue incluida para la sección de consultas anidadas.

# Consultas agrupadas

- Al “agrupar” consultas, lo que estaremos haciendo es un resumen de nuestra consulta, es decir, utilizando alguna función que permita hacerlo.

Funciones de agrupación	
AVG	Calcula el promedio sobre una columna
COUNT	Realiza el conteo sobre una columna
MAX	Calcula el máximo valor sobre una columna
MIN	Calcula el mínimo valor sobre una columna
STDDEV	Calcula la desviación estándar sobre una columna
SUM	Realiza la suma sobre una columna
VARIANCE	Calcula la varianza sobre una columna

- En la tabla se observan funciones que permiten calcular el promedio, cantidad de filas, numero máximo, mínimo, entre otros. Como se puede observar, todas se aplican a columnas que sean numéricas.

# Consultas agrupadas

- Ejemplo de agrupamiento

123 CODIGOPRODUCTO	123 CODIGOBOLETA	123 CANTIDAD	123 TOTAL
18	1	1	200.000
14	1	1	200.000
9	1	1	450.000
2	1	1	110.000
14	2	1	200.000

```
SELECT COUNT(*) AS TOTAL_FILAS FROM DETALLE  
GROUP BY CODIGOBOLETA;
```

```
SELECT COUNT(CANTIDAD) AS TOTAL_FILAS FROM DETALLE  
GROUP BY CODIGOBOLETA;
```

```
SELECT COUNT(TOTAL) AS TOTAL_FILAS FROM DETALLE  
GROUP BY CODIGOBOLETA;
```

123 TOTAL_FILAS
4

1

```
SELECT * FROM DETALLE  
GROUP BY CODIGOBOLETA;
```



SQL Error [979] [42000]: ORA-00979: not a GROUP BY expression

```
SELECT COUNT(TOTAL), TOTAL AS TOTAL_FILAS FROM DETALLE  
GROUP BY CODIGOBOLETA;
```



SQL Error [979] [42000]: ORA-00979: not a GROUP BY expression

# Consultas agrupadas

- **Ojo:** atributo de agrupamiento

123 CODIGOPRODUCTO	123 CODIGOBOLETA	123 CANTIDAD	123 TOTAL
18	1	1	200.000
14	1	1	200.000
9	1	1	450.000
2	1	1	110.000
14	2	1	200.000

```
SELECT CODIGOBOLETA, CODCOUNT(*) AS TOTAL_FILAS FROM DETALLE  
GROUP BY CODIGOBOLETA;
```

```
SELECT CODIGOBOLETA, COUNT(CANTIDAD) AS TOTAL_FILAS FROM DETALLE  
GROUP BY CODIGOBOLETA;
```

```
SELECT CODIGOBOLETA, COUNT(TOTAL) AS TOTAL_FILAS FROM DETALLE  
GROUP BY CODIGOBOLETA;
```

123 CODIGOBOLETA	123 TOTAL_FILAS
1	4
2	1

- Se puede observar que el atributo CODIGOBOLETA puede ser mostrado, pues este es el atributo por el cual se están agrupando.

# Consultas agrupadas

- Otro ejemplo más realista: obtener los totales de las boletas generadas.

```
SELECT CODIGOBOLETA, SUM(TOTAL) FROM DETALLE  
GROUP BY CODIGOBOLETA;
```

123 CODIGOBOLETA	123 SUM(TOTAL)
1	960.000
2	400.000
4	1.427.000
5	6.287.980
3	1.548.500

- Por cada boleta existe un detalle de los productos, el cual contiene un subtotal (cantidad de productos x precio unitario), por ende al sumarlos obtendremos el total de esa boleta.

# Consultas agrupadas

- Para filtrar en una consulta agrupada, si o si es necesario utilizar la sentencia HAVING.

Códigos de boleta que tengan mas de 2 productos en su detalle

```
SELECT COUNT(TOTAL) FROM DETALLE  
GROUP BY CODIGOBOLETA  
HAVING COUNT(TOTAL) > 2;
```

123 COUNT(TOTAL)
4

```
SELECT CODIGOBOLETA, COUNT(*) FROM DETALLE  
GROUP BY CODIGOBOLETA  
HAVING COUNT(*) > 2;
```

123 CODIGOBOLETA	123 COUNT(*)
1	4

```
SELECT CODIGOBOLETA, COUNT(CODIGOPRODUCTO) FROM DETALLE  
GROUP BY CODIGOBOLETA  
HAVING COUNT(CODIGOPRODUCTO) > 2;
```

123 CODIGOBOLETA	123 COUNT(CODIGOPRODUCTO)
1	4

Promedio del precio de los productos de cada boleta que sea superior a 200.000

```
SELECT CODIGOBOLETA, AVG(TOTAL) FROM DETALLE  
GROUP BY CODIGOBOLETA  
HAVING AVG(TOTAL) > 200000;
```

123 CODIGOBOLETA	123 AVG(TOTAL)
1	240.000

# Consultas agrupadas

- También es posible utilizar ORDER BY para ordenar los resultados de las consultas, pero debe ser utilizado después de aplicar la sentencia GROUP BY.

Promedio del precio de los productos de cada boleta ordenados por su promedio de mayor a menor

```
SELECT CODIGOBOLETA, AVG(TOTAL) FROM DETALLE  
GROUP BY CODIGOBOLETA  
ORDER BY AVG(TOTAL) DESC;
```

123 CODIGOBOLETA	123 AVG(TOTAL)
1	240.000
2	200.000

Promedio del precio de los productos de cada boleta ordenados por el código de la boleta de mayor a menor

```
SELECT CODIGOBOLETA, AVG(TOTAL) FROM DETALLE  
GROUP BY CODIGOBOLETA  
ORDER BY CODIGOBOLETA DESC;
```

123 CODIGOBOLETA	123 AVG(TOTAL)
2	200.000
1	240.000



# Consultas anidadas

- Las consultas anidadas son consultas dentro de una consulta mas grande. Pueden utilizarse en INSERT, UPDATE, DELETE y en otro SELECT. Nos sirven para buscar algún valor o un conjunto de valores para filtrar en la consulta mas general cuando se desconoce el valor.

```
SELECT column, column, ...  
FROM table  
WHERE (column, column, ...) IN  
      (SELECT column, column, ...  
       FROM table  
       WHERE condition);
```

# Consultas anidadas

- Ejemplo: Obtener los productos que tengan un precio mayor al promedio

```
SELECT AVG(P.PRECIO) FROM PRODUCTO P;
```

123 CODIGOPRODUCTO	ABC NOMBRE	123 PRECIO	ABC CATEGORIA	123 STOCK
1	NOTEBOOK LENOVO	380.000	PORTABLES	20
2	CELULAR MOTOROLA	110.000	TECNOLOGIA	1
3	AUDIFONOS MACROTEL	3.500	AUDIO	67
4	NOTEBOOK SAMSUNG	500.000	PORTABLES	23
5	MONITOR 17" AOC	67.990	MONITORES	6
6	MONITOR 21" DELL	81.990	MONITORES	7
7	NOTEBOOK HP	280.990	PORTABLES	5
8	CELULAR IPHONE 6S	560.000	TECNOLOGIA	8
9	CELULAR LG	450.000	TECNOLOGIA	14
10	MOUSE BLUETOOTH	28.990	ACCESORIOS	28
11	MONITOR 17" SAMSUNG	150.000	MONITORES	4
12	MONITOR 17" LENOVO	250.000	MONITORES	6
13	MONITOR 17" LG	125.000	MONITORES	6
14	MONITOR 15" LENOVO	200.000	MONITORES	7
15	MOTO G 2013	130.000	TECNOLOGIA	17
16	MOTO G 2015	180.000	TECNOLOGIA	16
17	MOTO X PLAY	370.000	TECNOLOGIA	14
18	IPAD MINI 2	200.000	TABLETS	18
19	IPAD AIR	260.000	TABLETS	14
20	NOTEBOOK MSI	850.000	PORTABLES	20
21	MOUSE MICROSOFT	8.500	ACCESORIOS	70

123 AVG(P.PRECIO)
246.998,0952380952

```
SELECT P.CODIGOPRODUCTO, P.NOMBRE, P.STOCK, P.PRECIO
FROM PRODUCTO P
WHERE P.PRECIO > (SELECT AVG(P.PRECIO) FROM PRODUCTO P);
```

123 CODIGOPRODUCTO	ABC NOMBRE	123 STOCK	123 PRECIO
1	NOTEBOOK LENOVO	20	380.000
4	NOTEBOOK SAMSUNG	23	500.000
7	NOTEBOOK HP	5	280.990
8	CELULAR IPHONE 6S	8	560.000
9	CELULAR LG	14	450.000
12	MONITOR 17" LENOVO	6	250.000
17	MOTO X PLAY	14	370.000
19	IPAD AIR	14	260.000
20	NOTEBOOK MSI	20	850.000

# Consultas anidadas

- Ejemplo (solo para complicar las cosas, porque sería mas sencillo): Obtener los productos que tengan un precio mayor al promedio y que no sean de una categoría que comiencen con la letra M o T.

123 CODIGOPRODUCTO	ABC NOMBRE	123 PRECIO	ABC CATEGORIA	123 STOCK
1	NOTEBOOK LENOVO	380.000	PORTABLES	20
2	CELULAR MOTOROLA	110.000	TECNOLOGIA	1
3	AUDIFONOS MACROTEL	3.500	AUDIO	67
4	NOTEBOOK SAMSUNG	500.000	PORTABLES	23
5	MONITOR 17" AOC	67.990	MONITORES	6
6	MONITOR 21" DELL	81.990	MONITORES	7
7	NOTEBOOK HP	280.990	PORTABLES	5
8	CELULAR IPHONE 6S	560.000	TECNOLOGIA	8
9	CELULAR LG	450.000	TECNOLOGIA	14
10	MOUSE BLUETOOTH	28.990	ACCESORIOS	28
11	MONITOR 17" SAMSUNG	150.000	MONITORES	4
12	MONITOR 17" LENOVO	250.000	MONITORES	6
13	MONITOR 17" LG	125.000	MONITORES	6
14	MONITOR 15" LENOVO	200.000	MONITORES	7
15	MOTO G 2013	130.000	TECNOLOGIA	17
16	MOTO G 2015	180.000	TECNOLOGIA	16
17	MOTO X PLAY	370.000	TECNOLOGIA	14
18	IPAD MINI 2	200.000	TABLETS	18
19	IPAD AIR	260.000	TABLETS	14
20	NOTEBOOK MSI	850.000	PORTABLES	20
21	MOUSE MICROSOFT	8.500	ACCESORIOS	70

```
SELECT c.NOMBRE_CATEGORIA
FROM CATEGORIA c
WHERE NOMBRE_CATEGORIA NOT LIKE 'M%'
AND NOMBRE_CATEGORIA NOT LIKE 'T%';
```

ABC NOMBRE_CATEGORIA
PORTABLES
AUDIO
ACCESORIOS

```
SELECT P.CODIGOPRODUCTO, P.NOMBRE, P.STOCK, P.PRECIO, P.CODIGO_CATEGORIA
FROM PRODUCTO P
WHERE P.PRECIO > ( SELECT AVG(P2.PRECIO)
                  FROM PRODUCTO P2 )
AND P.CODIGO_CATEGORIA NOT IN ( SELECT DISTINCT CODIGO_CATEGORIA
                                FROM CATEGORIA c
                                WHERE c.NOMBRE_CATEGORIA NOT LIKE 'M%'
                                AND c.NOMBRE_CATEGORIA NOT LIKE 'T%' );
```

123 CODIGOPRODUCTO	ABC NOMBRE	123 STOCK	123 PRECIO	123 CODIGO_CATEGORIA
17	MOTO X PLAY	14	370.000	2
9	CELULAR LG	14	450.000	2
8	CELULAR IPHONE 6S	8	560.000	2
19	IPAD AIR	14	260.000	6
12	MONITOR 17" LENOVO	6	250.000	4

# Consultas anidadas

- OJO:** si una subconsulta retorna solo un valor, este será reconocido como tal. Pero si la subconsulta retorna mas de una fila, esta será considerada como una lista. También se debe tener cuidado con consultar por mas de un atributo en una consulta.

```
SELECT P.CODIGOPRODUCTO, P.NOMBRE, P.STOCK, P.PRECIO, P.CODIGO_CATEGORIA
FROM PRODUCTO P
WHERE P.PRECIO > ( SELECT AVG(P2.PRECIO)
                  FROM PRODUCTO P2 )
AND P.CODIGO_CATEGORIA <> ( SELECT DISTINCT CODIGO_CATEGORIA
                           FROM CATEGORIA C
                           WHERE C.NOMBRE_CATEGORIA NOT LIKE 'M%'
                           AND C.NOMBRE_CATEGORIA NOT LIKE 'T%' );
```

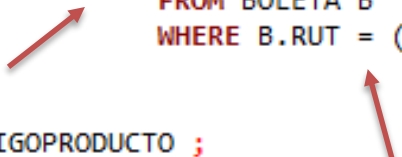


SQL Error [1427] [21000]: ORA-01427: single-row subquery returns more than one row

# Consultas anidadas

- Obtener el promedio de dinero utilizado en los productos comprados por Francisca

```
SELECT D.CODIGOPRODUCTO, AVG(D.TOTAL)
FROM DETALLE D
WHERE CODIGOBOLETA IN ( SELECT CODIGOBOLETA
                        FROM BOLETA B
                        WHERE B.RUT = ( SELECT RUT
                                      FROM CLIENTE C
                                      WHERE NOMBRE = 'Francisca'))
GROUP BY D.CODIGOPRODUCTO ;
```



123 CODIGOPRODUCTO	123 AVG(D.TOTAL)
2	110.000
3	8.750
8	1.120.000
9	450.000
11	300.000
14	200.000
18	200.000

- ¿Y qué nombre tienen esos productos 🤖?
- ¿Qué pasa si hay mas de una clienta llamada Francisca?