

BASE DE DATOS 2

Ing. Jacqueline Guzman



FUNCIONES DE AGREGACIÓN, SUBCONSULTAS, AGRUPACIONES

SELECT: FUNCIONES DE AGREGACIÓN

○ COUNT

- permite saber la cantidad de tuplas que cumplen determinada condición.
- Cuenta valores Nulos y valores Duplicados.
- Se puede combinar con DISTINCT para no contar ni nulos ni duplicados
- Resultado: un único valor numérico

○ SUM

- permite sumar los valores de un campo o expresión numérica
- Se puede usar el DISTINCT para no considerar repetidos.
- No considera nulos
- Resultado: un único valor numérico

○ AVG

- permite promediar los valores de un campo o expresión numérica
- Se puede usar el DISTINCT para no considerar repetidos
- No considera nulos
- Resultado: un único valor numérico



SELECT: FUNCIONES DE AGREGACIÓN

- MAX

- Valor Máximo de un campo o expresión numérica

- MIN

- Valor Mínimo de un campo o expresión numérica

- LAS FUNCIONES DE AGREGACION

Solo pueden utilizarse en la cláusula SELECT o en la cláusula HAVING.

No se pueden combinar con columnas que devuelvan más de un valor, a menos que la consulta contenga una cláusula GROUP BY.



SELECT: FUNCIONES DE AGREGACIÓN - EJEMPLOS

○ COUNT

-- cantidad de productos vendidos el dia 10/2/2016

```
SELECT COUNT(*) FROM Ventas V
WHERE V.fechaV = convert(date, '10/02/2016')
```

-- cantidad de productos DISTINTOS vendidos el dia 10/2/2016

```
SELECT COUNT(DISTINCT(*)) FROM Ventas V
WHERE V.fechaV = convert(date, '10/02/2016')
```

○ SUM

-- obtener el ingreso total de dinero por ventas del dia 10/2/2016

```
SELECT SUM(V.precioLista - V.importeDescontado)
FROM Ventas V
WHERE V.fechaV = convert(date, '10/02/2016')
```



SELECT: FUNCIONES DE AGREGACIÓN - EJEMPLOS

○ AVG

-- obtener el *PROMEDIO* de ingreso total de dinero por ventas del año 2016

```
SELECT AVG(V.precioLista - V.importeDescontado)
FROM Ventas V
WHERE Year(V.fechaV) = 2016
```

○ MAX

-- obtener el *mayor precio* de producto

```
SELECT MAX(P.precioP)
FROM Productos P
```

○ MIN

-- obtener el *menor precio* de producto

```
SELECT MIN(P.precioP)
FROM Productos P
```



SELECT: FUNCIONES DE AGREGACIÓN - EJEMPLOS

-- obtener la fecha de la ultima y primer venta del 2016, asi como la suma y promedio de ingresos por ventas del 2016

```
SELECT MAX(V.fechaV) , MIN(V.FechaV) ,  
        AVG(V.precioLista - V.importeDescontado) ,  
        SUM(V.precioLista - V.importeDescontado)  
FROM Ventas V  
WHERE Year(V.fechaV) = 2016
```

-- mejorando la salida,

```
SELECT MAX(V.fechaV) as 'VentaMasReciente'  
        MIN(V.FechaV) as 'VentaMasAntigua',  
        AVG(V.precioLista - V.importeDescontado) as 'PromVentas',  
        SUM(V.precioLista - V.importeDescontado) as 'SumaVentas'  
FROM Ventas V  
WHERE Year(V.fechaV) = 2016
```



SUBCONSULTAS: IN

```
SELECT * FROM NOMTBL WHERE Campo_J IN (subconsulta)
```

- La subconsulta debe devolver datos del mismo tipo de Campo_J

PRODUCTOS (codP, nombreP, descripP, precioP, codProv)

-- obtener los nombre de productos vendidos en el 2016

```
SELECT DISTINCT(P.nombreP)
FROM Productos P
WHERE P.codP IN (SELECT V.codP
                  FROM Ventas V
                  WHERE year(V.fechaV) = 2016
                )
```



SUBCONSULTAS: IN

IN: un campo de una consulta ppal esta dentro del conjunto resultado de una subconsulta

PRODUCTOS (codP, nombreP, descripP, precioP, codProv)

-- obtener los nombre de productos vendidos en el 2016

```
SELECT P.nombreP
FROM Productos P
WHERE P.codP IN (SELECT DISTINCT(V.codP)
                 FROM Ventas V
                 WHERE year(V.fechaV) = 2016
                )
```

- El tipo de datos de P.codP es igual a V.codP resultado de la subconsulta
- La subconsulta DEBE devolver números, no podría ser una tupla
- La subconsulta NO filtra repetidos, la subconsulta OPTIMA debería utilizar el DISTINCT para evitar repetidos

SUBCONSULTAS: EXISTS

EXISTS: una de las condiciones de la consulta ppal es la existencia de por lo menos una tupla en el resultado de una subconsulta

PRODUCTOS (codP, nombreP, descripP, precioP, codProv)

-- obtener los nombre de productos que han sido vendidos en el año actual

```
SELECT P.nombreP
FROM Productos P
WHERE EXISTS (SELECT *
              FROM Ventas V
              WHERE V.codP = P.codP AND
                    Year(V.fechaV) = 2016
              )
```

- La subconsulta siempre devuelve ‘*’
- La subconsulta esta relacionada al producto a través del código del producto de la venta

SUBCONSULTAS: EJEMPLOS

-- *Eliminar los Productos que no han sido vendidos*

```
DELETE FROM Productos
WHERE NOT EXISTS (SELECT *
                  FROM Ventas V
                  WHERE V.codP = Productos.codP)
```

-- Bajar en un 10% el precio de los productos para los cuales se han vendido menos de 5 veces en el año

```
UPDATE Productos
SET P.precioP = P.precioP - P.precioP*0.1
WHERE (SELECT Count(*)
       FROM Ventas V
       WHERE V.codP = PRODUCTOS.codP) < 5
```



SUBCONSULTAS: EJEMPLOS (UPDATE, DELETE)

-- *Eliminar los Productos que no han sido vendidos*

```
DELETE FROM Productos
WHERE NOT EXISTS (SELECT *
                  FROM Ventas V
                  WHERE V.codP = Productos.codP)
```

-- Bajar en un 10% el precio de los productos para los cuales se han vendido menos de 5 veces en el año

```
UPDATE Productos
SET P.precioP = P.precioP - P.precioP*0.1
WHERE (SELECT Count (*)
       FROM Ventas V
       WHERE V.codP = PRODUCTOS.codP) < 5
```



SUBCONSULTAS: EJEMPLOS (SELECT)

VENTAS (fechaV, codP, precioLista, importeDescontado, CodVend)

Analice las siguientes consultas y determine cual devuelve: los datos de la ultima Venta del vendedor 25

```
SELECT *  
FROM VENTAS  
WHERE  
CodVend = 25 AND  
fechaV =  
    (  
    SELECT MAX(fechaV)  
    FROM VENTAS  
    WHERE CodVend = 25  
    )
```

```
SELECT *  
FROM VENTAS  
WHERE CodVend = 25  
AND fechaV =  
    (  
    SELECT MAX(fechaV)  
    FROM VENTAS  
WHERE CodVend = 25  
    )
```

```
SELECT *  
FROM VENTAS  
WHERE CodVend = 25  
AND fechaV =  
    (  
    SELECT MAX(fechaV)  
    FROM VENTAS  
    WHERE CodVend = 25  
    )
```

SUBCONSULTAS: CARACTERISTICAS

- La referencia dentro de una subconsulta al valor de un atributo de alguna tabla de la consulta ppal se denomina referencia externa.
- Cuando se ejecuta una consulta que contiene una subconsulta con referencias externas, la subconsulta se ejecuta por cada fila de la consulta principal
- Se puede usar una subconsulta en los siguientes lugares de la consulta ppal:
 - La lista de selección
 - El FROM
 - El WHERE
 - El HAVING
- Las subconsultas no pueden:
 - contener la cláusula ORDER BY
 - ser la UNION de varias sentencias SELECT
 - Si la subconsulta aparece en la lista de selección, o esta asociada a un operador igual "=" solo puede devolver un único registro.



CONSULTAS REUNIÓN (VARIAS TABLAS)

SQL permite “reunir” dos o mas tablas para obtener información que se encuentra en distintas tablas. Cuidado tener presente las FK-PK

```
SELECT * FROM NOMTBL1, NOMTBL2, .....  
WHERE NOMTBL1.Campo_j = NOMTBL2.Campo_k ... AND/OR .....
```

PRODUCTOS (codP, nombreP, descripP, precioP, codProv)

VENTAS(fechaV, codP, precioLista, importeDescontado)

-- obtener los nombre de productos vendidos el 01/06/2016

```
SELECT P.nombreP  
FROM Productos P, Ventas V  
WHERE P.codP = V.codP AND  
      V.fechaV = convert(date, '01/06/2016')
```



CONSULTAS REUNIÓN (VARIAS TABLAS)

PRODUCTOS (codP, nombreP, descripP, precioP, codProv)

VENTAS(fechaV, codP, precioLista, importeDescontado)

-- obtener los datos de los productos vendidos en la primer venta realizada por la empresa

```
SELECT P.*
FROM Productos P, Ventas V
WHERE  P.codP = V.codP AND
        V.fechaV = ( SELECT MIN(fechaV)
                      FROM Ventas V2
                    )
```



FUNCION SOME

PRODUCTOS (codP, nombreP, descripP, precioP, codProv)


```
SELECT nombreProd
```

```
FROM Productos
```

```
WHERE precio > SOME
```

```
(Select precio  
  from Productos  
 where CodProv = 2)
```

Conjunto de precios



Obtiene nombre de los productos cuyos precios sean mayores a algun precio de producto del proveedor 2



FUNCION ALL

PRODUCTOS (codP, nombreP, descripP, precioP, codProv)


```
SELECT nombreProd
```

```
FROM Productos
```

```
WHERE precio > ALL
```

```
(Select precio  
  from Productos  
 where CodProv = 2)
```

Conjunto de precios



Obtiene nombre de los productos cuyos precios sean mayores a
TODOS los precio de productos del proveedor 2



CONSULTAS: GROUP BY

Permite agrupar los resultados de una consulta usando como criterio de agrupación los distintos valores que toma algún/os campos del resultado

```
SELECT campo_x, campo_y, campo_n  
FROM NOMTBL  
WHERE CONDICION  
GROUP BY campo_x, campo_y, campo_n
```

- En el SELECT SOLO pueden ir :
 - Atributos presentes en la cláusula GROUP BY.
 - Funciones de agregación sobre atributos.
 - Expresiones aritméticas que utilicen los anteriores.
- Primero será ejecutado el WHERE y las tuplas resultados se agrupan según lo indicado en el Group By

CONSULTAS: GROUP BY

Se quiere saber la cantidad de productos vendidos en el 2016 agrupados por producto.

VENTAS(fechaV, codP, precioLista, importeDescontado)

```
SELECT V.codP, count(*) FROM Ventas V
WHERE year(V.fechaV) = 2016 GROUP BY V.codP
```

Se quiere saber la cantidad de productos vendidos diariamente en el 2016.

```
SELECT V.fechaV, count(distinct(V.codP)) FROM Ventas V
WHERE year(V.fechaV) = 2016 GROUP BY V.fechaV
```



CONSULTAS: GROUP BY

Se quiere saber el código, nombre y la cantidad de productos vendidos en el 2016 agrupados por producto.

VENTAS(fechaV, codP, precioLista, importeDescontado)

```
SELECT V.codP, P.nombreP, count(*)  
FROM Ventas V, Productos P  
WHERE V.codP = P.codP AND year(V.fechaV) = 2016  
GROUP BY V.codP, P.nombreP
```



CONSULTAS: GROUP BY - HAVING

Permite filtrar el resultados del GROUP BY según una CONDICION indicada en la clausula HAVING

```
SELECT campo_x, campo_y, campo_n
FROM NOMTBL
WHERE CONDICION
GROUP BY campo_x, campo_y, campo_n
HAVING CONDICION_SOBRE_EL_GRUPO
```

Se quiere saber la cantidad de productos vendidos en el 2016 , cuyo precio sea mayor a 1000, agrupados por producto.

VENTAS(fechaV, codP, precioLista, importeDescontado)

```
SELECT V.codP, count(*) FROM Ventas V
WHERE year(V.fechaV) = 2016
GROUP BY V.codP
HAVING V.PrecioP > 1000
```

