



DML: Consultas Anidadas

- ✓ Consultas completas que se ubican dentro de la cláusula WHERE de otra consulta
- ✓ Ejemplo:

```
SELECT A1, ..., An
FROM R1, ..., Rm
WHERE (Aj, ..., Ak <op-comp>)
      (SELECT Bj, ..., Bk
       FROM S1, ..., Sn
       WHERE Q)
```

<op-comp> puede ser IN, = ANY, > ANY, = ALL, > ALL.



DML: Consultas anidadas

- ✓ Dar los nombres de los fabricantes que venden productos que además son vendidos por el fabricante número 1.

```
SELECT f.nombre
FROM FABS f, VENTAS v
WHERE f.#f = v.#f AND
      f.#f <> 1 AND
      v.#p IN
      (SELECT #p)
      FROM VENTAS
      WHERE VENTAS.#f = 1
```



DML: Función EXISTS

- ✓ Chequea si el resultado de una consulta anidada es vacío
- ✓ Ejemplo:
 - ✓ Dar los nombres de los fabricantes que sólo venden el producto número 2

```
SELECT nombre
FROM FABS
WHERE NOT EXISTS
  (SELECT *
FROM VENTAS
WHERE FABS.#f = VENTAS.#f
AND VENTAS.#p <> 2)
```



DML: Agrupamiento de tuplas

- ✓ Se quiere resolver: Dar, para cada fabricante, la cantidad de productos que vendió
- ✓ Esto implica:
 - ✓ Tomar cada grupo de tuplas en Ventas correspondiente a un fabricante
 - ✓ Contar la cantidad de tuplas en el grupo
- ✓ Para realizar esta operación es necesaria la cláusula GROUP BY



DML: Agrupamiento de tuplas

- ✓ Es una cláusula que se agrega al SELECT-FROM-WHERE

- ✓ Por ejemplo:

```
SELECT #f, count(*)  
FROM VENTAS  
GROUP BY #f
```

- ✓ Cómo funciona ?
 - ✓ Genera un grupo por cada #fabricante distinto
 - ✓ De cada grupo devuelve el #fabricante y la cantidad de tuplas de dicho grupo



DML: Agrupamiento de Tuplas

- ✓ Dar el número de fabricante y los promedios de precios a los cuales vendió

```
SELECT #f, avg(precio)
FROM VENTAS
GROUP BY #f
```

- ✓ Dar los nombres de fabricante y sus totales de precio vendidos

```
SELECT #f, nombre, sum(precio)
FROM FABS, VENTAS
WHERE FABS.#f = VENTAS.#f
GROUP BY #f, nombre
```



DML: Agrupamiento de Tuplas

- ✓ Regla sobre el GROUP BY
 - ✓ En una sentencia SQL que tiene cláusula GROUP BY, las expresiones en el SELECT pueden ser sólo:
 - ✓ Atributos presentes en la cláusula GROUP BY
 - ✓ Funciones de agregación sobre atributos
 - ✓ Expresiones aritméticas que utilicen los anteriores
- ✓ El agrupamiento se realiza después de aplicar el WHERE. O sea sobre las tuplas que cumplen la condición.



DML: Agrupamiento de tuplas

- ✓ Para cada pareja (#f, #p), dar la cantidad de ventas realizadas y el total de \$.

```
SELECT #f, #p, count(*), sum
      (precio)
FROM VENTAS
GROUP BY #f, #p
```

- ✓ Idem anterior, pero con precios en U\$S (3.00).

```
SELECT #f, nombre, count(*), sum
      (precio)/3.00
WHERE FABS.#f = VENTAS.#f
GROUP BY #f, nombre
```



DML: Condiciones sobre grupos

- ✓ Con la cláusula HAVING se pueden especificar condiciones sobre los grupos.
- ✓ Por ejemplo:
 - ✓ Dar el número de fabricante y los promedios de precios a los cuales vendió, pero para los fabricantes con más de 3 ventas.

```
SELECT #f, avg(precio)
FROM VENTAS
GROUP BY #f
HAVING count(*) > 3;
```



DML: Condiciones sobre grupos

- ✓ Dar los nombres de fabricantes que hicieron más de 5 ventas sobre productos con #p mayor que 2, junto con el total de precio vendido.

```
SELECT #f, nombre, sum(precio)
FROM FABS, VENTAS
WHERE FABS.#f = VENTAS.#f AND #p>2
GROUP BY #f, nombre
HAVING count(*) > 5;
```