

# SQL DML (3)

Bases de datos

# Agrupamiento de registros

- Las consultas de resumen obtienen totales de todas las filas
- Aquí vamos a calcular totales parciales
  - Totales agrupados según un campo
- Por ejemplo
  - Calcular el valor medio del sueldo en función del departamento
  - Calcular el número medio de pedidos que realiza cada cliente

...	8	a
...	5	b
...	4	a
...	7	a
...	2	c
...	1	b

⇒

...	8	a
...	4	a
...	7	a

...	5	b
...	1	b

...	2	c
-----	---	---

# Cláusula GROUP BY

- Group by -> columnas para agrupar
- Having -> condiciones de los grupos

```
SELECT columna1, columna2, ...  
FROM taboa1, taboa2, ...  
WHERE condición1, condición2, ...  
GROUP BY columna1, columna2, ...  
HAVING condición  
ORDER BY ordenación;
```

# Cláusula GROUP BY

```
SELECT  MAX(sal), deptno
FROM    emp
GROUP BY deptno
```

MAX(SAL)	DEPTNO
-----	-----
5000	10
3000	20
2850	30

```
SELECT  MAX(sal), deptno, empno
FROM    emp
GROUP BY deptno
```

# Cláusula GROUP BY

- Obtén el máximo, mínimo y la suma de los salarios de cada departamento, además del número de empleados de cada departamento

```
SELECT      MAX(sal), MIN(sal), SUM(sal), COUNT(empno), deptno
FROM        emp
GROUP BY    deptno
```

- Obtén el máximo, mínimo y la suma de los salarios de cada departamento, además del número de empleados, considerando los empleados cuyo salario es superior a 1500

```
SELECT      MAX(sal), MIN(sal), SUM(sal), COUNT(empno), deptno
FROM        emp
WHERE       sal > 1500
GROUP BY    deptno
```

# Cláusula GROUP BY

- Obtén los diferentes empleos que hay para cada departamento, incluyendo cuántos empleados lo desempeñan en cada departamento, ordenando por empleo y departamento

```
SELECT    job, deptno, COUNT(*)  
FROM      emp  
GROUP BY  deptno, job  
ORDER BY  job, deptno
```

# Cláusula HAVING

- WHERE indica una condición sobre las filas
- HAVING permite establecer una condición sobre los conjuntos

```
SELECT    MAX(sal), MIN(sal), SUM(sal), COUNT(empno), deptno
FROM      emp
WHERE     sal > 1500
GROUP BY  deptno
```

```
SELECT    MAX(sal), MIN(sal), SUM(sal), COUNT(empno), deptno
FROM      emp
WHERE     sal > 1500
GROUP BY  deptno
HAVING    MIN(sal) > 1600
```

# Orden de ejecución de las cláusulas

1. FROM (obligatoria)
  - a. Determina de qué tablas se seleccionan los datos
2. WHERE (optativa)
  - a. Indica un predicado que expresa la condición que debe cumplir cada fila
3. GROUP BY (optativa)
  - a. Forma conjuntos en función de la igualdad de valores en las columnas que se indiquen
4. HAVING (optativa)
  - a. Indica un predicado que expresa la condición que debe cumplir cada conjunto
5. SELECT (obligatoria)
  - a. Selecciona qué columnas se incluyen en la respuesta
6. ORDER BY (optativa)
  - a. Ordena las filas en el resultado



# Subconsultas

- En las consultas anteriores, las condiciones implicaban elementos conocidos
  - ... WHERE sal > 1200 ...
  - ... WHERE ename LIKE 'A%' ...
- Podemos expresar condiciones con algún elemento desconocido
  - Lo obtenemos con otra consulta

Ej.: Obtener el salario de los empleados cuyo salario es mayor que el del empleado de código 7900.

```
SELECT empno, ename, sal
FROM emp
WHERE sal > (SELECT sal
              FROM emp
              WHERE empno = 7900)
```

# Subconsultas

- Una subconsulta es ...
  - ... una sentencia SELECT ...
  - ... que aparece en un predicado de otra consulta
- El resultado de una subconsulta puede ser
  - Una fila

```
SELECT empno, ename, sal
FROM emp
WHERE sal > (SELECT sal
              FROM emp
              WHERE empno = 7900)
```

```
SELECT empno, ename, sal
FROM emp
WHERE sal >= (SELECT AVG(sal)
              FROM emp)
```

# Subconsultas

- El resultado de una subconsulta puede ser
  - Ninguna fila

```
SELECT * FROM emp WHERE sal >= (SELECT AVG(sal) FROM emp WHERE deptno = 40)
```

- Varias filas

```
SELECT empno, ename, sal
FROM emp
WHERE deptno IN (SELECT deptno
                  FROM dept
                  WHERE loc = 'DALLAS'
                  OR loc = 'CHICAGO')
```

```
SELECT *
FROM emp
WHERE sal = (SELECT sal
              FROM emp
              WHERE deptno = 20)
```

# Operadores cuantificados ALL/ANY

- Si la subconsulta devuelve más de una fila, podemos usar...
- ... operadores cuantificados
  - ALL
  - ANY

```
SELECT *  
FROM emp  
WHERE sal = ANY (SELECT sal  
                  FROM emp  
                  WHERE deptno = 30)
```

```
SELECT *  
FROM emp  
WHERE sal > ALL (SELECT sal  
                  FROM emp  
                  WHERE deptno = 30)
```

# Predicado EXISTS

- El predicado EXISTS comprueba si una subconsulta devuelve filas o no

... [NOT] EXISTS subconsulta

- Si existen empleados del dept 30, obtener los datos de todos los depts

```
SELECT      *  
FROM        dept d  
WHERE       EXISTS (SELECT *  
                    FROM    emp  
                    WHERE    deptno = 30)
```