

SQL El lenguaje de Consulta Estructurado

Funciones de agregación,
Anidamiento avanzado

Funciones de agregación

- Las funciones de agregación operan sobre un conjunto de tuplas y devuelven un único valor que será el resultado de dicha operación.
- Funciones:
 - **SUM** (expr): Suma los valores
 - **MIN** (expr) : Calcula el valor mínimo
 - **MAX** (expr) : Calcula el valor máximo
 - **COUNT** (expr) / **COUNT**(*) : Cuenta el número de tuplas
 - **AVG** (expr) : Calcula la media aritmética.
 - $AVG(expr) = \text{sum}(expr) / \text{count}(expr)$
- Se pueden utilizar para calcular información que está almacenada de forma implícita en la base de datos.

Funciones de agregación

- La mayoría, excepto COUNT(*), ignoran los valores NULL.
 - Se puede usar la función NVL para forzar que se considere los valores NULL como un valor determinado.
- Se puede utilizar DISTINCT en la expresión que se le pasa como parámetro a las funciones de agregación para tener en cuenta sólo los valores distintos.
- Las funciones de agregación devuelven NULL si no hay filas en la tabla necesaria para resolver la expresión o todas contienen NULL.
 - COUNT nunca devuelve NULL.

Funciones de agregación

- Contar cuántos teléfonos distintos de profesores hay en la base de datos.

```
SELECT COUNT(DISTINCT telefono) FROM profesores;
```

COUNT(DISTINCTTELEFONO)
5

- Contar cuántos teléfonos no nulos de profesores hay en la base de datos

```
SELECT COUNT(telefono) FROM profesores;
```

COUNT(TELEFONO)
7

- Contar cuántos profesores hay en la base de datos

```
SELECT COUNT(*) FROM profesores;
```

COUNT(*)
11

Funciones de agregación

- Mostrar el número de alumnos que se han matriculado en la asignatura 'Base de Datos' y la media de todas las notas de la asignatura, ignorando las calificaciones nulas.

```
SELECT COUNT(*) "Matriculados",  
       AVG(decode(calificacion,'MH',10,'SB',9,'NT',7,'AP',5,'SP', 0)) "Media"  
FROM matricular M, asignaturas A  
WHERE M.asignatura = A.codigo  
AND upper(A.nombre) = 'BASES DE DATOS' ;
```

Matriculados	Media
32	2,5

- ¿Qué habría que modificar si quisiéramos contar sólo los alumnos cuya calificación no es nula?
- ¿Qué habría que modificar si quisiéramos tener en cuenta las calificaciones nulas en la media?

GROUP BY

- La cláusula **GROUP BY** permite dividir las tuplas seleccionadas para el resultado por la cláusula WHERE en conjuntos de tuplas disjuntos, sobre cada uno de los cuales se va a aplicar funciones de agregación.

```
SELECT Agregación1, Agregación2,...  
FROM tablas  
WHERE predicado  
GROUP BY expr1, expr2, ..., exprn  
ORDER BY criterios_ordenación
```

- Se creará un grupo de tuplas por cada valor diferente de (*expr1,expr2,...,exprn*)
- La consulta aplica las funciones de agregación *Agregación1, Agregación2,..* a cada grupo, obteniéndose una tupla resultado para cada uno.

GROUP BY

- Mostrar el primer apellido más “pequeño” (en sentido lexicográfico) de los profesores de cada departamento.

```
SELECT MIN(apellido1) "Menor apellido", departamento  
FROM profesores  
GROUP BY departamento;
```



GROUP BY

- Queremos agrupar a los profesores por decenios (el decenio 2 lo forman los profesores con 20-29 años, el decenio 3, profesores con 30-39, ...). Necesitamos un listado ordenado por decenio, en el que se muestre el número de decenio y la fecha de nacimiento del profesor más joven del decenio.

```
SELECT trunc(months_between(sysdate, fecha_nacimiento)/12/10) "Decenio",  
       MAX(fecha_nacimiento) "Fecha"  
FROM profesores  
GROUP BY trunc(months_between(sysdate, fecha_nacimiento)/12/10)  
ORDER BY trunc(months_between(sysdate, fecha_nacimiento)/12/10);
```

Decenio	Fecha
2	14/09/92
3	21/12/80
4	01/01/68
6	01/08/49
7	24/01/42
8	01/08/31
(null)	(null)

```
SELECT trunc(months_between(sysdate, fecha_nacimiento)/12/10) "Decenio",  
       MAX(fecha_nacimiento) "Fecha"  
FROM profesores  
GROUP BY trunc(months_between(sysdate, fecha_nacimiento)/12/10)  
ORDER BY "Decenio"
```


GROUP BY

- Cuando se utiliza agrupación (GROUP BY) y/o funciones de agregación, en la lista de selección **SÓLO** podemos utilizar expresiones que tengan funciones de agregación
 - Se pueden usar expresiones que no utilicen funciones de agregación si son exactamente las expresiones que aparecen en el GROUP BY (ver ejemplo anterior)

- Ej:

```
SELECT MIN(apellido1) "Menor apellido", nombre  
FROM profesores  
GROUP BY departamento;
```

Error "not a GROUP BY expression"

- Si las tuplas se agrupan por valor de departamento, ¿cuál de todos los nombres de profesores de cada departamento tomamos para incluirlo en la única tupla resultado para ese departamento?

HAVING

- La cláusula **HAVING** permite filtrar los grupos que no satisfagan un determinado predicado.

```
SELECT Agregación1, Agregación2,...  
FROM tablas  
WHERE predicado_filtra_filas  
GROUP BY expr1, expr2, ..., exprn  
HAVING predicado_filtra_grupos  
ORDER BY criterios_ordenación
```

- El predicado de la cláusula WHERE se utiliza para eliminar filas que no se usarán en el resultado.
- La cláusula GROUP BY distribuye las filas que quedan en grupos.
- La cláusula HAVING se asegura de que los grupos que se usarán en la salida hacen verdad el predicado_filtra_grupos.

HAVING

- Listar el código, la nota media y el número de alumnos matriculados en cada una de las asignaturas, pero sólo de las asignaturas que tienen un grupo 'B' y más de 10 alumnos.

```
SELECT asignatura, COUNT(*) "Matriculados",  
       round(AVG(decode(calificacion,'MH',10,'SB',9,'NT',7,'AP',5,'SP',0)),2) "Media"  
FROM Matricular  
WHERE Grupo = 'B'  
GROUP BY asignatura  
HAVING COUNT(*) > 10 ;
```

ASIGNATURA	Matriculados	Media
144	25	2,33
113	23	2,67
116	11	2
112	19	1,15
110	12	5,75
114	15	1,75
115	14	1,5

HAVING

- Las funciones de agregación **sólo** pueden utilizarse en la lista de selección, teniendo en cuenta las limitaciones ya comentadas, en la cláusula HAVING y en la cláusula ORDER BY.
- Ej: Listar el código, la nota media y el número de alumnos matriculados en cada una de las asignaturas, pero sólo de las asignaturas que tienen un grupo 'B' y más de 10 alumnos. *El resultado ha de estar ordenado por número de alumnos matriculados.*

```
SELECT asignatura, COUNT(*) "Matriculados",  
       round(AVG(decode(calificacion,'MH',10,'SB',9,'NT',7,'AP',5,'SP',0)),2) "Media"  
FROM Matricular  
WHERE Grupo = 'B'  
GROUP BY asignatura  
HAVING COUNT(*) > 10  
ORDER BY COUNT(*);
```

Orden de Ejecución de la sentencia SELECT

5. SELECT lista_selección
1. FROM tablas
2. WHERE predicado_filtra_filas
3. GROUP BY expr1, . . . , exprk
4. HAVING predicado_filtra_grupos
6. ORDER BY criterios_ordenación

Ejercicios

1. Se desea conocer si los alumnos o las alumnas del curso '15/16' son más estudiosos. Para ello se necesita un listado con la nota media de los alumnos de dicho curso divididos por género.
2. Calcular la media de cada asignatura donde la nota mínima sea superior a 4.
3. Listado con el código de profesor, total de créditos que imparte y código de departamento al que pertenece ordenado por departamento. Queremos que salgan primero todos los profesores del departamento 4, después del 3, ...
4. Para los municipios de la provincia de Málaga, queremos agruparlos por primera letra de su nombre. Para cada grupo queremos calcular el total de habitantes de los municipios del grupo, el número de habitantes del municipio con menos habitantes del grupo y el número de habitantes del municipio con más habitantes del grupo.
5. ¿Qué habría que modificar si queremos visualizar sólo los grupos con menos de 10000 habitantes?

Anidamiento Avanzado

- El resultado de una sentencia SELECT sigue un esquema relacional.
- Dicho resultado se puede usar como una relación. Ya lo hemos hecho con las operaciones de conjunto (UNION, MINUS,...) y con las subconsultas.
- ¿Por qué no usarla en la cláusula **FROM**?
- Ventajas del anidamiento:
 - Estructurar la consulta.
 - Adaptar el orden de ejecución de la consulta a nuestras necesidades.

Anidamiento Avanzado

- Obtener toda la información de las dos asignaturas con más créditos.

```
SELECT *  
FROM asignaturas  
WHERE rownum <= 2 AND credits IS NOT NULL  
ORDER BY credits DESC;
```

CODIGO	NOMBRE	DEPARTAMENTO	CREDITOS	...
140	Prácticas en empresa	1	9	...
200	Teoria de la señal	4	6	...

Problema: ROWNUM (cláusula WHERE) se calcula antes del ORDER BY

La consulta extrae información de dos asignaturas y después las ordena.

Anidamiento Avanzado

- Obtener toda la información de las dos asignaturas con más créditos.

```
SELECT *  
FROM ( SELECT * FROM asignaturas  
      WHERE credits IS NOT NULL  
      ORDER BY credits DESC )  
WHERE rownum <= 2;
```

CODIGO	NOMBRE	DEPARTAMENTO	CREDITOS	...
140	Prácticas en empresa	1	9	...
112	Bases de Datos	1	7	...

En este caso se ordenan las asignaturas y después se seleccionan las dos primeras.

Anidamiento Avanzado

- Obtenga toda la información de las asignaturas, pero sólo de aquellas en las que haya exactamente 25 alumnos matriculados.

```
SELECT asig.*  
FROM asignaturas asig  
WHERE (25, asig.codigo) IN (  
    SELECT COUNT (DISTINCT alumno), asignatura  
    FROM matricular  
    GROUP BY asignatura)
```

CODIGO	NOMBRE	DEPARTAMENTO	CREDITOS	TEORICOS	PRACTICOS	CARACTER	CURSO	WEB	COD_MATERIA
113	Calculo Numerico	2	6	3	3	TR	2	www.fermat.uma.es	4
144	Modelos Computacionales	1	6	5	1	TR	3	(null)	(null)
112	Bases de Datos	1	7	5	3	TR	3	www.lcc.uma.es/~enciso	1

¿ Y si quisiéramos que saliera también el número de alumnos matriculados en esas asignaturas?

Anidamiento Avanzado

- Obtenga toda la información de las asignaturas junto al número de alumnos matriculados en ellas, pero sólo de aquellas en las que haya exactamente 25 alumnos matriculados.

```
SELECT asig.*, 25 "Matriculados"  
FROM asignaturas asig  
WHERE (25, asig.codigo) IN (  
    SELECT COUNT (DISTINCT alumno), asignatura  
    FROM matricular  
    GROUP BY asignatura)
```

CODIGO	NOMBRE	DEPARTAMENTO	CREDITOS	TEORICOS	PRACTICOS	CARA...	CURSO	WEB	COD_MATERIA	Matriculados
113	Calculo Numerico	2	6	3	3	TR	2	www.fermat.uma.es	4	25
144	Modelos Computacionales	1	6	5	1	TR	3	(null)	(null)	25
112	Bases de Datos	1	7	5	3	TR	3	www.lcc.uma.es/~enciso	1	25

Anidamiento Avanzado

- Obtenga la información de las asignaturas, pero sólo de aquellas en las que haya al menos 25 alumnos matriculados.

```
SELECT asig.*  
FROM asignaturas asig  
WHERE 25 >= (SELECT COUNT (DISTINCT alumno)  
             FROM matricular  
             WHERE asig.codigo = asignatura)
```

```
SELECT asig.*  
FROM asignaturas asig  
WHERE codigo IN  
      (SELECT asignatura FROM matricular  
       GROUP BY asignatura  
       HAVING COUNT (DISTINCT alumno) >=25)
```

CODIGO	NOMBRE	
113	Calculo Numerico	...
144	Modelos Computacionales	...
116	Logica Computacional	...
112	Bases de Datos	...
200	Teoria de la señal	...

Anidamiento Avanzado

- ¿Y si queremos añadir el número de alumnos de cada asignatura en la consulta anterior?

```
SELECT asig.*, numero
FROM asignaturas asig JOIN
    (SELECT COUNT (DISTINCT alumno) as numero, asignatura
    FROM matricular
    GROUP BY asignatura)
ON (asig.codigo = asignatura)
WHERE numero >=25
```

CODIGO	NOMBRE	NUMERO	...
113	Calculo Numerico	25	...
144	Modelos Computacionales	25	...
116	Logica Computacional	29	...
112	Bases de Datos	25	...
200	Teoria de la señal	87	...

Se asigna un identificador al
resultado de la expresión

Ejercicios

1. Mostrar los datos del profesor más antiguo de cada departamento.
2. Queremos dar un premio a los alumnos más estudiosos. Para ello se debe mostrar un listado con los tres alumnos (sólo 3) que hayan aprobado más asignaturas. Para cada alumno se ha de mostrar su dni, el número de asignaturas aprobadas y la nota media de las asignaturas aprobadas. Se considerará que la calificación 'MH' equivale a 10 puntos, el 'SB' a 9, el 'NT' a 7 y el 'AP' a 5. La nota media estará redondeada a dos decimales.
3. Obtener un informe con 4 columnas que muestre el dni, nombre completo, email y edad (en años) de los alumnos matriculados en la asignatura que tiene más alumnos matriculados.
4. La tasa de rendimiento de una titulación es la razón entre el número de créditos aprobados por los alumnos y el número de créditos en los que se han matriculado los alumnos. Se desea calcular dicha tasa para cada curso académico.