

DML 4

Funciones de Agrupación
Jerarquías

Funciones de Agrupación

- Las funciones de agrupación operan sobre un conjunto de tuplas para dar un resultado por grupo.
- Ignoran los valores NULL, aunque se puede usar la función NVL para forzar a que se considere a NULL como un valor determinado.
- Mediante la cláusula **GROUP BY**
- Todas las columnas que pongamos en el SELECT que no estén en el GROUP BY han de estar en una función de agrupación.

Funciones de Agrupación

- AVG: Calcula la media Aritmética
- SUM: Suma los valores
- MIN: Calcula el valor mínimo
- MAX: Calcula el valor máximo
- COUNT: Cuenta el número de tuplas

Funciones de Agrupación: Ejemplos

EMPLEADOS
Nombre
Inic
Apellido
NSS
Departamento
Sueldo

- Calcular el sueldo medio de todos los empleados
`select avg(sueldo) from Empleados`
- Considerar como cero los valores nulos de sueldo
`select avg(nvl(sueldo,o)) from Empleados`
- Calcular el sueldo medio de los empleados de cada departamento
`select departamento, avg(sueldo)
from Empleados
GROUP BY departamento`

Funciones de Agregación: Ejemplos

```
select departamento, avg(sueldo)
from Empleados
GROUP BY departamento
```

Si no ponemos
AVG, ¿Qué sueldo
se pondría aquí?

DEPARTAMENTO	AVG(SUELDO)
1	55000
5	33250
4	31000

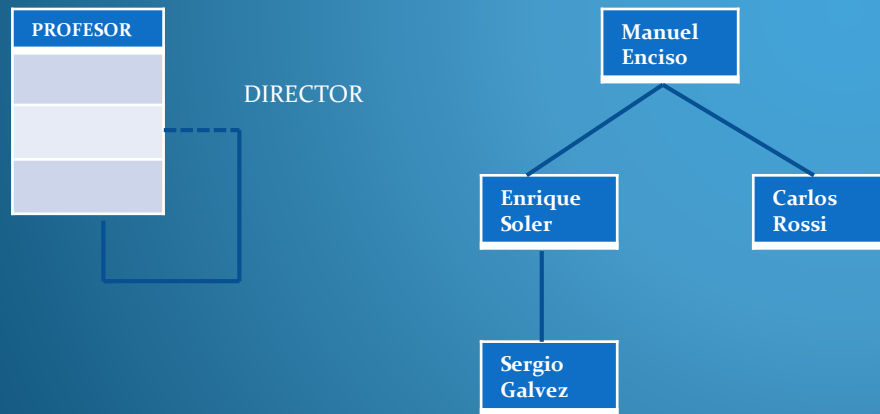
Funciones de Agregación: Having

- Si queremos filtrar el resultado NO se usa la cláusula WHERE, sino la cláusula **HAVING**.
- Calcular el sueldo máximo de cada departamento siempre y cuando el sueldo sea mayor que 2000.

```
select departamento, max(sueldo)
from Empleados
GROUP BY departamento
HAVING max(sueldo) > 2000
```

Jerarquías

- Las Jerarquías se modelan mediante claves foraneas.
- Son las relaciones reflexivas



CODIGO	NOMBRE	DIRECTOR
1001	Manuel Enciso	Null
1200	Enrique Soler	1001
4000	Carlos Rossi	1001
1045	Sergio Galvez	1200

Jerarquías

- SQL presenta limitaciones a la hora de consultar una jerarquía.
- Extraer la lista de todos los profesores y sus respectivos directores

```
select P.nombre "PROFESOR", D.nombre "DIRECTOR" from  
profesor P, profesor D  
where p.director = d.codigo
```

PROFESOR	DIRECTOR
Enrique Soler	Manuel Enciso
Carlos Rossi	Manuel Enciso
Sergio Galvez	Enrique Soler

- No muestra la jerarquía ni podemos especificar cuál es la raíz

Jerarquías

- Posible solución: Poner un nombre para especificar la raíz.
- Extraer la información de Manuel Enciso y de todos los profesores en la cadena de directores

```
select P.nombre "PROFESOR", D.nombre "DIRECTOR"  
from profesor P, profesor D where  
p.director = d.codigo AND P.nombre = 'Manuel Enciso'
```

PROFESOR	DIRECTOR
----------	----------

- No obtenemos ninguna información porque Enciso no tiene director

Jerarquías

- Problemas con los recorridos jerárquicos:
 - No se puede establecer un nodo de comienzo
 - No se puede extraer la raíz del árbol (porque su atributo conector es nulo)
 - No salen las tuplas ordenadas según el recorrido del árbol
- Solución: La cláusula **CONNECT BY**
 - Permite definir de forma opcional un nodo de comienzo
 - Permite especificar la condición de reunión (recursividad)

Jerarquías

- Extraer la lista de todos los profesores y sus respectivos directores

`select nombre, director from profesor start with
nombre = 'Manuel Enciso'`

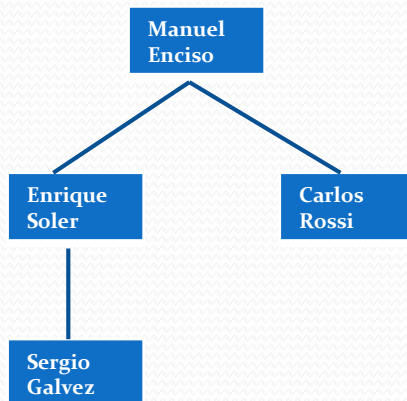
`connect by PRIOR codigo = director`

- **PRIOR**: Es el que marca en qué dirección se recorre el árbol
- **START WITH**: Indica la raíz del árbol. Si no indicamos nada, cada tupla actuará como raíz.

Jerarquías

- Ejemplo1: Cadena de directores hacia abajo empezando por Manuel Enciso

select nombre, director from profesor start with nombre = 'Manuel Enciso' connect by PRIOR codigo = director

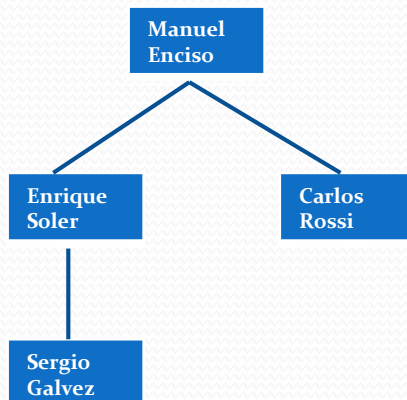


PROFESOR	DIRECTOR
Manuel Enciso	null
Enrique Soler	1001
Sergio Galvez	1200
Carlos Rossi	1001

Jerarquías

- Ejemplo2: Cadena de directores hacia arriba empezando por Manuel Enciso

select nombre, director from profesor start with nombre = 'Manuel Enciso' connect by PRIOR director = codigo

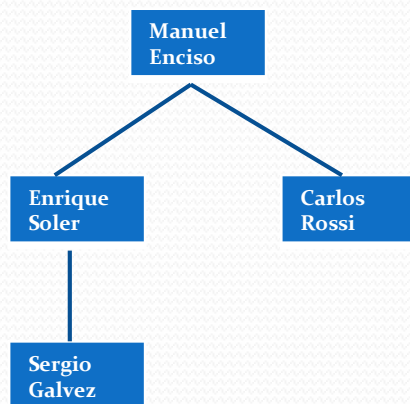


PROFESOR	DIRECTOR
Manuel Enciso	null

Jerarquías

- Ejemplo3: Cadena de directores hacia abajo empezando por Enrique Soler

select nombre, director from profesor start with nombre = 'Enrique Soler' connect by PRIOR codigo = director

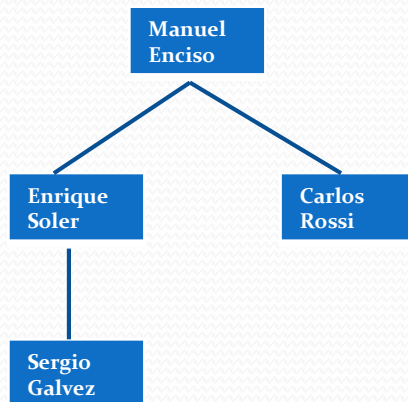


PROFESOR	DIRECTOR
Enrique Soler	1001
Sergio Galvez	1200

Jerarquías

- Ejemplo3: Cadena de directores hacia arriba empezando por Enrique Soler

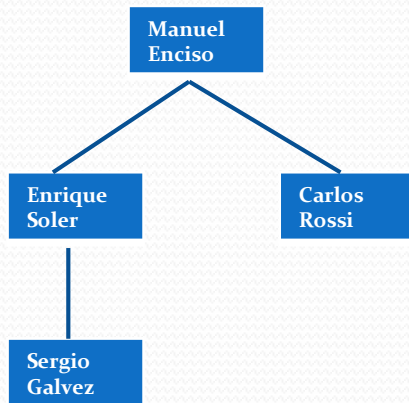
select nombre, director from profesor start with nombre = 'Enrique Soler' connect by PRIOR director = codigo



PROFESOR	DIRECTOR
Enrique Soler	1001
Manuel Enciso	null

Jerarquías

- Si omitimos la cláusula start with



PRIOR **codigo = director**

PROFESOR	DIRECTOR
Enrique Soler	1001
Sergio Galvez	1200
Carlos Rossi	1001
Sergio Galvez	1200
Manuel Enciso	Null
Enrique Soler	1001
Sergio Galvez	1200
Carlos Rossi	1001

PRIOR **director = codigo**

PROFESOR	DIRECTOR
Manuel Enciso	Null
Sergio Galvez	1200
Enrique Soler	1001
Manuel Enciso	Null
Enrique Soler	1001
Manuel Enciso	Null
Carlos Rossi	1001
Manuel Enciso	Null

Jerarquías

- **LEVEL**: Indica el nivel del árbol

select nombre, director, LEVEL from profesor start with nombre = 'Manuel Enciso' connect by prior codigo = director

PROFESOR	DIRECTOR	LEVEL
Manuel Enciso	null	1
Enrique Soler	1001	2
Sergio Galvez	1200	3
Carlos Rossi	1001	2

Jerarquías

- Orden de Ejecución
 1. Clausula WHERE si son de reunión
 2. Start with. Nodos raíz de la consulta
 3. Connect by. Nodos hijos del siguiente nivel
 4. Cláusula where que no tiene reunión

Jerarquías

- Diferencia entre condición en el WHERE o en el CONNECT BY
- Mostrar la cadena de directores empezando por Manuel Enciso que son TU

SELECT nombre, director

FROM profesor

WHERE categoria = 'T.U'

START WITH nombre = 'Manuel Enciso'

CONNECT BY prior codigo = director

jerarquías

CODIGO	NOMBRE	DIRECTOR	CATEGORIA
1001	Manuel Enciso	Null	CU
1200	Enrique Soler	1001	TU
4000	Carlos Rossi	1001	TU
1045	Sergio Galvez	1200	TU

PROFESOR	DIRECTOR
Enrique Soler	1001
Sergio Galvez	1200
Carlos Rossi	1001

- No obtenemos la raíz porque no cumple la condición.
Se genera la jerarquía completa y luego se filtra

Jerarquías

- Mostrar la cadena de directores empezando por Manuel Enciso que son TU

SELECT nombre, director

FROM profesor

START WITH nombre = 'Manuel Enciso'

CONNECT BY prior codigo = director and categoria = 'T.U'

jerarquías

CODIGO	NOMBRE	DIRECTOR	CATEGORIA
1001	Manuel Enciso	Null	CU
1200	Enrique Soler	1001	TU
4000	Carlos Rossi	1001	TU
1045	Sergio Galvez	1200	TU

PROFESOR	DIRECTOR
Manuel Enciso	null
Enrique Soler	1001
Sergio Galvez	1200
Carlos Rossi	1001

- Sale Manuel Enciso porque la raíz sale antes de la cláusula where

jerarquías

- Cambiamos la categoría de Sergio Galvez a TEU

CODIGO	NOMBRE	DIRECTOR	CATEGORIA
1001	Manuel Enciso	Null	CU
1200	Enrique Soler	1001	TU
4000	Carlos Rossi	1001	TU
1045	Sergio Galvez	1200	TEU

PROFESOR	DIRECTOR
Manuel Enciso	null
Enrique Soler	1001
Carlos Rossi	1001

- No sale Sergio Gálvez porque no cumple la condición. Sale Manuel Enciso porque la raíz sale antes de la cláusula where.

Ejercicios

- Extraer la nota media de 'Bases de Datos' y el número de alumnos matriculados en la asignatura
- Extraer la nota media y el número de alumnos matriculados en cada una de las asignaturas
- Extraer la nota media y el número de alumnos matriculados en cada una de las asignaturas para las asignaturas de los grupos de tarde que tienen más de 50 alumnos.