



## Unidad 9

### Elaboración de cláusulas avanzadas de selección.



Bases de datos 2011-12



## Agrupamiento de registros

- Un agrupamiento es el tratamiento único de varios registros que poseen un mismo valor de campo. Al realizar un agrupamiento, podemos aplicar a cada grupo de registros una determinada operación:
- Suma, SUM, Media Aritmética AVG, máximo, MAX, mínimo, MIN, Contar, COUNT



Bases de datos 2011-12

## FUNCIONES DE GRUPO



FUNCIONES	DESCRIPCIÓN DE LA FUNCIÓN
SUM( <i>expresión/DISTINCT nombre de columna</i> )	Calcula la suma de valores de la expresión o de la columna indicada dentro del paréntesis, teniendo en cuenta que la cláusula DISTINCT omite valores repetidos.
AVG( <i>expresión/DISTINCT nombre de columna</i> )	Calcula el valor medio de la expresión que se indique dentro del paréntesis, teniendo en cuenta que los valores NULL no son incluidos.
MIN( <i>expresión</i> )	Devuelve el valor mínimo de la expresión que le acompaña.
MAX( <i>expresión</i> )	Devuelve el valor máximo de la expresión que le acompaña.
COUNT( <i>nombre_columna/DISTINCT nombre_de_columna</i> )	Cuenta el número de valores de datos que hay en una columna, sin incluir los valores NULL.
COUNT(*)	Cuenta todas las filas de la tabla, sin considerar que en algunas columnas existan valores NULL.
STDDEV( <i>expresión</i> ) En Access se llama STDEV( <i>expresión</i> )	Calcula la desviación típica para los valores de la expresión.
VARIANCE( <i>expresión</i> ) En Access se llama VAR( <i>expresión</i> )	Calcula la varianza para los valores de la expresión.



Bases de datos 2011-12



## Ejemplo

JOB_ID	LAST_NAME	SALARY
AC_ACCOUNT	Gietz	8300
AC_MGR	Higgins	12000
AD_ASST	Vihinen	4400
AD_PRES	King	24000
AD_VP	De Haan	17000
AD_VP	Kochhar	17000
FI_ACCOUNT	Chen	8200
FI_ACCOUNT	Faviet	9000
FI_ACCOUNT	Poppo	6900
FI_ACCOUNT	Schierro	7700
FI_ACCOUNT	Uman	7800
FI_MGR	Greenberg	12000
HR_REP	Mavris	6500
IT_PROG	Hunold	9000
IT_PROG	Ernst	6000
IT_PROG	Lorentz	4200
IT_PROG	Pataballa	4800

En la tabla empleados queremos calcular el sueldo medio para cada tipo de empleo (job\_id).

Para ello tendremos que agrupar por tipo de empleo, job\_id y aplicar la función avg a los salarios de cada grupo.



Bases de datos 2011-12



## SENTENCIA SELECT

El formato básico de la sentencia select será

```
SELECT .....
FROM ...
GROUP BY columna1, columna2,...
```

Los datos que aparezcan en una select que lleva un group by deben ser: una constante, una función de grupo (SUM, COUNT, ...) o **una columna expresada** en el GROUP BY.



Bases de datos 2011-12



## Ejemplo

JOB_ID	AVG(SALARY)
IT_PROG	5760
AC_MGR	12000
AC_ACCOUNT	8300
ST_MAN	7280
PU_MAN	11000
AD_ASST	4400
AD_VP	17000
SH_CLERK	3215
FL_ACCOUNT	7920
FL_MGR	12000
PU_CLERK	2780
SA_MAN	12200
MK_MAN	13000
PR_REP	10000
AD_PRES	24000
SA_REP	8350

En la tabla empleados queremos calcular el sueldo medio para cada tipo de empleo(job\_id).

```
SELECT JOB_ID, AVG(SALARY)
FROM EMPLOYEES
GROUP BY JOB_ID
```



Bases de datos 2011-12



## Nota

Notar que en la cláusula GROUP BY, debemos poner columnas cuyo valor se repite en la tabla, lo que nos permitirá hacer los grupos. Si agrupamos por una columna cuyo valor no se repite nunca, por ejemplo por la clave primaria, no aparecerán grupos, más bien estos grupos tendrán un único registro



Bases de datos 2011-12



## Ejemplo

EMPLOYEE_ID	FIRST_NAME	JOB_ID
100	Steven	AD_PRES
101	Neena	AD_VP
102	Lex	AD_VP
103	Alexander	IT_PROG
104	Bruce	IT_PROG
105	David	IT_PROG
106	Valli	IT_PROG
107	Diana	IT_PROG
108	Nancy	PL_MGR
109	Daniel	PL_ACCOUNT
110	John	PL_ACCOUNT
111	Ismael	PL_ACCOUNT
112	Jose Manuel	PL_ACCOUNT
113	Luis	PL_ACCOUNT

Si en esta tabla agrupamos por employee\_id, como no vamos a encontrar valores de employee\_id repetidos en la tabla, los grupos nos saldrán con una sola fila, y habrá tantos como empleados distintos. En cambio, si agrupamos por job\_id, claramente se obtienen los distintos grupos



Bases de datos 2011-12



## Ejemplos

### 1. Obtener la masa salarial mensual de todos los empleados.

```
SQL>SELECT SUM(salario)
FROM empleados;
```

En este caso no hemos realizado ningún grupo, o si se prefiere tenemos un solo grupo formado por todas las filas de la tabla empleados.



Bases de datos 2011-12



### 2. Obtener los salarios máximo, mínimo y la diferencia existente entre ambos.

```
SELECT MAX(salario), MIN(salario), MAX(salario)-MIN(salario)
FROM empleados;
```



Bases de datos 2011-12



### 3. Obtener la fecha de alta más reciente.

```
SQL>SELECT MAX(fecha_alta) "Fecha alta"
FROM empleados;
```



Bases de datos 2011-12



## Ejemplos

### 4. Obtener los salarios medios por tipo de trabajo.

```
SQL>SELECT job_id, AVG(salary)
FROM employees
GROUP BY job_id;
```

JOB_ID	AVG(SALARY)
IT_PROG	5760
AC_MGR	12000
AC_ACCOUNT	8300
ST_MAN	7280
PU_MAN	11000
AD_ASST	4400
AD_VP	17000
SH_CLERK	3215
FI_ACCOUNT	7920
FI_MGR	12000
PU_CLERK	2780
SA_MAN	12200
MK_MAN	13000
PR_REP	10000
AD PRES	24000
SA_REP	8350



Bases de datos

**2. Obtener cuántos empleados hay en cada tipo de empleo.**

```
SELECT JOB_ID, COUNT(*)
FROM EMPLOYEES
GROUP BY JOB_ID
```

JOB_ID	COUNT(*)
AC_ACCOUNT	1
AC_MGR	1
AD_ASST	1
AD_PRES	1
AD_VP	2
FI_ACCOUNT	5
FI_MGR	1
HR_REP	1
IT_PROG	5
MK_MAN	1
MK_REP	1
PR_REP	1

Bases de datos

**3. Obtener el empleado que mayor salario tiene dentro de cada oficio, excluyendo al presidente.**

```
SQL>SELECT MAX(salary), JOB_ID
FROM employees
WHERE job_id != 'AD_PRES'
GROUP BY JOB_ID;
```

Bases de datos 2011-12

**Nota**


SQL Oracle realiza los siguientes pasos, De la tabla empleados, selecciona todos aquellos que no son presidente, los agrupa por tipo de oficio (jefe de ventas, representante,...) y finalmente, para cada uno de los grupos, calcula el salario máximo.

```
SQL>SELECT MAX(salario), JOB_ID
FROM empleados
WHERE job_id != 'AD_PRES'
GROUP BY JOB_ID;
```

Bases de datos 2011-12

MAX(SALARY)	JOB_ID
9000	IT_PROG
12000	AC_MGR
8300	AC_ACCOUNT
8200	ST_MAN
11000	PU_MAN
4400	AD_ASST
17000	AD_VP
4200	SH_CLERK
9000	FI_ACCOUNT
12000	FI_MGR
3100	PU_CLERK
14000	SA_MAN


Bases de datos 2011-12




## Función Count

- `Select count(*) from tabla`

Esta consulta devuelve el número de registros de la tabla indicada en el from



Bases de datos 2011-12




## Ejemplo


```
SELECT COUNT ( * )
FROM EMPLOYEES
```

Results Explain Des

COUNT(*)
107



Bases de datos 2011-12



## Ejemplo


- En este caso la consulta devuelve 106 filas ya que hay un empleado que no tiene asignado departamento

☒ Autocommit Rows 50


```
SELECT COUNT(department_id)
FROM EMPLOYEES
```

Results Explain Describe Saved SQL Hi

COUNT(DEPARTMENT_ID)
106



Bases de datos 2011-12




## Ejemplo


Si para cada tipo de trabajo, queremos contar cuantos empleados hay, emplearemos

```
SELECT job_id, COUNT(*)
FROM EMPLOYEES
group by job_id
```


Agrupar por job\_id y luego cuenta cuantas filas hay en cada grupo.




Bases de datos 2011-12



JOB_ID	COUNT(*)
AC_ACCOUNT	1
AC_MGR	1
AD_ASST	1
AD_PRES	1
AD_VP	2
FI_ACCOUNT	5
FI_MGR	1
HR_REP	1
IT_PROG	5
MK_MAN	1
MK_REP	1
PR_REP	1
PU_CLERK	5
PU_MAN	1
SA_MAN	5
SA_REP	30
SH_CLERK	20
ST_CLERK	20
ST_MAN	5




Bases de datos 2011-12




Si para cada tipo de trabajo, queremos contar cuantos empleados están asignados a un departamento, es decir su department\_id es no nulo, emplearemos

```
SELECT job_id, COUNT(department_id)
FROM EMPLOYEES
group by job_id
```




Bases de datos 2011-12




JOB_ID	COUNT(DEPARTMENT_ID)
IT_PROG	5
AC_MGR	1
AC_ACCOUNT	1
ST_MAN	5
PU_MAN	1
AD_ASST	1
AD_VP	2
SH_CLERK	20
FI_ACCOUNT	5
FI_MGR	1
PU_CLERK	5
SA_MAN	5
MK_MAN	1
PR_REP	1
AD_PRES	1
SA_REP	29
MK_REP	1
ST_CLERK	20
HR_REP	1

En este caso, para el tipo de empleo SA\_REP obtenemos 29 y en el caso anterior 30. Esto ocurre porque hay un empleado que es Sales Representative pero que no tiene departamento asignado, por eso no entra en el recuento




Bases de datos 2011-12




## Selección de grupos

Del mismo modo que la cláusula WHERE permite la selección de filas individuales en una sentencia SELECT, la cláusula **HAVING** permite realizar una selección sobre los grupos obtenidos por la cláusula **GROUP BY**.




Bases de datos 2011-12




## Formato de consulta con selección de grupos

```
SELECT funciones_de_columna
FROM lista_de_tablas
WHERE condición_de_selección DE FILAS
INDIVIDUALES
GROUP BY lista_de_columnas_para_agrupar
HAVING condición_de_selección para grupos de
filas
ORDER BY especificaciones_para_ordenar
```



Bases de datos 2011-12




## Ejemplos

1. **Seleccionar los oficios que tengan dos o más empleados:**  


```
SQL> SELECT job_id, count (*)
FROM empleados
GROUP BY job_id
HAVING COUNT (*) >= 2;
```

La cláusula **HAVING** **actúa como un filtro sobre filas agrupadas**, a diferencia de la cláusula **WHERE** que actúa sobre las filas antes de la agrupación.

Notar que en la cláusula Having, se utilizan condiciones de grupo, pueden aparecer funciones de grupo (SUM, COUNT, ...) o bien columnas de agrupación ( es decir columnas que aparezcan en el group by)



Bases de datos 2011-12




## 2. Seleccionar la población total de ASIA y EUROPA

```
SQL> SELECT continente, sum (poblacion)
FROM paises
WHERE continente in ('ASIA', 'EUROPA')
GROUP BY continente
```


- A partir de la tabla sobre la que se realiza la consulta, la cláusula WHERE actúa como un primer filtro que da como resultado una tabla interna cuyas filas cumplen la condición especificada en el WHERE. (En este caso primero se selecciona los países de Asia y Europa)

- La cláusula GROUP BY produce la agrupación de las filas de la segunda tabla, dando como resultado una tercera tabla con dos grupos, los países de Asia y los países de EUROPA.

- Después aplica el SELECT y selecciona el nombre del continente y la suma de su población




Bases de datos 2011-12



## Ejemplo

```
SELECT job_id, count (*), SUM(SALARY)
FROM EMPLOYEES
GROUP BY job_id
HAVING COUNT (*) >= 2
ORDER BY SUM(SALARY)
```

Notar que en Order BY con group by, solo pueden aparecer condiciones de grupo ( o bien funciones de grupo o columnas que aparezcan en el group by)



Bases de datos 2011-12



## Operadores Union, Intersect, minus

- Los operadores UNION, INTERSECT y MINUS son operadores de conjuntos. Los conjuntos son las filas resultantes de cualquier sentencia SELECT válida que permiten combinar los resultados de varios SELECT para obtener un único resultado.



Bases de datos 2011-12



## Formato

```
SELECT .....FROM ..... WHERE.....
  Operador_de_conjunto
SELECT ....FROM.... WHERE.....
```



Bases de datos 2011-12



## UNION

- UNION** combina los resultados de dos consultas. Las filas duplicadas que aparecen se reducen a una fila única.

```
SELECT col1, col2....FROM tabla1 WHERE condición
```

**UNION**

```
SELECT col1, col2,... FROM tabla2 WHERE condición
```



Bases de datos 2011-12



## Nota

- Dos consultas se pueden unir si tienen el mismo número de columnas y dominios compatibles.



Bases de datos 2011-12





## Ejemplo

```
Select nombre from alum  
UNION  
select nombre from nuevos
```



Bases de datos 2011-12



## UNION ALL

- Combina los resultados de dos consultas. Cualquier duplicación de filas que se dé en el resultado final aparecerá en la consulta.



Bases de datos 2011-12



## Ejemplo

```
Select nombre from alum  
UNION  
Select nombre from nuevos
```



Bases de datos 2011-12



## INTERSECT

- Devuelve las filas que son iguales en ambas consultas. Todas las filas duplicadas serán eliminadas antes de la generación del resultado final



Bases de datos 2011-12



## Nota

- Dos consultas se pueden intersecar si tienen el mismo número de columnas y dominios compatibles.



Bases de datos 2011-12



## Formato

```
SELECT col1, col2....FROM tabla1
WHERE condición
```

### **INTERSECT**

```
SELECT col1, col2,... FROM tabla2
WHERE condición;
```



Bases de datos 2011-12



## Ejemplo

```
Select nombre from alum
INTERSECT
Select nombre from antiguos
```



Bases de datos 2011-12



## Operador Minus

- Devuelve aquellas filas que están en la primera select y no están en la segunda select. Las filas duplicadas del primer conjunto se reducirán a una fila única antes de que empiece la comparación con el otro conjunto.



Bases de datos 2011-12



## Nota

- En dos consultas se puede hacer la diferencia si tiene el mismo número de columnas y dominios compatibles.



Bases de datos 2011-12



## Formato

```
SELECT col1, col2, ... FROM tabla1  
WHERE condición
```

### **MINUS**

```
SELECT col1, col2, ... FROM tabla2  
WHERE condición;
```



Bases de datos 2011-12



## Reglas de utilización

Estos operadores se pueden manejar con consultas de diferentes tablas, siempre que se apliquen las siguientes reglas:

- Las columnas de las dos consultas se relacionan en orden, de izquierda a derecha
- Los nombres de las columnas de la primera sentencia select no tienen por qué ser los mismos que los nombres de la segunda
- Las SELECT necesitan tener el mismo número de columnas
- Los tipos de datos deben coincidir, aunque la longitud no tiene que ser la misma



Bases de datos 2011-12