Bases de Datos

FACULTAD DE INFORMÁTICA

Funciones de Grupo

- AVG: Obtiene el valor promedio de la columna
- COUNT: Cuenta el total de filas de una columna
- MAX: Obtiene el valor más alto de la columna
- MIN: Obtiene el valor más bajo de la columna
- SUM: Suma los valores de la columna

Funciones de Grupo

```
SELECT [columnas], group_function(columna)
FROM   tabla
[WHERE condición]
[GROUP BY columna]
[ORDER BY columna];
```

Funciones **AVG** y **SUM**

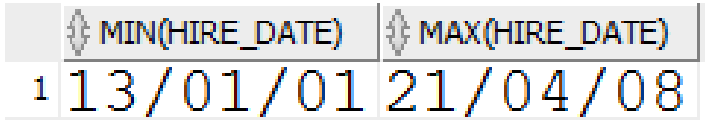
```
SELECT AVG(salary), MAX(salary), MIN(salary), SUM(salary)
FROM   employees
WHERE job_id LIKE '%REP%';
```

	AVG(SALARY)	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
1	8272.72	11500	6000	273000

* Las funciones de grupo AVG y SUM requieren datos numéricos.

Funciones MIN y MAX

```
SELECT MIN(hire_date), MAX(hire_date)
FROM   employees;
```



	MIN(HIRE_DATE)	MAX(HIRE_DATE)
1	13/01/01	21/04/08

* Las funciones de grupo MIN y MAX se pueden usar para cualquier tipo de dato.

Función **COUNT**

```
SELECT COUNT(*)
FROM   employees
WHERE department_id = 50;
```

	COUNT(*)
1	45

Función **COUNT**

- COUNT(*expr*) devuelve el número de filas con valores no nulos

```
SELECT COUNT(commission_pct)
FROM   employees
WHERE department_id = 80;
```

	COUNT(COMMISSION_PCT)
1	34

DISTINCT

- COUNT(DISTINCT *expr*) devuelve el número de valores distintos no nulos

```
SELECT COUNT(DISTINCT department_id)
FROM employees;
```

	COUNT(DISTINCTDEPARTMENT_ID)
1	11

Funciones de Grupo y NULLs

- Las funciones de grupo ignoran los valores nulos de la columna

```
SELECT AVG(commission_pct)
FROM employees;
```

	AVG(COMMISSION_PCT)
1	0.228571428

Funciones para valores NULL

```
SELECT commission_pct
FROM employees;
```

	COMMISSION_PCT
43	(null)
44	(null)
45	(null)
46	0.4
47	0.3
48	0.3
49	0.3
50	0.2
51	0.3
52	0.25

```
SELECT NVL(commission_pct, 0)
 FROM employees;
```

	NVL(COMMISSION_PCT,0)
43	0
44	0
45	0
46	0.4
47	0.3
48	0.3
49	0.3
50	0.2
51	0.3
52	0.25

Funciones de Grupo y NULLs

- Las funciones de grupo ignoran los valores nulos de la columna

```
SELECT AVG(NVL(commission_pct, 0))
FROM employees;
```

	AVG(NVL(COMMISSION_PCT,0))
1	0.0728971962

Cláusula **GROUP BY**

```
SELECT [columnas], group_function(columna)
FROM   tabla
[WHERE condición]
[GROUP BY columna]
[ORDER BY columna];
```

Divide las filas de una tabla en grupos más pequeños.

Cláusula **GROUP BY**

Todas las columnas de la lista SELECT que **no pertenezcan a una función de grupo** deben estar en la cláusula GROUP BY

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id;
```

	DEPARTMENT_ID	TRUNC(AVG(SALARY),2)
1	100	8738
2	30	4150
3	(null)	7000
4	90	21013.33
5	20	9500
6	70	10000
7	110	10154
8	50	3475.55
9	80	8955.88
10	40	6500

Cláusula **GROUP BY**

La columna de la cláusula GROUP BY no tiene que estar en la cláusula SELECT

```
SELECT AVG(salary)
FROM employees
GROUP BY department_id;
```

	TRUNC(AVG(SALARY),2)
1	8738
2	4150
3	7000
4	21013.33
5	9500
6	10000
7	10154
8	3475.55
9	8955.88
10	6500

GROUP BY con varias columnas

```
SELECT department_id, job_id, SUM(salary) NOMINA
FROM employees
GROUP BY department_id, job_id;
```

	DEPARTMENT_ID	JOB_ID	NOMINA
1	110	AC_ACCOUNT	8300
2	90	AD_VP	34000
3	50	ST_CLERK	55700
4	80	SA_REP	243500
5	50	ST_MAN	36400
6	80	SA_MAN	61000
7	110	AC_MGR	12008

Consultas no válidas

```
SELECT department_id, COUNT(last_name)
FROM employees;
```

```
ORA-00937: not a single-group group function
00937. 00000 -  "not a single-group group function"
*Cause:
*Action:
Error en la línea: 1, columna: 8
```

Falta la columna sin función de grupo del SELECT en la cláusula GROUP BY

Consultas no válidas

```
SELECT department_id, COUNT(last_name)
FROM employees
GROUP BY department_id;
```

	DEPARTMENT_ID	COUNT(LAST_NAME)
1	100	6
2	30	6
3	(null)	1
4	90	3
5	20	2
6	70	1
7	110	2

Consultas no válidas

```
SELECT department_id, AVG (salary)
FROM employees
WHERE AVG(salary) > 8000
GROUP BY department_id;
```

```
ORA-00934: group function is not allowed here
00934. 00000 -  "group function is not allowed here"
*Cause:
*Action:
Error en la línea: 3, columna: 7
```

Consultas no válidas

- No se puede utilizar la cláusula WHERE para filtrar **grupos**.
- No se pueden utilizar funciones de grupo en la cláusula WHERE.
- Se utiliza la cláusula HAVING para filtrar grupos.

```
SELECT department_id, AVG(salary)
FROM employees
HAVING AVG(salary) > 8000
GROUP BY department_id;
```

	DEPARTMENT_ID	TRUNC(AVG(SALARY),2)
1	100	8738
2	90	21013.33
3	20	9500
4	70	10000
5	110	10154
6	80	8955.88

Cláusula HAVING

La cláusula HAVING se utiliza para restringir grupos

1. Las filas se agrupan.
2. Se aplica la función de grupo.
3. Se muestran solo los grupos que coincidan con la cláusula HAVING

```
SELECT column, group_function
FROM table
[WHERE condition]
[GROUP BY group_by_expression]
[HAVING group_condition]
[ORDER BY column]
```

Cláusula **HAVING**

```
SELECT department_id, MAX(salary)
FROM employees
GROUP BY department_id
HAVING MAX(salary) > 10000;
```

	DEPARTMENT_ID	MAX(SALARY)
1	100	12008
2	30	11000
3	90	29040
4	20	13000
5	110	12008
6	80	14000

Cláusula **HAVING**

```
SELECT job_id, SUM(salary) NOMINA
FROM employees
WHERE job_id NOT LIKE '%REP%'
GROUP BY job_id
HAVING SUM(salary) > 13000;
```

	JOB_ID	NOMINA
1	PU_CLERK	13900
2	IT_PROG	28800
3	AD_PRES	29040
4	AD_VP	34000
5	ST_MAN	36400
6	FI_ACCOUNT	40420
7	ST_CLERK	55700
8	SA_MAN	61000
9	SH_CLERK	64300