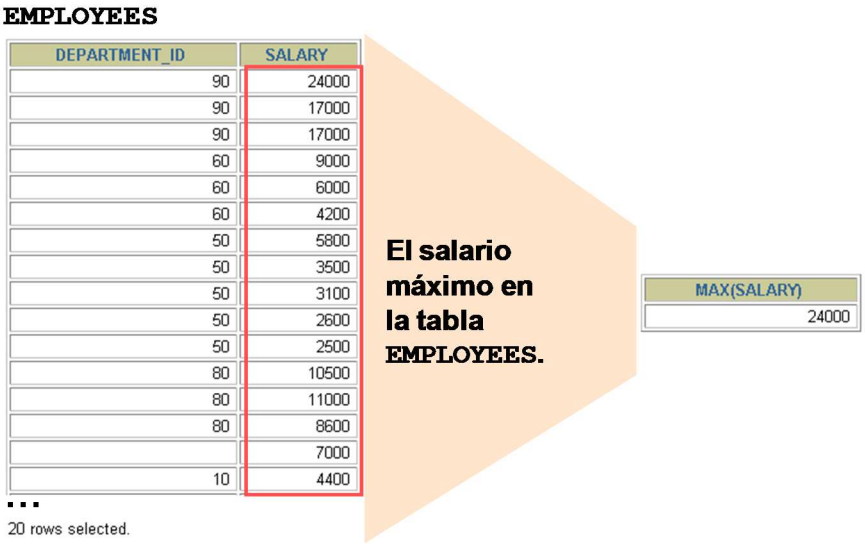
UNIDAD 5: FUNCIONES DE GRUPO

A diferencia de las funciones de una sola fila, las funciones de grupo operan sobre juegos de filas para proporcionar un resultado por grupo. Estos juegos pueden ser la tabla completa o la tabla dividida en grupos.



SINTAXIS DE LA FUNCIONES DE GRUPO

```
SELECT      [column,] group_function(column), ...
FROM        table
[WHERE      condition]
[GROUP BY   column]
[ORDER BY   column];
```

- DISTINCT hace que la función solamente considere valores no duplicados; ALL hace que considere todos los valores incluyendo duplicados. El valor por defecto es ALL y, por lo tanto, no es necesario especificarlo.
- Los tipos de dato para las funciones con un argumento expr pueden ser CHAR, VARCHAR2, NUMBER o DATE.
- Todas las funciones de grupo ignoran los valores nulos. Para sustituir un valor por valores nulos, utilice las funciones NVL, NVL2 o COALESCE.
- Oracle Server ordena implícitamente el juego de resultados en orden ascendente al utilizar una cláusula GROUP BY. Para sustituir este orden por defecto, se puede utilizar DESC en una cláusula ORDER BY.

USO DE LAS FUNCIONES AVG Y SUM

Puede utilizar las funciones AVG, SUM, MIN y MAX en columnas que almacenan datos numéricos. En el ejemplo de la transparencia se muestra la media, el salario mayor, el menor y la suma de los salarios mensuales de todos los representantes de ventas.

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

AVG(SALARY)	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
8150	11000	6000	32600

USO DE LAS FUNCIONES MINSU Y MAX

Puede utilizar las funciones MAX y MIN para cualquier tipo de dato. En el ejemplo de la transparencia se muestra al empleado de menor edad y al de mayor edad.

En el siguiente ejemplo se muestra el apellido del primer y del último empleado de una lista alfabética de todos los empleados.

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

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

MIN(HIRE_	MAX(HIRE_
17-JUN-87	29-JAN-00

USO DE LA FUNCION COUNT

La función COUNT tiene tres formatos:

- COUNT(*)
- COUNT(expr)
- COUNT(DISTINCT expr)

COUNT(*) devuelve el número de filas de una tabla que satisface los criterios de la sentencia SELECT, incluyendo filas duplicadas y filas que contengan valores nulos en cualquiera de las columnas. Si se incluye una cláusula WHERE en la sentencia SELECT, COUNT(*) devuelve el número de filas que satisface la condición de la cláusula WHERE..

En contraste, COUNT(*expr*) devuelve el número de valores no nulos de la columna identificada por *expr*.

COUNT(DISTINCT *expr*) devuelve el número de valores únicos no nulos de la columna identificada por *expr*.

En el ejemplo siguiente se muestra el número de empleados del departamento 50.

COUNT(*) devuelve el número de filas de una tabla.

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

COUNT(*)
5

En este ejemplo se muestra el número de empleados del departamento 80 que pueden percibir una comisión

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

COUNT(COMMISSION_PCT)
3

Otro ejemplo: mostrar el número total de empleados de la tabla EMPLOYEES

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

USO DE LA PALABRA RESERVADA DISTINCT

Utilice la palabra clave DISTINCT para suprimir el recuento de cualquier valor duplicado dentro de una columna. En el ejemplo de la figura se muestra el número de valores de departamento distintos de la tabla EMPLOYEES.

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

COUNT(DISTINCTDEPARTMENT_ID)
7

FUNCIONES DE GRUPO Y VALORES NULOS

Todas las funciones de grupo ignoran los valores nulos de la columna. En el ejemplo de la transparencia, la media se calcula basándose *solamente* en las filas de la tabla que almacenan un valor válido en la columna COMMISSION_PCT. La media se calcula como la comisión total pagada a todos los empleados dividida por el número de empleados que perciben una comisión (cuatro).

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

AVG(COMMISSION_PCT)
.2125

USO DE LA FUNCION NVL CON FUNCIONES DE GRUPO

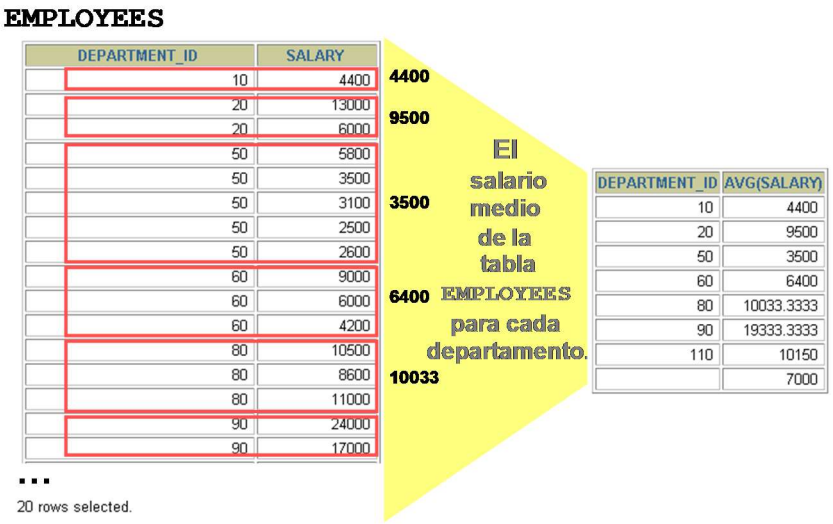
La función NVL fuerza a que las funciones de grupo incluyan valores nulos. En el ejemplo de la transparencia, la media se calcula basándose en *todas* las filas de la tabla, independientemente de si se almacenan valores nulos en la columna COMMISSION_PCT. La media se calcula como la comisión total pagada a todos los empleados dividida por el número total de empleados de la compañía (20).

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

AVG(NVL(COMMISSION_PCT,0))
.0425

CREACION DE GRUPOS DE DATOS

Hasta ahora, todas las funciones de grupo han tratado la tabla como un gran grupo de información. A veces, necesita dividir la tabla de información en grupos más pequeños. Esto se puede realizar utilizando la cláusula GROUP BY.



CREACION DE GRUPOS DE DATOS: SINTAXIS DE LA CLAUSULA GROUP BY

Puede utilizar la cláusula GROUP BY para dividir las filas de una tabla en grupos. A continuación, puede utilizar las funciones de grupo para devolver información de resumen para cada grupo.

En la sintaxis:

group_by_expression especifica columnas cuyos valores determinan la base para agrupar filas.

Instrucciones

- Si incluye una función de grupo en una cláusula SELECT, no puede seleccionar también resultados individuales, a menos que la columna individual aparezca en la cláusula GROUP BY. Recibirá un mensaje de error si no incluye la lista de columnas en la cláusula GROUP BY.
- Si utiliza una cláusula WHERE, puede excluir filas antes de dividirlas en grupos.
- Debe incluir las columnas en la cláusula GROUP BY.
- No se puede utilizar un alias de columna en la cláusula GROUP BY.
- Por defecto, las filas se ordenan en orden ascendente de las columnas incluidas en la lista GROUP BY. Puede sustituir este orden por defecto utilizando la cláusula ORDER BY.

```
SELECT      column, group_function(column)
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[ORDER BY   column];
```

Divida las filas de una tabla en grupos más pequeños utilizando la cláusula `GROUP BY`.

USO DE LA CLAUSULA GROUP BY

Al utilizar la cláusula GROUP BY, asegúrese de que todas las columnas de la lista SELECT que no son funciones de grupo están incluidas en la cláusula GROUP BY. En el ejemplo de la transparencia se muestra el número de departamento y el salario medio para cada departamento. A continuación, se muestra el modo en que se evalúa esta sentencia SELECT, que contiene una cláusula GROUP BY:

- La cláusula SELECT especifica las columnas que se van a recuperar:
  - Columna de número de departamento de la tabla EMPLOYEES
  - La media de todos los salarios en el grupo que ha especificado en la cláusula GROUP BY
- La cláusula FROM especifica las tablas a las que debe acceder la base de datos: la tabla EMPLOYEES.
- La cláusula WHERE especifica las filas que se van a recuperar. Como no hay ninguna cláusula WHERE, se recuperan todas las filas por defecto.
- La cláusula GROUP BY especifica cómo se deben agrupar las filas. Las filas se agrupan por número de departamento, por lo que la función AVG que se está aplicando a la columna de salario calculará el salario medio de cada departamento.

Todas las columnas de la lista `SELECT` que no estén en las funciones de grupo deben estar en la cláusula `GROUP BY`.

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

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333
90	19333.3333
110	10150
	7000

8 rows selected.

La columna GROUP BY no tiene que estar en la cláusula SELECT. Por ejemplo, la sentencia SELECT de la transparencia muestra los salarios medios para cada departamento sin mostrar los números de los departamentos respectivos. Sin embargo, sin los números de departamento, los resultados no parecen significativos.

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

La columna GROUP BY no tiene que estar en la lista SELECT.

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

AVG(SALARY)
4400
9500
3500
6400
10033.3333
19333.3333
10150
7000

AGRUPACION POR MÁS DE UNA COLUMNA

A veces necesita ver resultados para grupos dentro de grupos. En la transparencia se muestra un informe con el salario total que se paga a cada cargo, dentro de cada departamento.

La tabla EMPLOYEES se agrupa en primer lugar por número de departamento y, dentro de dicha agrupación, por cargo. Por ejemplo, los cuatro agentes de bolsa del departamento 50 están agrupados y se produce un resultado único (salario total) para todos los agentes de bolsa dentro del grupo.

EMPLOYEES

DEPARTMENT_ID	JOB_ID	SALARY
90	AD_PRES	24000
90	AD_VP	17000
90	AD_VP	17000
60	IT_PROG	9000
60	IT_PROG	6000
60	IT_PROG	4200
50	ST_MAN	5800
50	ST_CLERK	3500
50	ST_CLERK	3100
50	ST_CLERK	2600
50	ST_CLERK	2500
80	SA_MAN	10500
80	SA_REP	11000
80	SA_REP	8600
...		
20	MK_REP	6000
110	AC_MGR	12000
110	AC_ACCOUNT	8300

20 rows selected.

"Sume los salarios de la tabla EMPLOYEES para cada cargo, agrupado por departamento.

DEPARTMENT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD_PRES	24000
90	AD_VP	34000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
	SA_REP	7000

13 rows selected.

USO DE LA CLAUSULA GROUP BY EN VARIAS COLUMNAS

Puede devolver resultados resumidos para grupos y subgrupos incluyendo en una lista más de una columna GROUP BY. Puede determinar el orden por defecto de los resultados por el orden de las columnas en la cláusula GROUP BY. A continuación, se muestra el modo en que se evalúa la sentencia SELECT de la transparencia, que contiene una cláusula GROUP BY:

- La cláusula SELECT especifica la columna que se va a recuperar:
  - Número de departamento de la tabla EMPLOYEES
  - Identificador de cargo de la tabla EMPLOYEES
  - La suma de todos los salarios en el grupo que ha especificado en la cláusula GROUP BY
- La cláusula FROM especifica las tablas a las que debe acceder la base de datos: la tabla EMPLOYEES.
- La cláusula GROUP BY especifica cómo debe agrupar las filas:
  - En primer lugar, las filas se agrupan por número de departamento.
  - En segundo lugar, dentro de los grupos de números de departamento, las filas se agrupan por identificador de cargo.

De esta forma, la función SUM se está aplicando a la columna de salario para todos los identificadores de cargo dentro de cada grupo de números de departamento.

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

DEPT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD_PRES	24000
90	AD_VP	34000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
	SA_REP	7000

13 rows selected.

CONSULTAS NO VALIDAS UTILIZANDO FUNCIONES DE GRUPO

Siempre que utilice una mezcla de elementos individuales (DEPARTMENT_ID) y funciones de grupo (COUNT) en la misma sentencia SELECT, debe incluir una cláusula GROUP BY que especifique los elementos individuales (en este caso, DEPARTMENT_ID). Si falta la cláusula GROUP BY, aparecerá el mensaje de error "not a single-group group function" y un asterisco (*) señalará la columna incorrecta.

Toda columna o expresión de la lista `SELECT` que no sea una función agregada debe estar en la cláusula `GROUP BY`.

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

```
SELECT department_id, COUNT(last_name)
       *
ERROR at line 1:
ORA-00937: not a single-group group function
```

La columna falta en la cláusula `GROUP BY`

Puede corregir el error de la transparencia agregando la cláusula GROUP BY.

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

La cláusula WHERE no se puede utilizar para restringir grupos. La sentencia SELECT de la figura produce un error, pues utiliza la cláusula WHERE para restringir la visualización de salarios medios de los departamentos que tienen un salario medio mayor que $8000

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

```
WHERE  AVG(salary) > 8000
       *
ERROR at line 3:
ORA-00934: group function is not allowed here
```

No se puede utilizar la cláusula `WHERE` para restringir grupos

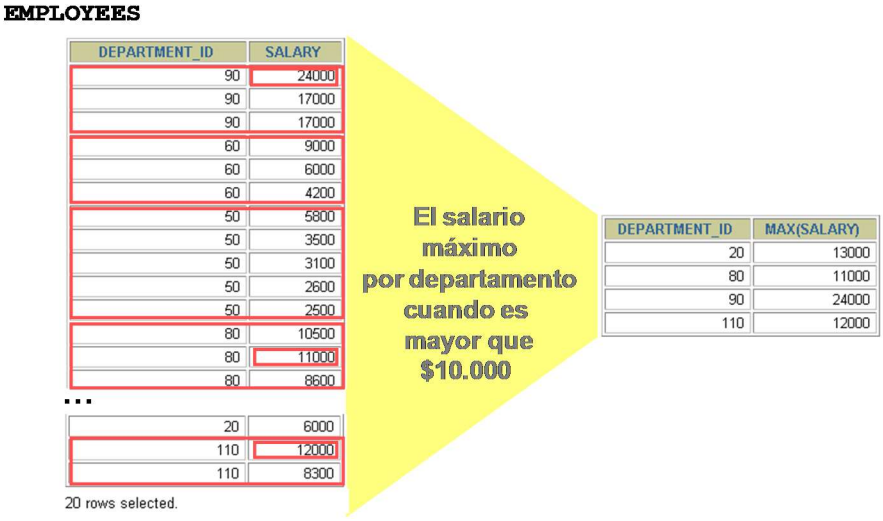
Se puede corregir el error de la transparencia utilizando la cláusula HAVING para restringir grupos:

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

EXCLUSION DE RESULTADOS DE GRUPO

De la misma forma que utiliza la cláusula WHERE para restringir las filas que selecciona, utilice la cláusula HAVING para restringir grupos. Para buscar el salario máximo de cada departamento, pero mostrando solamente los departamentos que tengan un salario máximo de más de $10.000, debe:

- Buscar el salario medio para cada departamento agrupando por número de departamento.
- Restringir los grupos a los departamentos con un salario máximo mayor que $10.000



EXCLUSION DE RESULTADOS DE GRUPO: CLAUSULA HAVING

Utilice la cláusula HAVING para especificar qué grupos se van a mostrar y, de esta forma, restringirá aún más los grupos sobre la base de información agregada.

En la sintaxis:

group_condition restringe los grupos de filas devueltas a los grupos para los que es verdadera la condición especificada.

Oracle Server realiza los siguientes pasos cuando se utiliza la cláusula HAVING:

- Las filas se agrupan.
- Se aplica al grupo la función de grupo.
- Se muestran los grupos que coinciden con los criterios de la cláusula HAVING.

La cláusula HAVING puede preceder a la cláusula GROUP BY, pero se recomienda que coloque en primer lugar la cláusula GROUP BY porque es más lógico. Se forman grupos y se calculan las funciones de grupo antes de que se aplique la cláusula HAVING a los grupos de la lista SELECT.

Utilice la cláusula `HAVING` para restringir grupos:

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

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

USO DE LA CLAUSULA HAVING

En el ejemplo de la figura se muestran números de departamento y salarios máximos para los departamentos cuyos salarios máximos son mayores que $10.000.

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

DEPARTMENT_ID	MAX(SALARY)
20	13000
80	11000
90	24000
110	12000

Puede utilizar la cláusula GROUP BY sin utilizar una función de grupo en la lista SELECT.

Si restringe las filas basándose en el resultado de una función de grupo, debe tener una cláusula GROUP BY además de la cláusula HAVING.

En el siguiente ejemplo se muestran los números de departamento y los salarios medios para los departamentos cuyos salarios máximos sean mayores que $10.000:

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

En el ejemplo de la siguiente figura se muestra el identificador de cargo y el salario mensual total para cada cargo con una nómina total superior a $13.000. El ejemplo excluye a representantes de ventas y ordena la lista según el salario mensual total.

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

JOB_ID	PAYROLL
IT_PROG	19200
AD_PRES	24000
AD_VP	34000

ANIDAMIENTO DE LAS FUNCIONES DE GRUPO

Las funciones de grupo se pueden anidar hasta una profundidad de dos. En el ejemplo de la transparencia se muestra el salario medio máximo.

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

MAX(AVG(SALARY))
19333.3333