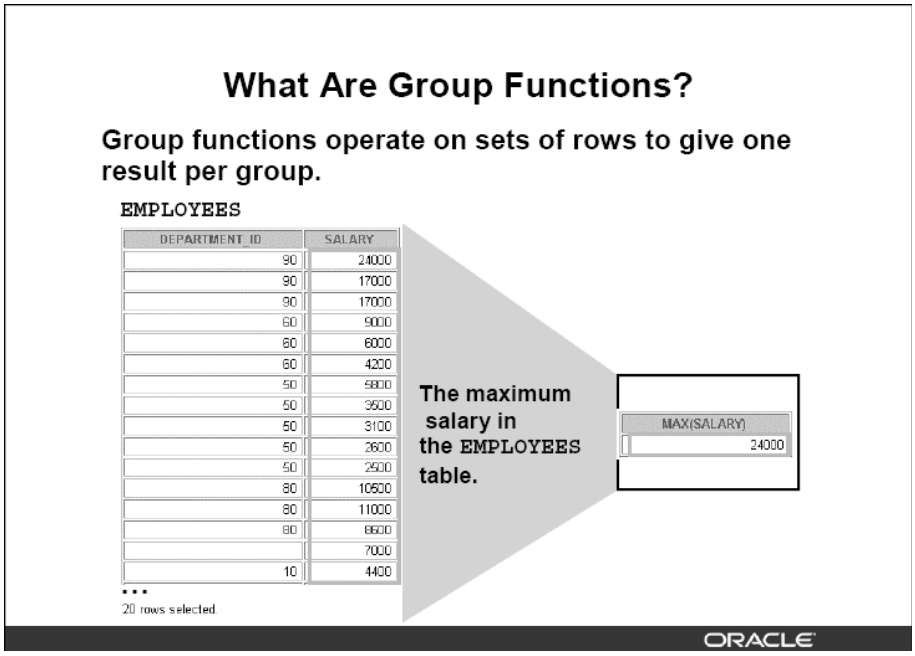
5.- Agregando datos utilizando funciones de grupo

Objetivo

Después de terminar este capítulo, conocerá lo siguiente:

- Identificar la disponibilidad de las funciones de grupo
- Describir el uso de las funciones de grupo
- Agrupar datos utilizando la cláusula GROUP BY
- Incluir o excluir filas agrupadas utilizando la cláusula HAVING



Funciones de Grupo

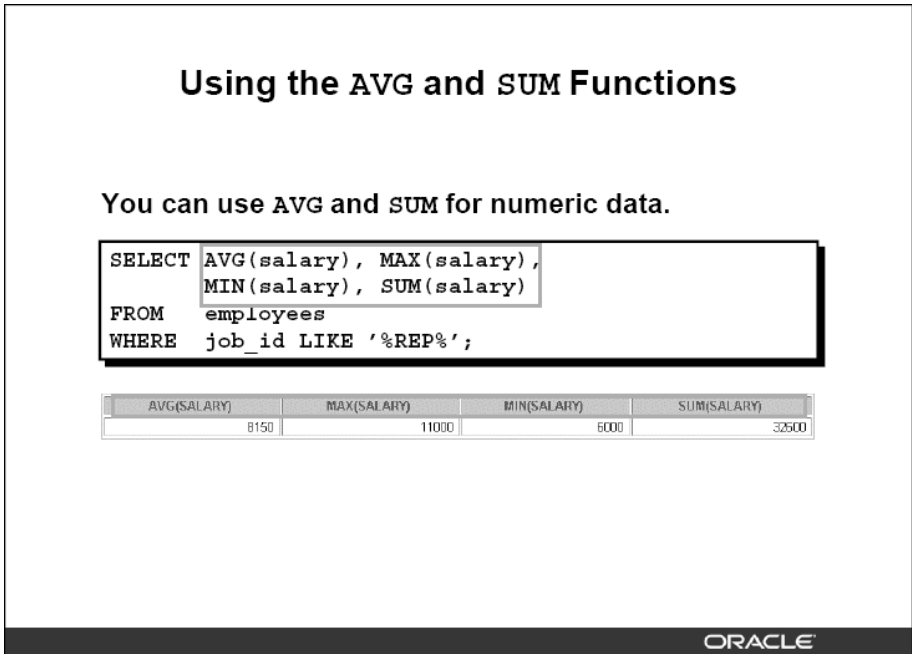
A diferencia de las funciones de línea única, las funciones de grupo funcionan en un conjunto de filas proporcionando un resultado por grupo. Estos conjuntos pueden ser de toda una tabla o fracciones de la tabla dentro de cada grupo.

Cada una de las funciones acepta un argumento. La siguiente tabla identifica las opciones que pueden utilizarse en su sintaxis:

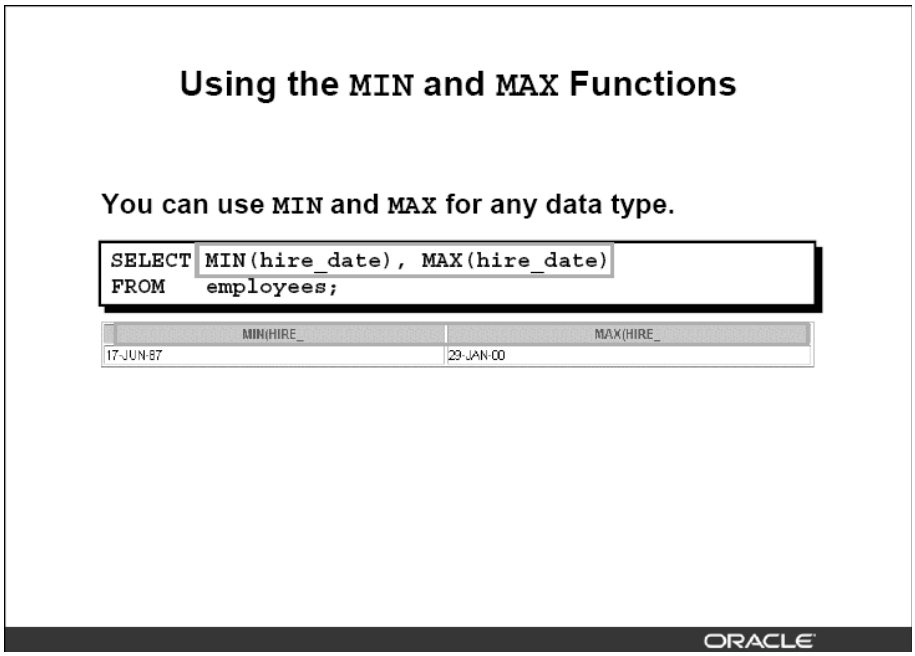
Función	Descripción
AVG([DISTINCT|ALL] *n*)	Promedia el valor de *n*, ignorando valores nulos
COUNT({* | [DISTINCT | ALL] *expr*})	Número de filas, donde *expr* es evaluado sin considerar nulos (Cuenta todas las filas usando *, incluyendo duplicadas y con valores nulos)
MAX([DISTINCT | ALL] *expr*)	Máximo valor de *expr,* ignorando valores nulos
MIN([DISTINCT | ALL] *expr*)	Mínimo valor de *expr,* ignorando valores nulos
STDDEV([DISTINCT | ALL] *n*)	Desviación estándar de *n*, ignorando valores nulos
SUM([DISTINCT | ALL] *n*)	Suma los valores de *n*, ignorando valores nulos
VARIANCE([DISTINCT | ALL] *n*)	Varianza de *n*, ignorando valores nulos

Normas para el uso de funciones de grupo

- DISTINCT hace que la función solo considere los valores no duplicados; ALL hace considera todos los valores incluyendo los duplicados. Por defecto es ALL y por consiguiente no necesita ser especificado.
- Los tipos de datos para las funciones con argumento *expr* pueden ser CHAR, VARCHAR2, NUMBER o DATE.
- Todas las funciones de grupo ignoran los valores nulos. Para sustituir un valor de un valor nulo utilice las funciones NVL, NVL2 o COALESCE.
- El servidor Oracle implícitamente ordena el resultado ascendentemente cuando se utiliza la cláusula GROUP BY. Para el ordenamiento por defecto, DESC puede ser usado en la cláusula ORDER BY.



Se pueden usar las funciones AVG, SUM, MIN y MAX con columnas que pueden almacenar datos numéricos. En el ejemplo anterior se despliega el promedio, el mayor, el menor y la suma de los salarios mensuales de todos los representantes de ventas.



Se pueden emplear las funciones MAX y MIN para cualquier tipo de dato. En el ejemplo se muestra al más reciente y más viejo empleado.
El siguiente ejemplo muestra el primer y último apellido de acuerdo al orden alfabético de todos los empleados.

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

MIN(LAST_NAME)	MAX(LAST_NAME)
Abel	Zlotkey

Nota: Las funciones AVG, SUM, VARIANCE y STDDEV pueden ser usadas únicamente con tipos de datos numéricos.

Función COUNT

COUNT(*) returns the number of rows in a table.

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

COUNT(*)
5

La función COUNT tiene tres formatos:

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

COUNT(*) regresa el número de filas en una tabla que satisface el criterio de la sentencia SELECT, incluyendo filas duplicadas y filas conteniendo valores nulos en alguna de sus columnas. Si una cláusula WHERE en la sentencia

SELECT, COUNT(*) regresa el número de filas que satisface la condición en la cláusula WHERE.
En contraste, COUNT(*expr*) regresa el número de valores no nulos en la columna identificada por *expr.*
COUNT(DISTINCT *expr*) regresa el número de valores no nulos y únicos en la columna identificada por *expr.*

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

COUNT(*)
5

En el ejemplo anterior se despliega el número de empleados en el departamento 50.

En el siguiente ejemplo se muestra el número de empleados en el departamento 80 que tienen una comisión.

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

COUNT(COMMISSION_PCT)
3

La palabra reservada DISTINCT

Use la palabra reservada DISTINCT para suprimir la cuenta de cualquier valor duplicado dentro de una columna. En el siguiente ejemplo se muestra el número de departamentos distintos en la tabla empleados.

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

COUNT(DISTINCTDEPARTMENT_ID)
7

Funciones de grupo y valores nulos

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

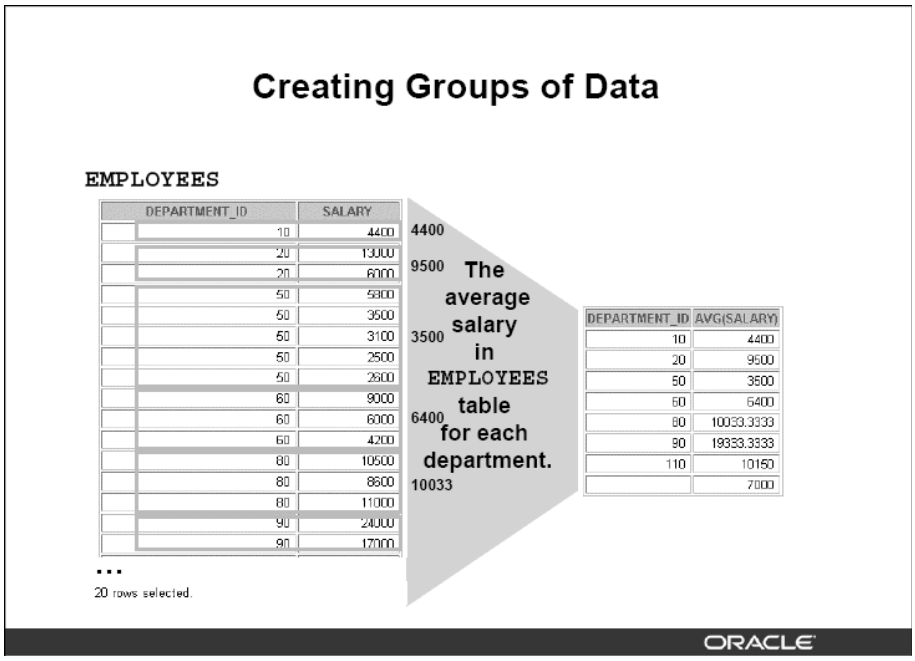
AVG(COMMISSION_PCT)
.2125

Todas las funciones de grupo ignoran los valores nulos en las columnas. En el ejemplo anterior, el promedio es calculado basado únicamente en las filas de la tabla donde un valor válido esté almacenado en la columna COMMISSION_PCT. El promedio es calculado como la comisión total pagada a todos los empleados dividido por el número de empleados que recibieron una comisión (4).

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

AVG(NVL(COMMISSION_PCT,0))
.0425

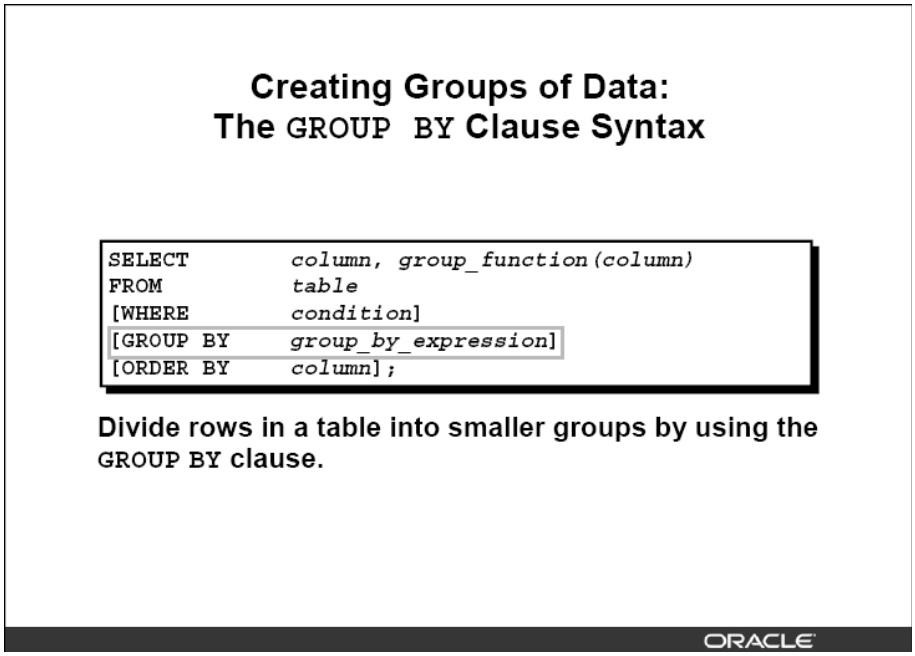
La función NVL forza al grupo de funciones a incluir los valores nulos. En el ejemplo anterior, el promedio es calculado considerando todas las columnas de la tabla, independientemente de que los valores nulos están en la columna COMMISSION_PCT. El promedio es calculado como la comisión total pagada a todos los empleados dividida por el número total de empleados en la compañía (20).



Grupos de datos

Hasta ahora, todas las funciones de grupo han tratado a las tablas como un largo grupo de información. En ocasiones, es necesario dividir la información de una tabla en grupos más pequeños. Esto puede ser realizado con el uso de la cláusula GROUP BY.

La cláusula GROUP BY

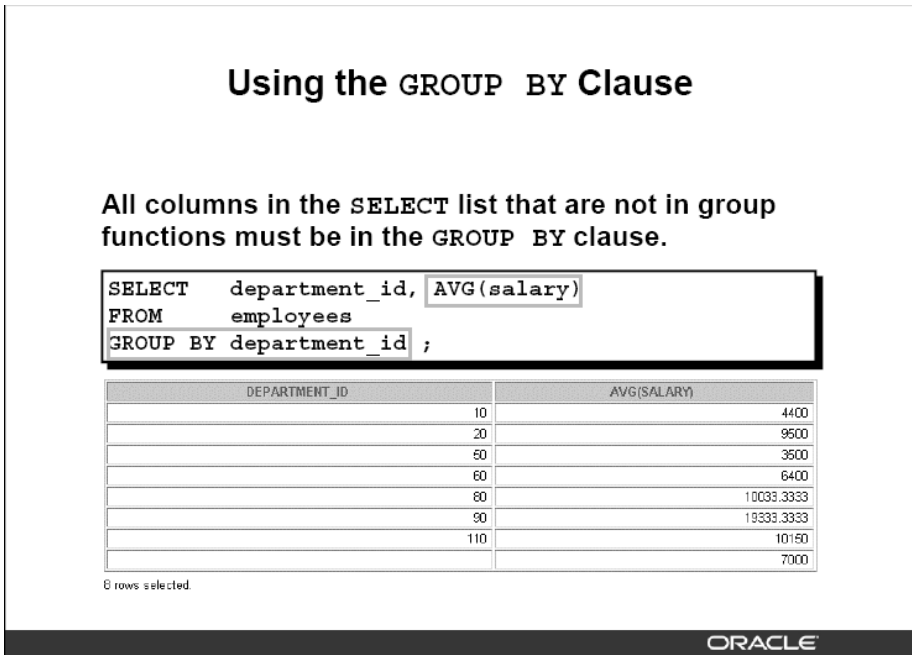


Se puede utilizar la cláusula GROUP BY para dividir las filas de una tabla en grupos. Se pueden emplear las funciones de grupo para obtener información resumida por cada grupo.
En la sintaxis:

g*roup_by_expression*	Especifica las columnas cuyos valores determinan las bases para la agrupación.

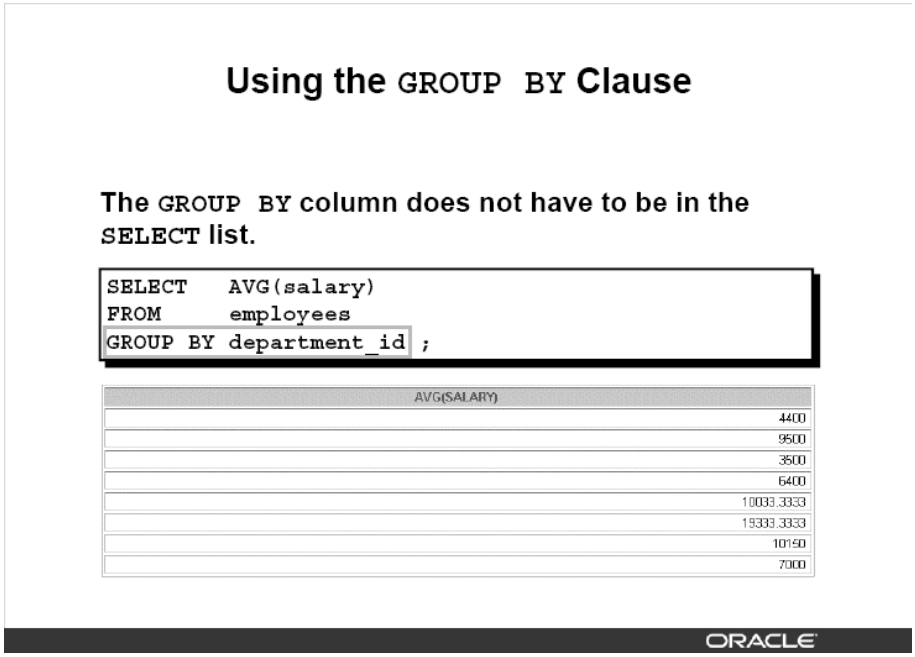
Normas a seguir

- Si incluyes una función de grupo en una cláusula SELECT, no se podrá seleccionar resultados individuales, a menos que la columna aparezca en la cláusula GROUP BY. Se obtendrá un mensaje de error si no se incluye dicha columna en la cláusula GROUP BY.
- Utilizando la cláusula WHERE, se pueden excluir filas antes de dividir estas en grupos.
- Se deben incluir las columnas en la cláusula GROUP BY.
- No se pueden utilizar alias de columnas en la cláusula GROUP BY.
- Por defecto, las filas son ordenadas ascendentemente en el orden de las columnas incluidas en la lista de la cláusula GROUP BY. Se puede sobrescribir este ordenamiento usando la cláusula ORDER BY.



Cuando se use la cláusula GROUP BY, asegúrese que todas las columnas en la lista del SELECT que no sean funciones de grupo, sean incluidas en la cláusula GROUP BY. En el ejemplo anterior se muestra el número de departamento y el salario promedio de cada departamento. Así es como esta sentencia SELECT, conteniendo una cláusula GROUP BY, es evaluada:

- La cláusula SELECT específica las columnas que serán recuperadas:
  - La columna número de departamento de la tabla EMPLOYEES
  - El promedio de todos los salarios en el grupo especificado en la cláusula GROUP BY
- La cláusula FROM especifica las tablas de la base de datos que se deben accesar: la tabla EMPLOYEES.
- La cláusula WHERE especifica las filas que serán recuperadas. Si no es especificada la cláusula WHERE, todas las filas son recuperadas por defecto.
- La cláusula GROUP BY especifica como las filas serán agrupadas. Las filas serán inicialmente agrupadas por número de departamento, así que la función AVG se iniciará para ser aplicada a la columna salary para calcular el salario promedio por cada departamento.

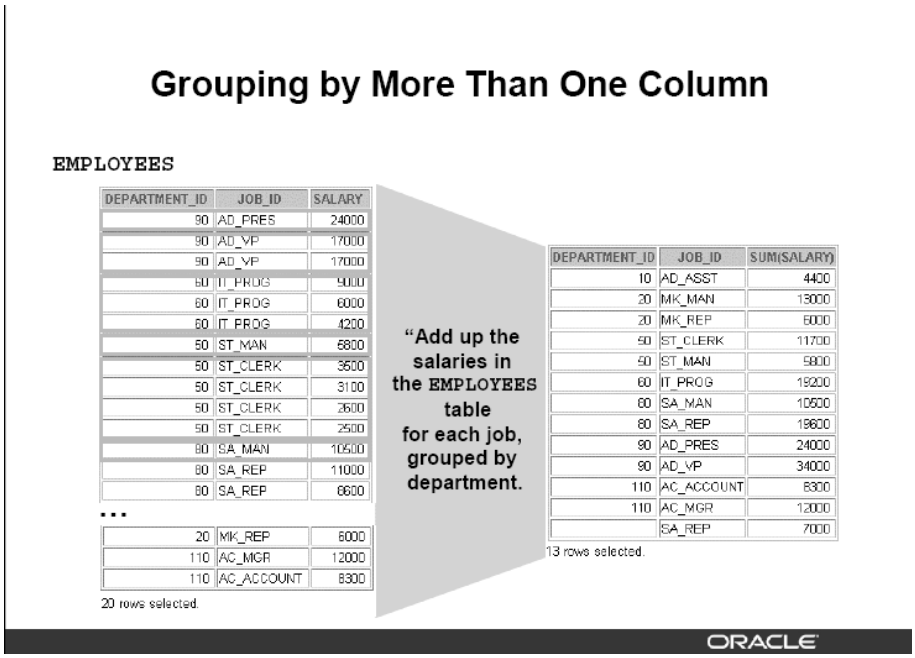


Las columnas incluidas en la cláusula GROUP BY no es necesario que estén en la cláusula SELECT. Por ejemplo, La sentencia SELECT anterior despliega el salario promedio por cada departamento, sin mostrar el respectivo número de departamento. Sin embargo, sin el número de departamento, el resultado no será significativo.

Se pueden usar las funciones de grupo en la cláusula ORDER BY.

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

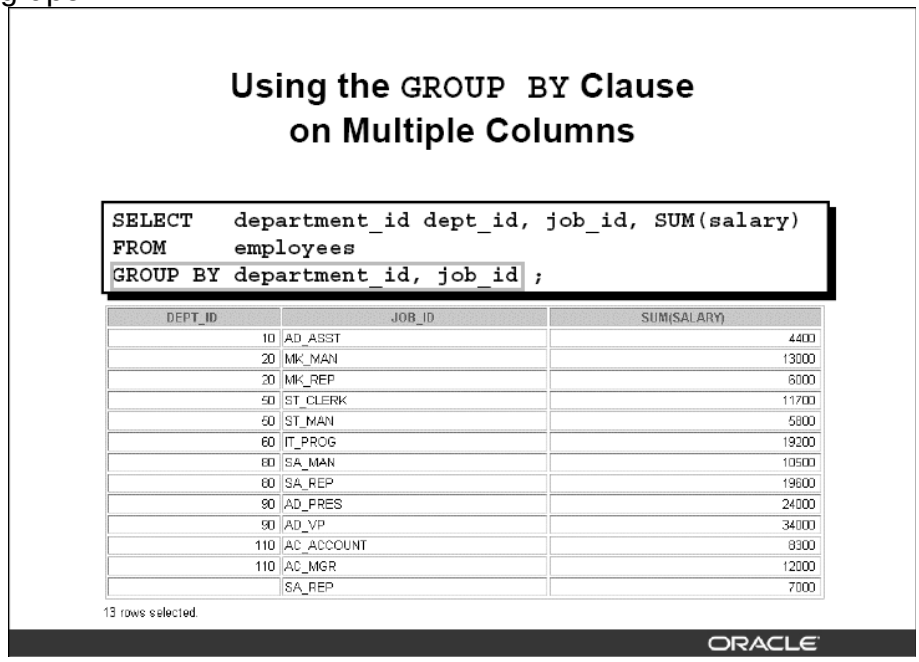
DEPARTMENT_ID	AVG(SALARY)
50	3500
10	4400
60	6400
...	
90	19333.3333



Grupos dentro de grupos

En algunas ocasiones es necesario ver los resultados de grupos dentro de otro grupo. En la imagen anterior se muestra un reporte que despliega el salario total pagado por cada puesto, dentro de cada departamento.

La tabla EMPLOYEES es agrupada primeramente por el número de departamento y, dentro de esta agrupación, por puesto. Por ejemplo, los cuatro empleados de almacén del departamento 50 están agrupados y un único resultado (el salario total) es producido para todos los empleados de almacén de ese grupo.

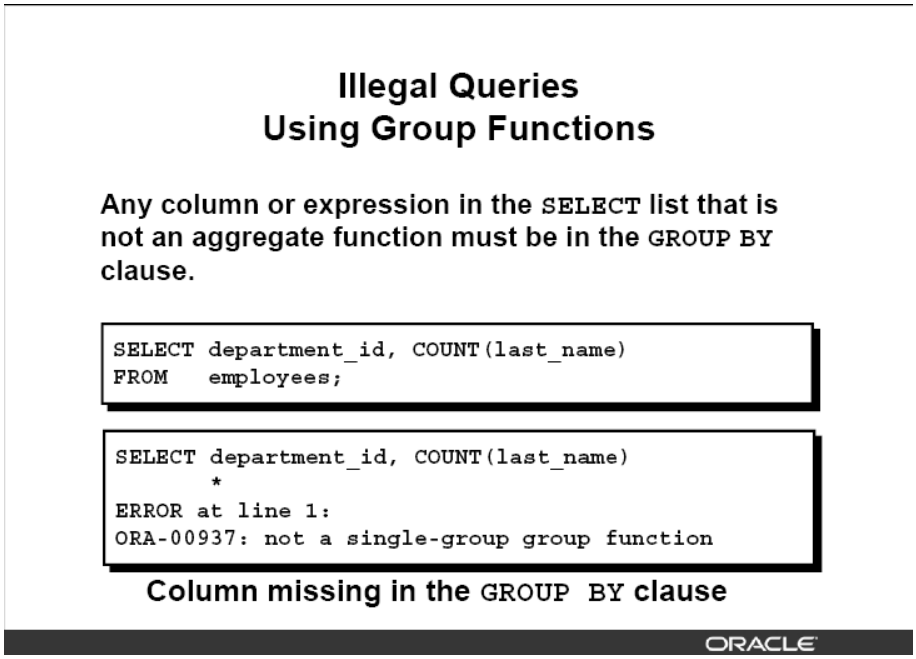


Se pueden obtener resultados resumidos para grupos y subgrupos por la lista de más de una columna en el GROUP BY. También es posible determinar el ordenamiento por defecto de los resultados por medio del orden de las

columnas en la cláusula GROUP BY. Así es como la sentencia SELECT de la imagen anterior es evaluada.

- La cláusula SELECT especifica las columnas que serán recuperadas:
  - El número de departamento de la tabla EMPLOYEES
  - El identificador del puesto de la tabla EMPLOYEES
  - La suma de todos los salarios en el grupo especificado en la cláusula GROUP BY
- La cláusula FROM especifica las tablas de la base de datos a las que se tendrá acceso: Tabla EMPLOYEES
- La cláusula GROUP BY especifica como se deben agrupar las filas:
  - Primero, las filas son agrupadas por número de departamento.
  - Segundo, con los grupos por número de departamento, las filas son agrupadas por puesto.

Por lo que la función SUM es aplicada a la columna salary para todos los puestos dentro de cada grupo de departamento.



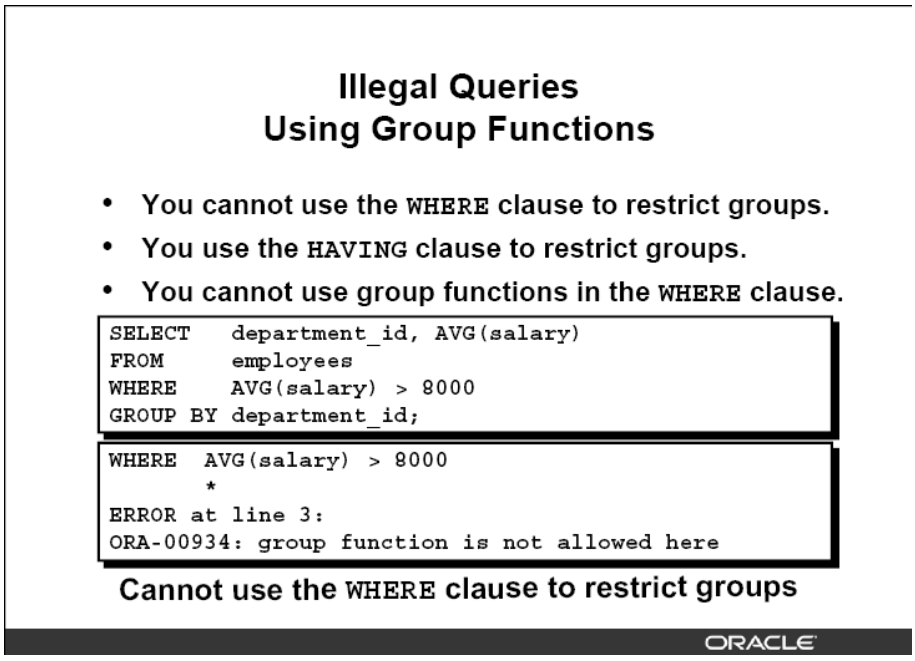
Consultas ilegales usando funciones de grupo

Cuantas veces se utilice una combinación de elementos individuales (DEPARTMENT_ID) y funciones de grupo (COUNT) en la misma sentencia SELECT, se debe incluir una cláusula GROUP BY que especifique el elemento individual (en este caso, DEPARTMENT_ID). Si la cláusula GROUP BY está ausente, entonces el mensaje de error "*not a single-group group function"* aparece y un asterisco (*) apunta a la columna que lo ocasiona. Se puede corregir este error agregando la cláusula GROUP BY.

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

DEPARTMENT_ID	COUNT(LAST_NAME)
10	1
20	2
...	
	1

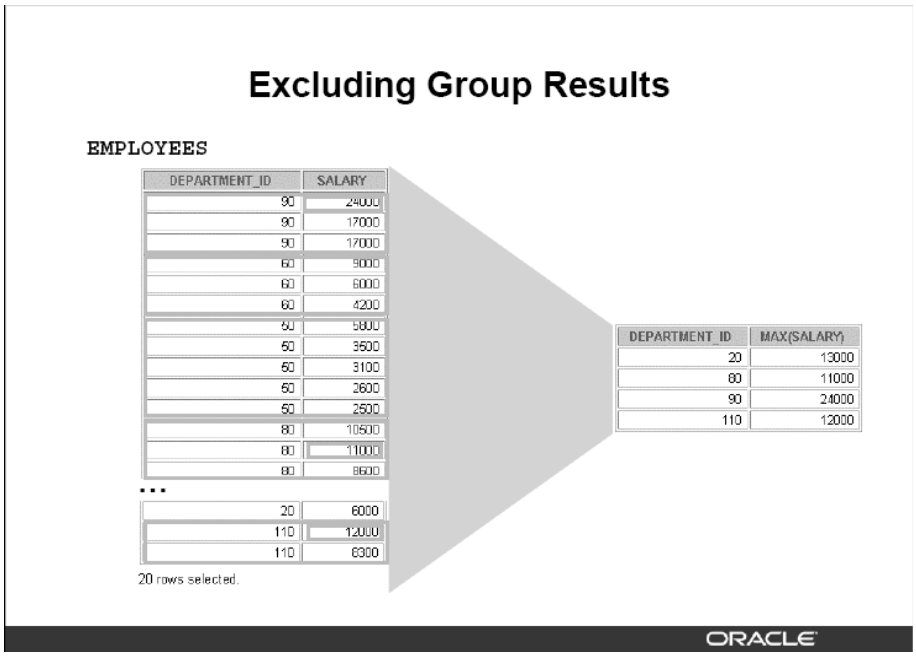
Algunas columnas o expresiones en la lista SELECT que no forman parte de una función deben estar en la cláusula GROUP BY.



La cláusula WHERE no puede ser usada para restringir grupos. La sentencia SELECT de la imagen anterior obtiene un error porque usa una cláusula WHERE para restringir que se muestren los salarios promedio de aquellos departamentos que tienen un salario promedio mayor a $8,000.
Para corregir el error se utiliza la cláusula HAVING que restringe grupos.

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

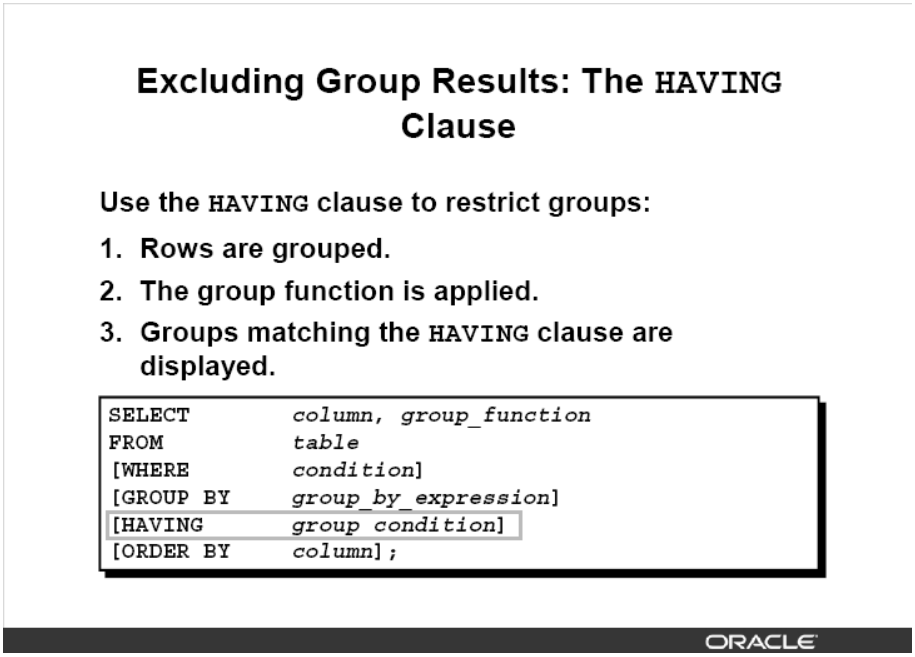
DEPARTMENT_ID	AVG(SALARY)
20	9500
80	10033.3333
90	19333.3333
110	10150



Restringiendo resultados de grupo

De la misma forma en la que se utiliza la cláusula WHERE para restringir las filas que se seleccionan, se puede usar la cláusula HAVING para restringir grupos. Para encontrar el salario máximo de cada departamento, pero solo mostrar los departamentos que tienen un salario máximo de más de $10,000, se necesita hacer lo siguiente:

1. Encontrar el salario promedio de cada departamento agrupando por número de departamento.
2. Restringir los grupos de aquellos departamentos con un salario máximo mayor a $10,000.



Cláusula HAVING

Se puede utilizar la cláusula HAVING para especificar que grupos serán mostrados, y así, restringir los grupos base de información adicional.
En la sintaxis:

g*roup_condition*	Restringe los grupos de filas obtenidas de aquellos grupos donde la condición especificada es verdadera

El servidor de Oracle ejecuta los siguientes pasos cuando se usa la cláusula SELECT:

1. Agrupa las filas
2. Las funciones de grupo son aplicadas al grupo
3. Los grupos que cumplen el criterio de la cláusula HAVING son mostrados

La cláusula HAVING puede ser precedida de la cláusula GROUP BY, pero es recomendado que la cláusula GROUP BY este primero puesto que es más lógico. Los grupos son formados y las funciones de grupo calculadas, antes que la cláusula HAVING sea aplicada.

Si se desea restringir filas basadas en el resultado de una función de grupo, se debe tener tanto una cláusula GROUP BY como una cláusula HAVING.

En el siguiente ejemplo se muestra el número de departamento y salario promedio de aquellos departamentos cuyo salario máximo es mayor de $10,000:

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

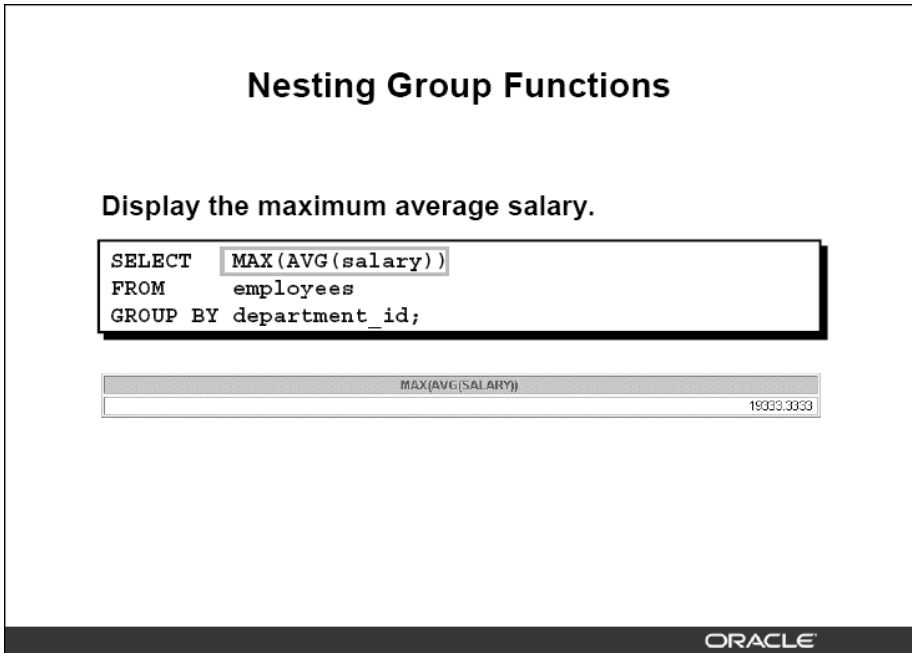
DEPARTMENT_ID	AVG(SALARY)
20	9500
80	10033.3333
90	19333.3333
110	10150

En el ejemplo siguiente, se despliega el puesto y el salario total mensual para cada puesto con una nómina total arriba de $13,000.

```
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

En el ejemplo se excluye al representante de ventas y se ordenan el resultado por el salario total mensual.



Funciones de grupo anidadas

Las funciones de grupo pueden ser anidadas hasta dos niveles. En el ejemplo anterior se muestra el salario promedio máximo.

Resumen

Siete funciones de grupo son disponibles en SQL:

- AVG
- COUNT
- MAX
- MIN
- SUM
- STDDEV
- VARIANCE

Se pueden crear subgrupos por medio de la cláusula GROUP BY. Los grupos pueden ser excluidos utilizando la cláusula HAVING.

Escriba las cláusulas HAVING y GROUP BY después de la cláusula WHERE en una sentencia. Escriba la cláusula ORDER BY al final.

El servidor Oracle evalúa las cláusulas en el siguiente orden:

1. Si la sentencia contiene una cláusula WHERE, el servidor establece las filas candidatas.
2. El servidor identifica los grupos especificados en la cláusula GROUP BY.
3. La cláusula HAVING restringe los resultados de los grupos que no cumplen el criterio establecido.

Práctica 5

Al finalizar esta práctica, se familiarizara con el uso de las funciones de grupo y seleccionara grupos de datos.

Nota: Alias de columnas son utilizadas para las consultas.

Determine la validez de las siguientes tres sentencias.

1. Las funciones de grupo trabajan con muchas filas para producir un resultado por grupo
   CIERTO FALSO

2. Las funciones de grupo incluyen valores NULOS en sus cálculos
   CIERTO FALSO

3. La cláusula WHERE restringe las filas antes de incluirlas en un calculo de grupo
   CIERTO FALSO

4. Despliegue el salario mayor, salario menor, la suma y promedio del salario de todos los empleados. Etiquete las columnas como Maximum, Minimum, Sum y Avarage respectivamente. Redondeé los resultados al número entero más cercano. Guarde la sentencia SELECT en el archivo lab5_4.sql

Maximum	Minimum	Sum	Average
24000	2500	175500	8775

5. Modifique la consulta del lab5_4.sql para desplegar el mínimo, máximo, la suma, y promedio de la columna salario por cada tipo de puesto. Guarde la sentencia en el archivo lab5_5.sql y ejecútelo.

JOB_ID	Maximum	Minimum	Sum	Average
AC_ACCOUNT	8300	8300	8300	8300
AC_MGR	12000	12000	12000	12000
AD_ASST	4400	4400	4400	4400
AD_PRES	24000	24000	24000	24000
AD_VP	17000	17000	34000	17000
IT_PROG	9000	4200	19200	6400
MK_MAN	13000	13000	13000	13000
MK_REP	6000	6000	6000	6000
SA_MAN	10500	10500	10500	10500
SA_REP	11000	7000	26600	8867
ST_CLERK	3500	2500	11700	2925
ST_MAN	5800	5800	5800	5800

6. Escriba una consulta que despliegue el número de gente con el mismo puesto

JOB_ID	COUNT(*)
AC_ACCOUNT	1
AC_MGR	1
AD_ASST	1
AD_PRES	1
AD_VP	2
IT_PROG	3
MK_MAN	1
MK_REP	1
SA_MAN	1
SA_REP	3
ST_CLERK	4
ST_MAN	1

7. Determine el número de jefes. Etiquete la columna como “Number of Managers”. Sugerencia: Utilice la columna MANAGER_ID para determinar el número de jefes.

Number of Managers
8

8. Escriba una consulta que despliegue la diferencia entre el más alto y más bajo salario. Etiquete la columna como DIFFERENCE.

DIFFERENCE
21500

Si tienes tiempo, completa los siguientes ejercicios

9. Despliega el número del jefe y el salario del empleado peor pagado de todos los jefes. Excluya cualquiera cuyo jefe no es conocido. Excluye cualquier grupo donde el salario mínimo sea menor a $6,000. Ordene el resultado descendentemente por la columna salary.

MANAGER_ID	MIN(SALARY)
102	9000
205	8300
149	7000

10. Escribe una consulta que despliegue cada nombre de departamento, localización, número de empleados y salario promedio de todos los empleados en los departamentos. Etiquete las columnas como Name, Location, Number of People y Salary respectivamente. Redondeé el salario promedio a 2 decimales.

Name	Location	Number of People	Salary
Accounting	1700	2	10150
Administration	1700	1	4400
Executive	1700	3	19333.33
IT	1400	3	6400
Marketing	1800	2	9500
Sales	2500	3	10033.33
Shipping	1500	5	3500

Si tienes tiempo extra, completa los siguientes ejercicios:

11. Cree una consulta que permita desplegar el número total de empleados y, del total, el número de empleados contratados en 1995, 1996, 1997 y 1998.

TOTAL	1995	1996	1997	1998
20	1	2	2	3

12. Genere una consulta matriz que despliegue el puesto, el salario para ese puesto basado en el numero de departamento y el salario total para el puesto por departamentos 20, 50, 80 y 90, tomando para cada columna una etiqueta apropiada.

Job	Dept 20	Dept 50	Dept 80	Dept 90	Total
AC_ACCOUNT					8300
AC_MGR					12000
AD_ASST					4400
AD_PRES				24000	24000
AD_VP				34000	34000
IT_PROG					19200
MK_MAN	13000				13000
MK_REP	6000				6000
SA_MAN			10500		10500
SA_REP			19600		26600
ST_CLERK		11700			11700
ST_MAN		5800			5800