



2

Restricción y Ordenación de Datos

ORACLE®
DATABASE

11^g

Objetivos

Al finalizar esta lección, debería estar capacitado para:

- **Limitar las filas recuperadas por una consulta**
- **Ordenar las filas recuperadas por una consulta**

Limitación de Filas Mediante una Selección

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
104	Ernst	IT_PROG	60
105	Austin	IT_PROG	60
106	Pataballa	IT_PROG	60
107	Lorentz	IT_PROG	60
108	Greenberg	FI_MGR	100
109	Faviet	FI_ACCOUNT	100

...

20 rows selected.

**“recuperar todos
los empleados del
departamento 90”**



EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

Limitación de las Filas Seleccionadas

- Restrinja las filas devueltas utilizando la cláusula **WHERE** .

```
SELECT      * | {DISTINCT} column|expression [alias],...}  
FROM        table  
[WHERE      condition (s)];
```

- La cláusula **WHERE** sigue a la cláusula **FROM**.

Uso de cláusula WHERE

```
SELECT  employee_id, last_name, job_id, department_id
FROM    employees
WHERE   department_id = 90;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

Cadenas de Caracteres y Fechas

- Las cadenas de caracteres y los valores de fechas se escriben entre comillas simples.
- Los valores de caracteres son sensibles a mayúsculas/minúsculas y los de fecha, al formato.
- El formato de fecha por defecto es DD-MON-RR.

```
SELECT last_name, job_id, department_id
FROM employees
WHERE last_name = 'Whalen';
```

Condiciones de Comparación

Operador	Significado
=	Igual que
>	Mayor que
>=	Mayor o igual que
<	Menor que
<=	Menor o igual que
<>	No igual a

Uso de Condiciones de Comparación

```
SELECT last_name, salary
FROM employees
WHERE salary <= 3000;
```

LAST_NAME	SALARY
Baida	2900
Tobias	2800
Himuro	2600
Colmenares	2500
Mikkilineni	2700
Landry	2400
Markle	2200
Atkinson	2800

Otras condiciones de Comparación

Operador	Significado
BETWEEN ...AND...	Entre dos valores (ambos inclusive)
IN (set)	Coincide con cualquiera de una lista de valores
LIKE	Coincide con un patrón de caracteres
IS NULL	Es un valor nulo

Uso de la condición BETWEEN

Utilice la condición BETWEEN para mostrar filas incluidas en un rango de valores.

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 2500 AND 3500;
```

Limite inferior

Limite superior

LAST_NAME	SALARY
Khoo	3100
Baida	2900
Tobias	2800
Himuro	2600
Colmenares	2500
Nayer	3200
Mikkilineni	2700
Bissot	3300
Atkinson	2800
Marlow	2500
Mallin	3300
Rogers	2900

Uso de la Condición IN

Utilice la condición **IN** para comprobar si hay valores en una lista.

```
SELECT employee_id, last_name, salary, manager_id
FROM   employees
WHERE  manager_id IN (100, 101, 201);
```

EMPLOYEE_ID	LAST_NAME	SALARY	MANAGER_ID
101	Kochhar	17000	100
102	De Haan	17000	100
108	Greenberg	12000	101
114	Raphaely	11000	100
120	Weiss	8000	100
121	Fripp	8200	100
122	Kaufling	7900	100
123	Vollman	6500	100
124	Mourgos	5800	100
145	Russell	14000	100
146	Partners	13500	100

Uso de la condición LIKE

- Utilice la condición LIKE para realizar búsquedas con comodines de valores de la cadena de búsqueda.
- Las condiciones de búsqueda pueden contener caracteres literales o números:
 - % indica cero o muchos caracteres cualesquiera
 - _ indica un carácter cualquiera

```
SELECT    first_name
FROM      employees
WHERE     first_name LIKE 's%';
```

Uso de la condición LIKE

- Puede combinar caracteres de coincidencia de patrones.

```
SELECT    last_name
FROM      employees
WHERE     last_name LIKE '_o%';
```

LAST_NAME
Kochhar
Lorentz
Popp

- Puede utilizar el identificador ESCAPE para buscar los símbolos % y _ reales.

```
SELECT    employee_id, last_name, job_id
FROM      employees
WHERE     job_id LIKE '%SA\_%' ESCAPE '\';
```

Uso de las condiciones NULL

Compruebe si hay valores nulos con el operador IS NULL.

```
SELECT  last_name, manager_id
FROM    employees
WHERE   manager_id IS NULL;
```

LAST_NAME	MANAGER_ID
King	

Condiciones Lógicas

Operador	Significado
AND	Devuelve TRUE si las <i>dos</i> condiciones componentes son verdaderas
OR	Devuelve TRUE si <i>alguna</i> de las condiciones componentes es verdadera
NOT	Devuelve TRUE si la siguiente condición es falsa

Uso del Operador AND

AND requiere que las dos condiciones sean verdaderas.

```
SELECT  employee_id, last_name, job_id, salary
FROM    employees
WHERE    salary >= 10000
AND      job_id LIKE '%MAN%';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
114	Raphaely	PU_MAN	11000
145	Russell	SA_MAN	14000
146	Partners	SA_MAN	13500
147	Errazuriz	SA_MAN	12000
148	Cambrault	SA_MAN	11000
149	Zlotkey	SA_MAN	10500
201	Hartstein	MK_MAN	13000

Uso del Operador OR

OR requiere que una de las condiciones sea verdadera.

```
SELECT  employee_id, last_name, job_id, salary
FROM    employees
WHERE    salary >= 10000
OR       job_id LIKE '%MAN%';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
102	De Haan	AD_VP	17000
108	Greenberg	FI_MGR	12000
114	Raphaely	PU_MAN	11000
120	Weiss	ST_MAN	8000
121	Fripp	ST_MAN	8200
122	Kaufling	ST_MAN	7900
123	Vollman	ST_MAN	6500
124	Mourgos	ST_MAN	5800

Uso del Operador NOT

```
SELECT  last_name, job_id
FROM    employees
WHERE   job_id
        NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP');
```

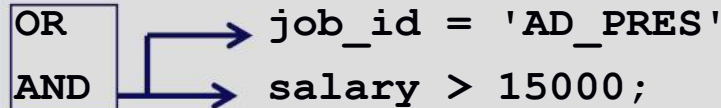
LAST_NAME	JOB_ID
King	AD_PRES
Kochhar	AD_VP
De Haan	AD_VP
Greenberg	FI_MGR
Faviet	FI_ACCOUNT
Chen	FI_ACCOUNT
Sciarra	FI_ACCOUNT
Urman	FI_ACCOUNT
Popp	FI_ACCOUNT
Raphaely	PU_MAN

Reglas de prioridad

Orden de Evaluación	Operador
1	Operadores aritméticos
2	Operador de Concatenación
3	Condiciones de comparación
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Condición lógica NOT
7	Condición lógica AND
8	Condición lógica OR

Reglas de Prioridad

```
SELECT    last_name, job_id, salary
FROM      employees
WHERE     job_id = 'SA_REP'
OR        job_id = 'AD_PRES'
AND       salary > 15000;
```

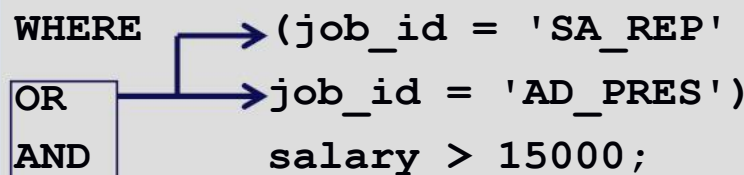


LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000
Tucker	SA_REP	10000
Bernstein	SA_REP	9500
Hall	SA_REP	9000
Olsen	SA_REP	8000
Cambrault	SA_REP	7500
Tuvault	SA_REP	7000
King	SA_REP	10000
Sully	SA_REP	9500
McEwen	SA_REP	9000

Reglas de Prioridad

Utilice paréntesis para forzar la prioridad.

```
SELECT      last_name, job_id, salary
FROM        employees
WHERE       (job_id = 'SA_REP'
OR          job_id = 'AD_PRES')
AND         salary > 15000;
```



LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000

Cláusula ORDER BY

- Ordene filas con la cláusula ORDER BY
 - ASC: orden ascendente, por defecto
 - DESC: orden descendente
- La cláusula ORDER BY se coloca en último lugar en la sentencia SELECT.

```
SELECT    last_name, job_id, department_id, hire_date
FROM      employees
ORDER BY  hire_date;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DAT
King	AD_PRES	90	17/06/87
Whalen	AD_ASST	10	17/09/87
Kochhar	AD_VP	90	21/09/89
Hunold	IT_PROG	60	03/01/90
Ernst	IT_PROG	60	21/05/91
De Haan	AD_VP	90	13/01/93
Mavris	HR_REP	40	07/06/94

Ordenación en Orden Descendente

```
SELECT    last_name, job_id, department_id, hire_date
FROM      employees
ORDER BY  hire_date DESC;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DAT
Banda	SA_REP	80	21/04/00
Kumar	SA_REP	80	21/04/00
Ande	SA_REP	80	24/03/00
Markle	ST_CLERK	50	08/03/00
Lee	SA_REP	80	23/02/00
Philtanker	ST_CLERK	50	06/02/00
Geoni	SH_CLERK	50	03/02/00
Zlotkey	SA_MAN	80	29/01/00
Marvins	SA_REP	80	24/01/00
Grant	SH_CLERK	50	13/01/00
Johnson	SA_REP	80	04/01/00
Perkins	SH_CLERK	50	19/12/99
Gee	ST_CLERK	50	12/12/99
Popp	FI_ACCOUNT	100	07/12/99

Ordenación según alias de Columna

```
SELECT    employee_id, last_name, salary*12 SALANUAL
FROM      employees
ORDER BY  SALANUAL;
```

EMPLOYEE_ID	LAST_NAME	SALANUAL
132	Olson	25200
128	Markle	26400
136	Philtanker	26400
127	Landry	28800
135	Gee	28800
119	Colmenares	30000
140	Patel	30000
144	Vargas	30000
191	Perkins	30000
182	Sullivan	30000
131	Marlow	30000
118	Himuro	31200

Ordenación según Múltiples Columnas

- El orden de la lista ORDER BY es el de ordenación.

```
SELECT    last_name, department_id, salary
FROM      employees
ORDER BY  department_id, salary DESC;
```

LAST_NAME	DEPARTMENT_ID	SALARY
Whalen	10	4400
Hartstein	20	13000
Fay	20	6000
Raphaely	30	11000
Khoo	30	3100
Baida	30	2900
Tobias	30	2800
Himuro	30	2600
Colmenares	30	2500
Mavris	40	6500
Fripp	50	8200
Weiss	50	8000

- Puede ordenar según una columna que no se encuentre en la lista SELECT.

Resumen

En esta lección, debería haber aprendido a:

- Utilizar la cláusula **WHERE** para restringir filas de salida:
 - Utilizar las condiciones de comparación
 - Utilizar las condiciones **BETWEEN**, **IN**, **LIKE**, y **NULL**
 - Aplicar los operadores lógicos **AND**, **OR**, y **NOT**
- Utilizar la cláusula **ORDER BY** para ordenar filas de salida.

```
SELECT      * | {[DISTINCT] column /expression [alias],...}  
FROM        table  
[WHERE      condition(s)]  
[ORDER BY {column, expr, alias} [ASC|DESC]];
```

Visión General de la Práctica 2

Esta práctica cubre los siguientes temas:

- **Selección de datos y cambio del orden de las filas mostradas**
- **Restricción de filas utilizando la cláusula WHERE**
- **Ordenación de filas utilizando la cláusula ORDER BY**