



Restricción y Ordenación de Datos

Objetivos

Al finalizar esta lección, debería estar capacitado para lo siguiente:





- Limitar las filas recuperadas por una consulta
- Ordenar las filas recuperadas por una consulta
- Usar la sustitución con ampersand para restringir y ordenar la salida en tiempo de ejecución

Agenda

- Limitación de filas con:
 - Cláusula `WHERE`
 - Condiciones de comparación con las condiciones `=`, `<=`, `BETWEEN`, `IN`, `LIKE` y `NULL`
 - Condiciones lógicas mediante los operadores `AND`, `OR` y `NOT`
- Reglas de prioridad de los operadores en una expresión
- Ordenación de filas mediante la cláusula `ORDER BY`
- Variables de sustitución
- Comandos `DEFINE` y `VERIFY`


Limitación de Filas con una Selección





EMPLOYEES

	 EMPLOYEE_ID	 LAST_NAME	 JOB_ID	 DEPARTMENT_ID
1	200	Whalen	AD_ASST	10
2	201	Hartstein	MK_MAN	20
3	202	Fay	MK_REP	20
4	205	Higgins	AC_MGR	110
5	206	Gietz	AC_ACCOUNT	110

...

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



	 EMPLOYEE_ID	 LAST_NAME	 JOB_ID	 DEPARTMENT_ID
1	100	King	AD_PRES	90
2	101	Kochhar	AD_VP	90
3	102	De Haan	AD_VP	90

Limitación de las Filas Seleccionadas

- Restringir las filas devueltas al utilizar la cláusula `WHERE`:

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

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

Uso de la 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
1	100	King	AD_PRES	90
2	101	Kochhar	AD_VP	90
3	102	De Haan	AD_VP	90

Fechas y Cadenas de Caracteres

- Las cadenas de caracteres y valores de fecha se incluyen entre comillas simples.
- Los valores de caracteres son sensibles a mayúsculas/minúsculas y los valores de datos son sensibles a formato.
- El formato de visualización de la fecha por defecto es DD-MON-RR.

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

```
SELECT last_name
FROM   employees
WHERE  hire_date = '17-FEB-96' ;
```

Operadores de Comparación

Operador	Significado
=	Igual que
>	Mayor que
>=	Mayor o igual que
<	Menor que
<=	Menor o igual que
<>	Distinto de
BETWEEN ...AND...	Entre dos valores (ambos incluidos)
IN (set)	Coincide con cualquiera de los
LIKE	Coincide con un patrón de
IS NULL	Es un valor nulo

Uso de Operadores de Comparación

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

	LAST_NAME	SALARY
1	Matos	2600
2	Vargas	2500

Uso de Condiciones de Rango mediante el Operador BETWEEN

Utilizar el operador BETWEEN para mostrar las filas basadas en un rango de valores:

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

Límite inferior Límite superior

	R2	LAST_NAME	R2	SALARY
1		Rajs		3500
2		Davies		3100
3		Matos		2600
4		Vargas		2500

Condición de Miembro mediante el Operador IN

Utilizar el operador `IN` para probar los valores de 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
1	201	Hartstein	13000	100
2	101	Kochhar	17000	100
3	102	De Haan	17000	100
4	124	Mourgos	5800	100
5	149	Zlotkey	10500	100
6	200	Whalen	4400	101
7	205	Higgins	12000	101
8	202	Fay	6000	201

Coincidencia de Patrones mediante el Operador LIKE

- Utilizar el operador `LIKE` para realizar búsquedas con comodines de valores de cadena de búsqueda válidos.
- Las condiciones de búsqueda pueden contener caracteres literales o números:
 - `%` indica cero o varios caracteres.
 - `_` indica un carácter.

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

Combinación de Caracteres Comodín

- Puede combinar los dos caracteres comodín (% , _) con caracteres literales para la coincidencia de patrones:

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

	LAST_NAME
1	Kochhar
2	Lorentz
3	Mourgos

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

Uso de las Condiciones NULL

Probar condiciones nulas con el operador `IS NULL`.

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

	LAST_NAME	MANAGER_ID
1	King	(null)

Definición de Condiciones mediante los Operadores Lógicos

Operador	Significado
AND	Devuelve TRUE si <i>ambas</i> condiciones de componente son verdaderas
OR	Devuelve TRUE si <i>cualquier</i> condición de componente es verdadera
NOT	Devuelve TRUE si la condición es falsa

Uso del Operador AND

AND necesita que ambas 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
1	201	Hartstein	MK_MAN	13000
2	149	Zlotkey	SA_MAN	10500

Uso del Operador OR

OR necesita que cualquier condición 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
1	201	Hartstein	MK_MAN	13000
2	205	Higgins	AC_MGR	12000
3	100	King	AD_PRES	24000
4	101	Kochhar	AD_VP	17000
5	102	De Haan	AD_VP	17000
6	124	Mourgos	ST_MAN	5800
7	149	Zlotkey	SA_MAN	10500
8	174	Abel	SA_REP	11000

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
1	De Haan	AD_VP
2	Fay	MK_REP
3	Gietz	AC_ACCOUNT
4	Hartstein	MK_MAN
5	Higgins	AC_MGR
6	King	AD_PRES
7	Kochhar	AD_VP
8	Mourgos	ST_MAN
9	Whalen	AD_ASST
10	Zlotkey	SA_MAN

Agenda

- Limitación de filas con:
 - Cláusula `WHERE`
 - Condiciones de comparación con las condiciones `=`, `<=`, `BETWEEN`, `IN`, `LIKE` y `NULL`
 - Condiciones lógicas mediante los operadores `AND`, `OR` y `NOT`
- Reglas de prioridad de los operadores en una expresión
- Ordenación de filas mediante la cláusula `ORDER BY`
- Variables de sustitución
- Comandos `DEFINE` y `VERIFY`

Reglas de Prioridad

Operador	Significado
1	Operadores aritméticos
2	Operador de concatenación
3	Condiciones de comparación
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Distinto de
7	Condición lógica NOT
8	Condición lógica AND
9	Condición lógica OR

Puede utilizar los paréntesis para sustituir las reglas de prioridad.

Reglas de Prioridad

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

1

	LAST_NAME	JOB_ID	SALARY
1	King	AD_PRES	24000
2	Abel	SA_REP	11000
3	Taylor	SA_REP	8600
4	Grant	SA_REP	7000

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

2

	LAST_NAME	JOB_ID	SALARY
1	King	AD_PRES	24000

Agenda

- Limitación de filas con:
 - Cláusula `WHERE`
 - Condiciones de comparación con las condiciones `=`, `<=`, `BETWEEN`, `IN`, `LIKE` y `NULL`
 - Condiciones lógicas mediante los operadores `AND`, `OR` y `NOT`
- Reglas de prioridad de los operadores en una expresión
- Ordenación de filas mediante la cláusula `ORDER BY`
- Variables de sustitución
- Comandos `DEFINE` y `VERIFY`

Uso de la Cláusula ORDER BY

- Ordenar las filas recuperadas con la cláusula ORDER BY:
 - ASC: orden ascendente, valor por defecto
 - DESC: orden descendente
- La cláusula ORDER BY es la última en una sentencia SELECT:

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


	LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
1	King	AD_PRES	90	17-JUN-87
2	Whalen	AD_ASST	10	17-SEP-87
3	Kochhar	AD_VP	90	21-SEP-89
4	Hunold	IT_PROG	60	03-JAN-90
5	Ernst	IT_PROG	60	21-MAY-91
6	De Haan	AD_VP	90	13-JAN-93

...

Ordenación


- Ordenar en orden descendente:

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



- Ordenar por alias de columna:


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



Ordenación


- Ordenar por posición numérica de la columna:

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY 3;
```



- Ordenar por varias columnas:

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



Agenda

- Limitación de filas con:
 - Cláusula `WHERE`
 - Condiciones de comparación con las condiciones `=`, `<=`, `BETWEEN`, `IN`, `LIKE` y `NULL`
 - Condiciones lógicas mediante los operadores `AND`, `OR` y `NOT`
- Reglas de prioridad de los operadores en una expresión
- Ordenación de filas mediante la cláusula `ORDER BY`
- Variables de sustitución
- Comandos `DEFINE` y `VERIFY`

Variables de Sustitución



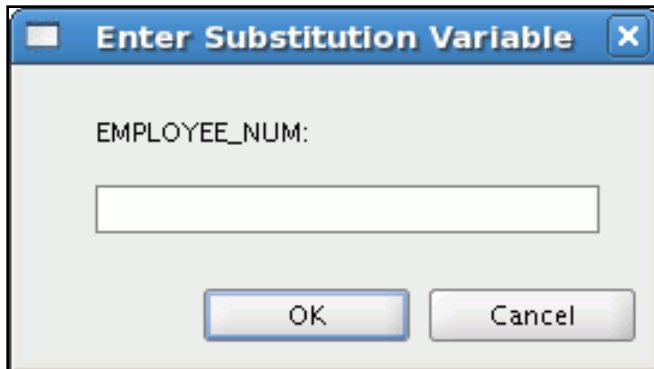
Variables de Sustitución

- Utilizar variables de sustitución para:
 - Almacenar valores temporalmente con una sustitución de un solo ampersand (&) y de dos ampersands (&&)
- Utilizar las variables de sustitución para complementar:
 - Condiciones `WHERE`
 - Cláusulas `ORDER BY`
 - Expresiones de columna
 - Nombres de tabla
 - Sentencias `SELECT` completas

Uso de la Variable de Sustitución de Un Solo Ampersand

Utilizar una variable prefijada con un ampersand (&) para solicitar al usuario un valor:

```
SELECT employee_id, last_name, salary, department_id  
FROM employees  
WHERE employee_id = &employee_num ;
```

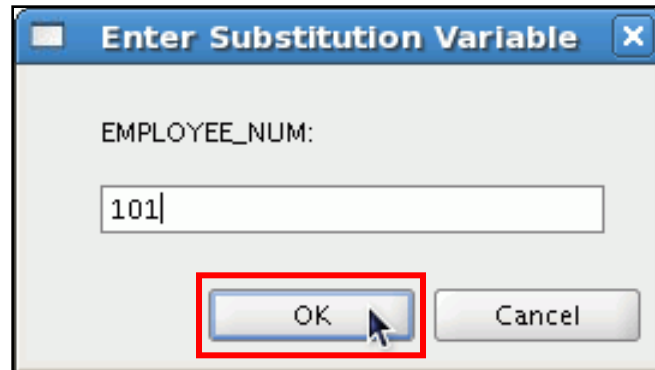


Enter Substitution Variable

EMPLOYEE_NUM:

OK Cancel

Uso de la Variable de Sustitución de Un Solo Ampersand

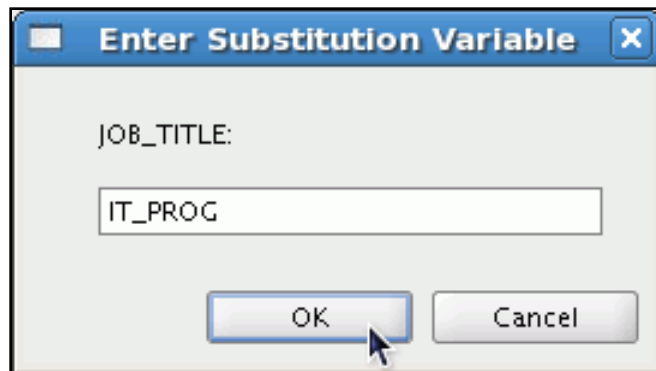


	EMPLOYEE_ID	LAST_NAME	SALARY	DEPARTMENT_ID
1	101	Kochhar	17000	90

Valores de Fecha y Carácter con Variables de Sustitución

Utilizar las comillas simples para los valores de fecha y carácter:

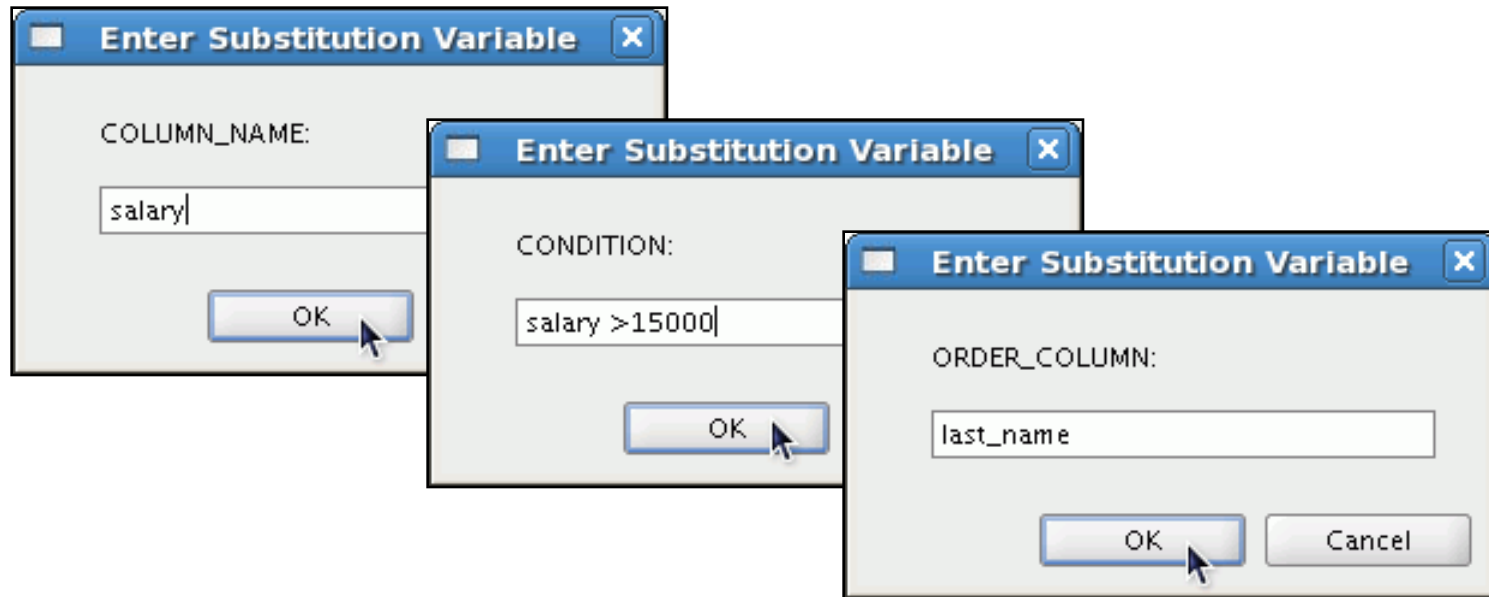
```
SELECT last_name, department_id, salary*12
FROM   employees
WHERE  job_id = '&job_title' ;
```



	LAST_NAME	DEPARTMENT_ID	SALARY*12
1	Hunold	60	108000
2	Ernst	60	72000
3	Lorentz	60	50400

Especificación de Nombres de Columna, Expresiones y Texto

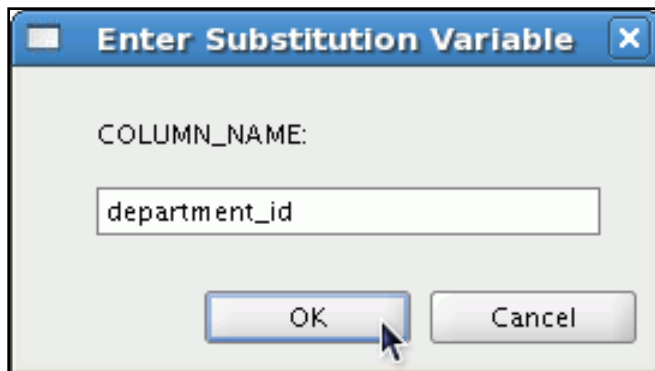
```
SELECT employee_id, last_name, job_id, &column_name  
FROM employees  
WHERE &condition  
ORDER BY &order_column ;
```



Uso de Variables de Sustitución de Dos Ampersands

Usar dos ampersands (&&) si se desea reutilizar el valor de la variable sin preguntar siempre al usuario:

```
SELECT  employee_id, last_name, job_id, &&column_name  
FROM    employees  
ORDER BY &column name ;
```



A dialog box titled "Enter Substitution Variable" with a close button (X). It contains a label "COLUMN_NAME:" and a text input field with the value "department_id". At the bottom are "OK" and "Cancel" buttons. A mouse cursor is pointing at the "OK" button.

	EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
1	200	Whalen	AD_ASST	10
2	201	Hartstein	MK_MAN	20
3	202	Fay	MK_REP	20

...


Agenda

- Limitación de filas con:
 - Cláusula `WHERE`
 - Condiciones de comparación con las condiciones `=`, `<=`, `BETWEEN`, `IN`, `LIKE` y `NULL`
 - Condiciones lógicas mediante los operadores `AND`, `OR` y `NOT`
- Reglas de prioridad de los operadores en una expresión
- Ordenación de filas mediante la cláusula `ORDER BY`
- Variables de sustitución
- Comandos `DEFINE` y `VERIFY`

Uso del Comando DEFINE

- Usar el comando `DEFINE` para crear y asignar un valor a una variable.
- Usar el comando `UNDEFINE` de iSQL*Plus para eliminar una variable.

```
DEFINE employee_num = 200  
  
SELECT employee_id, last_name, salary, department_id  
FROM employees  
WHERE employee_id = &employee_num ;  
  
UNDEFINE employee_num
```

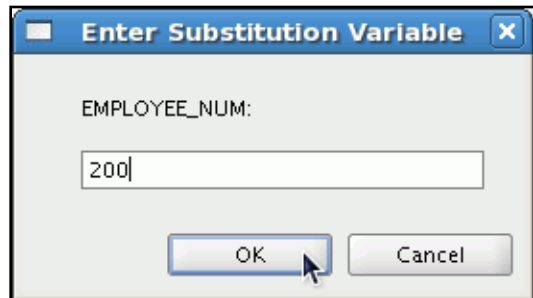
A red arrow points from the value '200' in the DEFINE statement to the '&employee_num' placeholder in the WHERE clause of the SQL query, illustrating how the variable's value is substituted.

Uso del Comando VERIFY

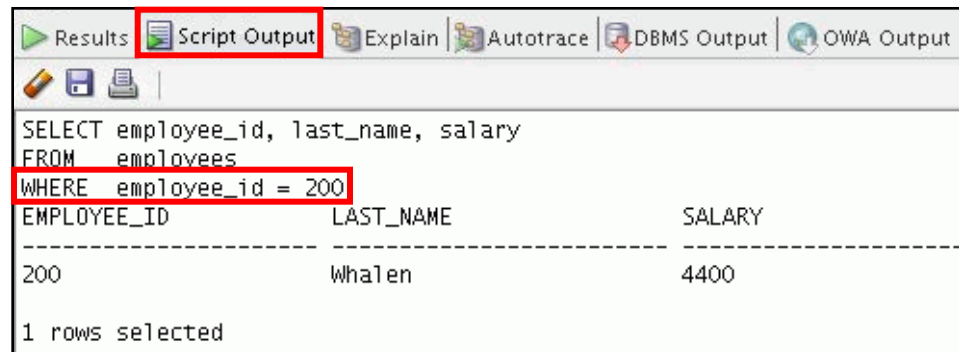
Usar el comando VERIFY para cambiar la visualización de la variable de sustitución, antes y después de que SQL Developer sustituya las variables de sustitución con los valores:

SET VERIFY ON

```
SELECT employee_id, last_name, salary
FROM   employees
WHERE  employee_id = &employee_num;
```



A dialog box titled "Enter Substitution Variable" with a close button (X). It contains a label "EMPLOYEE_NUM:" and a text input field with the value "200". At the bottom are "OK" and "Cancel" buttons. A mouse cursor is pointing at the "OK" button.



The "Script Output" window in SQL Developer. The "Script Output" tab is selected and highlighted with a red box. It shows the SQL script being executed, with the WHERE clause highlighted in red. Below the script, the results are displayed in a table format.

EMPLOYEE_ID	LAST_NAME	SALARY
200	Whalen	4400

1 rows selected

Prueba

¿Cuáles de los siguientes operadores son válidos para la cláusula WHERE?

1. >=
2. IS NULL
3. !=
4. IS LIKE
5. IN BETWEEN
6. <>

Resumen

En esta lección, debe haber aprendido lo siguiente:

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

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

- Usar la sustitución con ampersand para restringir y ordenar la salida en tiempo de ejecución

Práctica 2: Visión General

En esta práctica se abordan los siguientes temas:

- Selección de datos y cambio del orden de las filas que se muestran
- Restricción de filas mediante la cláusula `WHERE`
- Ordenación de filas mediante la cláusula `ORDER BY`
- Uso de las variables de sustitución para agregar flexibilidad a las sentencias SQL `SELECT`