



# Usando Restricciones en Sentencias SQL

**MDY2131**

# Experiencia de Aprendizaje y Competencia Asociada



ESCUELA DE  
INFORMÁTICA Y  
TELECOMUNICACIONES

Experiencia	Nombre	Unidad de Competencia Especialidad – Nivel de la Competencia de Empleabilidad
Nº 1	Construyendo sentencias simples de recuperación de datos	Desarrolla operaciones sobre la base de datos que permitan la obtención, actualización, inserción y eliminación de información estableciendo una conexión con una base de datos para cumplir con los requerimientos de la organización.
		Resolución de Problemas (N1)

- Cómo utilizar la cláusula WHERE para restringir las filas afectadas por una sentencias SQL.
- Cómo utilizar los diferentes operadores en una condición de comparación.
- Cómo utilizar los diferentes operadores lógicos en una condición.
- Crear tablas a partir de una Sentencia SELECT.
- Usar variable de sustitución para seleccionar, restringir y ordenar la información en tiempo de ejecución.

# Restringir la Filas Seleccionadas

\*Las sentencias de los ejemplos usan las tablas del esquema HR de la Base de Datos Oracle

# Restringiendo las Filas que son Seleccionadas

**¿Cómo mostrar sólo los empleados que trabajan en el departamento 90?**

EMPLOYEE_ID	JOB_ID	LAST_NAME	DEPARTMENT_ID
100	AD_PRES	King	90
101	AD_VP	Kochhar	90
102	AD_VP	De Haan	90
103	IT_PROG	Hunold	60
104	IT_PROG	Ernst	60
105	IT_PROG	Austin	60
106	IT_PROG	Pataballa	60
107	IT_PROG	Lorentz	60
108	FI_MGR	Greenberg	100
109	FI_ACCOUNT	Faviet	100
110	FI_ACCOUNT	Chen	100
.....			
.....			
204	PR_REP	Baer	70
205	AC_MGR	Higgins	110
206	AC_ACCOUNT	Gietz	110



# Restringiendo las Filas que son Seleccionadas

La cláusula **WHERE** restringe a que las filas deben cumplir con una condición para ser visualizadas, actualizadas o eliminadas

En una sentencia **SELECT** o **DELETE** va a continuación de la cláusula **FROM**.  
En una sentencia **UPDATE** va a continuación de la cláusula **SET**

Se pueden comparar valores entre columnas, valores literal, expresiones aritméticas o funciones

- Sintaxis:

```
SELECT * | { [ DISTINCT ] columna | expresión [alias], ... }  
FROM tabla  
[WHERE condición(es)_comparación]  
[ORDER BY {columna, alias, expresión, posición_numérica} [ASC|DESC]];
```

# Restringiendo las Filas que son Seleccionadas

- Elementos de una comparación:

... WHERE *columna\_a\_comparar* *condición\_comparación* *valor\_con\_el\_se\_compara*

Operador de  
Comparación

- Columna
- Constante
- Lista de valores
- Cadena de Caracteres
- Fechas literales
- Expresiones matemáticas

- Ejemplo:

```
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

# Restringiendo las Filas que son Seleccionadas

- Ejemplo:

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

	LAST_NAME	JOB_ID	DEPARTMENT_ID
1	Whalen	AD_ASST	10

- Ejemplo:

```
SELECT last_name, hire_date  
FROM employees  
WHERE hire_date > '01/03/2008';
```

	LAST_NAME	HIRE_DATE
1	Markle	08/03/2008
2	Ande	24/03/2008
3	Banda	21/04/2008
4	Kumar	21/04/2008



# Condiciones de Comparación



OPERADOR	DESCRIPCIÓN
=	Igual a
>	Mayor que
>=	Mayor o igual a
<	Menor que
<=	Menor o igual a
<>, !=, ^=	No igual o distinto a
BETWEEN ... AND ...	Entre dos valores (inclusivos)
IN (lista)	Valores que estén en la lista
LIKE	Valores que cumplan con una condición textual %: indica una serie cualquiera de caracteres _: indica un carácter cualquiera
IS NULL	Es un valor NULO

# Condición de Comparación: Operador <=

- Ejemplo:

```
SELECT last_name, salary  
FROM employees  
WHERE salary <= 2500;
```

	LAST_NAME	SALARY
1	Colmenares	2500
2	Landry	2400
3	Markle	2200
4	Marlow	2500
5	Olson	2100
6	Gee	2400
7	Philtanker	2200
8	Patel	2500
9	Vargas	2500
10	Sullivan	2500
11	Perkins	2500

# Condición de Comparación: Operador BETWEEN

- Ejemplo:

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

Límite Inferior

Límite Superior

	LAST_NAME	SALARY
1	Khoo	3100
2	Baida	2900
3	Tobias	2800
4	Himuro	2600
5	Colmenares	2500
6	Nayer	3200
7	Mikkilineni	2700

15	Patel	2500
16	Rajs	3500
17	Davies	3100

30	Walsh	3100
31	Feeney	3000
32	OConnell	2600
33	Grant	2600

# Condición de Comparación: Operador IN

- Ejemplo:

```
SELECT employee_id, last_name,  
       manager_id, department_id  
FROM employees  
WHERE last_name IN('Hartstein', 'Vargas');
```

EMPLOYEE_ID	LAST_NAME	MANAGER_ID	DEPARTMENT_ID
1	201 Hartstein	100	20
2	144 Vargas	124	50

- Ejemplo:

```
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	101 Kochhar	17000	100
2	102 De Haan	17000	100
3	114 Raphaely	11000	100
4	120 Weiss	8000	100
5	121 Fripp	8200	100
6	122 Kaufling	7900	100
7	123 Vollman	6500	100
8	124 Mourgós	5800	100
9	145 Russell	14000	100
10	146 Partners	13500	100

17	203 Mavris	6500	101
18	204 Baer	10000	101
19	205 Higgins	12008	101
20	202 Fay	6000	201

# Condición de Comparación: Operador LIKE

- Ejemplo:

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

	FIRST_NAME
1	Sundar
2	Shelli
3	Sarah
4	Shelley
5	Steven
6	Sundita
7	Steven
8	Susan
9	Samuel
10	Sarath
11	Stephen
12	Sigal
13	Shanta

# Condición de Comparación: Operador LIKE

- Ejemplo:

```
SELECT last_name, hire_date  
FROM employees  
WHERE hire_date LIKE '%03';
```

	LAST_NAME	HIRE_DATE
1	King	17/06/2003
2	Khoo	18/05/2003
3	Kaufling	01/05/2003
4	Ladwig	14/07/2003
5	Rajs	17/10/2003
6	Whalen	17/09/2003

- Ejemplo

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

	LAST_NAME
1	Colmenares
2	Doran
3	Fox
4	Johnson
5	Jones
6	Kochhar
7	Lorentz
8	Mourgos
9	Popp
10	Rogers
11	Tobias
12	Vollman



# Condición de Comparación: Operador NULL

- Ejemplo:

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

LAST_NAME	MANAGER_ID
King	(null)

- Ejemplo

```
SELECT last_name, job_id, commission_pct  
FROM employees  
WHERE commission_pct IS NOT NULL;
```

	LAST_NAME	JOB_ID	COMMISSION_PCT
1	Russell	SA_MAN	0,4
2	Partners	SA_MAN	0,3
3	Errazuriz	SA_MAN	0,3
4	Cambrault	SA_MAN	0,3
5	Zlotkey	SA_MAN	0,2
6	Tucker	SA_REP	0,3
7	Bernstein	SA_REP	0,25
8	Hall	SA_REP	0,25

33	Livingston	SA_REP	0,2
34	Grant	SA_REP	0,15
35	Johnson	SA_REP	0,1

# Operadores Lógicos



Una sentencia SQL  
puede requerir más  
de una condición  
para visualizar o  
actualizar  
información de las  
tablas

Para poder definir  
más de una  
condición en una  
sentencia SQL se  
deben usar  
Operadores Lógicos

- Los operadores que se pueden utilizar en una sentencia SQL son:

OPERADOR	DESCRIPCIÓN
AND	Retorna verdadero si todos los componentes de la condición son verdaderas.
OR	Retorna verdadero si alguna de las expresiones de la condición son verdaderas.
NOT	Retorna verdadero si la condición es falsa.

- Ejemplo:

```
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	114	Raphaely	PU_MAN	11000
2	145	Russell	SA_MAN	14000
3	146	Partners	SA_MAN	13500
4	147	Errazuriz	SA_MAN	12000
5	148	Cambrault	SA_MAN	11000
6	149	Zlotkey	SA_MAN	10500
7	201	Hartstein	MK_MAN	13000

- Ejemplo:

```
SELECT employee_id, salary, hire_date  
FROM employees  
WHERE salary >= 17000  
OR hire_date > '01/01/2008';
```

	EMPLOYEE_ID	SALARY	HIRE_DATE
1	100	24000	17/06/2003
2	101	17000	21/09/2005
3	102	17000	13/01/2001
4	128	2200	08/03/2008
5	136	2200	06/02/2008
6	149	10500	29/01/2008
7	164	7200	24/01/2008
8	165	6800	23/02/2008
9	166	6400	24/03/2008
10	167	6200	21/04/2008
11	173	6100	21/04/2008
12	179	6200	04/01/2008
13	183	2800	03/02/2008
14	199	2600	13/01/2008

# Usando Operador Lógico NOT

- Ejemplo:

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

	EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
1	100	King	AD_PRES	24000
2	101	Kochhar	AD_VP	17000
3	102	De Haan	AD_VP	17000
4	108	Greenberg	FI_MGR	12008
5	109	Faviet	FI_ACCOUNT	9000
6	110	Chen	FI_ACCOUNT	8200
7	111	Sciarra	FI_ACCOUNT	7700
8	112	Urman	FI_ACCOUNT	7800
9	113	Popp	FI_ACCOUNT	6900
10	114	Raphaely	PU_MAN	11000
.....				
24	203	Mavris	HR_REP	6500
25	204	Baer	PR_REP	10000
26	205	Higgins	AC_MGR	12008
27	206	Gietz	AC_ACCOUNT	8300

# Reglas de precedencia para los Operadores

ORDEN DE PRECEDENCIA	SIGNIFICADO
1	Operadores Aritméticos
2	Operador de Concatenación
3	Condiciones de Comparaciones
4	IS [NOT], NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	No igual
7	Condición lógica NOT
8	Condición lógica AND
9	Condición lógica OR



# Reglas de precedencia para los Operadores

- Ejemplo:

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id = 'SA_REP' } 2
      OR job_id = 'AD_PRES' } 1
      AND salary > 15000;
```

	LAST_NAME	JOB_ID	SALARY
1	King	AD_PRES	24000
2	Tucker	SA_REP	10000
3	Bernstein	SA_REP	9500
4	Hall	SA_REP	9000
5	Olsen	SA_REP	8000
6	Cambrault	SA_REP	7500
7	Tuvault	SA_REP	7000
8	King	SA_REP	10000
9	Sully	SA_REP	9500
10	McEwen	SA_REP	9000
.....			
28	Taylor	SA_REP	8600
29	Livingston	SA_REP	8400
30	Grant	SA_REP	7000
31	Johnson	SA_REP	6200

# Reglas de precedencia para los Operadores

- Ejemplo:

```
SELECT last_name, job_id, salary
FROM employees
WHERE (job_id = 'SA_REP'
      OR job_id = 'AD_PRES') } 1
      AND salary > 15000; } 2
```

	LAST_NAME	JOB_ID	SALARY
1	King	AD_PRES	24000

# Sentencia SELECT para crear una Tabla

# Usando Sentencia SELECT para crear Tablas

Crea una tabla e  
inserta filas  
mediante la  
combinación de la  
sentencia CREATE y  
la opción AS  
Subconsulta

Debe coincidir el  
número de columnas  
de la tabla con el  
número de columnas  
de la Subconsulta

La tabla se puede  
crear con los  
nombres y tipos de  
datos de las  
columnas  
seleccionadas

- Sintaxis:

```
CREATE TABLE nombre_tabla [(columna, columna...)]  
AS sentencia_select;
```

# Usando Sentencia SELECT para crear Tablas

- Ejemplo:

**CREATE TABLE employees\_copia**  
**AS SELECT \* FROM employees;**

	COLUMN_NAME	DATA_TYPE	NULLABLE
1	EMPLOYEE_ID	NUMBER (6, 0)	No
2	FIRST_NAME	VARCHAR2 (20 BYTE)	Yes
3	LAST_NAME	VARCHAR2 (25 BYTE)	No
4	EMAIL	VARCHAR2 (25 BYTE)	No
5	PHONE_NUMBER	VARCHAR2 (20 BYTE)	Yes
6	HIRE_DATE	DATE	No
7	JOB_ID	VARCHAR2 (10 BYTE)	No
8	SALARY	NUMBER (8, 2)	Yes
9	COMMISSION_PCT	NUMBER (2, 2)	Yes
10	MANAGER_ID	NUMBER (6, 0)	Yes
11	DEPARTMENT_ID	NUMBER (4, 0)	Yes

**TABLA EMPLOYEES\_COPIA**

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	100	Steven	King	SKING	515.123.4567	17/06/2003	AD_PRES	24000	(null)	(null)	90
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/2005	AD_VP	17000	(null)	100	90
3	102	Lex	De Haan	LDEHAAN	515.123.4569	13/01/2001	AD_VP	17000	(null)	100	90
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	03/01/2006	IT_PROG	9000	(null)	102	60
5	104	Bruce	Ernst	BERNST	590.423.4568	21/05/2007	IT_PROG	6000	(null)	103	60
6	105	David	Austin	DAUSTIN	590.423.4569	25/06/2005	IT_PROG	4800	(null)	103	60
7	106	Valli	Pataballa	VPATABAL	590.423.4560	05/02/2006	IT_PROG	4800	(null)	103	60
8	107	Diana	Lorentz	DLORENTZ	590.423.5567	07/02/2007	IT_PROG	4200	(null)	103	60
9	108	Nancy	Greenberg	NGREENBE	515.124.4569	17/08/2002	FI_MGR	12008	(null)	101	100
10	109	Daniel	Faviet	DFAVIET	515.124.4169	16/08/2002	FI_ACCOUNT	9000	(null)	108	100

105	204	Hermann	Baer	HBAER	515.123.8888	07/06/2002	PR_REP	10000	(null)	101	70
106	205	Shelley	Higgins	SHIGGINS	515.123.8080	07/06/2002	AC_MGR	12008	(null)	101	110
107	206	William	Gietz	WGIEZT	515.123.8181	07/06/2002	AC_ACCOUNT	8300	(null)	205	110

# Usando Sentencia SELECT para crear Tablas

- Ejemplo:

```
CREATE TABLE dept80  
AS SELECT employee_id, salary*12 salario_anual,  
        hire_date  
FROM employees  
WHERE department_id = 80;
```

	COLUMN_NAME	DATA_TYPE	NULLABLE
1	EMPLOYEE_ID	NUMBER (6,0)	Yes
2	SALARIO_ANUAL	NUMBER	Yes
3	HIRE_DATE	DATE	No

**TABLA DEPT80**

	EMPLOYEE_ID	SALARIO_ANUAL	HIRE_DATE
1	145	168000	01/10/2004
2	146	162000	05/01/2005
3	147	144000	10/03/2005
4	148	132000	15/10/2007
5	149	126000	29/01/2008
6	150	120000	30/01/2005
7	151	114000	24/03/2005
8	152	108000	20/08/2005
9	153	96000	30/03/2006
10	154	90000	09/12/2006

31	175	105600	19/03/2005
32	176	103200	24/03/2006
33	177	100800	23/04/2006
34	179	74400	04/01/2008

# Usando Sentencia SELECT para crear Tablas

- Ejemplo:

```
CREATE TABLE DEPT30  
(cod_empleado NOT NULL CONSTRAINTS PK_DEPT30 PRIMARY KEY,  
salario_anual NOT NULL,  
fecha_contrato NOT NULL)  
AS SELECT employee_id, salary*12, hire_date  
FROM employees  
WHERE department_id = 30;
```

**TABLA DEPT80**

	↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE
1	COD_EMPLEADO	NUMBER (6, 0)	No
2	SALARIO_ANUAL	NUMBER	No
3	FECHA_CONTRATO	DATE	No

	↕ COD_EMPLEADO	↕ SALARIO_ANUAL	↕ FECHA_CONTRATO
1	114	132000	07/12/2002
2	115	37200	18/05/2003
3	116	34800	24/12/2005
4	117	33600	24/07/2005
5	118	31200	15/11/2006
6	119	30000	10/08/2007





# Variables de Sustitución

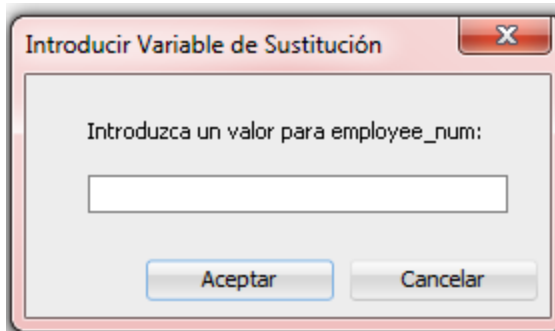


- Ejemplo:

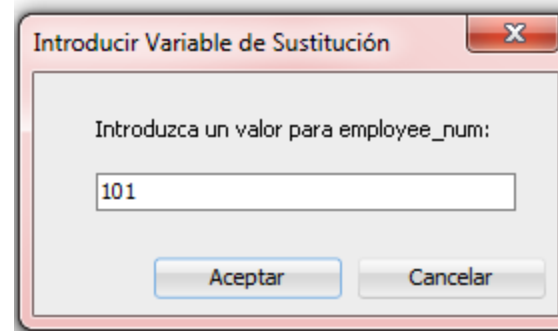
1

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

2



3



4

EMPLOYEE_ID	LAST_NAME	SALARY	DEPARTMENT_ID
101	Kochhar	17000	90

# Trabajando con Variables de Sustitución (&)

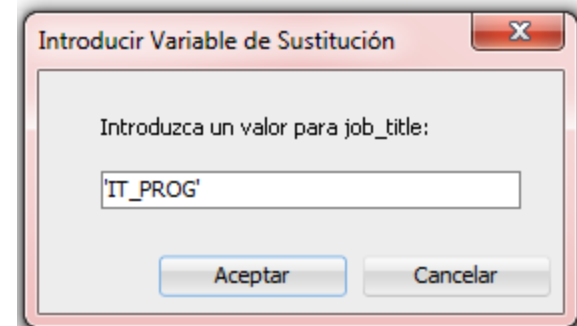
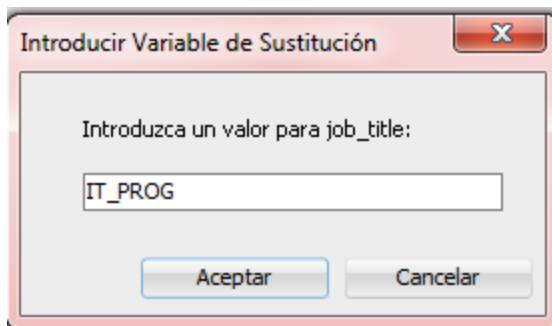
- Ejemplo:

1

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

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

2



3

	LAST_NAME	DEPARTMENT_ID	SALARY*12
1	Hunold	60	108000
2	Ernst	60	72000
3	Austin	60	57600
4	Pataballa	60	57600
5	Lorentz	60	50400

# Trabajando con Variables de Sustitución (&)

- Ejemplo:

1

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

2

Introducir Variable de Sustitución

Introduzca un valor para column\_name:

Aceptar Cancelar

Introducir Variable de Sustitución

Introduzca un valor para condition:

Aceptar Cancelar

Introducir Variable de Sustitución

Introduzca un valor para order\_column:

Aceptar Cancelar

3

	EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
1	102	De Haan	AD_VP	17000
2	100	King	AD_PRES	24000
3	101	Kochhar	AD_VP	17000

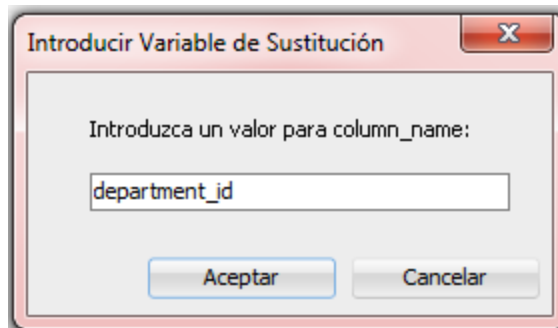
# Trabajando con Variables de Sustitución (&&)

- Ejemplo:

1

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

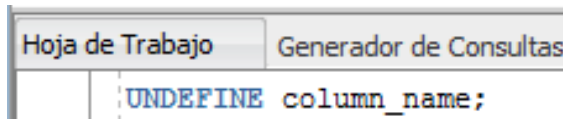
2



3

	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	114	Raphaely	PU_MAN	30
5	115	Khoo	PU_CLERK	30
6	116	Baida	PU_CLERK	30
7	117	Tobias	PU_CLERK	30
8	118	Himuro	PU_CLERK	30
9	119	Colmenares	PU_CLERK	30
10	203	Mavris	HR_REP	40

4



```
UNDEFINE column_name;
```

105	205	Higgins	AC_MGR	110
106	206	Gietz	AC_AC...	110
107	178	Grant	SA_REP	(null)

- Se explicó cómo utilizar la cláusula WHERE para restringir las filas afectadas por una sentencias SQL.
- Se explicó cómo utilizar los diferentes operadores en una condición de comparación.
- Se explicó cómo utilizar los diferentes operadores lógicos en una condición.
- Se explicó cómo crear tablas a partir de una Sentencia SELECT.
- Se explicó cómo usar variables de sustitución para seleccionar, restringir y ordenar la información en tiempo de ejecución