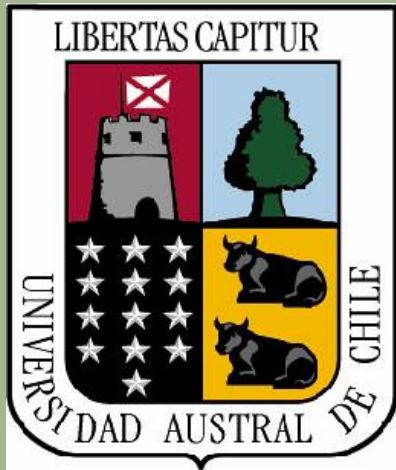


TALLER DE PROGRAMACIÓN

UNIDAD 2: BASES DE DATOS

PROFESOR GERMÁN BARRIENTOS

PROFESOR JOEL TORRES



QUÉ APRENDEREMOS

- 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.

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

- 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	Mourgos	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

USANDO 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.

USANDO OPERADOR LÓGICO AND

- 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

USANDO OPERADOR LÓGICO OR

- 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'
      OR job_id = 'AD_PRES'
      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')
AND salary > 15000;
```

	LAST_NAME	JOB_ID	SALARY
1	King	AD_PRES	24000

TALLER DE PROGRAMACIÓN

UNIDAD 2: BASES DE DATOS

PROFESOR GERMÁN BARRIENTOS

PROFESOR JOEL TORRES

