Universidad del Quindío - Ingeniería de Sistemas - Leonardo Hernández R.
leonardohernandez@telesat.com.co - www.geocities.com/leoher314
Mar.28/2002 Sep.28/2002 Mar.11/2003

Tutorial de MySQL
Lección III

4.1 Tabla de ejemplo para esta lección

Para esta sección usaremos la tabla Empleado (la misma de la lección II):

```
Empleado (  idEmpleado, nombre, apellido, sexo,
            estadoCivvil, fechaNacimiento, dependencia, sueldo)
```

El atributo *idEmpleado* es un número consecutivo que identifica el empleado. El atributo *dependencia* corresponde al nombre de la dependencia donde labora el empleado.

4.2 Operadores de conjuntos y rangos

Los Operadores *in*, *not in*, *between* y *not between* disminuyen el uso de operadores lógicos, reduciendo el tamaño de las cláusulas where. A continuación se dan algunos ejemplos de salidas a obtener y su correspondiente instrucción SQL:

a) Lista de los registros de empleados casados o viudos.

```
mysql>      select * from Empleado
            where estadoCivil in ( "c", "v");
```

b) Lista de empleados que NO son casados ni viudos.

```
mysql>      select * from Empleado
            where estadoCivil not in ( "c", "v");
```

c) Lista de los registros de empleado con sueldo entre $500.000 y $1’000.000 (incluidos $500.000 y $1’000.000).

```
mysql>      select * from Empleado
            where sueldo between 500000 and 1000000;
```

d) Lista de los empleados con sueldo menor estrictamente que $500.000 o mayor estrictamente que $1’000.000.

```
mysql>      select * from Empleado
            where sueldo not between 500000 and 1000000;
```

4.3 Ordenación

La cláusula *order by* permite ordenar la salida por uno o varios atributos. El orden será ascendente, a no ser que explícitamente se indique lo contrario. Veamos algunos ejemplos de salidas ordenadas y su correspondiente instrucción SQL:

a) Lista de empleados por orden alfabético de nombre (de la A a la Z).

```
mysql>      select * from Empleado
            order by nombre;
```

b) Lista de empleados ordenados por nombre, de la Z a la A.

```
mysql>      select * from Empleado
            order by nombre desc;
```

c) Lista de empleados por orden de sexo (primero las mujeres) - estado civil (en orden c, r, s, u, v)– nombre (de la A a la Z)

```
mysql>  select sexo, estadoCivil, nombre, sueldo from Empleado
        order by sexo, estadoCivil, nombre;
```

sexo	estadoCivil	nombre	sueldo
f	**c**	Magda	1000000
f	**s**	Ana Karina	300000
f	**s**	Luz	3000000
f	**u**	**Ana Karina**	500000
f	**u**	**Angela**	300000
f	**u**	**Diana**	300000
m	c	Tiberio	800000
m	s	Alejandro	400000
m	s	Diego	300000
m	s	Jaider	2000000
m	u	James	400000
m	u	Miguel	500000

d) Lista de empleados por orden de sexo (primero los hombres), estado civil (en orden c, r, s, u, v) – nombre (de la Z a la A).

```
mysql>  select sexo, estadoCivil, nombre, sueldo from Empleado
        order by sexo desc, estadoCivil, nombre desc;
```

sexo	estadoCivil	nombre	sueldo
m	c	Tiberio	800000
m	s	Jaider	2000000
m	s	Diego	300000
m	s	Alejandro	400000
m	u	Miguel	500000
m	u	James	400000
f	**c**	Magda	1000000
f	**s**	Luz	3000000
f	**s**	Ana Karina	300000
f	**u**	**Diana**	300000
f	**u**	**Angela**	300000
f	**u**	**Ana Karina**	500000

Observe que *desc* se aplica únicamente al atributo inmediatamente anterior y no a los demás.

e) Lista de empleados por orden de fecha de nacimiento (primero el más joven):

```
mysql>      select * from Empleado
```

```
order by fechaNacimiento desc;
```

4.4 Búsqueda por emparejamiento de patrones – like y not like

Los operadores like y not like permiten la búsqueda de registros por emparejamiento de patrones. Estos operadores se usan con dos comodines: el carácter de subrayado (_), que se empareja con exactamente un carácter (cualquiera) , y el signo de porcentaje (%), que se empareja con cero uno o más caracteres.

Por ejemplo, es posible obtener una lista de los empleados cuyo nombre empiece por "An" (Ejemplo: Ana Karina) con:

```
mysql> select nombre, sueldo from Empleado
    -> where nombre like "an%";

+------------+--------+
| nombre     | sueldo |
+------------+--------+
| Ana Karina | 300000 |
| Ana Karina | 500000 |
| Angela     | 300000 |
+------------+--------+
```

Los operadores *like* y *not like* no son sensibles a mayúsculas – minúsculas, por lo que, en el ejemplo, da lo mismo usar "AN%" o "aN%" o cualquier otra combinación de mayúsculas y minúsculas. También se puede usar la comilla sencilla: 'An%'

Otros ejemplos:

a) Una lista de los empleados cuyo nombre tenga exactamente cinco caracteres (Ejemplo: Pedro).

```
mysql>      select * from Empleado
            where nombre like "_____";
```

b) Una lista de los empleados cuyo nombre empiece con P y continúe con exactamente 4 caracteres (Ejemplo: Pedro).

```
mysql>      select * from Empleado
            where nombre like "P____";
```

También es muy útil el operador *not like.* Por ejemplo se puede obtener una lista de los empleados cuyo nombre NO comience con "an" (Ejemplo: Jimena) con:

```
mysql>      select * from Empleado
            where nombre not like "an%";
```

4.5 Búsqueda por emparejamiento de patrones – regexp y not regexp

Los operadores regexp y not regexp dan aún más flexibilidad para la búsqueda de registros por emparejamiento de patrones, estos operadores están disponibles en MySQL pero no en Oracle. Las convenciones son diferentes de las del operador like. regexp es una abreviatura de "expresión regular", un tipo de expresiones que se estudian en Teoría de Lenguajes Formales (Teoría de la Computación). Veamos algunos ejemplos.

a) Lista de empleados cuyo nombre tiene a "ri" por subcadena (Ejemplo: Ana Karina):

```
mysql> select nombre, sueldo from Empleado where nombre regexp "ri";

+------------+--------+
| nombre     | sueldo |
+------------+--------+
| Ana Karina | 300000 |
| Tiberio    | 800000 |
| Ana Karina | 500000 |
+------------+--------+
```

Los operadores regexp y no regexp tampoco son sensibles a mayúsculas – minúsculas. Igualmente, se puede usar la comilla sencilla o la doble (siempre que se cierre con la misma comilla que con la que se abra).

b) Lista de empleados cuyo nombre NO tiene a “in” por subcadena (Ejemplo: Jaider):

```
mysql>      select * from Empleado where nombre not regexp 'in';
```

c) Lista de empleados cuyo nombre tiene por lo menos una *o*, una *u* o una *k* (Ejemplo: Ana Karina):

```
mysql>      select * from Empleado
            where nombre regexp '[ouk]';
```

d) Lista de empleados tales que no sea cierto que su nombre tiene por lo menos una o, una u o una *k*; es decir, cuyo nombre no contiene ninguna de las letras *o*, *u* ni *k* (Ejemplo: Daniel):

```
mysql>      select * from Empleado
            where nombre not regexp '[ouk]';
```

e) Lista de empleados cuyo nombre tiene por lo menos una de las letras desde la m hasta la p; es decir, por lo menos una de las letras m, n, o, p (Ejemplo: Adriana):

```
mysql>      select * from Empleado
            where nombre regexp '[m-p]';
```

f) Lista de empleados cuyo nombre tiene por lo menos una de las letras m, n, o, p, q, i, j, k, l, c, f (Ejemplos: Adriana, Jaider):

```
mysql>      select * from Empleado
            where nombre regexp '[m-qi-lcf]';
```

g) Lista de empleados cuyo nombre comienza por “Cr” (Ejemplo: Cristina):

```
mysql>      select * from Empleado
            where nombre regexp '^cr';
```

h) Lista de empleados cuyo nombre termina en “ra” (Ejemplo: Sandra):

```
mysql>      select * from Empleado
            where nombre regexp 'ra$';
```

i) Lista de empleados cuyo nombre tiene en alguna parte una *a* seguida por un carácter cualquiera y luego por una *i* (Ejemplo: Ana Karina).

```
mysql>      select * from Empleado
            where nombre regexp 'a.i';
```

El punto es un comodín que se empareja con un carácter cualquiera.

j) Lista de empleados cuyo nombre tiene en alguna parte una *a* seguida de dos caracteres cualesquiera y luego de una *e* (Ejemplo: Jaider).

```
mysql>      select * from Empleado
            where nombre regexp 'a..e';
```

k) Lista de empleados cuyo nombre tenga al menos cinco caracteres (Ejemplo: Carolina).

```
mysql>      select * from Empleado
            where nombre regexp '.....';
```

l) Lista de empleados cuyo nombre tenga exactamente cinco caracteres (Ejemplo: Pedro):

```
mysql>      select * from Empleado where nombre regexp '^.....$';
```

Cuando están juntos, los caracteres ^ $ se interpretan como tener exactamente los caracteres indicados por la cadena patrón.

La siguiente instrucción es equivalente a la anterior. El número entre llaves evita tener que repetir un carácter en la cadena patrón.

```
mysql>      select * from Empleado where nombre regexp '^.{5}$';
```

m) Lista de empleados cuyo nombre contenga una i seguida de un número cualquiera de caracteres y de otra i (es decir, cuyo nombre tenga dos ies, como por ejemplo: Cristina):

```
mysql>      select * from Empleado where nombre regexp 'i.*i';
```

El asterisco se interpreta como cualquier número de veces lo inmediatamente anterior, en este caso .* representa cualquier número de puntos que se emparejarán con cualquier cantidad de caracteres (recuerde la cerradura de estrella en Teoría de la Computación). En conclusión .* se empareja con cualquier número de caracteres (incluido cero).

n) Lista de empleados de nombre conformado únicamente por caracteres de la a a la z; esto excluye las vocales con tildes, los dígitos etc.

```
mysql>      select * from Empleado where nombre regexp "^[a-z]*$";
```

4.6 Ejercicios

1. Usando los operadores para conjuntos y rangos elabore la instrucción SQL que realiza las siguientes consultas. Use solo una instrucción SQL por cada numeral. Con la instrucción se debe obtener exactamente lo solicitado y nada más:

a. Las columnas número de identificación y nombre de empleado cuyo número de identificación esté entre 400 y 600, incluidos 400 y 600. El resultado debe estar ordenado por nombre del empleado, de la *A* a la *Z*.

b. Lista de los empleados de las dependencias Presupuesto y Biblioteca. Deben salir primero los de biblioteca y después los de contabilidad.

2. Elabore las consultas SQL necesarias para producir los siguientes resultados. Use solo una instrucción SQL por cada numeral. Con la instrucción se debe obtener exactamente lo solicitado y nada más:

a) Una lista exactamente con el nombre, el sexo y la fecha de nacimiento de los empleados, en orden de sexo (primero las mujeres) – fecha de nacimiento (primero el menor).

b) Una lista de empleados en orden de sexo (primero los hombres) - fecha de nacimiento (primero el mayor).

3. Usando los operadores *like* y *not like*, escriba las instrucciones SQL que producen los siguientes resultados. Use solo una instrucción SQL por cada numeral. Con la instrucción se debe obtener exactamente lo solicitado y nada más:

a. Una lista de los empleados cuyo nombre termina en “a”.

b. Una lista de los empleados cuyo nombre termina en “an”.

c. Una lista de los empleados cuyo nombre tiene al menos una letra “t”.

d. Una lista de los empleados cuyo nombre tiene a “fa” por subcadena.

e. Una lista de los empleados cuyo nombre tiene al menos seis caracteres.

f. Una lista de los empleados cuyo nombre termina en “a” y tiene al menos una “r”.

g. Una lista de los empleados cuyo nombre tiene una “r” en la tercera posición.

h. Una lista de los empleados cuyo nombre comienza con “M” y tiene una “t” en la cuarta posición.

i. Una lista de los empleados cuyo nombre comienza con “S” y termina en “a” (Ejemplo: Sandra).

j. Una lista de los empleados cuyo nombre comienza por “S” o por “M”.

k. Una lista de empleados cuyo nombre tiene una “a” en la segunda posición y una “o” en la cuarta posición. (Ejemplo: Carolina)

l. Una lista de los empleados cuyo nombre tiene al menos una “i” y una “a”. (Ejemplos: Mario y Jimena).

4. Usando los operadores *regexp* y *not regexp*, escriba las instrucciones SQL que producen los siguientes resultados. Use solo una instrucción SQL por cada numeral. Con la instrucción se debe obtener exactamente lo solicitado y nada más:

a. Lista de los empleados cuyo nombre termina en *a* o en *o*.

b. Lista de los empleados cuyo nombre comienza con *c* o con *m*.

c. Lista de los empleados cuyo nombre comienza con *s* y termina en *a*.

d. Lista de los empleados cuyo nombre incluye solamente letras de la *a* a la *k* y de la *q* a la *s* (Ejemplo: Jaider).

e. Lista de los empleados cuyo nombre tiene alguna tilde.

f. Lista de los empleados cuyo nombre comienza con S, termina en a, y que además contiene una letra d (Ejemplo: Sandra).

4.7 Respuestas:

```
1.a.  select idEmpleado, nombre from Empleado where idEmpleado between 400 and 600
      order by nombre;

1.b.  select * from Empleado
      where dependencia in ( 'Presupuesto', 'Biblioteca')
      order by dependencia;

2.a.  select nombre, sexo, fechaNacimiento from Empleado
      order by sexo, fechaNacimiento desc;

2.b.  select * from Empleado order by sexo desc, fechaNacimiento;

3.a.  select * from Empleado where nombre like "%a";

3.b.  select * from Empleado where nombre like "%an";

3.c.  select * from Empleado where nombre like "%t%";

3.d.  select * from Empleado where nombre like "%fa%";

3.e.  select * from Empleado where nombre like "______%";

3.f.  select * from Empleado where nombre like "%r%a";

3.g.  select * from Empleado where nombre like "__r%";

3.h.  select * from Empleado where nombre like "M__t%";

3.i.  select * from Empleado where nombre like "s%a";

3.j.  select * from Empleado where nombre like "s%" or nombre like "m%";

3.k.  select * from Empleado where nombre like "_a_o%";

3.l.  select * from Empleado where nombre like "%i%a%" or nombre like "%a%i%";

4.a.  select * from Empleado where nombre regexp "[ao]$";

4.b.  select * from Empleado where nombre regexp "^[cm]";

4.c.  select * from Empleado where nombre regexp "^s.*a$";

4.d.  select * from Empleado where nombre regexp "^[a-kq-s]*$";

4.e.  select * from Empleado where nombre regexp "[áéíóú]";

4.f.  select * from Empleado where nombre regexp "^s.*d.*a$";
```