# BASES DE DATOS

FES Aragón

ICO

Dr. Omar Mendoza González

# Operadores SQL

| Operator | Significado |
|---|---|
| ALL | TRUE si el conjunto completo de comparaciones es TRUE. |
| AND | TRUE si ambas expresiones booleanas son TRUE. |
| ANY | TRUE si cualquier miembro del conjunto de comparaciones es TRUE. |
| BETWEEN | TRUE si el operando está dentro de un intervalo. |
| EXISTS | TRUE si una subconsulta contiene cualquiera de las filas. |
| IN | TRUE si el operando es igual a uno de la lista de expresiones. |
| LIKE | TRUE si el operando coincide con un patrón. |
| NOT | Invierte el valor de cualquier otro operador booleano. |
| OR | TRUE si cualquiera de las dos expresiones booleanas es TRUE. |
| SOME | TRUE si alguna de las comparaciones de un conjunto es TRUE. |

# Lenguaje de Manejo de Datos (LMD)

- OPERADORES

```
SELECT * FROM alumnos
WHERE sexo = 'M'
AND ciudad = 'QUERETARO';

SELECT * FROM alumnos
WHERE sexo = 'M'
OR ciudad = 'QUERETARO';
```

# Lenguaje de Manejo de Datos (LMD)

- OPERADORES

```
SELECT * FROM alumnos
WHERE estatura >= '1.60'
AND estatura <='1.68’;

SELECT * FROM alumnos
WHERE estatura
BETWEEN '1.60' AND '1.68’;
```

# Lenguaje de Manejo de Datos (LMD)

- OPERADORES

```
SELECT * FROM alumnos
WHERE estatura
NOT BETWEEN '1.60' AND '1.68’;
```

# Operadores de cadena

- LIKE
  - %
  - _
  - NOT

  - str LIKE 'abc%’ LEFT(str,3) = 'abc'
  - str LIKE '%abc’ RIGHT(str,3) = 'abc'

# Operadores de cadena

- BINARY

```
SELECT "HOLA" = "hola";

SELECT BINARY  "HOLA" = "hola";
```

# Lenguaje de Manejo de Datos (LMD)

- OPERADORES

```
SELECT * FROM alumnos
WHERE colonia LIKE 'LOMAS%';

SELECT * FROM alumnos
WHERE colonia NOT LIKE 'LOMAS%';
```

# Lenguaje de Manejo de Datos (LMD)

- OPERADORES

```
SELECT * FROM alumnos WHERE nombre LIKE 'A%';

SELECT * FROM alumnos WHERE nombre LIKE '%A';

SELECT * FROM alumnos WHERE nombre LIKE '__A%';

SELECT * FROM alumnos WHERE nombre LIKE '%A';

SELECT * FROM alumnos WHERE nombre LIKE '_____';
```

# Lenguaje de Manejo de Datos (LMD)

- OPERADORES

```
SELECT * FROM alumnos
 WHERE nombre IN ('RAFAEL', 'OSCAR', 'GABRIELA', 'FABIOLA');

SELECT * FROM alumnos
WHERE nombre NOT IN ('RAFAEL', 'OSCAR', 'GABRIELA', 'FABIOLA');
```

# Lenguaje de Manejo de Datos (LMD)

- OPERADORES

- IS NULL

```
SELECT * FROM alumnos
WHERE peso IS NULL ;
```

```
SELECT * FROM alumnos
WHERE peso IS NOT NULL ;
```

# Lenguaje de Manejo de Datos (LMD)

- Funciones de Agregación
  - AVG
  - COUNT
  - SUM
  - MAX
  - MIN

# Lenguaje de Manejo de Datos (LMD)

- Funciones de Agregación

```
SELECT AVG(peso),
COUNT(*),
SUM(peso),
MAX(peso),
MIN(peso)
FROM alumnos;
```

# Lenguaje de Manejo de Datos (LMD)

- Funciones de Agregación

```
SELECT sexo, AVG(peso), COUNT(*),
SUM(peso), MAX(peso), MIN(peso)
FROM alumnos
GROUP BY sexo;
```

# Operadores de relación

| Operador | Significado |
|---|---|
| = | Igual a |
| <> | Distinto de |
| > | Mayor que |
| >= | Mayor o igual que |
| < | Menor que |
| <= | Menor o igual que |
| Like | Coincidencia con un patrón |
| Not Like | No coincidencia con un patrón |
| Is Null | Igual a nulo (vacío) |
| Is Not Null | No es nulo (no está vacío) |
| Between | Rango de valores entre una cota inferior y otra superior |
| In | Pertenencia a un conjunto de valores o ser miembro de una subconsulta |
| Exist | Cierto si una subconsulta devuelve como mínimo un registro |
| Any | Compara un valor con cada valor devuelto por una subconsulta retornando cierto si uno cualquiera de ellos cumple la condición. |
| All | Compara un valor con cada valor devuelto por una subconsulta retornando cierto si todos ellos cumplen la condición. |

# Cadenas

- SELECT nombre, nombre = 'EMILIO', nombre != 'EMILIO', nombre >= 'EMILIO', nombre <= 'EMILIO' from alumnos

- SELECT nombre, nombre = 'emilio', BINARY nombre = 'emilio' from alumnos

# Funciones de cadena

- **CONCAT()**
  - *CONCAT(str1,str2,...)*
  - Devuelve la cadena resultante de concatenar los argumentos
  - SELECT CONCAT(nombre,' ',ap_paterno) from alumnos

# Funciones de cadena

- **INSERT()**
  - INSERT(str,pos,len,newstr)
  - Devueve la cadena str, con la subcadena que empieza en la posición pos y de len caracteres de longitud remplazada con la cadena newstr
  - SELECT nombre, ap_paterno, INSERT(nombre, 3, 4, ap_paterno) from alumnos

# Funciones de cadena

- **LOWER()**
  - LCASE()
- **UPPER**
  - UCASE()
  - SELECT LOWER(nombre), UPPER(NOMBRE) from alumnos

# Funciones de cadena

- **LEFT()**
  - LEFT(cadena,longitud)
  - Devuelve los 'longitud' caracteres de la izquierda de la 'cadena'
  - SELECT LEFT(nombre, 5) from alumnos

# Funciones de cadena

- **RIGHT()**
  - RIGHT(cadena,longitud)
  - Devuelve los 'longitud' caracteres de la derecha de la 'cadena'
  - SELECT RIGHT(nombre, 5) from alumnos

# Funciones de cadena

- **SUBSTRING()**
  - MID()
  - SUBSTRING(cadena,posicion)
  - SUBSTRING(cadena,posicion,longitud)
  - SELECT SUBSTRING(nombre, 5) from alumnos
  - SELECT SUBSTRING(nombre, 5, 3) from alumnos

# Funciones de cadena

- **LENGTH()**
  - LENGTH(str)
  - Devuelve la longitud de la cadena str,
  - SELECT LENGTH(nombre) from alumnos

# Funciones de cadena

- **LOCATE()**
  - POSITION()
  - LOCATE(substr,str)
  - LOCATE(substr,str,pos)
  - POSITION(substr IN str)
  - Devuelve la posición de la primer aparición de la cadena substr dentro de la cadena str.
  - SELECT LOCATE('A', nombre), LOCATE('A', nombre, 4) from alumnos

# Funciones de cadena

- **REPEAT()**
  - REPEAT(str,count)
  - Devuelve una cadena que consiste en la cadena str repetida count veces.
  - Si count <= 0, devuelve una cadena vacía. Devuelve NULL si str o count son NULL
  - 
  - SELECT REPEAT(nombre, 4) from alumnos

# Funciones de cadena

- **REPLACE()**
  - REPLACE(str,from_str,to_str)
  - Devuelve la cadena str con todas las apariciones de la cadena from_str sustituídas por la cadena to_str
  - 
  - SELECT REPLACE(nombre, ' ', '*') from alumnos

# Funciones de cadena

- **REVERSE()**
  - REVERSE(str)
  - Devuelve la cadena str con el orden de los caracteres invertido
  - 
  - SELECT REVERSE(nombre) from alumnos

# Funciones de cadena

- SELECT nombre from alumnos WHERE LEFT(nombre,1) >= 'n’
- SELECT CONCAT('Hola, ',USER( ),',Estas en BD')
- SELECT CONCAT(nombre,' termina en "O": ',IF(RIGHT(nombre,1)='o','SI','NO')) AS ’Termina en ”o"?' FROM alumnos

# Funciones de cadena

- UPDATE alumnos
  SET nombre = CONCAT(nombre,'ide')

- UPDATE alumnos
  SET nombre =
  LEFT(nombre,LENGTH(nombre)-3);