

INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE INGENIERÍA EN SISTEMAS COMPUTACIONALES
ACADEMIA DE BASES DE DATOS
UNIDAD DE APRENDIZAJE: BASES DE DATOS
Prof. Dra. Fabiola Ocampo Botello
Semestre Escolar Enero-Julio 2017
Grupos: 2CM3 y 2CM5

Práctica No. 3. Funciones para el manejo de caracteres y cadenas de caracteres

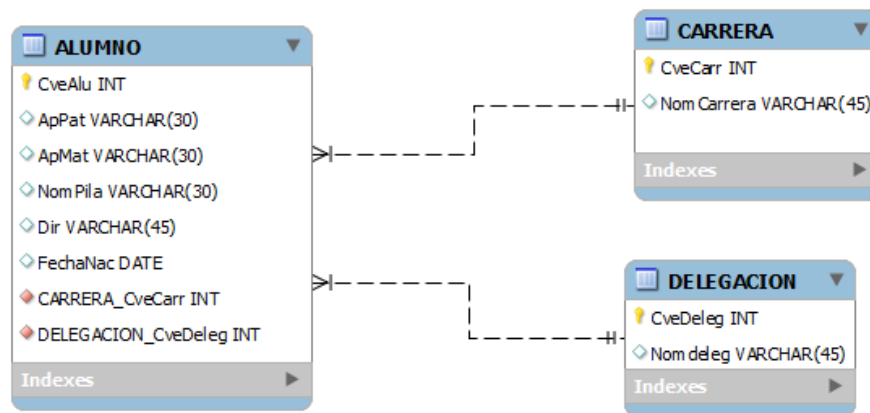


Figura 1. Modelo de la base de datos

1. Crear la base de datos con el siguiente comando:
SOURCE e: \BD Pract 3.txt;
2. Funciones para el manejo de caracteres.

Pregunta 1. Crea una vista llamada AlumEdad que almacene la clave, apellido paterno, apellido materno, nombre de pila y la edad de las personas.

- a) Introduzca la siguiente consulta.
- ```
SELECT CONCAT_WS(',', CveAlu, ApPat, ApMat, NomPila, Edad, '\n') FROM AlumEdad;
```

**Pregunta 2.** Con la sentencia INTO ... OUTFILE exporte la vista AlumEdad a un archivo de texto llamado AlumEdad.txt.

- b) Cree una tabla llamada EdadPrueba e importe los datos del archivo de texto AlumEdad.txt recién creado.

**Pregunta 3.** Sentencias de la creación de la tabla.

**Pregunta 4.** Sentencia para importar los datos.

- c) Ejemplo. Introduzca la siguiente sentencia.  
`SELECT CHAR (72,79,76,65);`
- d) Ejemplo. Introduzca la siguiente sentencia.  
`SELECT CONCAT (CAST(CveAlu AS CHAR), ': ', Appat, ': ', ApMat, ': ', nompila, ': ',  
CAST(Edad AS CHAR)) FROM EdadPrueba;`  
  
`SELECT CONCAT (Cvealu, ': ', Appat, ': ', ApMat, ': ', nompila, ': ', Edad) FROM  
EdadPrueba;`
- e) Ejemplo: Uso de variables y sentencia IF.  
`SET @Var1 = (SELECT Edad FROM EdadPrueba WHERE Cvealu = 10);  
SELECT IF (@Var1 > 30, 'Mayor de 30 años', 'Menor de 30 años');`
- f) Ejemplo: Sentencia ELT. Retorna str1 si N = 1, str2 if N = 2, y así. Retorna NULL si N es menor que 1 o mayor que el número de argumentos.  
`SET @Mes = (SELECT MONTH(FechaNac) FROM Alumno WHERE Cvealu = 8);  
SELECT ELT (@Mes%2 + 1, 'Nació en mes Par', 'Nació en mes Impar');`
- g) Ejemplo de funciones de mayúsculas y minúsculas.  
`SET @CadEj = 'El más grande de mis amigos es: '  
SET @MaxEdad = (SELECT MAX(edad) FROM AlumEdad);  
SELECT @MaxEdad;  
SET @Mayor = (SELECT nompila FROM AlumEdad WHERE Edad = @MaxEdad);  
SELECT @Mayor;  
SELECT LENGTH (@Mayor);  
SELECT LENGTH (@CadEj);  
  
SELECT CONCAT (@CadEj, @Mayor);  
SELECT LCASE (CONCAT (@CadEj, @Mayor));  
SELECT LOWER (CONCAT (@CadEj, @Mayor));  
SELECT UCASE (CONCAT (@CadEj, @Mayor));  
SELECT UPPER (CONCAT (@CadEj, @Mayor));`

- h) Ejemplo: Sentencia LEFT. Retorna la cantidad de caracteres especificados.  
`SELECT LEFT (@Mayor, 3);`
- i) Ejemplo: Sentencia LPAD. Regresa la cadena inicial, completando a la izquierda con el carácter especificado hasta alcanzar la longitud establecida.  
`SELECT LPAD (@Mayor, 10, '*');`
- j) Ejemplo: Sentencia LTRIM. Retorna la cadena sin los espacios de la izquierda.  
`SET @VarAux = ' Hola';`  
`SELECT (LENGTH(@VarAux));`  
`SELECT (LENGTH (LTRIM (@VarAux)));`
- k) Ejemplo: Sentencia RPAD. Regresa la cadena inicial, completando a la derecha con el carácter especificado hasta alcanzar la longitud establecida.  
`SELECT RPAD (@Mayor, 10, '*');`
- l) Ejemplo: Sentencia RTRIM. Retorna la cadena sin los espacios de la izquierda.  
`SET @VarAux = 'Hola ';`  
`SELECT (LENGTH(@VarAux));`  
`SELECT (LENGTH (RTRIM (@VarAux)));`
- m) Ejemplo: Sentencia REPEAT (str, conta). Regresa la cadena *str* *conta* veces.  
`SELECT REPEAT (@Mayor, 3);`
- n) Ejemplo: Sentencia REVERSE(str). Retorna la cadena *str* invertida.  
`SELECT REVERSE (@Mayor);`

### 3. Funciones de comparación de cadenas de caracteres.

MySQL convierte automáticamente los números a cadenas de caracteres y viceversa. Aunque también lo puede realizar mediante la función CAST.

LIKE puede ser utilizado para comparar una cadena de caracteres con un patrón específico. También se pueden comparar números. Con LIKE puede utilizar los siguientes comodines.

% Coincidencia de cualquier número de caracteres.

\_ Coincidencia de exactamente un carácter.

- o) Ejemplo: Uso de sentencia LIKE.
  - Para conocer cuántos alumnos se llaman Sofía.  
`SELECT COUNT(*) FROM Alumno`  
`WHERE nombra LIKE 'Sofia';`
  - Para saber cuántos apellidos paternos empiezan con 'H'.  
`SELECT COUNT(*) FROM Alumno`  
`WHERE appat LIKE 'H%';`
  - Mostrar todos los apellidos paternos que empiezan con 'G'  
`SELECT appat FROM Alumno`  
`WHERE appat LIKE 'G%';`

- Mostrar una sola vez los apellidos paternos que empiezan con 'G'  
SELECT DISTINCT appat FROM Alumno  
WHERE appat LIKE 'G%';
- Listar el nombre de los alumnos que viven en la delegación Coyoacán.  
SELECT cvealu, nompila, appat, apmat  
FROM Alumno  
WHERE cvedeleg IN (SELECT cvedeleg FROM Delegacion WHERE nomdeleg LIKE '%Coyo%');

Otra forma de responder la consulta:

```
SET @IdCoyo = (SELECT cvedeleg FROM Delegacion WHERE nomdeleg LIKE '%Coyo%');
SELECT @IdCoyo;
SELECT cvealu, nompila, appat, apmat
FROM Alumno
WHERE cvedeleg = @IdCoyo;
```

#### 4. Ejemplos de consultas.

A continuación se muestran algunos ejemplos de consultas anidadas aplicando las sentencias que ya se han visto a lo largo de las prácticas.

Recuerde los datos (atributos) que se han almacenado en cada una de las tablas. Esto es muy importante para que decida cuáles utilizar.

- p) Listar el nombre de los alumnos que viven en la delegación Coyoacán.  
Relaciones a utilizar: Alumno, Delegacion.  
DESCRIBE Alumno;  
DESCRIBE Delegacion;  
SELECT nompila, appat, apmat  
FROM Alumno  
WHERE cvedeleg = (SELECT cvedeleg FROM Delegacion WHERE nomdeleg LIKE '%Coyoacan%');
- q) Listar el nombre de los alumnos que viven en la delegación Álvaro Obregón y están en la carrera de Ingeniería.  
Relaciones a utilizar: Alumno, Delegacion, Carrera.  
SELECT nompila, appat, apmat  
FROM Alumno  
WHERE cvedeleg = (SELECT cvedeleg FROM Delegacion WHERE nomdeleg LIKE '%Alvaro%')  
AND  
cvecarr = (SELECT cvecarr FROM Carrera WHERE nomcarrera LIKE '%Ingenieria%');
- r) Encontrar el nombre de los alumnos que cumplen años en los meses de enero a junio.  
Relaciones a utilizar: Alumno.  
SELECT nompila, appat, apmat  
FROM Alumno  
WHERE MONTH(FechaNac) <= 6;

- s) Encontrar la cantidad de alumnos que cursan cada carrera.

Relaciones a utilizar: Alumno.

```
SELECT cvecarr, COUNT(*) FROM Alumno
GROUP BY cvecarr;
```

- t) Los alumnos que cursan la carrera de Economía y viven en la delegación Milpa Alta.

Relaciones a utilizar: Alumno, Delegacion, Carrera.

- Clave de la delegación Milpa Alta.

```
SET @CveMA = (SELECT cvedeleg FROM Delegacion WHERE LOWER(nomdeleg) LIKE
'%milpa%');
```

- Clave de la carrera de Ciencias.

```
SET @CveEcon = (SELECT cvecarr FROM Carrera WHERE LOWER (nomcarrera) LIKE
'%econ%');
```

```
SELECT cvealu, nompila, appat, apmat
FROM Alumno
WHERE cvedeleg = @CveMA AND cvecarr = @CveEcon;
```

Otra forma de la consulta, sin utilizar variables.

```
SELECT cvealu, nompila, appat, apmat
FROM Alumno
WHERE cvedeleg = (SELECT cvedeleg FROM Delegacion WHERE LOWER(nomdeleg) LIKE
'%milpa%')
AND
cvecarr = (SELECT cvecarr FROM Carrera WHERE LOWER (nomcarrera) LIKE '%econ%');
```

- u) Los alumnos que tienen edad entre 20 y 25 años.

```
SELECT *
```

```
FROM AlumEdad
```

```
WHERE Edad BETWEEN 20 AND 25;
```

**Pregunta 5.** Encontrar los alumnos que tienen una edad entre 20 y 30 años y viven en la delegación Tlalpan. Una consulta con variables y otra sin variables.

**SEGUNDA PARTE. Escriba con sus propias palabras la función que tienen los comandos vistos en esta práctica y otros más.**

Consulte el manual en línea de MySQL sólo para apoyarse en el entendimiento de las funciones. NO transcriba lo escrito en el manual.

**Tabla 1.** Comandos vistos en la práctica y otros más.

| Comando                                                  | Función |
|----------------------------------------------------------|---------|
| <b>Funciones para el manejo de caracteres</b>            |         |
| ASCII(str)                                               |         |
| BIN(N)                                                   |         |
| BIT_LENGTH(str)                                          |         |
| CHAR(N, ...)                                             |         |
| CHAR_LENGTH(str)                                         |         |
| CHARACTER_LENGTH(str)                                    |         |
| CONCAT(str1,str2,...)                                    |         |
| CONCAT_WS(separador,str1,str2,...)                       |         |
| CONV(N,DeBase,ABase)                                     |         |
| ELT(N,str1,str2,str3,...)                                |         |
| FIELD(str,str1,str2,str3,...)                            |         |
| FIND_IN_SET(str,listastr)                                |         |
| INSERT(str,pos,len,nuevastr)                             |         |
| INSTR(str,substr)                                        |         |
| LCASE(str)                                               |         |
| LOWER(str)                                               |         |
| LEFT(str,len)                                            |         |
| LENGTH(str)                                              |         |
| LOCATE(substr,str)                                       |         |
| LOCATE(substr,str,pos)                                   |         |
| LPAD(str,len,padstr)                                     |         |
| MID(str,pos,len)                                         |         |
| ORD()                                                    |         |
| SUBSTRING(str,pos,len)                                   |         |
| POSITION(substr IN str)                                  |         |
| REPEAT(str,contador)                                     |         |
| REPLACE(str,desde_str,hasta_str)                         |         |
| REVERSE(str)                                             |         |
| RIGHT(str,len)                                           |         |
| RPAD(str,len,padstr)                                     |         |
| RTRIM(str)                                               |         |
| SPACE(N)                                                 |         |
| TRIM                                                     |         |
| UCASE(str)                                               |         |
| UPPER (str)                                              |         |
|                                                          |         |
| <b>Funciones para el manejo de cadenas de caracteres</b> |         |
| DISTINCT                                                 |         |
| LIKE                                                     |         |
| NOT LIKE                                                 |         |
| STRCMP                                                   |         |
|                                                          |         |