

Nombre de la práctica	Introducción a SQL (UNIDAD 5)			No.	1
Asignatura:	FUNDAMENTOS DE BASES DE DATOS	Carrera:	INGENIERÍA EN SISTEMAS COMPUTACIONALES	Duración de la práctica (Hrs)	5 horas

NOMBRE DEL ALUMNO: Ana Edith Hernández
Hernández GRUPO: 3401

I. Competencia(s) específica(s):

Conoce y aplica el modelo relacional para la generación de esquemas de base de datos con el fin de organizar la información y atender necesidades del entorno.

Encuadre con CACEI: Registra el (los) atributo(s) de egreso y los criterios de desempeño que se evaluarán en esta práctica.

No. atributo	Atributos de egreso del PE que impactan en la asignatura	No. Criterio	Criterios de desempeño	No. Indicador	Indicadores
2	El estudiante diseñará esquemas de trabajo y procesos, usando metodologías congruentes en la resolución de problemas de ingeniería en sistemas computacionales	CD1	Identifica metodologías y procesos empleados en la resolución de problemas	I1	Identificación y reconocimiento de distintas metodologías para la resolución de problemas
				I2	Manejo de procesos específicos en la solución de problemas y/o detección de necesidades
		CD2	Diseña soluciones a problemas, empleando metodologías apropiadas al área	I1	Uso de metodologías para el modelado de la solución de sistemas y aplicaciones
3	El estudiante plantea soluciones basadas en tecnologías empleando su juicio ingenieril para valorar necesidades, recursos y resultados esperados.	CD1	Emplea los conocimientos adquiridos para el desarrollar soluciones	I1	Elección de metodologías, técnicas y/o herramientas para el desarrollo de soluciones
				I2	Uso de metodologías adecuadas para el desarrollo de proyectos
				I3	Generación de productos y/o proyectos
		CD2	Analiza y comprueba resultados	I1	Realizar pruebas a los productos obtenidos
				I2	Documentar información de las pruebas realizadas y los resultados

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):

Laboratorio de cómputo y equipo de cómputo personal.

III. Material empleado:

- Equipo de cómputo
- MySQL/MariaDB

IV. Desarrollo de la práctica:

UNIDAD 3

No. atributo	Atributos de egreso del PE que impactan en la asignatura	No. Criterio	Criterios de desempeño	No. Indicador	Indicadores
2	El estudiante diseñará esquemas de trabajo y procesos, usando metodologías congruentes en la resolución de problemas de ingeniería en sistemas computacionales	CD1	Identifica metodologías y procesos empleados en la resolución de problemas	I1	Identificación y reconocimiento de distintas metodologías para la resolución de problemas
				I2	Manejo de procesos específicos en problemas y/o detección de necesidades
		CD2	Diseña soluciones a problemas, empleando metodologías apropiadas al área	I1	Uso de metodologías para el desarrollo de soluciones de sistemas y aplicaciones
3	El estudiante plantea soluciones basadas en tecnologías empleando su juicio ingenieril para valorar necesidades, recursos y resultados esperados.	CD1	Emplea los conocimientos adquiridos para el desarrollo de soluciones	I1	Elección de metodologías y técnicas y/o para el desarrollo de soluciones
				I2	Uso de metodologías adecuadas para proyectos
				I3	Generación de productos y/o proyectos
		CD2	Analiza y comprueba resultados	I1	Realizar pruebas a los productos obtenidos
				I2	Documentar información de las pruebas y los resultados

UNIDAD V. ALGEBRA RELACIONAL

INSTRUCCIONES: Genera las instrucciones en sql según se solicite

ANIMALES

noAnimal	tipo	genero	Edad
1	Gato	Macho	2
2	Iguana	Hembra	1
3	Hamster	Hembra	2
4	Hamster	Macho	1
5	Perro	Hembra	2

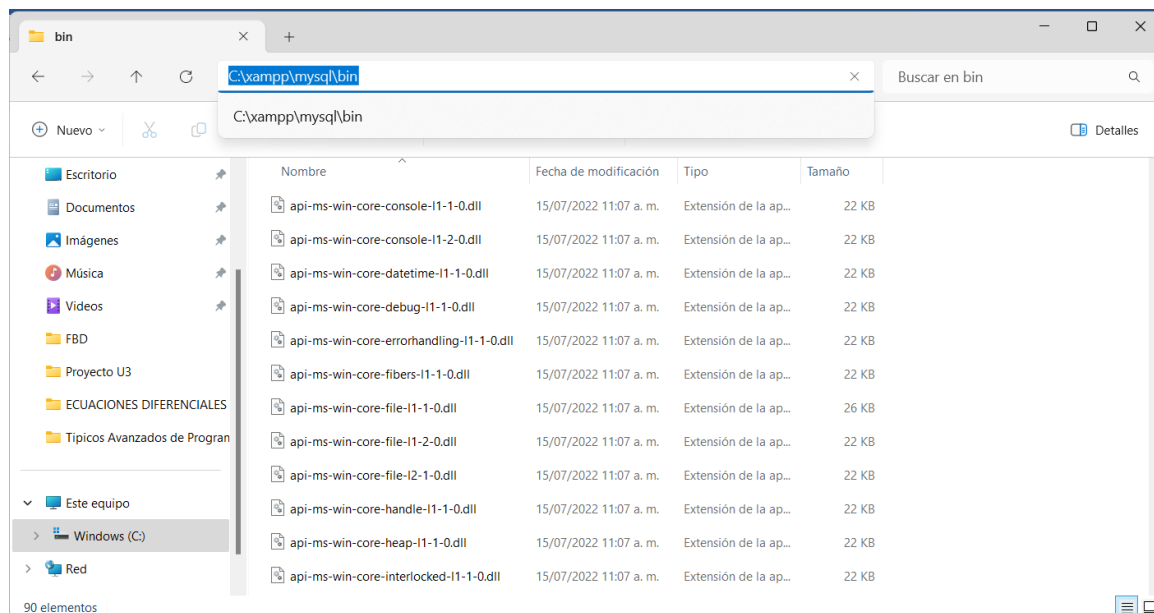
VACUNAS

VACUNACION ANIMAL

idVacuna	descripcion
A1	Parvovirus
A2	Tétanos
A3	Moquillo
A4	Rabia
A5	Pentavalente

noAnimal	idVacuna	fechaAplicacion
1	A2	20/03/12
2	A3	26/06/12
3	A4	27/06/12
4	A5	27/06/12
5	A1	25/06/11

1. Comenzaremos entando a Explorador de Archivos en la siguiente ruta:



Posterior a eso en la barra colocamos CMD y nos abrirá una terminal con **MySQL Maria DB**
Lo inicializamos con el siguiente comando **mysql -u root**

```
C:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

2. Genera código sql para definir la base de datos cuyo nombre es “ControlVeterinario”.

Creamos la base de datos:

```
MariaDB [(none)]> CREATE DATABASE ControlVeterinario;
Query OK, 1 row affected (0.013 sec)

MariaDB [(none)]> USE ControlVeterinario;
Database changed
```

Y la “activamos” para comenzarla a usar

2. Genera el código sql para crear la tabla animales, vacunas y animal.

Creamos tabla Animales:

```
MariaDB [ControlVeterinario]> CREATE TABLE animales (
    -> noAnimal INT PRIMARY KEY,
    -> tipo VARCHAR(50),
    -> genero VARCHAR(10),
    -> edad INT
    -> );
Query OK, 0 rows affected (0.033 sec)
```

Se hacen los registros

```
MariaDB [ControlVeterinario]> INSERT INTO animales (noAnimal, tipo, genero,
edad) VALUES (1, 'Gato', 'Macho', 2);
Query OK, 1 row affected (0.018 sec)

MariaDB [ControlVeterinario]> INSERT INTO animales (noAnimal, tipo, genero,
edad) VALUES (2, 'Iguana', 'Hembra', 1);
Query OK, 1 row affected (0.004 sec)

MariaDB [ControlVeterinario]> INSERT INTO animales (noAnimal, tipo, genero,
edad) VALUES (3, 'Hamster', 'Hembra', 1);
Query OK, 1 row affected (0.005 sec)

MariaDB [ControlVeterinario]> INSERT INTO animales (noAnimal, tipo, genero,
edad) VALUES (4, 'Hamster', 'Macho', 1);
Query OK, 1 row affected (0.013 sec)

MariaDB [ControlVeterinario]> INSERT INTO animales (noAnimal, tipo, genero,
edad) VALUES (5, 'Perro', 'Hembra', 2);
Query OK, 1 row affected (0.004 sec)

MariaDB [ControlVeterinario]> INSERT INTO vacunas (idVacuna, descripcion) VA
LUES ('A1', 'Parvovirus');
Query OK, 1 row affected (0.012 sec)
```

Creamos tabla Vacunas:

```
MariaDB [ControlVeterinario]> CREATE TABLE vacunas (
-> idVacuna VARCHAR(10) PRIMARY KEY,
-> descripcion VARCHAR(50)
-> );
Query OK, 0 rows affected (0.020 sec)
```

Se hacen los registros.

```
MariaDB [ControlVeterinario]> INSERT INTO vacunas (idVacuna, descripcion) VA
LUES ('A2', 'Tétanos');
Query OK, 1 row affected (0.004 sec)

MariaDB [ControlVeterinario]> INSERT INTO vacunas (idVacuna, descripcion) VA
LUES ('A3', 'Moquillo');
Query OK, 1 row affected (0.004 sec)

MariaDB [ControlVeterinario]> INSERT INTO vacunas (idVacuna, descripcion) VA
LUES ('A4', 'Rabia');
Query OK, 1 row affected (0.004 sec)

MariaDB [ControlVeterinario]> INSERT INTO vacunas (idVacuna, descripcion) VA
LUES ('A5', 'Pentavalente');
Query OK, 1 row affected (0.004 sec)
```

Creamos tabla Vacunacion_Animal:

```
MariaDB [ControlVeterinario]> CREATE TABLE vacunacion_animal (  
-> noAnimal INT,  
-> idVacuna VARCHAR(10),  
-> fechaAplicacion DATE,  
-> PRIMARY KEY (noAnimal, idVacuna),  
-> FOREIGN KEY (noAnimal) REFERENCES animales(noAnimal),  
-> FOREIGN KEY (idVacuna) REFERENCES vacunas(idVacuna)  
-> );  
Query OK, 0 rows affected (0.025 sec)
```

Se hacen los registros.

```
MariaDB [ControlVeterinario]> INSERT INTO vacunacion_animal (noAnimal, idVacuna, fechaAplicacion) VALUES (1, 'A2', '2012-03-20');  
Query OK, 1 row affected (0.007 sec)  
  
MariaDB [ControlVeterinario]> INSERT INTO vacunacion_animal (noAnimal, idVacuna, fechaAplicacion) VALUES (2, 'A3', '2012-06-26');  
Query OK, 1 row affected (0.004 sec)  
  
MariaDB [ControlVeterinario]> INSERT INTO vacunacion_animal (noAnimal, idVacuna, fechaAplicacion) VALUES (3, 'A4', '2012-06-27');  
Query OK, 1 row affected (0.004 sec)  
  
MariaDB [ControlVeterinario]> INSERT INTO vacunacion_animal (noAnimal, idVacuna, fechaAplicacion) VALUES (4, 'A5', '2012-06-27');  
Query OK, 1 row affected (0.010 sec)  
  
MariaDB [ControlVeterinario]> INSERT INTO vacunacion_animal (noAnimal, idVacuna, fechaAplicacion) VALUES (5, 'A1', '2011-06-25');  
Query OK, 1 row affected (0.004 sec)
```

3. Selecciona noAnimal, edad cuyo tipo sea Hamster

```
MariaDB [ControlVeterinario]> SELECT noAnimal, edad  
-> FROM animales  
-> WHERE tipo = "Hamster";  
+-----+-----+  
| noAnimal | edad |  
+-----+-----+  
|         3 |     2 |  
|         4 |     1 |  
+-----+-----+  
2 rows in set (0.007 sec)
```

4. ¿Cuál es el promedio de edades de los animales?

```
MariaDB [ControlVeterinario]> SELECT AVG(edad) AS PromedioEdad
-> FROM animales;
+-----+
| PromedioEdad |
+-----+
|          1.6000 |
+-----+
1 row in set (0.001 sec)
```

5. Muestra el noAnimal, tipo, género y tipo de vacuna que ha recibido cada animal

```
MariaDB [ControlVeterinario]> SELECT a.noAnimal, a.tipo, a.genero, v.descripcion
-> FROM animales a
-> JOIN vacunacion_animal va ON a.noAnimal = va.noAnimal
-> JOIN vacunas v ON va.idVacuna = v.idVacuna;
+-----+-----+-----+-----+
| noAnimal | tipo   | genero | descripcion |
+-----+-----+-----+-----+
|          5 | Perro  | Hembra | Parvovirus   |
|          1 | Gato   | Macho  | Tétanos      |
|          2 | Iguana | Hembra | Moquillo     |
|          3 | Hamster | Hembra | Rabia        |
|          4 | Hamster | Macho  | Pentavalente |
+-----+-----+-----+-----+
5 rows in set (0.010 sec)
```

6. ¿Con cuántos animales cuenta la veterinaria?

```
MariaDB [ControlVeterinario]> SELECT COUNT(*) AS TotalAnimales
-> FROM animales;
+-----+
| TotalAnimales |
+-----+
|          5 |
+-----+
1 row in set (0.001 sec)
```

7. ¿Cuántos animales se vacunaron contra el parvovirus?

```
MariaDB [ControlVeterinario]> SELECT COUNT(DISTINCT va.noAnimal) AS Animales
VacunadosParvovirus
  -> FROM vacunacion_animal va
  -> JOIN vacunas v ON va.idVacuna = v.idVacuna
  -> WHERE v.descripcion = 'Parvovirus';
+-----+
| AnimalesVacunadosParvovirus |
+-----+
|                            1 |
+-----+
1 row in set (0.002 sec)

MariaDB [ControlVeterinario]>
```

Conclusión:

SQL (Structured Query Language) es el lenguaje estándar utilizado para interactuar con bases de datos relacionales. A lo largo de esta introducción, hemos explorado cómo SQL nos permite definir, manipular y consultar datos de manera eficiente. La creación de una base de datos es un proceso fundamental en el que SQL juega un papel crucial, y entender sus comandos básicos es esencial para cualquier profesional que trabaje con datos.

En primer lugar, hemos visto que los comandos DDL (Data Definition Language) son esenciales para definir la estructura de la base de datos. Comandos como CREATE DATABASE, CREATE TABLE, ALTER TABLE y DROP TABLE nos permiten crear, modificar y eliminar estructuras de datos. Estos comandos aseguran que la base de datos tenga una organización coherente y que los datos estén correctamente almacenados y relacionados.

Además, los comandos DML (Data Manipulation Language) son vitales para interactuar con los datos almacenados. Comandos como INSERT INTO, UPDATE, DELETE y SELECT nos permiten añadir, actualizar, eliminar y recuperar datos respectivamente. La capacidad de realizar consultas complejas con SELECT, incluyendo filtros (WHERE), agrupaciones (GROUP BY), y ordenamientos (ORDER BY), hace de SQL una herramienta poderosa para el análisis de datos.

También hemos aprendido la importancia de mantener la integridad de los datos a través de restricciones (CONSTRAINTS) y la creación de claves primarias (PRIMARY KEY) y foráneas (FOREIGN KEY). Estas características garantizan que los datos sean únicos, coherentes y que las relaciones entre tablas sean válidas, lo cual es fundamental para la integridad referencial de la base de datos.