



El empleo
es de todos

Mintrabajo

Sesión ADSO

Tecnólogo en Análisis y Desarrollo de Software (ADSO)



@SENAcomunica

www.sena.edu.co



- ✓ Definición
- ✓ Lenguajes claves
- ✓ Query

SQL - Lenguaje de consulta estructurada



Como ya lo mencionamos, los sistemas de bases de datos proporcionan **dos lenguajes claves**:

a. DDL (Data Definition Language) □ Lenguaje de Definición de Datos

- Es útil para especificar el **esquema** de la base de datos
- Estas instrucciones se usan para definir o modificar **la estructura** de las bases de datos (tablas, índices, esquemas, etc.).

b. DML (Data Manipulation Language) □ Lenguaje de Manipulación de Datos

- Es útil para expresar las **consultas** y las **modificaciones** de la base de datos.
- Permite a los usuarios **tener acceso** a los datos
- Estas instrucciones se usan para **gestionar los datos** dentro de las estructuras definidas con DDL.

SQL - Lenguaje de consulta estructurada



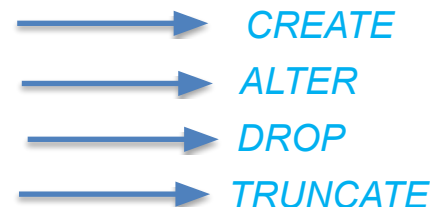
a. DDL (Data Definition Language) □ Lenguaje de Definición de Datos

- Es útil para especificar el **esquema** de la base de datos
- Estas instrucciones se usan para definir o modificar la estructura de las bases de datos (tablas, índices, esquemas, etc.).



- Algunas instrucciones DDL:

- Crear objetos en la base de datos.
- Modificar la estructura de un objeto ya existente.
- Eliminar objetos de la base de datos.
- Eliminar todos los registros de una tabla manteniendo la estructura de la tabla.



SQL - Lenguaje de consulta estructurada



b. DML (Data Manipulation Language) □ Lenguaje de Manipulación de Datos

- Es útil para expresar las **consultas** y las **modificaciones** de la base de datos.
- Permite a los usuarios **tener acceso** a los datos
- Algunos tipos de acceso son:

• Recuperación	→	<i>SELECT</i>
• Inserción	→	<i>INSERT</i>
• Borrado registros	→	<i>DELETE</i>
• Modificación	→	<i>UPDATE</i>



Nota:

Aunque SELECT a veces se clasifica como parte de DML, técnicamente pertenece a **DQL (Data Query Language)**, que se enfoca solo en consultas.

SQL - Lenguaje de consulta estructurada



Query / Consulta:

- Una consulta es una instrucción que solicita que se recupere información.
- La parte de los **LMD** que implicada en la recuperación de información se denomina ***lenguaje de consultas***.
- El lenguaje de consultas más ampliamente usado es SQL.
- En la realidad, estos dos lenguajes (el de definición y el de manipulación) **no son dos lenguajes diferentes**, al contrario, simplemente forman parte de un único lenguaje de bases de datos, como puede ser el muy usado SQL.

SQL - Lenguaje de consulta estructurada



En ocasiones, suele considerarse una categoría adicional de sentencias incluida en DML y basada en la sentencia SELECT. Esta categoría se denomina *Lenguaje de consulta de datos* DQL.

LENGUAJES DE DEFINICIÓN, MANIPULACIÓN Y CONTROL

6.1 EL LENGUAJE SQL

SQL es un lenguaje estándar completo de control e interacción con un sistema de gestión de base de datos relacional definido por la *American National Standards Institute* (ANSI) y la *International Standards Organization* (ISO). Se trata de un lenguaje para administrar, almacenar y recuperar información utilizado por la mayoría de los sistemas gestores de bases de datos actuales. Habitualmente incluye las siguientes categorías de sentencias:

- **DDL** (*Data Definition Language* o Lenguaje de Definición de Datos)
Utilizadas para crear, alterar o borrar objetos de base de datos, tales como tablas, columnas y vistas. Como ejemplos característicos tenemos las instrucciones o comandos CREATE, ALTER y DROP.
- **DML** (*Data Modification Language* o Lenguaje de Modificación de Datos)
Utilizadas en la interrogación y manipulación de datos en esquemas ya existentes. Como ejemplos característicos tenemos los comandos SELECT, INSERT, UPDATE y DELETE.
- **TCL** (*Transaction Control Language* o Lenguaje de Control de Transacciones)
Utilizadas para confirmar o restaurar transacciones de base de datos, que son unidades de trabajo que realizan una o más sentencias SQL relacionadas entre sí. Como ejemplos característicos tenemos los comandos COMMIT y ROLLBACK.
- **DCL** (*Data Control Language* o Lenguaje de Control de Datos)
Utilizadas en el control de acceso a datos en la base de datos. Como ejemplos característicos tenemos los comandos GRANT y REVOKE.

Script de base de datos



`archivo.sql`

Un script de base de datos es un conjunto de instrucciones SQL escritas en un archivo de texto o ejecutadas directamente, que se utilizan para crear, modificar, poblar o gestionar una base de datos.

Script de base de datos - Documentado



```
Ejemplo_SQL_Inicial - documen... x
Limit to 1000 rows

20
21  -- =====
22  -- DDL (Data Definition Language)
23  -- =====
24
25  -- DDL: CREATE DATABASE - Crea una nueva base de datos
26  • CREATE DATABASE Ejemplo;
27
28  -- Mostrar las bases de datos disponibles
29  • SHOW DATABASES;
30
31  -- Establacer o Seleccionar la base de datos que se va a usar
32  • USE Ejemplo;
33
34  -- DDL: CREATE TABLE - Crea una tabla llamada <<Persona>> con tres campos básicos
35  • CREATE TABLE Persona
36  (
37      id varchar(10),
38      nombre varchar(30),
39      salario float
40  );
```



MySQL es un Sistema de gestión de bases de datos relacional.



- Herramienta visual para manejar la administración de MySQL utilizando un navegador web.

- Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios y exportar datos en varios formatos.



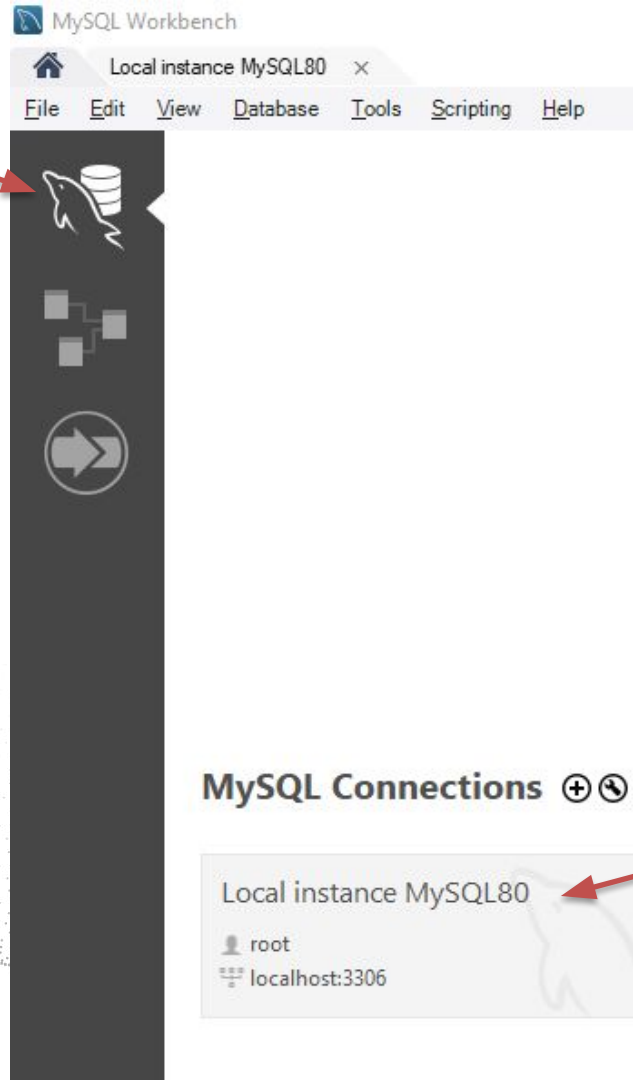
HeidiSQL

Software libre y de código abierto que permite conectarse a MySQL



Es una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL.

Paso 1



Paso 2

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Filter objects

- bd_api
- company
- db_personas
- directorio
- ejemploload
- exampleorders
- festivalbd
- fullstack
- fullstack_2
- hilton
- Tables

Administration Schemas

Information

No object selected

Dept* SQL File 4* SQL File 4* SQL File 5* SQL File 6* Hilton* SQL File 8 x

1

Limit to 1000 rows

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
---	------	--------	---------	------------------

Es necesario verificar que nos encontramos conectados al gestor de base de datos.



Ejecuta todo el script

- bdcolegio
- company
- employees
- empresadb
- libro
- metropolisdb
- mydb
- roaddb
- sept
- sys
- world

No object selected



Limit to 1000 rows

1

Ejecuta una sola línea

Tipo de datos



En la creación de una tabla debe indicarse un tipo de datos para cada uno de los atributos que la componen, en esta sección se analizarán los diferentes tipos de datos soportados por MySQL.

Cada uno de los tipos tiene asociado un espacio de almacenamiento en memoria del cual a su vez depende el rango admisible de valores que puede tomar el dato.

Es responsabilidad del administrador de la base datos hacer un buen diseño de la misma, en particular eligiendo adecuadamente los tipos de datos de cada uno de los atributos de la tabla. Con esta elección debe buscarse la optimización del espacio de memoria y la eficiencia y rapidez en la consulta de los datos.

Los diferentes tipos de datos pueden ser clasificados en varios grupos o categorías: numéricos, cadena de caracteres, fecha y hora, conjuntos y enumeraciones,...

Tipos Enteros

<i>Nombre</i>	<i>Rango</i>	<i>Almacenamiento</i>	<i>Descripción</i>
TINYINT	-128 a 127	1 byte	Entero pequeño
BIT	0 o 1	1 bit	Entero 0 o 1
BOOL	0 o 1	1 bit	Entero 0 o 1
SMALLINT	-32768 a 32767	2 bytes	Entero pequeño
MEDIUMINT	-8388608 a 8388607	3 bytes	Entero mediano
INT	-2147483648 a 2147483647	4 bytes	Entero normal
INTEGER	-2147483648 a 2147483647	4 bytes	Sinónimo de INT
BIGINT	-9223372036854775808 a 9223372036854775807	8 bytes	Entero grande

Tabla 14.2 *Tipos de datos numéricos enteros*

Numéricos




Como puede apreciarse en la Tabla 14.2, existen varias posibilidades para trabajar con tipos enteros, la elección concreta dependerá de lo que realmente se almacenará en ese argumento; por ejemplo, si se desea definir un argumento para guardar edades de personas, no parecería muy lógico utilizar el tipo `INTEGER` por el enorme desperdicio de espacio en memoria que implicaría.

Numéricos



A la hora de definir un argumento de tipo entero se puede utilizar el argumento UNSIGNED (sin signo) para impedir que dicho argumento acepte valores con signo negativo, es decir, solo se podrán utilizar números positivos. El uso de este atributo además permite duplicar el tamaño del número más grande que puede ser almacenado; por ejemplo, el rango de valores del tipo TINYINT comprende todos los números entre -128 y 127, en cambio, el rango de UNSIGNED TINYINT variará entre 0 y 255.

	Valores	Σ
TINYINT	-128 y 127	
Unsigned TINYINT	0 y 255	

Tipos numéricos en coma flotante

Básicamente hay dos tipos numéricos en coma flotante: FLOAT y DOUBLE; la diferencia entre ellos está en el número de bytes que se utilizan para almacenar los números; en el tipo FLOAT se utilizan cuatro bytes y en el DOUBLE el doble, es decir, 8 bytes. Además de estos dos tipos básicos, existen una serie de tipos "sinónimos" que tienen un comportamiento idéntico a ellos.

Numéricos



<i>Nombre</i>	<i>Rango</i>	<i>Almacenamiento</i>	<i>Descripción</i>
FLOAT	$\pm 1.175494351\text{E}-38$ a $\pm 3.402823466\text{E}+38$	4 bytes	Coma flotante de precisión única.
DOUBLE	$\pm 1.797693134862315\text{E}+308$ a $\pm 2.2507385850720\text{E}-308$	8 bytes	Coma flotante de precisión única.
PRECISION REAL			Sinónimos de double
DECIMAL NUMERIC DEC		Un byte para cada dígito del número	Número de coma flotante almacenado como cadena de caracteres

Tabla 14.3 *Tipos de datos numéricos en coma flotante*

Numéricos



A la hora de definir atributos de tipo numérico en coma flotante también es posible especificar el número de lugares decimales y el número de dígitos totales con los que los números serán visualizados. Para hacerlo se deben colocar entre paréntesis a continuación del nombre del tipo, y separados por coma, la longitud total de visualización del número y el número máximo de dígitos decimales.

Numéricos



Si se define el atributo PRECIO de la siguiente manera:

PRECIO FLOAT(5, 2)

Los valores serán visualizados con un máximo de 5 dígitos, dos de ellos correspondientes a la parte decimal. Conviene indicar que se permitiría almacenar números con un dígito más siempre que sean positivos.

Numéricos



Salvo en los tipos BIT y BOOL, en el resto de tipos de enteros se puede especificar entre paréntesis el número máximo de dígitos con el que se mostrará el correspondiente número. El valor máximo de este parámetro es 255.

Relacionado con esto, el argumento ZEROFILL utilizado para los valores numéricos, completa con ceros a la izquierda los valores hasta alcanzar esa longitud máxima. Si se le asigna a un atributo el modificador ZEROFILL automáticamente se le asignará UNSIGNED.

Un tipo de dato entero representa un subconjunto finito de los números enteros.

El número mayor que puede representar depende del tamaño del espacio usado por el dato y la posibilidad (o no) de representar números negativos.

Por ejemplo, si para almacenar un número entero disponemos de 2 bytes de memoria tenemos que:

Ejemplo:

SMALLINT -----> Almacenamiento: 2 bytes

2 Bytes = 2 Bytes x 8 bits = 16 bits

Con 16 bits se pueden representar $2^{16} = 65536$ valores

Solo positivos
0 hasta 65535

Entero con signo (Positivos y negativos)
-32768 hasta 32767

Tipos Enteros

<i>Nombre</i>	<i>Rango</i>	<i>Almacenamiento</i>	<i>Descripción</i>
TINYINT	-128 a 127	1 byte	Entero pequeño
BIT	0 o 1	1 bit	Entero 0 o 1
BOOL	0 o 1	1 bit	Entero 0 o 1
SMALLINT	-32768 a 32767	2 bytes	Entero pequeño
MEDIUMINT	-8388608 a 8388607	3 bytes	Entero mediano
INT	-2147483648 a 2147483647	4 bytes	Entero normal
INTEGER	-2147483648 a 2147483647	4 bytes	Sinónimo de INT
BIGINT	-9223372036854775808 a 9223372036854775807	8 bytes	Entero grande

Tabla 14.2 *Tipos de datos numéricos enteros*

Cadenas



En ambos casos, a la hora de definir un atributo con tipo CHAR o VARCHAR, debe indicarse entre paréntesis la longitud máxima que admitirá. La longitud máxima posible para esta cadena es, en cualquier caso, de 255 caracteres.

La principal ventaja de las cadenas de longitud fija respecto a las de longitud variable es que las primeras se procesan con mayor rapidez; pero en cambio son menos eficientes en el uso de memoria. Por ejemplo, si se almacena una cadena con 3 caracteres en un campo de tipo CHAR(10) se estarían usando 10 bytes (uno por cada carácter de la cadena más los 7 espacios en blanco que se añadirían) para guardar ese dato. En cambio, esa misma cadena en un tipo de dato VARCHAR(10) requeriría 4 bytes (3 para los caracteres y un byte adicional que sirve para identificar el fin de la cadena).

MySQL no admite en la construcción de una tabla mezclar el uso de cadenas de longitud fija y variable. Cuando el usuario defina una tabla en la que aparezca, simultáneamente tipos CHAR y VARCHAR, automáticamente todos serán convertidos a tipo VARCHAR. La única excepción a esta regla se produce cuando se definen tipos VARCHAR de 4 o menos caracteres, en este caso serán convertidas a CHAR.

Cadenas



<i>Nombre</i>	<i>Longitud máxima de caracteres</i>	<i>Almacenamiento</i>	<i>Descripción</i>
CHAR	1	1 bytes	Un carácter.
CHAR (N)	255	N bytes	Cadena de longitud fija de N caracteres.
VARCHAR (N)	255	N+1 bytes	Cadena de longitud variable de N caracteres.
TINYTEXT TINYBLOB	255	L+1 bytes	Cadenas u objetos binarios de tamaño pequeño
TEXT BLOB	65.535	L+2 bytes	Cadenas u objetos binarios de tamaño normal
MEDIUMTEXT MEDIUMBLOB	16.777.215 (1,6 MB)	L+3 bytes	Cadenas u objetos binarios de tamaño mediano
LONGTEXT LONGBLOB	4.294.967.295 (4,2 GB)	L+4 bytes	Cadenas u objetos binarios de tamaño grande

Tabla 14.4 *Tipos de cadenas de caracteres*

Fecha / Hora



MySQL no comprueba si una fecha es válida en el momento de la inserción de los datos, solo hace dos simples comprobaciones: que el mes esté comprendido entre 0 y 12 y el día esté comprendido entre 0 y 31. Esto hace que MySQL considere como fecha válida, por ejemplo, '2005-2-31' (31 de febrero de 2005).

Fecha / Hora



<i>Nombre</i>	<i>Rango</i>	<i>Tamaño de almacenamiento</i>	<i>Descripción</i>
DATE	1000-01-01 9999-12-31	3 bytes	Almacena una fecha con el formato YYYY-MM-DD
TIME	-838:59:59 838:59:59	3 bytes	Almacena una hora con el formato HH:MM:SS
DATETIME	1000-01-01 00:00:00 9999-12-31 23:59:59	8 bytes	Almacena una fecha y una hora con el formato YYYY-MM-DD HH:MM:SS
TIMESTAMP TIMESTAMP (M)	1970-01-01 00:00:00 2037-01-01 00:00:00	4 bytes	Instante de tiempo, combinación de fecha y hora según diferentes formatos
YEAR (2)	70-69 (de 1970 a 2069)	1 byte	Almacena un año con dos dígitos (p.e. 97 corresponde a 1997)
YEAR YEAR (4)	1901 - 2155	1 byte	Almacena un año con cuatro dígitos

Tabla 14.5 *Tipos de dato fecha-hora*

<i>Tipo especificado</i>	<i>Formato</i>
TIMESTAMP	AAAAMMDDHHMMSS
TIMESTAMP (14)	AAAAMMDDHHMMSS
TIMESTAMP (12)	AAMMDDHHMMSS
TIMESTAMP (8)	AAAAMMDD
TIMESTAMP (6)	AAMMDD
TIMESTAMP (4)	AAMM
TIMESTAMP (2)	AA

Tabla 14.6 *Formato de tipos TIMESTAMP*

NORMALIZACIÓN



- Proceso
- Implica: aplicar reglas -----> Reglas de normalización
- Obtener un “buen” modelo E/R
- Las bases de datos relacionales se normalizan para:
 - * Evitar la redundancia de los datos.
 - * Evitar problemas de actualización de los datos en las tablas.
 - * Proteger la integridad de los datos -----> *Restricciones*
 - * Tener almacenamiento con un menor espacio posible
 - * Eliminar errores lógicos
 - * Tener nuestros datos ordenados

- Formas normales:

- 1FN - Codd definió primera forma normal

- 2FN

- 3FN

- FNBC** - Forma Normal de Boyce-Codd

- 4FN

- 5FN

NORMALIZACIÓN



Hay que tener en cuenta que muchos diseñadores opinan que basta con llegar a la forma Boyce-Codd, ya que la cuarta, y sobre todo la quinta forma normal es polémica. Hay quien opina que hay bases de datos peores en quinta forma normal que en tercera. En cualquier caso debería ser obligatorio para cualquier diseñador llegar hasta la forma normal de Boyce-Codd.

LENGUAJES DE DEFINICIÓN, MANIPULACIÓN Y CONTROL



6.1 EL LENGUAJE SQL

SQL es un lenguaje estándar completo de control e interacción con un sistema de gestión de base de datos relacional definido por la *American National Standards Institute* (ANSI) y la *International Standards Organization* (ISO). Se trata de un lenguaje para administrar, almacenar y recuperar información utilizado por la mayoría de los sistemas gestores de bases de datos actuales. Habitualmente incluye las siguientes categorías de sentencias:

- **DDL** (*Data Definition Language* o Lenguaje de Definición de Datos)
Utilizadas para crear, alterar o borrar objetos de base de datos, tales como tablas, columnas y vistas. Como ejemplos característicos tenemos las instrucciones o comandos CREATE, ALTER y DROP.
- **DML** (*Data Modification Language* o Lenguaje de Modificación de Datos)
Utilizadas en la interrogación y manipulación de datos en esquemas ya existentes. Como ejemplos característicos tenemos los comandos SELECT, INSERT, UPDATE y DELETE.
- **TCL** (*Transaction Control Language* o Lenguaje de Control de Transacciones)
Utilizadas para confirmar o restaurar transacciones de base de datos, que son unidades de trabajo que realizan una o más sentencias SQL relacionadas entre sí. Como ejemplos característicos tenemos los comandos COMMIT y ROLLBACK.
- **DCL** (*Data Control Language* o Lenguaje de Control de Datos)
Utilizadas en el control de acceso a datos en la base de datos. Como ejemplos característicos tenemos los comandos GRANT y REVOKE.

Crear bases de datos



Create database Ejemplo;



Ejemplo

Importante:

- ✓ Todas las instrucciones terminan en punto y coma (;)
- ✓ En Windows, las instrucciones no son case sensitive.

Por ejemplo Windows interpreta igualmente las siguientes sentencias:

```
create database Ejemplo;  
Create DataBase Ejemplo;
```

Conocer bases de datos

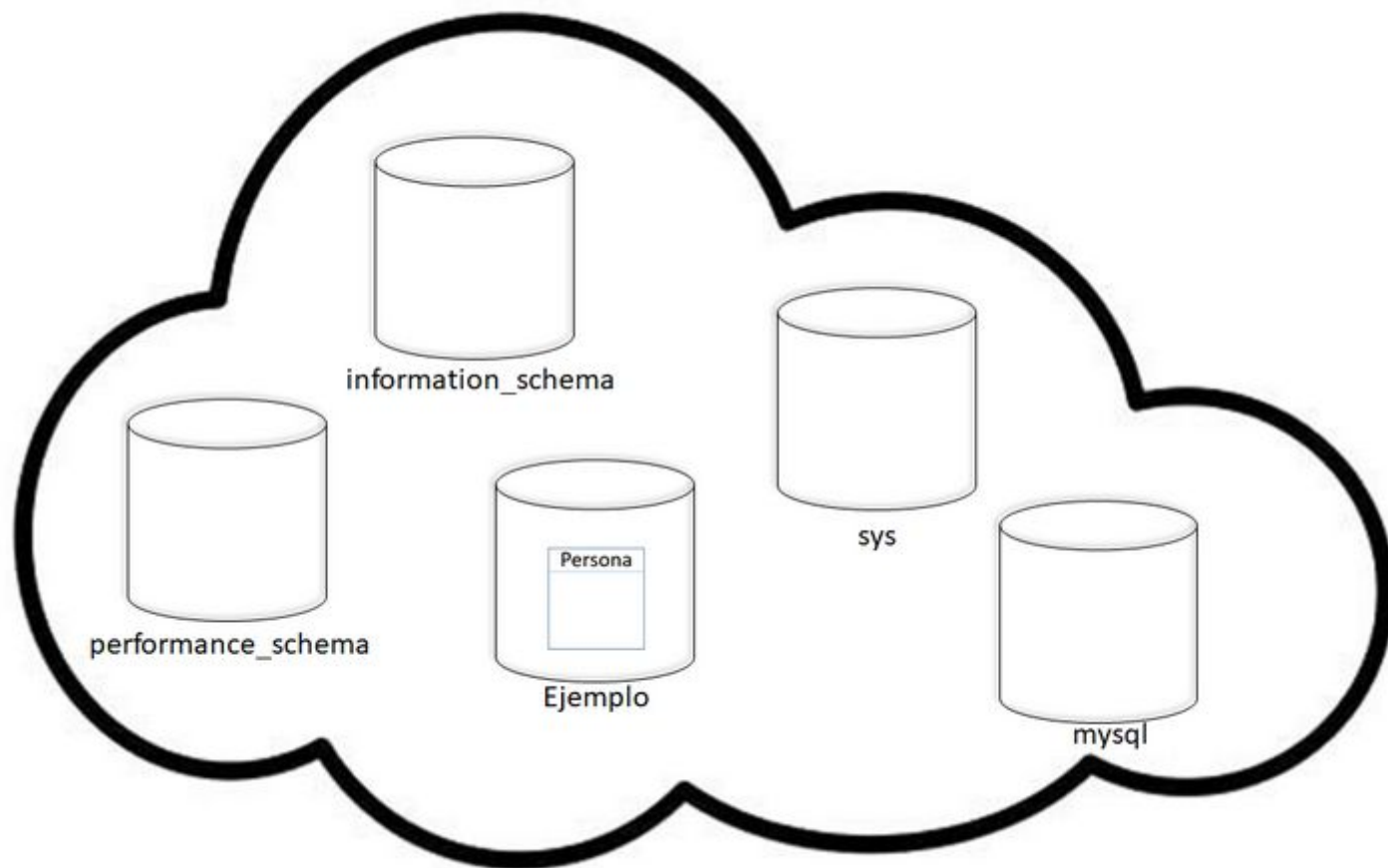


Show databases;

The screenshot shows a Windows command prompt window titled "c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe". The text inside the window is as follows:

```
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> create database Ejemplo;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| ejemplo |  
| mysql |  
| performance_schema |  
| test |  
+-----+  
5 rows in set (0.00 sec)  
  
mysql> _
```

An orange arrow points from the right side of the image to the "ejemplo" database name in the output table.





SGBD

3

EmpresaDB

TurismoApiaDB

4

ActivosInformacionDB

2

ColegioDB

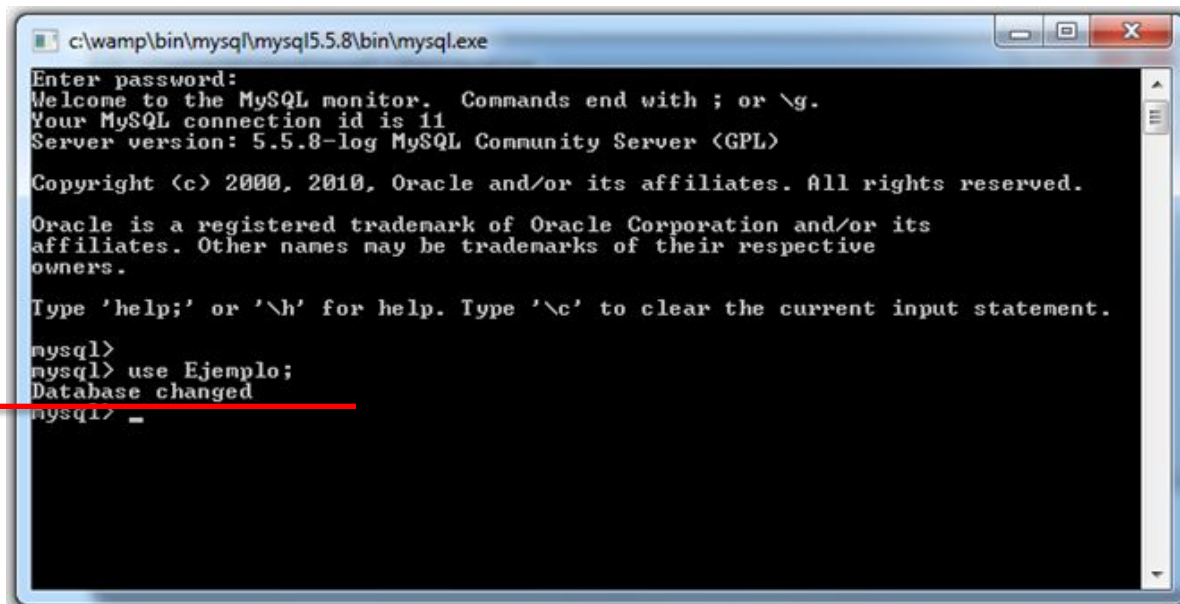
1

USAR BASES DE DATOS

(Establecer Como Bases De Datos Principal)



Use Ejemplo;

A screenshot of a Windows command prompt window titled "c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe". The window shows the MySQL command-line interface. The text displayed is: "Enter password:", "Welcome to the MySQL monitor. Commands end with ; or \g.", "Your MySQL connection id is 11", "Server version: 5.5.8-log MySQL Community Server (GPL)", "Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.", "Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.", "Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.", "mysql>", "mysql> use Ejemplo;", "Database changed", "mysql> _". A red horizontal line is drawn under the "mysql>" prompt on the line "mysql> _".

```
c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.5.8-log MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
mysql> use Ejemplo;
Database changed
mysql> _
```

CREAR TABLAS



Forma 1:

```
Create table Persona
```

```
(
```

```
    id varchar(10),
```

```
    nombre varchar(30),
```

```
    salario float
```

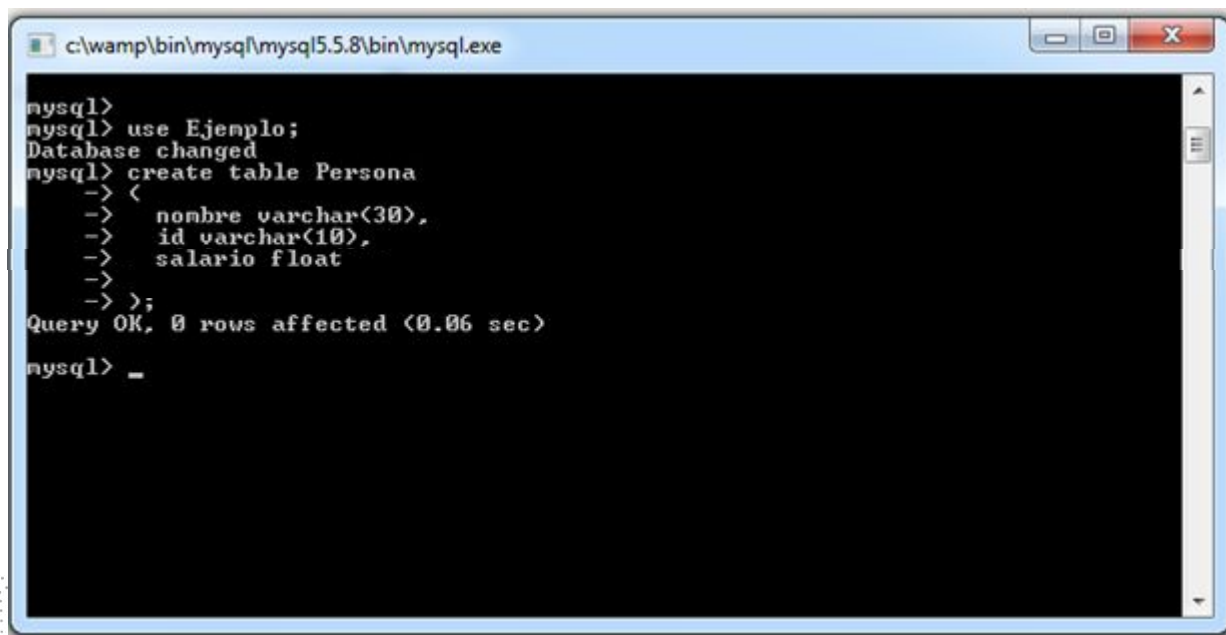
```
);
```



Ejemplo

Forma 2:

```
Create table Persona( id varchar(10), nombre varchar(30), salario float );
```


A screenshot of a MySQL command prompt window. The title bar shows the file path "c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe". The window has a black background with white text. The text shows the execution of SQL commands to use a database and create a table.

```
c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe

mysql>
mysql> use Ejemplo;
Database changed
mysql> create table Persona
-> (
->   nombre varchar(30),
->   id varchar(10),
->   salario float
-> );
Query OK, 0 rows affected (0.06 sec)

mysql> _
```

Ver tablas de una base de datos



Show tables;

A screenshot of a Windows command prompt window titled "c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe". The window shows the MySQL command-line interface. The user has entered the command "show tables;" after selecting the "Ejemplo" database. The output shows a single table named "persona". A red horizontal line is drawn across the "mysql> show tables;" command line.

```
c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.5.8-log MySQL Community Server <GPL>

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use Ejemplo;
Database changed
mysql> show tables;
+-----+
| Tables_in_ejemplo |
+-----+
| persona            |
+-----+
1 row in set (0.00 sec)

mysql>
```

Describir estructura de la tabla



Describe Persona;

```
c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe
Your MySQL connection id is 14
Server version: 5.5.8-log MySQL Community Server (GPL)
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> use ejemplo;
Database changed
mysql> describe Persona;
```

Field	Type	Null	Key	Default	Extra
nombre	varchar(30)	YES		NULL	
id	varchar(10)	YES		NULL	
salario	float	YES		NULL	

```
4 rows in set (0.01 sec)
mysql>
```

LENGUAJES DE DEFINICIÓN, MANIPULACIÓN Y CONTROL



6.1 EL LENGUAJE SQL

SQL es un lenguaje estándar completo de control e interacción con un sistema de gestión de base de datos relacional definido por la *American National Standards Institute* (ANSI) y la *International Standards Organization* (ISO). Se trata de un lenguaje para administrar, almacenar y recuperar información utilizado por la mayoría de los sistemas gestores de bases de datos actuales. Habitualmente incluye las siguientes categorías de sentencias:

- **DDL** (*Data Definition Language* o Lenguaje de Definición de Datos)
Utilizadas para crear, alterar o borrar objetos de base de datos, tales como tablas, columnas y vistas. Como ejemplos característicos tenemos las instrucciones o comandos CREATE, ALTER y DROP.
- **DML** (*Data Modification Language* o Lenguaje de Modificación de Datos)
Utilizadas en la interrogación y manipulación de datos en esquemas ya existentes. Como ejemplos característicos tenemos los comandos SELECT, INSERT, UPDATE y DELETE.
- **TCL** (*Transaction Control Language* o Lenguaje de Control de Transacciones)
Utilizadas para confirmar o restaurar transacciones de base de datos, que son unidades de trabajo que realizan una o más sentencias SQL relacionadas entre sí. Como ejemplos característicos tenemos los comandos COMMIT y ROLLBACK.
- **DCL** (*Data Control Language* o Lenguaje de Control de Datos)
Utilizadas en el control de acceso a datos en la base de datos. Como ejemplos característicos tenemos los comandos GRANT y REVOKE.

Instrucción SQL: Insert Into



La instrucción INSERT INTO se utiliza para insertar nuevos registros en una tabla.

Esta es la estructura de la instrucción:


```
INSERT INTO Tabla (campo1, campo2, campo3)  
VALUES (valor1, valor2, valor3);
```

- **Símbolo:** *
- **Descripción:** Recupera todos los registros de la tabla

```
Select * From Persona;
```

Eliminar todos los registros



 c:\wamp64\bin\mysql\mysql8.0.21\bin\mysql.exe

```
mysql> TRUNCATE TABLE persona;  
Query OK, 0 rows affected (0.02 sec)
```

Eliminar una tabla



DROP DATABASE ejemplo;



c:\wamp64\bin\mysql\mysql8.0.21\bin\mysql.exe

```
mysql> DROP TABLE Persona;  
Query OK, 0 rows affected (0.01 sec)
```


Actividad Práctica



1. Cree la base de datos ***Ejemplo***
2. Use a base de datos
3. Cree la tabla persona: código, nombre y fecha de nacimiento. ***Llave primaria será el código.***
4. Describa la estructura de la tabla.
5. Imprima la lista de tablas que tiene la BD
6. Inserte los datos de quince personas (registros).
7. Imprima la lista de las personas registradas

Entregables:

- Modelo relacional
- Script de la base de datos (Archivo ***.sql***)

Consultar

- Llave primaria
- Insertar fechas
- Auto incremental

Query 1 

1

Ejecuta una sola línea

No object selected

←→↻sqliteonline.com

📄☆📄📄PDF📄📄📄

PricingHelp🗨️+

☰⬆️ Import⬇️ Export

✕

+ → "Add DataBase"

Create a database linked to your account.

This is only available with a paid subscription.

📁 SQLite

📁 0.1.4 beta (Mem...)

Table

demo

Persona

MariaDB

PostgreSQL

MS SQL

▶ Run🏠 SQLite

📄🔗👤⚙️

```
1
2 CREATE TABLE Persona (
3     id INT AUTO_INCREMENT PRIMARY KEY,
4     nombre VARCHAR(30),
5     salario FLOAT
6 );
7
8
9 INSERT INTO Persona(id,nombre,salario)VALUES(1,'Pepe',1000);
10 INSERT INTO Persona(id,nombre,salario)VALUES(2,'María',2500);
11
12 SELECT * FROM persona;
13
```

id	nombre	salario
1	Pepe	1000
2	María	2500

⬆️📄📊📄☰

Syntax

History

Download remote DB (only SQLite):
[https://sqliteonline.com/#urldb=\(url\)](https://sqliteonline.com/#urldb=(url))

Example: Chinook | NorthWind | cnf.db | BasketBall | Sakila | Multiple DB (Chinook+NorthWind)

Download remote SQL Script:
[https://sqliteonline.com/#urlsql=\(url\)](https://sqliteonline.com/#urlsql=(url))

Example: Chinook SQL

all functions

comment

ALTER TABLE

ANALYZE

ATTACH DATABASE

<https://sqliteonline.com/>

```
CREATE TABLE Persona  
(  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(30),  
    salario FLOAT  
);
```

```
INSERT INTO Persona(id,nombre,salario)VALUES(1,'Pepe',1000);  
INSERT INTO Persona(id,nombre,salario)VALUES(2,'María',2500);
```

```
SELECT * from persona;
```



GRACIAS

Línea de atención al ciudadano: 018000 910270
Línea de atención al empresario: 018000 910682



@SENAcomunica

www.sena.edu.co