ALUMNO: Cuesta Alario David **DNI: 20857516-N**

MySQL – Comandos en Xampp

Trabajar con Xampp

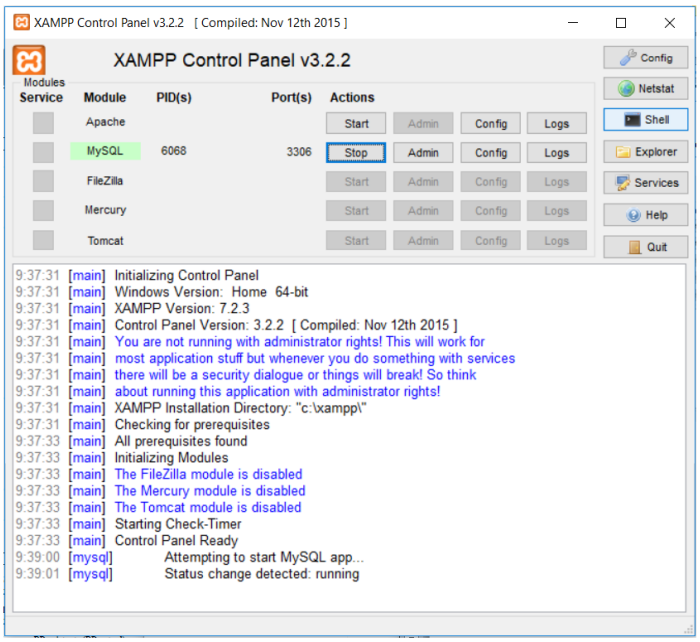
Una vez arrancado el servidor de MySQL con la acción Start se abre la ventana de comandos Shell y se introducen los siguientes comandos que permiten administrar las bases de datos como Root:

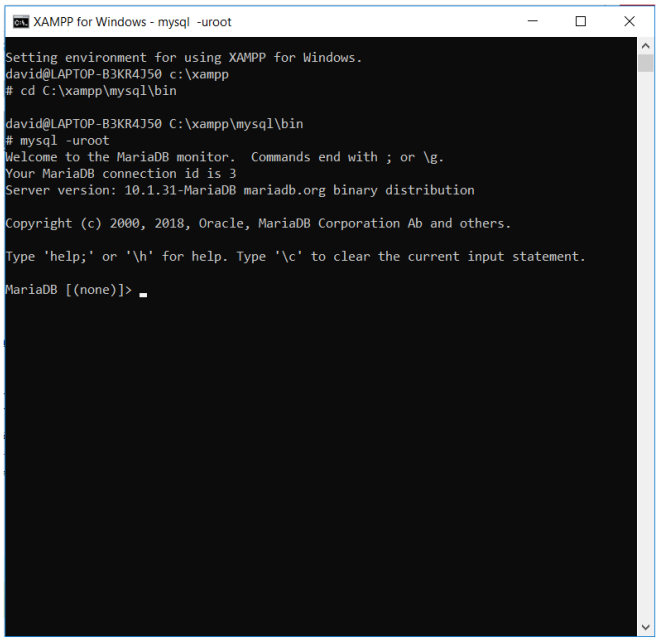
CD <dirección> **mysql –h**<host> **-u**<usuario> **-p**<contraseña>

- Donde la opción:
  - <dirección> representa el directorio en el que está instalado MySQL. Por defecto será → C:\xampp\mysql\bin
  - **–h**<host> corresponde al host (este campo no hace falta si estamos en la misma máquina que el servidor)
  - **-u**<usuario> indica el nombre del usuario que va a entrar
  - **-p**<contraseña> indica su contraseña

MySQL –uroot

Cuando se entra en MySQL, sale un mensaje de ayuda y el prompt de la línea de comandos cambia a **mysql >**





Para detener el servidor de MySQL desde el panel de XAMPP haremos clic en la acción stop del servidor MySQL

Cerrar la consola de MySQL:

mysql > QUIT;

Los siguientes comandos permiten acceder al manual de MySQL que incluye Ayuda para algunos comandos propios de:

mysql > HELP; los sitios locales y web

mysql > HELP CONTENTS; la sintaxis y funcionamiento de las instrucciones

Los siguientes comandos permiten moverse entre las diferentes tablas y bases de datos implementadas

mysql > STATUS;	Ver qué usuario somos y qué base de datos estamos usando
mysql > USE < Nombre Base Datos >**;**	Usar una base de datos existente
mysql > SHOW DATABASES;	Mostrar todas las bases de datos existentes
mysql > SHOW TABLES;	Mostrar las tablas de la base de datos actual
mysql > SHOW CREATE TABLE < Nombre Tabla >**;**	Mostrar la instrucción de creación de una tabla
mysql > DESCRIBE < Nombre Tabla >**;**	Mostrar los campos de una tabla

Grabar la sesión en un fichero

mysql> TEE < Nombre-Ruta Fichero.txt >

mysql> NOTEE (Si después ponemos simplemente TEE sigue añadiendo al fichero anterior)

Cambiar los permisos de root a permitido

Acedemos a la carpeta C:\xampp\apache\conf\extra

Abrimos el archivo httpd-xampp.conf

Y sustituimos el texto:

```
#
# New XAMPP security concept
#

Order deny,allow
Deny from all
Allow from ::1 127.0.0.0/8
```

```
#
# New XAMPP security concept
#
Order deny,allow
#Deny from all
Allow from all
```

Sintaxis de MySQL

Los comandos del estándar SQL siguen la sintaxis del estándar SQL92

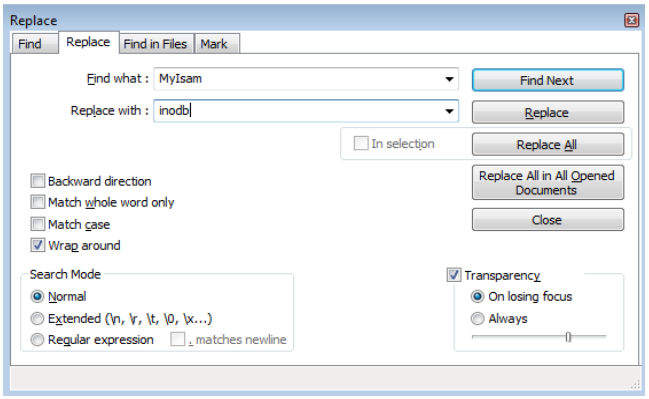
- Todas las instrucciones acaban con punto y coma ;
- A la hora de indicar la sintaxis de los distintos comandos de SQL se sigue la notación conocida como BNF:

SIMBOLO	SIGNIFICADO
MAYÚSCULAS	Palabra reservada de SQL
minúsculas	Nombre de un archivo, tabla, campo, etc…
Corchetes [ ]	El elemento que esta entre corchetes es opcional, apareciendo de 0 a 1 veces
Llaves { }	Se debe elegir una de las opciones que aparezcan entre llaves
Barra vertical |	OR lógico que separa las distintas opciones
Puntos suspensivos …	Repetición de lo que aparezca justo antes de los puntos suspensivos

Comentarios en los comandos

- Dos guiones al principio de la línea: **mysql> --** Consulta 1
- También se admiten los comentarios al estilo de C y Java en cualquier parte de las instrucciones: /* …. */

Ejecutar scripts



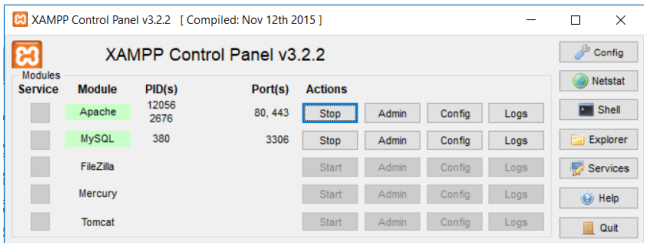
Los ficheros de script son ficheros de texto que contienen instrucciones en SQL. Para importarlos se debe sustituir las apariciones de la palabra MyIsam por inobd y ejecutar el siguiente comando:

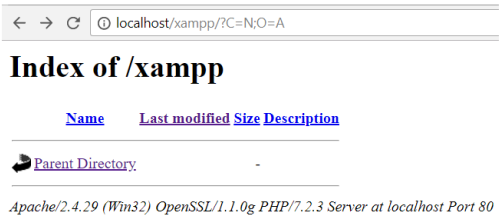
mysql > SOURCE <directorio>****<nombre script.sql>

Crear una copia de seguridad

Se crea un script con todo lo referente a la BD – tablas y datos):

- **Opción 1:** descargar una copia
  Arrancar Apache desde XAMPP
  Poner en el navegador la URL: localhost/Xampp
  Parent Directory **-->** phpMyAdmin **-->** seleccionar la BD **-->** Exportar
- **Opción 2:** Desde el directorio de instalación de MySQL
  Desde la línea de comandos (fuera de mysql pero en el mismo directorio c:\xampp\mysql\bin) escribir:
  mysqldump –uroot < Nombre base de datos > **>** < Nombre Script.sql >**;**





Importar datos desde un fichero de texto

LOAD DATA INFILE < Nombre Fichero.txt >

INTO TABLE < Nombre Tabla > **(** < Nombre Campo 1 > **,** < Nombre Campo i > **);**

- El fichero de texto a importar ha de estar en el directorio de esa base de datos, por defecto:
  C:\dir_mysql\DATA\carpeta_con_el_nombre_de_la_BD\
- Si el fichero de texto a importar está en otro directorio, hemos de poner su ruta completa al estilo UNIX

Creación de una base de datos

CREATE DATABASE IF NOT EXISTS < Nombre Base Datos >

CREATE TABLE IF NOT EXISTS < Nombre Tabla > **(**

< Nombre Campo 1 > < Tipo Dato 1 > < Opciones 1 >, (Debe existir un mínimo de dos campos)

< Nombre Campo 2 > < Tipo Dato 2 > < Opciones 2 >,

< Nombre Campo N > < Tipo Dato N > < Opciones N >,

PRYMARY KEY < Campo Clave Primaria >

- Debe ser un campo de la tabla, Debe existir al menos una y ser única, pueden haber varias

UNIQUE < Campos Clave Alternativa >

- Deben ser campos de la tabla, Puede haber entre ninguna hasta varias
- Contendrá los campos que sean únicos pero no hayan sido escogidos como clave principal

FOREIGN KEY < Campos Clave Externa > **REFERENCES** < Nombre Tablas Referenciadas >

- Deben ser campos de la tabla, Puede haber entre ninguna hasta varias)
- Contendrá los campos de la tabla en el mismo orden que las tablas con las que se relacionen

ON UPDATE < Opciones Actualizar >

ON DELETE < Opciones Borrar > **);**

Tipos de datos:

Numéricos	**Enteros**	**INTEGER**	**INT**	
	Reales	**DECIMAL(i , j)**	**DEC(i , j)**	**NUMERIC(i , j)**
		i representa la cantidad total de dígitos enteros j representa la cantidad de dígitos decimales		
		DOUBLE PRECISION	**FLOAT**	**REAL**
Cadena de caracteres	**Longitud fija:**	**CHAR(n)**	**CHARACTER(n)**	
		n representa el número de caracteres de la cadena		
	Longitud variable	**VARCHAR(n)**	**CHARACTER VARYING(n)**	**CHAR VARYING(n)**
		n representa el número máximo de caracteres definidos		
Cadena de Bits	**Longitud fija:**	**BIT(n)**		
		n representa el número de caracteres de la cadena		
	Longitud variable	**BIT VARYING(n)**		
		n representa el número máximo de caracteres definidos		
Fecha Hora	**Fecha y hora**	**TIMEDATE**	YYYY-MM-DD HH:MM:SS	
	Sólo fecha	**DATE**	YYYY-MM-DD	
	Sólo hora	**TIME**	HH:MM:SS	

Un dominio permite crear un nuevo tipo de dato personalizado a partir de los que ya están definidos en el estándar SQL:
CREATE DOMAIN < Nombre nuevo Tipo > **AS** < Tipo Dato Referencia>

Opciones para los datos:

El estándar SQL permite especificar algunas opciones dentro de cada campo de la tabla.

DEFAULT permite inicializar automáticamente los valores del campo al valor por defecto para ese tipo de dato

NOT NULL permite indicar que un campo de la tabla nunca podrá estar vacío.
Esta restricción la han de cumplir obligatoriamente todos los campos que formen parte de la clave primaria.

Opciones para las restricciones de integridad:

El estándar SQL permite establecer como afectara al resto de la base de datos el eliminar o actualizar un campo

SET NULL dejar el campo vacío

SET DEFAULT dejar el valor por defecto en función del tipo de dato

CASCADE modifica todos los campos relacionados al modificarse la tabla con este parámetro

RESTRICT sólo se borra si no hay restricciones ni vistas que le hagan referencia.

Actualización de una base de datos

Permite cambiar el diseño de la tabla que se indique: tanto el diseño de sus atributos como de sus restricciones de integridad

ALTER TABLE < Nombre Base Datos > **.** < Nombre Tabla >

ALTER < Nombre Campo o Código Restricción > (permite cambiar algo que ya existe)

DROP < Código opción >**;** (Para eliminar una opción de un campo)

SET < Código opción >**;** (Para añadir una opción a un campo)

ADD < Nombre Campo o Código Restricción >**;** (permite añadir algo nuevo al diseño)

Después de ADD se pone lo mismo que se indicaría en la instrucción CREATE TABLE

DROP < Nombre Campo o Código Restricción >**;** (permite quitar algo del diseño)

Para borrar una tabla entera de una base de datos se emplea:

DROP TABLE < Nombre Tabla >

CASCADE modifica todos los campos relacionados al modificarse la tabla con este parámetro

RESTRICT sólo se borra si no hay restricciones ni vistas que le hagan referencia.

Para borrar una base de datos entera se emplea:

DROP DATABASE < Nombre Base Datos >

CASCADE modifica todos los campos relacionados al modificarse la tabla con este parámetro

RESTRICT sólo se borra si no hay restricciones ni vistas que le hagan referencia.

Definición de datos

Insertar nuevas filas de datos en los campos de una tabla

INSERT INTO < Nombre Tabla > **VALUES** < Valor Campo 1 >**,** < Valor Campo 2 >**,** < Valor Campo N >**;**

- Los campos deben estar en el mismo orden en el que se especificaron los atributos al crear la tabla
- Las cadenas de caracteres van entre comillas simples

INSERT INTO < Nombre Tabla > **(** < Nombre Campo i > **)** **VALUES** < Valor Campo i >**;**

- Permite añadir solo los campos indicados
- El orden de los campos de las instancias INSERT y VALUES debe coincidir
- Los campos con la restricción NOT NULL no pueden omitirse
- Las cadenas de caracteres van entre comillas simples

Modificar datos de una tabla

Sustituirá el valor indicado de los campos descritos por la cláusula SET de aquellos campos que cumplan las condiciones indicadas por la cláusula WHERE

UPDATE < Nombre Tabla >

SET < Nombre Campo i > **=** < Nuevo Valor Campo i >

- Permite seleccionar los campos que se deben sustituir, así como su nuevo valor
- Admite múltiples cambios simultáneos separados por comas
- Otra posibilidad es efectuar una operación sobre un determinado campo

  SET < Nombre Campo i > **=** < Nombre Campo i > < Operación a realizar >

WHERE < Nombre Campo > <OPERADOR>**;**

Borrar filas enteras de la tabla

DELETE FROM < Nombre Tabla > **WHERE** < Nombre Campo i > { = != , > , < } < Valor Campo i >**;**

- Sin la cláusula WHERE se borraran todas las tuplas de la tabla

Consultas de datos

CREATE VIEW < Nombre Vista > **AS SELECT** <SUBCLAUSULA> < Nombre Campo 1 > **,** < Nombre Campo i >

- Permite seleccionar los campos los datos que se quieren recuperar separados por comas
- Los resultados se mostraran como una tabla en el mismo orden
- Admite cualquier tipo de atributos, fórmulas, funciones agregadas, concatenación de cadenas de caracteres con "||"
- Tiene algunas subclausulas opcionales:

  DISTINCT permite que se eliminen los datos duplicados

FROM < Nombre Tabla 1 >

- Permite indicar las tablas de las que se deben obtener los campos seleccionados
- Las tuplas con valores nulos en el atributo de reunión no se incluyen en el resultado, salvo si es una reunión externa
- Se puede anidar más de una tabla mediante paréntesis sucesivos empleando las cláusulas de reunión

  FROM < Nombre Tabla 1 > **INNER JOIN** < Nombre Tabla 2 > **ON** < Nombre Campo 1 > **=** < Nombre Campo 2 >
  - Las reuniones internas permiten anidar dos tablas cuando tienen un campo común con un nombre distinto
  - Se ha de indicar cuales son los campos relacionados

  FROM < Nombre Tabla 1 > **NATURAL JOIN** < Nombre Tabla 2 >
  - Las reuniones naturales permiten anidar dos tablas cuando tienen un campo común con el mismo nombre
  - Se asume que los campos con el mismo nombre son los que están relacionados

  FROM <Nombre Tabla 1> <OPERADOR> **OUTER JOIN** <Nombre Tabla 2> **ON** < Nombre Campo 1 > **=** < Nombre Campo 2 >
  - Las reuniones externas permiten representar los campos de las tablas aunque no cumplan la cláusula de reunión
  - Aquellos campos que no se puedan emparejar aparecerán representados con el valor NULL
  - Podemos encontrar tres tipos de reuniones externas identificadas con los operadores **LEFT** , **RIGHT** y **FULL**

WHERE < Nombre Campo > <OPERADOR>

- Permite seleccionar de entre los campos indicados únicamente aquellos datos que cumplan las condiciones descritas
- Admite los siguientes operadores:
  - De comparación { **=** **!=** , **>** , **<** } < Valor Campo >
  - Lógicos **AND** , **OR** o **NOT** para separar varias condiciones
  - Comparación de subcadenas con **LIKE**
    - **_** sustituye a un carácter arbitrario
    - **%** a un número indeterminado de caracteres arbitrarios
    - Las cadenas de caracteres van entre comillas simples
  - Conjuntos de valores

    IN (< Rango de Valores o clausula SELECT >**)**

    NOT IN (<Rango de Valores o clausula SELECT >**)**

    BETWEEN <Numero entero> **AND** < Numero entero >
  - las comparaciones de igualdad no funciona sobre un campo vacío por lo que se emplean los operadores:

    IS NULL y **IS NOT NULL**
- La omisión de WHERE indica una selección de tuplas incondicional en la que se cogen todas las filas

GROUP BY < Nombre Campo para agrupación de las funciones agregadas >

- En esta cláusula deben aparecer todos los atributos que aparezcan en SELECT
- En esta cláusula pueden aparecer otros atributos de la tabla que no aparezcan en SELECT

HAVING < condiciones sobre resultados asociados a GROUP BY >

ORDER BY < Nombre Campo > <SUBCLAUSULA>

- Permite ordenar los datos en función del campo especificado de manera **ASC** Ascendente o **DESC** Descendente
- Admite varios campos separados por comas

Funciones agregadas

COUNT (<SUBCLAUSULA> < Nombre Campo >**)** Cuenta el número filas no vacías (aunque se repitan)

- Permite poner ***** como campo para contar todas las filas de la tabla
- Admite de forma opcional la subclausula: **DISTINCT** para contar solo los valores distintos

SUM (< Nombre Campo >**)** Devuelve la suma de los datos de esa columna

AVG (< Nombre Campo >**)** Devuelve la media de esa columna

MAX (< Nombre Campo >**)** Devuelve el mayor valor de esa columna

MIN (< Nombre Campo >**)** Devuelve el menor valor de esa columna

Eliminar consultas

DROP VIEW < Nombre Vista >**;**

Otros conceptos

Calificación de atributos

Cuando coinciden los nombres de los atributos en varias tablas ha de indicarse de que tabla proviene cada campo mediante:

< Nombre Tabla > **.** < Nombre Campo >

El * selecciona todos los atributos de las relaciones

La cláusula **AS** que permite asociar dos campos consiguiendo:

- Hacer varias referencias a una misma relación mediante la asignación de un seudónimo
- Cambiar la representación de un campo en la tabla de resultados

```
SELECT E.NOMBRE AS NOMBRE_EMPLEADO,
       E.APELLIDO AS APEL_EMPLEADO,
       S.NOMBRE AS NOMBRE_SUPERVISOR,
       S.APELLIDO AS APEL_SUPERVISOR
FROM EMPLEADO AS E, EMPLEADO AS S
WHERE E.NSS SUPERV = S.NSS;
```

Funciones y Procedimientos

MySQL permite almacenan rutinas y programas del usuario para la base de datos en las que además de las instrucciones SQL, se pueden declarar variables, instrucciones de control, etc…

Funcionan de forma similar a cualquier lenguaje de programación.

Sintaxis de creación: **CREATE** { **FUNCTION** | **PROCEDURE** } < Nombre > **(**< Argumentos >**)** < Cuerpo >

Sintaxis de Llamada: **CALL** < Nombre > **(**< Argumentos >**)**

Triggers

Son procedimientos que se ejecuta automáticamente cuando se intentan insertar, modificar o borrar datos de una tabla.

Sintaxis de creación: **CREATE TRIGGER**

Sintaxis de borrado: **DROP TRIGGER**

Funciones incorporadas

Son funciones básicas adicionales que se ponen a disposición del usuario:

- Funciones matemáticas
- Funciones de fecha/hora
- Funciones de conversión entre tipos de datos
- Funciones avanzadas

Se pueden utilizar tanto en las instrucciones SQL como dentro del entorno de programación

Funciones sobre la fecha y hora

CURRENT_DATE devuelve la fecha actual en formato YYYY-MM-DD

CURRENT_TIME devuelve la hora actual en formato HH:MM:SS

NOW() devuelve la fecha y hora actual en formato YYYY-MM-DD HH:MM:SS

Extracción de partes de una fecha o hora:

YEAR (< Fecha >**)** devuelve el año de la fecha mediante un número entero

MONTH (< Fecha >**)** devuelve el mes de la fecha mediante un número entero

DAY (< Fecha >**)** devuelve el día de la fecha mediante un número entero

WEEKDAY (< Fecha >**)** devuelve el día de la semana de la fecha mediante un número entero del 1 al 7

DAYNAME (< Fecha >**)** devuelve el día de la semana de la fecha en texto

HOUR (< Hora >**)**

MINUTE (< Hora >**)**

SECOND (< Hora >**)**

Operaciones con fechas y horas:

DATE_ADD (< Fecha >**, INTERVAL (**< incremento > < tipo >**)**

- El incremento puede ser + para añadir ó – para quitar
- El tipo puede ser YEAR, MONTH, DAY, HOUR, MINUTE ó SECOND.

DATEDIFF (< Fecha más Moderna >**,** < Fecha más Antigua >**)**

- Días transcurridos entre dos fechas