# BASES DE DATOS II

## PRACTICA 2

INDICE:

* 1.- Crear una base de datos “prueba”.
* 2.- Crear una tabla con cada motor de almacenamiento indicado posteriormente:
  + InnoDB
  + MyISAM.
  + MEMORY
  + CSV
  + BLACKHOLE
  + Archive
* 3.- Insertar registros en cada tabla
* 4.- Consultar los registros insertados
* 5.- Observar y analizar los ficheros de datos ‘físicos’ creados por MySQL.

**1.- Crear una base de datos “prueba”.**

Primero entramos en la base de datos y accedemos a ella. Para ello ejecutamos el comando: *“mysql -u root -p”* en la dirección en la que se encuentre la base de datos.

Para crear una nueva base de datos ejecutamos el comando: “*CREATE DATABASE prueba;”*

Con el objetivo de comprobar que se ha creado la base de datos ejecutamos el comando “*SHOW DATABASES;”*

Finalmente para añadir tablas a esta base de datos ejecutamos el comando: “*USE prueba;”* y escribimos los métodos de creación de tablas pero diferenciándolas entre ellas con el motor con el que se ha creado cada una. Los cuales mostramos en el siguiente punto.

**2.- Crear una tabla con cada motor de almacenamiento indicado**

CREATE TABLE jugadoresInno (

id INT NOT NULL AUTO\_INCREMENT,

nombre VARCHAR(50) NOT NULL,

fecha\_inscrp DATE NOT NULL,

altura FLOAT NOT NULL,

mas18 BOOLEAN NOT NULL,

PRIMARY KEY (id)

) ENGINE=InnoDB;

CREATE TABLE jugadoresIsam (

id INT NOT NULL AUTO\_INCREMENT,

nombre VARCHAR(50) NOT NULL,

fecha\_inscrp DATE NOT NULL,

altura FLOAT NOT NULL,

mas18 BOOLEAN NOT NULL,

PRIMARY KEY (id)

) ENGINE=MyISAM;

CREATE TABLE jugadoresMemory (

id INT NOT NULL AUTO\_INCREMENT,

nombre VARCHAR(50) NOT NULL,

fecha\_inscrp DATE NOT NULL,

altura FLOAT NOT NULL,

mas18 BOOLEAN NOT NULL,

PRIMARY KEY (id)

) ENGINE=MEMORY;

CREATE TABLE jugadoresCSV (

id INT NOT NULL,

nombre VARCHAR(50) NOT NULL,

fecha\_inscrp DATE NOT NULL,

altura FLOAT NOT NULL,

mas18 BOOLEAN NOT NULL

) ENGINE=CSV;

CREATE TABLE jugadoresArchive (

id INT NOT NULL AUTO\_INCREMENT,

nombre VARCHAR(50) NOT NULL,

fecha\_inscrp DATE NOT NULL,

altura FLOAT NOT NULL,

mas18 BOOLEAN NOT NULL,

PRIMARY KEY (id)

) ENGINE=ARCHIVE;

CREATE TABLE jugadoresBH (

id INT NOT NULL AUTO\_INCREMENT,

nombre VARCHAR(50) NOT NULL,

fecha\_inscrp DATE NOT NULL,

altura FLOAT NOT NULL,

mas18 BOOLEAN NOT NULL,

PRIMARY KEY (id)

) ENGINE=BLACKHOLE;

**3.- Insertar registros en cada tabla**

Hemos insertado 1000 registros por tabla mediante un generador de consultas de inserción de datos.

SOURCE jugadoresArchive.sql;

SOURCE jugadoresBH.sql;

SOURCE jugadoresCSV.sql;

SOURCE jugadoresInno.sql;

SOURCE jugadoresISAM.sql;

SOURCE jugadoresMemory.sql;

**4.- Consultar los registros insertados**

Imagen que contiene Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Imagen que contiene Tabla

Descripción generada automáticamente

Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Texto

Descripción generada automáticamente con confianza media

Tiene las tablas:

Texto

Descripción generada automáticamente

Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media

Imagen que contiene Tabla

Descripción generada automáticamente

Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media

Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

Como se puede observar la tabla ARCHIVE es la que más tarda en imprimir los registros (0.04s). Este motor está pensado para grandes volúmenes de datos, para datos históricos de los cuales no se necesite hacer un UPDATE o DELETE. Por ello faltan este tipo de consultas.

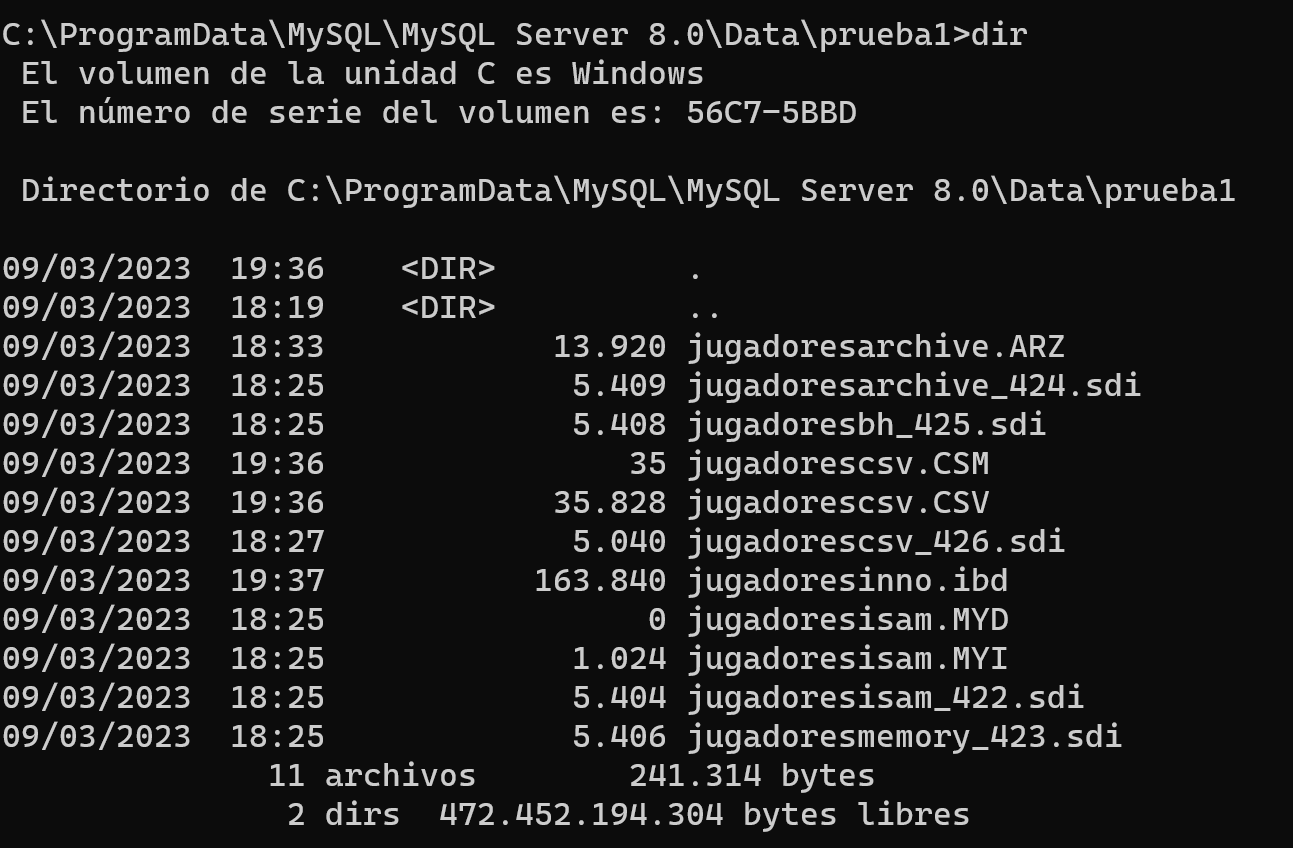
A la hora de borrar el motor que más tarda es el CSV puesto que los datos se almacenan en un fichero CSV, es decir, separado por comas, lo que ralentiza tanto la extracción de la información como la actualización de esta.

Finalmente, observando los tiempos de extracción, actualización y borrado de registros recomendamos los siguientes motores para estas distintas funcionalidades:

* **InnoDB**: Lo recomendamos para aplicaciones que tengan una gran cantidad de datos y se accedan por frecuencia debido a que sus tiempos de actualización y borrados están en la media y la extracción es rápida. Además son perfectas para las aplicaciones que requieren transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad
* **Archive**: Lo recomendamos para datos históricos que no necesiten ni actualizaciones ni borrados.
* **CSV**: Lo recomendamos solo si se necesita visualizar los archivos en formato .csv para abrirlo con aplicaciones del día a día como Xcel, Numbers, RStudio, …
* **Memory**: Este motor es perfecto para aplicaciones que requieren de un acceso my rápido y donde la pérdida de meoria no suponga un problema.
* **Blackhole**: Este motor es útil para copiar tablas o para aplicaciones que no necesiten almacenar datos de forma permanente, ya que puede descartar los datos o replicarlos a múltiples servidores para que estos sean procesados en otro lugar.
* **MyISAM**: Recomendamos este motor para aplicaciones que requieran una alta velocidad de lectura y no necesiten las transacciones ACID. Se puede utilizar como alternativa a InnoDB si no se necesita una alta carga de escritura (en este caso MyISAM no es muy fiable.

5-ANÁLISIS DE RENDIMIENTO DE LOS DIFERENTES MOTORES:

**Comportamineto en el disco:**

****

Se crean un total de 121archivos después de cargar todos los archivos .sql . estos ocupan un total de 241.314 bytes. Además solo se pueden ver una vez cargados los archivos por lo que al acceder a la carpeta data de la base de datos mysql, no encontramos dentro de la carpeta prueba 1(la base de datos creada para almacenar toda la información) los archivos mostrados en la imagen.

Los nombres de los archivos tienen el mismo nombre que sus respectivas tablas con la extensión del motor con la que se ha creado su respectiva tabla original.

**Contenido:**

jugadoresarchive.ARZ: Este archivo contiene datos de la tabla.

jugadoresarchive\_424.sdi : Este archivo contiene información sobre la ubicación física de los datos en el archivo de datos de la tabla.

jugadoresbh\_425.sdi: contiene la información

jugadorescsv.CSM: Este archivo contiene datos de la tabla.

jugadorescsv.CSV: Archivo de extensión csv, las tablas con el motor CSV se guardan en este formato de texto plano.

jugadorescsv\_426.sdi: Este archivo contiene información sobre la ubicación física de los datos en el archivo de datos de la tabla.

jugadoresinno.ibd: archivo de los datos de la tabla

0 jugadoresisam.MYD: contiene los registros de la tabla (se generan gracias al motor junto al archivo .MYI)

jugadoresisam.MYI: contiene los índices de la tabla

jugadoresisam\_422.sdi: Este archivo contiene información sobre la ubicación física de los datos en el archivo de datos de la tabla.

jugadoresmemory\_423.sdi: Este archivo contiene información sobre la ubicación física de los datos en el archivo de datos de la tabla.

Como se observa, no se ha guardado ningún archivo de la tabla jugadoresBH. Al ser un motor Blackhole todos los datos se envían a /dev/null.