



---

# PRÁCTICA 2

## ADMINISTRACIÓN DE BBDD

---

Adrián Alonso Ridao



2º ASIR  
1º EVALUACIÓN  
Administración de Base de Datos

## Contenido

Funcionamiento del Servidor MySQL .....	2
Ejercicio 1: Preguntas teóricas sobre la arquitectura .....	2
Ejercicio 2: Identifica el motor de almacenamiento .....	3
Ejercicio 3: Consultas al Diccionario de Datos (INFORMATION_SCHEMA) .....	3
Ejercicio 4: Comandos SHOW .....	5

# Funcionamiento del Servidor MySQL

## Ejercicio 1: Preguntas teóricas sobre la arquitectura

Responde brevemente a las siguientes preguntas:

**1. ¿Qué hace la "capa de manipulación" (management layer) cuando recibe una petición de un cliente?**

Recibe la petición, autentica al cliente, valida permisos, y solo si todo es correcto, envía la consulta al motor SQL para que sea analizada y ejecutada.

**2. Nombra dos responsabilidades de los motores de almacenamiento.**

- Gestión del almacenamiento físico de los datos
- Manejo de transacciones y bloqueo

**3. Además de las bases de datos, ¿qué otros tipos de información almacena MySQL en el disco?**

Metadatos, registros de actividad, temporales y archivos de control necesarios para la administración, seguridad y recuperación del sistema.

**4. Además de los gestores de conexión, ¿qué otros elementos almacena MySQL en la memoria?**

Mantiene en memoria buffers, cachés, metadatos y estructuras temporales que permiten acelerar el acceso a datos y reducir el trabajo de disco.

**5. ¿Para qué se utiliza el buffer de comunicaciones (net\_buffer\_length, max\_allowed\_packet) en el cliente?**

Se utiliza para gestionar y optimizar el intercambio de información entre el cliente y el servidor MySQL, asegurando que las consultas y resultados se transmitan de manera eficiente y sin errores por límites de tamaño.

## Ejercicio 2: Identifica el motor de almacenamiento

Lee los siguientes escenarios y elige el motor de almacenamiento más adecuado (MyISAM, InnoDB, MEMORY/HEAP). Justifica tu respuesta.

- 1. Una tabla de lookup (o de búsqueda) con 1000 filas de países y sus códigos, que es leída constantemente pero casi nunca se modifica. El rendimiento de lectura debe ser extremadamente rápido.**

MEMORY/HEAP, ya que prioriza la velocidad de lectura en consultas frecuentes y la tabla es suficientemente pequeña para mantenerse en RAM.

- 2. La tabla de "Pedidos" de una tienda online, donde es crucial que las operaciones (como registrar el pedido, descontar stock y registrar la transacción financiera) se completen todas correctamente o no se realice ninguna. Además, existen relaciones con otras tablas como "Clientes" y "Productos".**

InnoDB, porque garantiza consistencia, integridad y seguridad en transacciones críticas.

- 3. Una tabla que almacena las sesiones temporales de los usuarios en una aplicación web. Los datos son válidos solo durante un corto período de tiempo y se borran con frecuencia.**

MEMORY es la mejor opción para tablas de sesiones temporales que requieren velocidad y manejo frecuente de datos efímeros.

## Ejercicio 3: Consultas al Diccionario de Datos (INFORMATION\_SCHEMA)

Escribe las consultas SQL para obtener la siguiente información del diccionario de datos. Asume que existe una base de datos llamada mi\_tienda.

1. Mostrar una lista de todas las tablas que existen en la base de datos **mi\_tienda**.

```
mysql> show tables;
+-----+
| Tables_in_information_schema |
+-----+
| ADMINISTRABLE_ROLE_AUTHORIZATIONS |
| APPLICABLE_ROLES |
| CHARACTER_SETS |
| CHECK_CONSTRAINTS |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLLATIONS |
| COLUMN_PRIVILEGES |
| COLUMN_STATISTICS |
| COLUMNS |
| COLUMNS_EXTENSIONS |
| ENABLED_ROLES |
| ENGINES |
```

2. Mostrar todas las columnas de la tabla **productos** en la base de datos **mi\_tienda**.

```
mysql> desc tables
-> ;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| TABLE_CATALOG | varchar(64) | YES | | NULL | |
| TABLE_SCHEMA | varchar(64) | YES | | NULL | |
| TABLE_NAME | varchar(64) | YES | | NULL | |
| TABLE_TYPE | enum('BASE TABLE','VIEW','SYSTEM VIEW') | NO | | NULL | |
| ENGINE | varchar(64) | YES | | NULL | |
| VERSION | int | YES | | NULL | |
```

3. Mostrar el motor de almacenamiento (ENGINE) de todas las tablas de la base de datos **mi\_tienda**.

```
mysql> SELECT table_name, engine FROM information_schema.tables;
+-----+-----+
| TABLE_NAME | ENGINE |
+-----+-----+
| CHARACTER_SETS | NULL |
| CHECK_CONSTRAINTS | NULL |
| COLLATION_CHARACTER_SET_APPLICABILITY | NULL |
| COLLATIONS | NULL |
| COLUMN_PRIVILEGES | NULL |
| COLUMN_STATISTICS | NULL |
| COLUMNS | NULL |
| ENGINES | NULL |
| EVENTS | NULL |
```

## Ejercicio 4: Comandos SHOW

Ejecuta los siguientes comandos y observa el resultado. Si no, investiga para qué sirve cada uno.

### 1. SHOW VARIABLES.

```
mysql> SHOW VARIABLES
-> ;
```

Variable_name	Value
activate_all_roles_on_login	OFF
admin_address	
admin_port	33062
auto_generate_certs	ON

## 2. SHOW STATUS.

```
mysql> SHOW VARIABLES
-> ;
```

Variable_name	Value
activate_all_roles_on_login	OFF
admin_address	
admin_port	33062
auto_generate_certs	ON

## 3. SHOW ENGINES.

```
mysql> SHOW ENGINES;
```

Engine	Support	Comment	Tran
MEMORY	YES	Hash based, stored in memory, useful for temporary tables	NO
MRG_MYISAM	YES	Collection of identical MyISAM tables	NO
CSV	YES	CSV storage engine	NO
FEDERATED	NO	Federated MySQL storage engine	NULL
PERFORMANCE_SCHEMA	YES	Performance Schema	NO
MyISAM	YES	MyISAM storage engine	NO
InnoDB	DEFAULT	Supports transactions, row-level locking, and foreign keys	YES
BLACKHOLE	YES	/dev/null storage engine (anything you write to it disappears)	NO
ARCHIVE	YES	Archive storage engine	NO

9 rows in set (0.00 sec)

4. SHOW TABLES FROM information\_schema.

```
mysql> SHOW TABLES FROM information_schema;
```

Tables_in_information_schema
CHARACTER_SETS
CHECK_CONSTRAINTS
COLLATION_CHARACTER_SET_APPLICABILITY
COLLATIONS
COLUMN_PRIVILEGES
COLUMN_STATISTICS
COLUMNS
ENGINES
EVENTS
FILES
INNODB_BUFFER_PAGE
INNODB_BUFFER_PAGE_LRU
INNODB_BUFFER_POOL_STATS
INNODB_CACHED_INDEXES
INNODB_CMP
INNODB_CMP_PER_INDEX