

BASE DE DATOS APLICADA



Administración y Gestión de Bases de Datos con MySQL

**Claves para la
Operación,
Mantenimiento y
Seguridad de los
Datos**



Introducción a la Administración de Bases de Datos

- **Administración de Bases de Datos**
 - Conjunto de tareas y procesos para asegurar la **disponibilidad, rendimiento, seguridad e integridad** de los datos.
 - Va más allá de la instalación; es la gestión del ciclo de vida completo de la base de datos.
- **Rol del DBA (Administrador de Bases de Datos):**
 - Diseño, implementación, mantenimiento, monitorización, optimización y resolución de problemas.
- **Importancia:** Una buena administración garantiza que las aplicaciones funcionen sin problemas y que la información crítica esté siempre protegida y accesible.



Creación y Configuración de Base de Datos



- **Creación de una Base de Datos:**

```
CREATE DATABASE IF NOT EXISTS base
DEFAULT CHARACTER SET utf8mb4
DEFAULT COLLATE utf8mb4_unicode_ci;
```

¿Qué es un conjunto de caracteres?

Conjunto de caracteres: tabla o un mapa que asigna un código numérico a cada carácter, como letras, números y símbolos.

Ejemplo:

Conjunto de caracteres **ASCII** asigna un número a cada letra del alfabeto inglés, a los números y a algunos símbolos básicos.

Limitaciones ASCII: No puede manejar caracteres de otros idiomas, como la 'ñ' o los caracteres cirílicos y asiáticos.

Solución Se crean otros conjuntos de caracteres (**Unicode**), que incluye casi todos los caracteres de todos los idiomas del mundo. Implementación más utilizada de Unicode es **UTF-8**.

Creación y Configuración de Base de Datos

DEFAULT CHARACTER SET `utf8mb4`

`utf8mb4` es una implementación completa del estándar de codificación **UTF-8**

La principal característica de `utf8mb4` es que puede utilizar **hasta 4 bytes** para codificar un solo carácter.

`utf8mb4` asegura que la base de datos pueda almacenar y procesar **cualquier carácter Unicode**

Se evitan problemas de truncación de datos o aparición de caracteres extraños.



Creación y Configuración de Base de Datos



DEFAULT CHARACTER SET `utf8mb4`

¿Qué define `DEFAULT CHARACTER SET utf8mb4`?

Este ajuste establece que, por defecto, cualquier nueva tabla o columna de texto que crees en tu base de datos usará la codificación `utf8mb4`.

- **Alcance:** La configuración por defecto se aplica a nivel de **base de datos**

Caso especial: La configuración puede ser sobrescrita a nivel de **tabla** o **columna** en casos de datos que necesiten otra representación.

- **Compatibilidad:** `utf8mb4` es práctica recomendada. Garantiza compatibilidad global. Es el conjunto de caracteres predeterminado en las versiones más recientes de MySQL.
- **Se combina con Collation:** Este ajuste a menudo va de la mano con una **collation** (por ejemplo, `utf8mb4_0900_ai_ci`), que define las reglas para ordenar y comparar los datos de texto. El nombre de la collation indica el conjunto de caracteres (`utf8mb4`), la versión de Unicode (`0900`), y si la comparación es insensible a acentos (`ai`) y a mayúsculas/minúsculas (`ci`).

Creación y Configuración de Base de Datos



```
DEFAULT COLLATE utf8mb4_unicode_ci;
```

Define las reglas por defecto para la comparación y ordenación de cadenas de texto en la base de datos con codificación `utf8mb4`.

Partes de la cláusula

- **utf8mb4**: Conjunto de caracteres en base a la codificación Unicode.
- **unicode_ci**: Define las reglas de la collation.
 - **unicode**: Indica que las reglas de comparación y ordenación se basan en el **Unicode Collation Algorithm (UCA)**.
 - **ci**: Significa "**Case-Insensitive**" (insensible a mayúsculas y minúsculas). Al comparar cadenas de texto, la collation no distingue entre letras mayúsculas y minúsculas.

Ejemplo, 'A' será considerada igual a 'a', y 'Texto' será igual a 'texto'.

Creación y Configuración de Base de Datos



```
DEFAULT COLLATE utf8mb4_unicode_ci;
```

La cláusula define:

1. **Comparación insensible a mayúsculas y minúsculas:** Consultas `SELECT` con cláusulas `WHERE` que comparan cadenas de texto ignoran la diferencia entre mayúsculas y minúsculas.

2. Ejemplo:

```
SELECT * FROM usuarios WHERE nombre = 'JUAN'
```

 encontrará registros con nombres como "Juan", "juan", o "JUAN".

3. **Ordenación de texto basada en Unicode:** Al ordenar los resultados de una consulta `SELECT` con `ORDER BY`, el ordenamiento se hará de acuerdo con las reglas lingüísticas del estándar Unicode. Esto es beneficioso para manejar texto multilingüe de forma consistente.
4. `utf8mb4` se ha actualizado a `utf8mb4_0900_ai_ci`. Parte `ai` significa "Accent-Insensitive" (insensible a acentos)

Creación y Configuración de Base de Datos



Configuración Inicial

1. Archivos **my.cnf** / **my.ini**
 1. Ajustes para el rendimiento y comportamiento de MySQL.
2. **Parámetros clave:**
 - a. **innodb_buffer_pool_size**: Cantidad de RAM dedicada a InnoDB (motor de almacenamiento).
 - b. **max_connections**: Número máximo de conexiones simultáneas.
 - c. **log_error**: Ruta del archivo de log de errores.
 - d. **port**: Puerto de escucha (por defecto 3306).

Creación y Configuración de Base de Datos



innodb_buffer_pool_size

Define rendimiento de MySQL.

Se recomienda 70-80% de la RAM disponible del servidor (Para el caso de servidor MySQL dedicado).

InnoDB buffer pool es donde se almacenan en caché los datos y los índices. Su configuración adecuada reduce las operaciones de E/S del disco.

Ejemplo

- **Servidor con 8 GB de RAM:** `innodb_buffer_pool_size = 6G`
- **Servidor con 16 GB de RAM:** `innodb_buffer_pool_size = 12G`

Creación y Configuración de Base de Datos



max_connections

Define la cantidad máxima de conexiones simultáneas que el servidor puede manejar.

Configurar un número alto puede consumir mucha RAM.

Valor puede ser moderado para el caso en que las conexiones de cierren inmediatamente después de usarlas.

- **Aplicación web pequeña:** `max_connections = 250`
- **Servidor de base de datos con alta concurrencia:** `max_connections = 1000`

Creación y Configuración de Base de Datos



log_error

Especifica la ubicación del archivo de registro de errores de MySQL.

El archivo de registro de errores es importante para la resolución de problemas

Se registran errores, advertencias e información de inicio/cierre.

- **Configuración común:** `log_error = /var/log/mysql/error.log`

```
2025-08-14T11:45:01.023456Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.29) starting as process 12345
2025-08-14T11:45:01.023500Z 0 [Warning] [MY-010068] [Server] The user specified as "mysql" in the --user option does not exist. Using the current user.
2025-08-14T11:45:02.123456Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.29' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL
Community Server - GPL.
2025-08-14T11:45:05.567890Z 10 [Warning] [MY-011082] [Server] Invalid (old?) table or database name: 'my-old-database'.
2025-08-14T11:46:10.123456Z 15 [ERROR] [MY-010584] [Server] A connection attempt failed because the host '192.168.1.100' is not allowed to connect to this MySQL server.
2025-08-14T11:47:03.987654Z 20 [ERROR] [MY-010586] [Server] Can't start server: Bind on TCP/IP port: Address already in use
2025-08-14T11:47:03.987654Z 20 [ERROR] [MY-010119] [Server] Aborting
2025-08-14T11:47:04.123456Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown complete (mysqld 8.0.29).
```

Creación y Configuración de Base de Datos

port

Este parámetro establece el puerto TCP/IP que MySQL

Por este puerto escucha las conexiones de los clientes.

El puerto por defecto es el 3306 (recomendado),

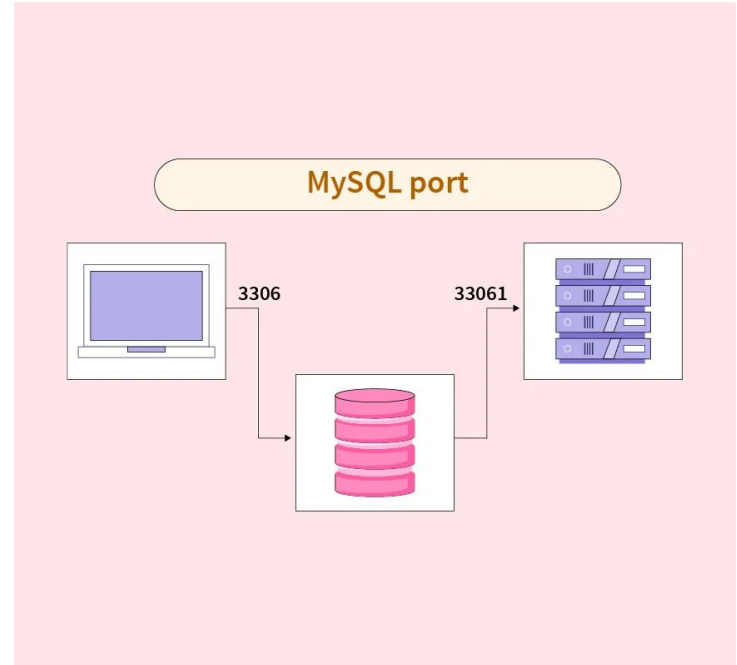
Cambio por razones de seguridad o si otro servicio ya está usando ese puerto.

- **Configuración predeterminada (recomendada):**

port = 3306

- **Configuración alternativa (si es necesario):**

port = 3307



Gestión de Usuarios y Permisos

- **Seguridad por Principio de Mínimo Privilegio:** Asignar solo los permisos necesarios.
- **Creación de Usuarios**

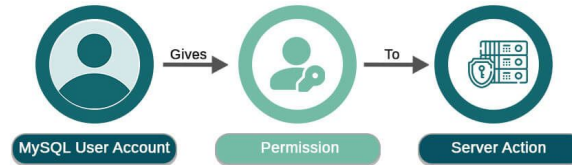
– Acceso desde localhost

```
CREATE USER 'app_user'@'localhost' IDENTIFIED BY 'pwd123';
```

– Acceso desde cualquier host

```
CREATE USER 'report_user'@'%' IDENTIFIED BY 'pwd123';
```

MySQL User Permissions



Gestión de Usuarios y Permisos

- **Asignación de Permisos (GRANT):**

– Permisos de S/I/U/D al usuario app_user

```
GRANT SELECT, INSERT, UPDATE, DELETE ON mi_aplicacion_db.* TO 'app_user'@'localhost';
```

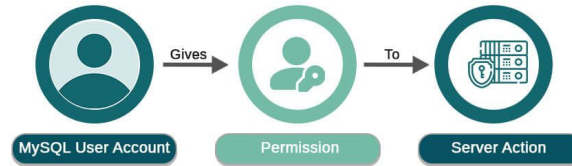
– Permiso de S al usuario report_user

```
GRANT SELECT ON mi_aplicacion_db.productos TO 'report_user'@'%';
```

-- Recarga los privilegios

```
FLUSH PRIVILEGES;
```

MySQL User Permissions

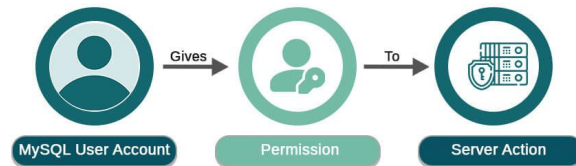


Gestión de Usuarios y Permisos

- **Revocación de Permisos (REVOKE):**

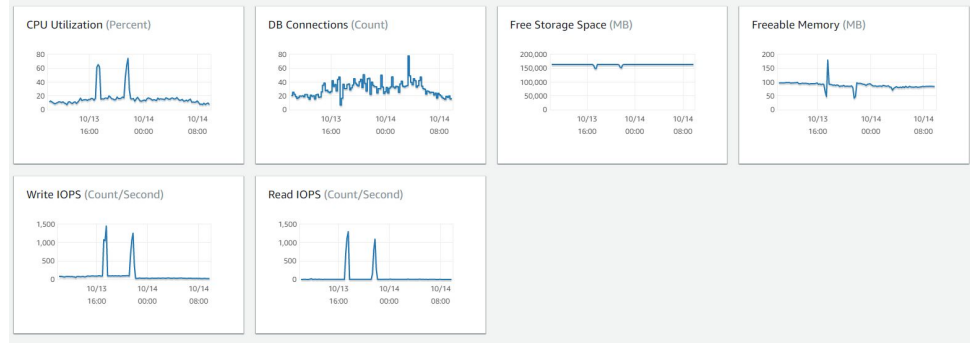
```
- Quita permiso de Delete al usuario app_user  
REVOKE DELETE ON mi_aplicacion_db.* FROM 'app_user'@'localhost';  
  
-- Recarga los privilegios  
FLUSH PRIVILEGES;
```

MySQL User Permissions



Monitorización de Rendimiento

- Identificar cuellos de botella, problemas de rendimiento y uso de recursos.
- **Métricas Clave:**
 - **Uso de CPU y RAM**
 - **Conexiones activas**
 - **Consultas lentas (Slow Queries):**
 - **Uso de disco**
 - **Buffer Pool Hit Ratio (Eficiencia del caché de InnoDB)**



Monitorización de Rendimiento

Herramientas:

Comandos SQL para información en tiempo real.

- **SHOW STATUS;**
- **SHOW GLOBAL STATUS LIKE 'Com_delete';**
- **SHOW PROCESSLIST;**
- **mysqldumpslow:** Para analizar logs de consultas lentas.

**mysqldumpslow -s t -t 10
/var/log/mysql/mysql-slow.log**

```
| Connections | 31 |
| Cpu_time    | 0.000000 |
| Created_tmp_disk_tables | 0 |
| Created_tmp_files | 4 |
| Created_tmp_tables | 3 |
| Delayed_errors | 0 |
| Delayed_insert_threads | 0 |
| Delayed_writes | 0 |
| Delete_scan | 0 |
| Empty_queries | 0 |
```

```
MariaDB [sakila]> SHOW PROCESSLIST;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Id | User | Host | db | Command | Time | State | Info | Progress |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 31 | root | localhost | sakila | Query | 0 | starting | SHOW PROCESSLIST | 0.000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [sakila]> 
```

```
Reading mysql-slow.log...
Count: 2 Time=2.12s (4s) Lock=0.00s (0s) Rows_sent=100.0 (200), Rows_examined=100.0 (200), Rows_affected=0.0 (0)
      SELECT id, nombre, apellido FROM usuarios WHERE id = S

Count: 5 Time=1.56s (7s) Lock=0.00s (0s) Rows_sent=10.0 (50), Rows_examined=1000.0 (5000), Rows_affected=0.0 (0)
      SELECT * FROM productos WHERE precio > N ORDER BY nombre
```

Mantenimiento de Base de Datos

Tareas de mantenimiento de bases

- **Optimización de Tablas:** Reorganiza el almacenamiento físico de los datos para mejorar el rendimiento.
 - `OPTIMIZE TABLE tabla;`
- **Análisis de Tablas:** Recopila estadísticas sobre las tablas para que el optimizador de consultas las use mejor.
 - `ANALYZE TABLE tabla;`
- **Revisión de Errores en Logs:** archivo `error.log`
- **Limpieza de Datos Antiguos/Innecesarios:** Borrar registros que ya no son útiles.
- **Automatización:** Se pueden programar las tareas de mantenimiento. (cron, task)

```
MariaDB [sakila]> optimize table actor;
```

Table	Op	Msg_type	Msg_text
sakila.actor	optimize	note	Table does not support optimize, doing recreate + analyze instead
sakila.actor	optimize	status	OK

```
2 rows in set (0,753 sec)
```

```
MariaDB [sakila]> 
```

```
MariaDB [sakila]> Analyze table actor;
```

Table	Op	Msg_type	Msg_text
sakila.actor	analyze	status	OK

```
1 row in set (0,038 sec)
```

```
MariaDB [sakila]> 
```

Seguridad en Bases de Datos

Enfoque integral

- **Seguridad a nivel de red:** Firewall.
- **Seguridad a nivel de sistema operativo:** Proteger el servidor, actualizaciones, antivirus.
- **Seguridad a nivel de SGBD:**
 - **Gestión de Usuarios y Permisos**
 - **Cifrado de datos**
 - **Inyección SQL** (Usar consultas parametrizadas).
 - **Auditoría.**

Ejemplo de Inyección SQL

```
SELECT * FROM usuarios WHERE username = 'admin' AND  
password = " OR '1'='1';
```

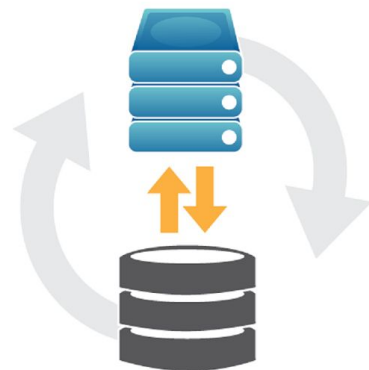
Seguridad en Base de Datos

Niveles de Seguridad



Backup y Restauración

- **Backup (Copia de Seguridad)**
- **Restauración** desde una copia de seguridad.
- **Estrategias de Respaldo:**
 - **Completo:** Copia toda la base de datos.
 - **Incremental:** Copia solo los datos que han cambiado desde el último backup (completo o incremental).
 - **Diferencial:** Copia los datos que han cambiado desde el último backup *completo*.
- **Herramientas en MySQL:**
 - **mysqldump:** Para backups lógicos (archivos SQL).
 - MySQL Enterprise Backup (MEB): Para backups físicos (más rápidos, para grandes DBs).
 - Snapshots de volúmenes (a nivel de SO/virtualización).



Backup y Restauración



- Backup

```
mysqldump -u [usuario] -p [nombre_base_de_datos] > [nombre_archivo].sql
```

Ejemplo:

```
mysqldump -u root -p nombre_base > nombre_base_backup.sql
```

- Restauración

```
mysql -u [usuario] -p [nombre_base_de_datos] < [nombre_archivo].sql
```

Ejemplo:

```
mysql -u root -p nombre_base < nombre_base_backup.sql
```

(Crear la base de datos destino de la restauración)

Plan de Respaldo y Pruebas de Restauración



- **Plan de Respaldo:** Documento que define:
 - **Qué** respaldar (bases de datos, tablas específicas).
 - **Cuándo** respaldar (frecuencia: diaria, semanal, horaria).
 - **Cómo** respaldar (herramienta, tipo de backup).
 - **Dónde** almacenar los backups (local, remoto, nube).
 - **Quién** es responsable.
 - **Retención:** Cuánto tiempo se guardan los backups.
- **Pruebas de Restauración:**
 - Simular una falla y restaurar un backup en un entorno de prueba.
 - Verificar la integridad y consistencia de los datos restaurados.
- **Registro de Respaldo:** Documentar cada backup realizado (fecha, hora, tamaño, estado)