



Sesión 19

Tema	Herramientas de gestión de bases de datos.
Propósito	Poner en conocimiento de los participantes los fundamentos y características de las herramientas de administración de bases de datos, mediante la correspondiente presentación y demostración del docente en una sesión expositiva-participativa..
Fecha	C.10.11.2025
Hora	17:00

Este es un aspecto fundamental en el mundo del desarrollo y administración de sistemas.

Herramientas de Administración de Bases de Datos Relacionales

Las herramientas se pueden categorizar en varios tipos según su función y modalidad:

■ Herramientas Principales por Proveedor

MySQL:

- MySQL Workbench (oficial)
- phpMyAdmin (web-based)
- HeidiSQL (Windows)
- Adminer (ligera, web-based)

PostgreSQL:

- pgAdmin (oficial)
- DBeaver (multi-plataforma)
- Navicat for PostgreSQL



SQL Server:

- SQL Server Management Studio (SSMS)
- Azure Data Studio

Oracle:

- SQL Developer
- Oracle Enterprise Manager

🌐 Herramientas Multiplataforma

- **DBeaver** - Open source, soporta múltiples motores
- **DataGrip** (JetBrains) - IDE profesional
- **TablePlus** - Interfaz moderna y rápida
- **Beekeeper Studio** - Open source y moderna

⌚ Funcionalidades Clave

- Gestión de esquemas y tablas
- Ejecución de consultas SQL
- Monitorización de rendimiento
- Backup y restauración
- Gestión de usuarios y permisos
- Optimización de índices



III Análisis Detallado de Herramientas de Administración

MySQL/MariaDB

MySQL Workbench (Oficial)

Características Técnicas:

- **Arquitectura:** Aplicación desktop nativa (C++/Python)
- **Conexiones:** MySQL 5.6+, MariaDB 10.0+
- **Características destacadas:**
 - Diseñador visual EER (Enhanced Entity Relationship)
 - Migración de bases de datos
 - Administración de usuarios y privilegios
 - Backup y restore con mysqldump integrado
 - Performance monitoring básico

Limitaciones:

- Solo para MySQL/MariaDB
- Consumo moderado de recursos
- Interfaz puede ser compleja para principiantes

phpMyAdmin (Web-based)

Características Técnicas:

- **Arquitectura:** Aplicación web (PHP/JavaScript)
- **Requisitos:** Servidor web + PHP + MySQL
- **Fortalezas:**
 - Acceso desde cualquier navegador
 - Bajo consumo en cliente



- Amplia adopción en hosting compartido

Debilidades:

- Seguridad dependiente de configuración
- Rendimiento con bases de datos muy grandes
- Funcionalidades limitadas vs herramientas desktop

Microsoft SQL Server

SQL Server Management Studio (SSMS) - Oficial

Características Técnicas:

- **Plataforma:** Windows only
- **Arquitectura:** .NET Framework
- **Características avanzadas:**
 - Query tuning y execution plan analysis
 - SQL Server Agent integration
 - Profiler y Extended Events
 - AlwaysOn y High Availability management

Puntos fuertes:

- Integración completa con el ecosistema Microsoft
- Herramientas de diagnóstico robustas
- Soporte para todas las características de SQL Server



Oracle Database

SQL Developer (Oficial)

Características Técnicas:

- **Plataforma:** Multiplataforma (Java)
- **Requisitos:** Java JDK 8+
- **Capacidades técnicas:**
 - Gestión de objetos PL/SQL avanzada
 - Data Modeler integrado
 - Soporte para Oracle Cloud
 - Migración desde terceros (MySQL, SQL Server)

Ventajas:

- Completamente gratuita
- Actualizaciones frecuentes
- Amplia documentación oficial

Herramientas Multiplataforma Free

DBeaver (Community Edition)

Características Técnicas:

- **Arquitectura:** Eclipse RCP (Java)
- **Soporte:** 80+ conexiones diferentes
- **Características destacables:**
 - Editor SQL con autocompletado avanzado
 - Data compare y sync
 - ER diagram generation



- Extensiones via plugins

Fortalezas:

- Una herramienta para múltiples motores
- Active development community
- Lightweight compared to official tools

HeidiSQL

Características Técnicas:

- **Plataforma:** Windows (via Wine en Linux/macOS)
- **Conexiones:** MySQL, MariaDB, SQL Server, PostgreSQL
- **Especialización:**
 - Interfaz muy limpia y rápida
 - Bulk data operations
 - Session monitoring
 - Export/import flexibles

☒ Cuadro Comparativo Detallado

Her-ramiento	Licencia	Multi-DB	Per-for-manc-e	Curva Aprendizaje	Carac-terístic-a Única
MySQL					
Work-bench	Free	No	Alta	Media	EER Diagram Designer



Her-ramienta	Licencia	Multi-DB	Per-forman-cia	Curva Aprendizaje	Carac-terístic-a Única
phpMyAdmin	GPL	No	Me-dia-Baja	Baja	Acceso web universal
SSMS	Free	No	Muy Alta	Media-Alta	SQL Server Profiler
SQL De-veloper	Free	Limit-ado	Alta	Media	Oracle Data Modeler
DBeaver CE	Free	Sí	Me-dia-Alta	Media	80+ Connectors
HeidiSQL	Free	Sí	Alta	Baja	Simpli-cidad y velocidad
Her-ramienta	Requisitos Sistema	Porta-ble	Co-muni-dad	Docu-mentació-n	Soport-e Cloud
MySQL Work-bench	2GB RAM, Windows/Linux/macOS	No	Activa	Excelente	AWS, Azure



Her-ramienta	Licencia	Multi-DB	Per-for-manc-e	Curva Aprendizaje	Carac-terístic-a Única
phpMyAdmin	Servidor LAMP/LEMP	Sí	Muy Activa	Buena	Limitado
SSMS	Windows, .NET 4.6+	No	Activa	Excelente	Azure SQL
SQL Developer	Java JDK 8+	Sí	Activa	Excelente	Oracle Cloud
DBeaver CE	Java JRE 8+	Sí	Muy Activa	Buena	Múltiples
HeidiSQL	Windows (Wine)	Sí	Activa	Regular	Limitado

💡 Recomendaciones de Uso por Escenario

Cuándo usar cada herramienta oficial:

MySQL Workbench:

- ✓ Desarrollo de nuevas bases de datos MySQL/MariaDB
- ✓ Diseño de esquemas complejos con EER
- ✓ Migraciones desde otros motores
- ✗ Evitar si necesitas multi-motor



SSMS:

- ✓ Entornos empresariales Microsoft
- ✓ Tuning de queries críticas
- ✓ Administración de alta disponibilidad
- ✗ No usar fuera del ecosistema Microsoft

SQL Developer:

- ✓ Desarrollo con Oracle Database
- ✓ Trabajo con PL/SQL avanzado
- ✓ Entornos multi-plataforma
- ✗ Limitado para otros motores

Cuándo usar herramientas multiplataforma:

DBeaver Community:

- ✓ Equipos con múltiples tecnologías de BD
- ✓ Desarrollo cross-platform
- ✓ Necesidad de herramienta unificada
- ✓ Presupuesto limitado

HeidiSQL:

- ✓ Administración rápida de MySQL/MariaDB
- ✓ Equipos con preferencia por interfaces simples
- ✓ Operaciones bulk de datos

phpMyAdmin:

- ✓ Hosting compartido o entornos web
- ✓ Acceso remoto sin instalación desktop



- ✓ Recursos limitados en cliente

🏆 Recomendación

Para equipos empresariales:

"**SSMS + DBeaver**" - Lo mejor de ambos mundos: especialización para SQL Server y versatilidad para otros motores.

Para desarrolladores freelance/startups:

"**DBeaver Community**" - Cubre el 95% de necesidades con una sola herramienta, multiplataforma y gratuita.

Para entornos educativos:

"**MySQL Workbench + HeidiSQL**" - Excelente balance entre capacidades avanzadas y simplicidad.

Para administradores Oracle:

"**SQL Developer**" - Especializada, gratuita y en constante evolución.

⌚ Tendencia Actual

El mercado se mueve hacia herramientas multiplataforma como **DBeaver** que reducen la curva de aprendizaje y permiten estandarización en equipos heterogéneos.



💻 Administración por Consola/Terminal

MySQL/MariaDB

Conexión y Autenticación

```
bash
# Conexión básica
mysql -u root -p

# Conexión con parámetros específicos
mysql -h localhost -u usuario -p -P 3306 nombre_base

# Conexión con socket
mysql -S /var/run/mysqld/mysqld.sock -u root -p
```

Gestión de Usuarios y Privilegios

```
sql
-- Crear usuario
CREATE USER 'nuevo_usuario'@'localhost' IDENTIFIED BY 'password_seguro';

-- Privilegios completos sobre una base de datos
GRANT ALL PRIVILEGES ON basedatos.* TO 'usuario'@'localhost';

-- Privilegios específicos
GRANT SELECT, INSERT, UPDATE ON basedatos.* TO 'usuario'@'%';

-- Ver privilegios de un usuario
SHOW GRANTS FOR 'usuario'@'localhost';

-- Revocar privilegios
REVOKE DELETE ON basedatos.* FROM 'usuario'@'localhost';

-- Eliminar usuario
DROP USER 'usuario'@'localhost';
```



Backup y Restauración

```
bash
# Backup completo de base de datos
mysqldump -u usuario -p nombre_base > backup_completo.sql

# Backup de todas las bases de datos
mysqldump -u usuario -p --all-databases > backup_total.sql

# Backup con compresión
mysqldump -u usuario -p nombre_base | gzip > backup.sql.gz

# Backup solo estructura
mysqldump -u usuario -p --no-data nombre_base > solo_estructura.sql

# Backup solo datos
mysqldump -u usuario -p --no-create-info nombre_base > solo_datos.sql

# Restauración
mysql -u usuario -p nombre_base < backup_completo.sql

# Restauración desde archivo comprimido
gunzip < backup.sql.gz | mysql -u usuario -p nombre_base
```

Operaciones de Auditoría y Monitoreo

```
sql
-- Ver procesos activos
SHOW PROCESSLIST;

-- Ver variables de configuración
SHOW VARIABLES LIKE '%timeout%';

-- Ver status del servidor
SHOW STATUS;

-- Ver motores de almacenamiento
SHOW ENGINES;
```

```
-- Ver logs activos
SHOW VARIABLES LIKE 'log_%';

-- Consultar información de tablas
SELECT TABLE_NAME, TABLE_ROWS, DATA_LENGTH
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'nombre_base';
```

Microsoft SQL Server

Conexión con sqlcmd

```
bash
# Conexión básica
sqlcmd -S localhost -U sa -P password

# Conexión con base de datos específica
sqlcmd -S localhost -d master -U sa -P password

# Ejecutar archivo SQL desde consola
sqlcmd -S localhost -U sa -P password -i archivo.sql
```

Gestión de Usuarios y Roles

```
sql
-- Crear Login de servidor
CREATE LOGIN [nuevo_usuario] WITH PASSWORD = 'Password123';

-- Crear usuario en base de datos
USE mi_base;
CREATE USER [usuario_bd] FOR LOGIN [nuevo_usuario];

-- Asignar roles
ALTER ROLE db_owner ADD MEMBER [usuario_bd];

-- Permisos específicos
GRANT SELECT, INSERT ON dbo.tabla TO [usuario_bd];
```

```
-- Ver permisos
EXEC sp_helprotect @username = 'usuario_bd';
```

Backup y Restauración

```
sql
-- Backup completo
BACKUP DATABASE mi_base
TO DISK = 'C:\backups\mi_base.bak'
WITH FORMAT, COMPRESSION;

-- Backup diferencial
BACKUP DATABASE mi_base
TO DISK = 'C:\backups\mi_base_diff.bak'
WITH DIFFERENTIAL;

-- Restauración completa
RESTORE DATABASE mi_base
FROM DISK = 'C:\backups\mi_base.bak'
WITH REPLACE, RECOVERY;

-- Ver backups realizados
SELECT name, backup_start_date, type
FROM msdb.dbo.backupset;
```

PowerShell para SQL Server

```
powershell
# Importar módulo SQL Server
Import-Module SqlServer

# Ejecutar query
Invoke-Sqlcmd -ServerInstance "localhost" -Database "master" -Query "SELECT @@VERSION"

# Backup via PowerShell
Backup-SqlDatabase -ServerInstance "localhost" -Database "mi_base" -BackupFile "C:\backups\backup.ps1.bak"
```



Oracle Database

Conexión con SQL*Plus

```
bash
# Conexión básica
sqlplus usuario/password@//localhost:1521/SID

# Conexión como sysdba
sqlplus / as sysdba

# Ejecutar archivo SQL
sqlplus usuario/password @archivo.sql
```

Gestión de Usuarios y Privilegios

```
sql
-- Crear usuario
CREATE USER usuario IDENTIFIED BY password
DEFAULT TABLESPACE users
QUOTA UNLIMITED ON users;

-- Privilegios básicos
GRANT CREATE SESSION TO usuario;
GRANT CREATE TABLE TO usuario;

-- Roles predefinidos
GRANT CONNECT, RESOURCE TO usuario;

-- Ver sesiones activas
SELECT username, sid, serial#, machine
FROM v$session;

-- Ver privilegios de usuario
SELECT * FROM user_sys_privs;
SELECT * FROM user_role_privs;
```



Backup con Data Pump (Moderno)

```
bash
```

```
# Export completo
```

```
expdp usuario/password DIRECTORY=backup_dir DUMPFILE=export_completo.dmp FULL=Y  
S
```

```
# Export por esquema
```

```
expdp usuario/password DIRECTORY=backup_dir DUMPFILE=esquema.dmp SCHEMAS=mi_esquema
```

```
# Import
```

```
impdp usuario/password DIRECTORY=backup_dir DUMPFILE=export_completo.dmp FULL=Y  
S
```

```
# Backup tradicional (Legacy)
```

```
exp usuario/password FILE=export.dmp OWNER=mi_esquema
```

Monitoreo y Auditoría

```
sql
```

```
-- Ver tablespaces
```

```
SELECT tablespace_name, used_space, free_space  
FROM dba_tablespace_usage_metrics;
```

```
-- Ver objetos grandes
```

```
SELECT owner, segment_name, bytes/1024/1024 MB  
FROM dba_segments  
WHERE bytes > 104857600  
ORDER BY bytes DESC;
```

```
-- Ver parámetros importantes
```

```
SELECT name, value FROM v$parameter WHERE name LIKE '%memory%';
```



❖ Comandos Esenciales Multi-Motor

Operaciones Comunes Entre Motores

Ver Conexiones Activas

```
sql
-- MySQL
SHOW PROCESSLIST;

-- SQL Server
SELECT * FROM sys.dm_exec_sessions;

-- Oracle
SELECT sid, serial#, username, machine FROM v$session;
```

Kill Process

```
sql
-- MySQL
KILL [process_id];

-- SQL Server
KILL [session_id];

-- Oracle
ALTER SYSTEM KILL SESSION '[sid],[serial#]';
```

Espacio en Base de Datos

```
sql
-- MySQL
SELECT table_schema "Base",
ROUND(SUM(data_length+index_length)/1024/1024,2) "MB"
FROM information_schema.tables
GROUP BY table_schema;

-- SQL Server
```



```
EXEC sp_spaceused;

-- Oracle
SELECT tablespace_name,
SUM(bytes)/1024/1024 "MB"
FROM dba_data_files
GROUP BY tablespace_name;
```

🔧 Scripts de Automatización Útiles

Backup Automatizado MySQL (Bash)

```
bash
#!/bin/bash
FECHA=$(date +%Y%m%d_%H%M%S)
BACKUP_DIR="/backups/mysql"
USUARIO="backup_user"
PASSWORD="password"

# Backup todas las bases de datos
mysqldump -u$USUARIO -p$PASSWORD --all-databases | gzip > $BACKUP_DIR/full_backup_$FECHA.sql.gz

# Eliminar backups antiguos (más de 30 días)
find $BACKUP_DIR -name "*.sql.gz" -mtime +30 -delete
```

Backup Automatizado SQL Server (PowerShell)

```
powershell
# backup_sql.ps1
$ServerInstance = "localhost"
$BackupPath = "C:\Backups\
$date = Get-Date -Format "yyyyMMdd_HHmmss"

Get-SqlDatabase -ServerInstance $ServerInstance | Where-Object {$_ .Name -ne 'tempdb'} | ForEach-Object {
```



```
$BackupFile = "$BackupPath\$($_.Name)_$Date.bak"  
Backup-SqlDatabase -ServerInstance $ServerInstance -Database $_.Name -Backup  
File $BackupFile -CompressionOption On  
}
```

⌚ Situaciones Donde la Consola es Esencial

Escenarios Críticos:

1. **Servidores remotos sin GUI** - SSH solamente
2. **Automation y scripts** - CI/CD pipelines
3. **Recuperación de desastres** - Cuando las herramientas GUI no están disponibles
4. **Performance tuning** - Comandos específicos de diagnóstico
5. **Bulk operations** - Operaciones masivas más eficientes

Ventajas de la Consola:

- ✓ Menor consumo de recursos
- ✓ Scripting y automatización
- ✓ Acceso en servidores headless
- ✓ Mayor control y precisión
- ✓ Logs detallados de operaciones

Desventajas:

- ✗ Curva de aprendizaje más pronunciada
- ✗ Fácil cometer errores críticos
- ✗ Visualización limitada de datos
- ✗ Menos intuitivo para principiantes



Recomendaciones de Seguridad en Consola

bash

```
# MySQL - Evitar passwords en Línea de comandos (usar archivo de opciones)
mysql --defaults-extra-file=~/my.cnf
```

```
# SQL Server - Usar trusted connections cuando sea posible
```

```
sqlcmd -S localhost -E
```

```
# Oracle - Usar wallet para credentials
```

```
sqlplus /@connection_string
```

► Práctica Guiada: Sistema de Reservas Hoteleras



Descripción del Proyecto

Implementaremos una base de datos para un sistema de reservas hoteleras con seguridad, automatización de backups y configuración segura.



Paso 1: Creación de la Base de Datos y Estructura

1.1 Conectarnos a MySQL

bash

```
# Conexión segura usando archivo de opciones (lo configuraremos después)
mysql --defaults-extra-file=~/my.cnf
```

1.2 Crear la Base de Datos

sql

```
-- Crear base de datos
```

```
CREATE DATABASE hotel_reservas;
```



```
USE hotel_reservas;
```

-- Tabla de hoteles

```
CREATE TABLE hoteles (
    hotel_id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    direccion VARCHAR(200),
    telefono VARCHAR(20),
    email VARCHAR(100),
    estrellas TINYINT,
    activo BOOLEAN DEFAULT TRUE,
    fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

-- Tabla de habitaciones

```
CREATE TABLE habitaciones (
    habitacion_id INT AUTO_INCREMENT PRIMARY KEY,
    hotel_id INT NOT NULL,
    numero_habitacion VARCHAR(10) NOT NULL,
    tipo ENUM('Individual', 'Doble', 'Suite', 'Familiar') NOT NULL,
    precio_noche DECIMAL(10,2) NOT NULL,
    capacidad TINYINT NOT NULL,
    descripcion TEXT,
    disponible BOOLEAN DEFAULT TRUE,
    FOREIGN KEY (hotel_id) REFERENCES hoteles(hotel_id) ON DELETE CASCADE,
    UNIQUE KEY unique_habitacion_hotel (hotel_id, numero_habitacion)
);
```

-- Tabla de clientes

```
CREATE TABLE clientes (
    cliente_id INT AUTO_INCREMENT PRIMARY KEY,
    dni VARCHAR(20) UNIQUE NOT NULL,
    nombre VARCHAR(100) NOT NULL,
    apellidos VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    telefono VARCHAR(20),
    pais VARCHAR(50),
    fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
```



```
);
```

-- Tabla de reservas

```
CREATE TABLE reservas (
    reserva_id INT AUTO_INCREMENT PRIMARY KEY,
    cliente_id INT NOT NULL,
    habitacion_id INT NOT NULL,
    fecha_entrada DATE NOT NULL,
    fecha_salida DATE NOT NULL,
    adultos TINYINT DEFAULT 1,
    ninos TINYINT DEFAULT 0,
    estado ENUM('Pendiente', 'Confirmada', 'En curso', 'Completada', 'Cancelada'
) DEFAULT 'Pendiente',
    total DECIMAL(10,2) NOT NULL,
    observaciones TEXT,
    fecha_reserva TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (cliente_id) REFERENCES clientes(cliente_id) ON DELETE CASCADE,
    FOREIGN KEY (habitacion_id) REFERENCES habitaciones(habitacion_id) ON DELETE
CASCADE,
    CHECK (fecha_salida > fecha_entrada)
);
```

-- Tabla de pagos

```
CREATE TABLE pagos (
    pago_id INT AUTO_INCREMENT PRIMARY KEY,
    reserva_id INT NOT NULL,
    monto DECIMAL(10,2) NOT NULL,
    metodo ENUM('Tarjeta', 'Efectivo', 'Transferencia', 'Online') NOT NULL,
    estado ENUM('Pendiente', 'Completado', 'Fallido', 'Reembolsado') DEFAULT 'Pe
ndiente',
    fecha_pago TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    transaccion_id VARCHAR(100),
    FOREIGN KEY (reserva_id) REFERENCES reservas(reserva_id) ON DELETE CASCADE
);
```

-- Tabla de auditoría

```
CREATE TABLE auditoria_reservas (
    auditoria_id INT AUTO_INCREMENT PRIMARY KEY,
```



```
reserva_id INT NOT NULL,  
accion ENUM('INSERT', 'UPDATE', 'DELETE') NOT NULL,  
usuario VARCHAR(100) NOT NULL,  
fecha_auditoria TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
datos_anteriores JSON,  
datos_nuevos JSON  
);
```

1.3 Insertar Datos de Ejemplo

```
sql  
-- Insertar hoteles  
INSERT INTO hoteles (nombre, direccion, telefono, email, estrellas) VALUES  
('Grand Palace Hotel', 'Av. Principal 123, Ciudad', '+34 912 345 678', 'info@grandpalace.com', 5),  
('Sea View Resort', 'Playa del Sol 45, Costa', '+34 913 456 789', 'reservas@seaview.com', 4),  
('Mountain Lodge', 'Sierra Nevada 78, Montaña', '+34 914 567 890', 'contact@mountainlodge.com', 3);  
  
-- Insertar habitaciones  
INSERT INTO habitaciones (hotel_id, numero_habitacion, tipo, precio_noche, capacidad, descripcion) VALUES  
(1, '101', 'Individual', 80.00, 1, 'Habitación individual con vista al jardín'),  
(1, '102', 'Doble', 120.00, 2, 'Habitación doble con cama king size'),  
(1, '201', 'Suite', 250.00, 3, 'Suite ejecutiva con sala de estar'),  
(2, '101', 'Individual', 70.00, 1, 'Habitación con vista al mar'),  
(2, '102', 'Familiar', 180.00, 4, 'Habitación familiar con dos camas dobles'),  
(3, '101', 'Doble', 90.00, 2, 'Habitación doble con chimenea');  
  
-- Insertar clientes  
INSERT INTO clientes (dni, nombre, apellidos, email, telefono, pais) VALUES  
('12345678A', 'María', 'González Pérez', 'maria.gonzalez@email.com', '+34 600 111 222', 'España'),  
('87654321B', 'Carlos', 'Rodríguez López', 'carlos.rodriguez@email.com', '+34 600 333 444', 'España'),
```



```
('11223344C', 'Anna', 'Smith Johnson', 'anna.smith@email.com', '+44 7700 123456'  
, 'Reino Unido');
```

👤 Paso 2: Gestión de Usuarios y Privilegios

2.1 Crear Usuarios Específicos por Rol

```
sql  
-- Crear usuario para aplicación web (privilegios limitados)  
CREATE USER 'app_hotel'@'localhost' IDENTIFIED BY 'AppHotel2024!';  
CREATE USER 'app_hotel'@'%' IDENTIFIED BY 'AppHotel2024!';  
  
-- Crear usuario para reportes (solo Lectura)  
CREATE USER 'reportes_hotel'@'localhost' IDENTIFIED BY 'Reportes2024!';  
  
-- Crear usuario para administración  
CREATE USER 'admin_hotel'@'localhost' IDENTIFIED BY 'AdminHotel2024!';  
  
-- Crear usuario para backups  
CREATE USER 'backup_user'@'localhost' IDENTIFIED BY 'Backup2024!';
```

2.2 Asignar Privilegios por Rol

```
sql  
-- Privilegios para aplicación web  
GRANT SELECT, INSERT, UPDATE ON hotel_reservas.clientes TO 'app_hotel'@'localhost';  
GRANT SELECT, INSERT, UPDATE ON hotel_reservas.reservas TO 'app_hotel'@'localhost';  
GRANT SELECT, INSERT ON hotel_reservas.pagos TO 'app_hotel'@'localhost';  
GRANT SELECT ON hotel_reservas.hoteles TO 'app_hotel'@'localhost';  
GRANT SELECT ON hotel_reservas.habitaciones TO 'app_hotel'@'localhost';  
  
-- Privilegios para reportes (solo Lectura)
```



```
GRANT SELECT ON hotel_reservas.* TO 'reportes_hotel'@'localhost';

-- Privilegios para administración
GRANT ALL PRIVILEGES ON hotel_reservas.* TO 'admin_hotel'@'localhost';

-- Privilegios para backups
GRANT SELECT, RELOAD, LOCK TABLES, PROCESS ON *.* TO 'backup_user'@'localhost';
GRANT EVENT ON hotel_reservas.* TO 'backup_user'@'localhost';

-- Aplicar los cambios
FLUSH PRIVILEGES;
```

2.3 Verificar Privilegios

```
sql
-- Ver privilegios de un usuario
SHOW GRANTS FOR 'app_hotel'@'localhost';
SHOW GRANTS FOR 'reportes_hotel'@'localhost';

-- Ver usuarios creados
SELECT user, host FROM mysql.user WHERE user LIKE '%hotel%';
```

🔒 Paso 3: Configuración Segura con Archivo de Opciones

3.1 Crear Archivo de Configuración Seguro

```
bash
# Crear directorio seguro para configuración
sudo mkdir -p /etc/mysql/conf.d/
sudo chmod 700 /etc/mysql/conf.d/

# Crear archivo de opciones para el usuario admin
```



```
nano ~/.my.cnf
```

3.2 Contenido del Archivo .my.cnf

```
ini
[client]
user = admin_hotel
password = AdminHotel2024!
host = localhost
database = hotel_reservas
```

```
[mysql]
prompt = "\u00u@\u00h [\u00d]> "
auto-rehash
safe-updates
```

```
[mysqldump]
quick
single-transaction
compress
```

3.3 Configurar Permisos del Archivo

```
bash
# Hacer el archivo solo readable por el usuario
chmod 600 ~/.my.cnf

# Verificar que funciona la conexión sin password
mysql --defaults-extra-file=~/my.cnf

# Probar conexión directa
mysql # Debería conectarse automáticamente con las credenciales
```



■ Paso 4: Automatización de Backups

4.1 Crear Script de Backup Completo

```
bash
# Crear directorio para backups
sudo mkdir -p /backups/mysql
sudo chown mysql:mysql /backups/mysql
sudo chmod 755 /backups/mysql

# Crear script de backup
sudo nano /usr/local/bin/backup_hotel.sh
```

4.2 Script de Backup Automatizado (/usr/local/bin/backup_hotel.sh)

```
bash
#!/bin/bash

# Configuración
BACKUP_DIR="/backups/mysql"
DATE=$(date +%Y%m%d_%H%M%S)
RETENTION_DAYS=7
MYSQL_CNF="/root/.my.cnf"
LOG_FILE="/var/log/mysql_backup.log"

# Funciones de Logging
log() {
    echo "$(date '+%Y-%m-%d %H:%M:%S') - $1" >> $LOG_FILE
}

# Verificar directorio de backup
if [ ! -d "$BACKUP_DIR" ]; then
    mkdir -p $BACKUP_DIR
fi
```



```
# Iniciar backup
log "Iniciando backup de bases de datos..."

# Backup completo de todas las bases de datos
BACKUP_FILE="$BACKUP_DIR/full_backup_${DATE}.sql.gz"
mysqldump --defaults-extra-file=$MYSQL_CNF --all-databases --single-transaction
--routines --triggers --events | gzip > $BACKUP_FILE

if [ $? -eq 0 ]; then
    log "Backup completado: $BACKUP_FILE"

    # Backup específico de hotel_reservas (adicional)
    BACKUP_HOTEL="$BACKUP_DIR/hotel_reservas_${DATE}.sql.gz"
    mysqldump --defaults-extra-file=$MYSQL_CNF --single-transaction --routines h
otel_reservas | gzip > $BACKUP_HOTEL

    if [ $? -eq 0 ]; then
        log "Backup hotel_reservas completado: $BACKUP_HOTEL"
    else
        log "ERROR en backup específico de hotel_reservas"
    fi
else
    log "ERROR en backup completo"
    exit 1
fi

# Limpiar backups antiguos
log "Limpiando backups antiguos (más de $RETENTION_DAYS días)..."
find $BACKUP_DIR -name "*.sql.gz" -mtime +$RETENTION_DAYS -delete

# Verificar espacio en disco
DISK_USAGE=$(df -h $BACKUP_DIR | awk 'NR==2 {print $5}')
log "Espacio en disco utilizado: $DISK_USAGE"

log "Proceso de backup finalizado"
```



4.3 Hacer Ejecutable el Script

```
bash
sudo chmod +x /usr/local/bin/backup_hotel.sh

# Probar el script manualmente
sudo /usr/local/bin/backup_hotel.sh

# Verificar que se crearon los backups
ls -la /backups/mysql/
```

4.4 Configurar Cron para Backups Automatizados

```
bash
# Editar crontab
sudo crontab -e

# Agregar las siguientes líneas:
# Backup completo diario a las 2:00 AM
0 2 * * * /usr/local/bin/backup_hotel.sh

# Backup incremental cada 6 horas (solo de hotel_reservas)
0 */6 * * * /usr/local/bin/backup_incremental.sh

# Verificar crontab
sudo crontab -l
```

4.5 Script de Backup Incremental

```
bash
sudo nano /usr/local/bin/backup_incremental.sh

bash
#!/bin/bash

# Backup incremental para hotel_reservas
BACKUP_DIR="/backups/mysql/incremental"
DATE=$(date +%Y%m%d_%H%M%S)
```



```
MYSQL_CNF="/root/.my.cnf"

# Crear directorio si no existe
mkdir -p $BACKUP_DIR

# Backup de tablas específicas que cambian frecuentemente
mysqldump --defaults-extra-file=$MYSQL_CNF \
    hotel_reservas reservas pagos auditoria_reservas \
    --where="fecha_reserva >= DATE_SUB(NOW(), INTERVAL 6 HOUR)" | gzip > $BACKUP
_DIR/incremental_$date.sql.gz

# Mantener solo los últimos 50 backups incrementales
ls -t $BACKUP_DIR/incremental_*.sql.gz | tail -n +51 | xargs -r rm
bash
sudo chmod +x /usr/local/bin/backup_incremental.sh
```

Q Paso 5: Verificación y Pruebas

5.1 Probar Conexiones con Diferentes Usuarios

```
bash
# Probar conexión como app_hotel
mysql -u app_hotel -p -e "SHOW DATABASES;"

# Probar conexión como reportes_hotel
mysql -u reportes_hotel -p -e "USE hotel_reservas; SHOW TABLES;"

# Probar conexión segura con archivo .my.cnf
mysql --defaults-extra-file=~/my.cnf -e "SELECT USER(), CURRENT_USER();"
```



5.2 Verificar Privilegios en Acción

```
sql
-- Conectarse como app_hotel y probar operaciones
-- (Esto debería funcionar)
INSERT INTO hotel_reservas.clientes (dni, nombre, apellidos, email)
VALUES ('99999999X', 'Test', 'User', 'test@email.com');

-- (Esto debería fallar - no tiene permisos DELETE)
DELETE FROM hotel_reservas.clientes WHERE dni = '99999999X';
```

5.3 Probar Backup y Restore

```
bash
# Probar restauración en base de datos de prueba
mysql -e "CREATE DATABASE hotel_reservas_test;"
zcat /backups/mysql/hotel_reservas_*.sql.gz | mysql hotel_reservas_test

# Verificar que los datos se restauraron correctamente
mysql -e "USE hotel_reservas_test; SHOW TABLES; SELECT COUNT(*) FROM clientes;"
```

III Paso 6: Monitoreo y Auditoría

6.1 Crear Vista para Reportes

```
sql
-- Vista para reportes de ocupación
CREATE VIEW vista_ocupacion AS
SELECT
    h.nombre AS hotel,
    hab.tipo_habitacion,
    COUNT(r.reserva_id) AS reservas_activas,
    AVG(hab.precio_noche) AS precio_promedio
FROM hoteles h
```



```
LEFT JOIN habitaciones hab ON h.hotel_id = hab.hotel_id
LEFT JOIN reservas r ON hab.habitacion_id = r.habitacion_id
AND r.estado IN ('Confirmada', 'En curso')
GROUP BY h.hotel_id, hab.tipo_habitacion;

-- Vista para ingresos
CREATE VIEW vista_ingresos AS
SELECT
    DATE(r.fecha_reserva) AS fecha,
    h.nombre AS hotel,
    SUM(p.monto) AS ingresos_totales,
    COUNT(DISTINCT r.reserva_id) AS total_reservas
FROM reservas r
JOIN habitaciones hab ON r.habitacion_id = hab.habitacion_id
JOIN hoteles h ON hab.hotel_id = h.hotel_id
JOIN pagos p ON r.reserva_id = p.reserva_id
WHERE p.estado = 'Completado'
GROUP BY DATE(r.fecha_reserva), h.hotel_id;
```

6.2 Trigger para Auditoría

```
sql
-- Trigger para auditar cambios en reservas
DELIMITER //
CREATE TRIGGER audit_reservas_update
AFTER UPDATE ON reservas
FOR EACH ROW
BEGIN
    INSERT INTO auditoria_reservas (reserva_id, accion, usuario, datos_anteriores, datos_nuevos)
    VALUES (
        OLD.reserva_id,
        'UPDATE',
        USER(),
        JSON_OBJECT(
            'estado', OLD.estado,
            'total', OLD.total,
```



```
'observaciones', OLD.observaciones
),
JSON_OBJECT(
    'estado', NEW.estado,
    'total', NEW.total,
    'observaciones', NEW.observaciones
)
);
END//  
DELIMITER ;
```

⌚ Resumen de la Implementación

✓ Lo que hemos implementado:

1. **Base de datos completa** para reservas hoteleras
2. **Usuarios y roles** con privilegios específicos
3. **Configuración segura** con archivo .my.cnf
4. **Sistema de backups** automatizado y incremental
5. **Mecanismos de auditoría** y monitoreo
6. **Vistas para reportes** y análisis

🔒 Mejoras de Seguridad Implementadas:

- Usuarios con privilegios mínimos necesarios
- Conexiones seguras mediante archivo de opciones
- Backups automatizados con retención controlada
- Auditoría de cambios críticos



🔗 Próximos Pasos Sugeridos:

- Implementar replicación para alta disponibilidad
- Configurar SSL para conexiones seguras
- Establecer políticas de retención de datos
- Implementar monitoring con alertas

💻 Ajustes para Windows

1. Configuración de Archivos y Rutas

Archivo de Opciones (.my.cnf) en Windows

```
ini
# En Windows, el archivo se llama my.ini y puede estar en varias ubicaciones:
# C:\ProgramData\MySQL\MySQL Server 8.0\my.ini
# C:\Users\[Usuario]\my.ini
# O crear my.cnf en C:\

[client]
user=admin_hotel
password=AdminHotel2024!
host=localhost
database=hotel_reservas

[mysql]
prompt="\u@\h [\d]> "
auto-rehash
safe-updates

[mysqldump]
quick
single-transaction
compress
```



Ubicaciones comunes en Windows:

```
cmd  
# Para MySQL instalado con instalador oficial  
C:\ProgramData\MySQL\MySQL Server 8.0\my.ini  
  
# Para XAMPP/WAMP  
C:\xampp\mysql\bin\my.ini  
C:\wamp\bin\mysql\mysql[version]\my.ini  
  
# Archivo de usuario  
C:\Users\%USERNAME%\my.ini
```

2. Scripts de Backup para Windows

Script Batch para Backup (backup_hotel.bat)

```
batch  
@echo off  
setlocal enabledelayedexpansion  
  
REM Configuración  
set BACKUP_DIR=C:\Backups\MySQL  
set DATE=%DATE:/=-%  
set TIME=%TIME::=-%  
set DATETIME=%DATE%_%TIME%  
set MYSQL_PATH="C:\Program Files\MySQL\MySQL Server 8.0\bin"  
set LOG_FILE=C:\Backups\MySQL\backup_log.txt  
  
REM Crear directorio de backups  
if not exist "%BACKUP_DIR%" mkdir "%BACKUP_DIR%"  
  
REM Log inicio  
echo %DATE% %TIME% - Iniciando backup >> %LOG_FILE%  
  
REM Backup completo de todas las bases de datos
```



```
"%MYSQL_PATH%\mysqldump.exe" --defaults-extra-file=C:\my.cnf --all-databases --single-transaction --routines --triggers --events > "%BACKUP_DIR%\full_backup_%DATETIME%.sql"

if !errorlevel! equ 0 (
    echo %DATE% %TIME% - Backup completo exitoso >> %LOG_FILE%

    REM Comprimir con PowerShell
    powershell -command "& {Get-Content '%BACKUP_DIR%\full_backup_%DATETIME%.sql'
    ' | Set-Content '%BACKUP_DIR%\full_backup_%DATETIME%.sql.txt'}"
    del "%BACKUP_DIR%\full_backup_%DATETIME%.sql"
) else (
    echo %DATE% %TIME% - ERROR en backup completo >> %LOG_FILE%
)

REM Backup específico de hotel_reservas
"%MYSQL_PATH%\mysqldump.exe" --defaults-extra-file=C:\my.cnf --single-transaction --routines hotel_reservas > "%BACKUP_DIR%\hotel_reservas_%DATETIME%.sql"

if !errorlevel! equ 0 (
    echo %DATE% %TIME% - Backup hotel_reservas exitoso >> %LOG_FILE%
    powershell -command "& {Get-Content '%BACKUP_DIR%\hotel_reservas_%DATETIME%.sql'
    ' | Set-Content '%BACKUP_DIR%\hotel_reservas_%DATETIME%.sql.txt'}"
    del "%BACKUP_DIR%\hotel_reservas_%DATETIME%.sql"
) else (
    echo %DATE% %TIME% - ERROR en backup hotel_reservas >> %LOG_FILE%
)

REM Limpiar backups antiguos (más de 7 días)
forfiles /p "%BACKUP_DIR%" /m "*.sql.txt" /d -7 /c "cmd /c del @path"

echo %DATE% %TIME% - Proceso de backup finalizado >> %LOG_FILE%
```

Script PowerShell Más Avanzado (backup_hotel.ps1)

```
powershell
# backup_hotel.ps1
param(
```



```
[string]$BackupDir = "C:\Backups\MySQL",
[string]$MyCnf = "C:\my.cnf",
[int]$RetentionDays = 7
)

# Configuraciòn
$Date = Get-Date -Format "yyyyMMdd_HH:mm:ss"
$MySQLPath = "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqldump.exe"
$LogFile = "$BackupDir\backup_log.txt"

# Crear directorio si no existe
if (!(Test-Path $BackupDir)) {
    New-Item -ItemType Directory -Path $BackupDir -Force
}

# Funciòn de Logging
function Write-Log {
    param([string]$Message)
    $Timestamp = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
    "$Timestamp - $Message" | Out-File -FilePath $LogFile -Append
    Write-Host $Message
}

# Iniciar backup
Write-Log "Iniciando proceso de backup..."

try {
    # Backup completo
    $FullBackup = "$BackupDir\full_backup_$Date.sql"
    & $MySQLPath --defaults-extra-file=$MyCnf --all-databases --single-transaction --routines --triggers --events --result-file=$FullBackup

    if ($LASTEXITCODE -eq 0) {
        Write-Log "Backup completo exitoso: $FullBackup"

        # Comprimir
        Compress-Archive -Path $FullBackup -DestinationPath "$FullBackup.zip"
        Remove-Item $FullBackup
```



```
    } else {
        throw "Error en backup completo"
    }

    # Backup específico de hotel_reservas
    $HotelBackup = "$BackupDir\hotel_reservas_$Date.sql"
    & $MysqlPath --defaults-extra-file=$MyCnf --single-transaction --routines hotel_reservas --result-file=$HotelBackup

    if ($LASTEXITCODE -eq 0) {
        Write-Log "Backup hotel_reservas exitoso: $HotelBackup"
        Compress-Archive -Path $HotelBackup -DestinationPath "$HotelBackup.zip"
        Remove-Item $HotelBackup
    } else {
        throw "Error en backup hotel_reservas"
    }

    # Limpiar backups antiguos
    Get-ChildItem -Path $BackupDir -Filter "*.zip" |
        Where-Object { $_.LastWriteTime -lt (Get-Date).AddDays(-$RetentionDays)}
    } |
        Remove-Item -Force

    Write-Log "Proceso de backup completado exitosamente"

} catch {
    Write-Log "ERROR: $($_.Exception.Message)"
    exit 1
}
```



3. Automatización con Task Scheduler (Reemplazo de Cron)

Configuración en Task Scheduler:

1. Abrir Task Scheduler:

- Windows + R → taskschd.msc
- buscar "Programador de tareas"

2. Crear tarea básica:

- **Nombre:** "MySQL Backup Diario"
- **Desencadenador:** Diariamente a las 2:00 AM
- **Acción:** Iniciar un programa
- **Programa:** C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
- **Argumentos:** -ExecutionPolicy Bypass -File "C:\Scripts\backup_hotel.ps1"

Comando para crear tarea desde línea de comandos:

```
cmd
schtasks /create /tn "MySQL Backup Diario" /tr "powershell -ExecutionPolicy Bypass -File \"C:\Scripts\backup_hotel.ps1\"" /sc daily /st 02:00 /ru SYSTEM
```

4. Gestión de Servicios MySQL en Windows

Comandos de Servicio:

```
cmd
# Ver estado del servicio MySQL
sc query MySQL80

# Iniciar servicio
net start MySQL80

# Detener servicio
net stop MySQL80
```



```
# Reiniciar servicio  
net stop MySQL80 && net start MySQL80
```

```
# Configurar inicio automático  
sc config MySQL80 start=auto
```

Ubicación de datos MySQL:

```
cmd  
# Datos por defecto MySQL Installer  
C:\ProgramData\MySQL\MySQL Server 8.0\Data\
```

```
# En XAMPP  
C:\xampp\mysql\data\  
  
# En WAMP  
C:\wamp\bin\mysql\mysql[version]\data\
```

5. Configuración de Permisos de Archivos en Windows

Permisos para directorio de backups:

```
cmd  
# Crear directorio  
mkdir C:\Backups\MySQL  
  
# Dar permisos al servicio MySQL (si corre como servicio específico)  
icacls "C:\Backups\MySQL" /grant "NT SERVICE\MySQL80:(OI)(CI)F"  
  
# O dar permisos al usuario SYSTEM  
icacls "C:\Backups\MySQL" /grant "SYSTEM:(OI)(CI)F"  
  
# O para todos los usuarios (menos seguro)  
icacls "C:\Backups\MySQL" /grant "Todos:(OI)(CI)RW"
```

Protección del archivo my.cnf:

```
cmd  
# Mover a ubicación segura
```



```
move my.cnf C:\ProgramData\MySQL\  
  
# Establecer permisos restrictivos  
icacls "C:\ProgramData\MySQL\my.cnf" /inheritance:r  
icacls "C:\ProgramData\MySQL\my.cnf" /grant:r "Administradores:F" "SYSTEM:F"
```

6. Conexión y Verificación en Windows

Comandos de conexión:

```
cmd  
# Con archivo de configuración  
mysql --defaults-extra-file=C:\my.cnf  
  
# Con parámetros directos  
mysql -u admin_hotel -pAdminHotel2024! -h localhost  
  
# Con MySQL Workbench (GUI)  
# Descargar desde: https://dev.mysql.com/downloads/workbench/
```

Verificar instalación:

```
cmd  
# Ver versión de MySQL  
mysql --version  
  
# Ver servicios MySQL  
sc query | findstr MySQL  
  
# Ver procesos MySQL  
tasklist | findstr mysql
```

7. Solución de Problemas Comunes en Windows

Problemas de permisos:

```
cmd  
# Ejecutar como administrador
```



```
# Click derecho → "Ejecutar como administrador"

# Ver permisos de archivo
icacls "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld.exe"

# Reparar permisos
icacls "C:\Backups\MySQL" /reset
icacls "C:\Backups\MySQL" /grant "Todos:(OI)(CI)RW"
```

Problemas de firewall:

```
cmd
# Agregar excepción para MySQL
netsh advfirewall firewall add rule name="MySQL" dir=in action=allow program="C:
\Program Files\MySQL\MySQL Server 8.0\bin\mysqld.exe" enable=yes
```

8. Script de Instalación/Configuración Automática

setup_mysql_windows.bat:

```
batch
@echo off
echo Configurando MySQL para sistema de reservas hoteleras...

REM Crear directorios
mkdir C:\Backups\MySQL 2>nul
mkdir C:\Scripts 2>nul

REM Copiar scripts
copy backup_hotel.ps1 C:\Scripts\ 2>nul
copy backup_hotel.bat C:\Scripts\ 2>nul

REM Crear archivo de configuración
echo [client] > C:\my.cnf
echo user=admin_hotel >> C:\my.cnf
echo password=AdminHotel2024! >> C:\my.cnf
echo host=localhost >> C:\my.cnf
echo database=hotel_reservas >> C:\my.cnf
```



```
REM Configurar permisos
icacls "C:\Backups\MySQL" /grant "Todos:(OI)(CI)RW"
icacls "C:\my.cnf" /inheritance:r
icacls "C:\my.cnf" /grant:r "Administradores:F"

echo Configuracion completada!
echo Archivos creados en:
echo - C:\Scripts\backup_hotel.ps1
echo - C:\Scripts\backup_hotel.bat
echo - C:\my.cnf
echo.
echo Recuerda programar la tarea en Task Scheduler!
pause
```

📝 Diferencias Clave Windows vs Linux

Aspecto	Linux	Windows
Archivo config	~/.my.cnf	



Bibliografía Recomendada

- "Fundamentos de Bases de Datos" de Abraham Silberschatz, Henry F. Korth y S. Sudarshan
- "Sistemas de Bases de Datos: un enfoque práctico" de Thomas M. Connolly y Carolyn Begg
- "Desarrollo de Bases de Datos: casos prácticos desde el análisis a la implementación" de Dolores Cuadra, Elena Castro, Ana M. Iglesias
- "Tecnología y Diseño de Bases de Datos" de Marcos, C. Calero y B. Vela
- <https://docs.oracle.com/en/database/>