



UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

UNITEC

Documentación proyecto del primer parcial

RDBMS para MySQL

Teoría de base de datos 2

Entrega: 23 de feb. de 26

San Pedro Sula, Cortés

Docente: Ing. Elvin Deras

Estudiante:

Jorge Alejandro Discua Morán – 22311096

1. Tabla de contenido

1.	Tabla de contenido	2
2.	Introducción.....	3
3.	Objetivos.....	3
	Objetivo General.....	3
	Objetivos específicos	3
4.	Instalación de la base de datos.....	4
4.1	Instalación de MySQL	4
4.2	Creación de Base de Datos de Prueba.....	4
4.3	Instalación de Dependencias en Python.....	4
4.4	Verificación de Conexión.....	5
5.	Tecnologías utilizadas.....	5
6.	Alcance del sistema.....	5
7.	Arquitectura del sistema	6
8.	Descripción de funcionalidades	6
6.1	Login	6
6.2	Explorador de objetos	6
6.3	Visualización de datos.....	6
6.4	Editor SQL.....	6
6.5	Creación visual de tablas	7
6.6	Creación visual de vistas.....	7
6.7	Generación de DDL	7
9.	Uso de comandos del motor.....	7
10.	Limitaciones.....	8
11.	Conclusiones.....	8

2. Introducción

El presente documento describe el desarrollo de una herramienta administrativa de bases de datos (RDBMS) implementada en Python, cuyo propósito es permitir la exploración, administración y manipulación básica de objetos de una base de datos MySQL.

La aplicación fue desarrollada como proyecto final de la asignatura Teoría de Base de Datos II, cumpliendo las restricciones establecidas, tales como el uso de SQL directo, la no utilización de ORMs y la no dependencia del esquema `information_schema` para la obtención de metadata principal.

El sistema proporciona una interfaz gráfica ‘amigable’ y sencilla que facilita al usuario interactuar con tablas, vistas, procedimientos almacenados, funciones y disparadores, así como ejecutar sentencias SQL desde un editor integrado.

3. Objetivos

Objetivo General

Desarrollar una aplicación de escritorio que funcione como gestor básico de bases de datos MySQL, permitiendo la administración y visualización de objetos mediante SQL directo.

Objetivos específicos

- Implementar un sistema de conexión y autenticación a MySQL.
- Visualizar los principales objetos de una base de datos.
- Permitir la creación de tablas y vistas mediante interfaz gráfica.
- Generar y mostrar el DDL de los objetos existentes.
- Integrar un editor SQL para la ejecución de consultas.

4. Instalación de la base de datos

Para el desarrollo y ejecución del sistema se requirió la instalación y configuración del sistema gestor de base de datos MySQL, así como del entorno de desarrollo en Python.

4.1 Instalación de MySQL

Se utilizó MySQL versión 8.

El proceso de instalación fue el siguiente:

1. Descarga del instalador oficial desde el sitio web de MySQL.
2. Ejecución del asistente de instalación.
3. Selección del tipo de instalación “Server Only” o “Developer Default”. Personalmente recomiendo incluir también la instalación de MySQL Workbench.
4. Configuración del puerto por defecto (3306).
5. Creación del usuario root y asignación de contraseña.
6. Inicio automático del servicio MySQL.

4.2 Creación de Base de Datos de Prueba

Para realizar las pruebas del sistema se creó una base de datos específica:

```
CREATE DATABASE tbd2_python;
USE tbd2_python;
```

En dicha base de datos se crearon tablas, vistas, procedimientos, funciones y triggers utilizados durante las pruebas del gestor desarrollado.

4.3 Instalación de Dependencias en Python

Se utilizó Python 3.13 junto con la librería mysql-connector-python versión 8.0.24.

Instalación mediante:

```
pip install mysql-connector-python==8.0.24
```

Se utilizó entorno virtual para aislar dependencias.

4.4 Verificación de Conexión

La conexión entre la aplicación y MySQL se realizó utilizando:

- Host: localhost
- Puerto: 3306
- Usuario: root (o usuario configurado)
- Base de datos: tbd2_python

Se validó la conexión ejecutando consultas de prueba desde la aplicación.

5. Tecnologías utilizadas

- Lenguaje de programación: Python 3.13
- Interfaz gráfica: Tkinter
- Conector: mysql-connector-python 8.0.24
- Sistema gestor de base de datos: MySQL 8.
- Entorno de desarrollo: PyCharm

6. Alcance del sistema

El sistema permite administrar los siguientes objetos:

- Tablas
- Vistas
- Procedimientos almacenados
- Funciones
- Triggers

- Índices (listado)
- Usuarios (listado)

Objetos como Packages, Sequences o Tablespaces no se implementan debido a que MySQL no los maneja de la misma forma que otros SGBD como Oracle.

7. Arquitectura del sistema

La aplicación se organiza en módulos:

- login.py: Maneja autenticación y conexión.
- main_window.py: Controla la interfaz principal.
- metadata.py: Contiene funciones para obtener información del motor.
- main.py: Punto de entrada del sistema.

Flujo general:

Login → Validación → Ventana principal → Interacción con objetos

Se utiliza una arquitectura modular para separar responsabilidades.

8. Descripción de funcionalidades

6.1 Login

Permite ingresar host, usuario, contraseña y base de datos.

6.2 Explorador de objetos

Se presenta un árbol lateral con los objetos de la base de datos.

6.3 Visualización de datos

Al seleccionar una tabla o vista, se muestran los registros en forma tabular.

6.4 Editor SQL

Área donde el usuario puede escribir y ejecutar sentencias SQL.

6.5 Creación visual de tablas

Interfaz que permite definir columnas, tipos de datos, PK y NOT NULL.

6.6 Creación visual de vistas

Formulario para definir nombre y consulta SELECT.

6.7 Generación de DDL

El sistema puede mostrar el DDL de:

- Tablas
- Vistas
- Procedures
- Functions
- Triggers

9. Uso de comandos del motor

El sistema utiliza comandos como:

- SHOW TABLES
- SHOW FULL TABLES
- SHOW CREATE TABLE
- SHOW CREATE VIEW
- SHOW CREATE PROCEDURE
- SHOW CREATE FUNCTION
- SHOW CREATE TRIGGER

- SHOW PROCEDURE STATUS
- SHOW FUNCTION STATUS

No se utiliza information_schema.

10. Limitaciones

- No se implementa administración avanzada de usuarios.
- No se crean índices ni triggers de forma gráfica.
- La herramienta es educativa, no empresarial.

11. Conclusiones

El desarrollo de esta herramienta permitió comprender de forma práctica el funcionamiento interno de un gestor de bases de datos y el uso de SQL directo para administrar objetos.

El proyecto cumple con los requisitos planteados y demuestra la aplicación de conceptos fundamentales de bases de datos y programación.