**Database Management Assignment: Sakila Database in MySQL Workbench**

# Objective

This assignment aims to enhance your understanding of database management by performing various operations on the **Sakila** database using **MySQL Workbench**. You will modify, insert, delete, and update records, design complex queries, work with transactions, and ensure data integrity and consistency.

# Assignment Tasks

## 1. Identifying Tools and Statements for Modifying Database Content

* Research and document different **SQL statements** used to modify database content (INSERT, UPDATE, DELETE, ALTER, etc.).

**INSERT INTO:** used for adding new records in tables, we specify table and/or columns and then VALUES.

**UPDATE:** modifies existing records in a table, we use a SET and the reference of the data with WHERE.

**DELETE:** removes records from table, WHERE as the ref.

**ALTER TABLE:** modifies the structure of the tables adding, deleting or modifying the columns and constraints of the table with ADD, DROP and MODIFY COLUMN.

**TRUNCATE TABLE:** empties a table completely but mantains its structure.

* Describe the functionalities of MySQL Workbench tools such as **SQL Editor, Schema Inspector, and Query Builder**.

**SQL Editor:** Used to write and execute queries.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**Schema Inspector:** Helps with database structure, indexes and constraint analisis.

Access via schemas panel.

Tabla

El contenido generado por IA puede ser incorrecto.

**Query Builder:** A visual database builder, provides a graphic way of editing databases easily.

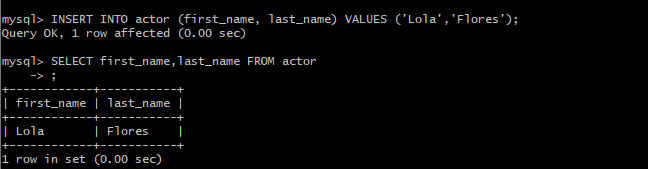
Imagen que contiene Diagrama

El contenido generado por IA puede ser incorrecto.

* **Deliverable:** A section in the final report summarizing SQL statements and MySQL Workbench tools.

## 2. Data Insertion, Deletion, and Update

* Using the **actor** table, insert a new record with fictitious data.



* Update an existing record by changing the last name of an actor.

Texto

El contenido generado por IA puede ser incorrecto.

* Delete an actor from the table.

Una captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

* **Deliverable:** Provide the executed SQL statements and their results.

○ **SQL File:** 02\_modify\_actor.sql

○ **Screenshots:** 02\_modify\_actor\_screenshots/

## 3. Creating a Table from a Query Result

* Execute a query to retrieve all movies released after 2005 from the **film** table.

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

* Store the result in a new table called **recent\_films**.

Texto

El contenido generado por IA puede ser incorrecto.

* **Deliverable:** The SQL script used and a screenshot of the newly created table.

○ **SQL File:** 03\_create\_recent\_films.sql

○ **Screenshot:** 03\_recent\_films\_screenshot.png

## 4. Designing Complex SQL Scripts

* Write an SQL script that:

○ Lists all customers who have rented a film in the last 30 days.

○ Identifies the most rented film in the database.

○ Displays the total revenue generated per store.

* **Deliverable:** The SQL script with comments explaining each query. ○ **SQL File:** 04\_complex\_queries.sql

## 5. Understanding Transactions

* Explain transactions and their importance in database management.
* Using MySQL Workbench, perform a transaction that:
  1. Inserts a new rental record.

○ Updates the inventory to reflect the rental.

○ Commits the transaction.

* **Deliverable:** 
  1. **SQL File:** 05\_transaction\_example.sql

○ **Section in the final report covering transactions.**

## 6. Rolling Back Transactions

* Demonstrate a scenario where a transaction is partially executed but later rolled back due to an error (e.g., an attempt to rent a movie that is out of stock).
* **Deliverable:**

○ **SQL File:** 06\_rollback\_example.sql

○ **Section in the final report covering rollback transactions.**

## 7. Understanding Record Locking Policies

* Research and document different types of record-locking mechanisms (pessimistic vs. optimistic locking).
* Test a scenario where two users attempt to update the same record simultaneously.
* **Deliverable:**

○ **Section in the final report covering record locking policies.**

## 8. Ensuring Data Integrity and Consistency

* Identify potential data integrity issues in the **Sakila** database.
* Implement foreign key constraints and triggers to maintain data consistency.
* **Deliverable:**

○ **SQL File:** 08\_data\_integrity.sql

○ **Section in the final report covering data integrity.**

# Submission Requirements

* **Folder Structure:** 
  1. Deliverables/

■ SQL\_Scripts/ (Contains all .sql files)

■ Screenshots/ (Contains screenshots of query executions, if applicable)

■ Final\_Report/ (Contains a single consolidated report in PDF format)

* **Final Report:** All previously separate reports should now be consolidated into a single PDF file:
  1. **Filename:** Surname\_Name\_Final\_Report\_Sakila.pdf

○ **Folder:** Deliverables/Final\_Report/

○ The report must contain all research, explanations, query results, and screenshots.

**Notes:**

* Ensure that all SQL statements are tested before submission.
* The final report should include reflections on challenges faced and how they were overcome.
* Cite any external references used in your research.