

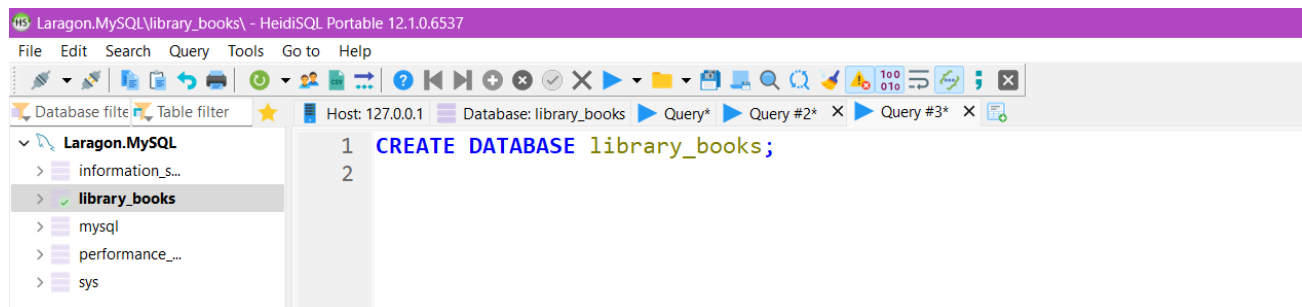
Activity 1: Database Design Challenge

Part #1

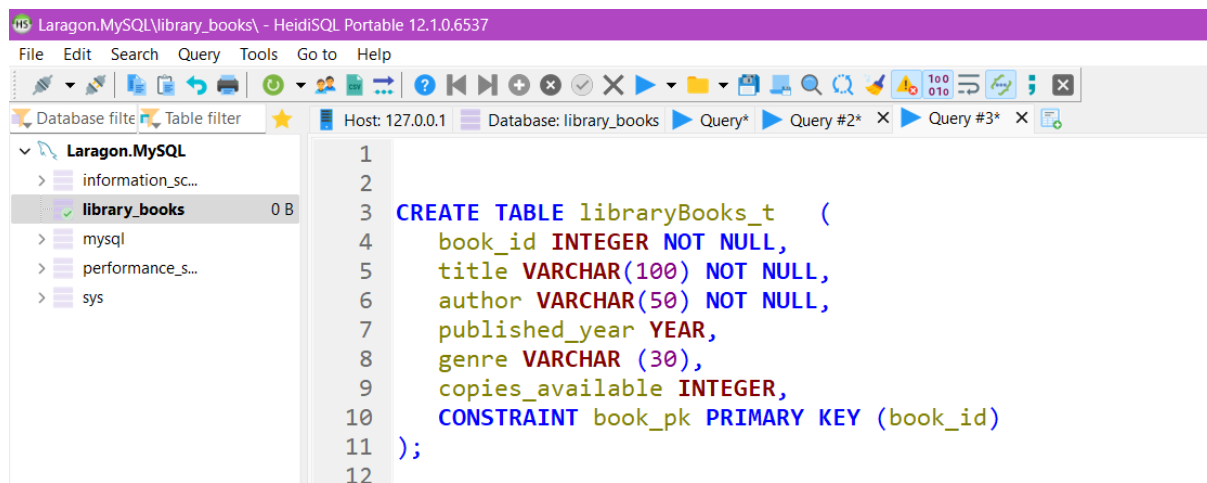
Instructions:

Design a database table for managing **library books**. Include the following requirements:

- Columns: book_id (Primary Key, Integer), title (VARCHAR(100)), author (VARCHAR(50)), published_year (YEAR), genre (VARCHAR(30)), copies_available (Integer).



- Write the SQL statement to create the table.



Columns: + Add - Remove ▲ Up ▼ Down							
#	Name	Datatype	Length/Set	Unsigned	Allow N...	Zerofill	Default
1	book_id	INT	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default
2	title	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default
3	author	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default
4	published_year	YEAR		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
5	genre	VARCHAR	30	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
6	copies_availab...	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

3. Write an SQL query to insert the following book details into the table:

- Book ID: 101, Title: "SQL Fundamentals", Author: "John Doe", Published Year: 2022, Genre: "Technology", Copies Available: 5.

The screenshot shows the HeidiSQL interface with the following SQL query in the editor:

```
1 USE library_books;
2
3 INSERT INTO librarybooks_t (book_id, title, author, published_year, genre, copies_available)
4 VALUES
5 (101, 'SQL Fundamentals', 'John Doe', 2002, 'Technology', 5)
```

The screenshot shows the result of the INSERT query in the HeidiSQL interface. The table 'library_books.librarybooks_t' contains 1 row. The data is as follows:

book_id	title	author	published_year	genre	copies_available
101	SQL Fundamentals	John Doe	2002	Technology	5

4. Write an SQL query to update the copies_available for the book with ID 101 to 7.

The screenshot shows the HeidiSQL interface with the following SQL query in the editor:

```
1 USE library_books;
2
3 UPDATE librarybooks_t
4 SET copies_available = 7
5 WHERE copies_available = 5 AND book_id = 101;
```

The screenshot shows the result of the UPDATE query in the HeidiSQL interface. The table 'library_books.librarybooks_t' contains 1 row. The data is as follows:

book_id	title	author	published_year	genre	copies_available
101	SQL Fundamentals	John Doe	2002	Technology	7

5. Write an SQL query to delete all books published before 2000.

The screenshot shows the HeidiSQL interface with the following SQL query in the editor:

```
1 USE library_books;
2
3 DELETE FROM librarybooks_t
4 WHERE published_year < 2000;
```

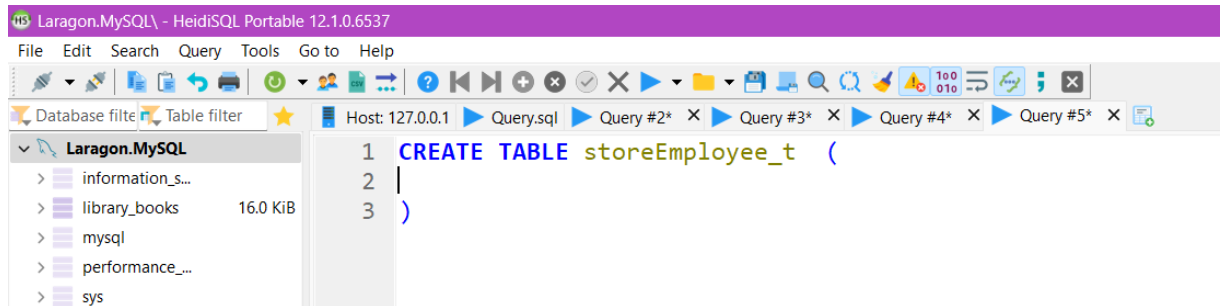
Part #2

Instructions:

For each scenario below, identify the most appropriate SQL command (CREATE, ALTER, DROP, SELECT, INSERT, UPDATE, DELETE, GRANT, REVOKE) and provide a brief justification for your choice.

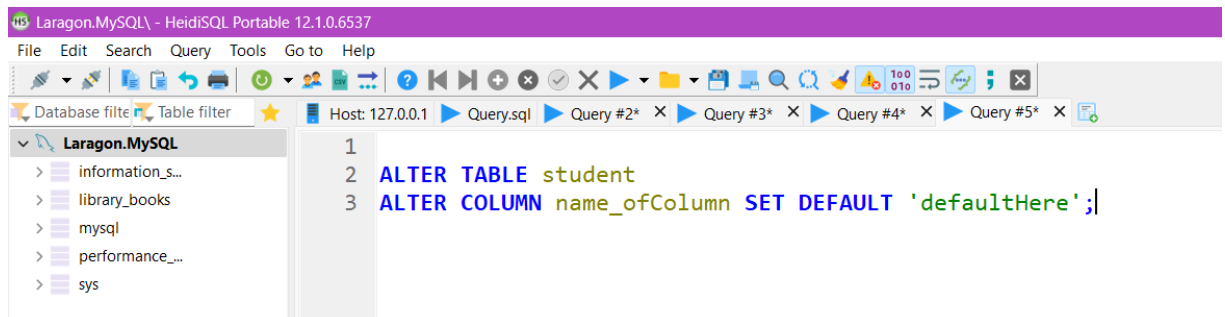
1. You need to create a new table in the database to store employee records.

- In this scenario, the most appropriate SQL Command is **CREATE** as only the CREATE SQL Command could create new databases and new tables inside a database.



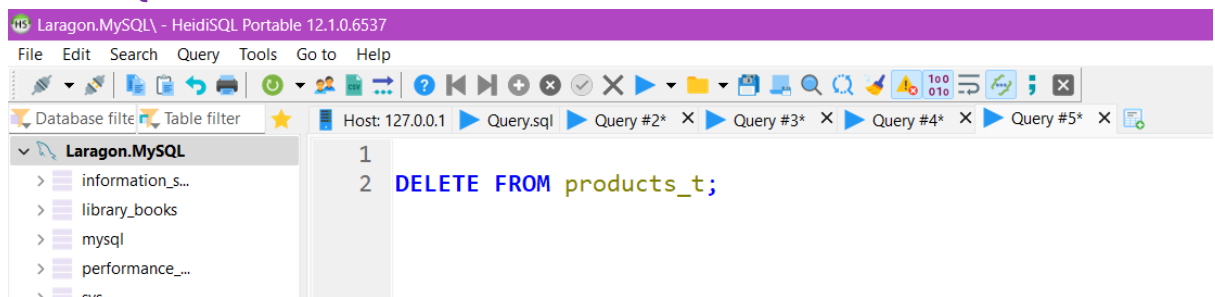
2. A column in the "student" table needs to have a default value updated.

- For this scenario, I would use **ALTER SQL Command** as it could add a constraint like *SET DEFAULT* to a column.



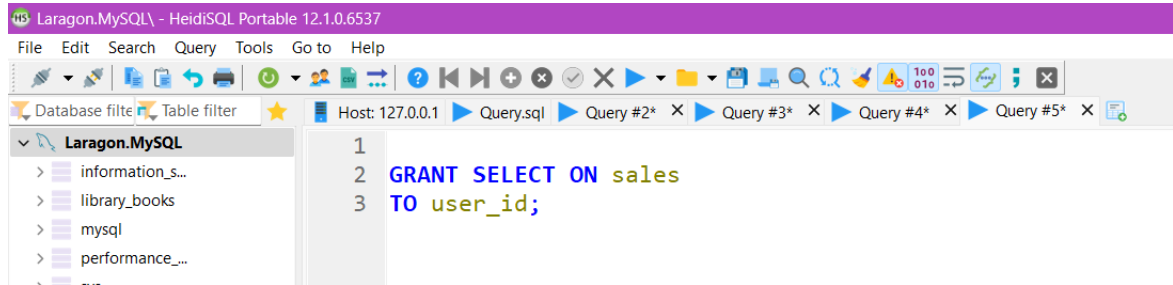
3. You want to delete all records from the "products" table but keep the table structure.

- If I would be to delete records from the "products" table but keep the table structure, I would use the **DELETE SQL Command** wherein I would not use a WHERE Clause to delete a whole row of data.



4. A user needs permission to view and query the "sales" table.

- If a user needs permission to view and query the "sales" table, the perfect SQL Command for this scenario is **GRANT** as it allows users to perform specific tasks when used.



The screenshot shows the HeidiSQL interface with the title bar 'Laragon.MySQL - HeidiSQL Portable 12.1.0.6537'. The menu bar includes File, Edit, Search, Query, Tools, Go to, and Help. The toolbar contains various icons for file operations, database connections, and query execution. The 'Database filter' and 'Table filter' are set to 'Laragon.MySQL'. The left sidebar shows a tree view of databases: information_s..., library_books, mysql, performance_..., and sys. The main query editor displays the following SQL command:

```
1  
2 GRANT SELECT ON sales  
3 TO user_id;
```

5. You need to remove the "inventory" table entirely from the database.

- To remove the whole inventory table entirely from the database, I would use **DROP SQL COMMAND** as it is the command that is used to drop tables from a specific database.



The screenshot shows the HeidiSQL interface with the title bar 'Laragon.MySQL - HeidiSQL Portable 12.1.0.6537'. The menu bar includes File, Edit, Search, Query, Tools, Go to, and Help. The toolbar contains various icons for file operations, database connections, and query execution. The 'Database filter' and 'Table filter' are set to 'Laragon.MySQL'. The left sidebar shows a tree view of databases: information_s..., library_books, mysql, performance_..., and sys. The main query editor displays the following SQL command:

```
1  
2  
3 DROP TABLE inventory_t;|
```