Module 4 – Introduction to DBMS Introduction to SQL

Theory Questions:

1. What is SQL, and why is it essential in database management?

=>**SQL** is a standard programming language designed for managing and manipulating relational databases. It is used to query, update, insert, and delete data within a database.

**Importance in Database Management:**

* Data Manipulation: SQL allows users to insert, update, delete, and retrieve data efficiently.
* Data Integrity: SQL provides the means to define constraints and rules, ensuring that data is accurate and consistent.

2. Explain the difference between DBMS and RDBMS.

=>**DBMS (Database Management System):** A DBMS is a software system that provides an interface for users to interact with databases. It supports basic operations such as storing, retrieving, and managing data. However, it does not necessarily organize the data in a relational format.

**RDBMS (Relational Database Management System):** An RDBMS is a type of DBMS that specifically uses a relational model for storing data. Data is organized into tables (relations), and relationships between these tables are established using keys (primary and foreign keys).

3. Describe the role of SQL in managing relational databases.

=>SQL is used to define the structure of databases through commands like CREATE TABLE, ALTER TABLE, and DROP TABLE.

SQL allows for inserting (INSERT), updating (UPDATE), and deleting (DELETE) data within the database.

SQL provides constraints like PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, and CHECK to enforce data integrity and ensure that the data is consistent and valid.

SQL can be used for controlling access to data with commands like GRANT and REVOKE, providing security and managing who can perform various operations on the database.

4. What are the key features of SQL?

=>

 Simplicity.

 Data **Querying**

 Data **Integrity**

 Scalability

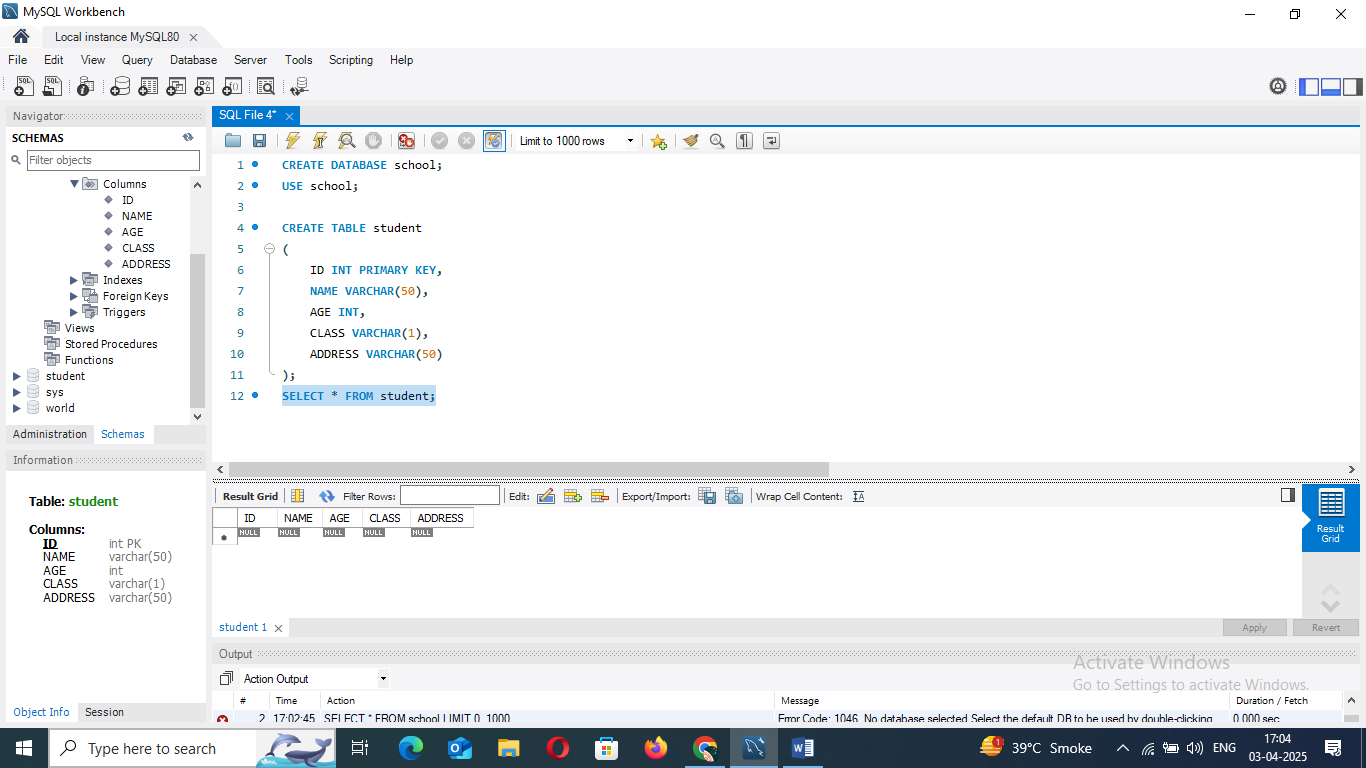
 Data **Definition**

 Data **Manipulation**

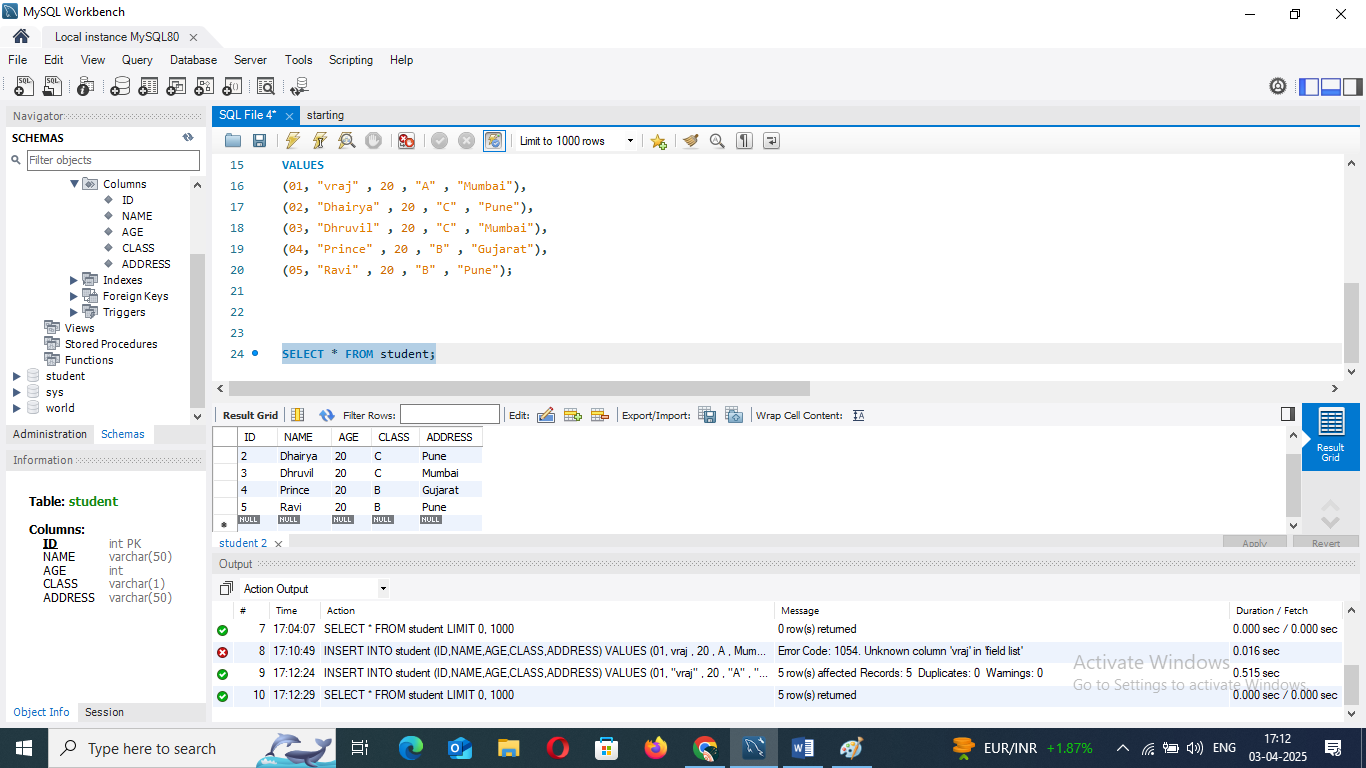
 Transaction **Control**

LAB EXERCISES:

• Lab 1: Create a new database named school\_db and a table called students with the following columns: student\_id, student\_name, age, class, and address.

=>

• Lab 2: Insert five records into the students table and retrieve all records using the SELECT statement.



2. SQL Syntax

Theory Questions:

1. What are the basic components of SQL syntax?

=> Reserved words that SQL recognizes and uses to perform operations. Examples include SELECT, INSERT, UPDATE, DELETE, FROM, WHERE, JOIN, ORDER BY, etc.

2. Write the general structure of an SQL SELECT statement.

=> SELECT column1, column2, ...

FROM table\_name

WHERE condition

GROUP BY column

HAVING condition

ORDER BY column [ASC|DESC]

LIMIT number;

3. Explain the role of clauses in SQL statements.

### => 1. SELECT Clause:

* **Role**: Specifies the columns that you want to retrieve from one or more tables.

**2. FROM Clause:**

* **Role**: Identifies the table or tables from which to retrieve the data.

**3. WHERE Clause:**

* **Role**: Filters the rows that are returned based on specified conditions.

**4. GROUP BY Clause:**

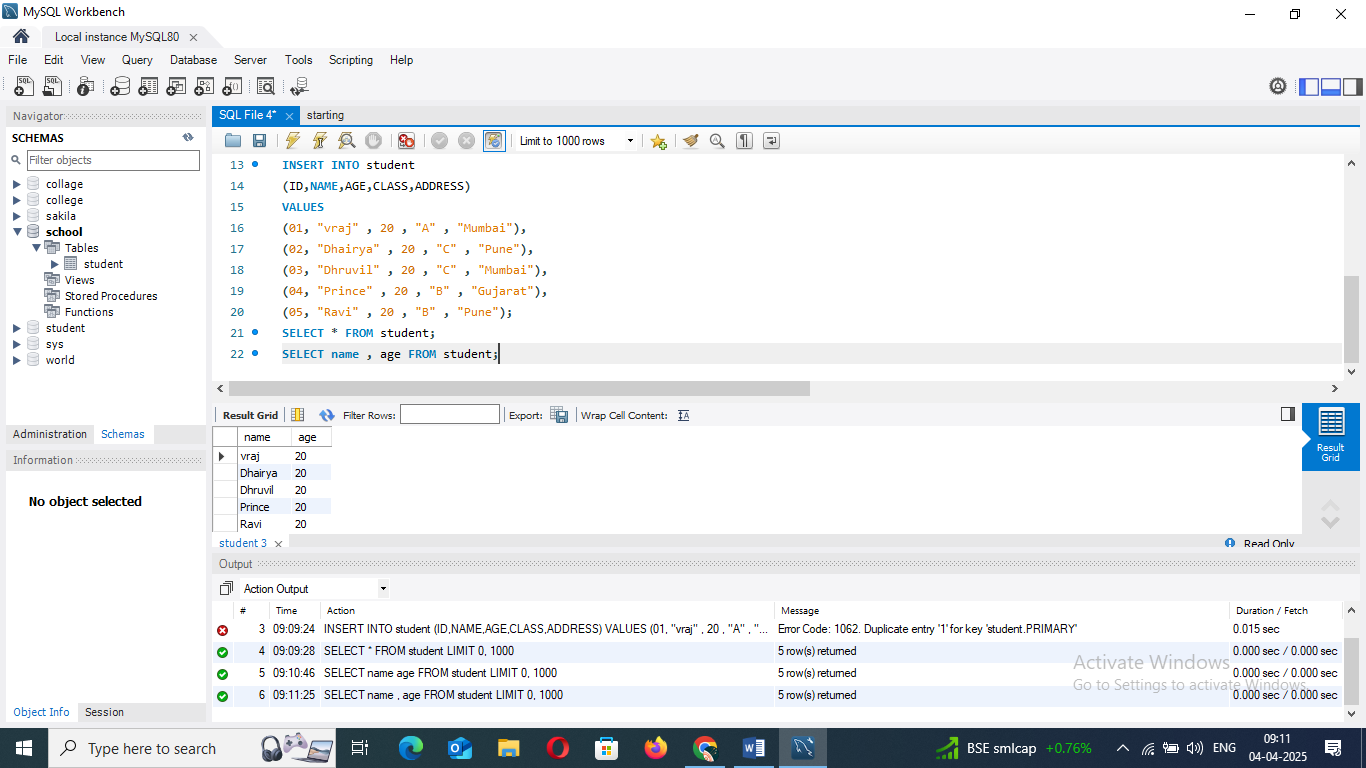
* **Role**: Groups rows that have the same values in specified columns into summary rows.

**5. HAVING Clause:**

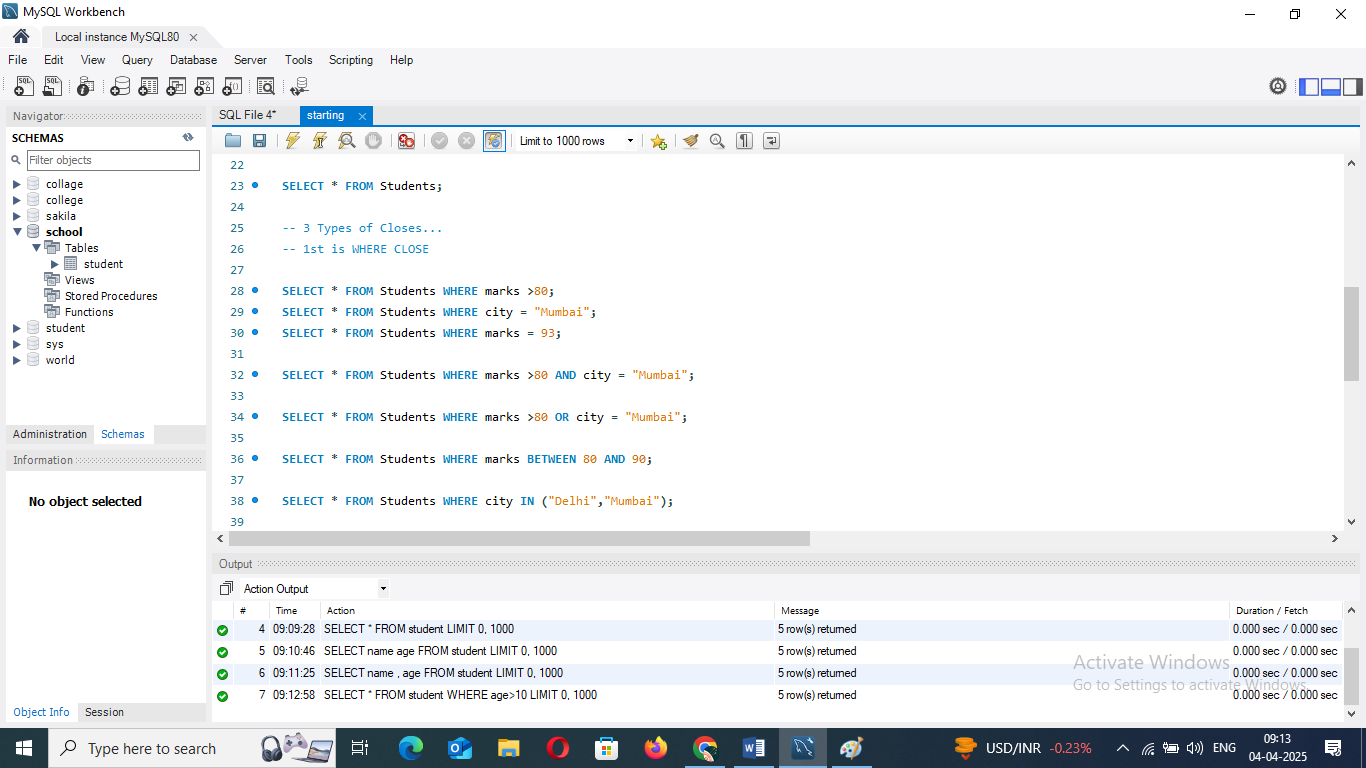
* **Role**: Filters the result of the GROUP BY operation. It works similarly to WHERE but operates on groups rather than individual rows.

LAB EXERCISES:

• Lab 1: Write SQL queries to retrieve specific columns (student\_name and age) from the students table.

=> 

• Lab 2: Write SQL queries to retrieve all students whose age is greater than 10.



3. SQL Constraints

Theory Questions:

1. What are constraints in SQL? List and explain the different types of constraints.

=>In SQL, **constraints** are rules that are applied to table columns to ensure the integrity, accuracy, and reliability of the data stored in the database.

**Types of Constraints in SQL**

1. NOT NULL Constraint
2. PRIMARY KEY Constraint:
3. FOREIGN KEY Constraint:
4. **CHECK Constraint:**
5. DEFAULT Constraint:

2. How do PRIMARY KEY and FOREIGN KEY constraints differ?

### => 1. PRIMARY KEY Constraint:

* **Purpose**: The PRIMARYKEY uniquely identifies each record in a table.
* **Usage**: Typically used to define the main identifier of a table (e.g., student\_id, employee\_id).

**2. FOREIGN KEY Constraint:**

* **Purpose**: The FOREIGNKEY establishes a relationship between two tables by linking a column in one table to the primary key (or a unique key) in another table.
* **Usage**: Used to maintain referential integrity by ensuring that values in the foreign key column exist in the referenced primary key column in the related table.

3. What is the role of NOT NULL and UNIQUE constraints?

=> **Role**: The **NOT NULL** constraint ensures that a column cannot have NULL values. This means every row in that column must have a valid value. It is used to guarantee that certain columns will always contain data and cannot be left empty.

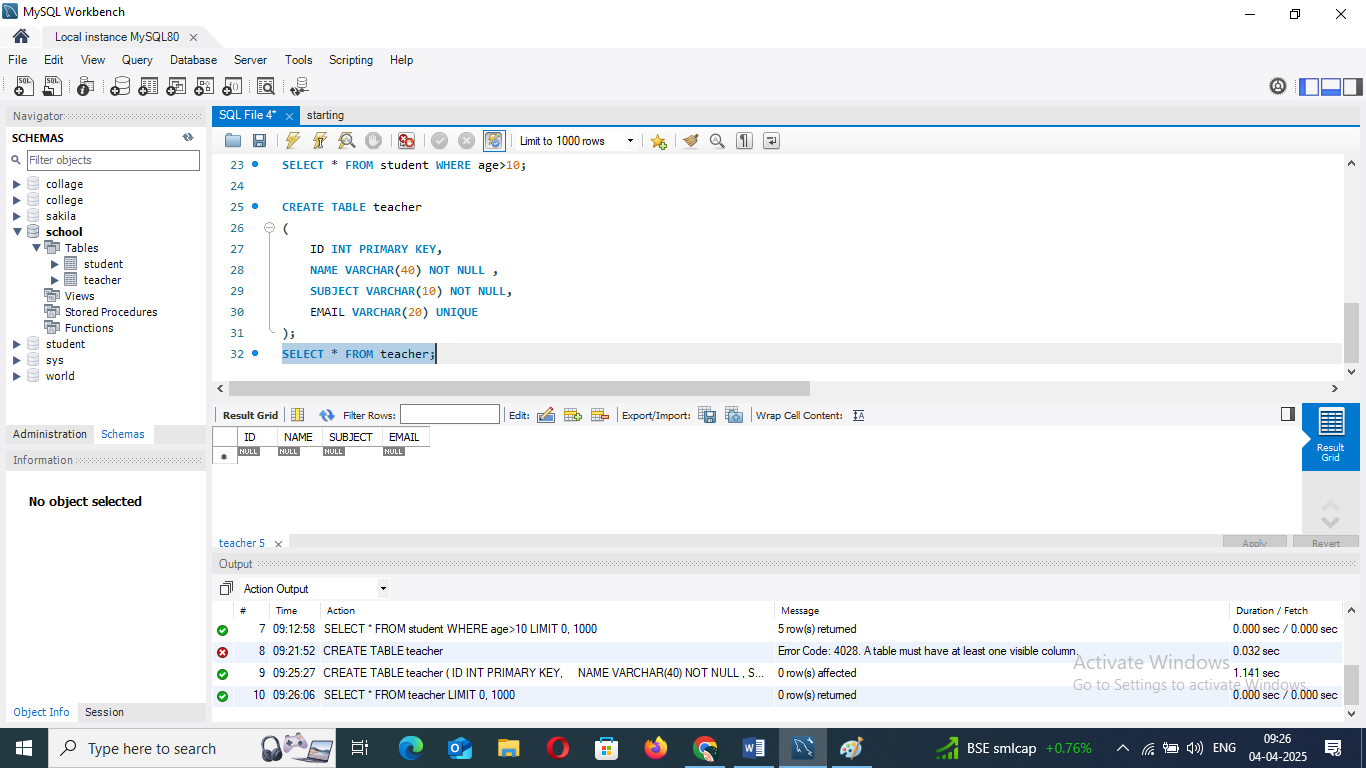
**Role**: The **UNIQUE** constraint ensures that all values in a column (or a combination of columns) are unique, meaning no two rows can have the same value(s) for that column(s).

Top of Form

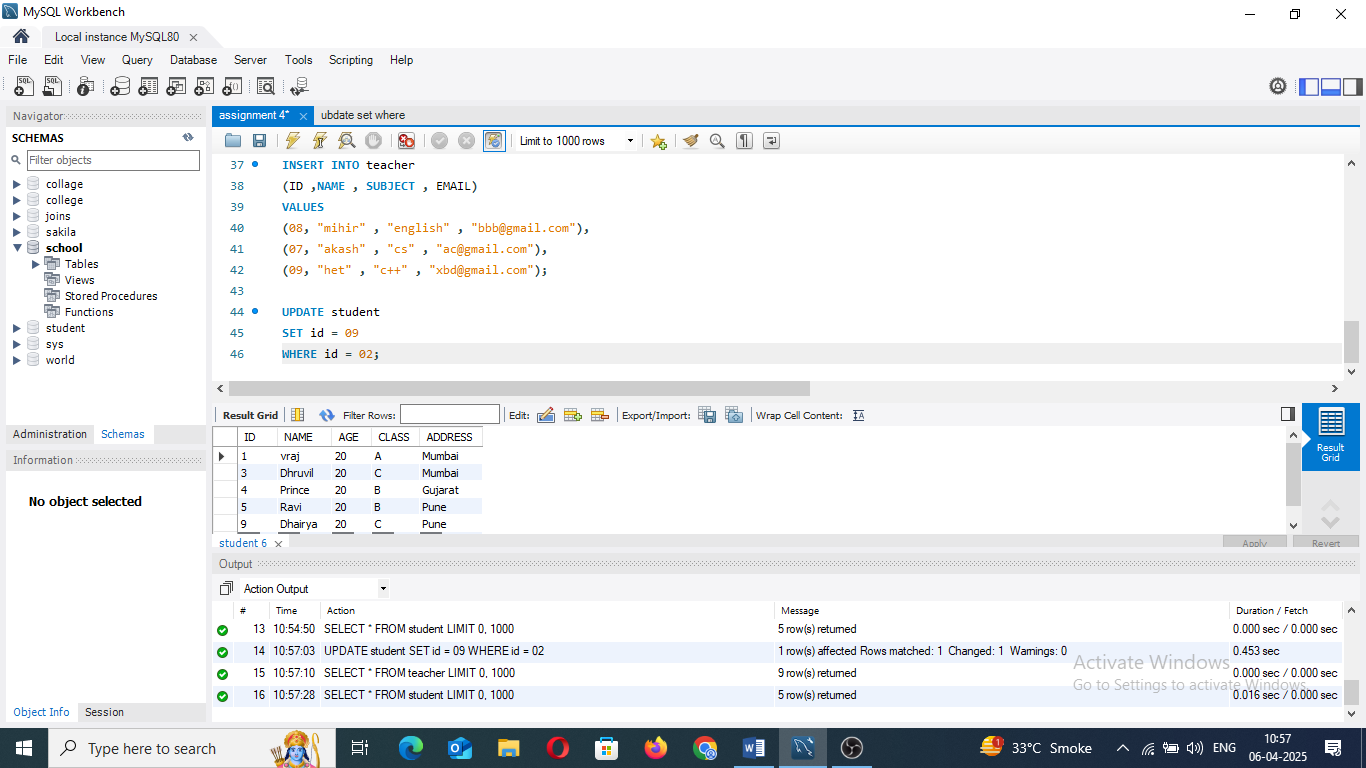
Bottom of Form

LAB EXERCISES:

• Lab 1: Create a table teachers with the following columns: teacher\_id (Primary Key), teacher\_name (NOT NULL), subject (NOT NULL), and email (UNIQUE).



• Lab 2: Implement a FOREIGN KEY constraint to relate the teacher\_id from the teachers table with the students table.



4. Main SQL Commands and Sub-commands (DDL)

Theory Questions:

1. Define the SQL Data Definition Language (DDL).

=>SQL Data Definition Language (DDL) refers to the subset of SQL (Structured Query Language) used to define, manage, and modify database structures, such as tables, schemas, indexes, and views.

2. Explain the CREATE command and its syntax.

=> CREATE TABLE table\_name (

column1 datatype [constraints],

column2 datatype [constraints],

...

);

3. What is the purpose of specifying data types and constraints during table creation?

=>Data types define the kind of data that can be stored in a column, which is essential for several reasons:

(Data Integrity

Memory Efficiency)

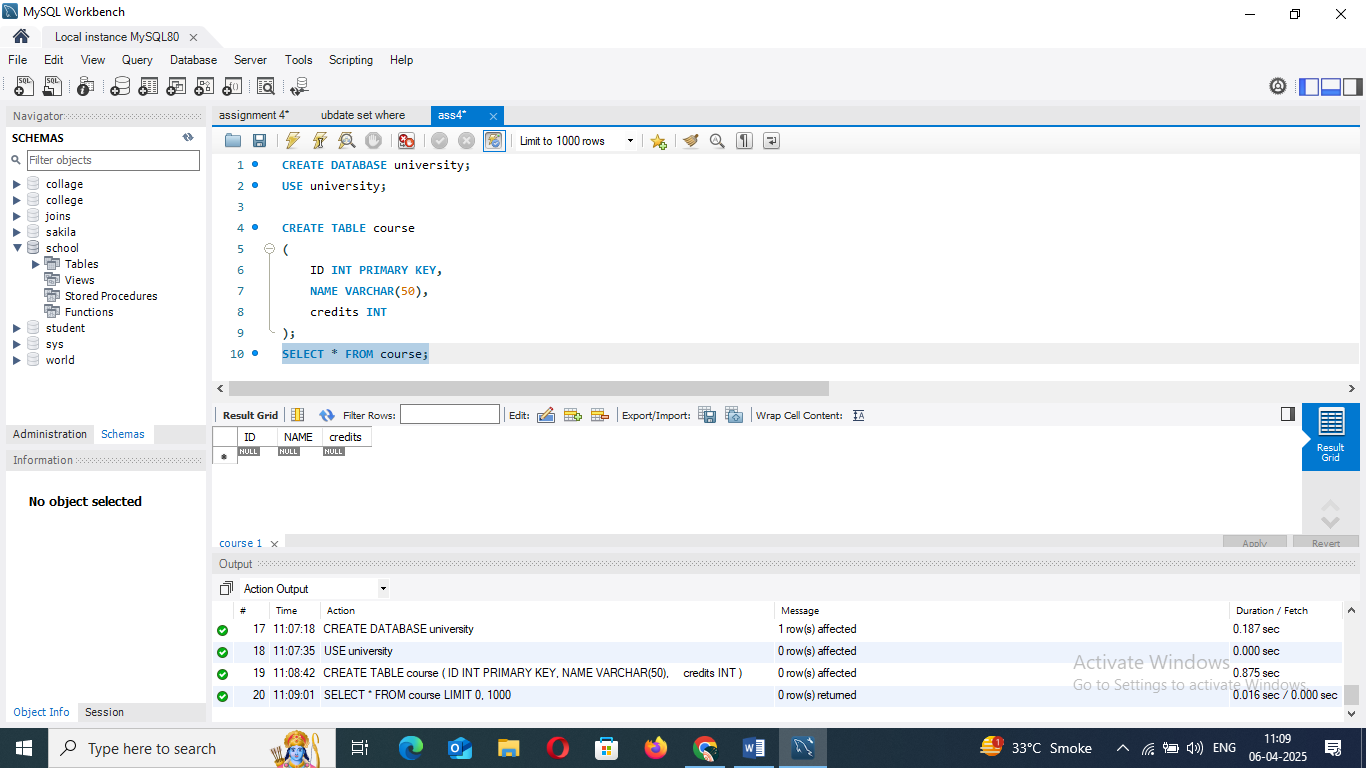
Constraints help enforce rules that ensure the quality and consistency of the data stored in the table. Here’s why they are important:

(Enforcing Uniqueness

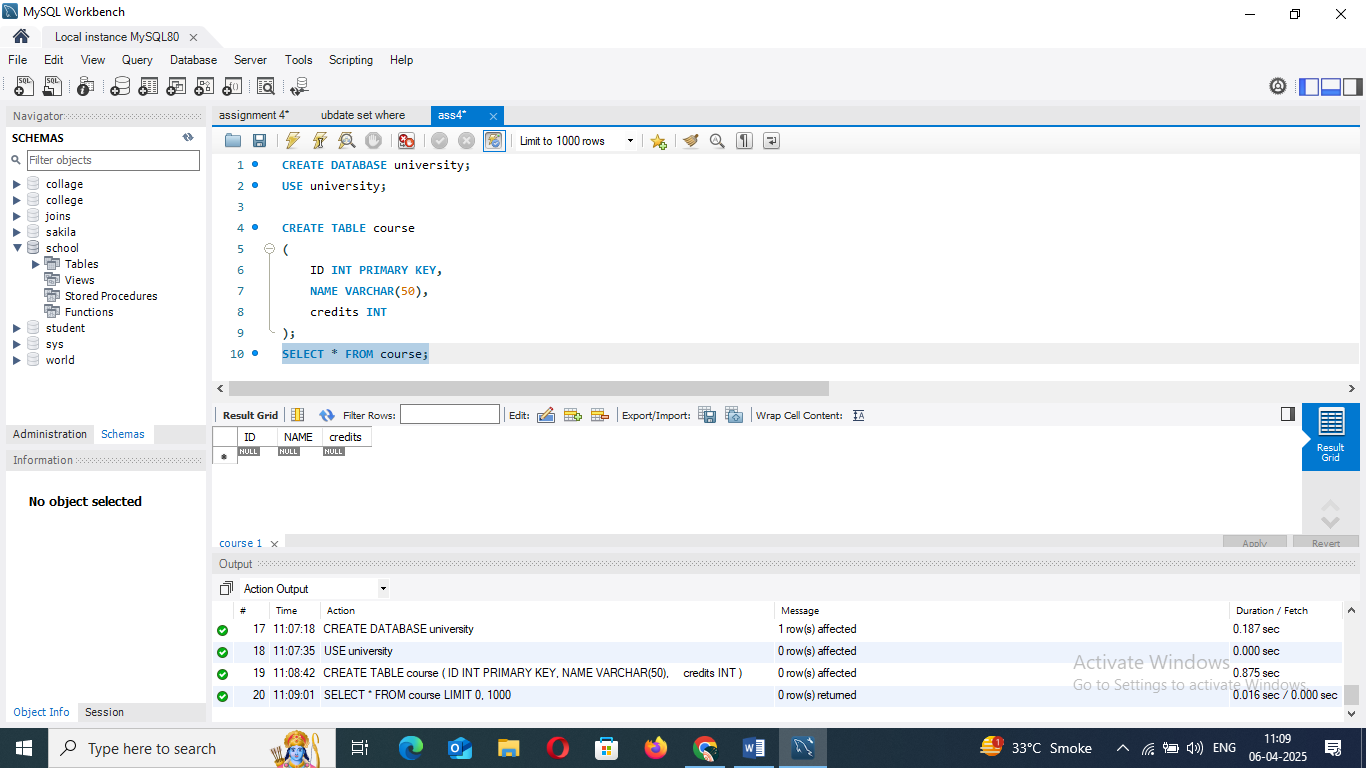
Ensuring Data Validity)

LAB EXERCISES:

• Lab 1: Create a table courses with columns: course\_id, course\_name, and course\_credits. Set the course\_id as the primary key.

=> 

• Lab 2: Use the CREATE command to create a database university\_db

=> 

5. ALTER Command

Theory Questions:

1. What is the use of the ALTER command in SQL?

=> The ALTER command in SQL is used to modify an existing database object, such as a table, view, or schema. Unlike the CREATE command, which is used to create new database objects, ALTER allows you to make changes to the structure of an existing object without having to drop and recreate it.

2. How can you add, modify, and drop columns from a table using ALTER?

=> ALTER TABLE table\_name

ADD column\_name datatype [constraints];

ALTER TABLE table\_name

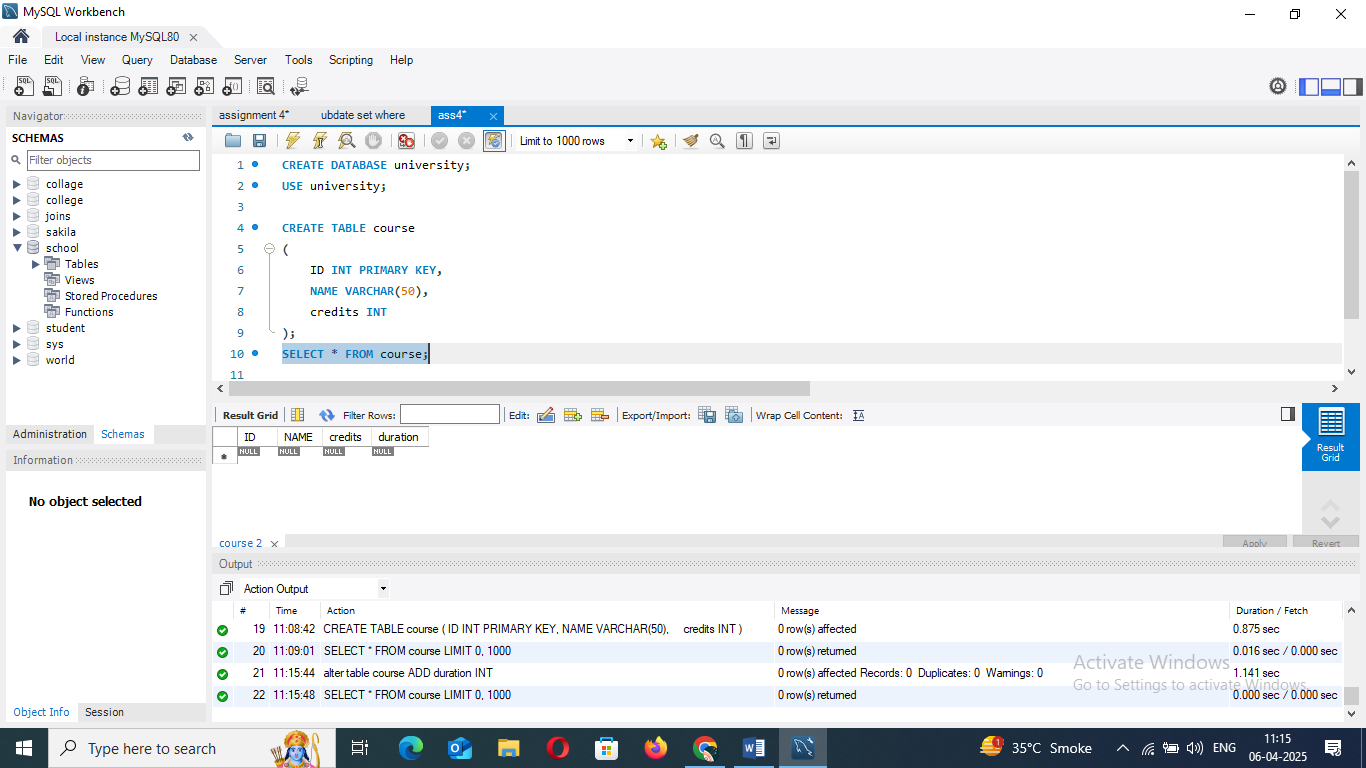
MODIFY column\_name new\_datatype [constraints];

ALTER TABLE table\_name

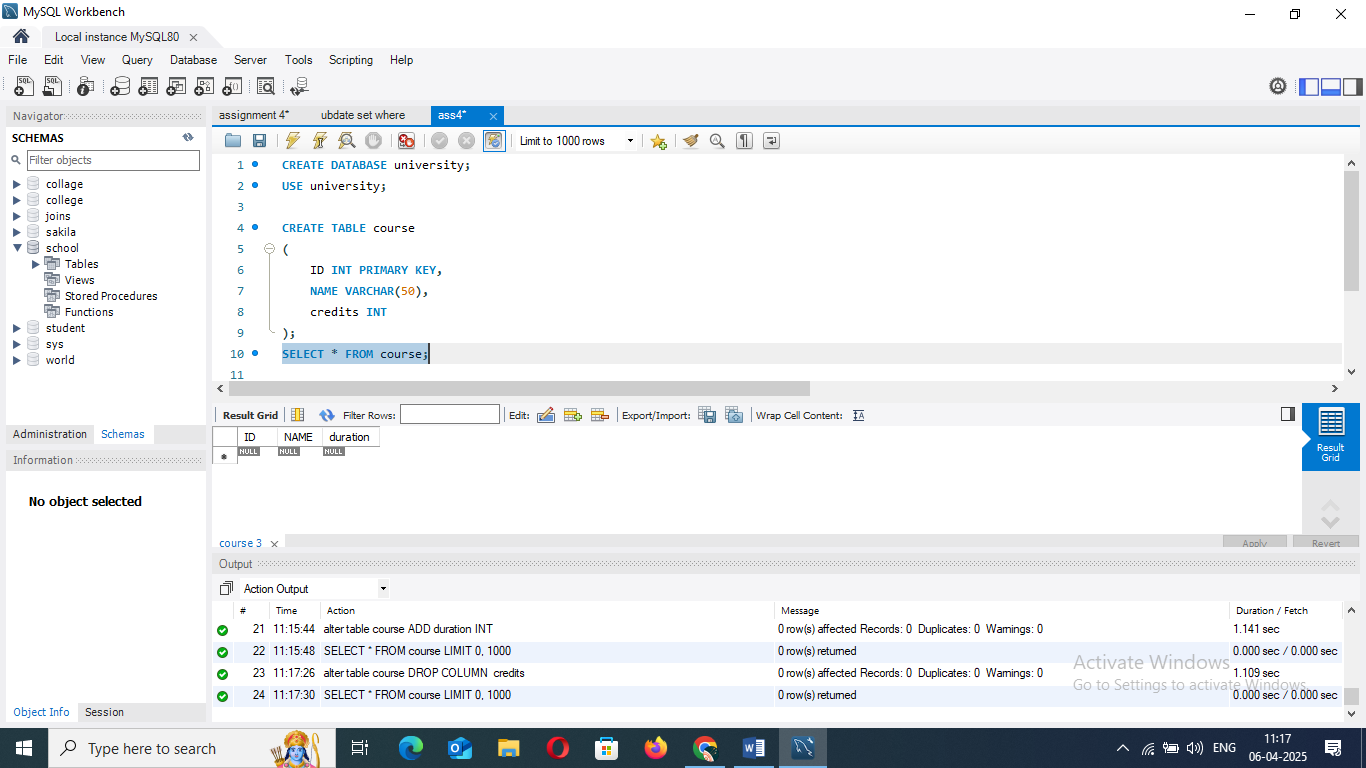
DROP COLUMN column\_name;

LAB EXERCISES:

• Lab 1: Modify the courses table by adding a column course\_duration using the ALTER command.



• Lab 2: Drop the course\_credits column from the courses table.



6. DROP Command

Theory Questions:

1. What is the function of the DROP command in SQL?

=> The DROP command in SQL is used to **remove database objects** such as tables, views, indexes, schemas, or even entire databases.

2. What are the implications of dropping a table from a database?

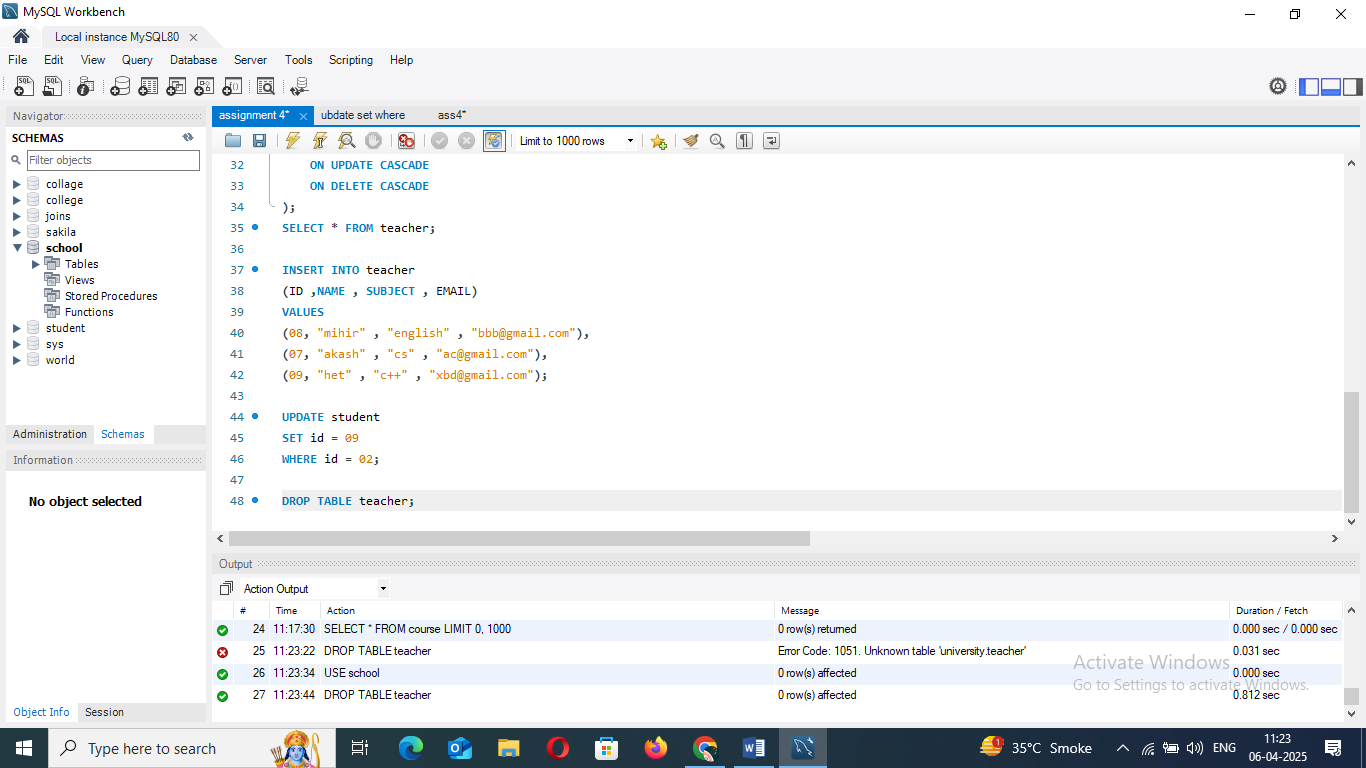
=> **1. Loss of Data**

**2. Removal of the Table Structure**

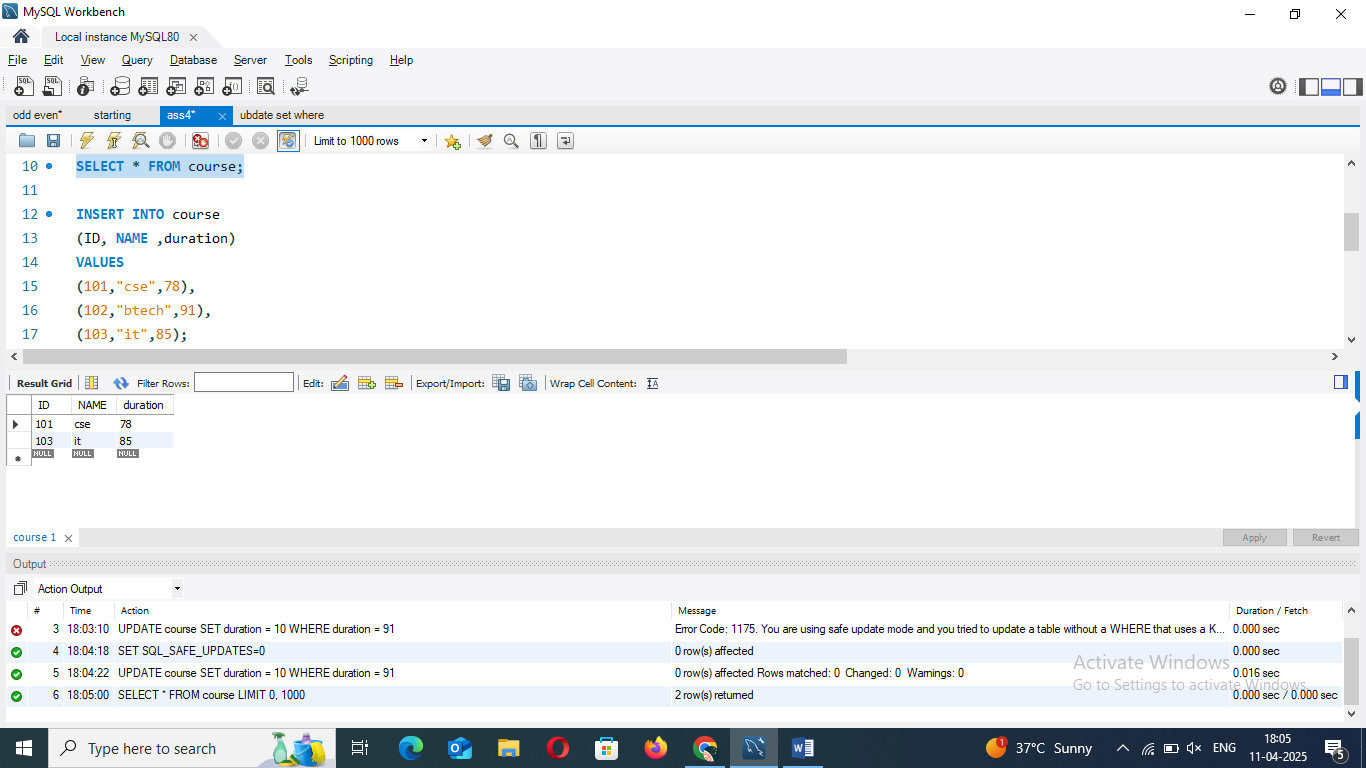
**3. Impact on Associated Indexes**

**LAB EXERCISES:**

• Lab 1: Drop the teachers table from the school\_db database.



• Lab 2: Drop the students table from the school\_db database and verify that the table has been removed.



7. Data Manipulation Language (DML)

Theory Questions:

1. Define the INSERT, UPDATE, and DELETE commands in SQL.

=> The INSERT command is used to **add new records (rows)** into a table.

The UPDATE command is used to **modify existing records** in a table.

The DELETE command is used to **remove records** from a table.

2. What is the importance of the WHERE clause in UPDATE and DELETE operations?

=>In UPDATE Statements

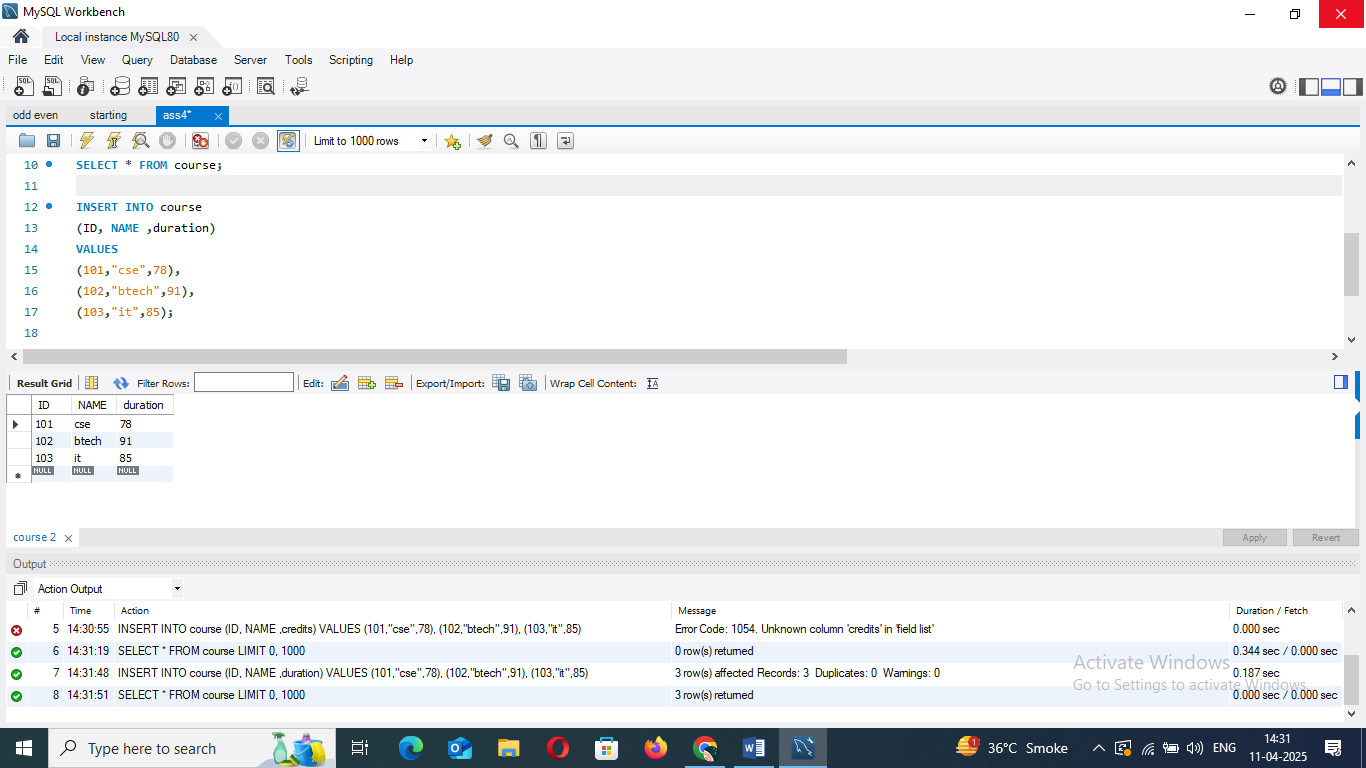
* **With** WHERE**:** Only specific rows are updated.
* **Without** WHERE**:** **All** rows are updated — which might not be what you intended.

In DELETE Statements

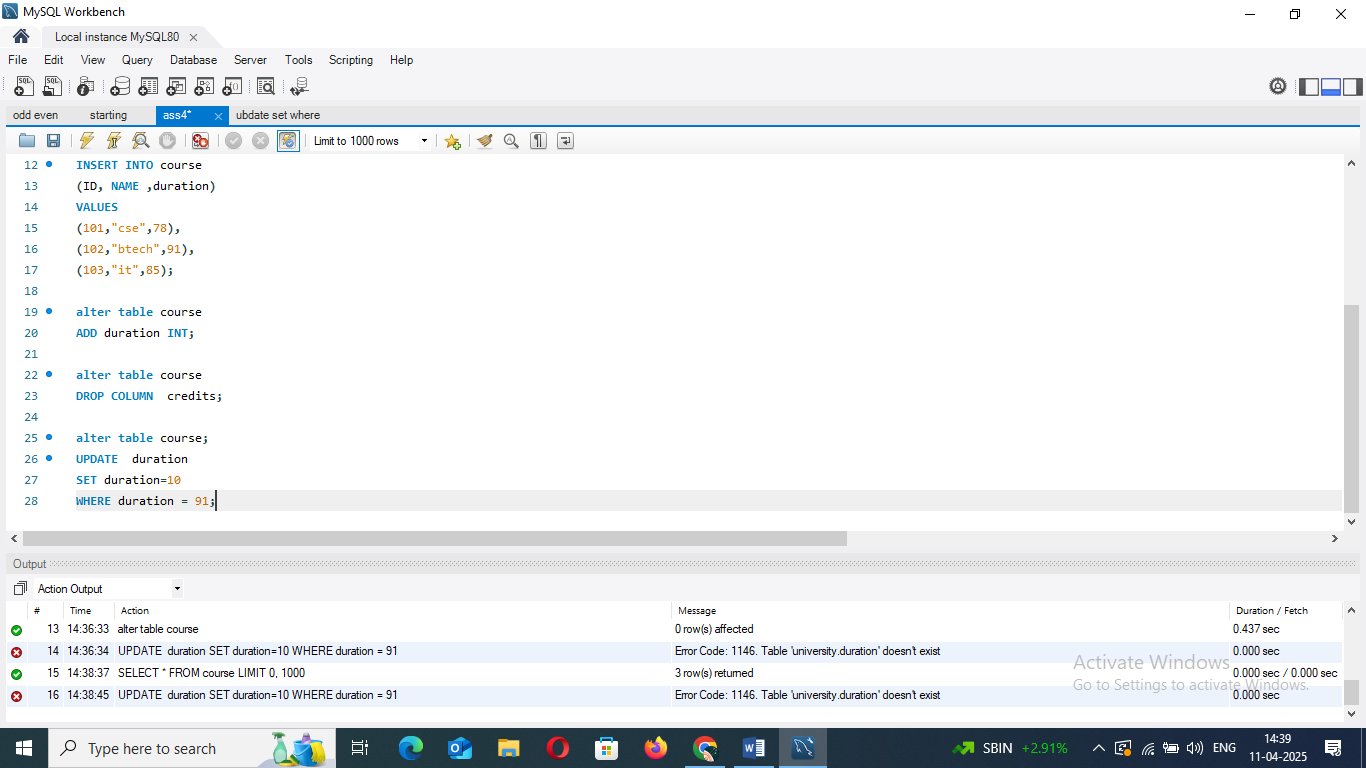
* **With** WHERE**:** Only specific rows are deleted.
* **Without** WHERE**:** The entire table is emptied — all records are gone!

LAB EXERCISES:

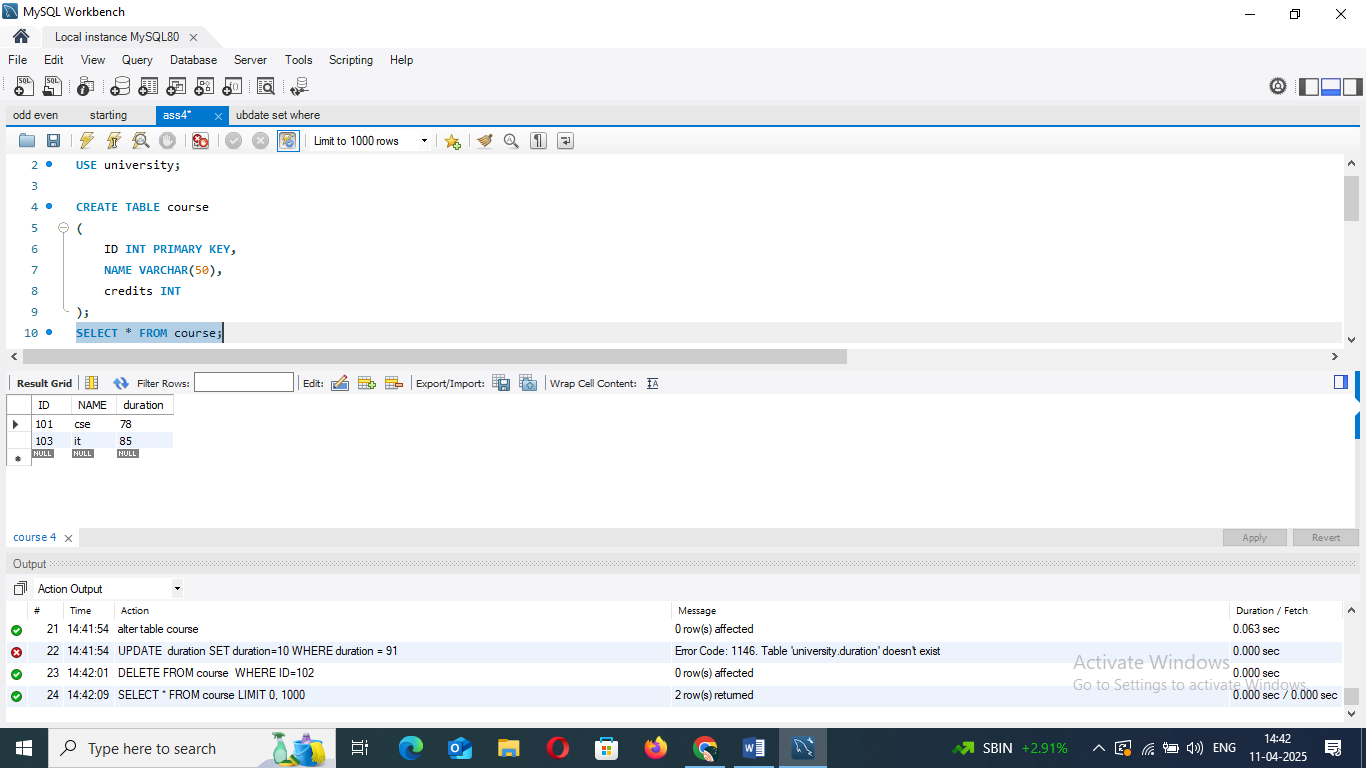
• Lab 1: Insert three records into the courses table using the INSERT command.



• Lab 2: Update the course duration of a specific course using the UPDATE command.



• Lab 3: Delete a course with a specific course\_id from the courses table using the DELETE command.



8. Data Query Language (DQL)

Theory Questions:

1. What is the SELECT statement, and how is it used to query data?

=>The SELECT statement is the most commonly used SQL command. It's used to **retrieve data from one or more tables** in a database.

2. Explain the use of the ORDER BY and WHERE clauses in SQL queries.

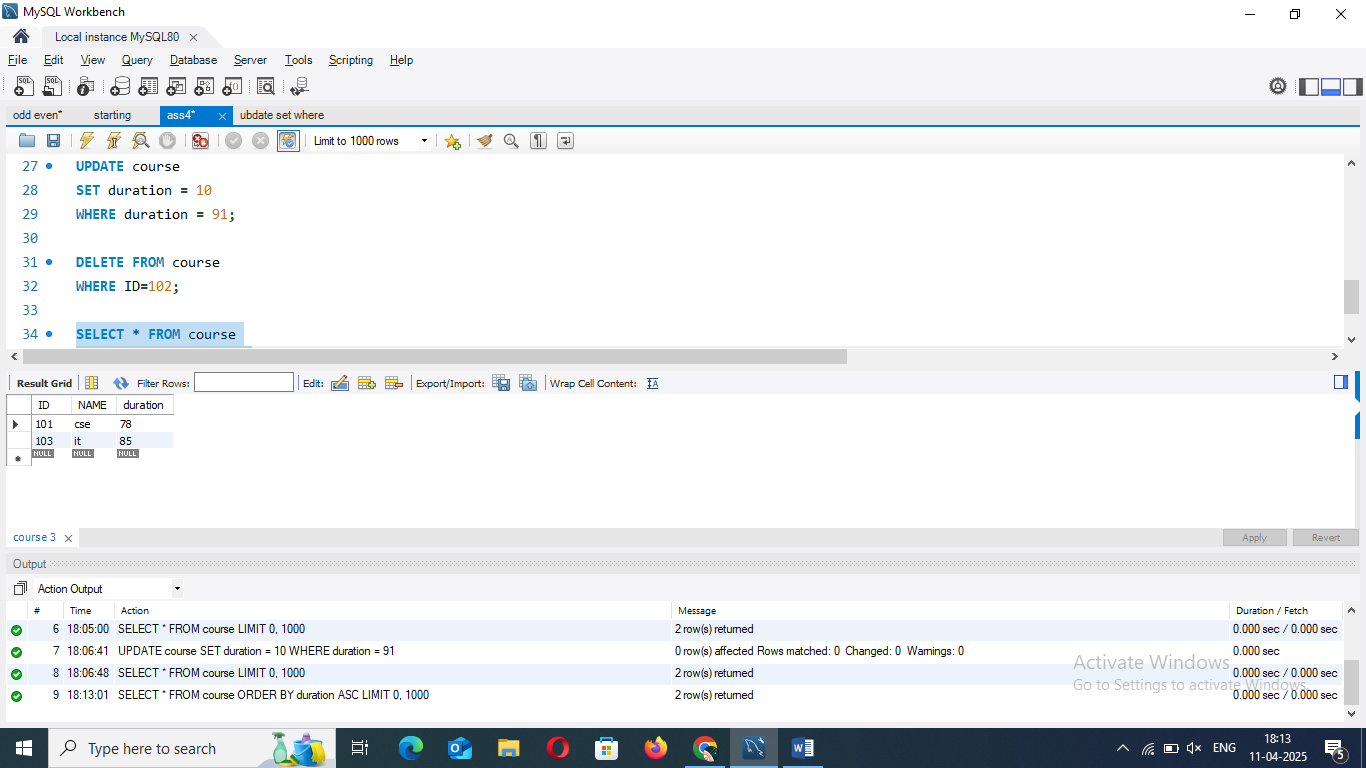
=>ORDER BY **Clause:** Used to **sort the results** returned by a SELECT statement by one or more columns.

-WHERE **Clause:** Used to **filter records** based on specific conditions.

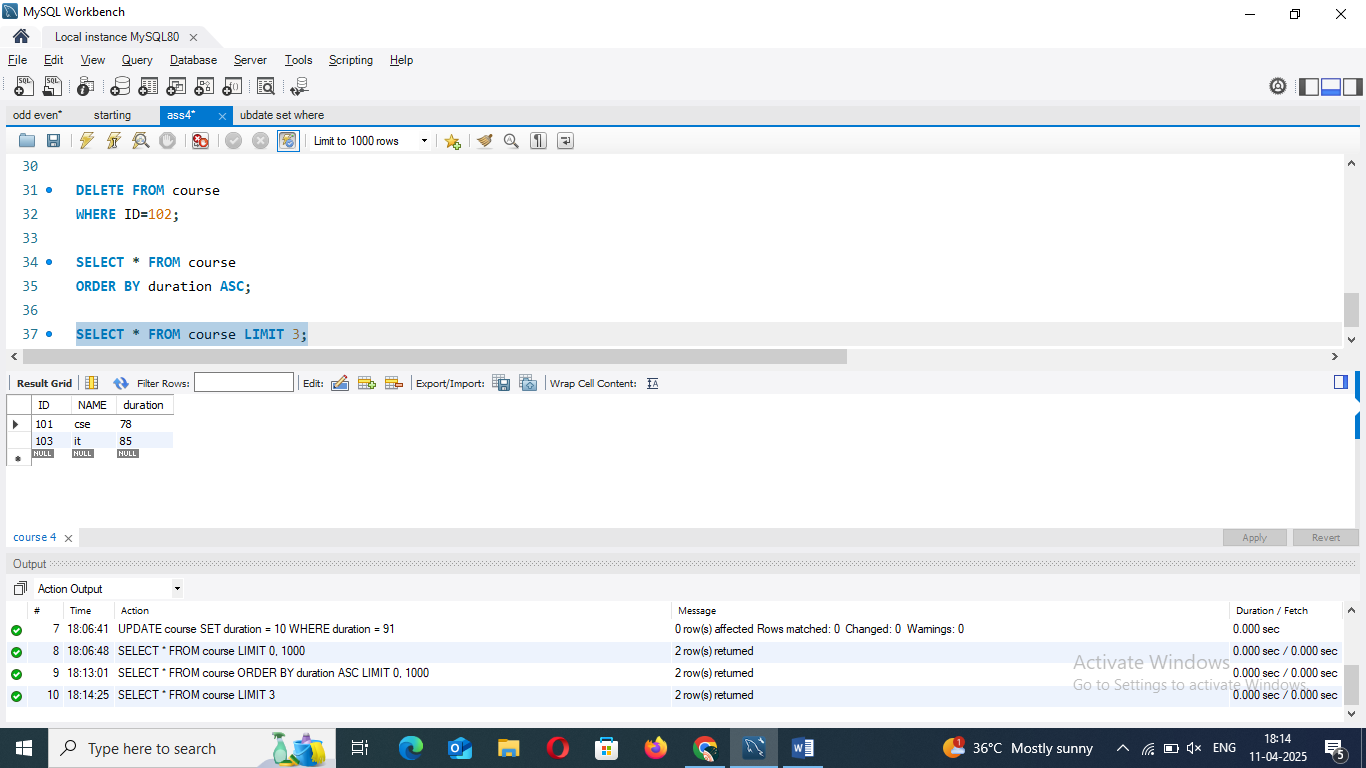
LAB EXERCISES:

• Lab 1: Retrieve all courses from the courses table using the SELECT statement.

• Lab 2: Sort the courses based on course\_duration in descending order using ORDER BY.



• Lab 3: Limit the results of the SELECT query to show only the top two courses using LIMIT.



9. Data Control Language (DCL)

Theory Questions:

1. What is the purpose of GRANT and REVOKE in SQL?

=> The GRANT statement gives **privileges** (permissions) to users so they can perform specific actions on database objects.

- The GRANT statement gives **privileges** (permissions) to users so they can perform specific actions on database objects.

2. How do you manage privileges using these commands?

=>

LAB EXERCISES:

• Lab 1: Create two new users user1 and user2 and grant user1 permission to SELECT from the courses table.

CREATE USER user1 IDENTIFIED BY 'password1';

CREATE USER user2 IDENTIFIED BY 'password2';

• Lab 2: Revoke the INSERT permission from user1 and give it to user2.

REVOKE INSERT ON database\_name.courses FROM user1;

GRANT INSERT ON database\_name.courses TO user2;

10. Transaction Control Language (TCL)

Theory Questions:

1. What is the purpose of the COMMIT and ROLLBACK commands in SQL?

### => COMMIT

* **Purpose:** Saves all changes made during the current transaction.

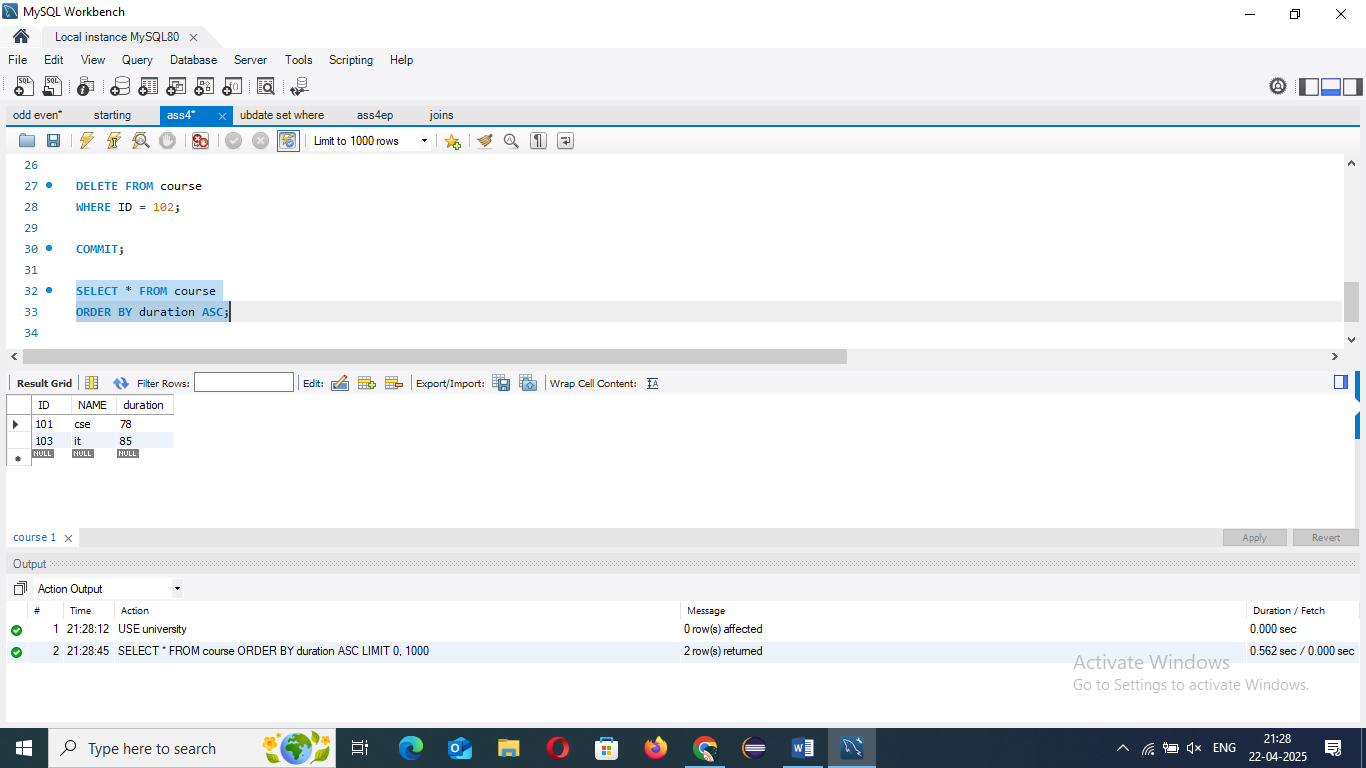
### ****ROLLBACK****

* **Purpose:** Undoes all changes made during the current transaction.

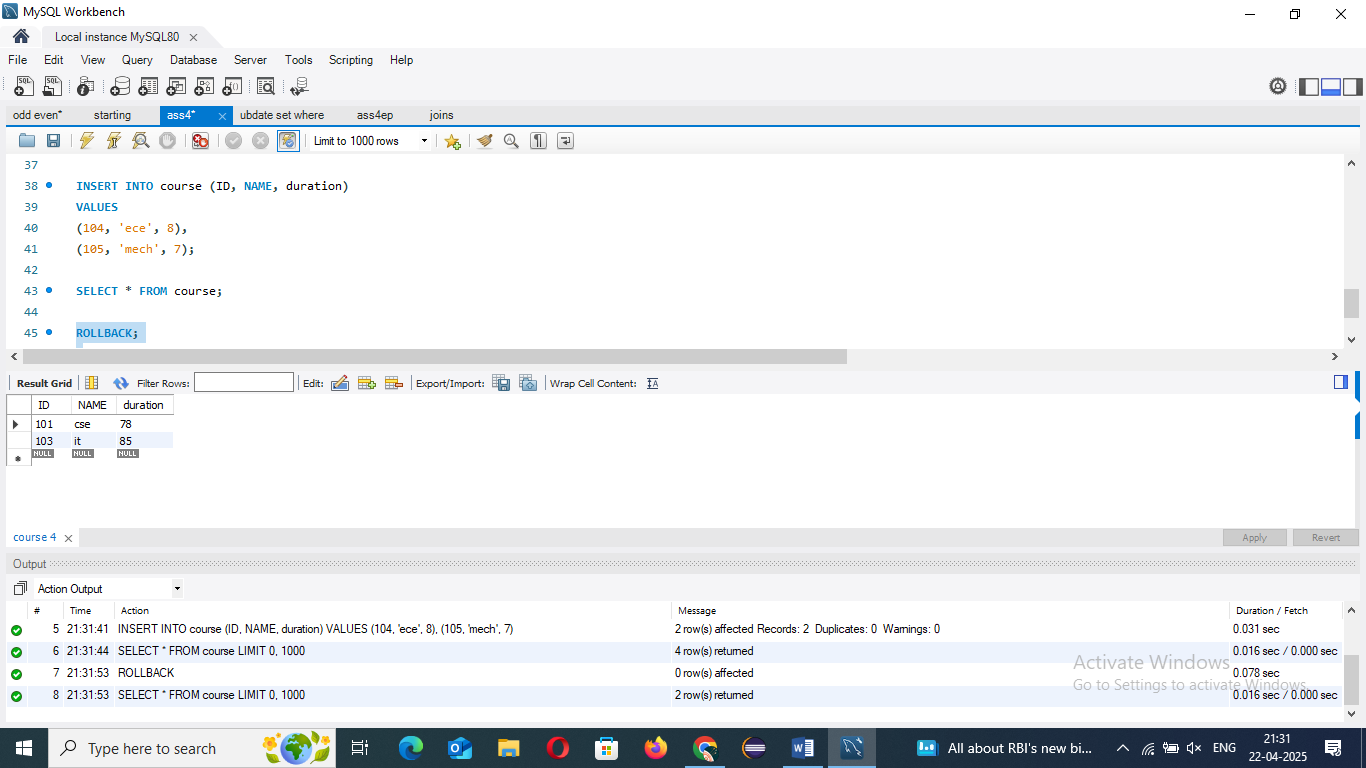
2. Explain how transactions are managed in SQL databases.

LAB EXERCISES:

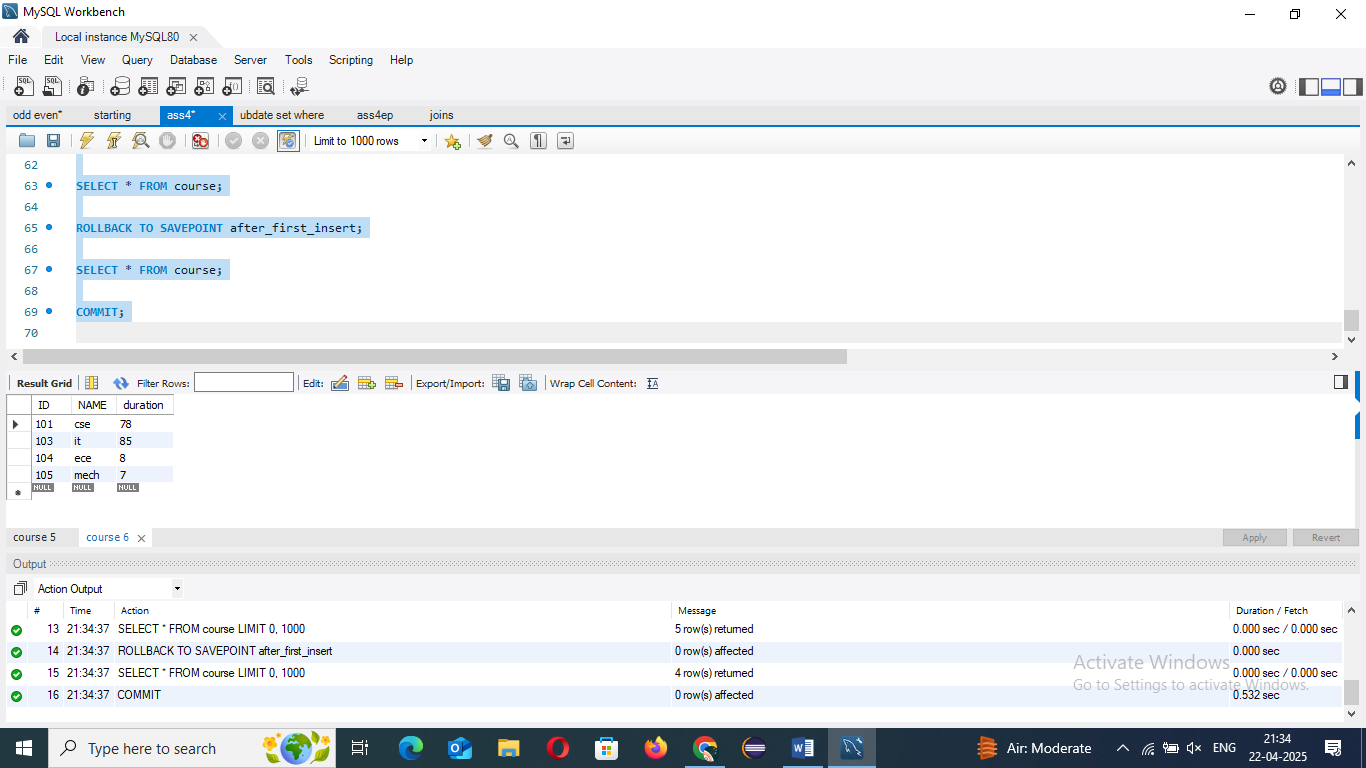
• Lab 1: Insert a few rows into the courses table and use COMMIT to save the changes.



• Lab 2: Insert additional rows, then use ROLLBACK to undo the last insert operation.



• Lab 3: Create a SAVEPOINT before updating the courses table, and use it to roll back specific changes.



11. SQL Joins Theory Questions:

1. Explain the concept of JOIN in SQL. What is the difference between INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN?

=> A **JOIN** is used to combine rows from twoormoretables based on a relatedcolumn between them — usually a primary key in one table and a foreign key in another.

1. 🟦 ***INNER JOIN***

* **Returns only matching rows** from both tables.

2. 🟨 ***LEFT JOIN*** (or ***LEFT OUTER JOIN***)

* Returns **all rows from the left table,** and matched rows from the right table.

3. 🟥 ***RIGHT JOIN*** (or ***RIGHT OUTER JOIN***)

* Opposite of LEFT JOIN.
* Returns **all rows from the right table,** and matched rows from the left.

4. 🟪 ***FULL OUTER JOIN***

* Returns **all rows from both tables.**
* If there's a match, you get merged data.

2. How are joins used to combine data from multiple tables?

=> JOINs work by using a commoncolumn between two (or more) tables — usually a primarykey in one table and a foreignkey in another.