SQL MOULE LAB – 2

By

Pedda Jagadeesh

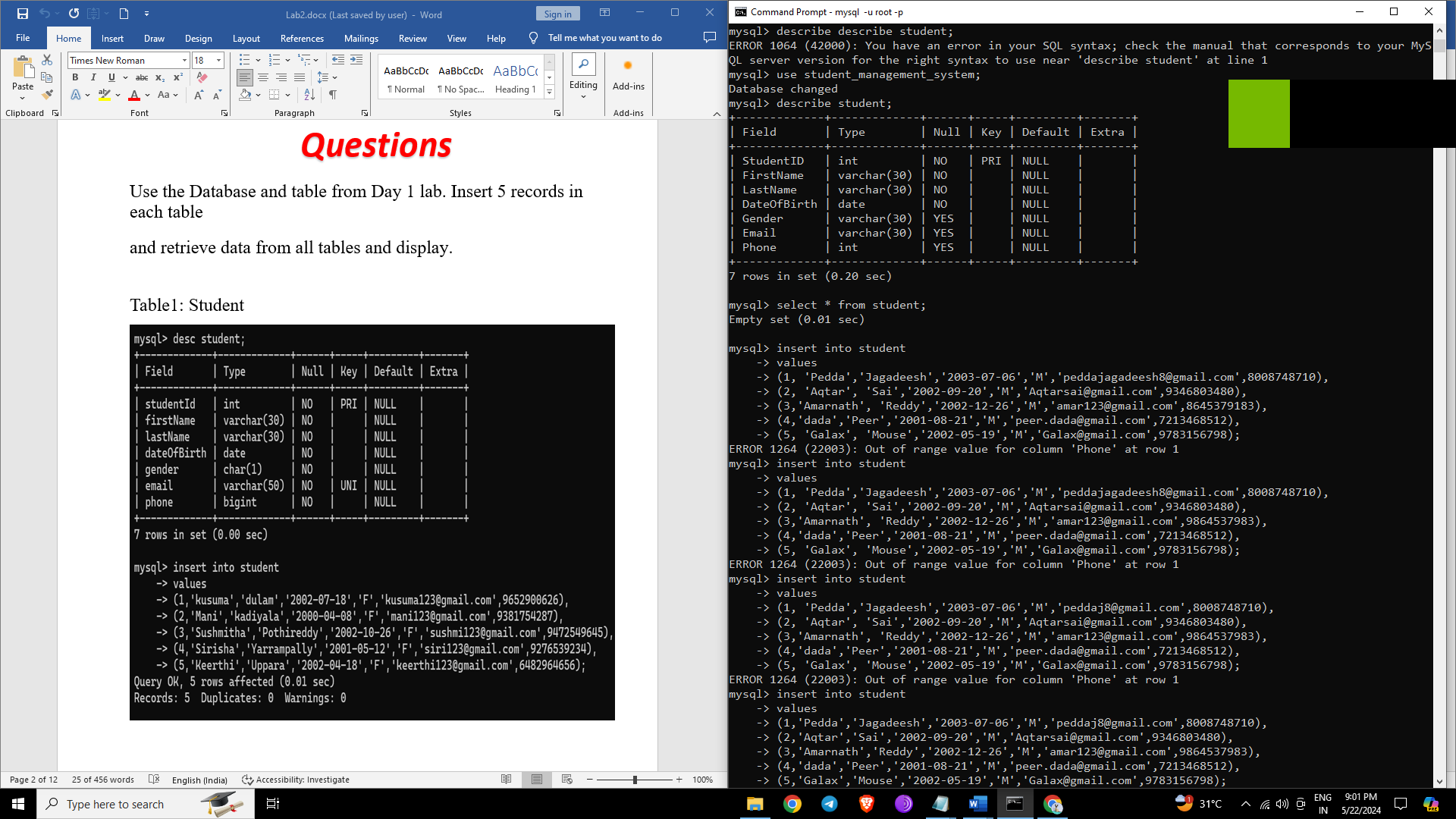
AF0366969

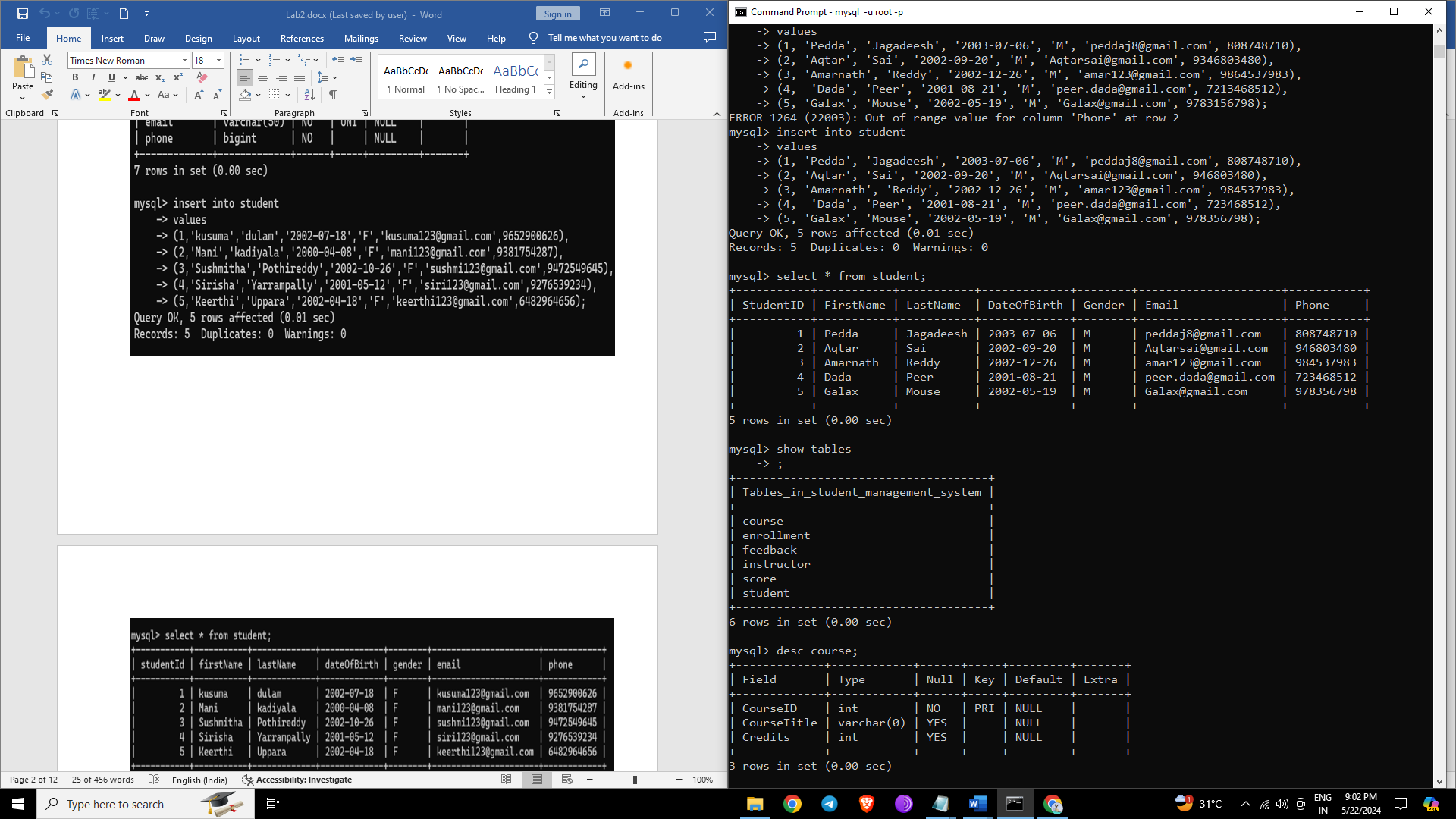
QUESTIONS

Use the Database and table from Day 1 lab. Insert 5 records in each table

and retrieve data from all tables and display.

Table1: Student





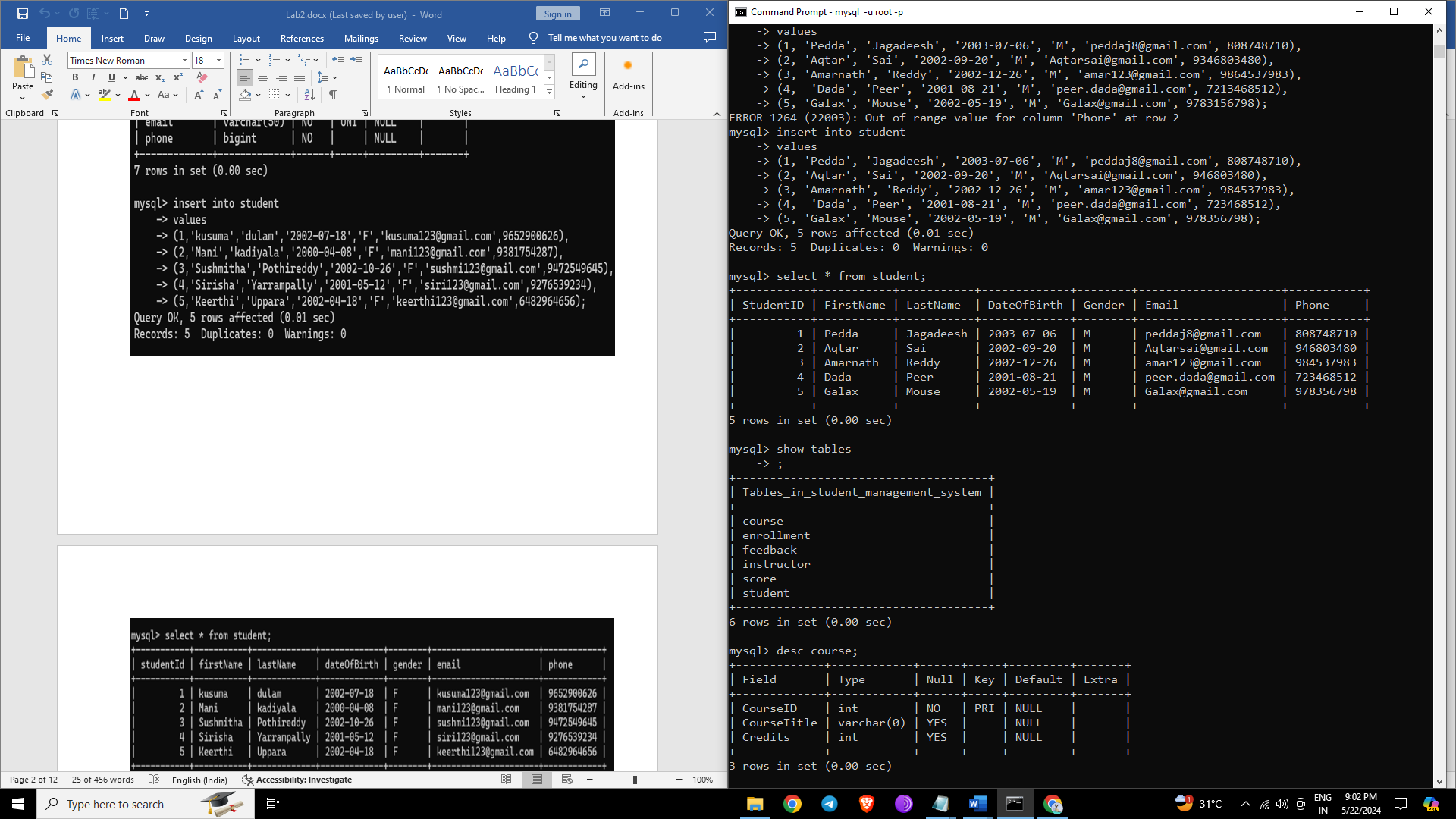
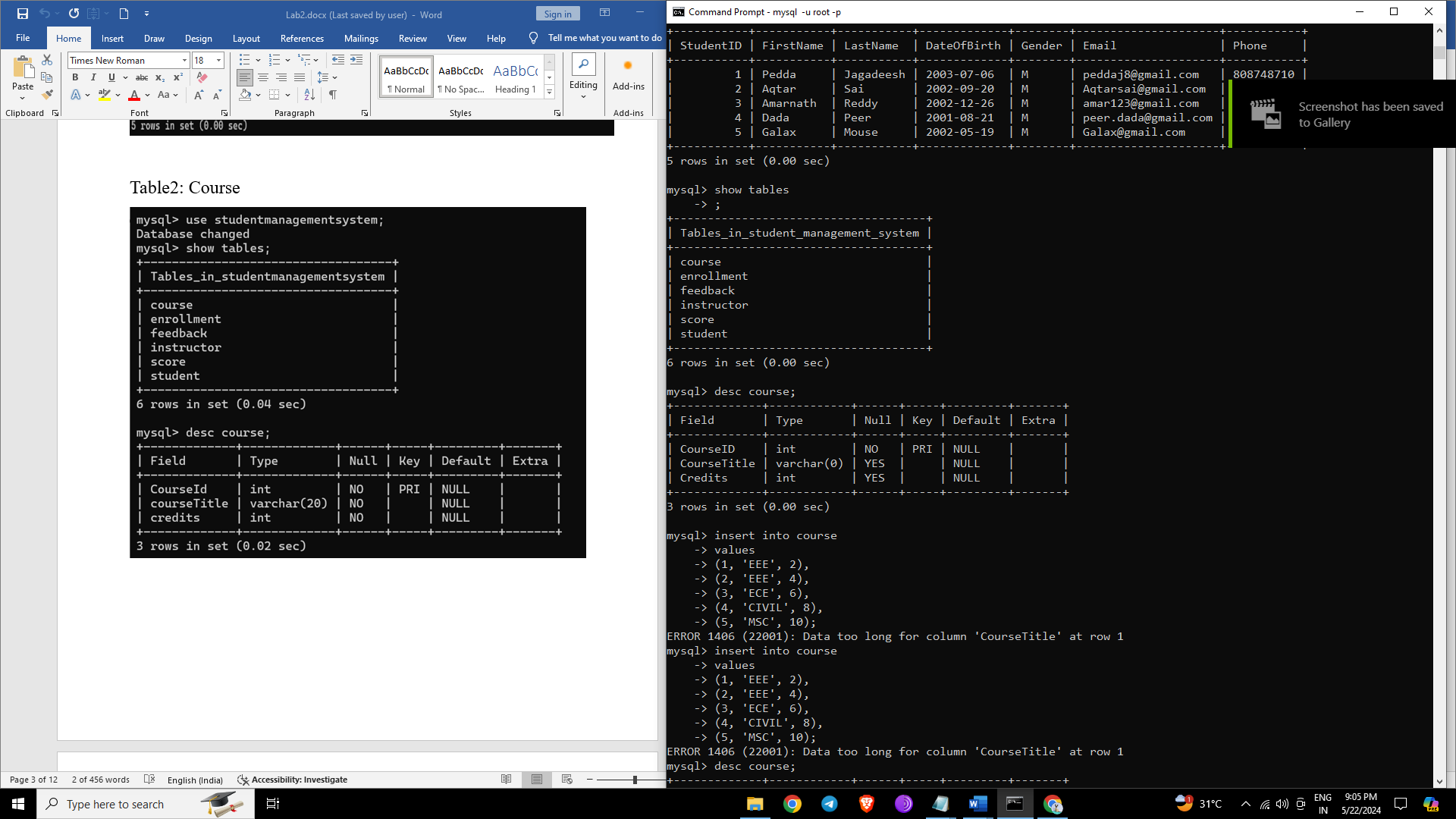
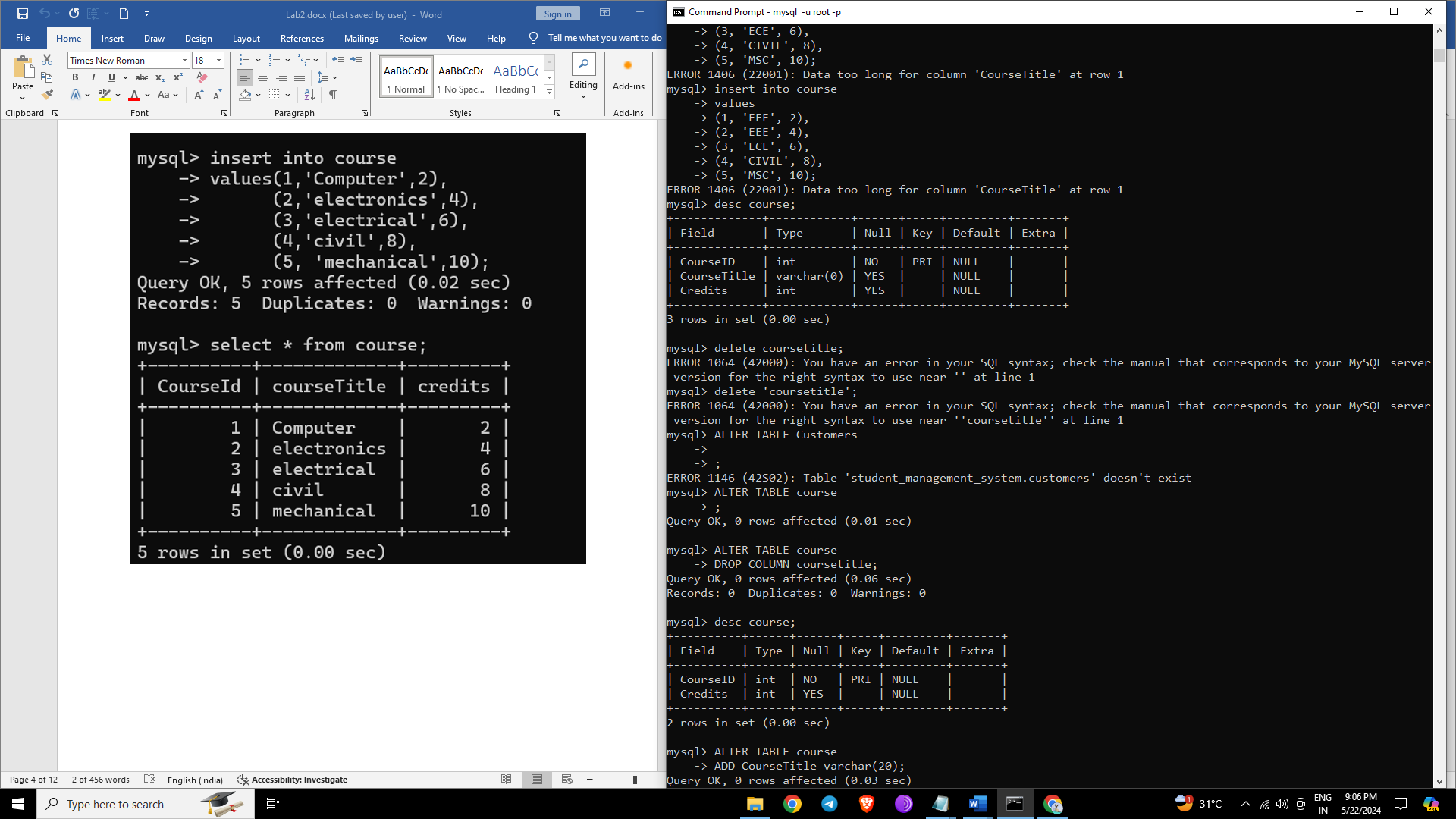


Table2: Course





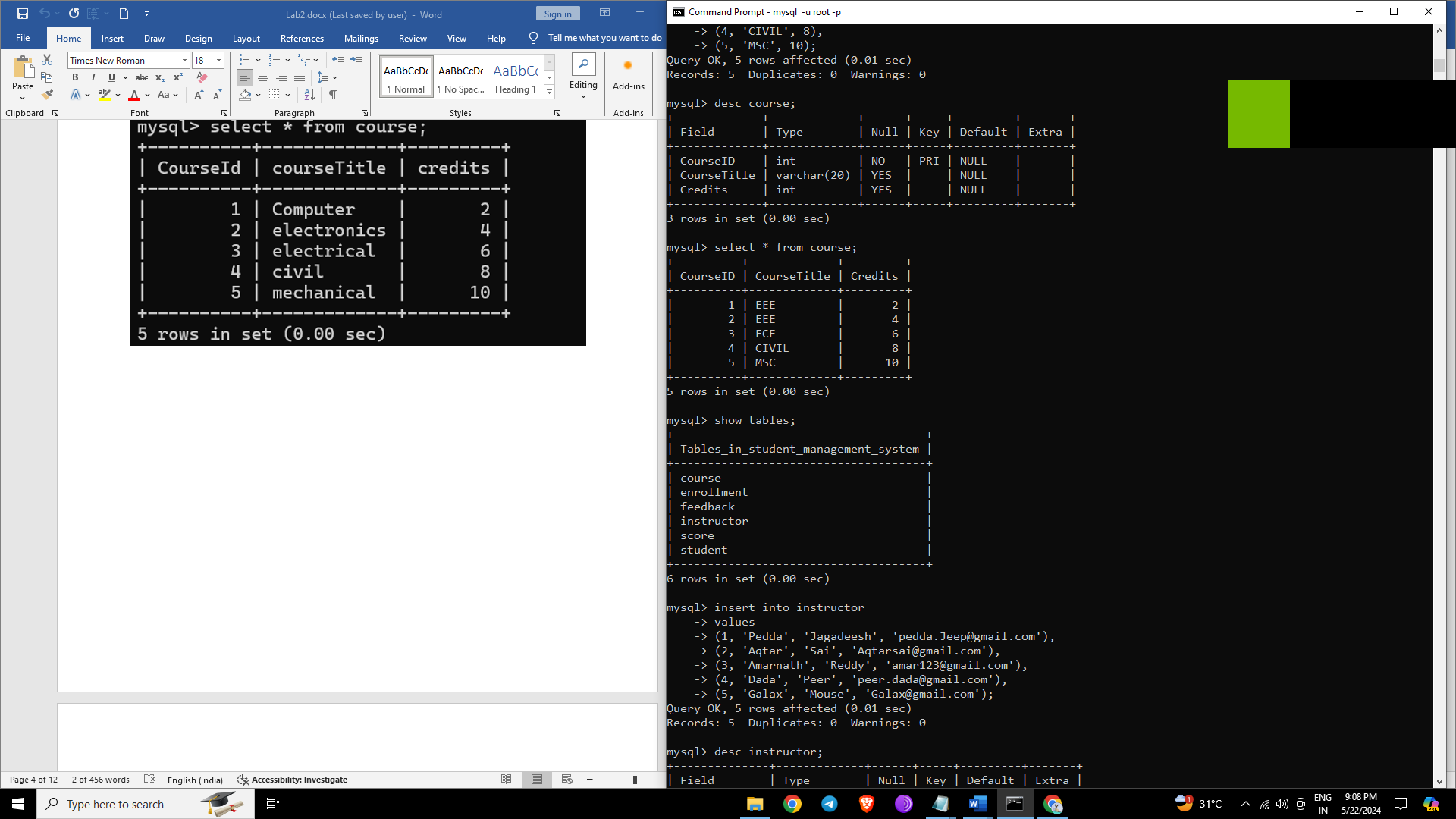


Table3: Instructor

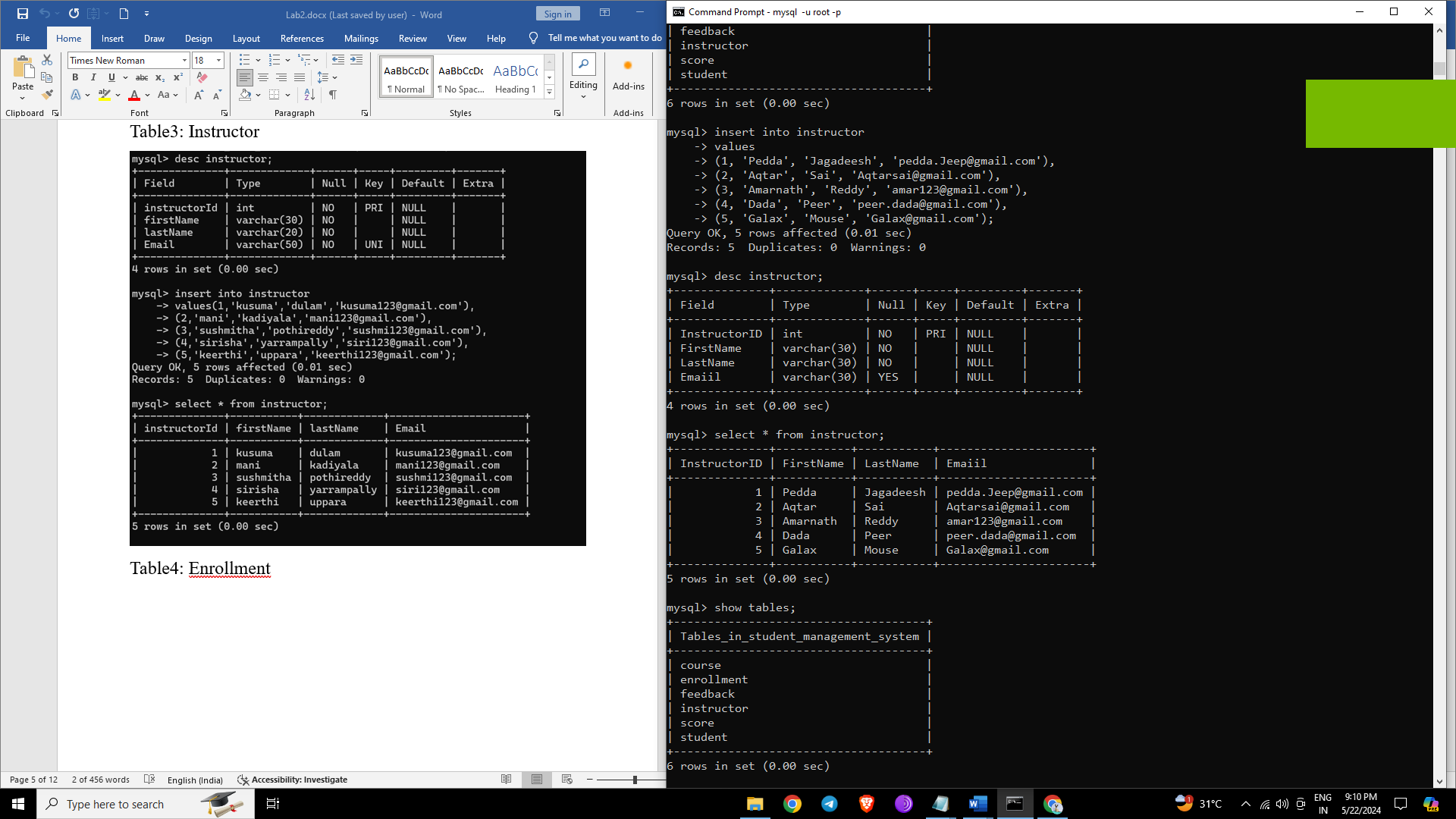


Table4: Enrollment

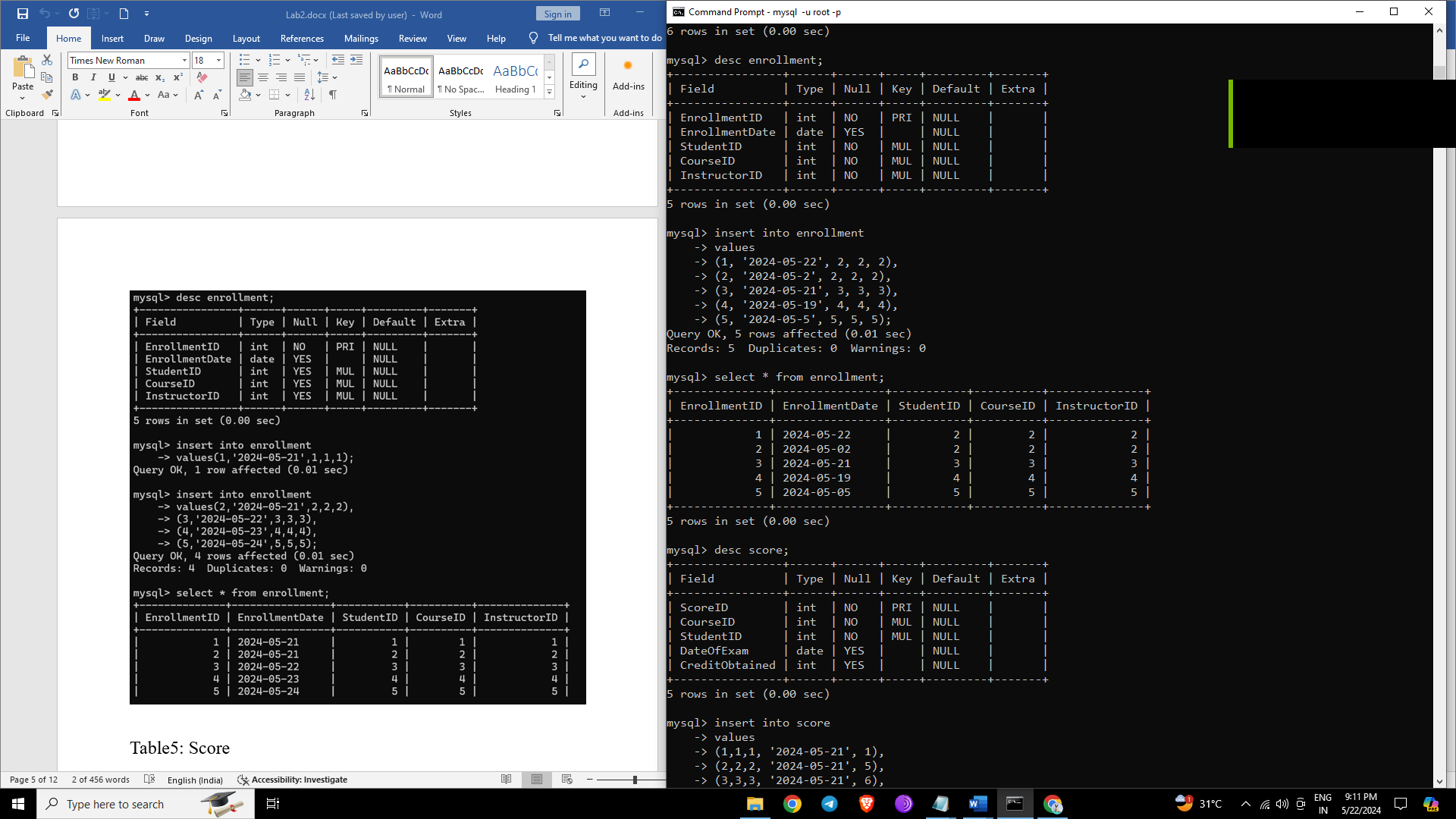


Table5: Score

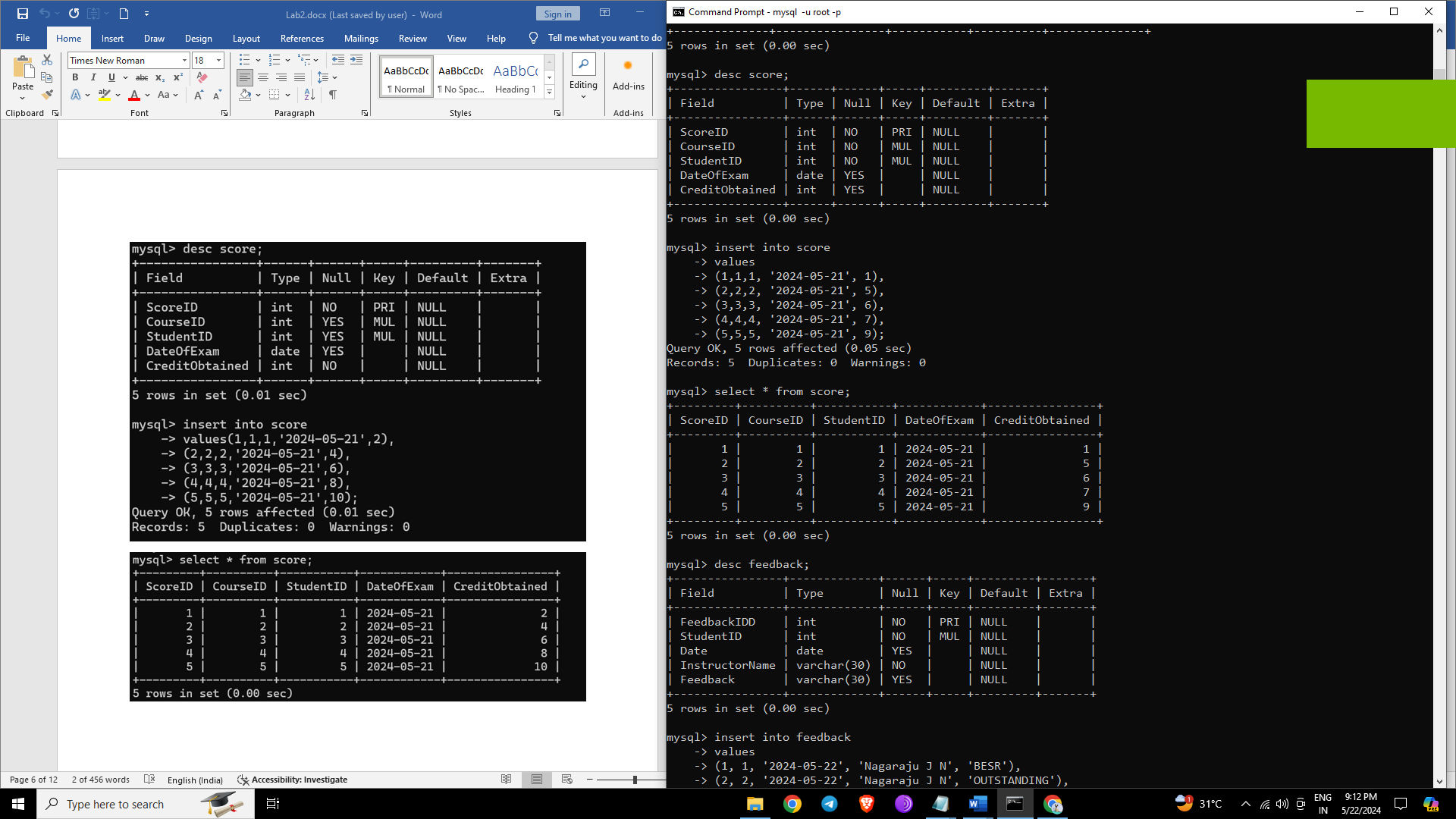
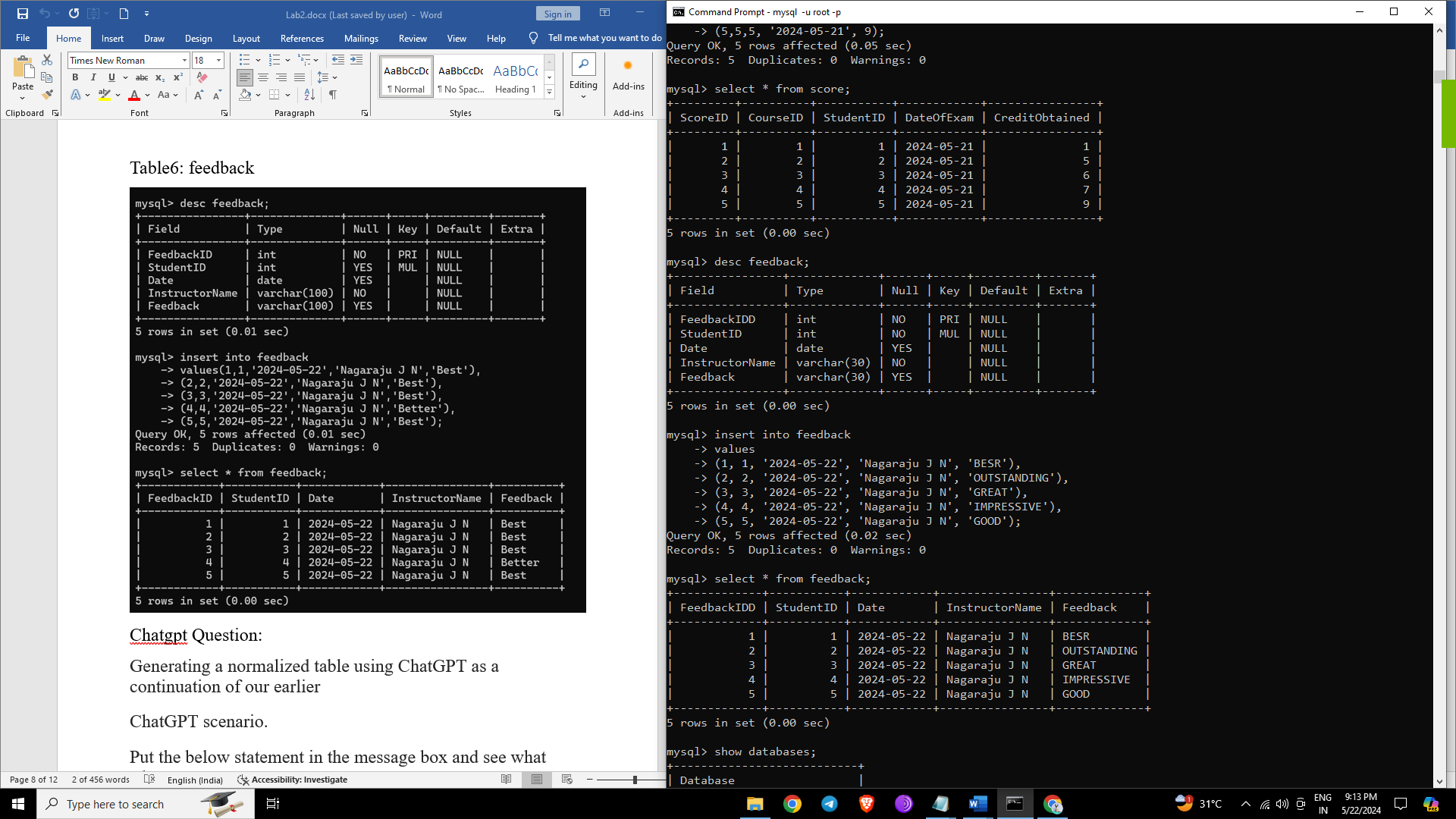


Table6: feedback



Chatgpt Question:

Generating a normalized table using ChatGPT as a continuation of our earlier

ChatGPT scenario.

Put the below statement in the message box and see what ChatGPT says:

I have created a database of emergency contact information. Kindly help me by

providing normalized tables with understanding.

I got following information from chatgpt

Creating a normalized database for emergency contact information involves organizing the data into multiple related tables to reduce redundancy and improve data integrity. Here’s an outline of what the normalized tables could look like along with explanations for each.

**Here's how you can structure and normalize a database for emergency contact information:**

**### Step 1: Identify the Entities and Attributes**

**1. \*\*Individuals\*\***

**- ID**

**- First Name**

**- Last Name**

**- Date of Birth**

**- Address**

**2. \*\*Contacts\*\***

**- ID**

**- Individual ID (Foreign Key)**

**- Contact Type (e.g., family, friend, coworker)**

**- Contact Name**

**- Phone Number**

**- Email**

**3. \*\*Addresses\*\***

**- ID**

**- Street**

**- City**

**- State**

**- Zip Code**

**### Step 2: Create the Normalized Tables**

**1. \*\*Individuals Table\*\***

**- \*\*individual\_id\*\* (Primary Key)**

**- first\_name**

**- last\_name**

**- date\_of\_birth**

**- address\_id (Foreign Key)**

**2. \*\*Contacts Table\*\***

**- \*\*contact\_id\*\* (Primary Key)**

**- individual\_id (Foreign Key)**

**- contact\_type**

**- contact\_name**

**- phone\_number**

**- email**

**3. \*\*Addresses Table\*\***

**- \*\*address\_id\*\* (Primary Key)**

**- street**

**- city**

**- state**

**- zip\_code**

**### Step 3: Define Relationships**

**1. \*\*Individuals and Contacts\*\*: Each individual can have multiple contacts, but each contact belongs to only one individual.**

**- Relationship: One-to-Many (Individual to Contacts)**

**2. \*\*Individuals and Addresses\*\*: Each individual has one address, but multiple individuals can share the same address.**

**- Relationship: Many-to-One (Individual to Address)**

**### Step 4: Example Data**

**\*\*Individuals Table\*\*:**

**| individual\_id | first\_name | last\_name | date\_of\_birth | address\_id |**

**|---------------|------------|-----------|---------------|------------|**

**| 1 | John | Doe | 1980-05-15 | 101 |**

**| 2 | Jane | Smith | 1990-07-20 | 102 |**

**\*\*Contacts Table\*\*:**

**| contact\_id | individual\_id | contact\_type | contact\_name | phone\_number | email |**

**|------------|---------------|--------------|--------------|--------------|-------------------|**

**| 1 | 1 | Family | Mary Doe | 123-456-7890 | mary.doe@example.com |**

**| 2 | 1 | Friend | Bob Johnson | 987-654-3210 | bob.johnson@example.com |**

**| 3 | 2 | Family | Sara Smith | 555-555-5555 | sara.smith@example.com |**

**\*\*Addresses Table\*\*:**

**| address\_id | street | city | state | zip\_code |**

**|------------|------------------|----------|-------|----------|**

**| 101 | 123 Main St | Cityville| CA | 90001 |**

**| 102 | 456 Elm St | Townsville| NY | 10001 |**

**### Explanation**

**1. \*\*Individuals Table\*\*: Stores personal details about each individual and references their address through `address\_id`.**

**2. \*\*Contacts Table\*\*: Contains information about the emergency contacts associated with individuals, referencing the individual via `individual\_id`.**

**3. \*\*Addresses Table\*\*: Holds the address information, which can be linked to multiple individuals.**

**By normalizing the tables in this way, redundancy is minimized (e.g., addresses are stored once and referenced), and data integrity is improved (e.g., ensuring consistent contact information for individuals).**

**Would you like to proceed with any specific queries or further details on any part of this normalization process?**