**COMPUTER SCIENCE DEPARTMENT**

LAB: Database System

**Lab Task # 02**

**Last date of Submission: 4th October 2024**

# Submitted To: Mam Kashia Riaz

**Student Name: UBAID-BIN-WARIS**

# Reg. Number: 2212416

# 

**Task 1:** Students will select the topic of their own choice and gather information regarding

that topic.

The provided SQL script creates a database structure for a table named user\_details and inserts sample data into it. The table contains various columns such as user\_id, username, first\_name, last\_name, gender, password, and status. The user\_id is the primary key and is set to auto-increment, ensuring each user has a unique identifier. The other fields hold basic information about users, such as their names, gender, and password, with passwords stored in what appear to be hashed forms for security. The status field is used to indicate whether a user is active (represented by a 1). Additionally, the table uses the MyISAM storage engine, which is suited for applications focused more on read-heavy operations, and the default character set is Latin-1. The script also inserts 89 records of users with diverse attributes into the table.

**Task 2:** Identify Entity, Attribute and Relationship among them

### **Entities**

1. **User**
   * This is the main entity represented by the table user\_details. Each record in the table represents a user.

### **Attributes**

Attributes are the properties or characteristics of an entity. For the **User** entity, the attributes are:

1. **user\_id**: A unique identifier for each user (Primary Key).
2. **username**: The username chosen by the user, used for login or identification.
3. **first\_name**: The user's first name.
4. **last\_name**: The user's last name.
5. **gender**: The user's gender.
6. **password**: The user's password, likely hashed for security.
7. **status**: The user's account status, where 1 represents active users and 0 represents inactive users.

### **Relationships**

Since the script only defines a single table (user\_details), there are no explicit relationships between multiple entities. If there were additional tables (e.g., an "Orders" table for users), relationships like "One-to-Many" (one user can have many orders) or "Many-to-Many" (many users can have many roles) could be identified. However, in this case:

* **User relationship**: Implicitly, each user has a **status** relationship (active or inactive), which determines their current account state.

**Task 3:** Draw ER diagram in MYSQL

