Lab 8: Implementing Security Protocols in SQL (User Authentication & Role Management, Views, Encryption)

Objectives:

- User Authentication & Role Management (already done in DBS)
- View-based access control for restricting data access,
- SQL injection prevention using secure coding techniques,
- Basic encryption/decryption of sensitive data.

Tool/Software requirements:

- MySQL workbench
- Anaconda

Reference book: Database Management Systems, 3rd Edition by Raghu Ramakrishnan, Johannes Gehrke

Description:

Example and Lab Tasks:

✅ **Part 1: View-Based Access Control**

Setup:

```
CREATE TABLE employees (
  emp_id INT PRIMARY KEY,
  name VARCHAR(50),
  department VARCHAR(50),
  salary DECIMAL(10,2)
);
INSERT INTO employees VALUES
(1, 'Alice', 'IT', 70000),
(2, 'Bob', 'HR', 60000),
(3, 'Charlie', 'IT', 80000);
```

Create a view for HR staff that only shows employee names and departments (no salary info).

```
CREATE VIEW employee_view_hr AS
```

SELECT name, department FROM employees;

-- Assuming a user named hr_user

GRANT SELECT ON employee_view_hr TO hr_user;

✅ Part 2: Data Encryption with MySQL

🔐 Task:

Encrypt user passwords using `AES_ENCRYPT` and decrypt them for verification.

📦 Table Setup:

```
CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50),
    password VARBINARY(255)
);
```

🔏 Insert Encrypted Passwords:

```
-- Replace 'my_secret_key' with your secure key
INSERT INTO users (username, password)
VALUES ('admin', AES_ENCRYPT('admin123', 'my_secret_key'));
```

🔓 Decrypt Passwords:

```
SELECT username, AES_DECRYPT(password, 'my_secret_key') AS decrypted_password
FROM users;
```

📁 Deliverables in Report

- SQL scripts for views, secure queries, and encryption.
- Screenshots of table creation, view outputs, and decrypted results.
- A short report (1–2 pages) answering:
    - What security threats are mitigated?
    - What are best practices in each method?
    - How could this be extended in a real application?

- Why are views better than granting access to base tables?
- Is symmetric encryption sufficient for storing passwords?

Lab Tasks:

Task 1: Create Users and Roles

Goal: Implement user-level access control.

1. **Create two roles:**
   - **role_manager**
   - **role_employee**
2. **Create two users and assign roles:**
   - **user_john → role_manager**
   - **user_anna → role_employee**

Task 2: Define Views for Data Access Control

Goal: Use views to expose only necessary data to specific roles.

1. **Create a view accessible by employees (hide salary and SSN):**

Task 3: Encrypt a Sensitive Field

Goal: Store and retrieve encrypted data.

Rubric:

Total Marks: 10

Criteria	Description	Marks
1. SQL Script Accuracy	- Correct and complete implementation of: ✓ View-based access control ✓ Injection prevention ✓ AES encryption/decryption - Queries should run without errors.	2
2. Explanation & Justification	- Written explanation for each SQL security technique used - Justification of chosen approach with examples and context	1.5
3. Originality & Customization	- Use of unique table/data names (e.g., with student roll no.) - Custom examples not copied directly from class material or AI responses	2
4. Screenshots of Execution	- Clear screenshots of working outputs with timestamps - Must include data insertions, query execution, and encrypted outputs	1

Criteria	Description	Marks
5. Written Reflection Report	- 1–2 pages explaining: ✓ Learning outcomes ✓ Real-world relevance ✓ Possible improvements	1.5
6. Viva/Oral Defense	- Student explains any part of their work in a short viva (live or recorded) - Questions may cover: ✓ Why use views? ✓ How injection works? ✓ How AES works?	2
Total		**10**

-