Database Security Lab

These 5 experiments cover all **major areas of database security**:

1. Access control
2. Injection attacks
3. Auditing
4. Encryption
5. Role-based access control

1. User Management with GRANT and REVOKE

Objective: To study database access control using user roles and privileges.

a) Create Users

```
CREATE USER 'student1'@'localhost' IDENTIFIED BY 'pass123';
CREATE USER 'student2'@'localhost' IDENTIFIED BY 'pass123';
```

b) Grant Privileges

```
GRANT ALL PRIVILEGES ON labdb.* TO 'student1'@'localhost';
GRANT SELECT, INSERT ON labdb.* TO 'student2'@'localhost';
```

c) Revoke Privilege

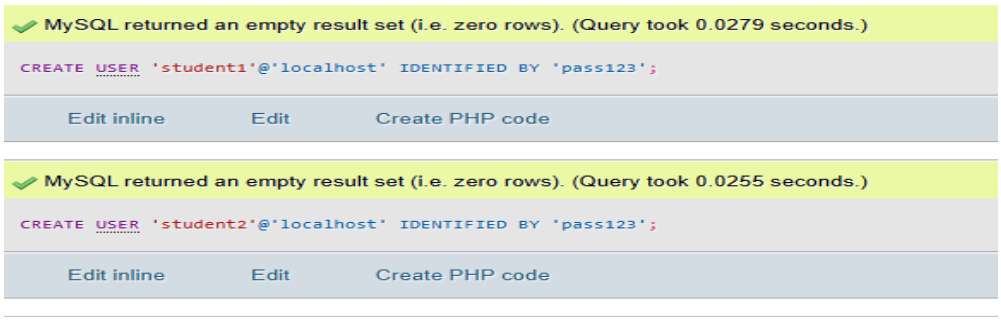
```
REVOKE INSERT ON labdb.* FROM 'student2'@'localhost';
```

d) Verify

```
SHOW GRANTS FOR 'student1'@'localhost';
SHOW GRANTS FOR 'student2'@'localhost';
```

Output:

a)



b)

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0334 seconds.)

```
GRANT ALL PRIVILEGES ON labdb.* TO 'student1'@'localhost';
```

Edit inline Edit Create PHP code

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0274 seconds.)

```
GRANT SELECT, INSERT ON labdb.* TO 'student2'@'localhost';
```

Edit inline Edit Create PHP code

c)

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0394 seconds.)

```
REVOKE INSERT ON labdb.* FROM 'student2'@'localhost';
```

Edit inline Edit Create PHP code

d)

Your SQL query has been executed successfully.

```
SHOW GRANTS FOR 'student1'@'localhost';
```

Profiling [Edit inline] [Edit] [Create PH

Extra options

Grants for student1@localhost

GRANT USAGE ON *.* TO `student1`@`localhost` IDENT...

GRANT ALL PRIVILEGES ON `labdb`.* TO `student1`@`l...

```
SHOW GRANTS FOR 'student2'@'localhost';
```

Extra options

Grants for student2@localhost

GRANT USAGE ON *.* TO `student2`@`localhost` IDENT...

GRANT SELECT ON `labdb`.* TO `student2`@`localhost`...

2. Experiment SQL Injection Demonstration

Objective: To demonstrate SQL injection and its prevention.

```
CREATE DATABASE labdb;
USE labdb;

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


INSERT INTO users (username, password) VALUES
('admin', 'admin123'),
('student', 'stud123');
```

Injection Example:

Input:

Username: admin

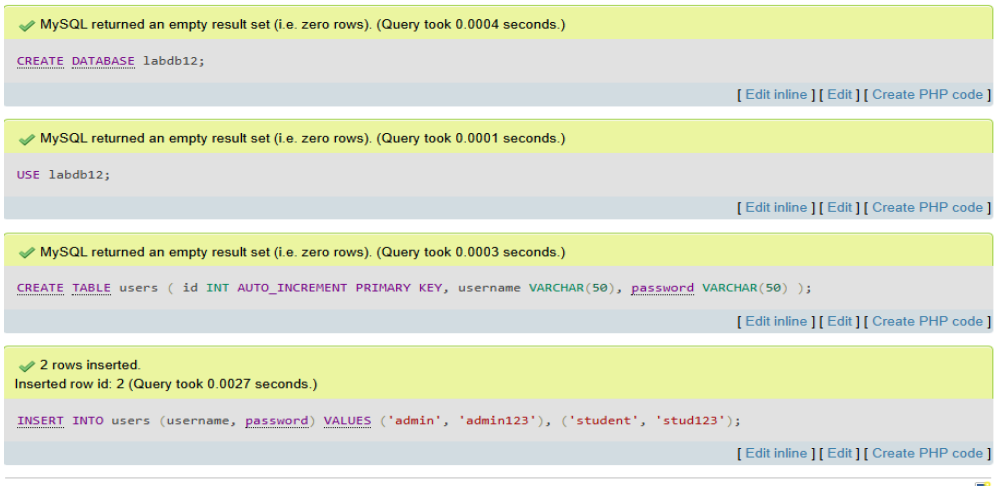
Password: ' OR '1'='1

Query becomes:

SELECT * FROM users WHERE username='admin' AND password='' OR '1'='1';

Bypasses authentication.

Prevention: Use prepared statements (parameterized queries)



3. Experiment Database Auditing

Objective: To enable logging and track changes to the database.

Procedure:

-- Enable general log (MySQL)

```
SET GLOBAL general_log = 'ON';
SET GLOBAL general_log_output = 'TABLE';
```

-- View logged queries

```
SELECT * FROM mysql.general_log ORDER BY event_time DESC LIMIT 10;
```

Output:

4. Experiment 4: Encryption in Database

Objective: To secure data using encryption functions.

a) CREATE TABLE secure_data (

id INT AUTO_INCREMENT PRIMARY KEY,
secret VARBINARY(255)
);

b) Insert encrypted data (AES)

INSERT INTO secure_data(secret) VALUES (AES_ENCRYPT('mysecretpassword', 'key123'));

c) Decrypt data

SELECT AES_DECRYPT(secret, 'key123') AS decrypted_value FROM secure_data;

Output:

5: Role-Based Access Control (RBAC)

Objective: To implement RBAC in a database.

Procedure:

a) Create a role

```
CREATE ROLE 'manager';
```

b) Assign privileges to role

```
GRANT SELECT, UPDATE ON labdb.* TO 'manager';
```

c) Create user and assign role

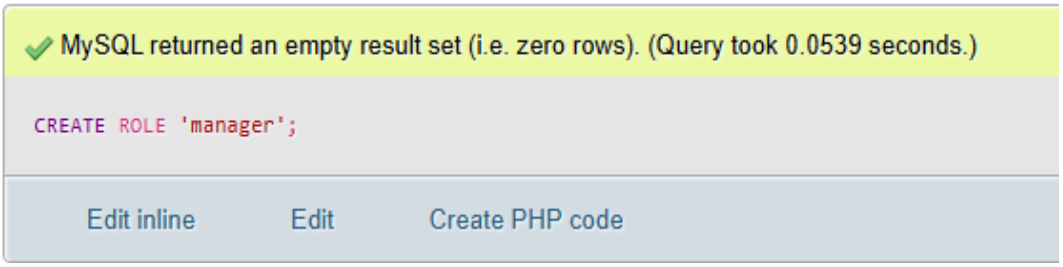
```
CREATE USER 'alice'@'localhost' IDENTIFIED BY 'alice123';
GRANT 'manager' TO 'alice'@'localhost';
```

d) Verify

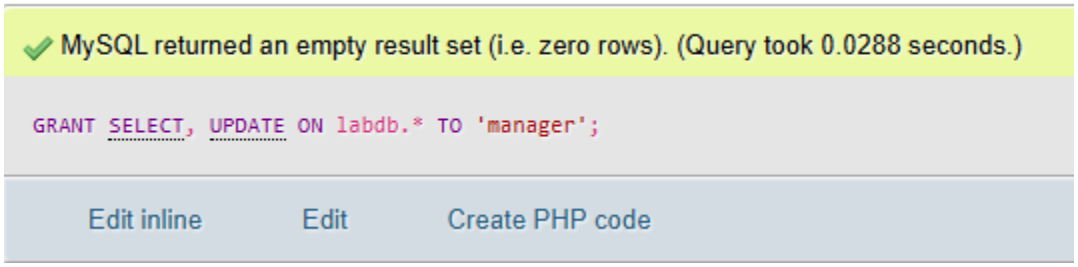
```
SHOW GRANTS FOR 'alice'@'localhost';
```

Output:

a)



b)



c)

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0264 seconds.)

```
CREATE USER 'alice'@'localhost' IDENTIFIED BY 'alice123';
```

Edit inline Edit Create PHP code

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0394 seconds.)

```
GRANT 'manager' TO 'alice'@'localhost';
```

Edit inline Edit Create PHP code

d)

Your SQL query has been executed successfully.

```
SHOW GRANTS FOR 'alice'@'localhost';
```

Profiling Edit inline Edit Create PHP code Refresh

Extra options

Grants for alice@localhost
GRANT USAGE ON *.* TO `alice`@`localhost`
GRANT `manager`@`%` TO `alice`@`localhost`