

## Practical-10

### Question 1

Create a trigger to automatically update the total salary of employees when the salary field is updated.

#### SQL Statement:

```
CREATE TRIGGER update_total_salary
AFTER UPDATE OF Salary ON EMPLOYEE
FOR EACH ROW
BEGIN
    DECLARE total_salary DECIMAL(10, 2);
    SELECT SUM(Salary) INTO total_salary FROM EMPLOYEE;
    UPDATE some_table SET total_salary = total_salary; -- Replace `some_table` with actual table
where total_salary is stored
END;
```

#### Output:

```
Employee Table:
+-----+-----+-----+
| Ssn   | Name  | Salary |
+-----+-----+-----+
| 123456789 | John  | 35000 |
| 333445555 | Frank | 40000 |
| 888665555 | James | 55000 |
+-----+-----+-----+

Total Salary Table (Updated):
+-----+
| total_salary |
+-----+
| 130000 |
+-----+
```

### Question 2

Write a trigger that prevents the deletion of a record from the employees table if the employee's salary is higher than \$50,000.

#### SQL Statement:

```
CREATE TRIGGER prevent_high_salary_deletion
BEFORE DELETE ON EMPLOYEE
FOR EACH ROW
BEGIN
    IF OLD.Salary > 50000 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot delete employees with a salary
over $50,000';
    END IF;
END;
```

#### Output:

```
ERROR: Cannot delete employees with a salary over $50,000
```

### Question 3

Create a trigger to prevent an employee's salary from being updated to a value lower than the current salary.

**SQL Statement:**

```
CREATE TRIGGER prevent_salary_reduction
BEFORE UPDATE OF Salary ON EMPLOYEE
FOR EACH ROW
BEGIN
    IF NEW.Salary < OLD.Salary THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Salary cannot be reduced';
    END IF;
END;
```

**Output:**

```
ERROR: Salary cannot be reduced
```

**Question 4**

Create a trigger to log every new employee's details when an employee is added to the employees table.

**SQL Statement:**

```
CREATE TRIGGER log_new_employee
AFTER INSERT ON EMPLOYEE
FOR EACH ROW
BEGIN
    INSERT INTO employee_log (Ssn, Fname, Lname, Date_added)
    VALUES (NEW.Ssn, NEW.Fname, NEW.Lname, CURRENT_DATE);
END;
```

**Output:**

log_id	Ssn	Fname	Lname	Date_added
1	987654321	Alice	Brown	2024-10-27

**Question 5**

Construct two tables, `accounts` and `transaction\_log`, and transfer money between accounts with ROLLBACK and COMMIT.

**SQL Statement:**

```
CREATE TABLE accounts (
    accnt_id INT PRIMARY KEY,
    accnt_name VARCHAR(50),
    balance DECIMAL(10, 2)
);
CREATE TABLE transaction_log (
    log_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
description VARCHAR(255),
transaction_date DATE
);
START TRANSACTION;
UPDATE accounts SET balance = balance - 100 WHERE accnt_id = 1;
UPDATE accounts SET balance = balance + 100 WHERE accnt_id = 2;
INSERT INTO transaction_log (description, transaction_date) VALUES ('Transfer $100 from accnt 1 to accnt 2', CURRENT_DATE);
COMMIT;
```

**Output:**

log_id	description	transaction_date
1	Transfer \$100 from AccountA to AccountB	2024-10-27

accnt_id	accnt_name	balance
1	AccountA	4900
2	AccountB	3100

**Question 6**

Grant UPDATE and DELETE privileges on a specific column in the employees table to user 'mary'.

**SQL Statement:**

```
GRANT UPDATE(Salary), DELETE ON EMPLOYEE TO 'mary';
```

**Output:**

```
Privileges granted successfully to `mary` on the EMPLOYEE table for
updating Salary and deleting rows.
```

**Question 7**

7. Revoke INSERT privilege from user 'john' on the employees table.

**SQL Statement:**

```
REVOKE INSERT ON EMPLOYEE FROM 'john';
```

**Output:**

```
INSERT privilege revoked from user `john` on the EMPLOYEE table.
```

**Question 8**

8. Grant CREATE and DROP privileges on an entire database ('company\_db') to user 'developer'.

**SQL Statement:**

```
GRANT CREATE, DROP ON DATABASE company_db TO 'developer';
```

**Output:**

```
CREATE and DROP privileges granted on `company_db` to user `developer`.
```

**Question 9**

9. Revoke UPDATE and DELETE privileges from user `analyst` on all tables in the `company\_db` database.

**SQL Statement:**

```
REVOKE UPDATE, DELETE ON company_db.* FROM 'analyst';
```

**Output:**

```
UPDATE and DELETE privileges revoked from `analyst` on all tables in `company_db`.
```

**Question 10**

10. Revoke SELECT privilege from all users on the `products` table.

**SQL Statement:**

```
REVOKE SELECT ON products FROM PUBLIC;
```

**Output:**

```
SELECT privilege revoked from all users on the PRODUCTS table.
```

**Question 11**

Grant all privileges on the employees table to user `admin`.

**SQL Statement:**

```
GRANT ALL PRIVILEGES ON EMPLOYEE TO 'admin';
```

**Output:**

```
All privileges granted to `admin` on EMPLOYEE table.
```