Lab3 Advanced Sql

Q1: Insert new student and his score in exam in different subjects as transaction.

```
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO student (id, fName, lName, email, address, gender, birthDate)
    -> VALUES (6, 'Alice', 'Brown', 'alice.brown@example.com', '789 Elm Street', 'female', '2002-05-15');
Query OK, 1 row affected (0.04 sec)
```

```
mysql> INSERT INTO exam (stuId, subId, stuScore) values
    ->      (6, 103, 88),(6,102,90),(6,101,9);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> COMMIT;
Query OK, 0 rows affected (0.01 sec)
```

Q2: Display the date of exam as the following: day 'month name' year.

```
mysql> SELECT DATE_FORMAT(examDate, '%d \'%M\' %Y') AS formattedExamDate from exam;
+--------------------+
| formattedExamDate  |
+--------------------+
| 01 'January' 2024  |
| 02 'January' 2024  |
| 03 'January' 2024  |
| 04 'January' 2024  |
| 06 'May' 2024      |
| 26 'May' 2024      |
| 02 'May' 2024      |
+--------------------+
7 rows in set (0.00 sec)
```

Q3: Display name and age of each students

```
mysql> SELECT
    ->     CONCAT(fName, ' ', lName) AS fullName,
    ->     TIMESTAMPDIFF(YEAR, birthDate, CURDATE()) AS age
    -> FROM student;
+---------------+------+
| fullName      | age  |
+---------------+------+
| John Doe      |   23 |
| Jane Smith    |   22 |
| Emily Davis   |   21 |
| Michael Brown |   24 |
| Alice Brown   |   22 |
+---------------+------+
5 rows in set (0.01 sec)
```

Q4: Display the name of students with their Rounded score in each Exam

```
mysql> SELECT
    -> CONCAT(st.fName, ' ', st.lName) AS fullName,
    -> s.name AS subjectName,
    -> ROUND(e.stuScore, 0) AS roundedScore
    -> FROM student st
    -> JOIN exam e ON st.id = e.stuId
    -> JOIN subject s ON e.subId = s.id;
+---------------+-------------+--------------+
| fullName      | subjectName | roundedScore |
+---------------+-------------+--------------+
| John Doe      | Mathematics |           85 |
| Alice Brown   | Mathematics |            9 |
| Jane Smith    | Physics     |           90 |
| Alice Brown   | Physics     |           90 |
| Emily Davis   | Chemistry   |           78 |
| Alice Brown   | Chemistry   |           88 |
| Michael Brown | Biology     |           88 |
+---------------+-------------+--------------+
7 rows in set (0.00 sec)
```

Q5: Display the name of students with the year of Birthdate

```
mysql> select Concat(fName,' ',lName) as fullName,YEAR(birthDate) as yearOfBirth from student;
+---------------+-------------+
| fullName      | yearOfBirth |
+---------------+-------------+
| John Doe      |        2001 |
| Jane Smith    |        2002 |
| Emily Davis   |        2003 |
| Michael Brown |        2000 |
| Alice Brown   |        2002 |
+---------------+-------------+
5 rows in set (0.00 sec)
```

Q6: Add new exam result, in date column use NOW

```
mysql> INSERT INTO exam (stuId, subId, stuScore, examDate) VALUES (5, 104, 95, NOW());
Query OK, 1 row affected, 1 warning (0.01 sec)
```

Q7: Create Hello world function which take username and return welcome message to user using his name

```
mysql> DELIMITER $
mysql> CREATE FUNCTION HelloWorld(username VARCHAR(100))
    -> RETURNS VARCHAR(200)
    -> DETERMINISTIC
    -> BEGIN    RETURN CONCAT('Welcome, ', username, '!');
    -> END $
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> DELIMITER ;
mysql> Select HelloWorld("esraa");
+---------------------+
| HelloWorld("esraa") |
+---------------------+
| Welcome, esraa!     |
+---------------------+
1 row in set (0.01 sec)
```

Q8: Create multiply function which take two number and return the multiply of them

```
mysql> CREATE FUNCTION MultiplyNumbers(num1 DOUBLE, num2 DOUBLE)
    -> RETURNS DOUBLE
    -> DETERMINISTIC
    -> BEGIN
    ->     RETURN num1 * num2;
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> Select MultiplyNumbers(20,30
    -> );
+-------------------------+
| MultiplyNumbers(20,30
) |
+-------------------------+
|                     600 |
+-------------------------+
1 row in set (0.00 sec)
```

Q9: Create function which takes student id and Exam id and return score the student in Exam.

```
mysql> DELIMITER $$
mysql> CREATE FUNCTION GetStudentScore(stuId INT, subId INT)
    -> RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->  return (SELECT stuScore FROM exam  WHERE stuId = stuId AND subId = subId);
    -> END$$
Query OK, 0 rows affected (0.01 sec)
```

Q10: Create function which takes Exam id and return the number of students who failed in a Exam (Score less than 50).

```
mysql> CREATE FUNCTION CountFailedStudents(subId INT)
    -> RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->  RETURN (SELECT COUNT(*) FROM exam WHERE subId = subId AND stuScore < 50);
    -> END$$
Query OK, 0 rows affected (0.01 sec)
```

Q11: Create function which take subject name and return the average of max grades for subject

```
mysql> DELIMITER $$
mysql> CREATE FUNCTION AverageExamGrade(subjectName VARCHAR(100))
    -> RETURNS DOUBLE
    -> DETERMINISTIC
    -> BEGIN
    ->  RETURN (SELECT AVG(e.stuScore) FROM exam e JOIN subject s ON e.subId = s.id WHERE s.name = subjectName);
    -> END$$
Query OK, 0 rows affected (0.01 sec)
```

Q12: Create Table called Deleted_Students which will hold the deleted students info(same columns as in student tables)

```
mysql> CREATE TABLE Deleted_Students (
    ->     Id INT PRIMARY KEY,
    ->     email VARCHAR(100),
    ->     address VARCHAR(255),
    ->     gender ENUM('male', 'female'),
    ->     birthDate DATE,
    ->     fName VARCHAR(50),
    ->     lName VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.07 sec)
```

Q13: Create trigger to save the deleted student from Student table to Deleted_Students.

```
mysql> CREATE TRIGGER AfterStudentDelete
    -> AFTER DELETE ON student
    -> FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO Deleted_Students (Id, email, address, gender, birthDate, fName, lName)
    ->     VALUES (OLD.Id, OLD.email, OLD.address, OLD.gender, OLD.birthDate, OLD.fName, OLD.lName);
    -> END$$
Query OK, 0 rows affected (0.02 sec)
```

Q14: Create trigger to save the newly added students to Student table to Backup_Students.

```
mysql> CREATE TABLE Backup_Students (
    ->     Id INT PRIMARY KEY,
    ->     email VARCHAR(100),
    ->     address VARCHAR(255),
    ->     gender ENUM('male', 'female'),
    ->     birthDate DATE,
    ->     fName VARCHAR(50),
    ->     lName VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> CREATE TRIGGER AfterStudentInsert
    -> AFTER INSERT ON student
    -> FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO Backup_Students (Id, email, address, gender, birthDate, fName, lName)
    ->     VALUES (NEW.Id, NEW.email, NEW.address, NEW.gender, NEW.birthDate, NEW.fName, NEW.lName);
    -> END$$
Query OK, 0 rows affected (0.02 sec)
```

Q15: (Bouns) Create trigger to keep track the changes of contact info table (add/update rows); it will logs the time of action and description of action to another table.

1.create table

```
mysql> CREATE TABLE ContactInfo (
    ->   id INT AUTO_INCREMENT PRIMARY KEY,    action_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   action_type ENUM('INSERT', 'UPDATE') NOT NULL,    description VARCHAR(255) NOT NULL);
Query OK, 0 rows affected (0.03 sec)
```

2.trigger insert

```
mysql> DELIMITER $$
mysql> CREATE TRIGGER AfterPhoneInsert
    -> AFTER INSERT ON phone
    -> FOR EACH ROW
    -> BEGIN
    ->  INSERT INTO ContactInfo (action_type, description)    VALUES ('INSERT', CONCAT('Inserted
 new contact info with ID: ', NEW.stuId));
    -> END$$
Query OK, 0 rows affected (0.02 sec)
```

3.trigger update

```
mysql> CREATE TRIGGER AfterPhoneUpdate
    -> AFTER UPDATE ON phone
    -> FOR EACH ROW
    -> BEGIN
    -> INSERT INTO ContactInfo (action_type, description)    VALUES ('UPDATE', CONCAT('Updated c
ontact info with ID: ', OLD.stuId, ' to new values.'));
    -> END$$
Query OK, 0 rows affected (0.01 sec)
```

Q16: Dump your database into SQL file.

```
C:\Users\mass>mysqldump -u root -p advancedsql > advancedSqlLabDatabase.sql
Enter password: ****
```

advancedSqlLabDatabase.sql ٢٠٢٥/٠١/١٣ م ٠٨:٠٨ SQL Text File 15 KB

Q17: Dump Students table into file.

studentTable.sql ٢٠٢٥/٠١/١٣ م ٠٨:١١ SQL Text File 5 KB

```
C:\Users\mass>mysqldump -u root -p advancedsql student> studentTable.sql
Enter password: ****
```

Q18: Import SQL file into your backup database (Grading_Backup Database)

```
mysql> create database backup_database
    -> ;
Query OK, 1 row affected (0.01 sec)
```

```
C:\Users\mass>mysql -u root -p backup_database < advancedSqlLabDatabase.sql
Enter password: ****
```

```
mysql> show tables
    -> ;
+---------------------------+
| Tables_in_backup_database |
+---------------------------+
| backup_students           |
| contactinfo               |
| deleted_students          |
| exam                      |
| phone                     |
| student                   |
| subject                   |
+---------------------------+
7 rows in set (0.00 sec)
```