SQL TASK – 6 - Creating & Using Views

Overview

This project demonstrates the creation and usage of SQL views to manage and analyze student marks efficiently. The goal is to simplify data retrieval and identify students who have passed all their subjects using reusable views.

1. Create Database and Table

SQL Query:

```
CREATE DATABASE Student_TASK;
USE Student_TASK;
CREATE TABLE students_Name(
            student_id INT AUTO_INCREMENT PRIMARY KEY,
            name VARCHAR(50)
            );

```

```
CREATE TABLE Scores (
            student_id INT,
            subject VARCHAR(20),
            score INT,
            FOREIGN KEY (student_id) REFERENCES students_Name(student_id)
            );
```

***Purpose**: To define a normalized schema separating student personal information and their scores.*

***Observation**: Two linked tables are created using foreign keys to maintain referential integrity.*

2. Insert Sample Records

SQL Query:

```
15 •  INSERT INTO students_Name (name) VALUES
16                              ('Abhijith'),
17                              ('Bhavana'),
18                              ('Chandrika'),
19                              ('Dev');

21 •  INSERT INTO Scores VALUES
22        (1, 'Math', 85), (1, 'Science', 90),
23        (2, 'Math', 35), (2, 'Science', 42),
24        (3, 'Math', 70), (3, 'Science', 65),
25        (4, 'Math', 38), (4, 'Science', 39);
26
```

***Purpose**: To populate the tables with realistic sample data for analysis.*

***Observation**: 4 students with different performance levels. Subjects: Math and Science.*

student_id	name
1	Abhijith
2	Bhavana
3	Chandrika
4	Dev

student_id	subject	score
1	Math	85
1	Science	90
2	Math	35
2	Science	42
3	Math	70
3	Science	65
4	Math	38
4	Science	39

3. Task 1: Create View - Student Scores

SQL Query:

```
CREATE VIEW student_scores AS
SELECT sm.student_id, sm.name, m.subject, m.score
FROM students_Name sm
JOIN Scores m ON sm.student_id = m.student_id;
```

```
SELECT * FROM student_scores;
```

***Purpose**: To simplify access to combined student details and their scores.*

***Observation**:*
- *Combines student names and scores across subjects.*
- *View acts as a virtual table for easier analysis.*

Query Explanation:

This view joins the students_name and Scores tables to show each student's name, subject, and score.
It helps to quickly see all subject-wise scores of students.

Result :

student_id	name	subject	score
1	Abhijith	Math	85
1	Abhijith	Science	90
2	Bhavana	Math	35
2	Bhavana	Science	42
3	Chandrika	Math	70
3	Chandrika	Science	65
4	Dev	Math	38
4	Dev	Science	39

4. Task 2: Create View - Passed Students

SQL Query:

```
37 •  CREATE VIEW passed_students AS
38    SELECT sm.student_id, sm.name
39    FROM students_Name sm
40    WHERE sm.student_id NOT IN (
41        SELECT student_id
42        FROM Scores
43        WHERE score < 40
44    );
45
46 •  SELECT * FROM passed_students;
```

***Purpose**: To identify students who scored at least 40 in every subject.*

***Observation**:*
- *Uses a subquery to filter out students with any failing score.*
- *Final view contains only those who passed all subjects.*

***Query Explanation**: This view lists students who passed all subjects (scored 40 or above in every subject). It filters out any student with a score below 40.*

Result:

student_id	name
1	Abhijith
3	Chandrika