Experiment 7

FS19CO042

Title: Write SQL code for creating of View. Perform Insert ,Modify,
Delete records through view, Delete the View.
Working with Nested -Query.

Theory:

1. Simple View

➔Simple View in SQL is the view created by involving only single table. In other words we can say that there is only one base table in case of Simple View in SQL

Syntax :
```
CREATE VIEW view-name AS
SELECT column-name
FROM table-name
WHERE condition
```

Example :
```
CREATE VIEW Brazil_Customers AS
SELECT CustomerName, ContactName
FROM Customers
WHERE Country = 'Brazil';
```

2. Complex view

➔On other hand, Complex View is created by involving more than one table i.e., multiple tables get projected in Complex view.

Syntax :
```
CREATE VIEW view-name AS
SELECT column-name
   FROM    table-name
   JOIN aggregate-function
   GROUP BY    column-name
```

Example :
```
CREATE VIEW dept_income AS
SELECT d.Name as DepartmentName, sum(e.salary) as
TotalSalary
FROM Employees e
JOIN Departments d on e.DepartmentId = d.id
GROUP BY d.Name
```

3. INSERT query syntax and example.

➔ *INSERT* command is used to insert data into the row of a table.

Syntax : INSERT INTO TABLE_NAME (col1, col2, col3,. ...col N)
VALUES (value1, value2, value3,..... valueN);

Example : INSERT INTO fyfs (name, roll) VALUES ("Rupesh", "FS43");

4. UPDATE query syntax and example**.**

➔*UPDATE* command is used to update or modify the value of a column in the table.

Syntax : UPDATE table_name SET [column_name1=value1] [WHERE CONDITION]

Example : UPDATE students SET User_Name = "Rupesh" WHERE Student_Id = '43'

5. DELETE query syntax and example**.**

➔ *DELETE* is used to remove one or more row from a table.

Syntax : DELETE FROM table_name [WHERE condition];

Example : DELETE FROM javatpoint WHERE Author="Rupesh";

6. DROP VIEW

➔*DROP VIEW* command is used to delete the view

Syntax : DROP VIEW *view_name*

Example : DROP VIEW [Brazil Customers]

7. NESTED QUERY

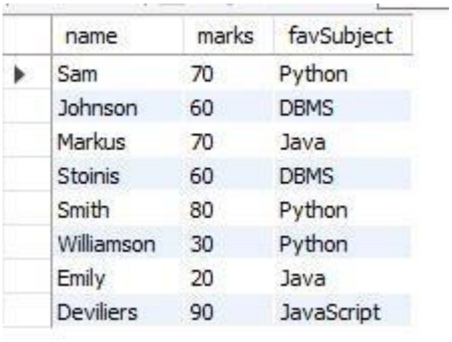
➔Query written inside a query is called as SQL Nested Query

Syntax : SELECT Column1,Column2… From Table_Name
WHERE Column_Name Operator(Select Column1,Column From Table_Name_2)
Operator (Select Column1,Column2…..From Table_Name_3)

Example : ***SELECT Model FROM Product***
WHERE ManufacturerID IN (SELECT ManufacturerID FROM
Manufacturer
WHERE Manufacturer = 'Dell')

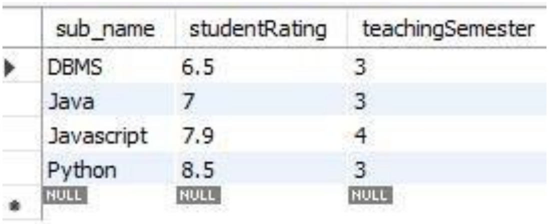
Outputs:
Initial tables setup:

Students table:



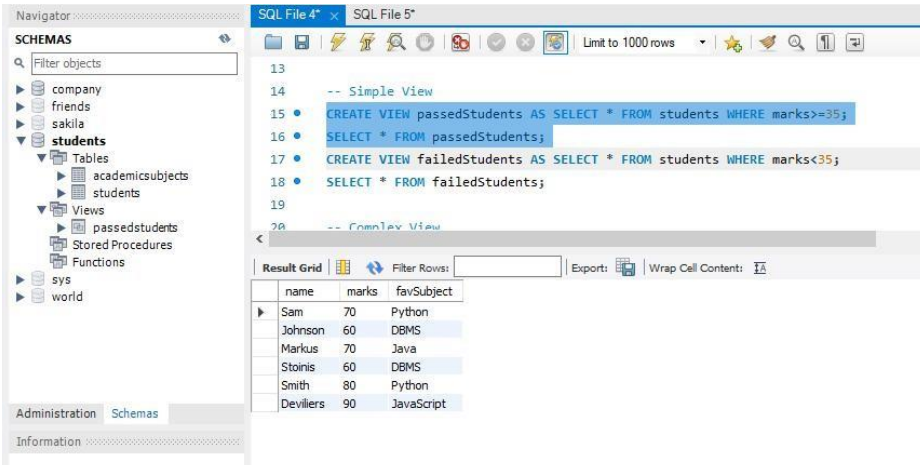
name	marks	favSubject
Sam	70	Python
Johnson	60	DBMS
Markus	70	Java
Stoinis	60	DBMS
Smith	80	Python
Williamson	30	Python
Emily	20	Java
Deviliers	90	JavaScript

academicSubjects table:



sub_name	studentRating	teachingSemester
DBMS	6.5	3
Java	7	3
Javascript	7.9	4
Python	8.5	3
NULL	NULL	NULL

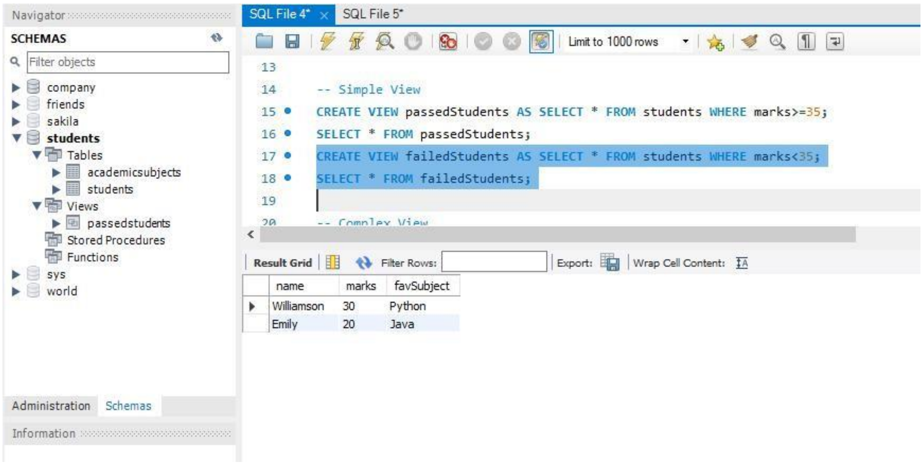
Creating Simple view:



```
-- Simple View
CREATE VIEW passedStudents AS SELECT * FROM students WHERE marks>=35;
SELECT * FROM passedStudents;
CREATE VIEW failedStudents AS SELECT * FROM students WHERE marks<35;
SELECT * FROM failedStudents;

-- Complex View
```

name	marks	favSubject
Sam	70	Python
Johnson	60	DBMS
Markus	70	Java
Stoinis	60	DBMS
Smith	80	Python
Deviliers	90	JavaScript

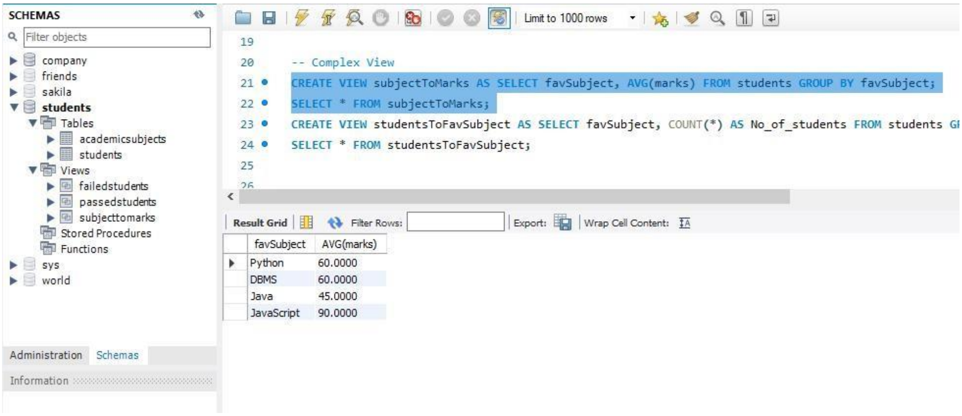


```
-- Simple View
CREATE VIEW passedStudents AS SELECT * FROM students WHERE marks>=35;
SELECT * FROM passedStudents;
CREATE VIEW failedStudents AS SELECT * FROM students WHERE marks<35;
SELECT * FROM failedStudents;

-- Complex View
```

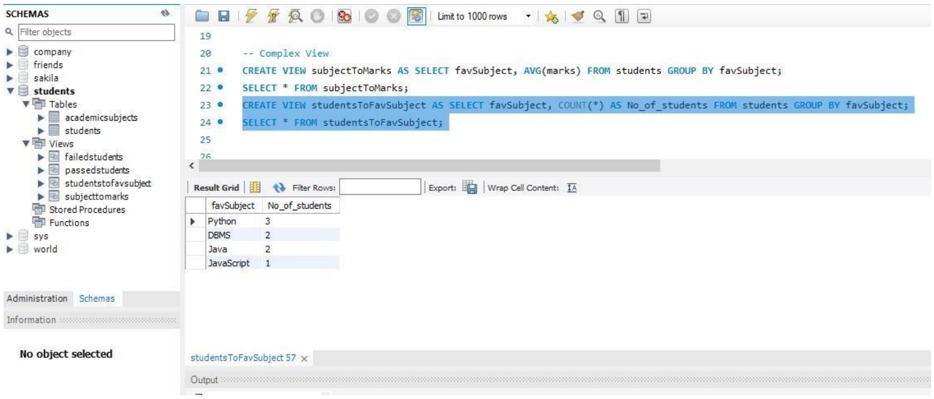
name	marks	favSubject
Williamson	30	Python
Emily	20	Java

Creating complex view:



```
-- Complex View
CREATE VIEW subjectToMarks AS SELECT favSubject, AVG(marks) FROM students GROUP BY favSubject;
SELECT * FROM subjectToMarks;
CREATE VIEW studentsToFavSubject AS SELECT favSubject, COUNT(*) AS No_of_students FROM students GI
SELECT * FROM studentsToFavSubject;
```

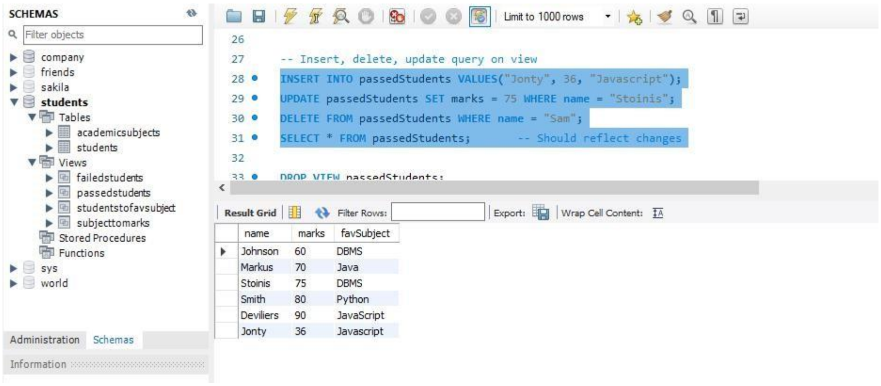
favSubject	AVG(marks)
Python	60.0000
DBMS	60.0000
Java	45.0000
JavaScript	90.0000



```
-- Complex View
CREATE VIEW subjectToMarks AS SELECT favSubject, AVG(marks) FROM students GROUP BY favSubject;
SELECT * FROM subjectToMarks;
CREATE VIEW studentsToFavSubject AS SELECT favSubject, COUNT(*) AS No_of_students FROM students GROUP BY favSubject;
SELECT * FROM studentsToFavSubject;
```

favSubject	No_of_students
Python	3
DBMS	2
Java	2
JavaScript	1

Insert delete and update on view:

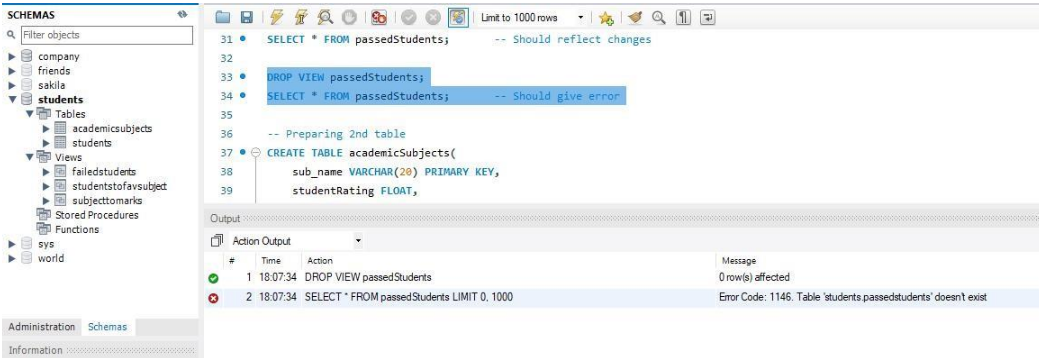


```
-- Insert, delete, update query on view
INSERT INTO passedStudents VALUES("Jonty", 36, "Javascript");
UPDATE passedStudents SET marks = 75 WHERE name = "Stoinis";
DELETE FROM passedStudents WHERE name = "Sam";
SELECT * FROM passedStudents;        -- Should reflect changes

DROP VIEW passedStudents;
```

name	marks	favSubject
Johnson	60	DBMS
Markus	70	Java
Stoinis	75	DBMS
Smith	80	Python
Deviliers	90	JavaScript
Jonty	36	Javascript

Drop view:



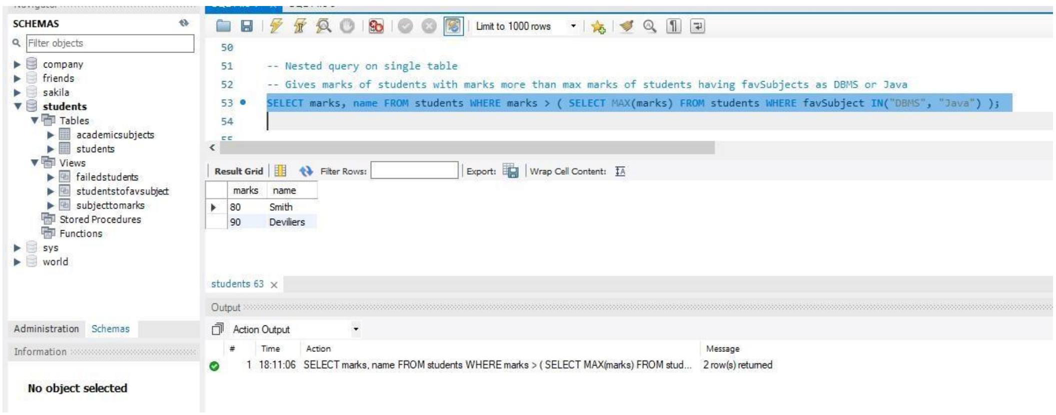
```sql
SELECT * FROM passedStudents;        -- Should reflect changes

DROP VIEW passedStudents;
SELECT * FROM passedStudents;        -- Should give error

-- Preparing 2nd table
CREATE TABLE academicSubjects(
    sub_name VARCHAR(20) PRIMARY KEY,
    studentRating FLOAT,
```

```
#   Time      Action                                         Message
1   18:07:34  DROP VIEW passedStudents                       0 row(s) affected
2   18:07:34  SELECT * FROM passedStudents LIMIT 0, 1000     Error Code: 1146. Table 'students.passedstudents' doesn't exist
```

Nested query on
single table:

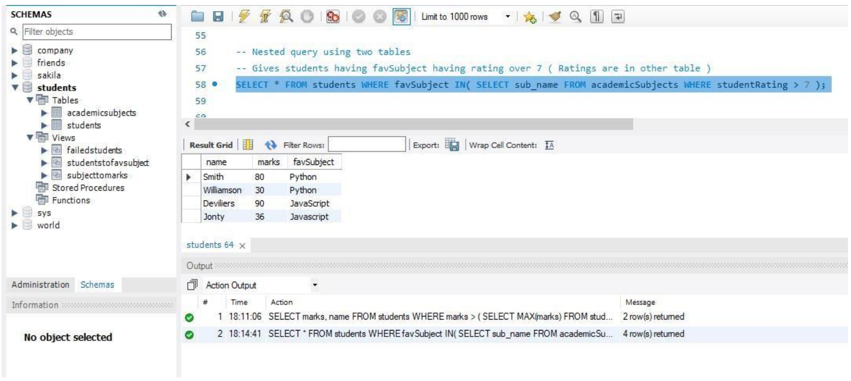


```sql
-- Nested query on single table
-- Gives marks of students with marks more than max marks of students having favSubjects as DBMS or Java
SELECT marks, name FROM students WHERE marks > ( SELECT MAX(marks) FROM students WHERE favSubject IN("DBMS", "Java") );
```

marks	name
80	Smith
90	Deviliers

```
#   Time      Action                                                                   Message
1   18:11:06  SELECT marks, name FROM students WHERE marks > ( SELECT MAX(marks) FROM stud...   2 row(s) returned
```

Nested query using
two tables:



```sql
-- Nested query using two tables
-- Gives students having favSubject having rating over 7 ( Ratings are in other table )
SELECT * FROM students WHERE favSubject IN( SELECT sub_name FROM academicSubjects WHERE studentRating > 7 );
```

name	marks	favSubject
Smith	80	Python
Williamson	30	Python
Deviliers	90	JavaScript
Jonty	36	Javascript

```
#   Time      Action                                                                   Message
1   18:11:06  SELECT marks, name FROM students WHERE marks > ( SELECT MAX(marks) FROM stud...   2 row(s) returned
2   18:14:41  SELECT * FROM students WHERE favSubject IN( SELECT sub_name FROM academicSu...    4 row(s) returned
```

Conclusion: Thus, we created, modified(Inserted updated deleted) and deleted views