Professor's Support System

VIEWS

```
CREATE OR REPLACE VIEW TC AS SELECT Batch, SUM(Classes) AS
TotalClasses FROM WEEK_INFO GROUP BY BATCH;


CREATE OR REPLACE VIEW Theory_Marks AS SELECT StudentId,
(Quizes + InSem1 + InSem2 + EndSem) AS TheoryMarks FROM
STUDENT NATURAL JOIN CLASS_PERFORMANCE WHERE Batch = 2016;


CREATE OR REPLACE VIEW Project_Marks AS SELECT StudentId,
Marks AS ProjectMarks FROM TEAM_MEMBERS AS t NATURAL JOIN
PROJECT_PERFORMANCE AS p WHERE t.TeamNo = p.TeamNo;


CREATE OR REPLACE VIEW Lab_Marks AS SELECT
l.StudentId,(SUM(Submission) + SUM(Viva)) AS LabMarks FROM
STUDENT NATURAL JOIN LAB_PERFORMANCE AS l WHERE Batch = 2016
GROUP BY l.StudentId;


CREATE OR REPLACE VIEW Practical_Marks AS SELECT l.StudentId,
(LabMarks + ProjectMarks) AS PracticalMarks FROM Project_Marks
NATURAL JOIN Lab_Marks AS l;


CREATE OR REPLACE VIEW Marks_Details AS SELECT t.StudentId,
TheoryMarks, PracticalMarks FROM Theory_Marks AS t NATURAL
JOIN Practical_Marks;


CREATE OR REPLACE VIEW PER_Marks_Details AS SELECT StudentID,
(TheoryMarks/(SELECT MAX(TheoryMarks) FROM Marks_Details)) AS
PerTheory, (PracticalMarks/(SELECT MAX(PracticalMarks) FROM
Marks_Details)) AS PerPracticals FROM Marks_Details;


CREATE OR REPLACE VIEW Per_Class_Att AS SELECT c.StudentId,
((SUM(Presents)*1.0/(SELECT TotalClasses FROM TC WHERE Batch =
2018))*100) AS PerClassAttendance FROM CLASS_ATTENDANCE AS c
NATURAL JOIN STUDENT WHERE Batch = 2016 GROUP BY c.StudentID;


CREATE OR REPLACE VIEW Per_Lab_Att AS SELECT StudentID,
((COUNT(CASE WHEN LabAttendance THEN 1 END)*1.0/COUNT(DISTINCT
```

```
Lab))*100) AS PerLabAttendance FROM LAB_PERFORMANCE NATURAL
JOIN STUDENT WHERE Batch = 2016 GROUP BY StudentID;
```

SQL QUERIES

1. **Number of students in each Batch**

   ```
   SELECT Batch, COUNT(StudentID) AS Strength FROM STUDENT
   GROUP BY Batch;
   ```

2. **Marks of each student(Batch - 2016)**

   ```
   SELECT Batch, StudentId, (Quizes + InSem1 + InSem2 +
   EndSem) AS TheoryMarks FROM STUDENT NATURAL JOIN
   CLASS_PERFORMANCE WHERE Batch = 2016;
   ```

3. **Lab Marks of each student(Batch - 2016)**

   ```
   SELECT l.StudentId,(SUM(Submission) + SUM(Viva))  AS
   LabMarks FROM STUDENT NATURAL JOIN LAB_PERFORMANCE AS l
   WHERE Batch = 2016 GROUP BY l.StudentId;
   ```

4. **Project Marks of each team(Batch-wise)**

   ```
   SELECT TeamNo, Marks AS ProjectMarks FROM
   PROJECT_PERFORMANCE;
   ```

5. **Project Marks of each team(Batch - 2016)**

   ```
   SELECT TeamNo, Marks AS ProjectMarks FROM
   PROJECT_PERFORMANCE WHERE TeamNo LIKE '2016%';
   ```

6. **Percentage of class attendance of each student(Batch - 2016)**

   ```
   SELECT c.StudentId, ((SUM(Presents)*1.0/(SELECT
   TotalClasses FROM TC WHERE Batch = 2018))*100) AS
   PerClassAttendance FROM CLASS_ATTENDANCE AS c NATURAL
   JOIN STUDENT WHERE Batch = 2018 GROUP BY c.StudentID;
   ```

7. **Percentage of lab attendance of each student(Batch - 2016)**

   ```
   SELECT StudentID, ((COUNT(CASE WHEN LabAttendance THEN 1
   END)*1.0/COUNT(DISTINCT Lab))*100) AS PerLabAttendance
   FROM LAB_PERFORMANCE NATURAL JOIN STUDENT WHERE Batch =
   2018 GROUP BY StudentID;
   ```

8. **Grade of students in Theory(Batch - 2016)**

```
CREATE OR REPLACE FUNCTION get_theory_grade (g_StudentID
Decimal(9,0))
RETURNS VARCHAR (2) AS $$
DECLARE
    PTheory NUMERIC(3,2) ;
    TheoryGrade VARCHAR (2) ;
BEGIN
-- get the theory grade for given student id
     SELECT INTO PTheory Pertheory FROM per_marks_details
WHERE StudentID = g_StudentID ;

   CASE
      WHEN PTheory > 0.90 THEN
          TheoryGrade = 'AA' ;
      WHEN PTheory > 0.80 THEN
          TheoryGrade = 'AB' ;
      WHEN PTheory > 0.70 THEN
          TheoryGrade = 'BB' ;
      WHEN PTheory > 0.60 THEN
          TheoryGrade = 'BC' ;
      WHEN PTheory > 0.50 THEN
          TheoryGrade = 'CC' ;
      WHEN PTheory > 0.40 THEN
          TheoryGrade = 'CD' ;
      ELSE
          TheoryGrade = 'FF' ;
   END CASE ;

   RETURN      TheoryGrade ;
END ; $$
LANGUAGE plpgsql;

SELECT get_theory_grade(201651005) as TheoryGrade;
```

9. Grade of students in Practical (Batch - 2016)

```
CREATE OR REPLACE FUNCTION get_practical_grade
(g_StudentID Decimal(9,0))
RETURNS VARCHAR (2) AS $$
DECLARE
    PPractical NUMERIC(3,2) ;
    PracticalGrade VARCHAR (2) ;
BEGIN
-- get the practical grade for given student id
     SELECT INTO PPractical PerPracticals FROM
per_marks_details WHERE StudentID = g_StudentID ;

   CASE
      WHEN PPractical > 0.90 THEN
          PracticalGrade = 'AA' ;
```

```
        WHEN PPractical > 0.80 THEN
            PracticalGrade = 'AB' ;
        WHEN PPractical > 0.70 THEN
            PracticalGrade = 'BB' ;
        WHEN PPractical > 0.60 THEN
            PracticalGrade = 'BC' ;
        WHEN PPractical > 0.50 THEN
            PracticalGrade = 'CC' ;
        WHEN PPractical > 0.40 THEN
            PracticalGrade = 'CD' ;
        ELSE
            PracticalGrade = 'FF' ;
    END CASE ;

    RETURN      PracticalGrade ;
END ; $$
LANGUAGE plpgsql;

SELECT get_practical_grade(201651005) as PracticalGrade;
```

10. Students in Batch – 2016

```
SELECT StudentId,Name FROM STUDENT WHERE Batch = 2016;
```

11. Course and Slide Links for all batches for a week (WeekId - 1)

```
SELECT Batch, Course, SlideLink FROM BATCH_DETAILS
NATURAL JOIN WEEK_INFO WHERE WeekId = 1;
```

12. Complete Team details of a batch(Batch - 2016)

```
SELECT TeamNo, StudentId, Topic, LeaderId, OnTimeSub,
Marks FROM PROJECT_TEAMS AS p NATURAL JOIN TEAM_MEMBERS
NATURAL JOIN PROJECT_PERFORMANCE NATURAL JOIN STUDENT AS
s WHERE BATCH = 2016;
```

13. Students in low Class attendance zone(Batch-2016)

```
SELECT StudentID,Name FROM per_class_att NATURAL
JOIN Student WHERE (CPI >= 8.0 AND perclassattendance <
70.00) OR (CPI < 8.0 AND perclassattendance < 75.00);
```

14. Students in low Lab attendance zone(Batch-2016)

```
SELECT StudentID,Name FROM per_lab_att NATURAL JOIN
Student WHERE (CPI >= 8.0 AND perlabattendance < 70.00)
OR (CPI < 8.0 AND perlabattendance < 75.00);
```

15. Average Theory Grade (Batch - 2016)

```
CREATE OR REPLACE FUNCTION get_avg_theory_grade ( )
RETURNS VARCHAR (2) AS $$
DECLARE
    AvgTheory NUMERIC(3,2) ;
    AvgTheoryGrade VARCHAR (2) ;
BEGIN
-- get the avg theory grade for given batch
     SELECT INTO AvgTheory (SUM(Pertheory)*1.0/COUNT(*))
FROM per_marks_details;

   CASE
      WHEN AvgTheory > 0.90 THEN
          AvgTheoryGrade = 'AA' ;
      WHEN AvgTheory > 0.80 THEN
          AvgTheoryGrade = 'AB' ;
      WHEN AvgTheory > 0.70 THEN
          AvgTheoryGrade = 'BB' ;
      WHEN AvgTheory > 0.60 THEN
          AvgTheoryGrade = 'BC' ;
      WHEN AvgTheory > 0.50 THEN
          AvgTheoryGrade = 'CC' ;
      WHEN AvgTheory > 0.40 THEN
          AvgTheoryGrade = 'CD' ;
      ELSE
          AvgTheoryGrade = 'FF' ;
   END CASE ;

   RETURN AvgTheoryGrade ;
END ; $$
LANGUAGE plpgsql;

SELECT get_avg_theory_grade( ) as AvgTheoryGrade;
```

16. Average Practical Grade (Batch - 2016)

```
CREATE OR REPLACE FUNCTION get_avg_practical_grade ( )
RETURNS VARCHAR (2) AS $$
DECLARE
    AvgPractical NUMERIC(3,2) ;
    AvgPracticalGrade VARCHAR (2) ;
BEGIN
-- get the avg theory grade for given batch
     SELECT INTO AvgPractical
(SUM(Perpracticals)*1.0/COUNT(*)) FROM per_marks_details;

   CASE
      WHEN AvgPractical > 0.90 THEN
```

```
            AvgPracticalGrade = 'AA' ;
        WHEN AvgPractical > 0.80 THEN
            AvgPracticalGrade = 'AB' ;
        WHEN AvgPractical > 0.70 THEN
            AvgPracticalGrade = 'BB' ;
        WHEN AvgPractical > 0.60 THEN
            AvgPracticalGrade = 'BC' ;
        WHEN AvgPractical > 0.50 THEN
            AvgPracticalGrade = 'CC' ;
        WHEN AvgPractical > 0.40 THEN
            AvgPracticalGrade = 'CD' ;
        ELSE
            AvgPracticalGrade = 'FF' ;
    END CASE ;

    RETURN AvgPracticalGrade ;
END ; $$
LANGUAGE plpgsql;

SELECT get_avg_practical_grade( ) as AvgPracticalGrade;
```

17. Time schedule of the Professor

```
SELECT * FROM Time_Schedule;
```

18. All details of students(Batch - 2016)

```
SELECT StudentId, Name, TeamNo, CPI, TheoryMarks,
PracticalMarks, PerClassAttendance, PerLabAttendance FROM
STUDENT NATURAL JOIN MARKS_DETAILS NATURAL JOIN
Per_Class_Att NATURAL JOIN Per_Lab_Att;
```