Professor's Support System

SQL QUERIES

1. Number of students in each Batch

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

```
201651058=> SELECT Batch, COUNT(StudentID) AS Strength FROM STUDENT GROUP BY Batch;
 batch | strength
-------+----------
  2017 |       15
  2016 |       15
  2018 |       15
(3 rows)
```

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

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

```
201651058=> SELECT Batch, StudentId, (Quizes + InSem1 + InSem2 + EndSem) AS TheoryMarks
201651058-> FROM STUDENT NATURAL JOIN CLASS_PERFORMANCE
201651058-> WHERE Batch = 2016;
 batch | studentid | theorymarks
-------+-----------+-------------
  2016 | 201651001 |        95.0
  2016 | 201651002 |        80.5
  2016 | 201651003 |        77.5
  2016 | 201651004 |        60.0
  2016 | 201651005 |        70.0
  2016 | 201651006 |        99.0
  2016 | 201651007 |        73.0
  2016 | 201651008 |        57.0
  2016 | 201651009 |        97.0
  2016 | 201651010 |        77.0
  2016 | 201651011 |        62.0
  2016 | 201651012 |        89.0
  2016 | 201651013 |        38.0
  2016 | 201651014 |        53.0
  2016 | 201651015 |        86.0
(15 rows)
```

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

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

```
201651058=> SELECT l.StudentId,(SUM(Submission) + SUM(Viva)) AS LabMarks
201651058-> FROM STUDENT NATURAL JOIN LAB_PERFORMANCE AS l
201651058-> WHERE Batch = 2016
201651058-> GROUP BY l.StudentId;
 studentid | labmarks
-----------+----------
 201651011 |     66.0
 201651004 |     64.0
 201651009 |     79.0
 201651007 |     79.0
 201651002 |     69.0
 201651003 |     65.0
 201651005 |     65.0
 201651001 |     74.0
 201651008 |     68.0
 201651010 |     65.0
 201651013 |     66.0
 201651006 |     64.0
 201651014 |     74.0
 201651015 |     81.0
 201651012 |     66.0
(15 rows)
```

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

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

```
201651058=> SELECT TeamNo, Marks AS ProjectMarks
201651058-> FROM PROJECT_PERFORMANCE;
 teamno | projectmarks
--------+--------------
 2016_1 |         18.0
 2016_2 |         12.0
 2016_3 |         17.0
 2016_4 |         15.0
 2016_5 |         20.0
 2017_1 |         11.0
 2017_2 |         20.0
 2017_3 |         18.0
 2017_4 |         19.0
 2017_5 |          9.0
 2018_1 |          8.0
 2018_2 |         17.0
 2018_3 |          5.0
 2018_4 |         17.0
 2018_5 |         19.0
(15 rows)
```

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

```
1.  SELECT TeamNo, Marks AS ProjectMarks
2.  FROM PROJECT_PERFORMANCE
3.  WHERE TeamNo LIKE '2016%';
```

```
201651058=> SELECT TeamNo, Marks AS ProjectMarks
201651058-> FROM PROJECT_PERFORMANCE
201651058-> WHERE TeamNo LIKE '2016%';
 teamno | projectmarks
--------+--------------
 2016_1 |         18.0
 2016_2 |         12.0
 2016_3 |         17.0
 2016_4 |         15.0
 2016_5 |         20.0
(5 rows)
```

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

```
1.  SELECT c.StudentId, ((SUM(Presents)*1.0/(SELECT TotalClasses FROM TC WHERE Batch =
    2018))*100) AS PerClassAttendance
2.  FROM CLASS_ATTENDANCE AS c NATURAL JOIN STUDENT
3.  WHERE Batch = 2018
4.  GROUP BY c.StudentID;
5.
```

```
201651058=> SELECT c.StudentId, ((SUM(Presents)*1.0/(SELECT TotalClasses FROM TC WHERE Batch = 2018))*100) AS PerClassAttendance
201651058-> FROM CLASS_ATTENDANCE AS c NATURAL JOIN STUDENT
201651058-> WHERE Batch = 2018
201651058-> GROUP BY c.StudentID;
 studentid |    perclassattendance
-----------+--------------------------
 201851002 |  72.72727272727272727300
 201851010 |  63.63636363636363636400
 201851003 |     109.0909090909090900
 201851007 |  72.72727272727272727300
 201851008 |  81.81818181818181818200
 201851015 |  81.81818181818181818200
 201851001 | 100.00000000000000000000
 201851006 |  81.81818181818181818200
 201851011 |  63.63636363636363636400
 201851014 |  72.72727272727272727300
 201851004 |  90.90909090909090909100
 201851009 |  90.90909090909090909100
 201851013 | 100.00000000000000000000
 201851005 |  72.72727272727272727300
 201851012 |  90.90909090909090909100
(15 rows)
```

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

```sql
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;
```

```
201651058=>    SELECT StudentID, ((COUNT(CASE WHEN LabAttendance THEN 1 END)*1.0/COUNT(DISTINCT Lab))*100) AS PerLabAttendance
201651058-> FROM LAB_PERFORMANCE NATURAL JOIN STUDENT
201651058-> WHERE Batch = 2018
201651058-> GROUP BY StudentID;
 studentid |     perlabattendance
-----------+------------------------
 201851001 |  80.0000000000000000000
 201851002 |  60.0000000000000000000
 201851003 |  60.0000000000000000000
 201851004 |  70.0000000000000000000
 201851005 |  70.0000000000000000000
 201851006 |  70.0000000000000000000
 201851007 |  90.0000000000000000000
 201851008 |  70.0000000000000000000
 201851009 |  90.0000000000000000000
 201851010 |  60.0000000000000000000
 201851011 |  50.0000000000000000000
 201851012 |  50.0000000000000000000
 201851013 |  50.0000000000000000000
 201851014 |  60.0000000000000000000
 201851015 | 100.0000000000000000000
(15 rows)
```

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

```sql
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;
```

```
201651058=>       CREATE OR REPLACE FUNCTION get_theory_grade (g_StudentID Decimal(9,0))
201651058-> RETURNS VARCHAR (2) AS $$
201651058$> DECLARE
201651058$>     PTheory NUMERIC(3,2) ;
201651058$>     TheoryGrade VARCHAR (2) ;
201651058$> BEGIN
201651058$> -- get the theory grade for given student id
201651058$>     SELECT INTO PTheory Pertheory FROM per_marks_details WHERE StudentID = g_StudentID ;
201651058$>
201651058$>    CASE
201651058$>       WHEN PTheory > 0.90 THEN
201651058$>         TheoryGrade = 'AA' ;
201651058$>       WHEN PTheory > 0.80 THEN
201651058$>         TheoryGrade = 'AB' ;
201651058$>     WHEN PTheory > 0.70 THEN
201651058$>         TheoryGrade = 'BB' ;
201651058$>     WHEN PTheory > 0.60 THEN
201651058$>         TheoryGrade = 'BC' ;
201651058$>     WHEN PTheory > 0.50 THEN
201651058$>         TheoryGrade = 'CC' ;
201651058$>     WHEN PTheory > 0.40 THEN
201651058$>         TheoryGrade = 'CD' ;
201651058$>       ELSE
201651058$>         TheoryGrade = 'FF' ;
201651058$>    END CASE ;
201651058$>
201651058$>    RETURN   TheoryGrade ;
201651058$> END ; $$
201651058-> LANGUAGE plpgsql;
CREATE FUNCTION
201651058=>
201651058=> SELECT get_theory_grade(201651005) as TheoryGrade;
 theorygrade
-------------
 BB
(1 row)
```

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

```
1.  CREATE OR REPLACE FUNCTION get_practical_grade (g_StudentID Decimal(9,0))
2.  RETURNS VARCHAR (2) AS $$
3.  DECLARE
4.      PPractical NUMERIC(3,2) ;
5.      PracticalGrade VARCHAR (2) ;
6.  BEGIN
7.  -- get the practical grade for given student id
8.      SELECT INTO PPractical PerPracticals FROM per_marks_details WHERE StudentID = g
    _StudentID ;
9.
10.    CASE
11.       WHEN PPractical > 0.90 THEN
12.         PracticalGrade = 'AA' ;
13.       WHEN PPractical > 0.80 THEN
14.         PracticalGrade = 'AB' ;
15.     WHEN PPractical > 0.70 THEN
16.         PracticalGrade = 'BB' ;
17.     WHEN PPractical > 0.60 THEN
18.         PracticalGrade = 'BC' ;
19.     WHEN PPractical > 0.50 THEN
20.         PracticalGrade = 'CC' ;
21.     WHEN PPractical > 0.40 THEN
22.         PracticalGrade = 'CD' ;
23.       ELSE
24.         PracticalGrade = 'FF' ;
25.    END CASE ;
26.
27.    RETURN   PracticalGrade ;
```

```
28. END ; $$
29. LANGUAGE plpgsql;
30.
31. SELECT get_practical_grade(201651005) as PracticalGrade;
```

```
201651058=> CREATE OR REPLACE FUNCTION get_practical_grade (g_StudentID Decimal(9,0))
201651058-> RETURNS VARCHAR (2) AS $$
201651058$> DECLARE
201651058$>     PPractical NUMERIC(3,2) ;
201651058$>     PracticalGrade VARCHAR (2) ;
201651058$> BEGIN
201651058$> -- get the practical grade for given student id
201651058$>     SELECT INTO PPractical PerPracticals FROM per_marks_details WHERE StudentID = g_StudentID ;
201651058$>
201651058$>    CASE
201651058$>       WHEN PPractical > 0.90 THEN
201651058$>         PracticalGrade = 'AA' ;
201651058$>       WHEN PPractical > 0.80 THEN
201651058$>         PracticalGrade = 'AB' ;
201651058$>     WHEN PPractical > 0.70 THEN
201651058$>         PracticalGrade = 'BB' ;
201651058$>     WHEN PPractical > 0.60 THEN
201651058$>         PracticalGrade = 'BC' ;
201651058$>     WHEN PPractical > 0.50 THEN
201651058$>         PracticalGrade = 'CC' ;
201651058$>     WHEN PPractical > 0.40 THEN
201651058$>         PracticalGrade = 'CD' ;
201651058$>       ELSE
201651058$>         PracticalGrade = 'FF' ;
201651058$>    END CASE ;
201651058$>
201651058$>    RETURN   PracticalGrade ;
201651058$> END ; $$
201651058-> LANGUAGE plpgsql;
CREATE FUNCTION
201651058=>
201651058=> SELECT get_practical_grade(201651005) as PracticalGrade;
 practicalgrade
----------------
 BB
(1 row)
```

10. Students in Batch – 2016

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

```
201651058=> SELECT StudentId,Name FROM STUDENT WHERE Batch = 2016;
 studentid |       name
-----------+------------------
 201651001 | Aashutosh Rathi
 201651002 | Aastha Nehru
 201651003 | Bhavya Singh
 201651004 | Bhavesh Rajput
 201651005 | Cady Jacob
 201651006 | Charlie Smith
 201651007 | Diwanshu Jain
 201651008 | Divya Maheedharan
 201651009 | Divyesh Sinha
 201651010 | Devank Singhai
 201651011 | Divya Singh
 201651012 | Era Bhowmik
 201651013 | Farah Khan
 201651014 | Farhat Khan
 201651015 | Fatima Ahmed
(15 rows)
```

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

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

```
201651058=> SELECT Batch, Course, SlideLink
201651058-> FROM BATCH_DETAILS NATURAL JOIN WEEK_INFO
201651058-> WHERE WeekId = 1;
 batch |        course        |                                 slidelink
-------+----------------------+---------------------------------------------------------------------------
  2016 | DBMS                 | https://www.google.co.in/?gfe_rd=cr&dcr=0&ei=ZG_PWoKlHovK8AfJuKLICA&gws_rd=ssls
  2017 | OOP                  | https://www.google.co.in/?gfe_rd=cr&dcr=0&ei=ZG_PWoKlHovK8AfJuKLICA&gws_rd=ssls
  2018 | Software Engineering | https://www.google.co.in/?gfe_rd=cr&dcr=0&ei=ZG_PWoKlHovK8AfJuKLICA&gws_rd=ssls
(3 rows)
```

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

```sql
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;
```

```
201651058=> SELECT TeamNo, StudentId, Topic, LeaderId, OnTimeSub, Marks
201651058-> FROM PROJECT_TEAMS AS p NATURAL JOIN TEAM_MEMBERS NATURAL JOIN PROJECT_PERFORMANCE NATURAL JOIN STUDENT AS s
201651058-> WHERE BATCH = 2016;
 teamno | studentid |           topic            | leaderid  | ontimesub | marks
--------+-----------+----------------------------+-----------+-----------+-------
 2016_1 | 201651003 | Library Management System  | 201651001 |         4 |  18.0
 2016_1 | 201651002 | Library Management System  | 201651001 |         4 |  18.0
 2016_1 | 201651001 | Library Management System  | 201651001 |         4 |  18.0
 2016_2 | 201651006 | Hospital Management System | 201651004 |         2 |  12.0
 2016_2 | 201651005 | Hospital Management System | 201651004 |         2 |  12.0
 2016_2 | 201651004 | Hospital Management System | 201651004 |         2 |  12.0
 2016_3 | 201651009 | Hotel Management System    | 201651007 |         4 |  17.0
 2016_3 | 201651008 | Hotel Management System    | 201651007 |         4 |  17.0
 2016_3 | 201651007 | Hotel Management System    | 201651007 |         4 |  17.0
 2016_4 | 201651012 | Railway Enquiry System     | 201651010 |         3 |  15.0
 2016_4 | 201651011 | Railway Enquiry System     | 201651010 |         3 |  15.0
 2016_4 | 201651010 | Railway Enquiry System     | 201651010 |         3 |  15.0
 2016_5 | 201651015 | Holiday Trip planner       | 201651013 |         5 |  20.0
 2016_5 | 201651014 | Holiday Trip planner       | 201651013 |         5 |  20.0
 2016_5 | 201651013 | Holiday Trip planner       | 201651013 |         5 |  20.0
(15 rows)
```

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

```sql
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);
```

```
201651058=> SELECT StudentID,Name FROM per_class_att NATURAL JOIN Student
201651058-> WHERE (CPI >= 8.0 AND perclassattendance < 70.00) OR (CPI < 8.0 AND perclassattendance < 75.00);
 studentid |       name
-----------+------------------
 201651011 | Divya Singh
 201651004 | Bhavesh Rajput
 201651009 | Divyesh Sinha
 201651008 | Divya Maheedharan
 201651010 | Devank Singhai
 201651006 | Charlie Smith
 201651014 | Farhat Khan
 201651015 | Fatima Ahmed
 201651012 | Era Bhowmik
(9 rows)
```

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

```sql
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);
```

```
201651058=> SELECT StudentID,Name
201651058-> FROM per_lab_att NATURAL JOIN Student
201651058-> WHERE (CPI >= 8.0 AND perlabattendance < 70.00) OR (CPI < 8.0 AND perlabattendance < 75.00);
 studentid |      name
-----------+---------------
 201651003 | Bhavya Singh
 201651004 | Bhavesh Rajput
 201651005 | Cady Jacob
 201651006 | Charlie Smith
 201651010 | Devank Singhai
 201651011 | Divya Singh
 201651012 | Era Bhowmik
 201651013 | Farah Khan
 201651014 | Farhat Khan
(9 rows)
```

15. Average Theory Grade (Batch - 2016)

```sql
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;
```

```
201651058=> CREATE OR REPLACE FUNCTION get_avg_theory_grade ( )
201651058-> RETURNS VARCHAR (2) AS $$
201651058$> DECLARE
201651058$>     AvgTheory NUMERIC(3,2) ;
201651058$>     AvgTheoryGrade VARCHAR (2) ;
201651058$> BEGIN
201651058$> -- get the avg theory grade for given batch
201651058$>     SELECT INTO AvgTheory (SUM(Pertheory)*1.0/COUNT(*)) FROM per_marks_details;
201651058$>
201651058$>    CASE
201651058$>       WHEN AvgTheory > 0.90 THEN
201651058$>         AvgTheoryGrade = 'AA' ;
201651058$>       WHEN AvgTheory > 0.80 THEN
201651058$>         AvgTheoryGrade = 'AB' ;
201651058$>     WHEN AvgTheory > 0.70 THEN
201651058$>         AvgTheoryGrade = 'BB' ;
201651058$>     WHEN AvgTheory > 0.60 THEN
201651058$>         AvgTheoryGrade = 'BC' ;
201651058$>     WHEN AvgTheory > 0.50 THEN
201651058$>         AvgTheoryGrade = 'CC' ;
201651058$>     WHEN AvgTheory > 0.40 THEN
201651058$>         AvgTheoryGrade = 'CD' ;
201651058$>       ELSE
201651058$>         AvgTheoryGrade = 'FF' ;
201651058$>    END CASE ;
201651058$>
201651058$>    RETURN AvgTheoryGrade ;
201651058$> END ; $$
201651058-> LANGUAGE plpgsql;
CREATE FUNCTION
201651058=>
201651058=> SELECT get_avg_theory_grade( ) as AvgTheoryGrade;
 avgtheorygrade
----------------
 BB
(1 row)
```

16. Average Practical Grade (Batch - 2016)

```
1.  CREATE OR REPLACE FUNCTION get_avg_practical_grade ( )
2.  RETURNS VARCHAR (2) AS $$
3.  DECLARE
4.      AvgPractical NUMERIC(3,2) ;
5.      AvgPracticalGrade VARCHAR (2) ;
6.  BEGIN
7.  -- get the avg theory grade for given batch
8.      SELECT INTO AvgPractical (SUM(Perpracticals)*1.0/COUNT(*)) FROM per_marks_detai
    ls;
9.
10.    CASE
11.       WHEN AvgPractical > 0.90 THEN
12.         AvgPracticalGrade = 'AA' ;
13.       WHEN AvgPractical > 0.80 THEN
14.         AvgPracticalGrade = 'AB' ;
15.     WHEN AvgPractical > 0.70 THEN
16.         AvgPracticalGrade = 'BB' ;
17.     WHEN AvgPractical > 0.60 THEN
18.         AvgPracticalGrade = 'BC' ;
19.     WHEN AvgPractical > 0.50 THEN
20.         AvgPracticalGrade = 'CC' ;
21.     WHEN AvgPractical > 0.40 THEN
22.         AvgPracticalGrade = 'CD' ;
23.       ELSE
24.         AvgPracticalGrade = 'FF' ;
25.    END CASE ;
26.
```

```
27.    RETURN AvgPracticalGrade ;
28. END ; $$
29. LANGUAGE plpgsql;
30.
31. SELECT get_avg_practical_grade( ) as AvgPracticalGrade;
```

```
201651058=> CREATE OR REPLACE FUNCTION get_avg_practical_grade ( )
201651058-> RETURNS VARCHAR (2) AS $$
201651058$> DECLARE
201651058$>     AvgPractical NUMERIC(3,2) ;
201651058$>     AvgPracticalGrade VARCHAR (2) ;
201651058$> BEGIN
201651058$> -- get the avg theory grade for given batch
201651058$>     SELECT INTO AvgPractical (SUM(Perpracticals)*1.0/COUNT(*)) FROM per_marks_details;
201651058$>
201651058$>    CASE
201651058$>       WHEN AvgPractical > 0.90 THEN
201651058$>         AvgPracticalGrade = 'AA' ;
201651058$>       WHEN AvgPractical > 0.80 THEN
201651058$>         AvgPracticalGrade = 'AB' ;
201651058$>     WHEN AvgPractical > 0.70 THEN
201651058$>         AvgPracticalGrade = 'BB' ;
201651058$>     WHEN AvgPractical > 0.60 THEN
201651058$>         AvgPracticalGrade = 'BC' ;
201651058$>     WHEN AvgPractical > 0.50 THEN
201651058$>         AvgPracticalGrade = 'CC' ;
201651058$>     WHEN AvgPractical > 0.40 THEN
201651058$>         AvgPracticalGrade = 'CD' ;
201651058$>       ELSE
201651058$>         AvgPracticalGrade = 'FF' ;
201651058$>    END CASE ;
201651058$>
201651058$>    RETURN AvgPracticalGrade ;
201651058$> END ; $$
201651058-> LANGUAGE plpgsql;
CREATE FUNCTION
201651058=>
201651058=> SELECT get_avg_practical_grade( ) as AvgPracticalGrade;
 avgpracticalgrade
-------------------
 AB
(1 row)
```

17. Time schedule of the Professor

```
1.         SELECT * FROM Time_Schedule;
```

```
201651058=> SELECT * FROM Time_Schedule;
 batch |    day    |   time
-------+-----------+----------
  2016 | Monday    | 08:45:00
  2016 | Wednesday | 08:45:00
  2016 | Thursday  | 10:30:00
  2017 | Tuesday   | 08:45:00
  2017 | Thursday  | 08:45:00
  2017 | Friday    | 10:30:00
  2018 | Monday    | 10:30:00
  2018 | Wednesday | 10:30:00
  2018 | Thursday  | 08:45:00
(9 rows)
```

18. All details of students(Batch - 2016)

```
1. SELECT StudentId, Name, TeamNo, CPI, TheoryMarks,PracticalMarks, PerClassAttendance
   , PerLabAttendance
2. FROM STUDENT NATURAL JOIN MARKS_DETAILS NATURAL JOIN Per_Class_Att NATURAL JOIN Per
   _Lab_Att;
```

```
201651058=> SELECT StudentId, Name, TeamNo, CPI, TheoryMarks,PracticalMarks, PerClassAttendance, PerLabAttendance
201651058-> FROM STUDENT NATURAL JOIN MARKS_DETAILS NATURAL JOIN Per_Class_Att NATURAL JOIN Per_Lab_Att;
 studentid |       name        | teamno |  cpi  | theorymarks | practicalmarks |    perclassattendance     |    perlabattendance
-----------+-------------------+--------+-------+-------------+----------------+---------------------------+-------------------------
 201651001 | Aashutosh Rathi   | 2016_1 |  9.32 |        95.0 |           92.0 | 100.0000000000000000000000 | 70.0000000000000000000000
 201651002 | Aastha Nehru      | 2016_1 |  8.76 |        80.5 |           87.0 |     109.0909090909090900 | 70.0000000000000000000000
 201651003 | Bhavya Singh      | 2016_1 | 10.00 |        77.5 |           83.0 |  81.81818181818181818200 | 60.0000000000000000000000
 201651004 | Bhavesh Rajput    | 2016_2 |  6.34 |        60.0 |           76.0 |  72.72727272727272727300 | 70.0000000000000000000000
 201651005 | Cady Jacob        | 2016_2 |  7.35 |        70.0 |           77.0 |  90.90909090909090909100 | 60.0000000000000000000000
 201651006 | Charlie Smith     | 2016_2 |  5.69 |        99.0 |           76.0 |  72.72727272727272727300 | 70.0000000000000000000000
 201651007 | Diwanshu Jain     | 2016_3 |  9.76 |        73.0 |           96.0 |  72.72727272727272727300 | 90.0000000000000000000000
 201651008 | Divya Maheedharan | 2016_3 |  8.80 |        57.0 |           85.0 |  63.63636363636363636400 | 70.0000000000000000000000
 201651009 | Divyesh Sinha     | 2016_3 |  4.48 |        97.0 |           96.0 |  72.72727272727272727300 | 90.0000000000000000000000
 201651010 | Devank Singhai    | 2016_4 |  9.10 |        77.0 |           80.0 |  54.54545454545454545500 | 60.0000000000000000000000
 201651011 | Divya Singh       | 2016_4 |  7.60 |        62.0 |           81.0 |  63.63636363636363636400 | 50.0000000000000000000000
 201651012 | Era Bhowmik       | 2016_4 |  6.80 |        89.0 |           81.0 |  72.72727272727272727300 | 50.0000000000000000000000
 201651013 | Farah Khan        | 2016_5 |  8.50 |        38.0 |           86.0 | 100.0000000000000000000000 | 50.0000000000000000000000
 201651014 | Farhat Khan       | 2016_5 |  7.20 |        53.0 |           94.0 |  72.72727272727272727300 | 70.0000000000000000000000
 201651015 | Fatima Ahmed      | 2016_5 |  6.90 |        86.0 |          101.0 |  72.72727272727272727300 | 90.0000000000000000000000
(15 rows)
```