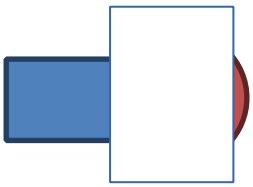


- \i D:/Web/ITI/ITI_Tasks/PostgreSQL/'Lab 3'/Function.sql

1. Create multiply function which take two number and return the multiply of them
   - SELECT multiply(20,30);

```
CREATE OR REPLACE FUNCTION multiply (num1 int, num2 int)
RETURNS int AS $$
DECLARE
  muliple int;
BEGIN
  muliple = num1 * num2;
  RETURN muliple;
END;
$$LANGUAGE plpgsql;
```

2. Create Hello world function which take username and return welcome message to user using his name
   - SELECT hello_World('Sarah');

```
CREATE OR REPLACE FUNCTION hello_world (username text)
RETURNS text AS $$
BEGIN
  RETURN CONCAT('Welcome ',username);
END;
$$LANGUAGE plpgsql;
```

3. Create function which takes number and return if this number is odd or even.
   - SELECT odd_or_even(6);

```sql
CREATE OR REPLACE FUNCTION odd_or_even (num int)
RETURNS text AS $$
DECLARE
  result text;
BEGIN
  if num % 2 = 0 THEN
    result := 'Even';
  else
    result := 'Odd';
  END IF;
  RETURN CONCAT('The Number is: ',result);
END;
```

4. Create AddNewStudent function which take Student firstName and lastname and birthdate and TrackName and add this new student info at database
   - SELECT AddNewStudent('Eman', 'Moaaz', '7-1-2000', 'Power BI');

```sql
CREATE OR REPLACE FUNCTION AddNewStudent (firstname varchar(25), lastname varchar(25), birthdate date, trackName text)
RETURNS text AS $$
DECLARE
  s_id smallint;
  trackid smallint;
BEGIN
  SELECT id INTO s_id FROM student ORDER BY id DESC LIMIT 1;
  s_id = s_id + 1;
  SELECT id INTO trackid FROM track WHERE name = trackName;
  INSERT INTO student (id, trackid, firstname, lastname, birthdate)
  VALUES (s_id, trackid, firstname, lastname, birthdate);
  RETURN 'Column Added Successfully';
END;
$$ LANGUAGE plpgsql;
```

5. Create function which takes StudentId and return the string/text that describe the use info(firstname, last name, age, TrackName).
   - SELECT studentInfo(3);

```
CREATE OR REPLACE FUNCTION studentInfo (s_id int)
RETURNS text AS $$
DECLARE
  studentData student%ROWTYPE;
  trackName text;
BEGIN
  SELECT * INTO studentData FROM student WHERE id = s_id;
  SELECT name INTO trackName FROM track WHERE id = studentData.trackid;
  RETURN CONCAT(studentData.firstname, ' ', studentData.lastname, ' registered in ', trackName, ' track, She is ', age(studentData.birthdate));
END;
$$ LANGUAGE plpgsql;
```

6. Create function which takes TrackName and return the students names in this track.
   - SELECT student_track('Full Stack Web Development Using Python');

```
CREATE OR REPLACE FUNCTION student_track (trackName text)
RETURNS table("FullName" text) AS $$
BEGIN
  RETURN QUERY
  SELECT CONCAT(s.firstname, ' ', s.lastname) AS "Full name"
  FROM student s, track t
  WHERE s.trackid = t.id
  AND t.name = trackName;
END;
$$ LANGUAGE plpgsql;
```

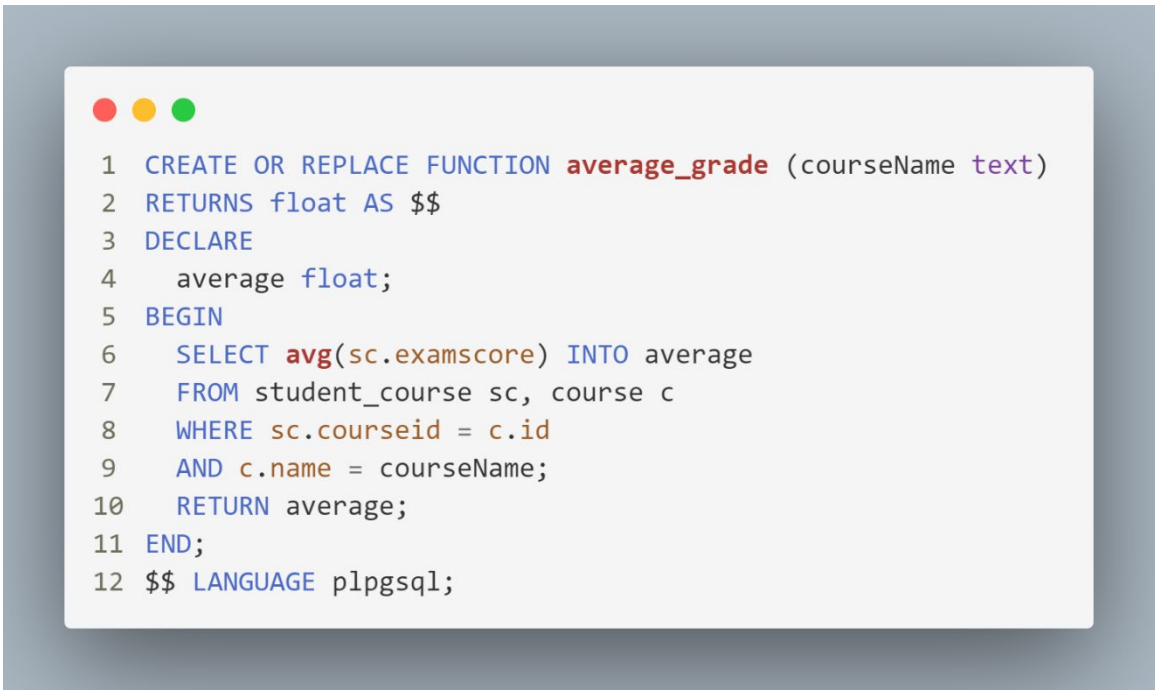
7. Create function which takes student id and subject id and return score the student in subject.
   - SELECT score(1,2);

```
CREATE OR REPLACE FUNCTION score (s_id int, c_id int)
RETURNS text AS $$
DECLARE
  studentName text;
  courseName text;
  studentScore float;
BEGIN
  SELECT CONCAT(firstname, ' ', lastname) AS "FullName" INTO studentName FROM student WHERE id = s_id;
  SELECT name INTO courseName FROM course WHERE id = c_id;
  SELECT examscore INTO studentScore FROM student_course WHERE studentid = s_id AND courseid = c_id;
  RETURN CONCAT(studentName, ' got a score ',studentScore,' in ',courseName);
END;
$$ LANGUAGE plpgsql;
```

8. Create function which takes subject id and return the number of students who failed in a subject (Score less than 50).
   - SELECT failing_student(1);

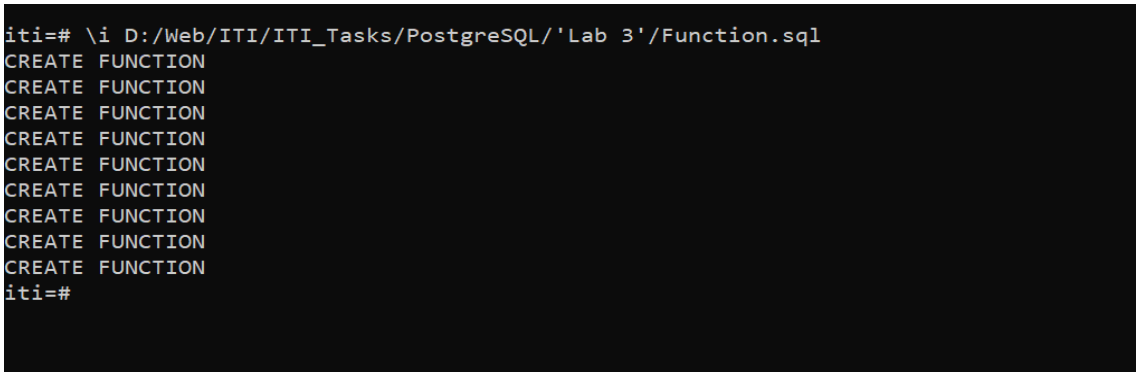
```
CREATE OR REPLACE FUNCTION failing_student(c_id int)
RETURNS int AS $$
DECLARE
  count int;
BEGIN
  SELECT count(examscore) INTO count
  FROM student_course
  WHERE courseid = c_id
  AND examscore < 60;
  RETURN count;
END;
$$ LANGUAGE plpgsql;
```

9. Create function which take subject name and return the average grades for subject
   - SELECT average_grade('HTML & CSS');



```sql
CREATE OR REPLACE FUNCTION average_grade (courseName text)
RETURNS float AS $$
DECLARE
  average float;
BEGIN
  SELECT avg(sc.examscore) INTO average
  FROM student_course sc, course c
  WHERE sc.courseid = c.id
  AND c.name = courseName;
  RETURN average;
END;
$$ LANGUAGE plpgsql;
```

10. Import SQL file into your database.
    - \i D:/Web/ITI/ITI_Tasks/PostgreSQL/'Lab 3'/Function.sql



```
iti=# \i D:/Web/ITI/ITI_Tasks/PostgreSQL/'Lab 3'/Function.sql
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
iti=#
```