DBMS Project

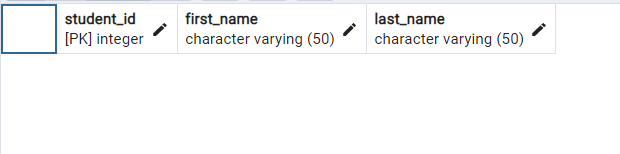
Student Grade Calculator

* Create a table for students :

CREATE TABLE students ( student\_id SERIAL PRIMARY KEY, first\_name VARCHAR(50),

last\_name VARCHAR(50) );

* Output :



* Insert values into the students table :

INSERT INTO students (student\_id, first\_name, last\_name) VALUES

(1, 'Rajesh', 'Kumar'), (2, 'Sita', 'Sharma'), (3, 'Amit', 'Patel'), (4, 'Poojan', 'Thakker'),

(5, 'Anil', 'Gupta'), (6, 'Priya', 'Singh'), (7, 'Vikram', 'Yadav'), (8, 'Mala', 'Rajput'),

(9, 'Nitin', 'Mishra'),(10, 'Deepa', 'Jha'), (11, 'Arun', 'Bansal'), (12, 'Geeta', 'Reddy'),

(13, 'Rahul', 'Choudhary'), (14, 'Kavita', 'Mehta'), (15, 'Vijay', 'Rao'),

(16, 'Anita', 'Pandey'),(17, 'Sanjay', 'Shukla'),(18, 'Neha', 'Chopra'),

(19, 'Rajendra', 'Malik'),(20, 'Suman', 'Dubey'),(21, 'Ashok', 'Agarwal'),

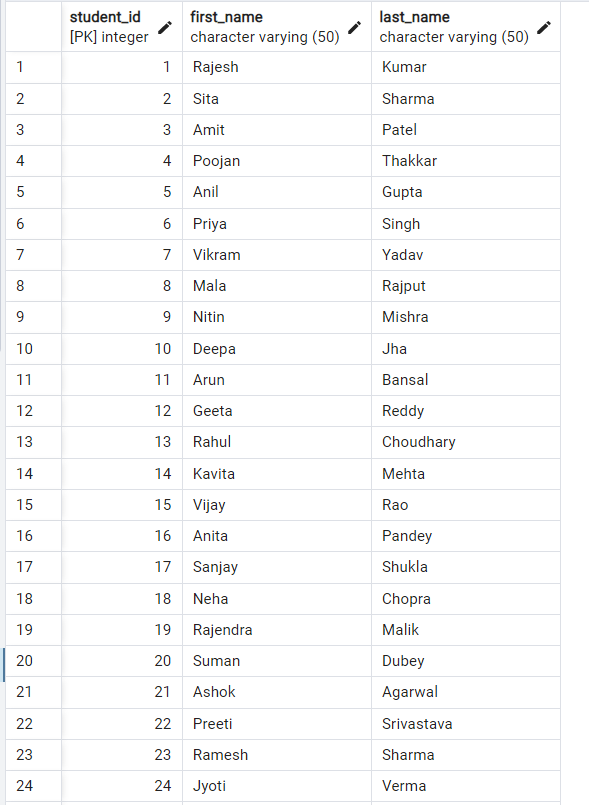
(22, 'Preeti', 'Srivastava'),(23, 'Ramesh', 'Sharma'),(24, 'Jyoti', 'Verma'),

(25, 'Harish', 'Yadav');

* Display all data from students :

select \* from students;

* Output :

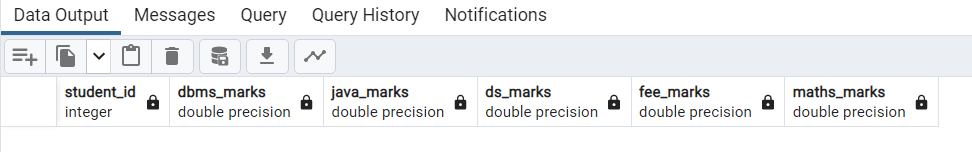


* Create table for student\_marks :

create table Student\_marks( student\_id int ,DBMS\_marks float,JAVA\_marks float,

DS\_marks float, FEE\_marks float, Maths\_marks float );

* Output :



* Insert values into the Student\_marks table :

INSERT INTO Student\_marks (student\_id, DBMS\_marks, JAVA\_marks, DS\_marks, FEE\_marks, Maths\_marks)

VALUES (1, 85.5, 78.0, 92.5, 90.0, 88.5), (2, 75.0, 89.5, 76.5, 82.0, 94.0),

(3, 92.0, 80.5, 85.0, 78.5, 91.0), (4, 88.5, 94.0, 81.5, 89.0, 85.5),

(5, 78.0, 86.5, 90.0, 75.5, 87.0),(6, 90.5, 92.5, 88.0, 93.0, 82.5),

(7, 86.0, 75.0, 89.5, 80.5, 79.0), (8, 83.5, 91.5, 87.0, 88.5, 92.0),

(9, 79.5, 88.0, 90.5, 81.0, 84.5),(10, 92.5, 84.0, 76.5, 87.5, 79.0),

(11, 85.0, 79.5, 91.0, 88.0, 83.5),(12, 89.0, 87.0, 82.5, 90.5, 85.0),

(13, 77.5, 93.5, 84.5, 76.0, 91.5),(14, 92.0, 85.5, 78.0, 89.5, 87.0),

(15, 80.5, 91.0, 87.5, 82.5, 90.0),(16, 88.5, 78.0, 94.0, 89.0, 76.5),

(17, 92.5, 89.5, 83.0, 90.5, 85.0), (18, 76.0, 87.0, 91.5, 79.5, 88.0),

(19, 85.0, 80.5, 93.0, 86.5, 82.0), (20, 91.0, 92.0, 84.5, 89.5, 78.5),

(21, 77.5, 88.0, 85.5, 76.0, 90.5),(22, 90.0, 81.5, 92.5, 88.5, 79.0),

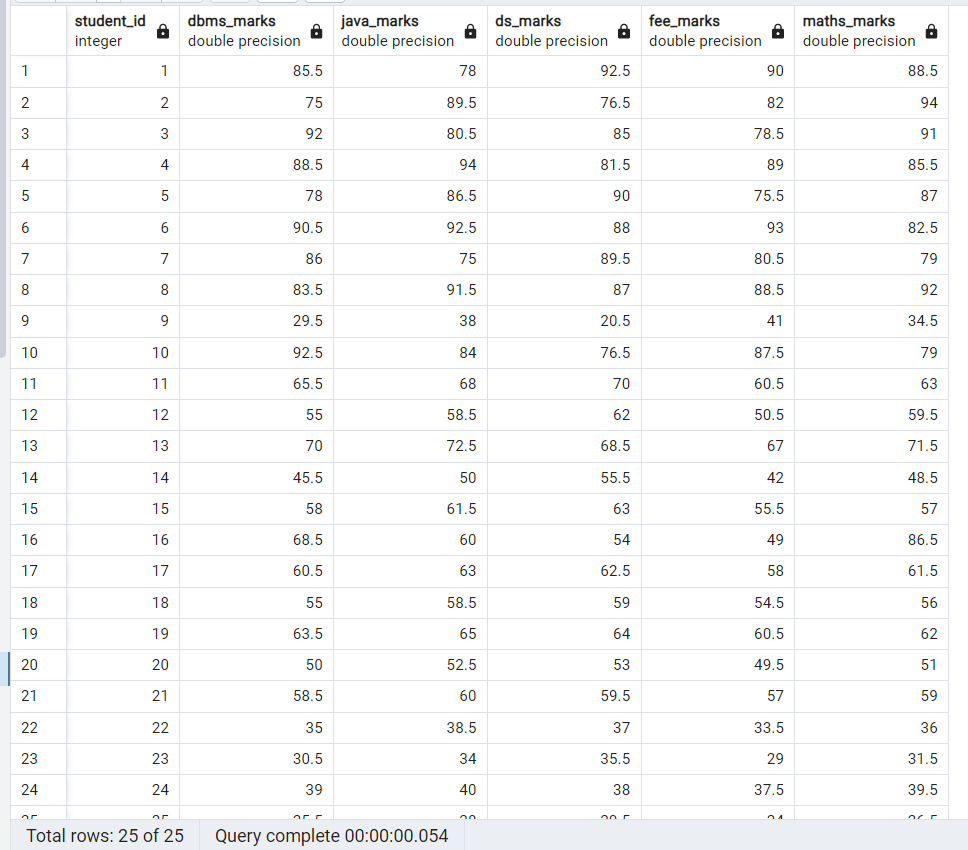
(23, 83.0, 79.5, 91.0, 84.5, 87.5),(24, 92.5, 86.0, 80.0, 89.0, 76.5),

(25, 78.5, 90.5, 85.0, 77.5, 91.0);

* Display all data from Student\_marks :

select \* from Student\_marks;

* Output :



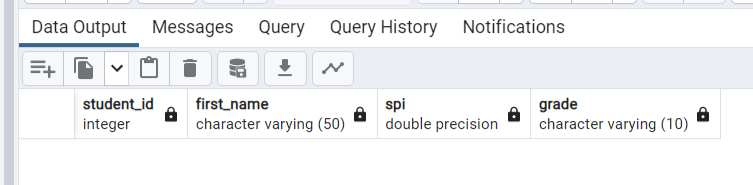
* Create table for result :

create table Result(student\_id int,first\_name varchar(50),spi float,grad varchar(10));

* Display all data from result :

select \* from Result;

* Output :



* Create Procedure name as inser\_result() :

create or replace procedure insert\_result()

as $$

declare

Student\_result CURSOR FOR

SELECT student\_id, first\_name FROM students;

sid students.student\_id%type;

sname students.first\_name%type;

begin

OPEN Student\_result;

LOOP

FETCH Student\_result INTO sid,sname;

EXIT WHEN NOT FOUND;

INSERT INTO Result (student\_id ,first\_name,spi ) values(sid,sname,0.0);

END LOOP;

CLOSE Student\_result;

end;

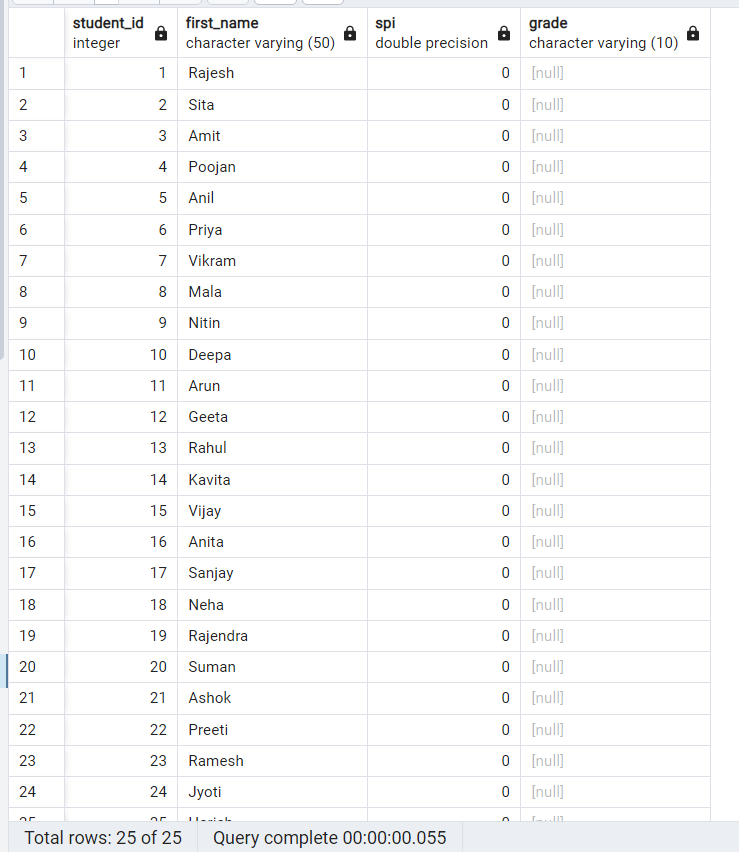
$$

LANGUAGE plpgsql;

* Call the Procedure :

call insert\_result();

* Output :



* Create Procedure name as Calculate\_spi() :

CREATE OR REPLACE PROCEDURE calculate\_spi() AS $$

DECLARE

java\_credit int :=6;

dbms\_credit int :=6;

ds\_credit int :=6;

fee\_credit int :=4;

maths\_credit int :=4;

total\_credits INT := java\_credit+dbms\_credit+ds\_credit+fee\_credit+maths\_credit;

sid student\_marks.student\_id%type;

javamarks student\_marks.java\_marks%type;

dbmsmarks student\_marks.dbms\_marks%type;

dsmarks student\_marks.ds\_marks%type;

feemarks student\_marks.fee\_marks%type;

mathsmarks student\_marks.maths\_marks%type;

Marks\_cursor cursor for

select student\_id, dbms\_marks,java\_marks,ds\_marks,fee\_marks,maths\_marks from student\_marks;

total\_grade\_points FLOAT := 0;

student\_spi FLOAT;

subject\_rec RECORD;

BEGIN

-- Open the cursor

OPEN Marks\_cursor;

-- Loop through subjects

LOOP

FETCH Marks\_cursor INTO sid,javamarks,dbmsmarks,dsmarks,feemarks,mathsmarks ;

EXIT WHEN NOT FOUND;

student\_spi := (javamarks\*java\_credit + dbmsmarks\*dbms\_credit +

dsmarks\*ds\_credit + feemarks\*fee\_credit \* mathsmarks\*maths\_credit)/(500\*total\_credits);

update Result set spi=student\_spi where Result.student\_id = sid;

END LOOP;

-- Close the cursor

CLOSE Marks\_cursor;

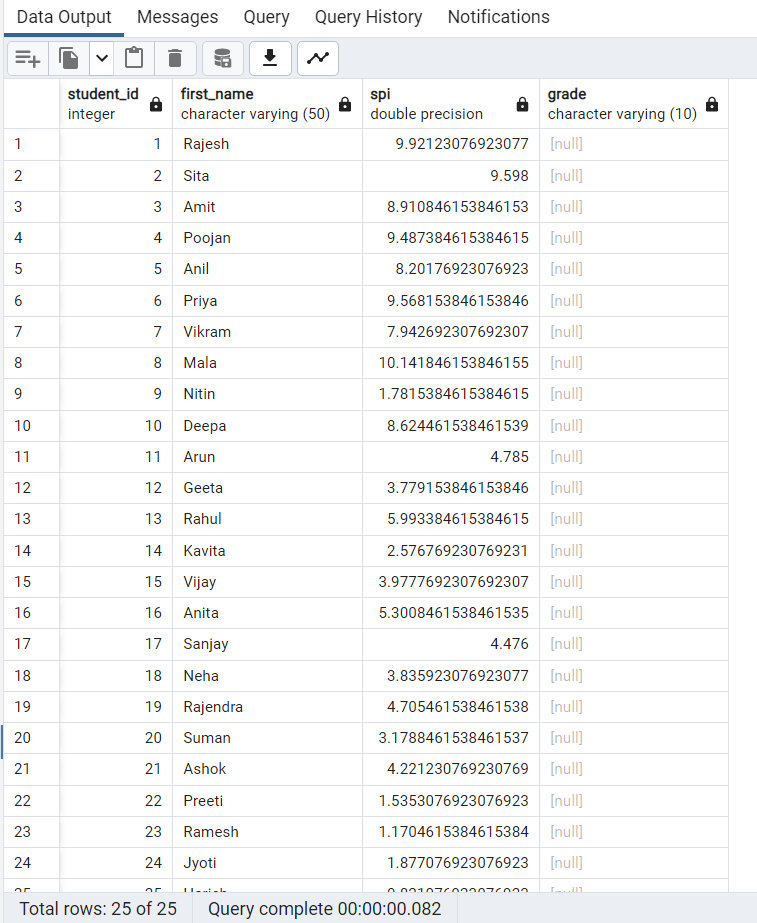
END;

$$ LANGUAGE plpgsql;

* Call the Procedure :

CALL calculate\_spi();

* Output :



* Create Procedure name as Grade\_calculator() :

create or replace procedure grade\_calculator()

as $$

declare

Student\_grade CURSOR FOR

SELECT student\_id, spi FROM Result;

rid Result.student\_id%type;

rspi Result.spi%type;

r\_grade varchar:='';

begin

OPEN Student\_grade;

-- Loop through subjects

LOOP

FETCH Student\_grade INTO rid,rspi;

EXIT WHEN NOT FOUND;

if rspi>9 then

r\_grade :='O++';

elsif rspi<9 and rspi>=8.5

then r\_grade :='O+';

elsif rspi<8.5 and rspi>=8

then r\_grade :='O';

elsif rspi<8 and rspi>=7.5

then r\_grade :='A++';

elsif rspi<7.5 and rspi>=7

then r\_grade :='A+';

elsif rspi<7 and rspi>=6.5

then r\_grade :='A';

elsif rspi<6.5 and rspi>=6

then r\_grade :='B++';

elsif rspi<6 and rspi>=5.5

then r\_grade :='B+';

elsif rspi<5.5 and rspi>=5

then r\_grade :='B';

elsif rspi<5 and rspi>=4

then r\_grade :='C';

else

r\_grade :='F';

insert into Result (grad) values (r\_grade) ;--where Result.student\_id = rid;

END if;

end loop;

-- Close the cursor

CLOSE Student\_grade;

end;

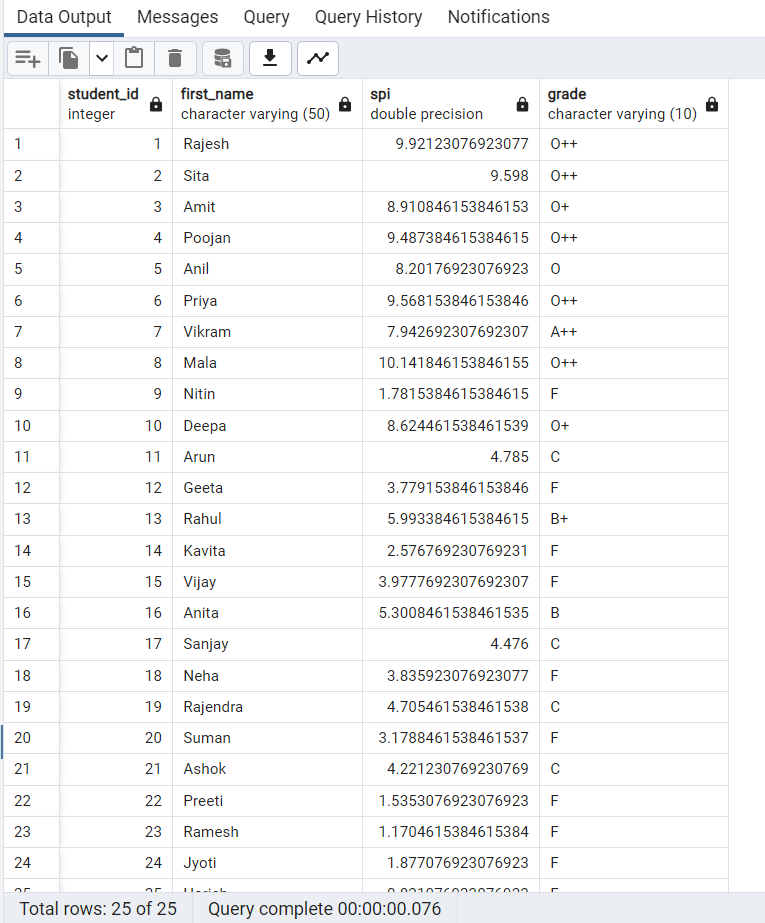
$$

LANGUAGE plpgsql;

* Call the Procedure :

call grade\_calculator();

* Output :



* Pl/pg sql block to display subject & credits :

do

$$

begin

raise notice'Credit of JAVA is 6.';

raise notice'Credit of DBMS is 6.';

raise notice'Credit of DS is 6.';

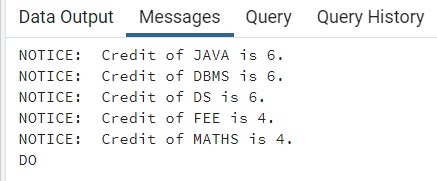
raise notice'Credit of FEE is 4.';

raise notice'Credit of MATHS is 4.';

END;

$$

* Output :



* Procedure to display the result of a spesify student :

create or replace procedure result\_student(sid in int)

as $$

declare

Student\_spesify\_result Result%rowtype;

begin

select \* into Student\_spesify\_result from Result where result.student\_id=sid;

if Student\_spesify\_result.student\_id=sid then

raise notice 'Student id = % ',Student\_spesify\_result.student\_id;

raise notice 'Student Name = % ',Student\_spesify\_result.first\_name;

raise notice 'Student SPI = % ',Student\_spesify\_result.spi;

raise notice 'Student Grade = % ',Student\_spesify\_result.grade;

else

raise notice 'id not found.';

end if;

end;

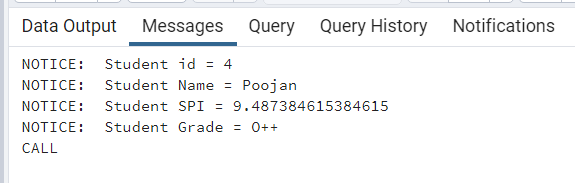
$$

LANGUAGE plpgsql;

* Call Procedure :

call result\_student(12);

* Output :



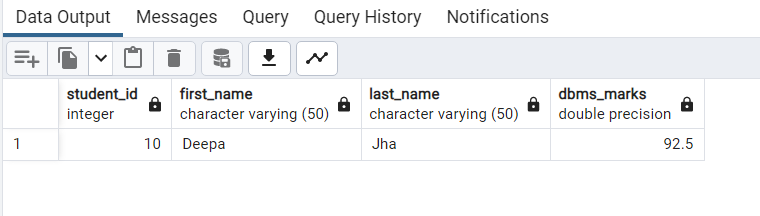
* Some Qurey above table :
* Qurey – 1 :
* Retrieve the students who scored the highest in DBMS:

SELECT students.student\_id, students.first\_name,students.last\_name, student\_marks.DBMS\_marks

FROM students INNER JOIN Student\_marks

ON students.student\_id = student\_marks.student\_id WHERE student\_marks.DBMS\_marks = (SELECT MAX(DBMS\_marks) FROM Student\_marks);

* Output :



* Qurey – 2 :
* Retrieve the student with the highest total marks (sum of all subjects) :

SELECT students.first\_name, result.spi, (

SELECT SUM(DBMS\_marks + JAVA\_marks + DS\_marks + FEE\_marks + Maths\_marks)

FROM Student\_marks

WHERE Student\_marks.student\_id = result.student\_id

) AS total\_marks

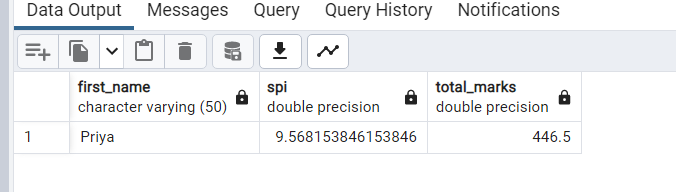
FROM students

JOIN Result ON students.student\_id = result.student\_id

ORDER BY total\_marks DESC

LIMIT 1;

* Output :



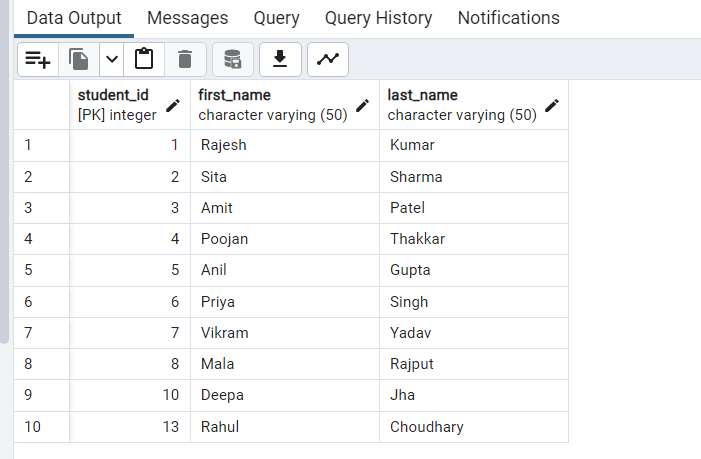
* Qurey – 3 :
* Retrieve the students who scored higher than the average SPI:

SELECT students.\* FROM students

WHERE students.student\_id IN (

SELECT result.student\_id FROM Result WHERE result.spi >

( SELECT AVG(result.spi) FROM Result ) );

Output : 

* Qurey – 4 :
* Retrieve the students who scored less than 'Geeta' in Java:

SELECT students.\* FROM students

WHERE students.student\_id <> 12

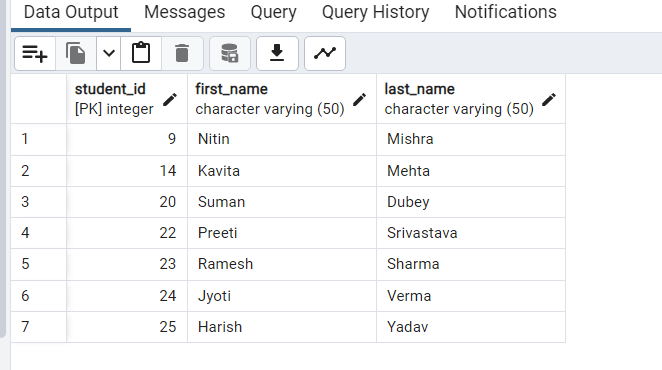
AND students.student\_id IN ( SELECT Student\_marks.student\_id

FROM Student\_marks WHERE Student\_marks.JAVA\_marks < (

SELECT JAVA\_marks FROM Student\_marks

WHERE student\_id = 12 ) );

* Output :



* Qurey – 5 :
* Retrieve the grades of students with SPI between 7 and 8:

Do

$$

DECLARE

student\_records cursor for select \* from result;

ss\_id result.student\_id%type;

f\_name result.first\_name%type;

s\_grade result.grade%type;

s\_spi result.spi%type;

BEGIN

OPEN Student\_records;

LOOP

FETCH Student\_records INTO ss\_id,f\_name,s\_spi,s\_grade;

EXIT WHEN NOT FOUND;

IF s\_spi >= 7 AND s\_spi <= 8 THEN

RAISE NOTICE 'Student Name: % Grade: % SPI: %', f\_name,s\_grade,s\_spi;

END IF;

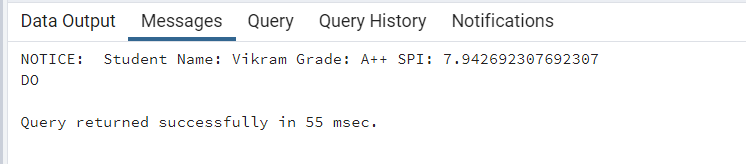
END LOOP;

CLOSE Student\_records;

END;

$$

* Output :



* Qurey – 6 :
* Retrieve the total marks obtained by each student, ordered by total marks in descending order:

SELECT

students .first\_name,

SUM(Student\_marks.DBMS\_marks + Student\_marks.JAVA\_marks + Student\_marks.DS\_marks + Student\_marks.FEE\_marks + Student\_marks.Maths\_marks) AS total\_marks

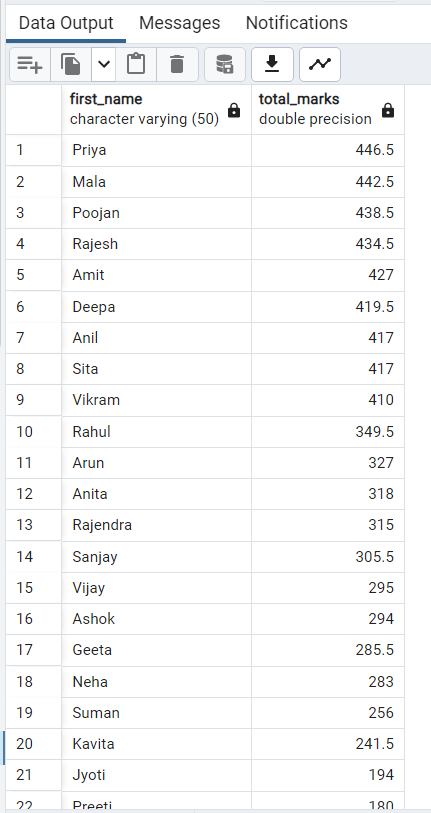
FROM students

JOIN Student\_marks ON Students.student\_id = Student\_marks.student\_id

GROUP BY students.first\_name

ORDER BY total\_marks DESC;

* Output :



* Qurey – 7 :
* List all students with their all subject marks and spi using a view:

CREATE OR REPLACE VIEW Student\_Marks\_Spi AS

SELECT result.student\_id,result.first\_name, Student\_marks.java\_marks,Student\_marks.dbms\_marks,

Student\_marks.ds\_marks,Student\_marks.fee\_marks,Student\_marks.maths\_marks,

Result.spi

FROM Student\_marks

JOIN Result ON student\_marks.student\_id = Result.student\_id;

SELECT \* FROM Student\_Marks\_Spi;

* Output :

