**College Data Management**

Name : Manav Mehta Name : Shivang Kosty

Roll No : 11 RollNo : 14

Enrollment No : 22002170110074 Enrollment No : 22002170110059

Branch : C.E. Batch : D-1 Branch : C.E. Batch : D-1

Name : Ansh Patel Name : Dhrumil Trada

Roll No : 12 RollNo : 25

Enrollment No : 22002170110104 Enrollment No : 22002170110194

Branch : C.E. Batch : D-1 Branch : C.E. Batch : D-1

Name : Jay Patel

Roll No : 26

Enrollment No : 22002170210070

Branch : I.T. Batch : D-1

Tables and their respective syntax :

1. CREATE TABLE Students (

Student\_ID SERIAL PRIMARY KEY,

Enrollment\_No BIGINT ,

First\_Name VARCHAR(50) NOT NULL,

Last\_Name VARCHAR(50) NOT NULL,

Date\_Of\_Birth DATE,

Gender VARCHAR(10) CHECK (Gender IN ('Male','Female')),

Contact\_Number VARCHAR(10) UNIQUE,

Email VARCHAR(100) UNIQUE,

Address VARCHAR(255),

Division VARCHAR(5),

Admission\_year INT,

Department\_ID INT,

FOREIGN KEY (Department\_ID) REFERENCES Departments(Department\_ID)

);

2. CREATE TABLE Courses (

Course\_ID INT PRIMARY KEY,

Course\_Name VARCHAR(50) NOT NULL,

Department\_ID INT,

Credits INT CHECK(Credits BETWEEN 0 AND 10),

FOREIGN KEY (Department\_ID) REFERENCES Departments(Department\_ID)

);

3. CREATE TABLE Results (

Result\_ID SERIAL PRIMARY KEY,

Student\_ID INT,

Course\_ID INT,

Score INT CHECK (Score BETWEEN 0 AND 100),

FOREIGN KEY (Student\_ID) REFERENCES Students(Student\_ID),

FOREIGN KEY (Course\_ID) REFERENCES Courses(Course\_ID)

);

4. CREATE TABLE Professors (

Professor\_ID SERIAL PRIMARY KEY,

First\_Name VARCHAR(50) NOT NULL,

Last\_Name VARCHAR(50) NOT NULL,

Date\_Of\_Birth DATE,

Gender VARCHAR(10) CHECK (Gender IN ('Male','Female')),

Contact\_Number VARCHAR(10) UNIQUE,

Email VARCHAR(100) UNIQUE,

Course\_ID INT,

FOREIGN KEY (Course\_ID) REFERENCES Courses(Course\_ID)

);

5. CREATE TABLE Departments (

Department\_ID SERIAL PRIMARY KEY,

Department\_Name VARCHAR(50)

);

6. CREATE TABLE Timetable (

Timetable\_ID SERIAL PRIMARY KEY,

Division VARCHAR(10),

Professor\_ID INT,

Day\_Of\_Week VARCHAR(15),

Start\_Time TIME,

End\_Time TIME,

Class\_ID INT,

FOREIGN KEY (Professor\_ID) REFERENCES Professors(Professor\_ID),

FOREIGN KEY (Class\_ID) REFERENCES Classes(Class\_ID)

);

7. CREATE TABLE Classes (

Class\_ID SERIAL PRIMARY KEY,

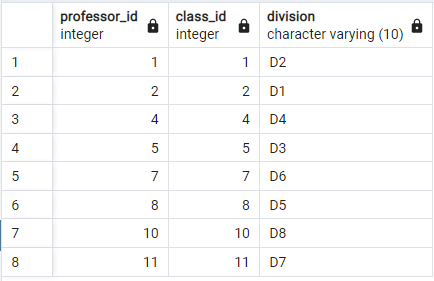
Class\_Number VARCHAR(10)

);

Queries for different questions and their respective output :

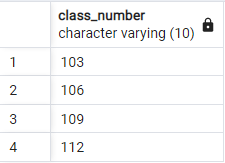
1. **Query to find professor at a certain time and the division he/she is teaching at particular day and time.**

* select professor\_id, class\_id, division from timetable where day\_of\_week = 'TUESDAY' and Start\_time < '12:00:00' and end\_time > '12:00:00' order by professor\_id;



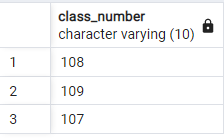
1. **Query to find which classes are empty at particular day and time.**

* select Class\_number from classes where class\_id not in(select class\_id from timetable where day\_of\_week = 'MONDAY' and Start\_time < '12:00:00' and end\_time > '12:00:00' );



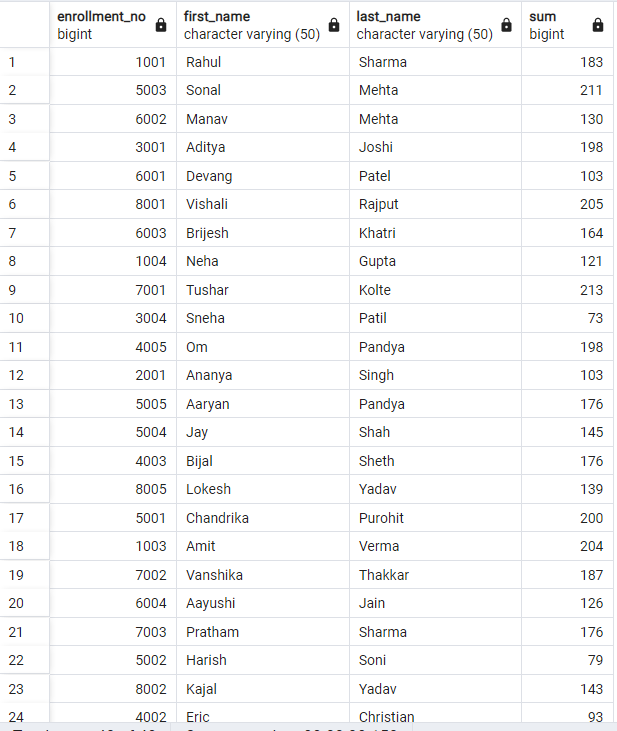
1. **Query to find which classes are assigned to particular department.**

* select class\_number from classes where class\_id in(select class\_id from timetable where professor\_id in (select professor\_id from professors where course\_id in (select course\_id from courses where department\_id in (select department\_id from departments where department\_name = 'Civil'))))

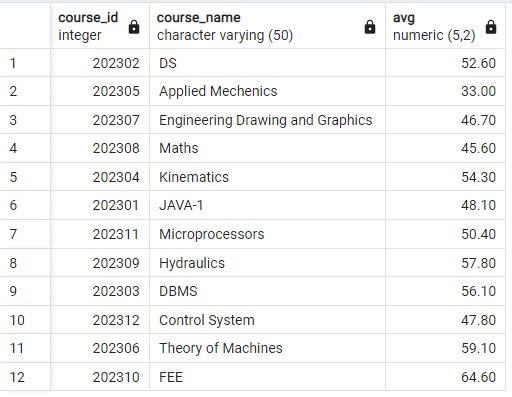


1. **Query to find total marks of all students in all subjects.**

* select enrollment\_no,first\_name,last\_name,sum(score) from students join results on students.Student\_id = results.student\_id group by results.Student\_id,enrollment\_no,first\_name,last\_name;

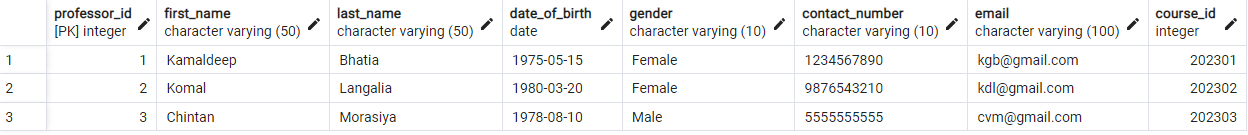


1. **Query to find average marks course/subject wise.**

* select results.course\_id,course\_name,cast(avg(score) as decimal(5,2)) from courses join results on courses.course\_id = results.course\_id group by results.course\_id,course\_name;
* 

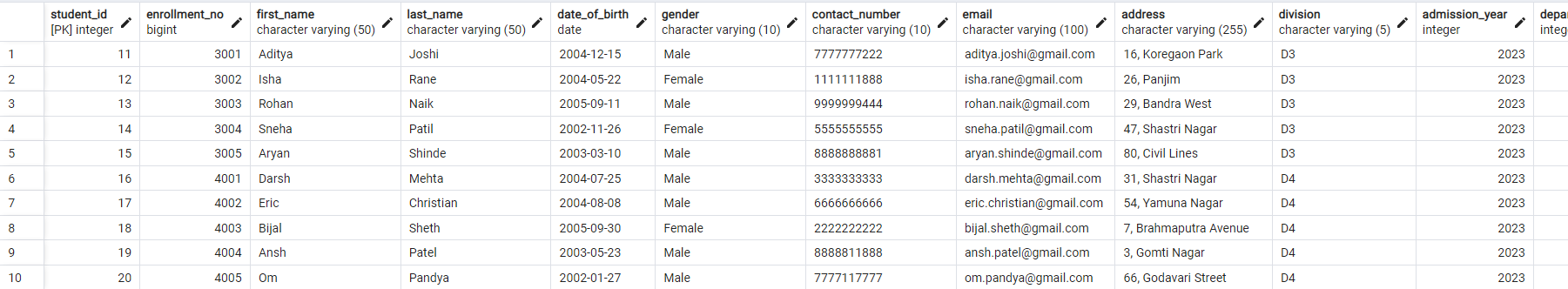
1. **Query to display details of professors in particular department.**

* select \* from professors where course\_id in (select course\_id from courses where department\_id in (select department\_id from departments where department\_name ='CS'));



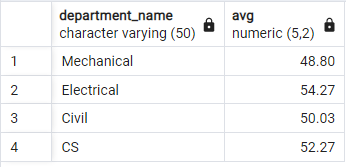
1. **Query to display details of students in particular department.**

* select \* from students where department\_id in (select department\_id from departments where department\_name ='Mechanical');



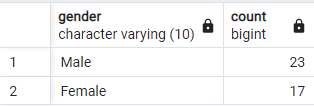
1. **Query to display average marks of all department.**

* select department\_name,cast(avg(score) as decimal(5,2)) from departments join courses on departments.department\_id = courses.department\_id join results on results.course\_id = courses.course\_id group by department\_name;



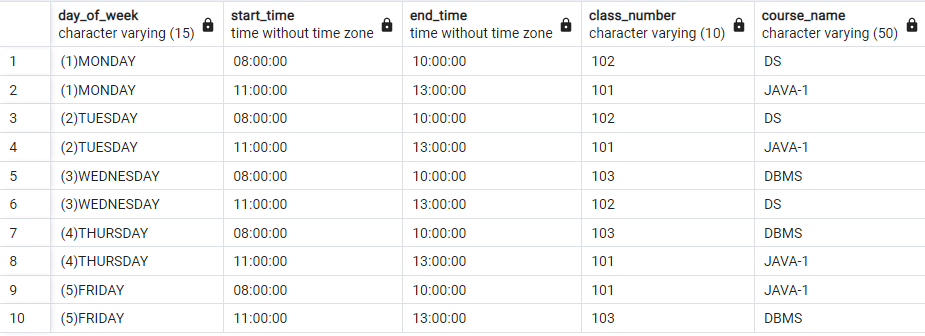
1. **Query to find count of male and female students.**

* select gender,count(gender) from students group by gender;



1. **Query to print timetable of a particular department.**

* select day\_of\_week,start\_time,end\_time,class\_number,course\_name from timetable,classes,professors,courses where timetable.class\_id = classes.class\_id and timetable.professor\_id = professors.professor\_id and professors.course\_id=courses.course\_id and Division = 'D2' order by day\_of\_week,start\_time ;



**FUNCTIONS :**

1. **To display students by their given divisions :**

* CREATE OR REPLACE PROCEDURE display\_division\_wise(div VARCHAR) AS $$

DECLARE

var students%ROWTYPE;

c1 CURSOR FOR SELECT \* FROM students WHERE Division = div;

BEGIN

OPEN c1;

LOOP

FETCH c1 INTO var;

EXIT WHEN NOT FOUND;

RAISE NOTICE 'Enrollment\_No = % , Name = % % , Gender = % , DOB = % , Email = %',var.Enrollment\_No,var.First\_Name,var.Last\_Name,var.Gender, var.Date\_Of\_Birth, var.Email ;

END LOOP;

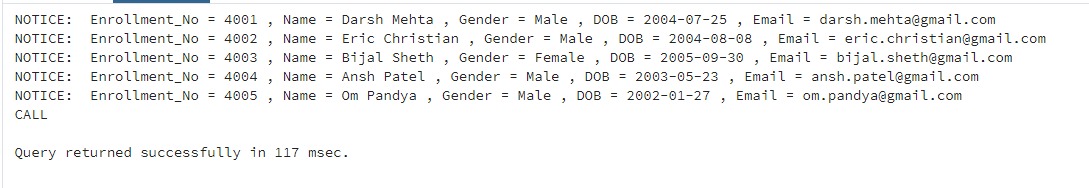
CLOSE c1;

END;

$$

LANGUAGE plpgsql;

* CALL display\_division\_wise('D4');



1. **To calculate spi from given enrolment number of a student :**

* CREATE OR REPLACE PROCEDURE calculate\_spi(enroll\_no INT) AS $$

DECLARE

sub NUMERIC = 0;

cred INT = 0;

total NUMERIC(5,2) = 0;

c\_id INT;

var Students%ROWTYPE;

c1 CURSOR FOR SELECT Score,Course\_ID FROM Results WHERE Student\_id IN (SELECT Student\_id FROM Students WHERE Enrollment\_No = enroll\_no);

BEGIN

SELECT \* FROM Students WHERE Enrollment\_No = enroll\_no INTO var;

OPEN c1;

LOOP

FETCH c1 INTO sub,c\_id;

EXIT WHEN NOT FOUND;

SELECT Credits FROM Courses WHERE course\_id = c\_id INTO cred;

total = total + (sub \* cred);

END LOOP;

CLOSE c1;

total = total / 15;

RAISE NOTICE 'Enrollment\_No : %',var.Enrollment\_No;

RAISE NOTICE 'Name : % %',var.First\_Name,var.Last\_Name;

RAISE NOTICE 'SPI : %',total;

END;

$$

LANGUAGE plpgsql;

* CALL calculate\_spi(1002);

