

**SCHOOL OF COMPUTER SCIENCE ENGINEERING**

**AND INFORMATION SYSTEMS**

**FALL SEMESTER 2024-2025**

**PMCA503P – DATABASE SYSTEMS LAB**

**CYCLESHEET – SQL**

**SUBMITTED ON: 08 – SEPT - 2024**

**SUBMITTED BY-**

**AKASH KUMAR BANIK**

**PROGRAM: MCA**

**REGISTER No.: 24MCA0242**

**CYCLESHEET – SQL**

Consider the following relational database schema for teaching-learning process in a university.

(Source: Database Systems – Coronel & Morris)

PROFESSOR(Prof\_id, Prof\_name, Email, Mobile, Specialty, Dept\_id)

SCHOOL(SCode, Scl\_name, Prof\_id, Location)

DEPARTMENT(Dept\_id, Dname, SCode, Prof\_id)

COURSE(Crs\_code, Crs\_name, Description, Credits, Hours)

CLASS(Cls\_code, Slot, Stime, Etime, Crs\_code, Prof\_id, Room\_no, Sem\_code, Day\_of\_week)

SEMESTER(Sem\_code, Term, Year, Sdate, Edate)

STUDENT(Reg\_no, Sname, Address, DoB, Email, Mobile, Dept\_id, Prof\_id)

ENROLL(Cls\_code, Reg\_no, Enroll\_time, Grade)

STUDENT\_VISA(Reg\_no, Visa\_status)

PROGRAMME(Prog\_code, Prog\_name, Prog\_preamble, Scode, Dept\_id)

The primary keys are underlined and foreign keys are self-explanatory. The Dept\_id column in professor table stands for the department the professor belongs to and Prof\_id column in the school table stands for the professor who chairs the school, the same column in the department table stands for the professor who heads the department, the domain of Term column in semester table is {Winter, Fall}.

**1. Create the above tables.**

**PROFESSOR TABLE:**

CREATE TABLE PROFESSOR\_24MCA0242 (

    Prof\_id VARCHAR2(5),

    Prof\_name VARCHAR2(30) NOT NULL,

    Email VARCHAR2(30) NOT NULL,

    Mobile NUMBER(10) NOT NULL,

    Specialty VARCHAR2(30),

    Dept\_id VARCHAR2(5),

    CONSTRAINT pk\_professor PRIMARY KEY (Prof\_id),

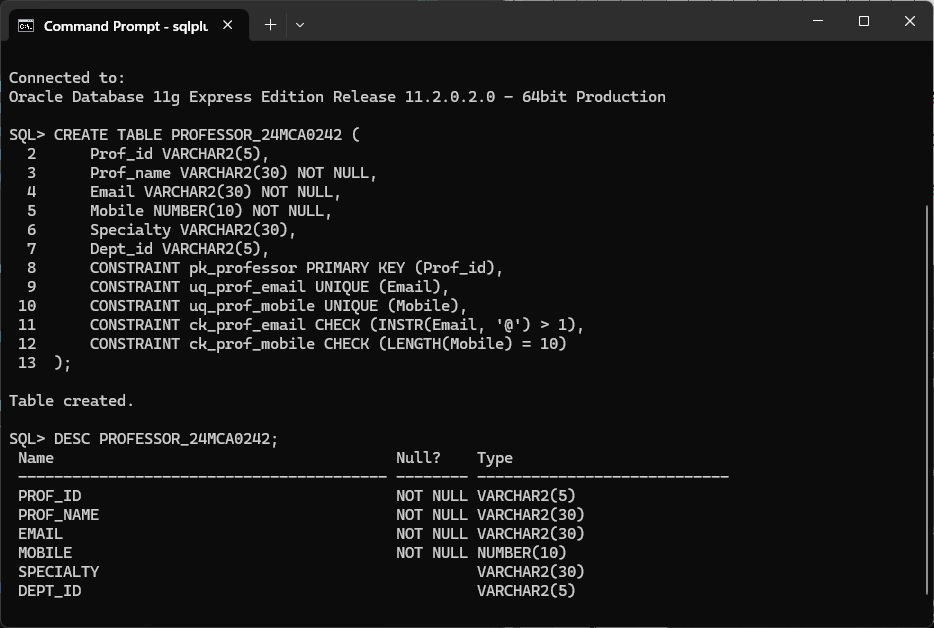
    CONSTRAINT uq\_prof\_email UNIQUE (Email),

    CONSTRAINT uq\_prof\_mobile UNIQUE (Mobile),

    CONSTRAINT ck\_prof\_email CHECK (INSTR(Email, '@') > 1),

    CONSTRAINT ck\_prof\_mobile CHECK (LENGTH(Mobile) = 10)

);



**SCHOOL TABLE:**

CREATE TABLE SCHOOL\_24MCA0242 (

    SCode VARCHAR2(5),

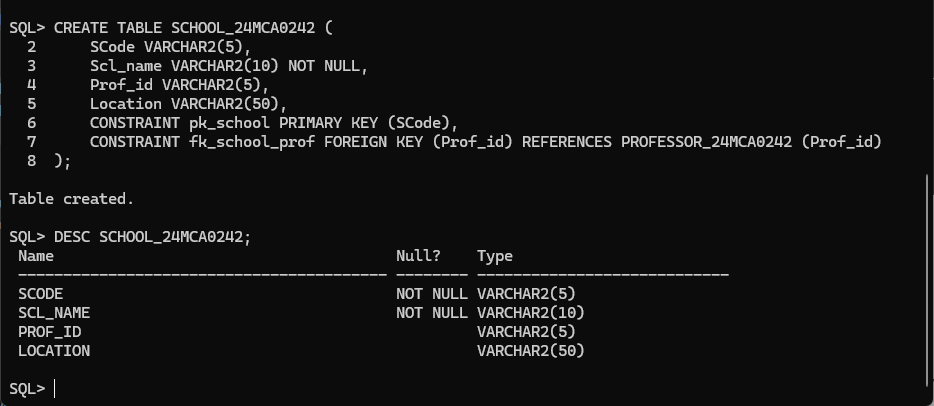
    Scl\_name VARCHAR2(10) NOT NULL,

    Prof\_id VARCHAR2(5),

    Location VARCHAR2(50),

    CONSTRAINT pk\_school PRIMARY KEY (SCode),

CONSTRAINT fk\_school\_prof FOREIGN KEY (Prof\_id) REFERENCES PROFESSOR\_24MCA0242 (Prof\_id)

); 

**DEPARTMENT TABLE:**

CREATE TABLE DEPARTMENT\_24MCA0242 (

    Dept\_id VARCHAR2(5),

    Dname VARCHAR2(50) NOT NULL,

    SCode VARCHAR2(5),

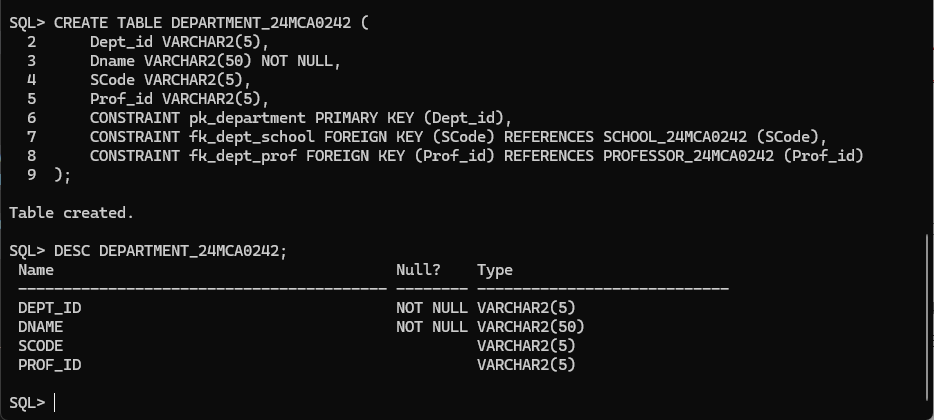
    Prof\_id VARCHAR2(5),

    CONSTRAINT pk\_department PRIMARY KEY (Dept\_id),

CONSTRAINT fk\_dept\_school FOREIGN KEY (SCode) REFERENCES SCHOOL\_24MCA0242 (SCode),

CONSTRAINT fk\_dept\_prof FOREIGN KEY (Prof\_id) REFERENCES PROFESSOR\_24MCA0242 (Prof\_id)

);



**COURSE TABLE:**

CREATE TABLE COURSE\_24MCA0242 (

    Crs\_code VARCHAR2(5),

    Crs\_name VARCHAR2(50) NOT NULL,

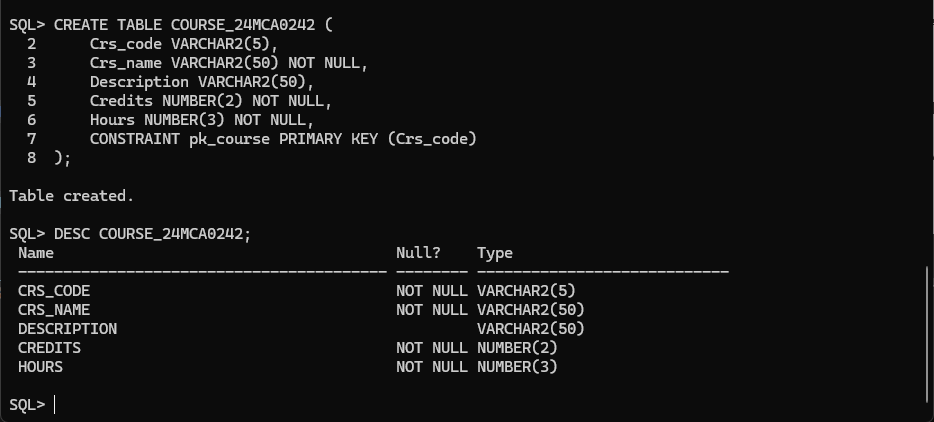
    Description VARCHAR2(50),

    Credits NUMBER(2) NOT NULL,

    Hours NUMBER(3) NOT NULL,

    CONSTRAINT pk\_course PRIMARY KEY (Crs\_code)

);



**STUDENT TABLE:**

CREATE TABLE STUDENT\_24MCA0242 (

    Reg\_no VARCHAR2(5),

    Sname VARCHAR2(30) NOT NULL,

    Address VARCHAR2(50),

    DoB DATE NOT NULL,

    Email VARCHAR2(30) NOT NULL,

    Mobile NUMBER(10) NOT NULL,

    Dept\_id VARCHAR2(5),

    Prof\_id VARCHAR2(5),

    CONSTRAINT pk\_student PRIMARY KEY (Reg\_no),

CONSTRAINT fk\_student\_dept FOREIGN KEY (Dept\_id) REFERENCES DEPARTMENT\_24MCA0242 (Dept\_id),

CONSTRAINT fk\_student\_prof FOREIGN KEY (Prof\_id) REFERENCES PROFESSOR\_24MCA0242 (Prof\_id),

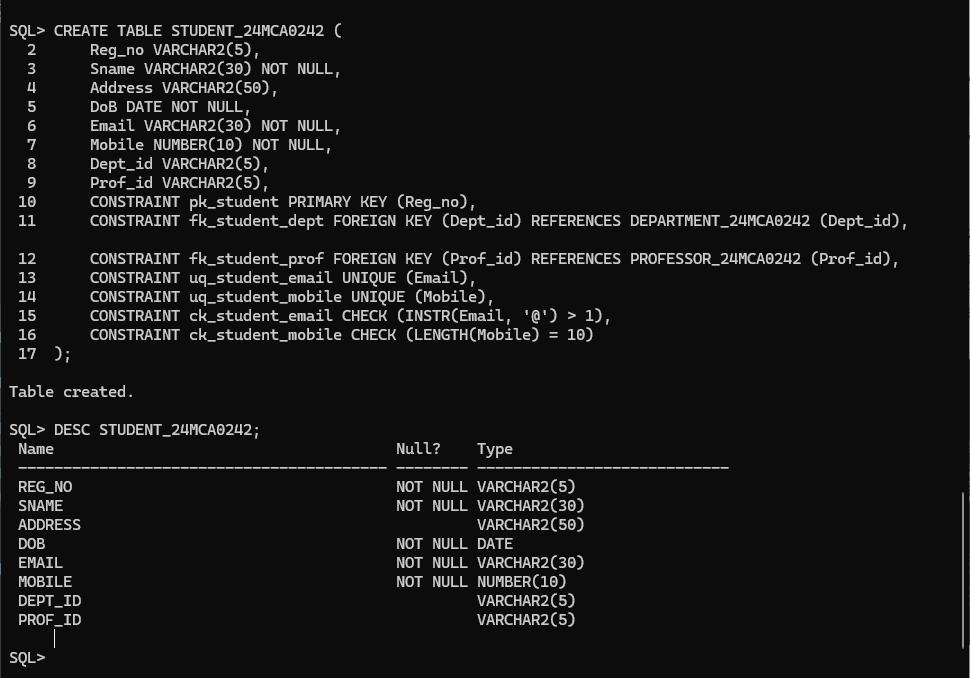
    CONSTRAINT uq\_student\_email UNIQUE (Email),

    CONSTRAINT uq\_student\_mobile UNIQUE (Mobile),

    CONSTRAINT ck\_student\_email CHECK (INSTR(Email, '@') > 1),

    CONSTRAINT ck\_student\_mobile CHECK (LENGTH(Mobile) = 10)

);



**PROGRAMME TABLE:**

CREATE TABLE PROGRAMME\_24MCA0242 (

    Prog\_code VARCHAR2(5),

    Prog\_name VARCHAR2(30) NOT NULL,

    Prog\_preamble VARCHAR2(50),

    Scode VARCHAR2(5),

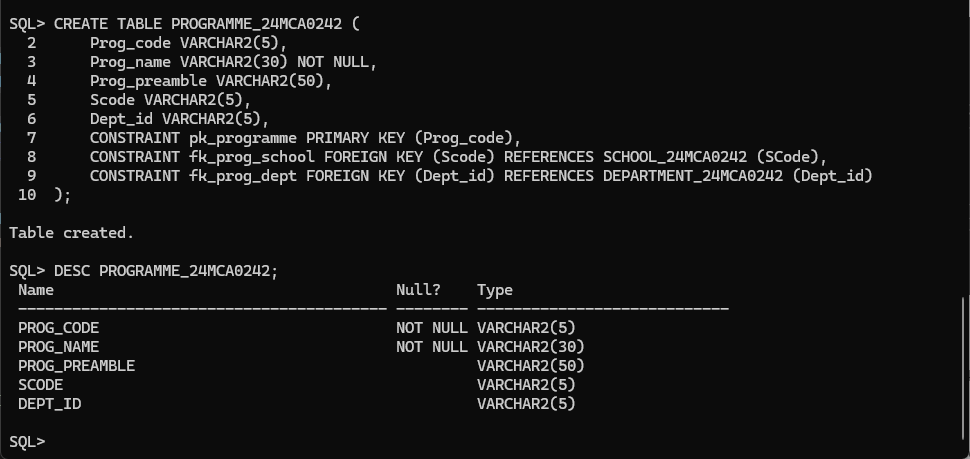
    Dept\_id VARCHAR2(5),

    CONSTRAINT pk\_programme PRIMARY KEY (Prog\_code),

CONSTRAINT fk\_prog\_school FOREIGN KEY (Scode) REFERENCES SCHOOL\_24MCA0242 (SCode),

CONSTRAINT fk\_prog\_dept FOREIGN KEY (Dept\_id) REFERENCES DEPARTMENT\_24MCA0242 (Dept\_id)

);



**SEMESTER TABLE:**

CREATE TABLE SEMESTER\_24MCA0242 (

    Sem\_code VARCHAR2(10),

    Term VARCHAR2(5) CHECK (Term IN ('Winter', 'Fall')),

    Year NUMBER(4) NOT NULL,

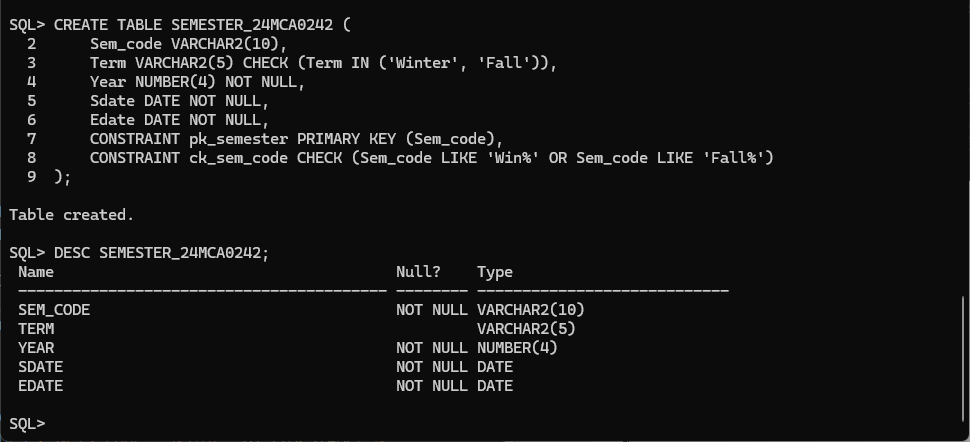
    Sdate DATE NOT NULL,

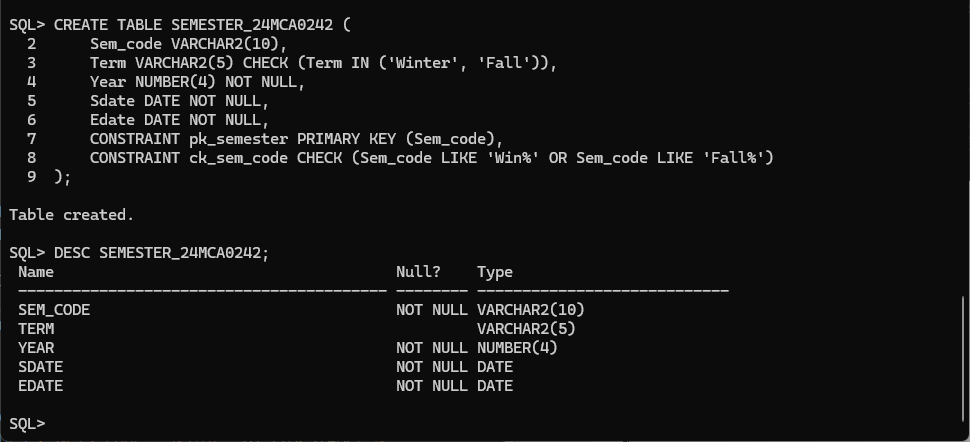
    Edate DATE NOT NULL,

    CONSTRAINT pk\_semester PRIMARY KEY (Sem\_code),

CONSTRAINT ck\_sem\_code CHECK (Sem\_code LIKE 'Win%' OR Sem\_code LIKE 'Fall%')

);





**CLASS TABLE:**

CREATE TABLE CLASS\_24MCA0242 (

    Cls\_code VARCHAR2(5),

    Slot VARCHAR2(10) NOT NULL,

    Stime TIMESTAMP(0) NOT NULL,

    Etime TIMESTAMP(0) NOT NULL,

    Crs\_code VARCHAR2(5),

    Prof\_id VARCHAR2(5),

    Room\_no VARCHAR2(10),

    Sem\_code VARCHAR2(10),

    Day\_of\_week VARCHAR2(10),

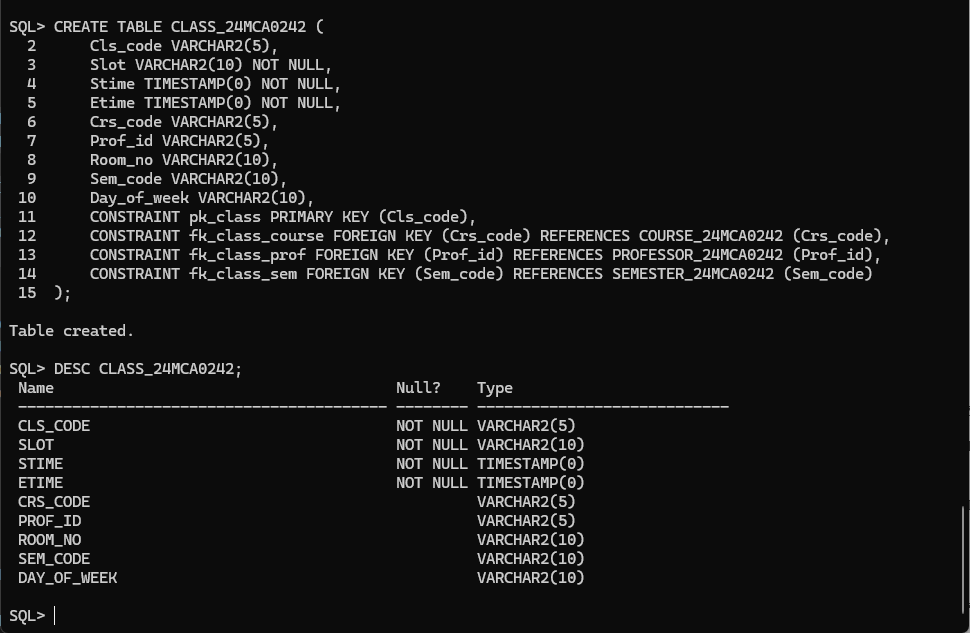
    CONSTRAINT pk\_class PRIMARY KEY (Cls\_code),

CONSTRAINT fk\_class\_course FOREIGN KEY (Crs\_code) REFERENCES COURSE\_24MCA0242 (Crs\_code),

CONSTRAINT fk\_class\_prof FOREIGN KEY (Prof\_id) REFERENCES PROFESSOR\_24MCA0242 (Prof\_id),

CONSTRAINT fk\_class\_sem FOREIGN KEY (Sem\_code) REFERENCES SEMESTER\_24MCA0242 (Sem\_code)

);



**ENROLL TABLE:**

CREATE TABLE ENROLL\_24MCA0242 (

    Cls\_code VARCHAR2(5),

    Reg\_no VARCHAR2(5),

    Enroll\_time TIMESTAMP(0) NOT NULL,

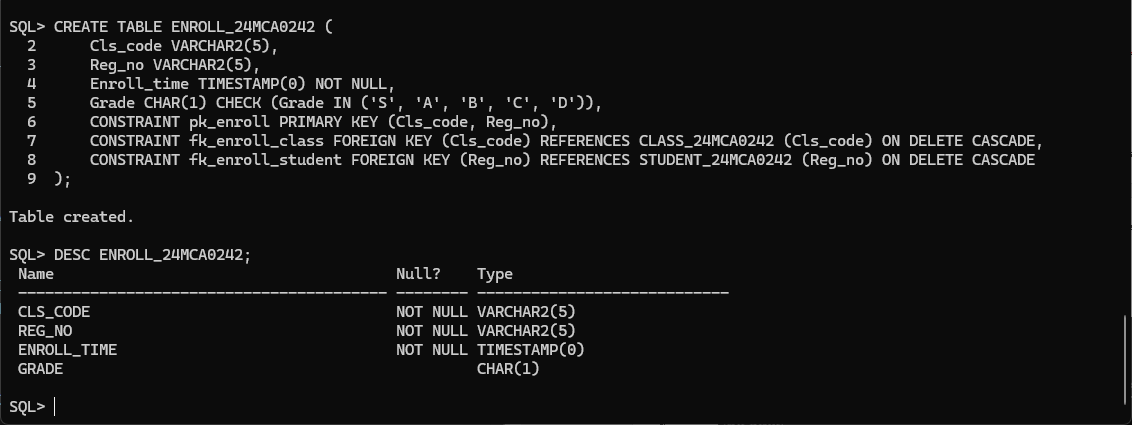
    Grade CHAR(1) CHECK (Grade IN ('S', 'A', 'B', 'C', 'D')),

    CONSTRAINT pk\_enroll PRIMARY KEY (Cls\_code, Reg\_no),

CONSTRAINT fk\_enroll\_class FOREIGN KEY (Cls\_code) REFERENCES CLASS\_24MCA0242 (Cls\_code) ON DELETE CASCADE,

CONSTRAINT fk\_enroll\_student FOREIGN KEY (Reg\_no) REFERENCES STUDENT\_24MCA0242 (Reg\_no) ON DELETE CASCADE

);



**STUDENT\_VISA TABLE:**

CREATE TABLE STUDENT\_VISA\_24MCA0242 (

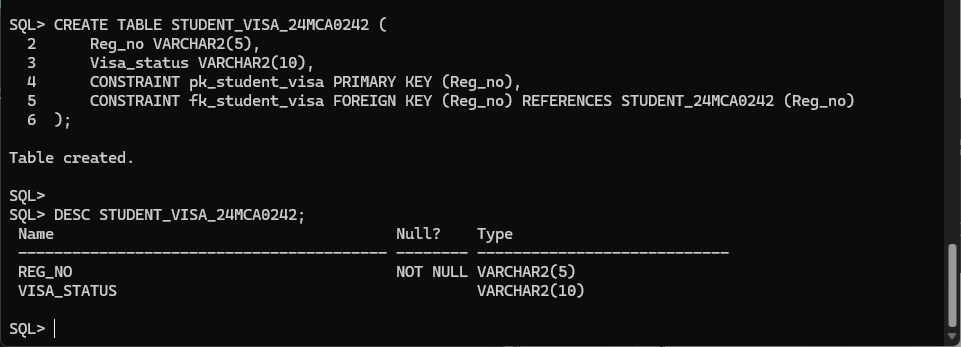
    Reg\_no VARCHAR2(5),

    Visa\_status VARCHAR2(10),

    CONSTRAINT pk\_student\_visa PRIMARY KEY (Reg\_no),

CONSTRAINT fk\_student\_visa FOREIGN KEY (Reg\_no) REFERENCES STUDENT\_24MCA0242 (Reg\_no)

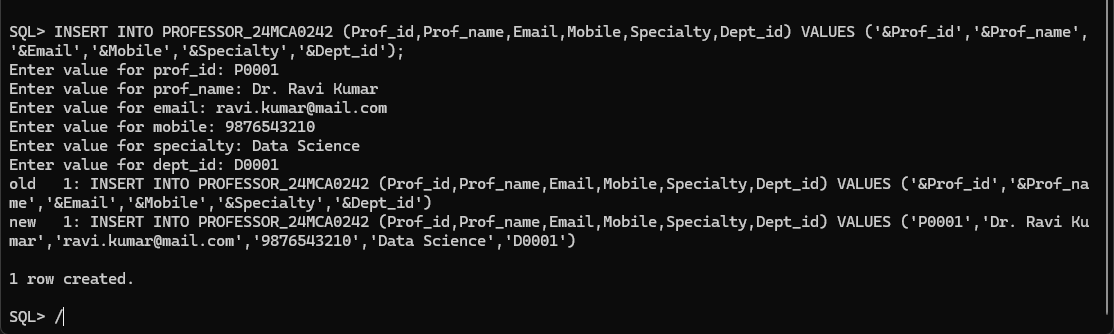
);



**2. Enter data into the above tables. Display the content of each table. Use column formatting while displaying.**

**PROFESSOR TABLE:**

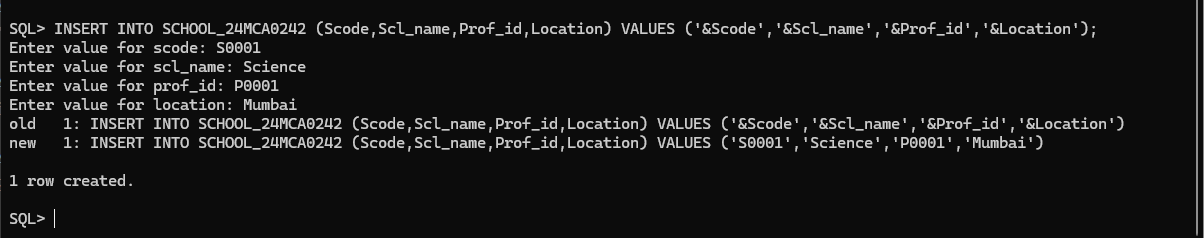
INSERT INTO PROFESSOR\_24MCA0242 (Prof\_id, Prof\_name, Email, Mobile, Specialty, Dept\_id) VALUES ('&Prof\_id','&Prof\_name','&Email','&Mobile','&Specialty','&Dept\_id');





**SCHOOL TABLE:**

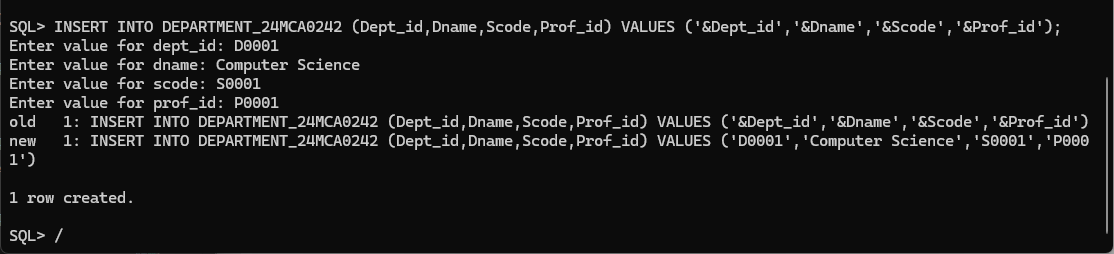
INSERT INTO SCHOOL\_24MCA0242 (Scode, Scl\_name, Prof\_id, Location) VALUES ('&Scode', '&Scl\_name', '&Prof\_id', '&Location');

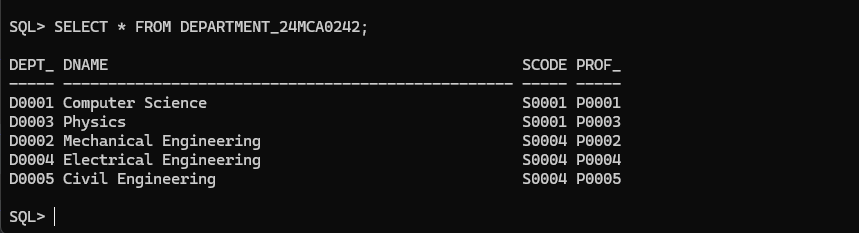




**DEPARTMENT TABLE:**

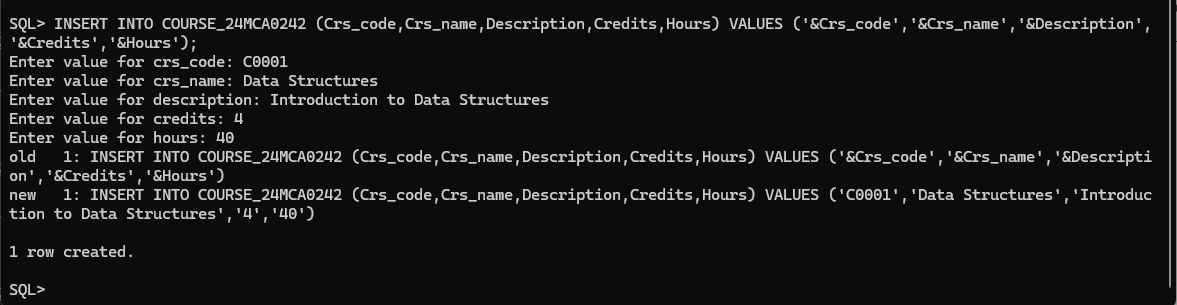
INSERT INTO DEPARTMENT\_24MCA0242 (Dept\_id, Dname, Scode, Prof\_id) VALUES ('&Dept\_id', '&Dname', '&Scode', '&Prof\_id');





**COURSE TABLE:**

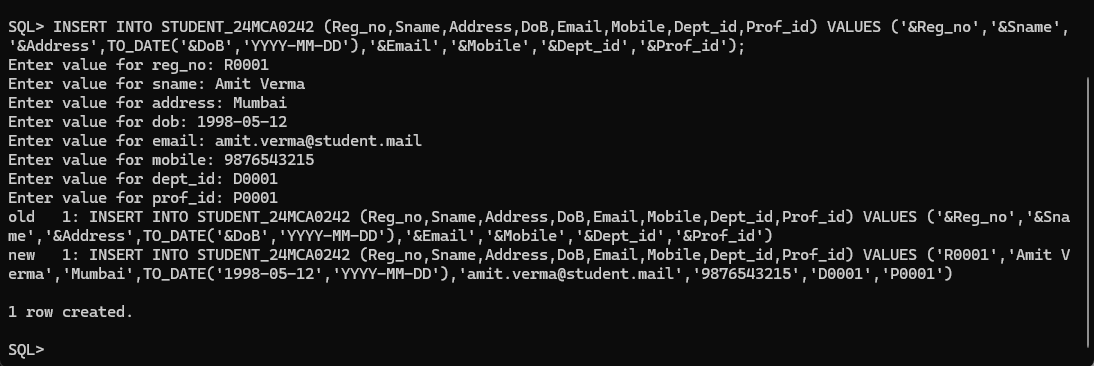
INSERT INTO COURSE\_24MCA0242 (Crs\_code, Crs\_name, Description, Credits, Hours) VALUES ('&Crs\_code', '&Crs\_name', '&Description', '&Credits', '&Hours');





**STUDENT TABLE:**

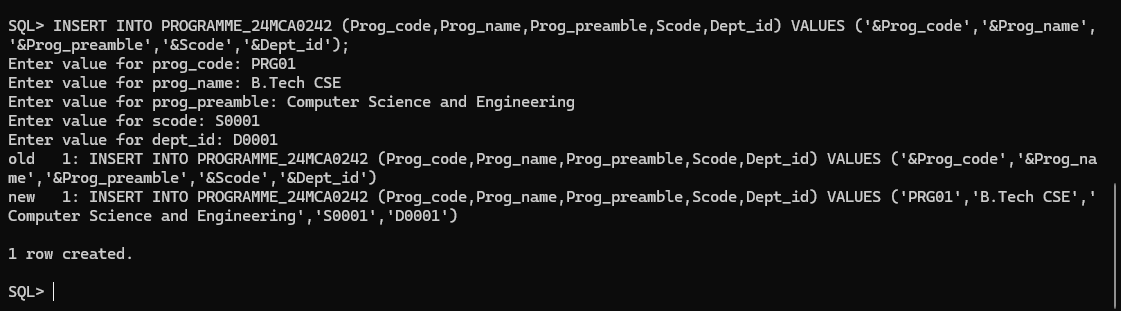
INSERT INTO STUDENT\_24MCA0242 (Reg\_no, Sname, Address, DoB, Email, Mobile, Dept\_id, Prof\_id) VALUES ('&Reg\_no', '&Sname', '&Address', TO\_DATE('&DoB','YYYY-MM-DD'), '&Email', '&Mobile', '&Dept\_id', '&Prof\_id');

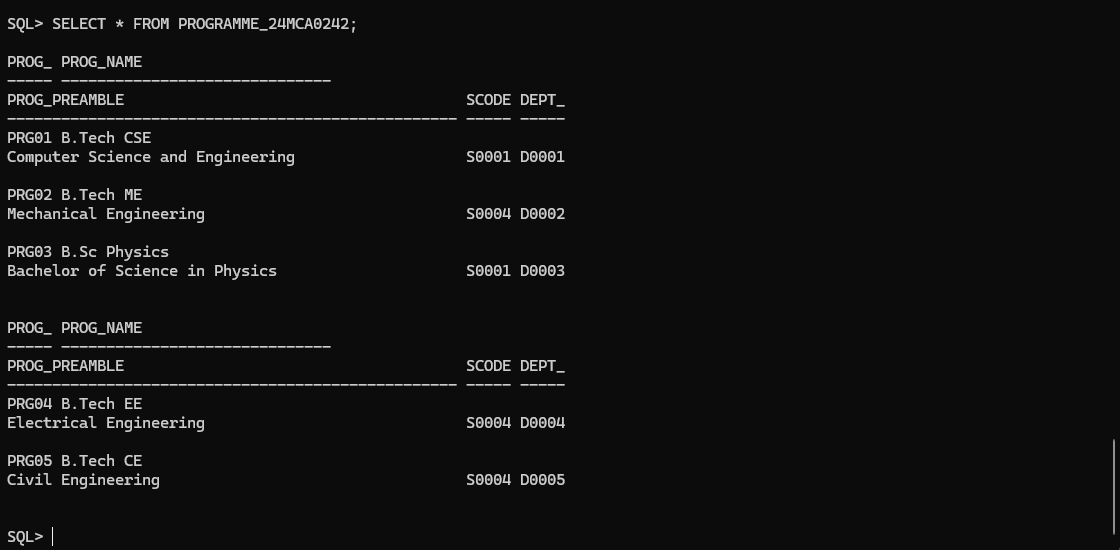




**PROGRAMME TABLE:**

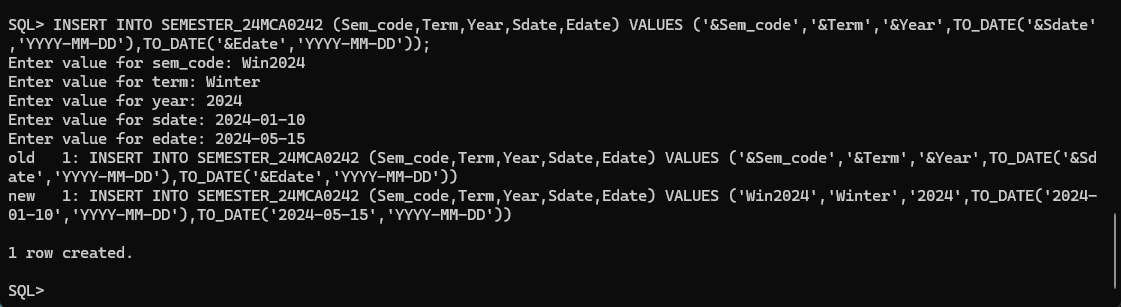
INSERT INTO PROGRAMME\_24MCA0242 (Prog\_code, Prog\_name, Prog\_preamble, Scode, Dept\_id) VALUES ('&Prog\_code', '&Prog\_name', '&Prog\_preamble', '&Scode', '&Dept\_id');





**SEMESTER TABLE:**

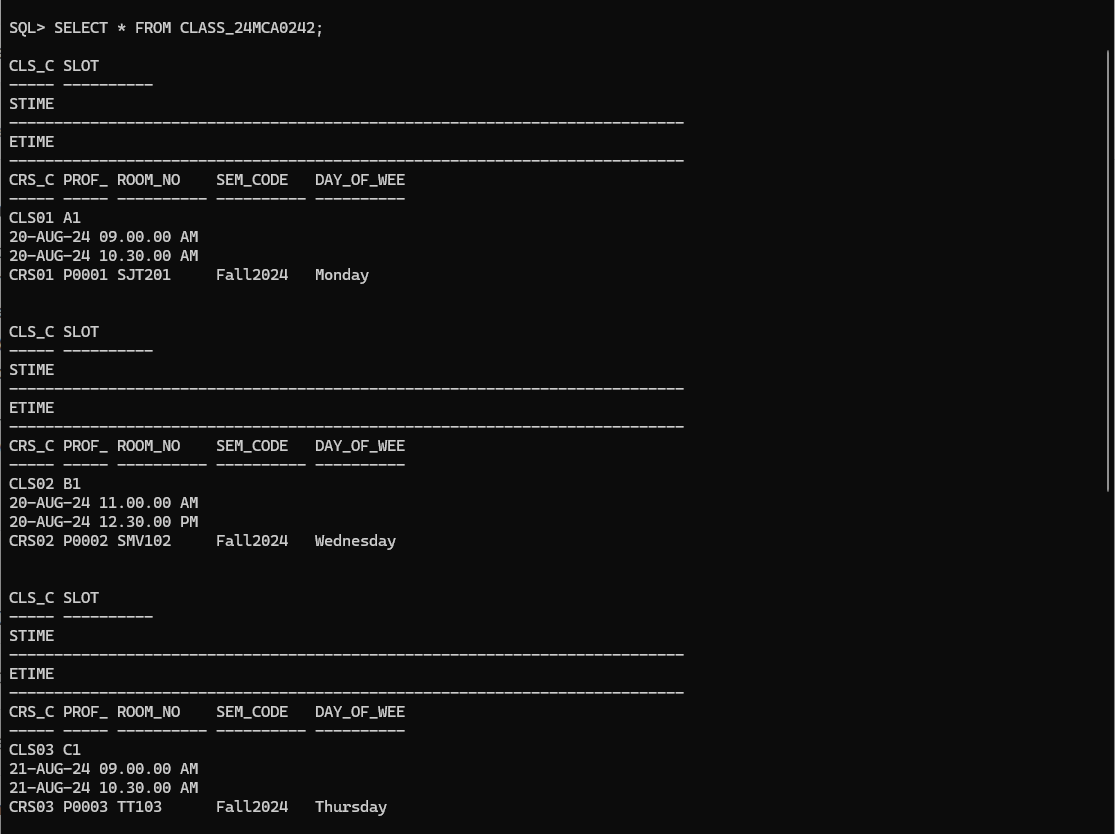
INSERT INTO SEMESTER\_24MCA0242 (Sem\_code, Term, Year, Sdate, Edate) VALUES ('&Sem\_code','&Term','&Year', TO\_DATE('&Sdate', 'YYYY-MM-DD'), TO\_DATE('&Edate', 'YYYY-MM-DD'));





**CLASS TABLE:**

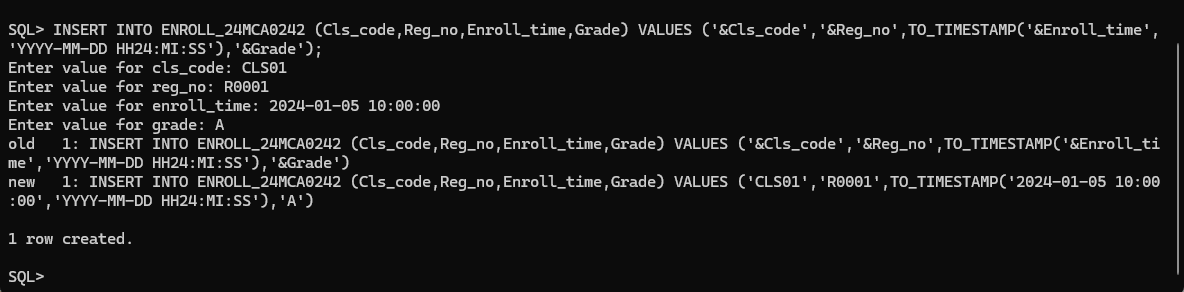
INSERT INTO CLASS\_24MCA0242 (Cls\_code, Slot, Stime, Etime, Crs\_code, Prof\_id, Room\_no, Sem\_code, Day\_of\_week) VALUES ('&Cls\_code', '&Slot', TO\_TIMESTAMP('&Stime', 'YYYY-MM-DD HH24:MI:SS'), TO\_TIMESTAMP('&Etime', 'YYYY-MM-DD HH24:MI:SS'), '&Crs\_code', '&Prof\_id', '&Room\_no', '&Sem\_code', '&Day\_of\_week');





**ENROLL TABLE:**

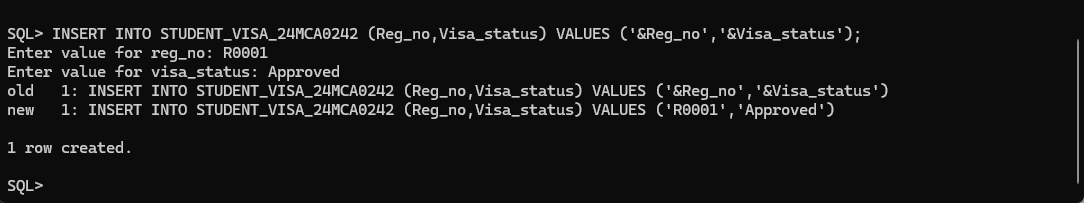
INSERT INTO ENROLL\_24MCA0242 (Cls\_code, Reg\_no, Enroll\_time, Grade) VALUES ('&Cls\_code', '&Reg\_no', TO\_TIMESTAMP('&Enroll\_time','YYYY-MM-DD HH24:MI:SS'), '&Grade');

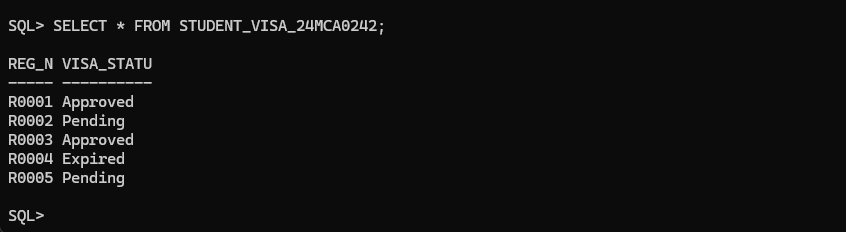




**STUDENT\_VISA TABLE:**

INSERT INTO STUDENT\_VISA\_24MCA0242 (Reg\_no, Visa\_status) VALUES ('&Reg\_no', '&Visa\_status');





**3. Alter or Recreate the above tables with primary key and foreign key and the following integrity constraints assigning name to integrity constraint.**

1. Prof\_id must have exactly five characters and their email and mobile number are unique. The email address must have @ as one of the characters and mobile number must have exactly ten characters.

ALTER TABLE PROFESSOR\_24MCA0242

ADD CONSTRAINT ck\_prof\_id\_length CHECK (LENGTH(Prof\_id) = 5);

ALTER TABLE PROFESSOR\_24MCA0242

ADD CONSTRAINT uq\_prof\_email UNIQUE (Email);

ALTER TABLE PROFESSOR\_24MCA0242

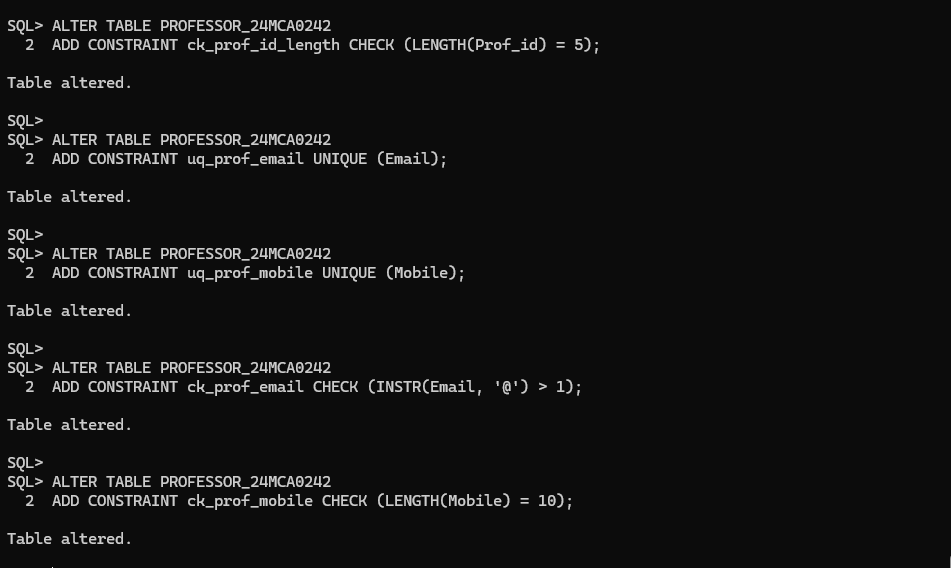
ADD CONSTRAINT uq\_prof\_mobile UNIQUE (Mobile);

ALTER TABLE PROFESSOR\_24MCA0242

ADD CONSTRAINT ck\_prof\_email CHECK (INSTR(Email, '@') > 1);

ALTER TABLE PROFESSOR\_24MCA0242

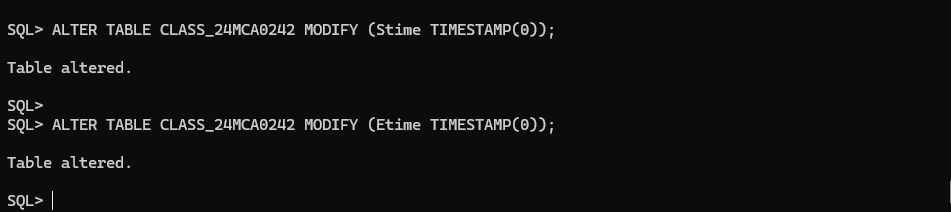
ADD CONSTRAINT ck\_prof\_mobile CHECK (LENGTH(Mobile) = 10);



1. Use timestamp data type without fractional parts of seconds for start time and end time column of class table.

ALTER TABLE CLASS\_24MCA0242 MODIFY (Stime TIMESTAMP(0));

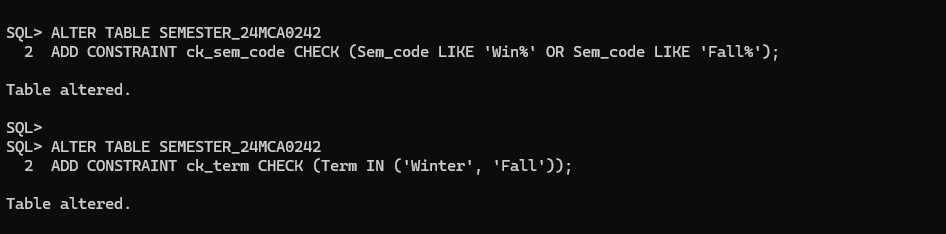
ALTER TABLE CLASS\_24MCA0242 MODIFY (Etime TIMESTAMP(0));



1. The Sem\_code should start with either ‘Win’ or ‘Fall’ and Term column can assume only one of two values {Winter, Fall}.

ALTER TABLE SEMESTER\_24MCA0242 ADD CONSTRAINT ck\_sem\_code CHECK (Sem\_code LIKE 'Win%' OR Sem\_code LIKE 'Fall%');

ALTER TABLE SEMESTER\_24MCA0242 ADD CONSTRAINT ck\_term CHECK (Term IN ('Winter', 'Fall'));



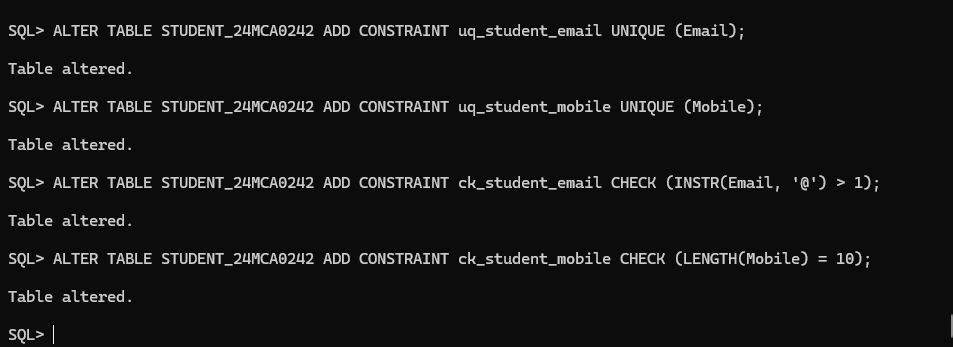
1. Email and mobile column in student table should have same characteristics as those in professor table.

ALTER TABLE STUDENT\_24MCA0242 ADD CONSTRAINT uq\_student\_email UNIQUE (Email);

ALTER TABLE STUDENT\_24MCA0242 ADD CONSTRAINT uq\_student\_mobile UNIQUE (Mobile);

ALTER TABLE STUDENT\_24MCA0242 ADD CONSTRAINT ck\_student\_email CHECK (INSTR(Email, '@') > 1);

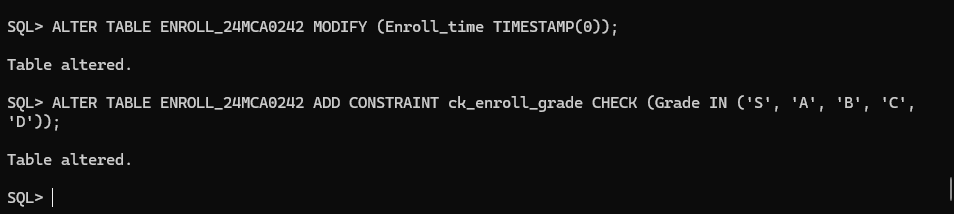
ALTER TABLE STUDENT\_24MCA0242 ADD CONSTRAINT ck\_student\_mobile CHECK (LENGTH(Mobile) = 10);



1. The enroll\_time in the enroll table should be of timestamp data type without fractional parts of seconds. The grade may assume one of the values in {‘S’, ‘A’, ‘B’, ‘C’, ‘D’}

ALTER TABLE ENROLL\_24MCA0242 MODIFY (Enroll\_time TIMESTAMP(0));

ALTER TABLE ENROLL\_24MCA0242 ADD CONSTRAINT ck\_enroll\_grade CHECK (Grade IN ('S', 'A', 'B', 'C', 'D'));

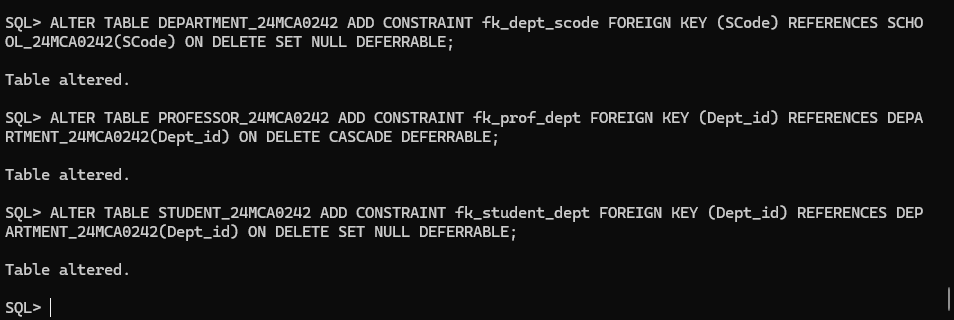


1. Use ‘on delete cascade’ or ‘on delete set null’ clause as requirements. Use deferrable constraint, if required.

ALTER TABLE DEPARTMENT\_24MCA0242 ADD CONSTRAINT fk\_dept\_scode FOREIGN KEY (SCode) REFERENCES SCHOOL\_24MCA0242(SCode) ON DELETE SET NULL DEFERRABLE;

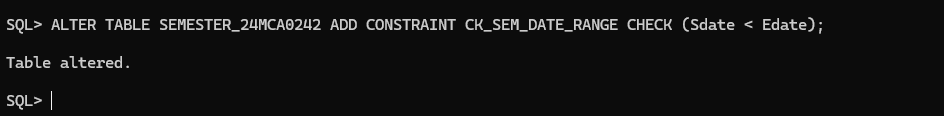
ALTER TABLE PROFESSOR\_24MCA0242 ADD CONSTRAINT fk\_prof\_dept FOREIGN KEY (Dept\_id) REFERENCES DEPARTMENT\_24MCA0242(Dept\_id) ON DELETE CASCADE DEFERRABLE;

ALTER TABLE STUDENT\_24MCA0242 ADD CONSTRAINT fk\_student\_dept FOREIGN KEY (Dept\_id) REFERENCES DEPARTMENT\_24MCA0242(Dept\_id) ON DELETE SET NULL DEFERRABLE;



1. Additional (innovative) integrity constraints, if any, may be specified by you.

ALTER TABLE SEMESTER\_24MCA0242 ADD CONSTRAINT CK\_SEM\_DATE CHECK (Sdate < Edate);

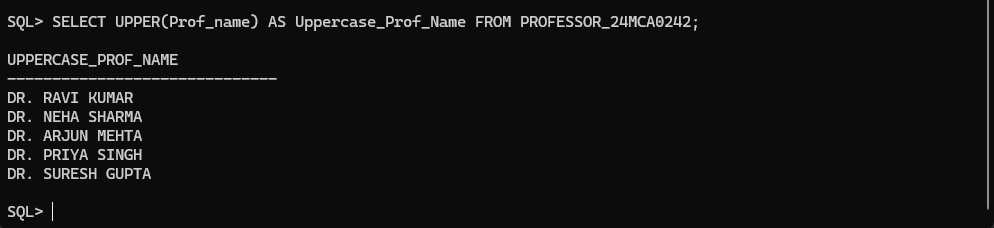


**4. In built functions**

1. Test the string manipulation functions – UPPER, LOWER, INITCAP, LENGTH, LPAD, RPAD, LTRIM, RTRIM and TRIM, using select queries on data present in the tables. Use one query each for demonstration of one function.
2. UPPER

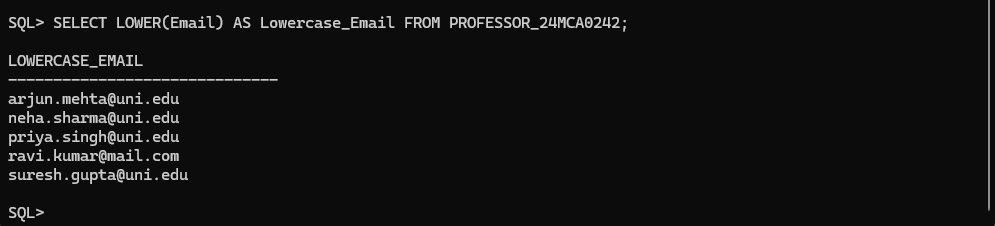
SELECT UPPER(Prof\_name) AS Uppercase\_Prof\_Name

FROM PROFESSOR\_24MCA0242;



1. LOWER

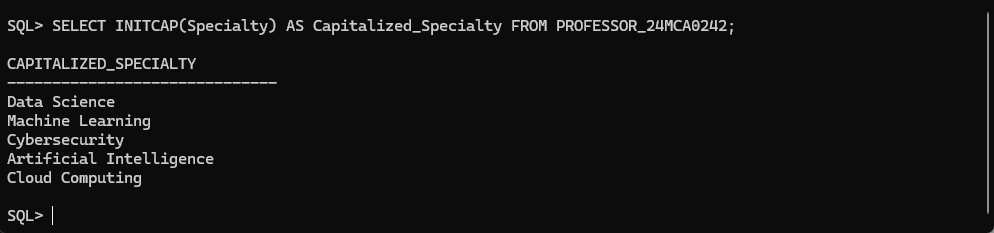
SELECT LOWER(Email) AS Lowercase\_Email FROM PROFESSOR\_24MCA0242;



1. INITCAP

SELECT INITCAP(Speciality) AS Capitalized\_Speciality

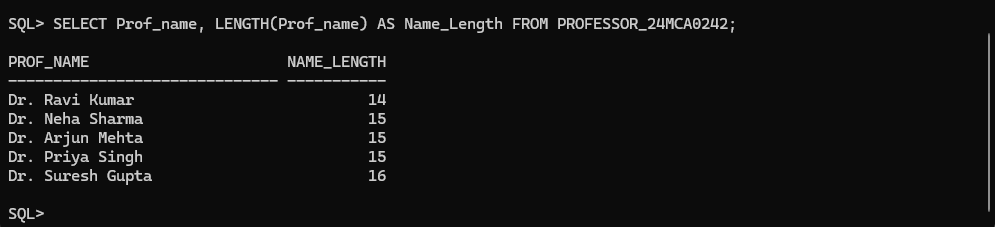
FROM PROFESSOR\_24MCA0242;



1. LENGTH

SELECT Prof\_name, LENGTH(Prof\_name) AS Name\_Length

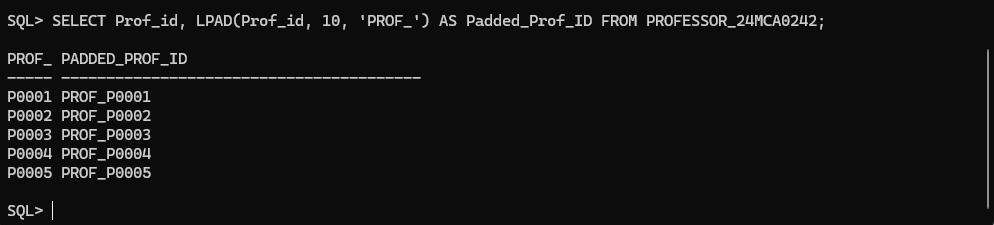
FROM PROFESSOR\_24MCA0242;



1. LPAD

SELECT Prof\_id, LPAD(Prof\_id, 10, 'PROF\_') AS Padded\_Prof\_ID

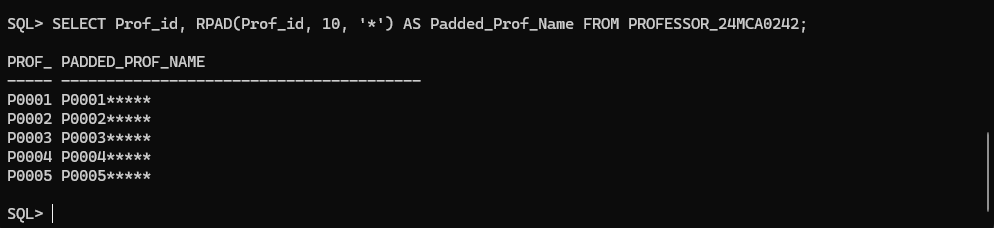
FROM PROFESSOR\_24MCA0242;



1. RPAD

SELECT Prof\_id, RPAD(Prof\_id, 10, '\*') AS Padded\_Prof\_Name

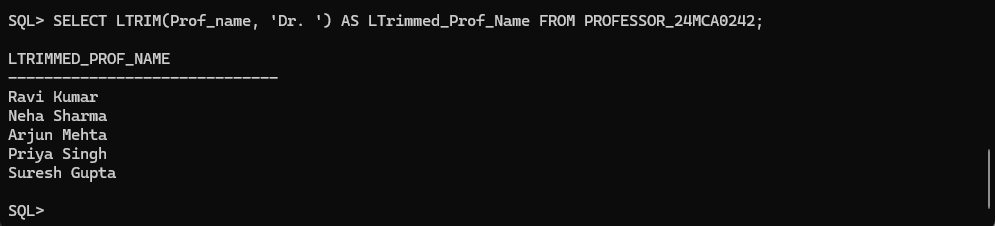
FROM PROFESSOR\_24MCA0242;



1. LTRIM

SELECT LTRIM(Prof\_name, 'Dr. ') AS Trimmed\_Prof\_Name

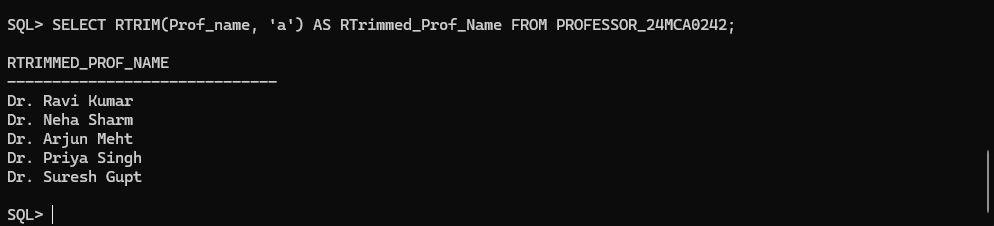
FROM PROFESSOR\_24MCA0242;



1. RTRIM

SELECT RTRIM(Prof\_name, 'a') AS RTrimmed\_Prof\_Name

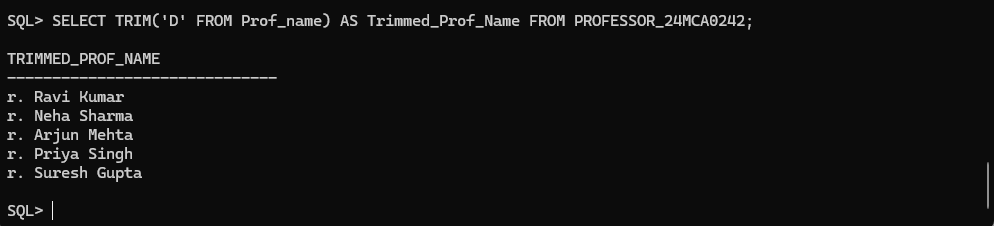
FROM PROFESSOR\_24MCA0242;



1. TRIM

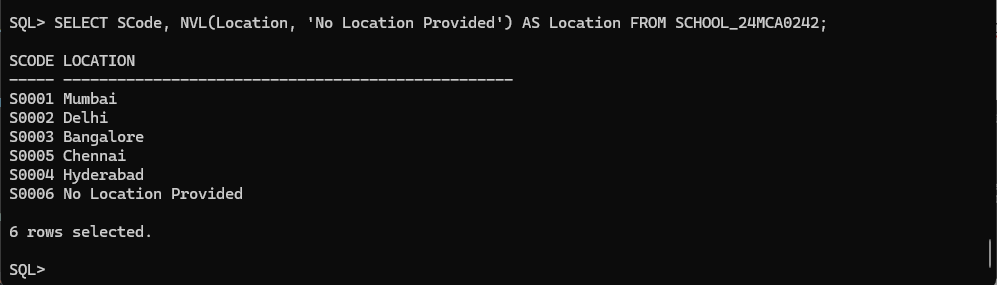
SELECT TRIM( 'D' FROM Prof\_name) AS Trimmed\_Prof\_Name

FROM PROFESSOR\_24MCA0242;



1. Write query to illustrate usage of NVL function and NULLIF function.
2. NVL

SELECT SCode, NVL(Location, 'No Location Provided') AS Location FROM SCHOOL\_24MCA0242;



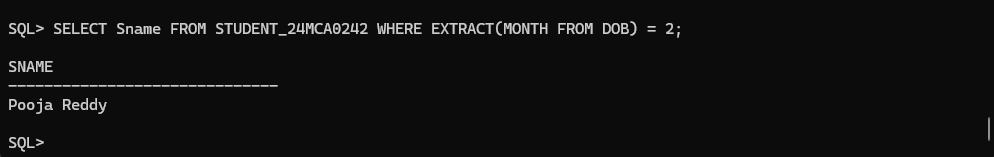
1. NULLIF

SELECT SCode, NULLIF(Location, 'No Location Provided') AS Location FROM SCHOOL\_24MCA0242;



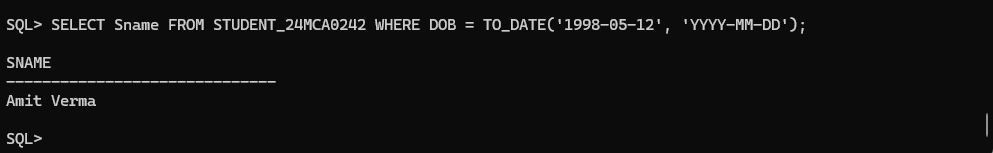
1. Display the name of the students who were born on a specified month.

SELECT Sname FROM STUDENT\_24MCA0242 WHERE EXTRACT(MONTH FROM DOB) = 2;



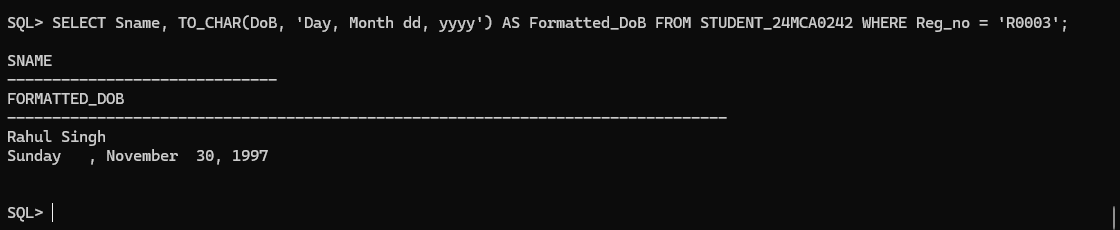
1. Display the name of the students with a specified date of birth.

SELECT Sname FROM STUDENT\_24MCA0242 WHERE DOB = TO\_DATE('1998-05-12', 'YYYY-MM-DD');



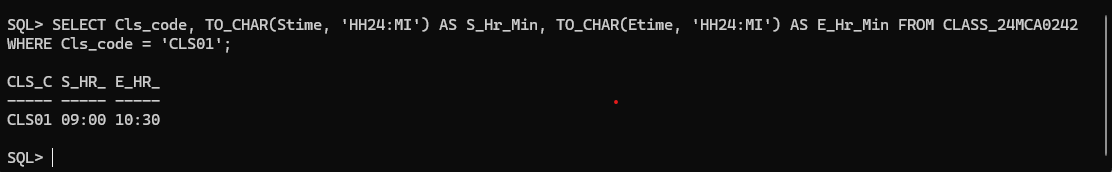
1. Display the date of birth of a specified student in the format ‘Day of week, Month dd, yyyy’.

SELECT Sname, TO\_CHAR(DoB, 'Day, Month dd, yyyy') AS Formatted\_DoB FROM STUDENT\_24MCA0242 WHERE Reg\_no = 'R0003';



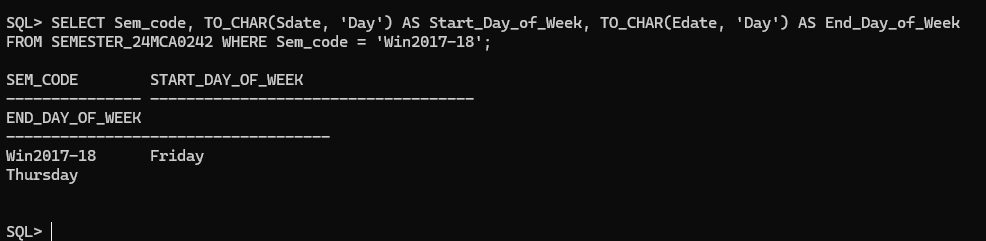
1. Display the hour and minutes of the start time and end time of a specified slot.

SELECT Cls\_code, TO\_CHAR(Stime, 'HH24:MI') AS S\_Hr\_Min, TO\_CHAR(Etime, 'HH24:MI') AS E\_Hr\_Min FROM CLASS\_24MCA0242 WHERE Cls\_code = 'CLS01';



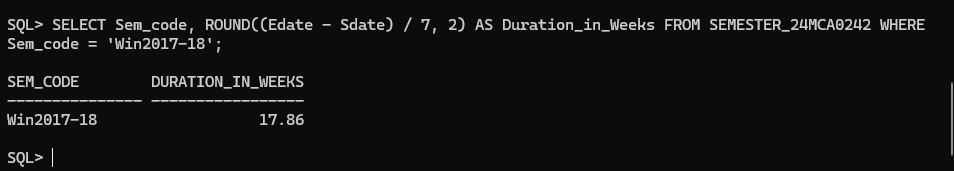
1. Display the day of week of the start date and end date of Winter semester 17–18.

SELECT Sem\_code, TO\_CHAR (Sdate, 'Day') AS Start\_Day\_of\_Week, TO\_CHAR (Edate, 'Day') AS End\_Day\_of\_Week FROM SEMESTER\_24MCA0242 WHERE Sem\_code = 'Win2017';



1. Display the duration of Winter semester 17–18 in terms of number of weeks.

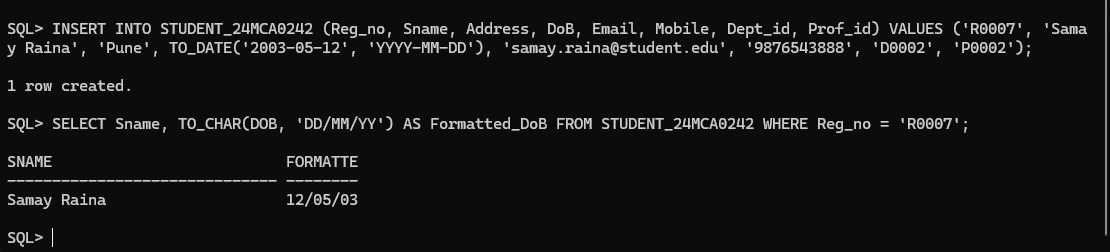
SELECT Sem\_code, ROUND((Edate - Sdate) / 7, 2) AS Duration\_in\_Weeks FROM SEMESTER\_24MCA0242 WHERE Sem\_code = 'Win2017-18';



1. Store date in the format dd/mm/yy for DOB of newly admitted student.

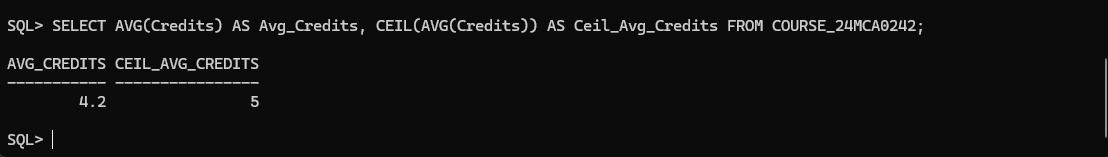
INSERT INTO STUDENT\_24MCA0242 (Reg\_no, Sname, Address, DoB, Email, Mobile, Dept\_id, Prof\_id) VALUES ('R0007', 'Samay Raina', 'Pune', TO\_DATE('2003-05-12', 'YYYY-MM-DD'), 'samay.raina@student.edu', '9876543888', 'D0002', 'P0002');

SELECT Sname, TO\_CHAR(DOB, 'DD/MM/YY') AS Formatted\_DoB FROM STUDENT\_24MCA0242 WHERE Reg\_no = 'R0007';



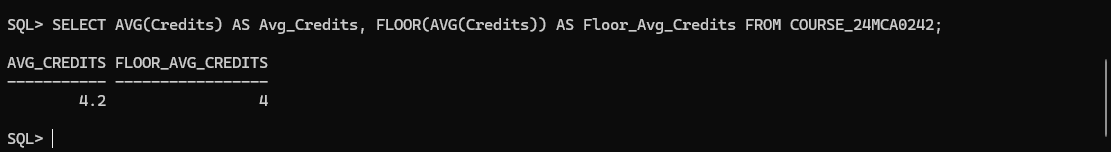
1. Test the numeric functions – CEIL, FLOOR, TRUCATE, MIN, MAX, AVG,COUNT using select queries on data present in the tables. Use one query each for demonstration of one function.
2. CEIL

SELECT AVG(Credits) AS Avg\_Credits, CEIL(AVG(Credits)) AS Ceil\_Avg\_Credits FROM COURSE\_24MCA0242;



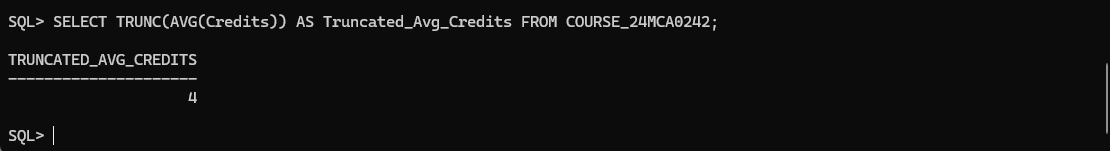
1. FLOOR

SELECT AVG(Credits) AS Avg\_Credits, FLOOR(AVG(Credits)) AS Floor\_Avg\_Credits FROM COURSE\_24MCA0242;



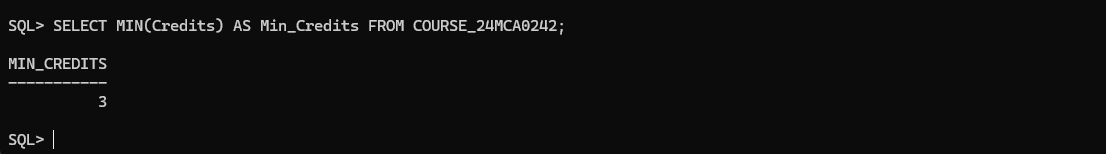
1. TRUNCATE

SELECT TRUNC(AVG(Credits)) AS Truncated\_Avg\_Credits FROM COURSE\_24MCA0242;



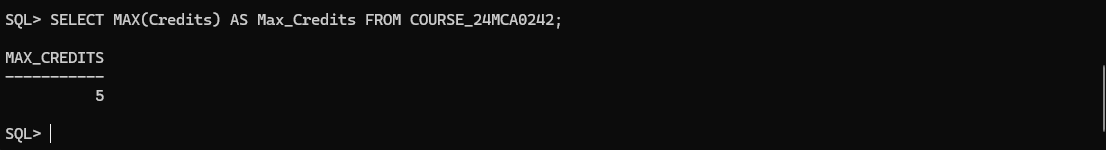
1. MIN

SELECT MIN(Credits) AS Min\_Credits FROM COURSE\_24MCA0242;



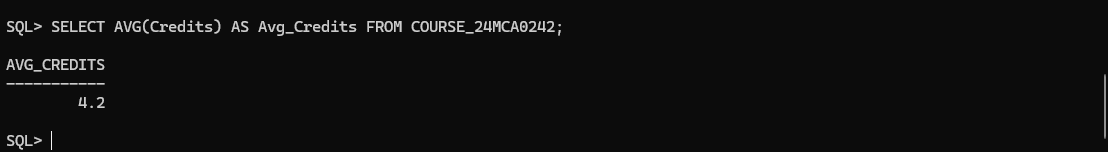
1. MAX

SELECT MAX(Credits) AS Max\_Credits FROM COURSE\_24MCA0242;



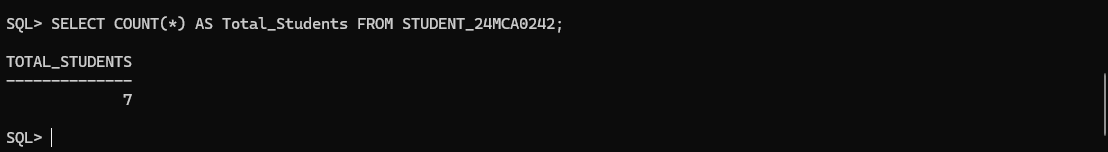
1. AVG

SELECT AVG(Credits) AS Avg\_Credits FROM COURSE\_24MCA0242;



1. COUNT

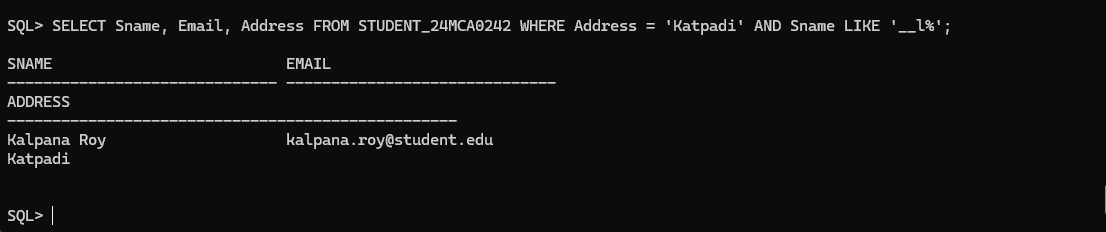
SELECT COUNT(\*) AS Total\_Students FROM STUDENT\_24MCA0242;



**5. Write queries for**

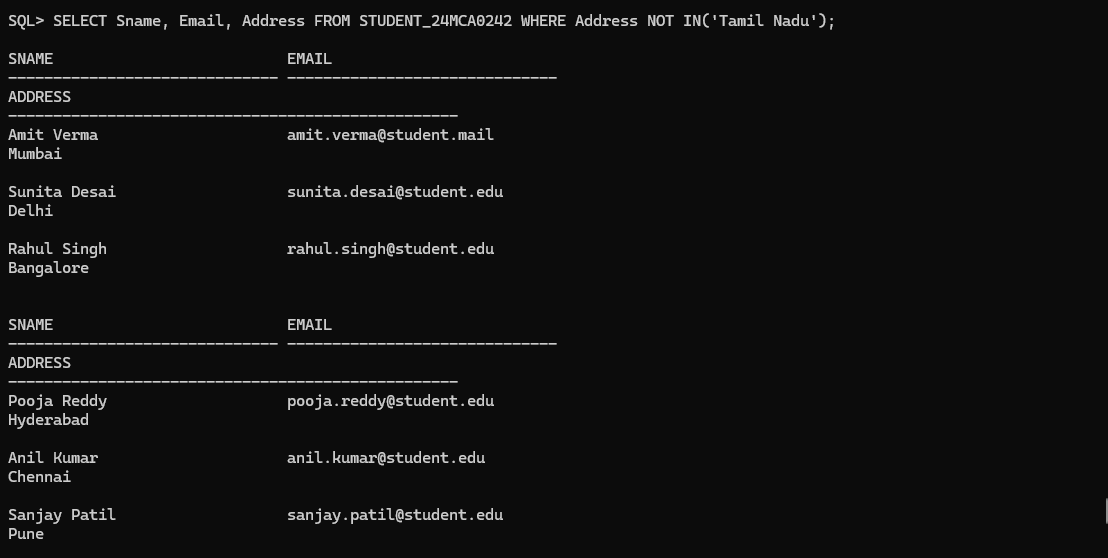
1. Display name, email address and address for those students who live in Katpadi area and whose name has an l as the third character.

SELECT Sname, Email, Address FROM STUDENT\_24MCA0242 WHERE Address = 'Katpadi' AND Sname LIKE '\_\_l%';



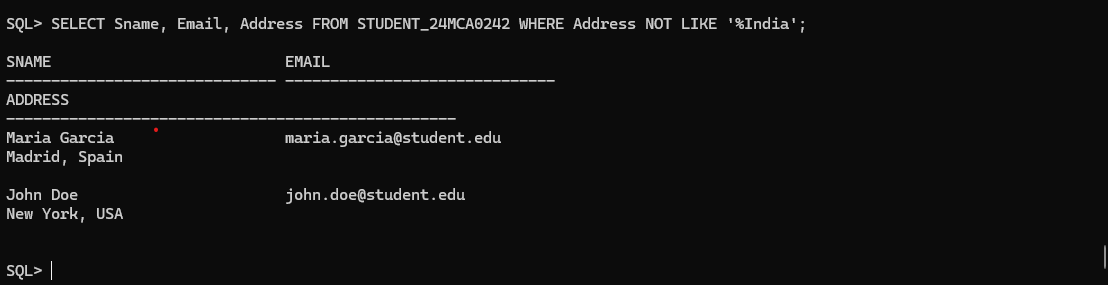
1. Display name, email address and address for those students who are not from Tamil Nadu.

SELECT Sname, Email, Address FROM STUDENT\_24MCA0242 WHERE Address NOT IN('Tamil Nadu');



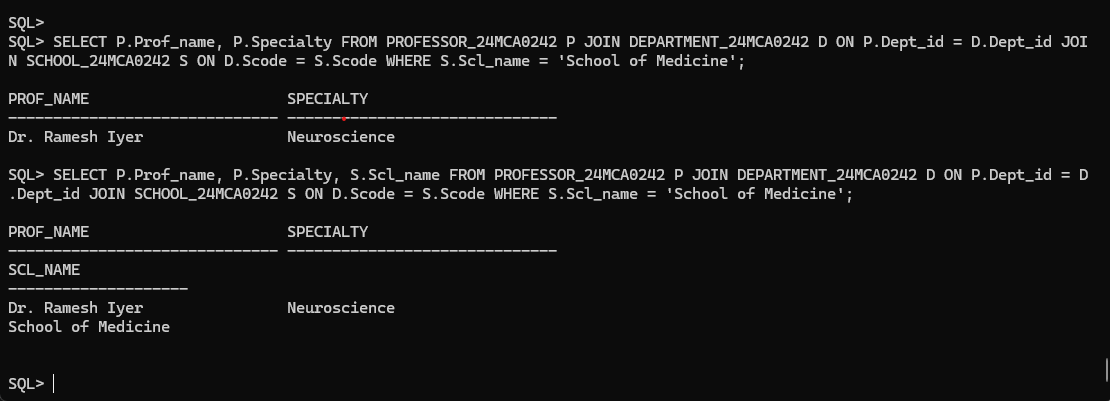
1. Display name, email address and address of foreign students only.

SELECT Sname, Email, Address FROM STUDENT\_24MCA0242 WHERE Address NOT LIKE '%India';



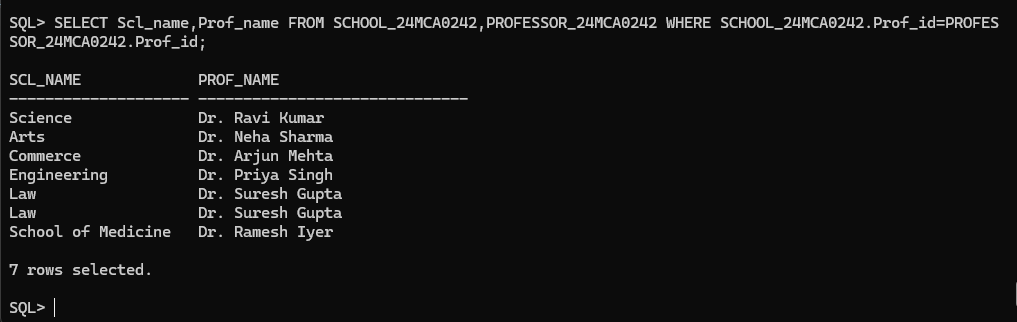
1. List the name of professors along with their specialty who belong to School of Medicine.

SELECT P.Prof\_name, P.Specialty, S.Scl\_name FROM PROFESSOR\_24MCA0242 P JOIN DEPARTMENT\_24MCA0242 D ON P.Dept\_id = D.Dept\_id JOIN SCHOOL\_24MCA0242 S ON D.Scode = S.Scode WHERE S.Scl\_name = 'School of Medicine';



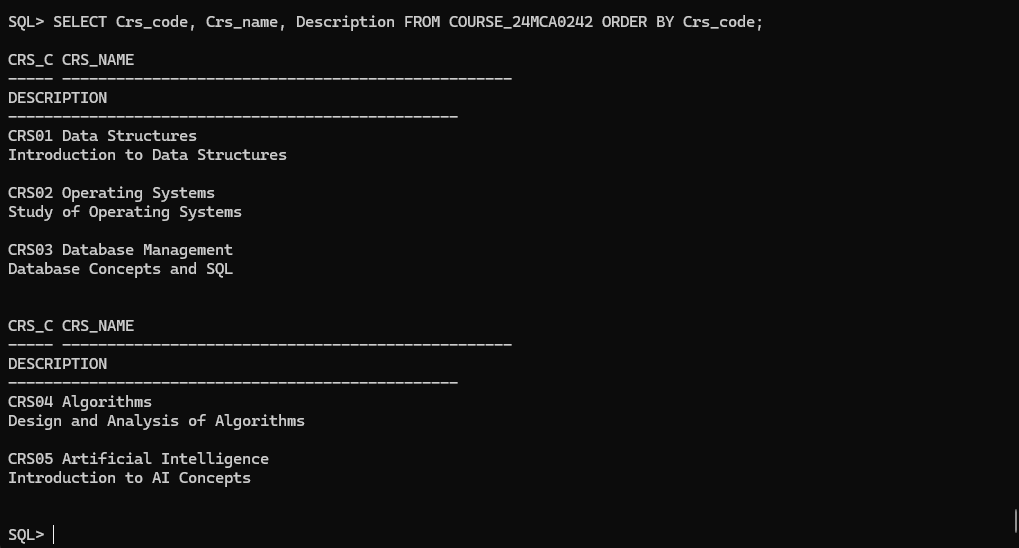
1. Display name of the school and name of professor who chairs the school.

SELECT Scl\_name, Prof\_name FROM SCHOOL\_24MCA0242, PROFESSOR\_24MCA0242 WHERE SCHOOL\_24MCA0242.Prof\_id = PROFESSOR\_24MCA0242.Prof\_id;



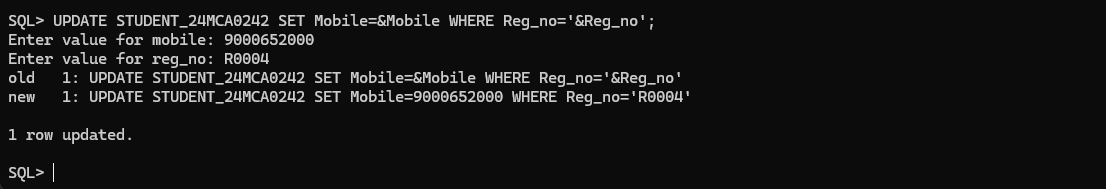
1. List course code, course name and course description in alphabetic order of course code.

SELECT Crs\_code, Crs\_name, Description FROM COURSE\_24MCA0242 ORDER BY Crs\_code;



1. Change the mobile number of a student interactively.

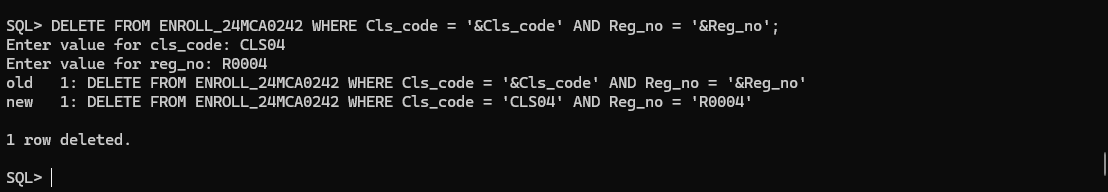
UPDATE STUDENT\_24MCA0242 SET Mobile = &Mobile WHERE Reg\_no = '&Reg\_no';



1. Remove enrollment information of a student from a particular course interactively. How would you recover the data?



DELETE FROM ENROLL\_24MCA0242 WHERE Cls\_code = '&Cls\_code' AND Reg\_no = '&Reg\_no';





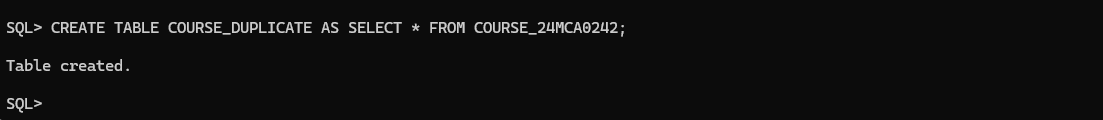
ROLLBACK;





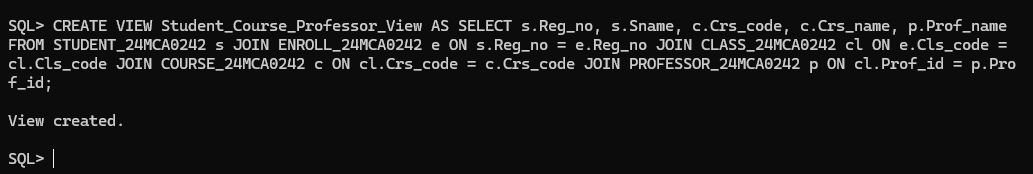
1. Create a duplicate of course table.

CREATE TABLE COURSE\_DUPLICATE AS SELECT \* FROM COURSE\_24MCA0242;

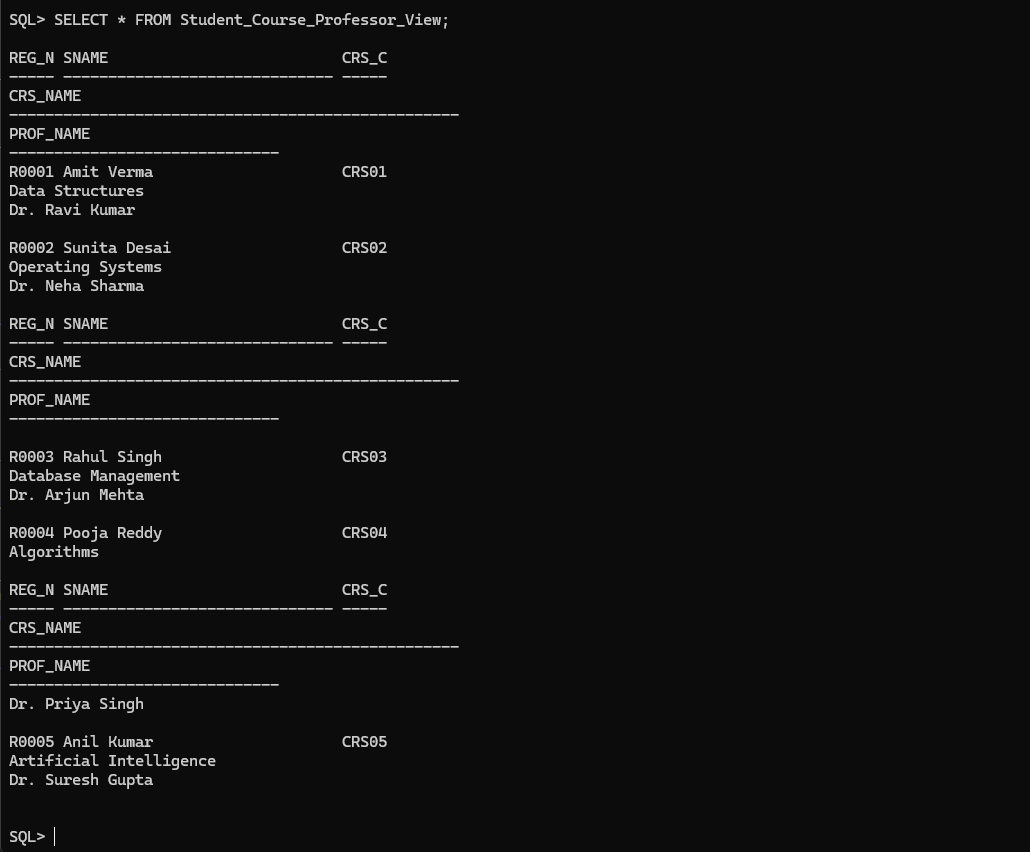


1. Create a view for list of students (Reg\_no, Sname) and the courses they have registered along with name of professors teaching the course.

CREATE VIEW Student\_Course\_Professor\_View AS SELECT s.Reg\_no, s.Sname, c.Crs\_code, c.Crs\_name, p.Prof\_name FROM STUDENT\_24MCA0242 s JOIN ENROLL\_24MCA0242 e ON s.Reg\_no = e.Reg\_no JOIN CLASS\_24MCA0242 cl ON e.Cls\_code = cl.Cls\_code JOIN COURSE\_24MCA0242 c ON cl.Crs\_code = c.Crs\_code JOIN PROFESSOR\_24MCA0242 p ON cl.Prof\_id = p.Prof\_id;



SELECT \* FROM Student\_Course\_Professor\_View;



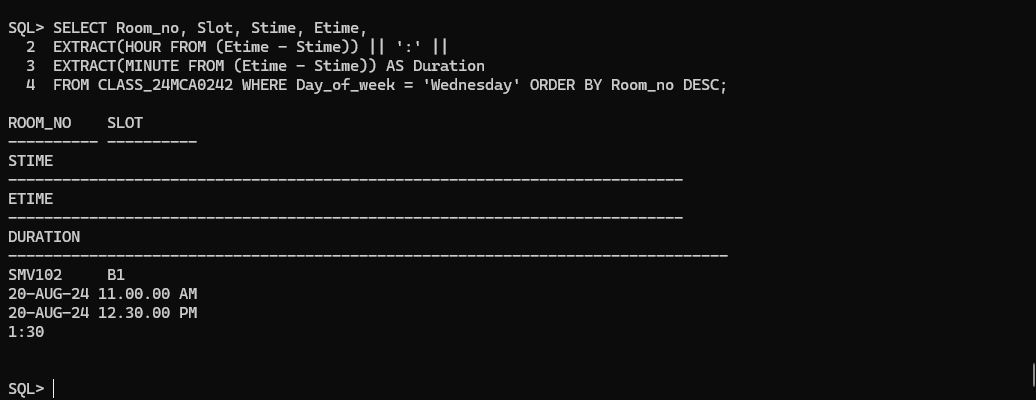
1. List the room number, slot, start time, end time and duration of every class held on Wednesdays in descending order of room number.

SELECT Room\_no, Slot, Stime, Etime,

EXTRACT(HOUR FROM (Etime - Stime)) || ':' ||

EXTRACT(MINUTE FROM (Etime - Stime)) AS Duration

FROM CLASS\_24MCA0242 WHERE Day\_of\_week = 'Wednesday' ORDER BY Room\_no DESC;



1. Display the name and grade of a student in different courses underwent in fall semester 2017 – 18.

SELECT s.Sname, e.Grade, c.Crs\_name FROM STUDENT\_24MCA0242 s

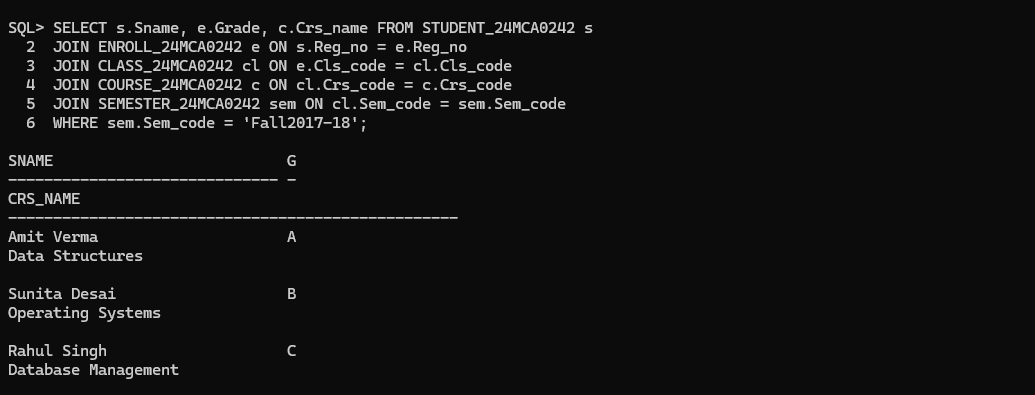
JOIN ENROLL\_24MCA0242 e ON s.Reg\_no = e.Reg\_no

JOIN CLASS\_24MCA0242 cl ON e.Cls\_code = cl.Cls\_code

JOIN COURSE\_24MCA0242 c ON cl.Crs\_code = c.Crs\_code

JOIN SEMESTER\_24MCA0242 sem ON cl.Sem\_code = sem.Sem\_code

WHERE sem.Sem\_code = 'Fall2017-18';



1. Find out name of students who have taken Database Systems course as well as Operating Systems course in fall semester 2016 – 17.

SELECT s.Sname FROM STUDENT\_24MCA0242 s

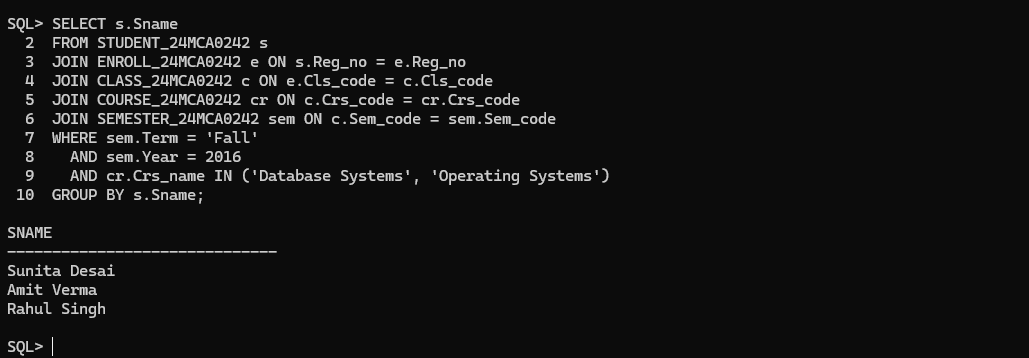
JOIN ENROLL\_24MCA0242 e ON s.Reg\_no = e.Reg\_no

JOIN CLASS\_24MCA0242 c ON e.Cls\_code = c.Cls\_code

JOIN COURSE\_24MCA0242 cr ON c.Crs\_code = cr.Crs\_code

JOIN SEMESTER\_24MCA0242 sem ON c.Sem\_code = sem.Sem\_code

WHERE sem.Term = 'Fall' AND sem.Year = 2016 AND cr.Crs\_name IN ('Database Systems', 'Operating Systems') GROUP BY s.Sname;



1. Find out name of students who have taken Database Systems course but have not taken Operating Systems course in winter semester 2017 – 18.

SELECT s.SNAME FROM STUDENT\_24MCA0242 s

JOIN ENROLL\_24MCA0242 e ON s.REG\_No = e.REG\_No

JOIN CLASS\_24MCA0242 c ON e.CLS\_Code = c.CLS\_Code

JOIN COURSE\_24MCA0242 cr ON c.CRS\_Code = cr.CRS\_Code

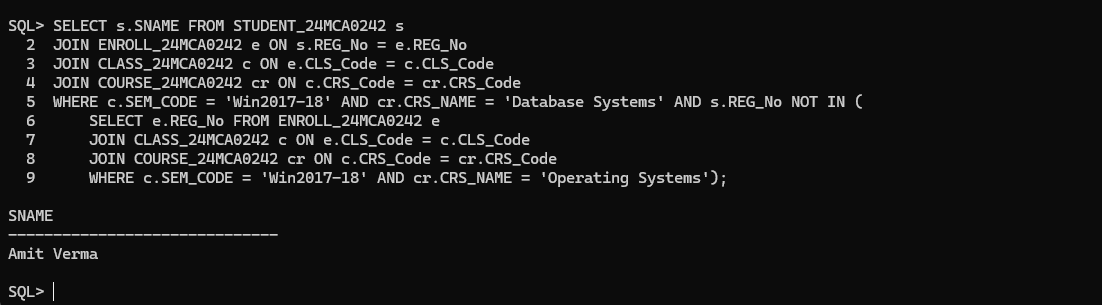
WHERE c.SEM\_CODE = 'Win2017-18' AND cr.CRS\_NAME = 'Database Systems' AND s.REG\_No NOT IN (

SELECT e.REG\_No FROM ENROLL\_24MCA0242 e

JOIN CLASS\_24MCA0242 c ON e.CLS\_Code = c.CLS\_Code

JOIN COURSE\_24MCA0242 cr ON c.CRS\_Code = cr.CRS\_Code

WHERE c.SEM\_CODE = 'Win2017-18' AND cr.CRS\_NAME = 'Operating Systems');



1. List the registration number and name of the students who have registered for maximum number of credits in Winter 17-18 semester.

SELECT s.REG\_No, s.SNAME, SUM(cr.Credits) AS TotalCredits

FROM STUDENT\_24MCA0242 s

JOIN ENROLL\_24MCA0242 e ON s.REG\_No = e.REG\_No

JOIN CLASS\_24MCA0242 c ON e.CLS\_Code = c.CLS\_Code

JOIN COURSE\_24MCA0242 cr ON c.CRS\_Code = cr.CRS\_Code

WHERE c.SEM\_CODE = 'Win2017-18'

GROUP BY s.REG\_No, s.SNAME

HAVING SUM(cr.Credits) = (

SELECT MAX(TotalCredits)

FROM (

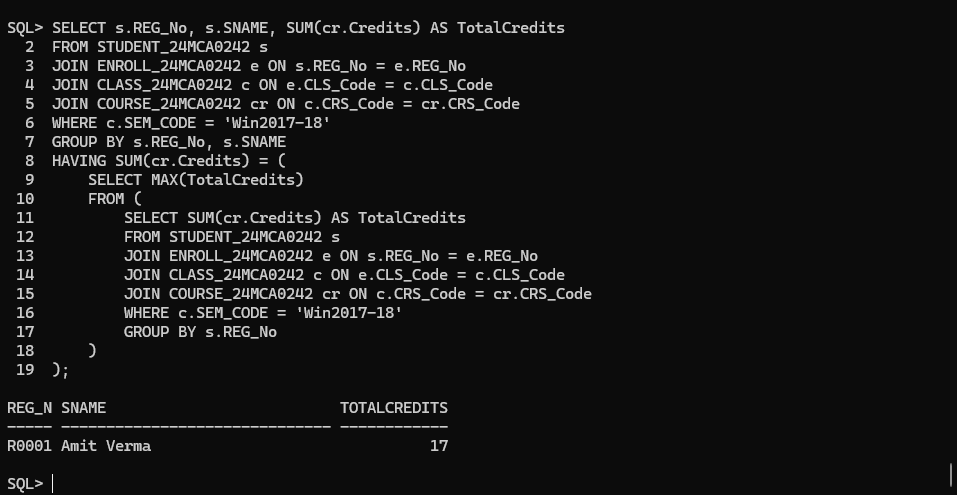
SELECT SUM(cr.Credits) AS TotalCredits FROM STUDENT\_24MCA0242 s

JOIN ENROLL\_24MCA0242 e ON s.REG\_No = e.REG\_No

JOIN CLASS\_24MCA0242 c ON e.CLS\_Code = c.CLS\_Code

JOIN COURSE\_24MCA0242 cr ON c.CRS\_Code = cr.CRS\_Code

WHERE c.SEM\_CODE = 'Win2017-18' GROUP BY s.REG\_No ));



1. List the name of the course and the number of students registered in each slot for course under different faculty members.

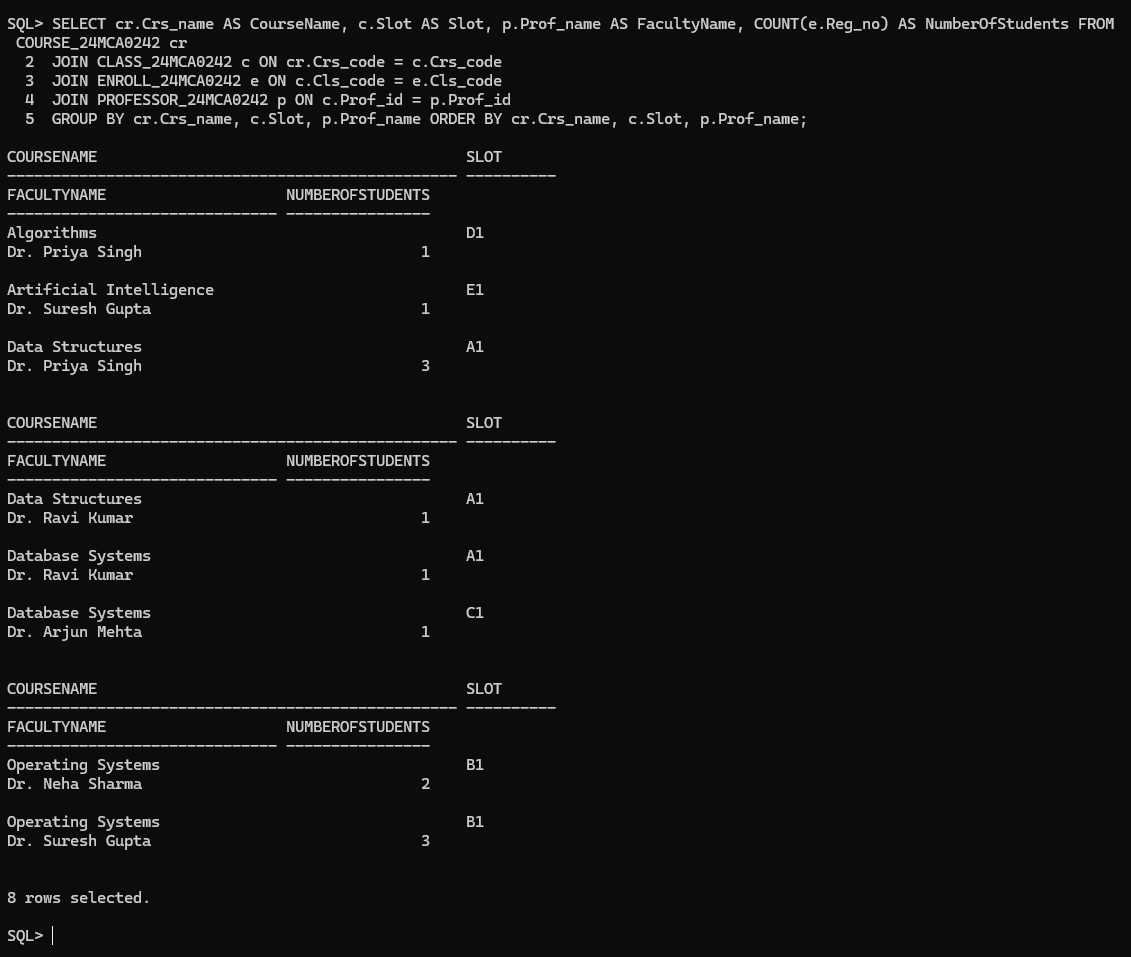
SELECT cr.Crs\_name AS CourseName, c.Slot AS Slot, p.Prof\_name AS FacultyName, COUNT(e.Reg\_no) AS NumberOfStudents FROM COURSE\_24MCA0242 cr

JOIN CLASS\_24MCA0242 c ON cr.Crs\_code = c.Crs\_code

JOIN ENROLL\_24MCA0242 e ON c.Cls\_code = e.Cls\_code

JOIN PROFESSOR\_24MCA0242 p ON c.Prof\_id = p.Prof\_id

GROUP BY cr.Crs\_name, c.Slot, p.Prof\_name ORDER BY cr.Crs\_name, c.Slot, p.Prof\_name;



1. Find out the name of the students who have registered in all the courses being taught by Prof. O’Brien in Winter 17-18.

SELECT s.Sname FROM STUDENT\_24MCA0242 s

JOIN ENROLL\_24MCA0242 e ON s.REG\_No = e.REG\_No

JOIN CLASS\_24MCA0242 c ON e.CLS\_Code = c.CLS\_Code

WHERE c.Crs\_code IN (

SELECT c2.Crs\_code FROM CLASS\_24MCA0242 c2

JOIN PROFESSOR\_24MCA0242 p ON c2.Prof\_id = p.Prof\_id

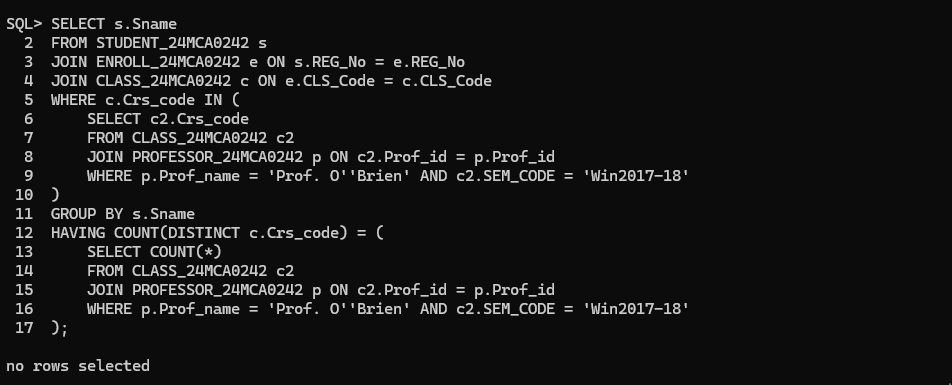
WHERE p.Prof\_name = 'Prof. O''Brien' AND c2.SEM\_CODE = 'Win2017-18')

GROUP BY s.Sname HAVING COUNT(DISTINCT c.Crs\_code) = (

SELECT COUNT(\*) FROM CLASS\_24MCA0242 c2

JOIN PROFESSOR\_24MCA0242 p ON c2.Prof\_id = p.Prof\_id

WHERE p.Prof\_name = 'Prof. O''Brien' AND c2.SEM\_CODE = 'Win2017-18');



1. List the registration number of the students who registered in Database Systems course on November 17, 2017.

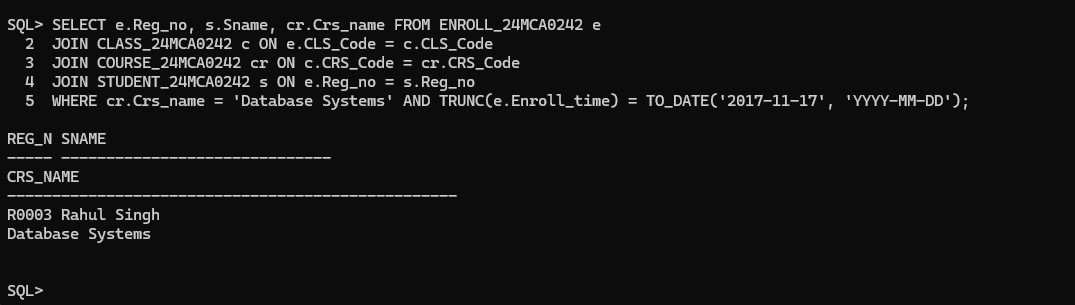
SELECT e.Reg\_no, s.Sname, cr.Crs\_name FROM ENROLL\_24MCA0242 e

JOIN CLASS\_24MCA0242 c ON e.CLS\_Code = c.CLS\_Code

JOIN COURSE\_24MCA0242 cr ON c.CRS\_Code = cr.CRS\_Code

JOIN STUDENT\_24MCA0242 s ON e.Reg\_no = s.Reg\_no

WHERE cr.Crs\_name = 'Database Systems' AND TRUNC(e.Enroll\_time) = TO\_DATE('2017-11-17', 'YYYY-MM-DD');



1. Write a query to display the grade of a student given his/her registration number and the course name for Fall semester 17–18.

SELECT e.Reg\_no, s.Sname, cr.Crs\_name, e.Grade FROM ENROLL\_24MCA0242 e

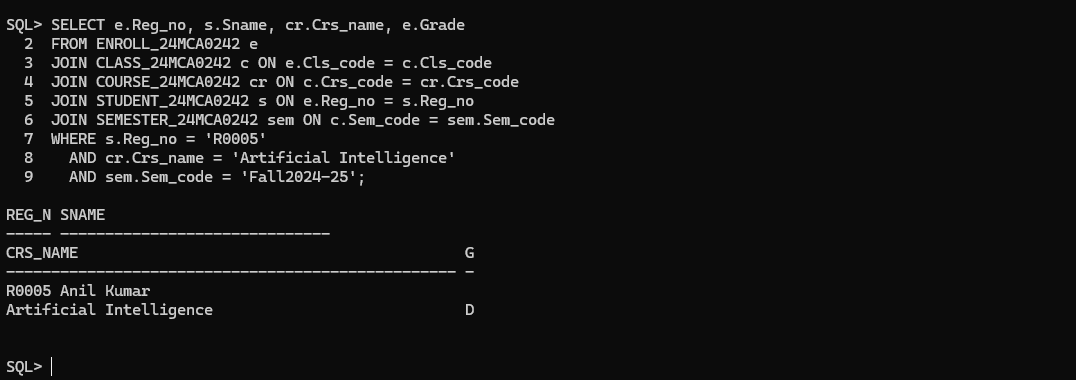
JOIN CLASS\_24MCA0242 c ON e.Cls\_code = c.Cls\_code

JOIN COURSE\_24MCA0242 cr ON c.Crs\_code = cr.Crs\_code

JOIN STUDENT\_24MCA0242 s ON e.Reg\_no = s.Reg\_no

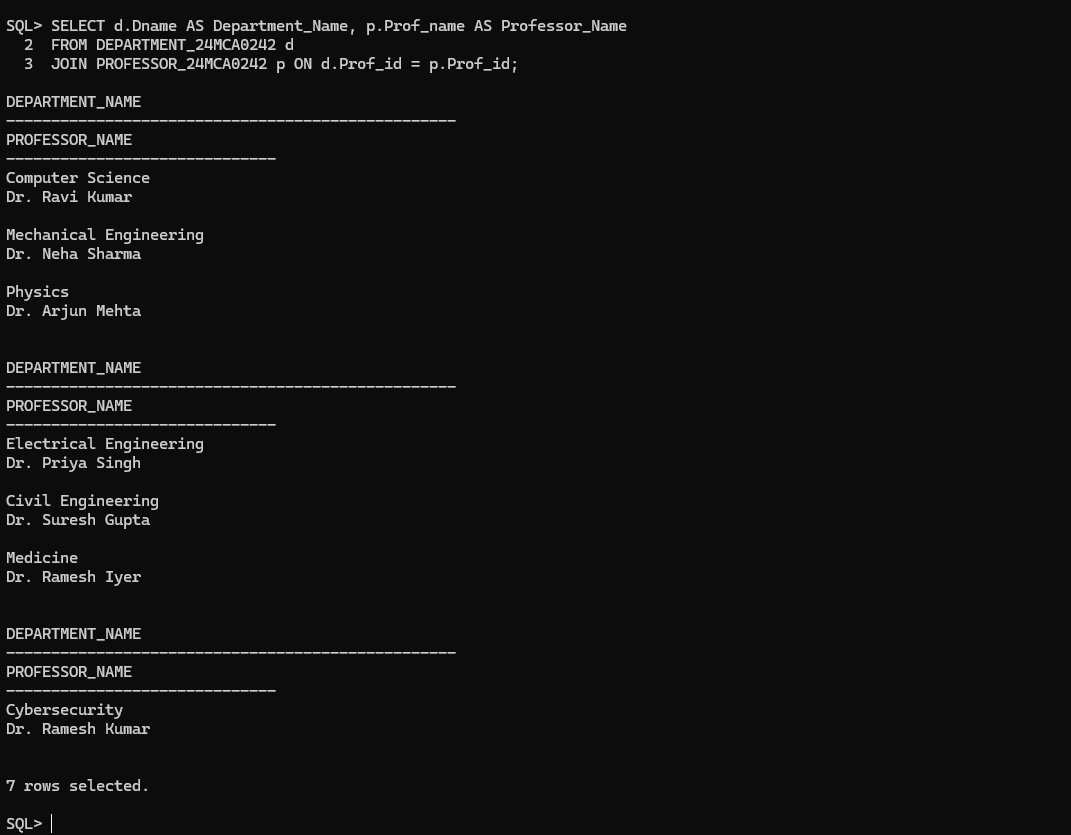
JOIN SEMESTER\_24MCA0242 sem ON c.Sem\_code = sem.Sem\_code

WHERE s.Reg\_no = 'R0005' AND cr.Crs\_name = 'Artificial Intelligence' AND sem.Sem\_code = 'Fall2024-25';



1. List the name of departments and the name professors who is in charge of the department.

SELECT d.Dname AS Department\_Name, p.Prof\_name AS Professor\_Name FROM DEPARTMENT\_24MCA0242 d JOIN PROFESSOR\_24MCA0242 p ON d.Prof\_id = p.Prof\_id;

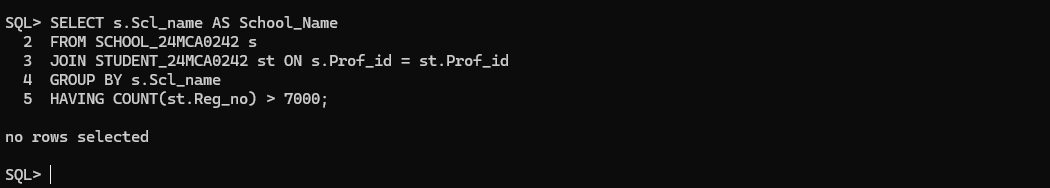


1. List the name of schools with students’ strength higher than 7000.

SELECT s.Scl\_name AS School\_Name FROM SCHOOL\_24MCA0242 s

JOIN STUDENT\_24MCA0242 st ON s.Prof\_id = st.Prof\_id

GROUP BY s.Scl\_name HAVING COUNT(st.Reg\_no) > 7000;



1. List the name of the department(s) under school of medicine with student strength higher than the average students of all the departments in the school.

SELECT sc.Scl\_name, d.Dname FROM DEPARTMENT\_24MCA0242 d

JOIN SCHOOL\_24MCA0242 sc ON d.SCode = sc.SCode

JOIN STUDENT\_24MCA0242 s ON d.Dept\_id = s.Dept\_id

WHERE sc.Scl\_name = 'School of Medicine' GROUP BY sc.Scl\_name, d.Dname

HAVING COUNT(s.Reg\_no) > (

SELECT AVG(Student\_Count) FROM (

SELECT d.Dept\_id, COUNT(s.Reg\_no) AS Student\_Count

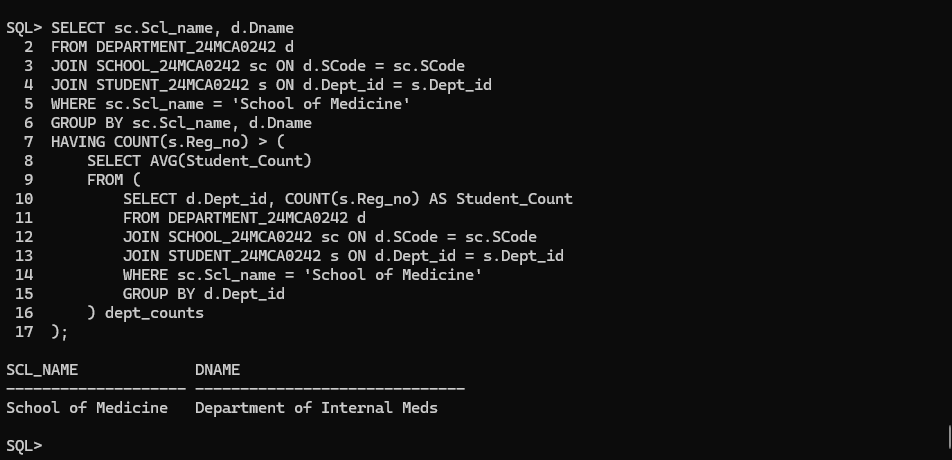
FROM DEPARTMENT\_24MCA0242 d

JOIN SCHOOL\_24MCA0242 sc ON d.SCode = sc.SCode

JOIN STUDENT\_24MCA0242 s ON d.Dept\_id = s.Dept\_id

WHERE sc.Scl\_name = 'School of Medicine' GROUP BY d.Dept\_id

) dept\_counts );



1. Given the registration number of a student, display the total credits registered by him/her in Winter 17–18.

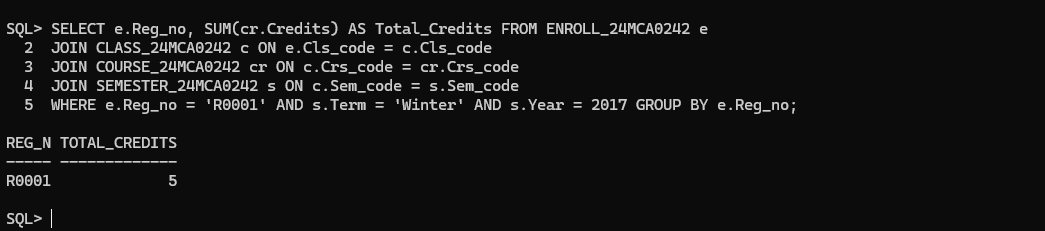
SELECT e.Reg\_no, SUM(cr.Credits) AS Total\_Credits FROM ENROLL\_24MCA0242 e

JOIN CLASS\_24MCA0242 c ON e.Cls\_code = c.Cls\_code

JOIN COURSE\_24MCA0242 cr ON c.Crs\_code = cr.Crs\_code

JOIN SEMESTER\_24MCA0242 s ON c.Sem\_code = s.Sem\_code

WHERE e.Reg\_no = 'R0001' AND s.Term = 'Winter' AND s.Year = 2017 GROUP BY e.Reg\_no;



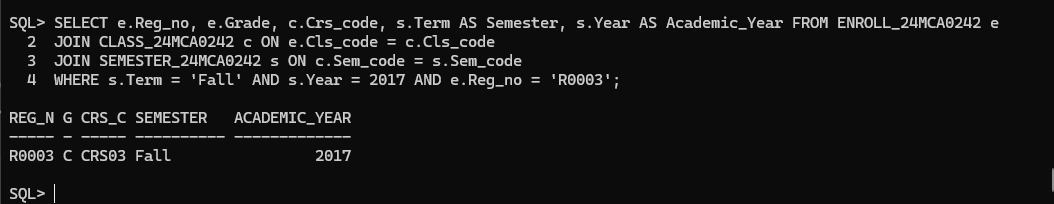
1. Given the registration number of a student, display her/his grade in the course she/he registered in Fall 17–18.

SELECT e.Reg\_no, e.Grade, c.Crs\_code, s.Term AS Semester, s.Year AS Academic\_Year FROM ENROLL\_24MCA0242 e

JOIN CLASS\_24MCA0242 c ON e.Cls\_code = c.Cls\_code

JOIN SEMESTER\_24MCA0242 s ON c.Sem\_code = s.Sem\_code

WHERE s.Term = 'Fall' AND s.Year = 2017 AND e.Reg\_no = 'R0003';



1. Display the name of the courses that are not being offered in Winter 17–18.

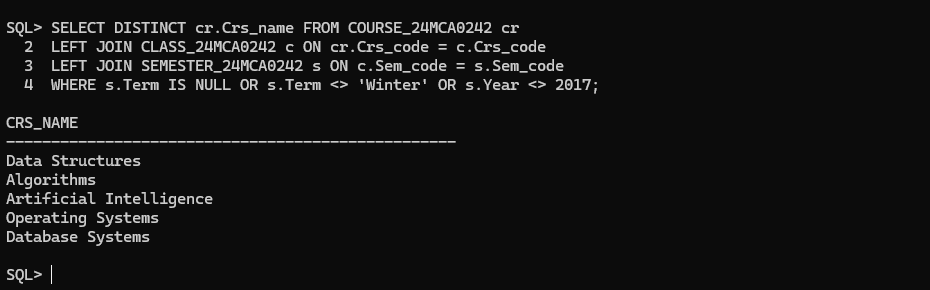
SELECT DISTINCT cr.Crs\_name

FROM COURSE\_24MCA0242 cr

LEFT JOIN CLASS\_24MCA0242 c ON cr.Crs\_code = c.Crs\_code

LEFT JOIN SEMESTER\_24MCA0242 s ON c.Sem\_code = s.Sem\_code

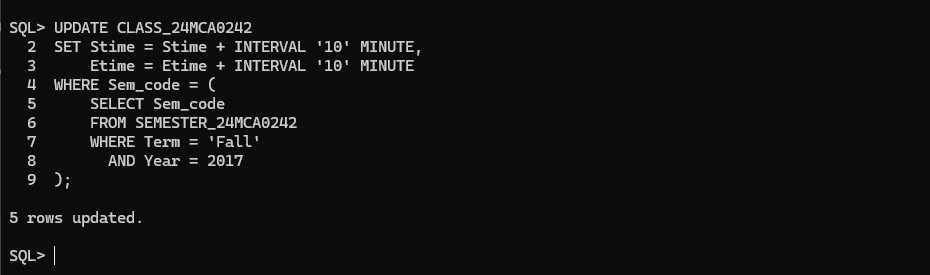
WHERE s.Term IS NULL OR s.Term <> 'Winter' OR s.Year <> 2017;



1. Write necessary SQL statement to advance the start time and end time of every class by ten minutes in Fall 17-18.

UPDATE CLASS\_24MCA0242 SET Stime = Stime + INTERVAL '10' MINUTE, Etime = Etime + INTERVAL '10' MINUTE

WHERE Sem\_code = (SELECT Sem\_code FROM SEMESTER\_24MCA0242 WHERE Term = 'Fall' AND Year = 2017);



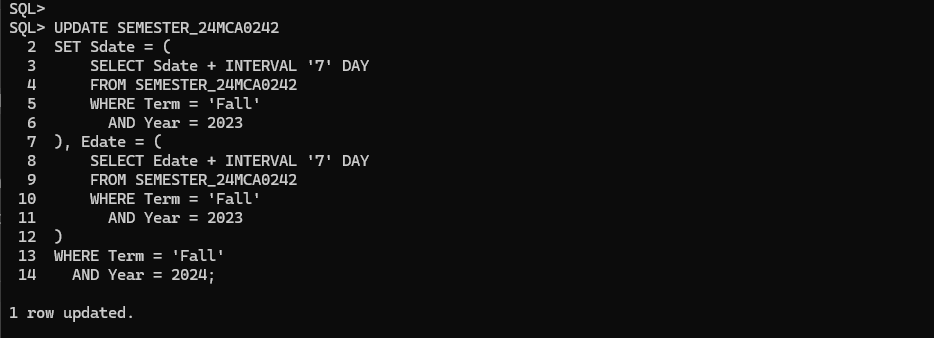
1. Write necessary SQL statement to advance the start date and end date of Fall 24–25 semester by one week with respect to Fall semester of 23 – 24.

UPDATE SEMESTER\_24MCA0242

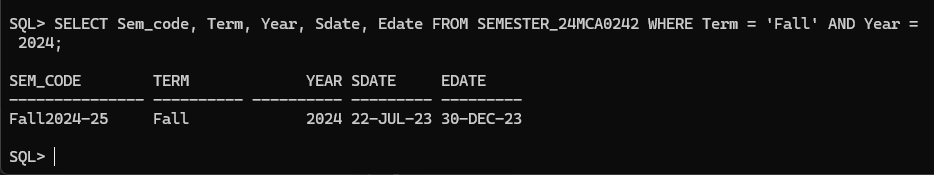
SET Sdate = (SELECT Sdate + INTERVAL '7' DAY FROM SEMESTER\_24MCA0242 WHERE Term = 'Fall' AND Year = 2023),

Edate = (SELECT Edate + INTERVAL '7' DAY FROM SEMESTER\_24MCA0242 WHERE Term = 'Fall' AND Year = 2023)

WHERE Term = 'Fall' AND Year = 2024;



SELECT Sem\_code, Term, Year, Sdate, Edate FROM SEMESTER\_24MCA0242 WHERE Term = 'Fall' AND Year = 2024;



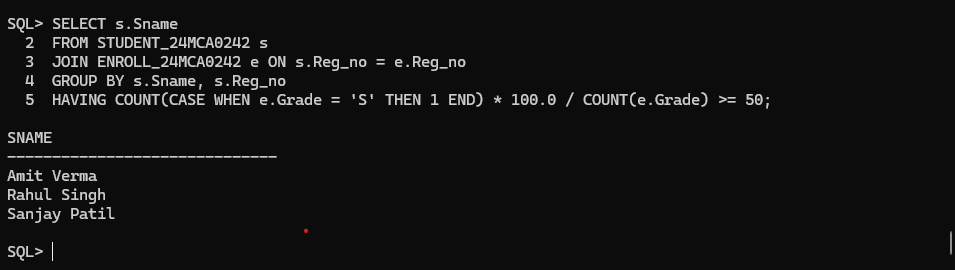
1. Find out the name list of students who had secured ‘S’ grade in at least 50% of the courses cleared by her/him.

SELECT s.Sname FROM STUDENT\_24MCA0242 s

JOIN ENROLL\_24MCA0242 e ON s.Reg\_no = e.Reg\_no

GROUP BY s.Sname, s.Reg\_no

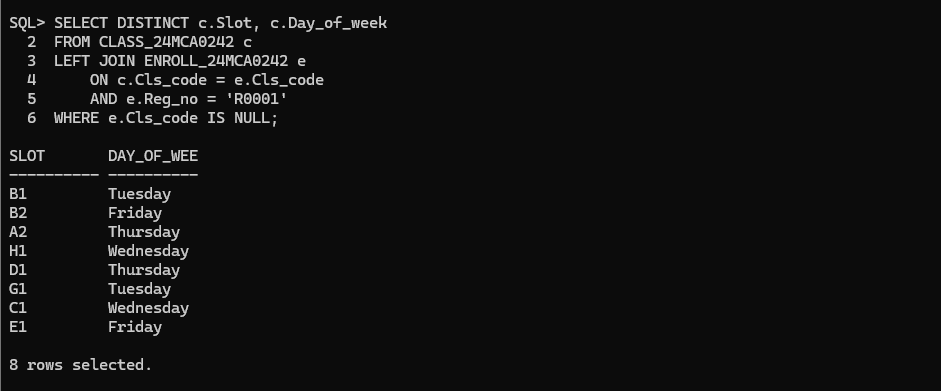
HAVING COUNT(CASE WHEN e.Grade = 'S' THEN 1 END) \* 100.0 / COUNT(e.Grade) >= 50;



1. Given the registration number of a student, find out his/her free slots.

SELECT DISTINCT c.Slot, c.Day\_of\_week FROM CLASS\_24MCA0242 c

LEFT JOIN ENROLL\_24MCA0242 e ON c.Cls\_code = e.Cls\_code AND e.Reg\_no = 'R0001' WHERE e.Cls\_code IS NULL;



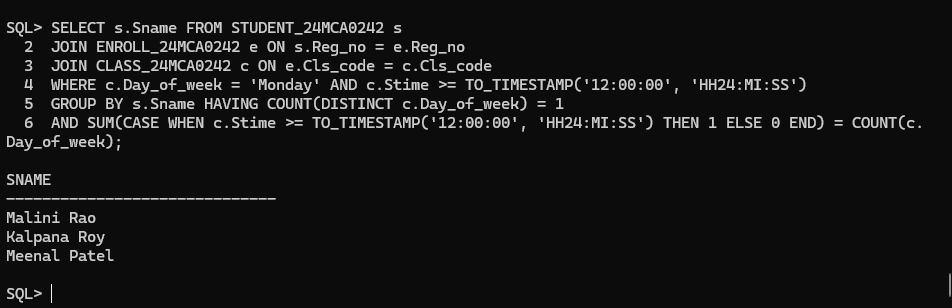
1. Find out the name list of students who have classes in the afternoon session only a specific day of the week.

SELECT s.Sname FROM STUDENT\_24MCA0242 s

JOIN ENROLL\_24MCA0242 e ON s.Reg\_no = e.Reg\_no

JOIN CLASS\_24MCA0242 c ON e.Cls\_code = c.Cls\_code

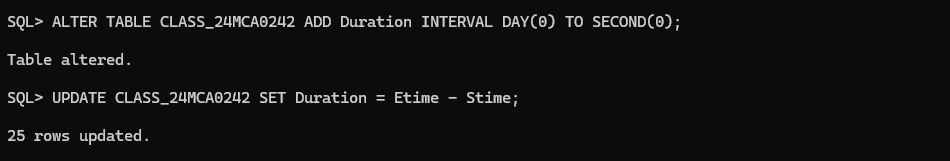
WHERE c.Day\_of\_week = 'Monday' AND c.Stime >= TO\_TIMESTAMP('12:00:00', 'HH24:MI:SS') GROUP BY s.Sname HAVING COUNT(DISTINCT c.Day\_of\_week) = 1 AND SUM(CASE WHEN c.Stime >= TO\_TIMESTAMP('12:00:00', 'HH24:MI:SS') THEN 1 ELSE 0 END) = COUNT(c.Day\_of\_week);



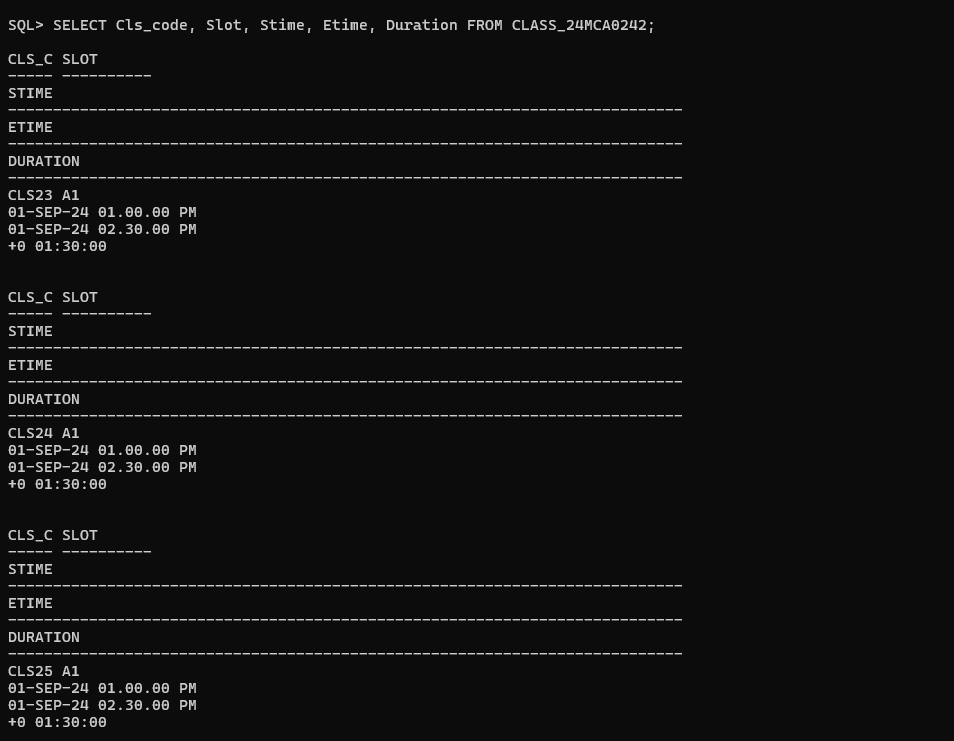
1. Add a column named ‘Duration’ (to indicate duration of a class) with appropriate data type to the CLASS table and populate the column from values of start time and end time columns.

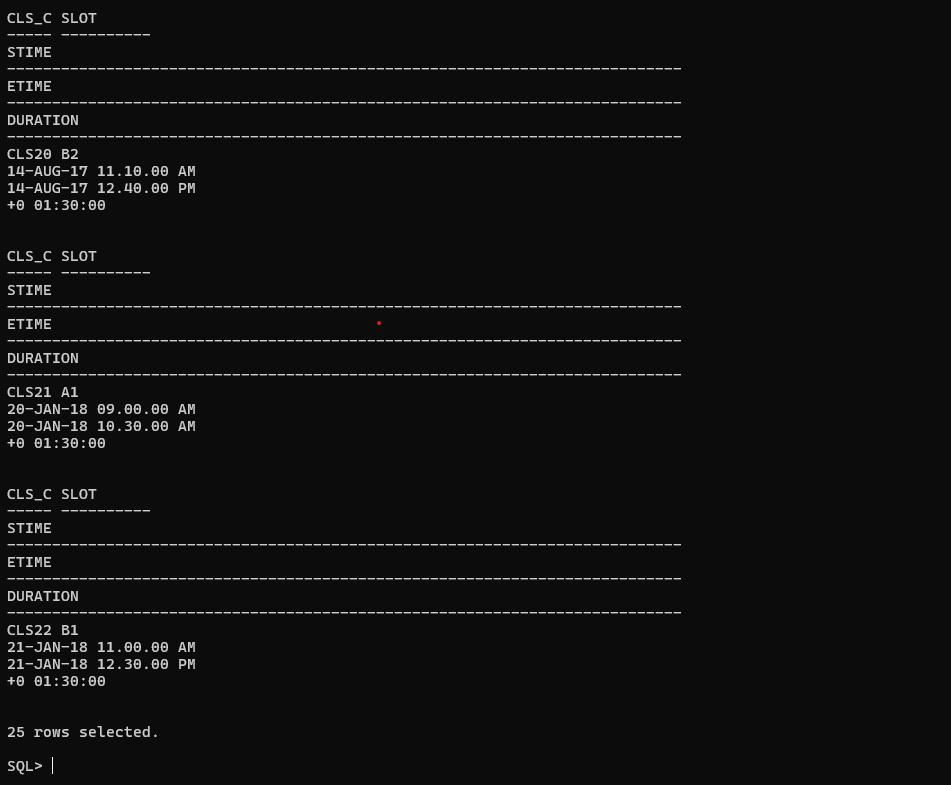
ALTER TABLE CLASS\_24MCA0242 ADD Duration INTERVAL DAY(0) TO SECOND(0);

UPDATE CLASS\_24MCA0242 SET Duration = Etime - Stime;



SELECT Cls\_code, Slot, Stime, Etime, Duration FROM CLASS\_24MCA0242;



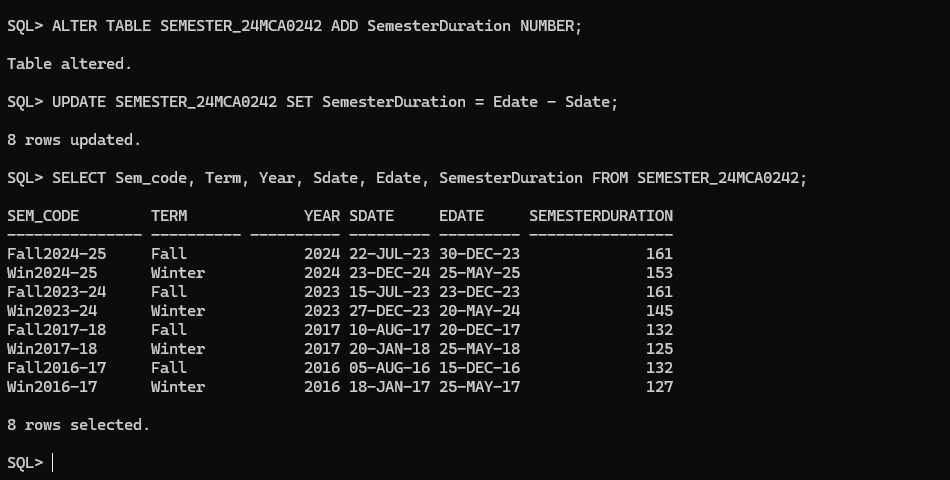


1. Add a column named ‘SemesterDuration’ (indicating duration of a semester) with appropriate data type to the SEMESTER table and populate the column from values of start date and end date columns.

ALTER TABLE SEMESTER\_24MCA0242 ADD SemesterDuration NUMBER;

UPDATE SEMESTER\_24MCA0242 SET SemesterDuration = Edate - Sdate;

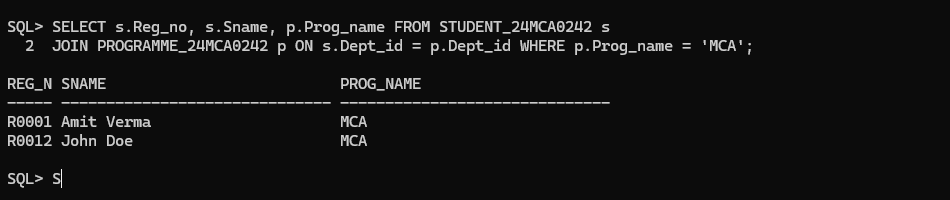
SELECT Sem\_code, Term, Year, Sdate, Edate, SemesterDuration FROM SEMESTER\_24MCA0242;



1. Find out the list of students who are undergoing MCA program.

SELECT s.Reg\_no, s.Sname, p.Prog\_name FROM STUDENT\_24MCA0242 s

JOIN PROGRAMME\_24MCA0242 p ON s.Dept\_id = p.Dept\_id WHERE p.Prog\_name = 'MCA';

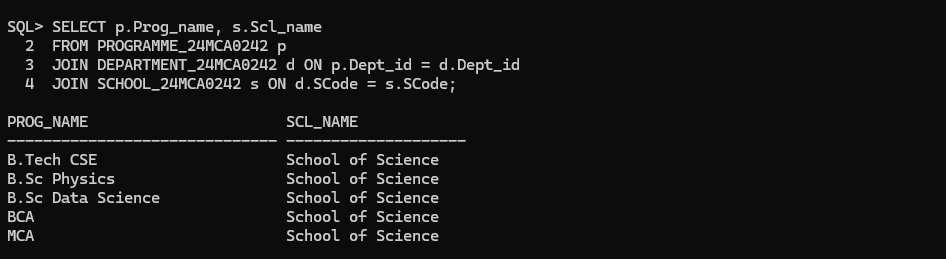


1. Display the name of programs and the name of school offering the program.

SELECT p.Prog\_name, s.Scl\_name FROM PROGRAMME\_24MCA0242 p

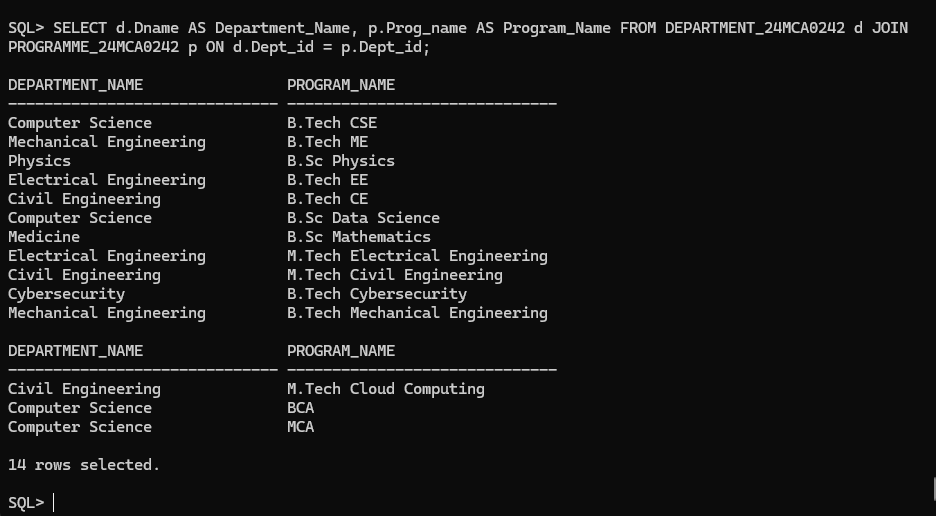
JOIN DEPARTMENT\_24MCA0242 d ON p.Dept\_id = d.Dept\_id

JOIN SCHOOL\_24MCA0242 s ON d.SCode = s.SCode;



1. Display the name of the departments and the name of the program controlled by the department.

SELECT d.Dname AS Department\_Name, p.Prog\_name AS Program\_Name FROM DEPARTMENT\_24MCA0242 d JOIN PROGRAMME\_24MCA0242 p ON d.Dept\_id = p.Dept\_id;



1. Find the school which has highest school strength (i.e number of students)

SELECT Scl\_name FROM (

SELECT s.Scl\_name, COUNT(st.Reg\_no) AS student\_count

FROM SCHOOL\_24MCA0242 s

JOIN DEPARTMENT\_24MCA0242 d ON s.SCode = d.SCode

JOIN STUDENT\_24MCA0242 st ON d.Dept\_id = st.Dept\_id

GROUP BY s.Scl\_name

ORDER BY student\_count DESC)

WHERE ROWNUM = 1;

