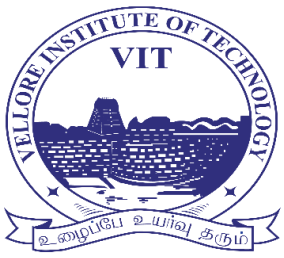


VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

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)

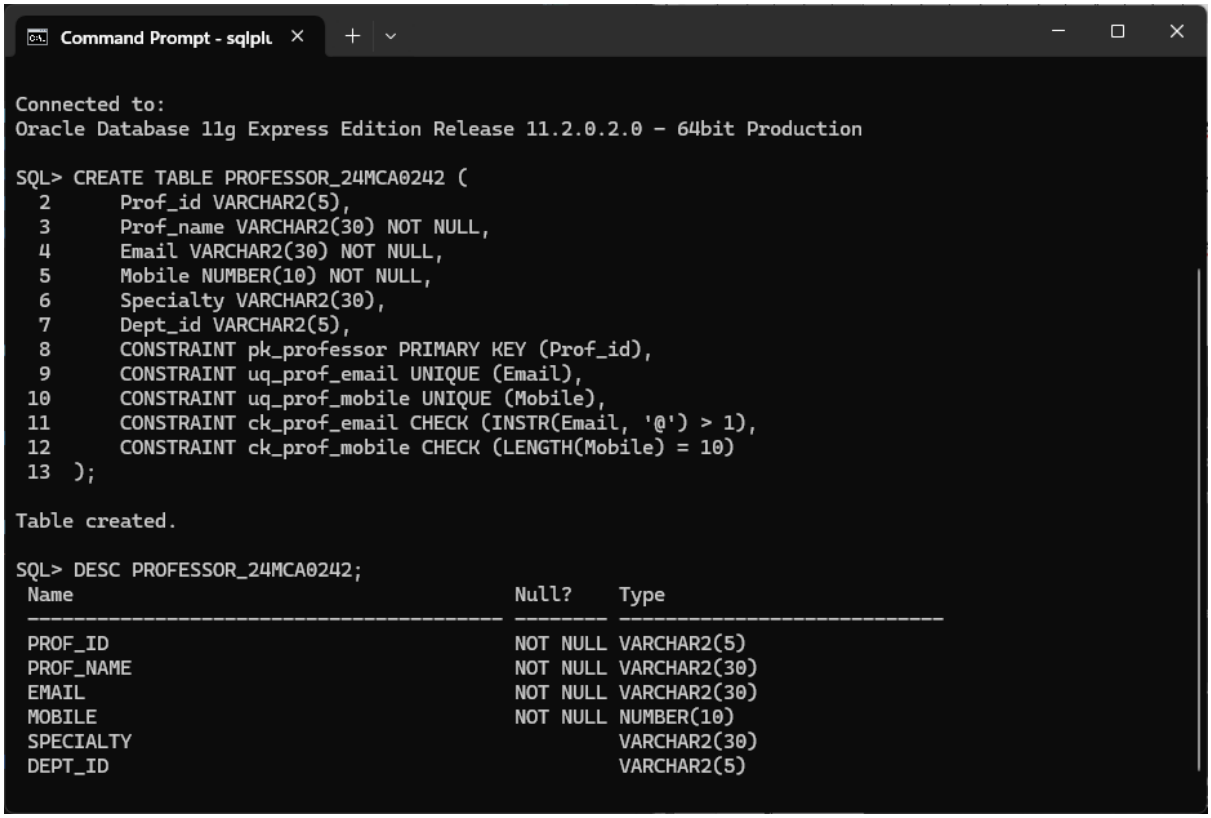
The primary keys are underlined and foreign keys are self-explanatory. The Dept_id column in professor table stands for the department the professor belongs to and Prof_id column in the school table stands for the professor who chairs the school, the same column in the department table stands for the 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)

);

```
SQL> CREATE TABLE SCHOOL_24MCA0242 (
  2      SCode VARCHAR2(5),
  3      Scl_name VARCHAR2(10) NOT NULL,
  4      Prof_id VARCHAR2(5),
  5      Location VARCHAR2(50),
  6      CONSTRAINT pk_school PRIMARY KEY (SCode),
  7      CONSTRAINT fk_school_prof FOREIGN KEY (Prof_id) REFERENCES PROFESSOR_24MCA0242 (Prof_id)
  8  );

Table created.

SQL> DESC SCHOOL_24MCA0242;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SCODE                                     NOT NULL VARCHAR2(5)
 SCL_NAME                                  NOT NULL VARCHAR2(10)
 PROF_ID                                            VARCHAR2(5)
 LOCATION                                           VARCHAR2(50)

SQL>
```

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)

);

```
SQL> CREATE TABLE DEPARTMENT_24MCA0242 (
  2      Dept_id VARCHAR2(5),
  3      Dname VARCHAR2(50) NOT NULL,
  4      SCode VARCHAR2(5),
  5      Prof_id VARCHAR2(5),
  6      CONSTRAINT pk_department PRIMARY KEY (Dept_id),
  7      CONSTRAINT fk_dept_school FOREIGN KEY (SCode) REFERENCES SCHOOL_24MCA0242 (SCode),
  8      CONSTRAINT fk_dept_prof FOREIGN KEY (Prof_id) REFERENCES PROFESSOR_24MCA0242 (Prof_id)
  9  );

Table created.

SQL> DESC DEPARTMENT_24MCA0242;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPT_ID                                   NOT NULL VARCHAR2(5)
 DNAME                                     NOT NULL VARCHAR2(50)
 SCODE                                              VARCHAR2(5)
 PROF_ID                                            VARCHAR2(5)

SQL>
```

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)

);

```
SQL> CREATE TABLE COURSE_24MCA0242 (
  2      Crs_code VARCHAR2(5),
  3      Crs_name VARCHAR2(50) NOT NULL,
  4      Description VARCHAR2(50),
  5      Credits NUMBER(2) NOT NULL,
  6      Hours NUMBER(3) NOT NULL,
  7      CONSTRAINT pk_course PRIMARY KEY (Crs_code)
  8  );

Table created.

SQL> DESC COURSE_24MCA0242;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CRS_CODE                                  NOT NULL VARCHAR2(5)
 CRS_NAME                                  NOT NULL VARCHAR2(50)
 DESCRIPTION                                        VARCHAR2(50)
 CREDITS                                   NOT NULL NUMBER(2)
 HOURS                                     NOT NULL NUMBER(3)

SQL>
```

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)
);
```

```
SQL> CREATE TABLE STUDENT_24MCA0242 (
  2      Reg_no VARCHAR2(5),
  3      Sname VARCHAR2(30) NOT NULL,
  4      Address VARCHAR2(50),
  5      DoB DATE NOT NULL,
  6      Email VARCHAR2(30) NOT NULL,
  7      Mobile NUMBER(10) NOT NULL,
  8      Dept_id VARCHAR2(5),
  9      Prof_id VARCHAR2(5),
 10      CONSTRAINT pk_student PRIMARY KEY (Reg_no),
 11      CONSTRAINT fk_student_dept FOREIGN KEY (Dept_id) REFERENCES DEPARTMENT_24MCA0242 (Dept_id),

 12      CONSTRAINT fk_student_prof FOREIGN KEY (Prof_id) REFERENCES PROFESSOR_24MCA0242 (Prof_id),
 13      CONSTRAINT uq_student_email UNIQUE (Email),
 14      CONSTRAINT uq_student_mobile UNIQUE (Mobile),
 15      CONSTRAINT ck_student_email CHECK (INSTR(Email, '@') > 1),
 16      CONSTRAINT ck_student_mobile CHECK (LENGTH(Mobile) = 10)
 17  );

Table created.

SQL> DESC STUDENT_24MCA0242;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 REG_NO                                    NOT NULL VARCHAR2(5)
 SNAME                                     NOT NULL VARCHAR2(30)
 ADDRESS                                            VARCHAR2(50)
 DOB                                       NOT NULL DATE
 EMAIL                                     NOT NULL VARCHAR2(30)
 MOBILE                                    NOT NULL NUMBER(10)
 DEPT_ID                                            VARCHAR2(5)
 PROF_ID                                            VARCHAR2(5)

SQL>
```

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)

);

```
SQL> CREATE TABLE PROGRAMME_24MCA0242 (
  2      Prog_code VARCHAR2(5),
  3      Prog_name VARCHAR2(30) NOT NULL,
  4      Prog_preamble VARCHAR2(50),
  5      Scode VARCHAR2(5),
  6      Dept_id VARCHAR2(5),
  7      CONSTRAINT pk_programme PRIMARY KEY (Prog_code),
  8      CONSTRAINT fk_prog_school FOREIGN KEY (Scode) REFERENCES SCHOOL_24MCA0242 (SCode),
  9      CONSTRAINT fk_prog_dept FOREIGN KEY (Dept_id) REFERENCES DEPARTMENT_24MCA0242 (Dept_id)
 10  );

Table created.

SQL> DESC PROGRAMME_24MCA0242;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PROG_CODE                                 NOT NULL VARCHAR2(5)
 PROG_NAME                                 NOT NULL VARCHAR2(30)
 PROG_PREAMBLE                                      VARCHAR2(50)
 SCODE                                              VARCHAR2(5)
 DEPT_ID                                            VARCHAR2(5)
```

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%')
);
```

```
SQL> CREATE TABLE SEMESTER_24MCA0242 (
  2      Sem_code VARCHAR2(10),
  3      Term VARCHAR2(5) CHECK (Term IN ('Winter', 'Fall')),
  4      Year NUMBER(4) NOT NULL,
  5      Sdate DATE NOT NULL,
  6      Edate DATE NOT NULL,
  7      CONSTRAINT pk_semester PRIMARY KEY (Sem_code),
  8      CONSTRAINT ck_sem_code CHECK (Sem_code LIKE 'Win%' OR Sem_code LIKE 'Fall%')
  9  );

Table created.
```

```
SQL> DESC SEMESTER_24MCA0242;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SEM_CODE                                  NOT NULL VARCHAR2(10)
 TERM                                               VARCHAR2(5)
 YEAR                                      NOT NULL NUMBER(4)
 SDATE                                     NOT NULL DATE
 EDATE                                     NOT NULL DATE
```

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)

);

```
SQL> CREATE TABLE CLASS_24MCA0242 (
  2      Cls_code VARCHAR2(5),
  3      Slot VARCHAR2(10) NOT NULL,
  4      Stime TIMESTAMP(0) NOT NULL,
  5      Etime TIMESTAMP(0) NOT NULL,
  6      Crs_code VARCHAR2(5),
  7      Prof_id VARCHAR2(5),
  8      Room_no VARCHAR2(10),
  9      Sem_code VARCHAR2(10),
 10      Day_of_week VARCHAR2(10),
 11      CONSTRAINT pk_class PRIMARY KEY (Cls_code),
 12      CONSTRAINT fk_class_course FOREIGN KEY (Crs_code) REFERENCES COURSE_24MCA0242 (Crs_code),
 13      CONSTRAINT fk_class_prof FOREIGN KEY (Prof_id) REFERENCES PROFESSOR_24MCA0242 (Prof_id),
 14      CONSTRAINT fk_class_sem FOREIGN KEY (Sem_code) REFERENCES SEMESTER_24MCA0242 (Sem_code)
 15  );

Table created.

SQL> DESC CLASS_24MCA0242;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLS_CODE                                  NOT NULL VARCHAR2(5)
 SLOT                                      NOT NULL VARCHAR2(10)
 STIME                                     NOT NULL TIMESTAMP(0)
 ETIME                                     NOT NULL TIMESTAMP(0)
 CRS_CODE                                           VARCHAR2(5)
 PROF_ID                                            VARCHAR2(5)
 ROOM_NO                                            VARCHAR2(10)
 SEM_CODE                                           VARCHAR2(10)
 DAY_OF_WEEK                                        VARCHAR2(10)

SQL>
```

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
);
```

```
SQL> CREATE TABLE ENROLL_24MCA0242 (
  2      Cls_code VARCHAR2(5),
  3      Reg_no VARCHAR2(5),
  4      Enroll_time TIMESTAMP(0) NOT NULL,
  5      Grade CHAR(1) CHECK (Grade IN ('S', 'A', 'B', 'C', 'D')),
  6      CONSTRAINT pk_enroll PRIMARY KEY (Cls_code, Reg_no),
  7      CONSTRAINT fk_enroll_class FOREIGN KEY (Cls_code) REFERENCES CLASS_24MCA0242 (Cls_code) ON DELETE CASCADE,
  8      CONSTRAINT fk_enroll_student FOREIGN KEY (Reg_no) REFERENCES STUDENT_24MCA0242 (Reg_no) ON DELETE CASCADE
  9  );

Table created.

SQL> DESC ENROLL_24MCA0242;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLS_CODE                                  NOT NULL VARCHAR2(5)
 REG_NO                                    NOT NULL VARCHAR2(5)
 ENROLL_TIME                               NOT NULL TIMESTAMP(0)
 GRADE                                              CHAR(1)

SQL>
```

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)

);

```
SQL> CREATE TABLE STUDENT_VISA_24MCA0242 (
  2      Reg_no VARCHAR2(5),
  3      Visa_status VARCHAR2(10),
  4      CONSTRAINT pk_student_visa PRIMARY KEY (Reg_no),
  5      CONSTRAINT fk_student_visa FOREIGN KEY (Reg_no) REFERENCES STUDENT_24MCA0242 (Reg_no)
  6  );

Table created.

SQL>
SQL> DESC STUDENT_VISA_24MCA0242;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 REG_NO                                    NOT NULL VARCHAR2(5)
 VISA_STATUS                                        VARCHAR2(10)

SQL>
```

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');

```
SQL> INSERT INTO PROFESSOR_24MCA0242 (Prof_id,Prof_name,Email,Mobile,Specialty,Dept_id) VALUES ('&Prof_id','&Prof_name',
'&Email','&Mobile','&Specialty','&Dept_id');
Enter value for prof_id: P0001
Enter value for prof_name: Dr. Ravi Kumar
Enter value for email: ravi.kumar@mail.com
Enter value for mobile: 9876543210
Enter value for specialty: Data Science
Enter value for dept_id: D0001
old   1: INSERT INTO PROFESSOR_24MCA0242 (Prof_id,Prof_name,Email,Mobile,Specialty,Dept_id) VALUES ('&Prof_id','&Prof_na
me','&Email','&Mobile','&Specialty','&Dept_id')
new   1: INSERT INTO PROFESSOR_24MCA0242 (Prof_id,Prof_name,Email,Mobile,Specialty,Dept_id) VALUES ('P0001','Dr. Ravi Ku
mar','ravi.kumar@mail.com','9876543210','Data Science','D0001')

1 row created.

SQL> /
```

```
SQL> SELECT * FROM PROFESSOR_24MCA0242;

PROF_ PROF_NAME                    EMAIL                          MOBILE
----- ---------------------------- ------------------------------ ----------
SPECIALTY                  DEPT_
-------------------------- -----
P0001 Dr. Ravi Kumar               ravi.kumar@mail.com            9876543210
Data Science               D0001

P0002 Dr. Neha Sharma              neha.sharma@uni.edu            9876543211
Machine Learning           D0002

P0003 Dr. Arjun Mehta              arjun.mehta@uni.edu            9876543212
Cybersecurity              D0003


PROF_ PROF_NAME                    EMAIL                          MOBILE
----- ---------------------------- ------------------------------ ----------
SPECIALTY                  DEPT_
-------------------------- -----
P0004 Dr. Priya Singh              priya.singh@uni.edu            9876543213
Artificial Intelligence    D0004

P0005 Dr. Suresh Gupta             suresh.gupta@uni.edu           9876543214
Cloud Computing            D0005


SQL>
```

SCHOOL TABLE:

INSERT INTO SCHOOL_24MCA0242 (Scode, Scl_name, Prof_id, Location) VALUES ('&Scode', '&Scl_name', '&Prof_id', '&Location');

```
SQL> INSERT INTO SCHOOL_24MCA0242 (Scode,Scl_name,Prof_id,Location) VALUES ('&Scode','&Scl_name','&Prof_id','&Location');
Enter value for scode: S0001
Enter value for scl_name: Science
Enter value for prof_id: P0001
Enter value for location: Mumbai
old   1: INSERT INTO SCHOOL_24MCA0242 (Scode,Scl_name,Prof_id,Location) VALUES ('&Scode','&Scl_name','&Prof_id','&Location')
new   1: INSERT INTO SCHOOL_24MCA0242 (Scode,Scl_name,Prof_id,Location) VALUES ('S0001','Science','P0001','Mumbai')

1 row created.

SQL>
```

```
SQL> SELECT * FROM SCHOOL_24MCA0242;

SCODE SCL_NAME   PROF_ LOCATION
----- ---------- ----- --------------------------------------------------
S0001 Science    P0001 Mumbai
S0002 Arts       P0002 Delhi
S0003 Commerce   P0003 Bangalore
S0005 Law        P0005 Chennai

SQL>
```

DEPARTMENT TABLE:

INSERT INTO DEPARTMENT_24MCA0242 (Dept_id, Dname, Scode, Prof_id) VALUES ('&Dept_id', '&Dname', '&Scode', '&Prof_id');

```
SQL> INSERT INTO DEPARTMENT_24MCA0242 (Dept_id,Dname,Scode,Prof_id) VALUES ('&Dept_id','&Dname','&Scode','&Prof_id');
Enter value for dept_id: D0001
Enter value for dname: Computer Science
Enter value for scode: S0001
Enter value for prof_id: P0001
old   1: INSERT INTO DEPARTMENT_24MCA0242 (Dept_id,Dname,Scode,Prof_id) VALUES ('&Dept_id','&Dname','&Scode','&Prof_id')
new   1: INSERT INTO DEPARTMENT_24MCA0242 (Dept_id,Dname,Scode,Prof_id) VALUES ('D0001','Computer Science','S0001','P000
1')

1 row created.

SQL> /
```

```
SQL> SELECT * FROM DEPARTMENT_24MCA0242;

DEPT_ DNAME                                              SCODE PROF_
----- -------------------------------------------------- ----- -----
D0001 Computer Science                                   S0001 P0001
D0003 Physics                                            S0001 P0003
D0002 Mechanical Engineering                             S0004 P0002
D0004 Electrical Engineering                             S0004 P0004
D0005 Civil Engineering                                  S0004 P0005

SQL>
```

COURSE TABLE:

INSERT INTO COURSE_24MCA0242 (Crs_code, Crs_name, Description, Credits, Hours) VALUES ('&Crs_code', '&Crs_name', '&Description', '&Credits', '&Hours');

```
SQL> INSERT INTO COURSE_24MCA0242 (Crs_code,Crs_name,Description,Credits,Hours) VALUES ('&Crs_code','&Crs_name','&Description',
'&Credits','&Hours');
Enter value for crs_code: C0001
Enter value for crs_name: Data Structures
Enter value for description: Introduction to Data Structures
Enter value for credits: 4
Enter value for hours: 40
old   1: INSERT INTO COURSE_24MCA0242 (Crs_code,Crs_name,Description,Credits,Hours) VALUES ('&Crs_code','&Crs_name','&Descripti
on','&Credits','&Hours')
new   1: INSERT INTO COURSE_24MCA0242 (Crs_code,Crs_name,Description,Credits,Hours) VALUES ('C0001','Data Structures','Introduc
tion to Data Structures','4','40')

1 row created.

SQL>
```

```
SQL> SELECT * FROM COURSE_24MCA0242;

CRS_C CRS_NAME
----- --------------------------------------------------
DESCRIPTION                                        CREDITS      HOURS
-------------------------------------------------- ---------- ----------
CRS01 Data Structures
Introduction to Data Structures                             4         40

CRS02 Operating Systems
Study of Operating Systems                                  3         33

CRS03 Database Management
Database Concepts and SQL                                   5         55


CRS_C CRS_NAME
----- --------------------------------------------------
DESCRIPTION                                        CREDITS      HOURS
-------------------------------------------------- ---------- ----------
CRS04 Algorithms
Design and Analysis of Algorithms                           4         45

CRS05 Artificial Intelligence
Introduction to AI Concepts                                 5         60


SQL>
```

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');

```
SQL> 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');
Enter value for reg_no: R0001
Enter value for sname: Amit Verma
Enter value for address: Mumbai
Enter value for dob: 1998-05-12
Enter value for email: amit.verma@student.mail
Enter value for mobile: 9876543215
Enter value for dept_id: D0001
Enter value for prof_id: P0001
old   1: INSERT INTO STUDENT_24MCA0242 (Reg_no,Sname,Address,DoB,Email,Mobile,Dept_id,Prof_id) VALUES ('&Reg_no','&Sna
me','&Address',TO_DATE('&DoB','YYYY-MM-DD'),'&Email','&Mobile','&Dept_id','&Prof_id')
new   1: INSERT INTO STUDENT_24MCA0242 (Reg_no,Sname,Address,DoB,Email,Mobile,Dept_id,Prof_id) VALUES ('R0001','Amit V
erma','Mumbai',TO_DATE('1998-05-12','YYYY-MM-DD'),'amit.verma@student.mail','9876543215','D0001','P0001')

1 row created.

SQL>
```

```
SQL> SELECT * FROM STUDENT_24MCA0242;

REG_N SNAME
----- ------------------------------
ADDRESS                                            DOB
-------------------------------------------------- ---------
EMAIL                          MOBILE DEPT_ PROF_
------------------------------ ---------- ----- -----
R0001 Amit Verma
Mumbai                                             12-MAY-98
amit.verma@student.mail        9876543215 D0001 P0001

R0002 Sunita Desai
Delhi                                              20-JUL-99
sunita.desai@student.edu       9876543216 D0002 P0002

REG_N SNAME
----- ------------------------------
ADDRESS                                            DOB
-------------------------------------------------- ---------
EMAIL                          MOBILE DEPT_ PROF_
------------------------------ ---------- ----- -----

R0003 Rahul Singh
Bangalore                                          30-NOV-97
rahul.singh@student.edu        9876543217 D0003 P0003

R0004 Pooja Reddy
Hyderabad                                          15-FEB-96

REG_N SNAME
----- ------------------------------
ADDRESS                                            DOB
-------------------------------------------------- ---------
EMAIL                          MOBILE DEPT_ PROF_
------------------------------ ---------- ----- -----
pooja.reddy@student.edu        9876543218 D0004 P0004

R0005 Anil Kumar
Chennai                                            05-SEP-98
anil.kumar@student.edu         9876543219 D0005 P0005


SQL>
```

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');

```
SQL> INSERT INTO PROGRAMME_24MCA0242 (Prog_code,Prog_name,Prog_preamble,Scode,Dept_id) VALUES ('&Prog_code','&Prog_name',
'&Prog_preamble','&Scode','&Dept_id');
Enter value for prog_code: PRG01
Enter value for prog_name: B.Tech CSE
Enter value for prog_preamble: Computer Science and Engineering
Enter value for scode: S0001
Enter value for dept_id: D0001
old   1: INSERT INTO PROGRAMME_24MCA0242 (Prog_code,Prog_name,Prog_preamble,Scode,Dept_id) VALUES ('&Prog_code','&Prog_na
me','&Prog_preamble','&Scode','&Dept_id')
new   1: INSERT INTO PROGRAMME_24MCA0242 (Prog_code,Prog_name,Prog_preamble,Scode,Dept_id) VALUES ('PRG01','B.Tech CSE','
Computer Science and Engineering','S0001','D0001')

1 row created.

SQL>
```

```
SQL> SELECT * FROM PROGRAMME_24MCA0242;

PROG_ PROG_NAME
----- ------------------------------
PROG_PREAMBLE                                      SCODE DEPT_
-------------------------------------------------- ----- -----
PRG01 B.Tech CSE
Computer Science and Engineering                   S0001 D0001

PRG02 B.Tech ME
Mechanical Engineering                             S0004 D0002

PRG03 B.Sc Physics
Bachelor of Science in Physics                     S0001 D0003


PROG_ PROG_NAME
----- ------------------------------
PROG_PREAMBLE                                      SCODE DEPT_
-------------------------------------------------- ----- -----
PRG04 B.Tech EE
Electrical Engineering                             S0004 D0004

PRG05 B.Tech CE
Civil Engineering                                  S0004 D0005


SQL>
```

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'));

```
SQL> 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'));
Enter value for sem_code: Win2024
Enter value for term: Winter
Enter value for year: 2024
Enter value for sdate: 2024-01-10
Enter value for edate: 2024-05-15
old   1: INSERT INTO SEMESTER_24MCA0242 (Sem_code,Term,Year,Sdate,Edate) VALUES ('&Sem_code','&Term','&Year',TO_DATE('&Sd
ate','YYYY-MM-DD'),TO_DATE('&Edate','YYYY-MM-DD'))
new   1: INSERT INTO SEMESTER_24MCA0242 (Sem_code,Term,Year,Sdate,Edate) VALUES ('Win2024','Winter','2024',TO_DATE('2024-
01-10','YYYY-MM-DD'),TO_DATE('2024-05-15','YYYY-MM-DD'))

1 row created.

SQL>
```

```
SQL> SELECT * FROM SEMESTER_24MCA0242;

SEM_CODE   TERM             YEAR SDATE     EDATE
---------- ---------- ---------- --------- ---------
Win2024    Winter           2024 10-JAN-24 15-MAY-24
Fall2024   Fall             2024 10-AUG-24 20-DEC-24
Win2023    Winter           2023 15-JAN-23 20-MAY-23
Fall2023   Fall             2023 15-AUG-23 25-DEC-23
Win2022    Winter           2022 20-JAN-22 25-MAY-22

SQL>
```

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');

```
SQL> SELECT * FROM CLASS_24MCA0242;

CLS_C SLOT
----- ----------
STIME
---------------------------------------------------------------------------
ETIME
---------------------------------------------------------------------------
CRS_C PROF_ ROOM_NO    SEM_CODE   DAY_OF_WEE
----- ----- ---------- ---------- ----------
CLS01 A1
20-AUG-24 09.00.00 AM
20-AUG-24 10.30.00 AM
CRS01 P0001 SJT201     Fall2024   Monday


CLS_C SLOT
----- ----------
STIME
---------------------------------------------------------------------------
ETIME
---------------------------------------------------------------------------
CRS_C PROF_ ROOM_NO    SEM_CODE   DAY_OF_WEE
----- ----- ---------- ---------- ----------
CLS02 B1
20-AUG-24 11.00.00 AM
20-AUG-24 12.30.00 PM
CRS02 P0002 SMV102     Fall2024   Wednesday


CLS_C SLOT
----- ----------
STIME
---------------------------------------------------------------------------
ETIME
---------------------------------------------------------------------------
CRS_C PROF_ ROOM_NO    SEM_CODE   DAY_OF_WEE
----- ----- ---------- ---------- ----------
CLS03 C1
21-AUG-24 09.00.00 AM
21-AUG-24 10.30.00 AM
CRS03 P0003 TT103      Fall2024   Thursday
```

```
CLS_C SLOT
----- ----------
STIME
---------------------------------------------------------------------------
ETIME
---------------------------------------------------------------------------
CRS_C PROF_ ROOM_NO    SEM_CODE   DAY_OF_WEE
----- ----- ---------- ---------- ----------
CLS04 D1
22-AUG-24 02.00.00 PM
22-AUG-24 03.30.00 PM
CRS04 P0004 SJT104     Fall2024   Friday


CLS_C SLOT
----- ----------
STIME
---------------------------------------------------------------------------
ETIME
---------------------------------------------------------------------------
CRS_C PROF_ ROOM_NO    SEM_CODE   DAY_OF_WEE
----- ----- ---------- ---------- ----------
CLS05 E1
23-AUG-24 09.00.00 AM
23-AUG-24 10.30.00 AM
CRS05 P0005 TT105      Fall2024   Monday


SQL>
```

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');

```
SQL> 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');
Enter value for cls_code: CLS01
Enter value for reg_no: R0001
Enter value for enroll_time: 2024-01-05 10:00:00
Enter value for grade: A
old   1: INSERT INTO ENROLL_24MCA0242 (Cls_code,Reg_no,Enroll_time,Grade) VALUES ('&Cls_code','&Reg_no',TO_TIMESTAMP('&Enroll_ti
me','YYYY-MM-DD HH24:MI:SS'),'&Grade')
new   1: INSERT INTO ENROLL_24MCA0242 (Cls_code,Reg_no,Enroll_time,Grade) VALUES ('CLS01','R0001',TO_TIMESTAMP('2024-01-05 10:00
:00','YYYY-MM-DD HH24:MI:SS'),'A')

1 row created.

SQL>
```

```
SQL> SELECT * FROM ENROLL_24MCA0242;

CLS_C REG_N
----- -----
ENROLL_TIME                                                                 G
--------------------------------------------------------------------------- -
CLS01 R0001
05-JAN-24 10.00.00 AM                                                       A

CLS02 R0002
08-AUG-21 09.00.00 AM                                                       B

CLS03 R0003
12-JAN-22 11.00.00 AM                                                       S


CLS_C REG_N
----- -----
ENROLL_TIME                                                                 G
--------------------------------------------------------------------------- -
CLS04 R0004
10-AUG-22 02.00.00 PM                                                       C

CLS05 R0005
18-JAN-23 12.00.00 PM                                                       D


SQL>
```

STUDENT_VISA TABLE:

INSERT INTO STUDENT_VISA_24MCA0242 (Reg_no, Visa_status) VALUES ('&Reg_no', '&Visa_status');

```
SQL> INSERT INTO STUDENT_VISA_24MCA0242 (Reg_no,Visa_status) VALUES ('&Reg_no','&Visa_status');
Enter value for reg_no: R0001
Enter value for visa_status: Approved
old   1: INSERT INTO STUDENT_VISA_24MCA0242 (Reg_no,Visa_status) VALUES ('&Reg_no','&Visa_status')
new   1: INSERT INTO STUDENT_VISA_24MCA0242 (Reg_no,Visa_status) VALUES ('R0001','Approved')

1 row created.

SQL>
```

```
SQL> SELECT * FROM STUDENT_VISA_24MCA0242;

REG_N VISA_STATU
----- ----------
R0001 Approved
R0002 Pending
R0003 Approved
R0004 Expired
R0005 Pending

SQL>
```

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

(i) 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);

```
SQL> ALTER TABLE PROFESSOR_24MCA0242
  2  ADD CONSTRAINT ck_prof_id_length CHECK (LENGTH(Prof_id) = 5);

Table altered.

SQL>
SQL> ALTER TABLE PROFESSOR_24MCA0242
  2  ADD CONSTRAINT uq_prof_email UNIQUE (Email);

Table altered.

SQL>
SQL> ALTER TABLE PROFESSOR_24MCA0242
  2  ADD CONSTRAINT uq_prof_mobile UNIQUE (Mobile);

Table altered.

SQL>
SQL> ALTER TABLE PROFESSOR_24MCA0242
  2  ADD CONSTRAINT ck_prof_email CHECK (INSTR(Email, '@') > 1);

Table altered.

SQL>
SQL> ALTER TABLE PROFESSOR_24MCA0242
  2  ADD CONSTRAINT ck_prof_mobile CHECK (LENGTH(Mobile) = 10);

Table altered.
```

(ii) 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));

```
SQL> ALTER TABLE CLASS_24MCA0242 MODIFY (Stime TIMESTAMP(0));

Table altered.

SQL>
SQL> ALTER TABLE CLASS_24MCA0242 MODIFY (Etime TIMESTAMP(0));

Table altered.

SQL>
```

(iii) 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'));

```
SQL> ALTER TABLE SEMESTER_24MCA0242
  2  ADD CONSTRAINT ck_sem_code CHECK (Sem_code LIKE 'Win%' OR Sem_code LIKE 'Fall%');

Table altered.

SQL>
SQL> ALTER TABLE SEMESTER_24MCA0242
  2  ADD CONSTRAINT ck_term CHECK (Term IN ('Winter', 'Fall'));

Table altered.
```

(iv) 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);

```
SQL> ALTER TABLE STUDENT_24MCA0242 ADD CONSTRAINT uq_student_email UNIQUE (Email);

Table altered.

SQL> ALTER TABLE STUDENT_24MCA0242 ADD CONSTRAINT uq_student_mobile UNIQUE (Mobile);

Table altered.

SQL> ALTER TABLE STUDENT_24MCA0242 ADD CONSTRAINT ck_student_email CHECK (INSTR(Email, '@') > 1);

Table altered.

SQL> ALTER TABLE STUDENT_24MCA0242 ADD CONSTRAINT ck_student_mobile CHECK (LENGTH(Mobile) = 10);

Table altered.

SQL>
```

(v) 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'));

```
SQL> ALTER TABLE ENROLL_24MCA0242 MODIFY (Enroll_time TIMESTAMP(0));

Table altered.

SQL> ALTER TABLE ENROLL_24MCA0242 ADD CONSTRAINT ck_enroll_grade CHECK (Grade IN ('S', 'A', 'B', 'C',
'D'));

Table altered.

SQL>
```

(vi) 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;

```
SQL> ALTER TABLE DEPARTMENT_24MCA0242 ADD CONSTRAINT fk_dept_scode FOREIGN KEY (SCode) REFERENCES SCHO
OL_24MCA0242(SCode) ON DELETE SET NULL DEFERRABLE;

Table altered.

SQL> ALTER TABLE PROFESSOR_24MCA0242 ADD CONSTRAINT fk_prof_dept FOREIGN KEY (Dept_id) REFERENCES DEPA
RTMENT_24MCA0242(Dept_id) ON DELETE CASCADE DEFERRABLE;

Table altered.

SQL> ALTER TABLE STUDENT_24MCA0242 ADD CONSTRAINT fk_student_dept FOREIGN KEY (Dept_id) REFERENCES DEP
ARTMENT_24MCA0242(Dept_id) ON DELETE SET NULL DEFERRABLE;

Table altered.

SQL>
```

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

ALTER TABLE SEMESTER_24MCA0242 ADD CONSTRAINT CK_SEM_DATE CHECK (Sdate < Edate);

```
SQL> ALTER TABLE SEMESTER_24MCA0242 ADD CONSTRAINT CK_SEM_DATE_RANGE CHECK (Sdate < Edate);

Table altered.

SQL>
```

4. In built functions

(i) 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.

a) UPPER

SELECT UPPER(Prof_name) AS Uppercase_Prof_Name

FROM PROFESSOR_24MCA0242;

```
SQL> SELECT UPPER(Prof_name) AS Uppercase_Prof_Name FROM PROFESSOR_24MCA0242;

UPPERCASE_PROF_NAME
---------------------------
DR. RAVI KUMAR
DR. NEHA SHARMA
DR. ARJUN MEHTA
DR. PRIYA SINGH
DR. SURESH GUPTA

SQL>
```

b) LOWER

SELECT LOWER(Email) AS Lowercase_Email FROM PROFESSOR_24MCA0242;

```
SQL> SELECT LOWER(Email) AS Lowercase_Email FROM PROFESSOR_24MCA0242;

LOWERCASE_EMAIL
---------------------------
arjun.mehta@uni.edu
neha.sharma@uni.edu
priya.singh@uni.edu
ravi.kumar@mail.com
suresh.gupta@uni.edu

SQL>
```

c) INITCAP

SELECT INITCAP(Speciality) AS Capitalized_Speciality

FROM PROFESSOR_24MCA0242;

```
SQL> SELECT INITCAP(Specialty) AS Capitalized_Specialty FROM PROFESSOR_24MCA0242;

CAPITALIZED_SPECIALTY
---------------------------
Data Science
Machine Learning
Cybersecurity
Artificial Intelligence
Cloud Computing

SQL>
```

d) LENGTH

SELECT Prof_name, LENGTH(Prof_name) AS Name_Length

FROM PROFESSOR_24MCA0242;

```
SQL> SELECT Prof_name, LENGTH(Prof_name) AS Name_Length FROM PROFESSOR_24MCA0242;

PROF_NAME                    NAME_LENGTH
---------------------------- -----------
Dr. Ravi Kumar                        14
Dr. Neha Sharma                       15
Dr. Arjun Mehta                       15
Dr. Priya Singh                       15
Dr. Suresh Gupta                      16

SQL>
```

e) LPAD

SELECT Prof_id, LPAD(Prof_id, 10, 'PROF_') AS Padded_Prof_ID

FROM PROFESSOR_24MCA0242;

```
SQL> SELECT Prof_id, LPAD(Prof_id, 10, 'PROF_') AS Padded_Prof_ID FROM PROFESSOR_24MCA0242;

PROF_ PADDED_PROF_ID
----- ----------------------------------------
P0001 PROF_P0001
P0002 PROF_P0002
P0003 PROF_P0003
P0004 PROF_P0004
P0005 PROF_P0005

SQL>
```

f) RPAD

SELECT Prof_id, RPAD(Prof_id, 10, '*') AS Padded_Prof_Name

FROM PROFESSOR_24MCA0242;

```
SQL> SELECT Prof_id, RPAD(Prof_id, 10, '*') AS Padded_Prof_Name FROM PROFESSOR_24MCA0242;

PROF_ PADDED_PROF_NAME
----- ----------------------------------------
P0001 P0001*****
P0002 P0002*****
P0003 P0003*****
P0004 P0004*****
P0005 P0005*****

SQL>
```

g) LTRIM

SELECT LTRIM(Prof_name, 'Dr. ') AS Trimmed_Prof_Name

FROM PROFESSOR_24MCA0242;

```
SQL> SELECT LTRIM(Prof_name, 'Dr. ') AS LTrimmed_Prof_Name FROM PROFESSOR_24MCA0242;

LTRIMMED_PROF_NAME
------------------------------
Ravi Kumar
Neha Sharma
Arjun Mehta
Priya Singh
Suresh Gupta

SQL>
```

h) RTRIM

SELECT RTRIM(Prof_name, 'a') AS RTrimmed_Prof_Name

FROM PROFESSOR_24MCA0242;

```
SQL> SELECT RTRIM(Prof_name, 'a') AS RTrimmed_Prof_Name FROM PROFESSOR_24MCA0242;

RTRIMMED_PROF_NAME
------------------------------
Dr. Ravi Kumar
Dr. Neha Sharm
Dr. Arjun Meht
Dr. Priya Singh
Dr. Suresh Gupt

SQL>
```

i) TRIM

SELECT TRIM('D' FROM Prof_name) AS Trimmed_Prof_Name

FROM PROFESSOR_24MCA0242;

```
SQL> SELECT TRIM('D' FROM Prof_name) AS Trimmed_Prof_Name FROM PROFESSOR_24MCA0242;

TRIMMED_PROF_NAME
------------------------------
r. Ravi Kumar
r. Neha Sharma
r. Arjun Mehta
r. Priya Singh
r. Suresh Gupta

SQL>
```

(ii) Write query to illustrate usage of NVL function and NULLIF function.

a) NVL

SELECT SCode, NVL(Location, 'No Location Provided') AS Location FROM SCHOOL_24MCA0242;

```
SQL> SELECT SCode, NVL(Location, 'No Location Provided') AS Location FROM SCHOOL_24MCA0242;

SCODE LOCATION
----- --------------------------------------------------
S0001 Mumbai
S0002 Delhi
S0003 Bangalore
S0005 Chennai
S0004 Hyderabad
S0006 No Location Provided

6 rows selected.

SQL>
```

b) NULLIF

SELECT SCode, NULLIF(Location, 'No Location Provided') AS Location FROM SCHOOL_24MCA0242;

```
SQL> SELECT SCode, NULLIF(Location, 'No Location Provided') AS Location FROM SCHOOL_24MCA0242;

SCODE LOCATION
----- --------------------------------------------------
S0001 Mumbai
S0002 Delhi
S0003 Bangalore
S0005 Chennai
S0004 Hyderabad
S0006

6 rows selected.

SQL>
```

(iii) Display the name of the students who were born on a specified month.

SELECT Sname FROM STUDENT_24MCA0242 WHERE EXTRACT(MONTH FROM DOB) = 2;

```
SQL> SELECT Sname FROM STUDENT_24MCA0242 WHERE EXTRACT(MONTH FROM DOB) = 2;

SNAME
------------------------------
Pooja Reddy

SQL>
```

(iv) 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');

```
SQL> SELECT Sname FROM STUDENT_24MCA0242 WHERE DOB = TO_DATE('1998-05-12', 'YYYY-MM-DD');

SNAME
------------------------------
Amit Verma

SQL>
```

(v) 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';

```
SQL> SELECT Sname, TO_CHAR(DoB, 'Day, Month dd, yyyy') AS Formatted_DoB FROM STUDENT_24MCA0242 WHERE Reg_no = 'R0003';

SNAME
------------------------------
FORMATTED_DOB
--------------------------------------------------------------------------------
Rahul Singh
Sunday   , November  30, 1997


SQL>
```

(vi) 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';

```
SQL> 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';

CLS_C S_HR_ E_HR_
----- ----- -----
CLS01 09:00 10:30

SQL>
```

(vii) 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';

```
SQL> 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 = 'Win2017-18';

SEM_CODE        START_DAY_OF_WEEK
--------------- ------------------------------------
END_DAY_OF_WEEK
------------------------------------
Win2017-18      Friday
Thursday


SQL>
```

(viii) 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';

```
SQL> SELECT Sem_code, ROUND((Edate - Sdate) / 7, 2) AS Duration_in_Weeks FROM SEMESTER_24MCA0242 WHERE
Sem_code = 'Win2017-18';

SEM_CODE        DURATION_IN_WEEKS
--------------- -----------------
Win2017-18                17.86

SQL>
```

(ix) 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';

```
SQL> INSERT INTO STUDENT_24MCA0242 (Reg_no, Sname, Address, DoB, Email, Mobile, Dept_id, Prof_id) VALUES ('R0007', 'Sama
y Raina', 'Pune', TO_DATE('2003-05-12', 'YYYY-MM-DD'), 'samay.raina@student.edu', '9876543888', 'D0002', 'P0002');

1 row created.

SQL> SELECT Sname, TO_CHAR(DOB, 'DD/MM/YY') AS Formatted_DoB FROM STUDENT_24MCA0242 WHERE Reg_no = 'R0007';

SNAME                          FORMATTE
------------------------------ --------
Samay Raina                    12/05/03

SQL>
```

(x) 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.

a) CEIL

SELECT AVG(Credits) AS Avg_Credits, CEIL(AVG(Credits)) AS Ceil_Avg_Credits FROM COURSE_24MCA0242;

```
SQL> SELECT AVG(Credits) AS Avg_Credits, CEIL(AVG(Credits)) AS Ceil_Avg_Credits FROM COURSE_24MCA0242;

AVG_CREDITS CEIL_AVG_CREDITS
----------- ----------------
        4.2                5

SQL>
```

b) FLOOR

SELECT AVG(Credits) AS Avg_Credits, FLOOR(AVG(Credits)) AS Floor_Avg_Credits FROM COURSE_24MCA0242;

```
SQL> SELECT AVG(Credits) AS Avg_Credits, FLOOR(AVG(Credits)) AS Floor_Avg_Credits FROM COURSE_24MCA0242;

AVG_CREDITS FLOOR_AVG_CREDITS
----------- -----------------
        4.2                 4

SQL>
```

c) TRUNCATE

SELECT TRUNC(AVG(Credits)) AS Truncated_Avg_Credits FROM COURSE_24MCA0242;

```
SQL> SELECT TRUNC(AVG(Credits)) AS Truncated_Avg_Credits FROM COURSE_24MCA0242;

TRUNCATED_AVG_CREDITS
---------------------
                    4

SQL>
```

d) MIN

SELECT MIN(Credits) AS Min_Credits FROM COURSE_24MCA0242;

```
SQL> SELECT MIN(Credits) AS Min_Credits FROM COURSE_24MCA0242;

MIN_CREDITS
-----------
          3

SQL>
```

e) MAX

SELECT MAX(Credits) AS Max_Credits FROM COURSE_24MCA0242;

```
SQL> SELECT MAX(Credits) AS Max_Credits FROM COURSE_24MCA0242;

MAX_CREDITS
-----------
          5

SQL>
```

f) AVG

SELECT AVG(Credits) AS Avg_Credits FROM COURSE_24MCA0242;

```
SQL> SELECT AVG(Credits) AS Avg_Credits FROM COURSE_24MCA0242;

AVG_CREDITS
-----------
        4.2

SQL>
```

g) COUNT

SELECT COUNT(*) AS Total_Students FROM STUDENT_24MCA0242;

```
SQL> SELECT COUNT(*) AS Total_Students FROM STUDENT_24MCA0242;

TOTAL_STUDENTS
--------------
             7

SQL>
```

5. Write queries for

(i) 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%';

```
SQL> SELECT Sname, Email, Address FROM STUDENT_24MCA0242 WHERE Address = 'Katpadi' AND Sname LIKE '__l%';

SNAME                          EMAIL
------------------------------ ------------------------------
ADDRESS
--------------------------------------------------
Kalpana Roy                    kalpana.roy@student.edu
Katpadi


SQL>
```

(ii) 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');

```
SQL> SELECT Sname, Email, Address FROM STUDENT_24MCA0242 WHERE Address NOT IN('Tamil Nadu');

SNAME                          EMAIL
------------------------------ ------------------------------
ADDRESS
--------------------------------------------------
Amit Verma                     amit.verma@student.mail
Mumbai

Sunita Desai                   sunita.desai@student.edu
Delhi

Rahul Singh                    rahul.singh@student.edu
Bangalore


SNAME                          EMAIL
------------------------------ ------------------------------
ADDRESS
--------------------------------------------------
Pooja Reddy                    pooja.reddy@student.edu
Hyderabad

Anil Kumar                     anil.kumar@student.edu
Chennai

Sanjay Patil                   sanjay.patil@student.edu
Pune
```

(iii) Display name, email address and address of foreign students only.

SELECT Sname, Email, Address FROM STUDENT_24MCA0242 WHERE Address NOT LIKE '%India';

```
SQL> SELECT Sname, Email, Address FROM STUDENT_24MCA0242 WHERE Address NOT LIKE '%India';

SNAME                          EMAIL
------------------------------ ------------------------------
ADDRESS
--------------------------------------------------
Maria Garcia                   maria.garcia@student.edu
Madrid, Spain

John Doe                       john.doe@student.edu
New York, USA


SQL>
```

(iv) 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';

```
SQL>
SQL> SELECT P.Prof_name, P.Specialty FROM PROFESSOR_24MCA0242 P JOIN DEPARTMENT_24MCA0242 D ON P.Dept_id = D.Dept_id JOI
N SCHOOL_24MCA0242 S ON D.Scode = S.Scode WHERE S.Scl_name = 'School of Medicine';

PROF_NAME                      SPECIALTY
------------------------------ --------------------------
Dr. Ramesh Iyer                Neuroscience

SQL> 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';

PROF_NAME                      SPECIALTY
------------------------------ ------------------------------
SCL_NAME
--------------------
Dr. Ramesh Iyer                Neuroscience
School of Medicine


SQL>
```

(v) 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;

```
SQL> SELECT Scl_name,Prof_name FROM SCHOOL_24MCA0242,PROFESSOR_24MCA0242 WHERE SCHOOL_24MCA0242.Prof_id=PROFES
SOR_24MCA0242.Prof_id;

SCL_NAME             PROF_NAME
-------------------- ------------------------------
Science              Dr. Ravi Kumar
Arts                 Dr. Neha Sharma
Commerce             Dr. Arjun Mehta
Engineering          Dr. Priya Singh
Law                  Dr. Suresh Gupta
Law                  Dr. Suresh Gupta
School of Medicine   Dr. Ramesh Iyer

7 rows selected.

SQL>
```

(vi) 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;

```
SQL> SELECT Crs_code, Crs_name, Description FROM COURSE_24MCA0242 ORDER BY Crs_code;

CRS_C CRS_NAME
----- --------------------------------------------------
DESCRIPTION
--------------------------------------------------
CRS01 Data Structures
Introduction to Data Structures

CRS02 Operating Systems
Study of Operating Systems

CRS03 Database Management
Database Concepts and SQL


CRS_C CRS_NAME
----- --------------------------------------------------
DESCRIPTION
--------------------------------------------------
CRS04 Algorithms
Design and Analysis of Algorithms

CRS05 Artificial Intelligence
Introduction to AI Concepts


SQL>
```

(vii) Change the mobile number of a student interactively.

UPDATE STUDENT_24MCA0242 SET Mobile = &Mobile WHERE Reg_no = '&Reg_no';

```
SQL> UPDATE STUDENT_24MCA0242 SET Mobile=&Mobile WHERE Reg_no='&Reg_no';
Enter value for mobile: 9000652000
Enter value for reg_no: R0004
old   1: UPDATE STUDENT_24MCA0242 SET Mobile=&Mobile WHERE Reg_no='&Reg_no'
new   1: UPDATE STUDENT_24MCA0242 SET Mobile=9000652000 WHERE Reg_no='R0004'

1 row updated.

SQL>
```

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

```
SQL> SELECT * FROM Enroll_24MCA0242;

CLS_C REG_N
----- -----
ENROLL_TIME                                                                 G
--------------------------------------------------------------------------- -
CLS01 R0001
05-JAN-24 10.00.00 AM                                                       A

CLS02 R0002
08-AUG-21 09.00.00 AM                                                       B

CLS03 R0003
12-JAN-22 11.00.00 AM                                                       S


CLS_C REG_N
----- -----
ENROLL_TIME                                                                 G
--------------------------------------------------------------------------- -
CLS04 R0004
10-AUG-22 02.00.00 PM                                                       C

CLS05 R0005
18-JAN-23 12.00.00 PM                                                       D
```

DELETE FROM ENROLL_24MCA0242 WHERE Cls_code = '&Cls_code' AND Reg_no = '&Reg_no';

```
SQL> DELETE FROM ENROLL_24MCA0242 WHERE Cls_code = '&Cls_code' AND Reg_no = '&Reg_no';
Enter value for cls_code: CLS04
Enter value for reg_no: R0004
old   1: DELETE FROM ENROLL_24MCA0242 WHERE Cls_code = '&Cls_code' AND Reg_no = '&Reg_no'
new   1: DELETE FROM ENROLL_24MCA0242 WHERE Cls_code = 'CLS04' AND Reg_no = 'R0004'

1 row deleted.

SQL>
```

```
SQL> SELECT * FROM Enroll_24MCA0242;

CLS_C REG_N
----- -----
ENROLL_TIME                                                                 G
--------------------------------------------------------------------------- -
CLS01 R0001
05-JAN-24 10.00.00 AM                                                       A

CLS02 R0002
08-AUG-21 09.00.00 AM                                                       B

CLS03 R0003
12-JAN-22 11.00.00 AM                                                       S


CLS_C REG_N
----- -----
ENROLL_TIME                                                                 G
--------------------------------------------------------------------------- -
CLS05 R0005
18-JAN-23 12.00.00 PM                                                       D
```

ROLLBACK;

```
SQL> ROLLBACK;

Rollback complete.
```

```
SQL> SELECT * FROM Enroll_24MCA0242;

CLS_C REG_N
----- -----
ENROLL_TIME                                                                 G
--------------------------------------------------------------------------- -
CLS01 R0001
05-JAN-24 10.00.00 AM                                                       A

CLS02 R0002
08-AUG-21 09.00.00 AM                                                       B

CLS03 R0003
12-JAN-22 11.00.00 AM                                                       S


CLS_C REG_N
----- -----
ENROLL_TIME                                                                 G
--------------------------------------------------------------------------- -
CLS04 R0004
10-AUG-22 02.00.00 PM                                                       C

CLS05 R0005
18-JAN-23 12.00.00 PM                                                       D
```

(ix) Create a duplicate of course table.

CREATE TABLE COURSE_DUPLICATE AS SELECT * FROM COURSE_24MCA0242;

```
SQL> CREATE TABLE COURSE_DUPLICATE AS SELECT * FROM COURSE_24MCA0242;

Table created.

SQL>
```

(x) 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;

```
SQL> 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.Pro
f_id;

View created.

SQL>
```

SELECT * FROM Student_Course_Professor_View;

```
SQL> SELECT * FROM Student_Course_Professor_View;

REG_N SNAME                          CRS_C
----- ------------------------------ -----
CRS_NAME
------------------------------------------------
PROF_NAME
------------------------------
R0001 Amit Verma                     CRS01
Data Structures
Dr. Ravi Kumar

R0002 Sunita Desai                   CRS02
Operating Systems
Dr. Neha Sharma

REG_N SNAME                          CRS_C
----- ------------------------------ -----
CRS_NAME
------------------------------------------------
PROF_NAME
------------------------------

R0003 Rahul Singh                    CRS03
Database Management
Dr. Arjun Mehta

R0004 Pooja Reddy                    CRS04
Algorithms

REG_N SNAME                          CRS_C
----- ------------------------------ -----
CRS_NAME
------------------------------------------------
PROF_NAME
------------------------------
Dr. Priya Singh

R0005 Anil Kumar                     CRS05
Artificial Intelligence
Dr. Suresh Gupta
```

(xi) 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;

```
SQL> SELECT Room_no, Slot, Stime, Etime,
  2  EXTRACT(HOUR FROM (Etime - Stime)) || ':' ||
  3  EXTRACT(MINUTE FROM (Etime - Stime)) AS Duration
  4  FROM CLASS_24MCA0242 WHERE Day_of_week = 'Wednesday' ORDER BY Room_no DESC;

ROOM_NO    SLOT
---------- ----------
STIME
---------------------------------------------------------------------------
ETIME
---------------------------------------------------------------------------
DURATION
-------------------------------------------------------------------------------
SMV102     B1
20-AUG-24 11.00.00 AM
20-AUG-24 12.30.00 PM
1:30
```

(xii) 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';
```

```
SQL> SELECT s.Sname, e.Grade, c.Crs_name FROM STUDENT_24MCA0242 s
  2  JOIN ENROLL_24MCA0242 e ON s.Reg_no = e.Reg_no
  3  JOIN CLASS_24MCA0242 cl ON e.Cls_code = cl.Cls_code
  4  JOIN COURSE_24MCA0242 c ON cl.Crs_code = c.Crs_code
  5  JOIN SEMESTER_24MCA0242 sem ON cl.Sem_code = sem.Sem_code
  6  WHERE sem.Sem_code = 'Fall2017-18';

SNAME                          G
------------------------------ -
CRS_NAME
--------------------------------------------------
Amit Verma                     A
Data Structures

Sunita Desai                   B
Operating Systems

Rahul Singh                    C
Database Management
```

(xiii) 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;
```

```
SQL> SELECT s.Sname
  2  FROM STUDENT_24MCA0242 s
  3  JOIN ENROLL_24MCA0242 e ON s.Reg_no = e.Reg_no
  4  JOIN CLASS_24MCA0242 c ON e.Cls_code = c.Cls_code
  5  JOIN COURSE_24MCA0242 cr ON c.Crs_code = cr.Crs_code
  6  JOIN SEMESTER_24MCA0242 sem ON c.Sem_code = sem.Sem_code
  7  WHERE sem.Term = 'Fall'
  8    AND sem.Year = 2016
  9    AND cr.Crs_name IN ('Database Systems', 'Operating Systems')
 10  GROUP BY s.Sname;

SNAME
----------------------------
Sunita Desai
Amit Verma
Rahul Singh

SQL>
```

(xiv) 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');

```
SQL> SELECT s.SNAME FROM STUDENT_24MCA0242 s
  2  JOIN ENROLL_24MCA0242 e ON s.REG_No = e.REG_No
  3  JOIN CLASS_24MCA0242 c ON e.CLS_Code = c.CLS_Code
  4  JOIN COURSE_24MCA0242 cr ON c.CRS_Code = cr.CRS_Code
  5  WHERE c.SEM_CODE = 'Win2017-18' AND cr.CRS_NAME = 'Database Systems' AND s.REG_No NOT IN (
  6      SELECT e.REG_No FROM ENROLL_24MCA0242 e
  7      JOIN CLASS_24MCA0242 c ON e.CLS_Code = c.CLS_Code
  8      JOIN COURSE_24MCA0242 cr ON c.CRS_Code = cr.CRS_Code
  9      WHERE c.SEM_CODE = 'Win2017-18' AND cr.CRS_NAME = 'Operating Systems');

SNAME
----------------------------
Amit Verma

SQL>
```

(xv) 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 ));
```

```
SQL> SELECT s.REG_No, s.SNAME, SUM(cr.Credits) AS TotalCredits
  2  FROM STUDENT_24MCA0242 s
  3  JOIN ENROLL_24MCA0242 e ON s.REG_No = e.REG_No
  4  JOIN CLASS_24MCA0242 c ON e.CLS_Code = c.CLS_Code
  5  JOIN COURSE_24MCA0242 cr ON c.CRS_Code = cr.CRS_Code
  6  WHERE c.SEM_CODE = 'Win2017-18'
  7  GROUP BY s.REG_No, s.SNAME
  8  HAVING SUM(cr.Credits) = (
  9      SELECT MAX(TotalCredits)
 10      FROM (
 11          SELECT SUM(cr.Credits) AS TotalCredits
 12          FROM STUDENT_24MCA0242 s
 13          JOIN ENROLL_24MCA0242 e ON s.REG_No = e.REG_No
 14          JOIN CLASS_24MCA0242 c ON e.CLS_Code = c.CLS_Code
 15          JOIN COURSE_24MCA0242 cr ON c.CRS_Code = cr.CRS_Code
 16          WHERE c.SEM_CODE = 'Win2017-18'
 17          GROUP BY s.REG_No
 18      )
 19  );

REG_N SNAME                          TOTALCREDITS
----- ------------------------------ ------------
R0001 Amit Verma                             17

SQL>
```

(xvi) 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;

```
SQL> 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
  2  JOIN CLASS_24MCA0242 c ON cr.Crs_code = c.Crs_code
  3  JOIN ENROLL_24MCA0242 e ON c.Cls_code = e.Cls_code
  4  JOIN PROFESSOR_24MCA0242 p ON c.Prof_id = p.Prof_id
  5  GROUP BY cr.Crs_name, c.Slot, p.Prof_name ORDER BY cr.Crs_name, c.Slot, p.Prof_name;

COURSENAME                                         SLOT
-------------------------------------------------- ----------
FACULTYNAME                    NUMBEROFSTUDENTS
------------------------------ ----------------
Algorithms                                         D1
Dr. Priya Singh                                 1

Artificial Intelligence                            E1
Dr. Suresh Gupta                                1

Data Structures                                    A1
Dr. Priya Singh                                 3


COURSENAME                                         SLOT
-------------------------------------------------- ----------
FACULTYNAME                    NUMBEROFSTUDENTS
------------------------------ ----------------
Data Structures                                    A1
Dr. Ravi Kumar                                  1

Database Systems                                   A1
Dr. Ravi Kumar                                  1

Database Systems                                   C1
Dr. Arjun Mehta                                 1


COURSENAME                                         SLOT
-------------------------------------------------- ----------
FACULTYNAME                    NUMBEROFSTUDENTS
------------------------------ ----------------
Operating Systems                                  B1
Dr. Neha Sharma                                 2

Operating Systems                                  B1
Dr. Suresh Gupta                                3


8 rows selected.

SQL>
```

(xvii) 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');
```

```
SQL> SELECT s.Sname
  2  FROM STUDENT_24MCA0242 s
  3  JOIN ENROLL_24MCA0242 e ON s.REG_No = e.REG_No
  4  JOIN CLASS_24MCA0242 c ON e.CLS_Code = c.CLS_Code
  5  WHERE c.Crs_code IN (
  6      SELECT c2.Crs_code
  7      FROM CLASS_24MCA0242 c2
  8      JOIN PROFESSOR_24MCA0242 p ON c2.Prof_id = p.Prof_id
  9      WHERE p.Prof_name = 'Prof. O''Brien' AND c2.SEM_CODE = 'Win2017-18'
 10  )
 11  GROUP BY s.Sname
 12  HAVING COUNT(DISTINCT c.Crs_code) = (
 13      SELECT COUNT(*)
 14      FROM CLASS_24MCA0242 c2
 15      JOIN PROFESSOR_24MCA0242 p ON c2.Prof_id = p.Prof_id
 16      WHERE p.Prof_name = 'Prof. O''Brien' AND c2.SEM_CODE = 'Win2017-18'
 17  );

no rows selected
```

(xviii) 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');
```

```
SQL> SELECT e.Reg_no, s.Sname, cr.Crs_name FROM ENROLL_24MCA0242 e
  2  JOIN CLASS_24MCA0242 c ON e.CLS_Code = c.CLS_Code
  3  JOIN COURSE_24MCA0242 cr ON c.CRS_Code = cr.CRS_Code
  4  JOIN STUDENT_24MCA0242 s ON e.Reg_no = s.Reg_no
  5  WHERE cr.Crs_name = 'Database Systems' AND TRUNC(e.Enroll_time) = TO_DATE('2017-11-17', 'YYYY-MM-DD');

REG_N SNAME
----- ------------------------------
CRS_NAME
--------------------------------------------------
R0003 Rahul Singh
Database Systems


SQL>
```

(xix) 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';

```
SQL> SELECT e.Reg_no, s.Sname, cr.Crs_name, e.Grade
  2  FROM ENROLL_24MCA0242 e
  3  JOIN CLASS_24MCA0242 c ON e.Cls_code = c.Cls_code
  4  JOIN COURSE_24MCA0242 cr ON c.Crs_code = cr.Crs_code
  5  JOIN STUDENT_24MCA0242 s ON e.Reg_no = s.Reg_no
  6  JOIN SEMESTER_24MCA0242 sem ON c.Sem_code = sem.Sem_code
  7  WHERE s.Reg_no = 'R0005'
  8    AND cr.Crs_name = 'Artificial Intelligence'
  9    AND sem.Sem_code = 'Fall2024-25';

REG_N SNAME
----- ------------------------------
CRS_NAME                                           G
-------------------------------------------------- -
R0005 Anil Kumar
Artificial Intelligence                            D


SQL>
```

(xx) 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;

```
SQL> SELECT d.Dname AS Department_Name, p.Prof_name AS Professor_Name
  2  FROM DEPARTMENT_24MCA0242 d
  3  JOIN PROFESSOR_24MCA0242 p ON d.Prof_id = p.Prof_id;

DEPARTMENT_NAME
--------------------------------------------------
PROFESSOR_NAME
------------------------------
Computer Science
Dr. Ravi Kumar

Mechanical Engineering
Dr. Neha Sharma

Physics
Dr. Arjun Mehta


DEPARTMENT_NAME
--------------------------------------------------
PROFESSOR_NAME
------------------------------
Electrical Engineering
Dr. Priya Singh

Civil Engineering
Dr. Suresh Gupta

Medicine
Dr. Ramesh Iyer


DEPARTMENT_NAME
--------------------------------------------------
PROFESSOR_NAME
------------------------------
Cybersecurity
Dr. Ramesh Kumar


7 rows selected.

SQL>
```

(xxi) 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;

```
SQL> SELECT s.Scl_name AS School_Name
  2  FROM SCHOOL_24MCA0242 s
  3  JOIN STUDENT_24MCA0242 st ON s.Prof_id = st.Prof_id
  4  GROUP BY s.Scl_name
  5  HAVING COUNT(st.Reg_no) > 7000;

no rows selected

SQL>
```

(xxii) 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);

```
SQL> SELECT sc.Scl_name, d.Dname
  2  FROM DEPARTMENT_24MCA0242 d
  3  JOIN SCHOOL_24MCA0242 sc ON d.SCode = sc.SCode
  4  JOIN STUDENT_24MCA0242 s ON d.Dept_id = s.Dept_id
  5  WHERE sc.Scl_name = 'School of Medicine'
  6  GROUP BY sc.Scl_name, d.Dname
  7  HAVING COUNT(s.Reg_no) > (
  8      SELECT AVG(Student_Count)
  9      FROM (
 10          SELECT d.Dept_id, COUNT(s.Reg_no) AS Student_Count
 11          FROM DEPARTMENT_24MCA0242 d
 12          JOIN SCHOOL_24MCA0242 sc ON d.SCode = sc.SCode
 13          JOIN STUDENT_24MCA0242 s ON d.Dept_id = s.Dept_id
 14          WHERE sc.Scl_name = 'School of Medicine'
 15          GROUP BY d.Dept_id
 16      ) dept_counts
 17  );

SCL_NAME           DNAME
------------------ ------------------------------
School of Medicine   Department of Internal Meds

SQL>
```

(xxiii) 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;

```
SQL> SELECT e.Reg_no, SUM(cr.Credits) AS Total_Credits FROM ENROLL_24MCA0242 e
  2  JOIN CLASS_24MCA0242 c ON e.Cls_code = c.Cls_code
  3  JOIN COURSE_24MCA0242 cr ON c.Crs_code = cr.Crs_code
  4  JOIN SEMESTER_24MCA0242 s ON c.Sem_code = s.Sem_code
  5  WHERE e.Reg_no = 'R0001' AND s.Term = 'Winter' AND s.Year = 2017 GROUP BY e.Reg_no;

REG_N TOTAL_CREDITS
----- -------------
R0001             5

SQL>
```

(xxiv) 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';

```
SQL> SELECT e.Reg_no, e.Grade, c.Crs_code, s.Term AS Semester, s.Year AS Academic_Year FROM ENROLL_24MCA0242 e
  2  JOIN CLASS_24MCA0242 c ON e.Cls_code = c.Cls_code
  3  JOIN SEMESTER_24MCA0242 s ON c.Sem_code = s.Sem_code
  4  WHERE s.Term = 'Fall' AND s.Year = 2017 AND e.Reg_no = 'R0003';

REG_N G CRS_C SEMESTER   ACADEMIC_YEAR
----- - ----- ---------- -------------
R0003 C CRS03 Fall               2017

SQL>
```

(xxv) 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;

```
SQL> SELECT DISTINCT cr.Crs_name FROM COURSE_24MCA0242 cr
  2  LEFT JOIN CLASS_24MCA0242 c ON cr.Crs_code = c.Crs_code
  3  LEFT JOIN SEMESTER_24MCA0242 s ON c.Sem_code = s.Sem_code
  4  WHERE s.Term IS NULL OR s.Term <> 'Winter' OR s.Year <> 2017;

CRS_NAME
--------------------------------------------------
Data Structures
Algorithms
Artificial Intelligence
Operating Systems
Database Systems

SQL>
```

(xxvi) 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);

```
SQL> UPDATE CLASS_24MCA0242
  2  SET Stime = Stime + INTERVAL '10' MINUTE,
  3      Etime = Etime + INTERVAL '10' MINUTE
  4  WHERE Sem_code = (
  5      SELECT Sem_code
  6      FROM SEMESTER_24MCA0242
  7      WHERE Term = 'Fall'
  8        AND Year = 2017
  9  );

5 rows updated.

SQL>
```

(xxvii) 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;

```
SQL>
SQL> UPDATE SEMESTER_24MCA0242
  2  SET Sdate = (
  3      SELECT Sdate + INTERVAL '7' DAY
  4      FROM SEMESTER_24MCA0242
  5      WHERE Term = 'Fall'
  6        AND Year = 2023
  7  ), Edate = (
  8      SELECT Edate + INTERVAL '7' DAY
  9      FROM SEMESTER_24MCA0242
 10      WHERE Term = 'Fall'
 11        AND Year = 2023
 12  )
 13  WHERE Term = 'Fall'
 14    AND Year = 2024;

1 row updated.
```

SELECT Sem_code, Term, Year, Sdate, Edate FROM SEMESTER_24MCA0242 WHERE Term = 'Fall' AND Year = 2024;

```
SQL> SELECT Sem_code, Term, Year, Sdate, Edate FROM SEMESTER_24MCA0242 WHERE Term = 'Fall' AND Year =
 2024;

SEM_CODE        TERM            YEAR SDATE     EDATE
--------------- ---------- ---------- --------- ---------
Fall2024-25     Fall             2024 22-JUL-23 30-DEC-23

SQL>
```

(xxviii) 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;

```
SQL> SELECT s.Sname
  2  FROM STUDENT_24MCA0242 s
  3  JOIN ENROLL_24MCA0242 e ON s.Reg_no = e.Reg_no
  4  GROUP BY s.Sname, s.Reg_no
  5  HAVING COUNT(CASE WHEN e.Grade = 'S' THEN 1 END) * 100.0 / COUNT(e.Grade) >= 50;

SNAME
----------------------------
Amit Verma
Rahul Singh
Sanjay Patil

SQL>
```

(xxix) 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;

```
SQL> SELECT DISTINCT c.Slot, c.Day_of_week
  2  FROM CLASS_24MCA0242 c
  3  LEFT JOIN ENROLL_24MCA0242 e
  4       ON c.Cls_code = e.Cls_code
  5       AND e.Reg_no = 'R0001'
  6  WHERE e.Cls_code IS NULL;

SLOT       DAY_OF_WEE
---------- ----------
B1         Tuesday
B2         Friday
A2         Thursday
H1         Wednesday
D1         Thursday
G1         Tuesday
C1         Wednesday
E1         Friday

8 rows selected.
```

(xxx) 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);

```
SQL> SELECT s.Sname FROM STUDENT_24MCA0242 s
  2  JOIN ENROLL_24MCA0242 e ON s.Reg_no = e.Reg_no
  3  JOIN CLASS_24MCA0242 c ON e.Cls_code = c.Cls_code
  4  WHERE c.Day_of_week = 'Monday' AND c.Stime >= TO_TIMESTAMP('12:00:00', 'HH24:MI:SS')
  5  GROUP BY s.Sname HAVING COUNT(DISTINCT c.Day_of_week) = 1
  6  AND SUM(CASE WHEN c.Stime >= TO_TIMESTAMP('12:00:00', 'HH24:MI:SS') THEN 1 ELSE 0 END) = COUNT(c.
Day_of_week);

SNAME
------------------------------
Malini Rao
Kalpana Roy
Meenal Patel

SQL>
```

(xxxi) 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;

```
SQL> ALTER TABLE CLASS_24MCA0242 ADD Duration INTERVAL DAY(0) TO SECOND(0);

Table altered.

SQL> UPDATE CLASS_24MCA0242 SET Duration = Etime - Stime;

25 rows updated.
```

SELECT Cls_code, Slot, Stime, Etime, Duration FROM CLASS_24MCA0242;

```
SQL> SELECT Cls_code, Slot, Stime, Etime, Duration FROM CLASS_24MCA0242;

CLS_C SLOT
----- ----------
STIME
---------------------------------------------------------------------------
ETIME
---------------------------------------------------------------------------
DURATION
---------------------------------------------------------------------------
CLS23 A1
01-SEP-24 01.00.00 PM
01-SEP-24 02.30.00 PM
+0 01:30:00


CLS_C SLOT
----- ----------
STIME
---------------------------------------------------------------------------
ETIME
---------------------------------------------------------------------------
DURATION
---------------------------------------------------------------------------
CLS24 A1
01-SEP-24 01.00.00 PM
01-SEP-24 02.30.00 PM
+0 01:30:00


CLS_C SLOT
----- ----------
STIME
---------------------------------------------------------------------------
ETIME
---------------------------------------------------------------------------
DURATION
---------------------------------------------------------------------------
CLS25 A1
01-SEP-24 01.00.00 PM
01-SEP-24 02.30.00 PM
+0 01:30:00
```

```
CLS_C SLOT
----- ----------
STIME
--------------------------------------------------------------------------------
ETIME
--------------------------------------------------------------------------------
DURATION
--------------------------------------------------------------------------------
CLS21 A1
20-JAN-18 09.00.00 AM
20-JAN-18 10.30.00 AM
+0 01:30:00


CLS_C SLOT
----- ----------
STIME
--------------------------------------------------------------------------------
ETIME
--------------------------------------------------------------------------------
DURATION
--------------------------------------------------------------------------------
CLS22 B1
21-JAN-18 11.00.00 AM
21-JAN-18 12.30.00 PM
+0 01:30:00


25 rows selected.

SQL>
```

(xxxii) 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;

```
SQL> ALTER TABLE SEMESTER_24MCA0242 ADD SemesterDuration NUMBER;

Table altered.

SQL> UPDATE SEMESTER_24MCA0242 SET SemesterDuration = Edate - Sdate;

8 rows updated.

SQL> SELECT Sem_code, Term, Year, Sdate, Edate, SemesterDuration FROM SEMESTER_24MCA0242;

SEM_CODE        TERM            YEAR SDATE     EDATE     SEMESTERDURATION
--------------- ---------- ---------- --------- --------- ----------------
Fall2024-25     Fall            2024 22-JUL-23 30-DEC-23              161
Win2024-25      Winter          2024 23-DEC-24 25-MAY-25              153
Fall2023-24     Fall            2023 15-JUL-23 23-DEC-23              161
Win2023-24      Winter          2023 27-DEC-23 20-MAY-24              145
Fall2017-18     Fall            2017 10-AUG-17 20-DEC-17              132
Win2017-18      Winter          2017 20-JAN-18 25-MAY-18              125
Fall2016-17     Fall            2016 05-AUG-16 15-DEC-16              132
Win2016-17      Winter          2016 18-JAN-17 25-MAY-17              127

8 rows selected.

SQL>
```

(xxxiii) 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';
```

```
SQL> SELECT s.Reg_no, s.Sname, p.Prog_name FROM STUDENT_24MCA0242 s
  2  JOIN PROGRAMME_24MCA0242 p ON s.Dept_id = p.Dept_id WHERE p.Prog_name = 'MCA';

REG_N SNAME                          PROG_NAME
----- ------------------------------ ------------------------------
R0001 Amit Verma                     MCA
R0012 John Doe                       MCA

SQL> S
```

(xxxiv) 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;
```

```
SQL> SELECT p.Prog_name, s.Scl_name
  2  FROM PROGRAMME_24MCA0242 p
  3  JOIN DEPARTMENT_24MCA0242 d ON p.Dept_id = d.Dept_id
  4  JOIN SCHOOL_24MCA0242 s ON d.SCode = s.SCode;

PROG_NAME                      SCL_NAME
------------------------------ --------------------
B.Tech CSE                     School of Science
B.Sc Physics                   School of Science
B.Sc Data Science              School of Science
BCA                            School of Science
MCA                            School of Science
```

(xxxv) 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;
```

```
SQL> 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;

DEPARTMENT_NAME               PROGRAM_NAME
----------------------------- -----------------------------
Computer Science              B.Tech CSE
Mechanical Engineering        B.Tech ME
Physics                       B.Sc Physics
Electrical Engineering        B.Tech EE
Civil Engineering             B.Tech CE
Computer Science              B.Sc Data Science
Medicine                      B.Sc Mathematics
Electrical Engineering        M.Tech Electrical Engineering
Civil Engineering             M.Tech Civil Engineering
Cybersecurity                 B.Tech Cybersecurity
Mechanical Engineering        B.Tech Mechanical Engineering

DEPARTMENT_NAME               PROGRAM_NAME
----------------------------- -----------------------------
Civil Engineering             M.Tech Cloud Computing
Computer Science              BCA
Computer Science              MCA

14 rows selected.

SQL>
```

(xxxvi) 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;
```

```
SCL_NAME            STUDENT_COUNT
------------------- -------------
School of Science               4

SQL> SELECT Scl_name
  2  FROM (
  3      SELECT s.Scl_name, COUNT(st.Reg_no) AS student_count
  4      FROM SCHOOL_24MCA0242 s
  5      JOIN DEPARTMENT_24MCA0242 d ON s.SCode = d.SCode
  6      JOIN STUDENT_24MCA0242 st ON d.Dept_id = st.Dept_id
  7      GROUP BY s.Scl_name
  8      ORDER BY student_count DESC)
  9  WHERE ROWNUM = 1;

SCL_NAME
--------------------
School of Science

SQL>
```