Online Academic Platform and College Management System

ASSIGNMENT 3

SECTION B

Faizan Pervaz , Haider Yar
20I-0565 | 20I-0879

REPORT:

The following report contains,

1. Explanation of ERD (relationships and cardinality ratios, participation, and constraints)

2. Relational Database Schema

3. SQL Queries and their Results

1. Explanation of ERD

1.1 Entities:

1. COLLEGE (Represents the academic institutions offering courses).
2. DEPARTMENT (Represents the academic departments within a college).
3. FACULTY (Represents the teaching staff associated with a department).
4. ADMINISTRATOR (Represents administrators overseeing college management).
5. TIMESLOT (Represents time slots for scheduling courses).
6. COURSE (Represents academic courses offered, scheduled, and associated with departments, faculties, and time slots).
7. PROGRAM (Represents academic programs associated with departments).
8. STAFF (Represents non-teaching staff associated with academic programs).
9. STUDENT (Represents students enrolled in departments).
10. ENROLLMENT (Represents the enrollment of students in programs and courses).

1.2 Relationships Between Entities:

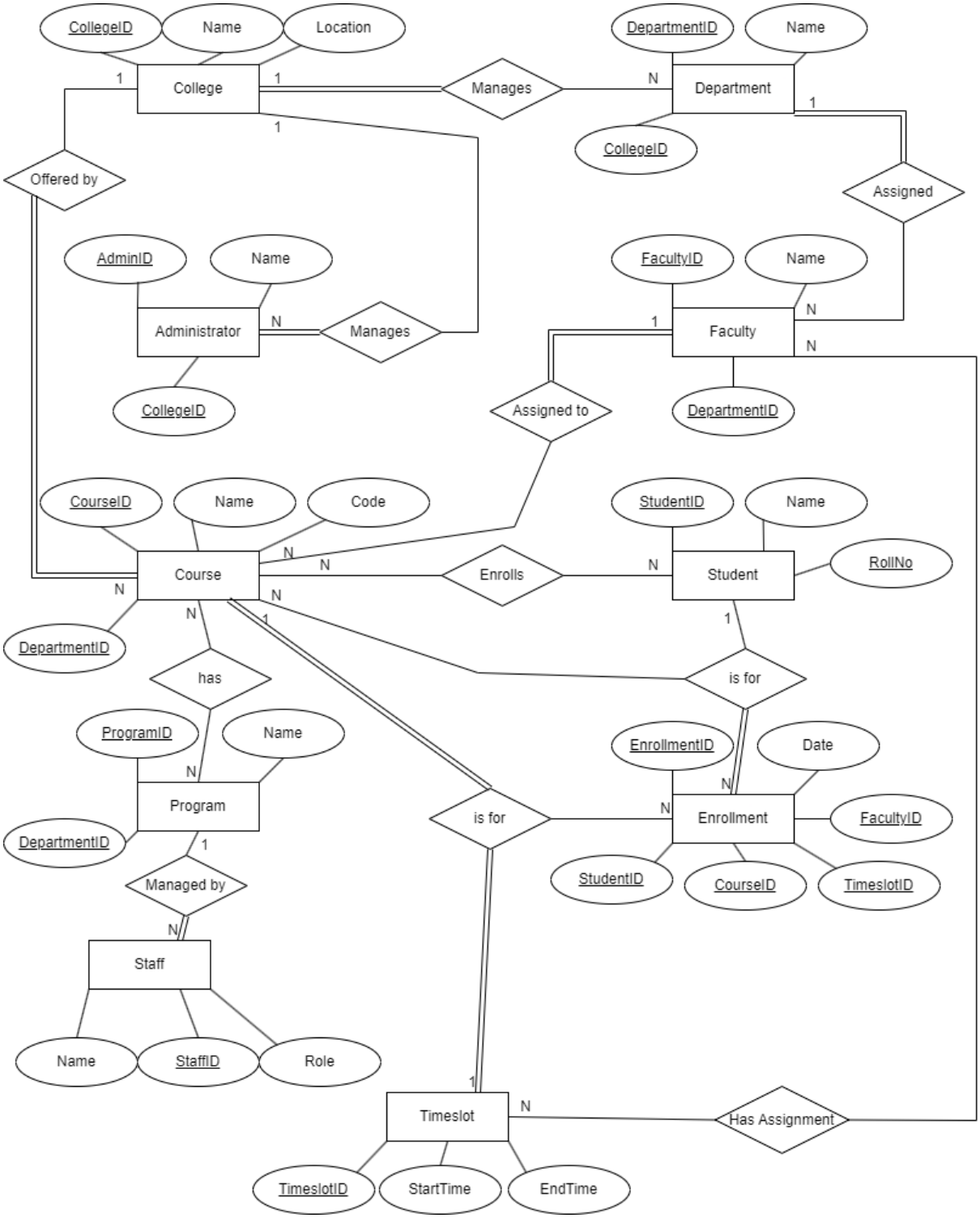
1. One-to-Many: A college can have multiple departments, but a department belongs to only one college.
2. One-to-Many: A department can have multiple faculties, but a faculty is associated with only one department.
3. One-to-Many: A college can have multiple administrators, but an administrator belongs to only one college.
4. Many-to-Many: A course can have multiple faculties, and a faculty can teach multiple courses.
5. Many-to-One: A course is associated with one time slot, but a time slot can be associated with multiple courses.
6. Many-to-One: A course belongs to one department, but a department can have multiple courses.
7. One-to-Many: A department can offer multiple programs, but a program belongs to only one department.
8. Many-to-Many: Programs can have multiple courses, and courses can belong to multiple programs.
9. One-to-Many: A staff member can be associated with one program, but a program can have multiple staff members.

10. Many-to-One: A student belongs to one department, but a department can have multiple students.
11. Many-to-Many: Students can enroll in multiple courses, and courses can have multiple enrolled students.
12. Many-to-Many: Faculties can be assigned to multiple courses, and a course can have multiple assigned faculties.
13. Many-to-One: An enrollment is associated with one student, one course, and one program.

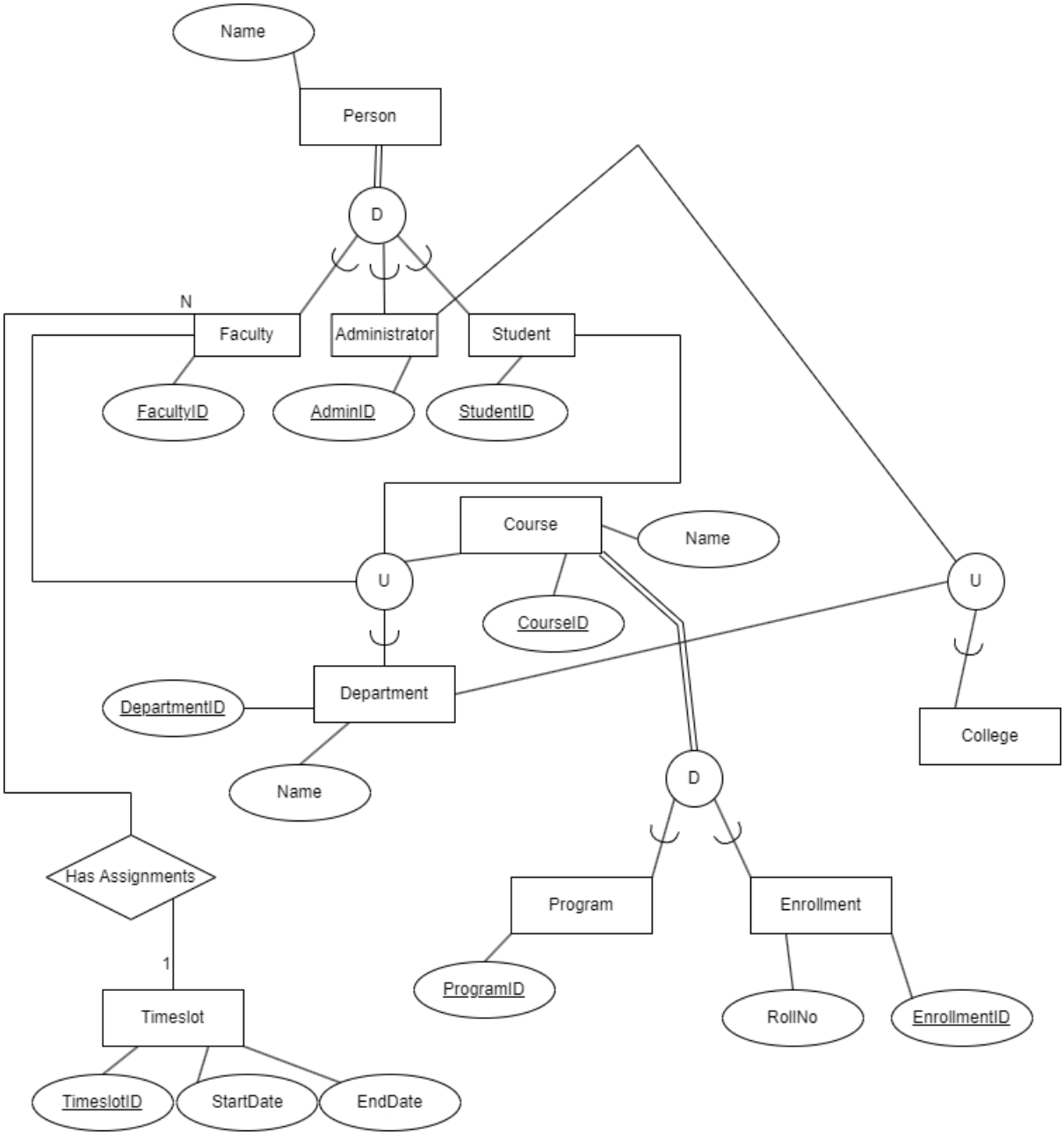
1.3 Participation of Entities in Relationship:

1. Total Participation:
   In most relationships, all entities have total participation, ensuring that every entity in the relationship participates in the relationship.
2. Partial Participation:
   In certain relationships like ENROLLS and ASSIGNED, partial participation is allowed, indicating that not every entity is required to participate in the relationship.

ERD DIAGRAM


CollegeID
Name
Location
College
Manages
DepartmentID
Name
Department
CollegeID
Offered by
Assigned
AdminID
Name
Administrator
Manages
FacultyID
Name
Faculty
CollegeID
Assigned to
DepartmentID
CourseID
Name
Code
Course
Enrolls
StudentID
Name
Student
RollNo
DepartmentID
has
is for
ProgramID
Name
Program
DepartmentID
Managed by
Staff
Name
StaffID
Role
is for
EnrollmentID
Date
Enrollment
FacultyID
StudentID
CourseID
TimeslotID
Timeslot
Has Assignment
TimeslotID
StartTime
EndTime

EERD DIAGRAM :



1.4 Constraints:

1.4.1 Domain Constraints:

Different domains have been rightly defined like varchar (255), date, numeric (20) for all types of attributes.

Attributes such as StartTime and EndTime in the TIMESLOT entity have domain constraints ensuring valid time values.

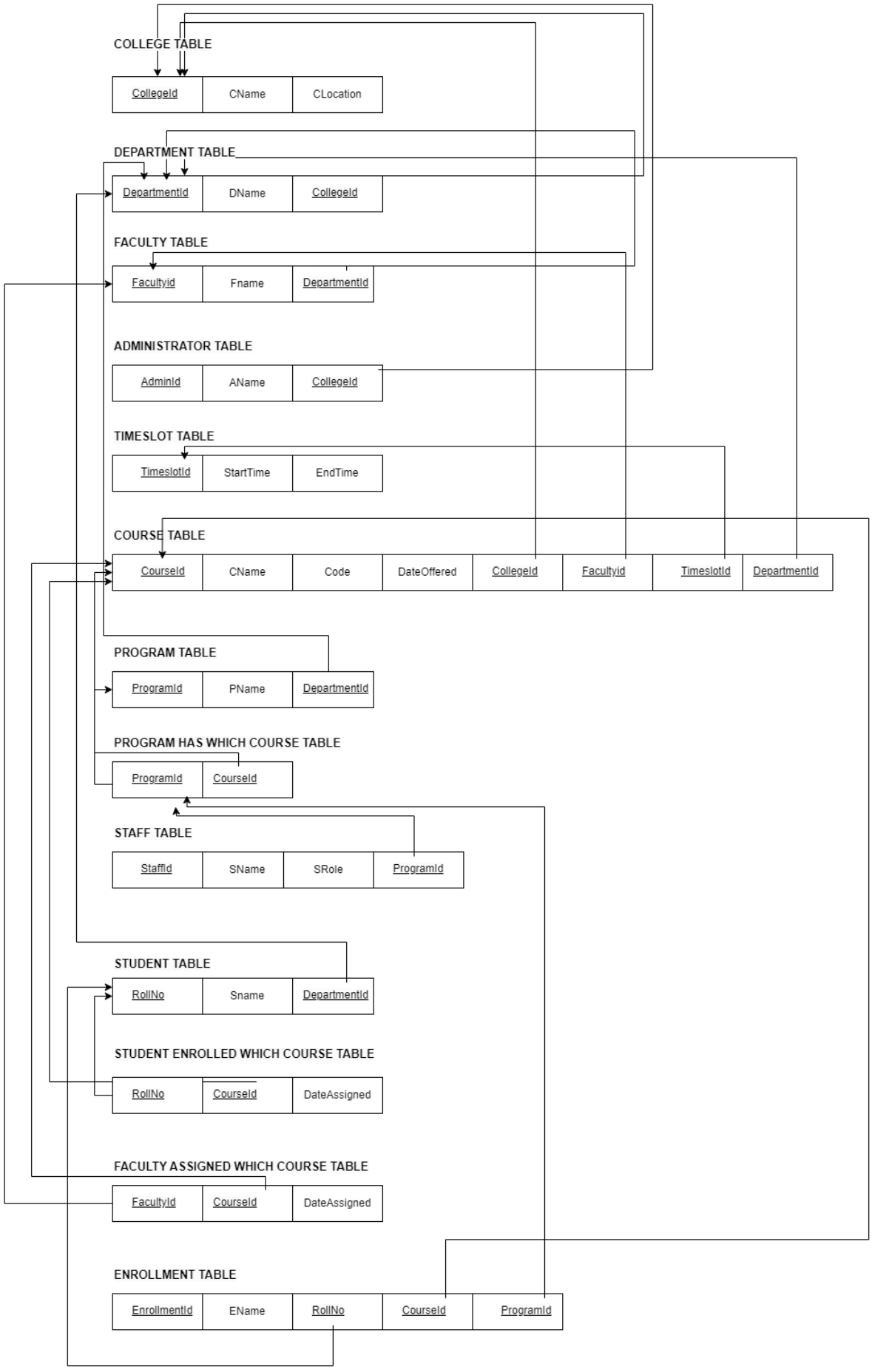
1.4.2 Key Constraints:

Primary keys are enforced for each entity to ensure entity uniqueness.

1.4.3 Referential Integrity Constraints:

Foreign keys have been defined for all relations and the on cascade delete and on cascade update is also applied accordingly.

2.0 Relational Database Schema:


COLLEGE TABLE
CollegeId
CName
CLocation
DEPARTMENT TABLE
DepartmentId
DName
CollegeId
FACULTY TABLE
Facultyid
Fname
DepartmentId
ADMINISTRATOR TABLE
AdminId
AName
CollegeId
TIMESLOT TABLE
TimeslotId
StartTime
EndTime
COURSE TABLE
CourseId
CName
Code
DateOffered
CollegeId
Facultyid
TimeslotId
DepartmentId
PROGRAM TABLE
ProgramId
PName
DepartmentId
PROGRAM HAS WHICH COURSE TABLE
ProgramId
CourseId
STAFF TABLE
StaffId
SName
SRole
ProgramId
STUDENT TABLE
RollNo
Sname
DepartmentId
STUDENT ENROLLED WHICH COURSE TABLE
RollNo
CourseId
DateAssigned
FACULTY ASSIGNED WHICH COURSE TABLE
FacultyId
CourseId
DateAssigned
ENROLLMENT TABLE
EnrollmentId
EName
RollNo
CourseId
ProgramId

3.1 Tuples:

COLLEGE

	CollegeId	CName	CLocation
1	1	College Edwrd	Location1
2	2	College2	Location2
3	3	College3	Location3
4	4	College4	Location4
5	5	College5	Location5
6	6	College6	Location6
7	7	College7	Location7
8	8	College8	Location8

FACULTY

	FacultyId	FName	DepartmentId
1	1	Faculty1	1
2	2	Faculty2	2
3	3	Faculty3	3
4	4	Faculty4	4
5	5	Faculty5	5
6	6	Faculty6	6
7	7	Faculty7	7
8	8	Faculty8	8
9	9	Faculty9	9
10	10	Faculty10	10

DEPARTMENT

	DepartmentId	DName	CollegeId
1	1	Engineering	1
2	2	Business	1
3	3	Arts	2
4	4	Science	2
5	5	Social Sciences	3
6	6	History	4
7	7	Chemistry	5
8	8	Economics	6
9	9	Political Science	7
10	10	Psychology	8

ADMINISTRATOR

	AdminId	AName	CollegeId
1	1	Admin1	1
2	2	Admin2	2
3	3	Admin3	3
4	4	Admin4	4
5	5	Admin5	5
6	6	Admin6	6
7	7	Admin7	7
8	8	Admin8	8

TIMESLOT

	TimeslotId	StartTime	EndTime
1	1	08:00:00.0000000	10:00:00.0000000
2	2	10:00:00.0000000	12:00:00.0000000
3	3	13:00:00.0000000	15:00:00.0000000
4	4	15:00:00.0000000	17:00:00.0000000
5	5	17:30:00.0000000	19:00:00.0000000
6	6	19:30:00.0000000	21:00:00.0000000
7	7	21:30:00.0000000	23:00:00.0000000
8	8	23:30:00.0000000	00:00:00.0000000

COURSE

	CourseId	CName	Code	DateOffered	CollegeId	FacultyId	TimeslotId	DepartmentId
1	1	Course1	101	2023-11-01	1	1	1	1
2	2	Course2	102	2023-11-15	1	2	2	2
3	3	Course3	103	2023-11-11	2	3	3	3
4	4	Course4	104	2023-11-12	2	4	4	4
5	5	Course5	105	2023-11-15	3	5	1	5
6	6	Course6	106	2023-11-20	4	6	5	6
7	7	Course7	107	2023-11-22	5	7	6	7
8	8	Course8	108	2023-11-25	6	8	7	8
9	9	Course9	109	2023-11-28	7	9	8	9
10	10	Course10	110	2023-11-30	8	10	1	10

PROGRAM

	ProgramId	PName	DepartmentId
1	1	Program1	1
2	2	Program2	2
3	3	Program3	3
4	4	Program4	4
5	5	Program5	5
6	6	Program6	6
7	7	Program7	7
8	8	Program8	8
9	9	Program9	9
10	10	Program10	10

PROGRAM HAS COURSES

	ProgramId	CourseId
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10

STAFF

	StaffId	SName	SRole	ProgramId
1	1	Staff1	Role1	1
2	2	Staff2	Role2	2
3	3	Staff3	Role3	3
4	4	Staff4	Role4	4
5	5	Staff5	Role5	5
6	6	Staff6	Role6	6
7	7	Staff7	Role7	7
8	8	Staff8	Role8	8
9	9	Staff9	Role9	9
10	10	Staff10	Role10	10

STUDENT

	RollNo	SName	DepartmentId
1	101	Student1	1
2	102	Student2	2
3	103	Student3	3
4	104	Student4	4
5	105	Student5	5
6	106	Student6	6
7	107	Student7	7
8	108	Student8	8
9	109	Student9	9
10	110	Student10	10

STUDENT ENROLLS IN COURSE

	RollNo	CourseId	DateEnrolled
1	101	6	2023-11-02
2	101	7	2023-11-03
3	102	8	2023-11-15
4	102	9	2023-11-16
5	103	10	2023-11-11
6	103	6	2023-11-12
7	104	7	2023-11-12
8	104	8	2023-11-13
9	105	9	2023-11-15
10	105	10	2023-11-16
11	101	1	2023-11-02
12	101	1	2023-11-03
13	102	2	2023-11-15
14	102	2	2023-11-16
15	103	3	2023-11-11
16	103	3	2023-11-12
17	104	4	2023-11-12
18	104	4	2023-11-13
19	105	5	2023-11-15
20	105	5	2023-11-16

FACULTY ASSIGNED TO COURSE

	FacultyId	CourseId	DateAssigned
1	1	1	2023-11-01
2	2	2	2023-11-02
3	3	3	2023-11-11
4	4	4	2023-11-12
5	5	5	2023-11-15
6	6	6	2023-11-20
7	7	7	2023-11-22
8	8	8	2023-11-25
9	9	9	2023-11-28
10	10	10	2023-11-30
11	1	6	2023-11-01
12	2	7	2023-11-02
13	3	8	2023-11-11
14	4	9	2023-11-12
15	5	10	2023-11-15

ENROLLMENT

	EnrollmentId	EName	RollNo	CourseId	ProgramId
1	1	Enrollment1	101	1	1
2	2	Enrollment2	102	2	2
3	3	Enrollment3	103	3	3
4	4	Enrollment4	104	4	4
5	5	Enrollment5	105	5	5
6	6	Enrollment6	106	6	6
7	7	Enrollment7	107	7	7
8	8	Enrollment8	108	8	8
9	9	Enrollment9	109	9	9
10	10	Enrollment10	110	10	10

3.2 Tables Creation

```sql
CREATE TABLE COLLEGE (
    CollegeId INT PRIMARY KEY,
    CName VARCHAR(255),
    CLocation VARCHAR(255)
);

CREATE TABLE DEPARTMENT (
    DepartmentId INT PRIMARY KEY,
    DName VARCHAR(255),
    CollegeId INT,
    FOREIGN KEY (CollegeId) REFERENCES COLLEGE(CollegeId)
);

CREATE TABLE FACULTY (
    FacultyId INT PRIMARY KEY,
    FName VARCHAR(255),
    DepartmentId INT,
    FOREIGN KEY (DepartmentId) REFERENCES DEPARTMENT(DepartmentId)
);

CREATE TABLE ADMINISTRATOR (
    AdminId INT PRIMARY KEY,
    AName VARCHAR(255),
    CollegeId INT,
    FOREIGN KEY (CollegeId) REFERENCES COLLEGE(CollegeId)
);

CREATE TABLE TIMESLOT (
    TimeslotId INT PRIMARY KEY,
    StartTime TIME,
    EndTime TIME
);

CREATE TABLE COURSE (
    CourseId INT PRIMARY KEY,
    CName VARCHAR(255),
    Code INT,
    DateOffered DATE,
    CollegeId INT,
    FacultyId INT,
    TimeslotId INT,
    DepartmentId INT,
    FOREIGN KEY (DepartmentId) REFERENCES DEPARTMENT(DepartmentId),
    FOREIGN KEY (FacultyId) REFERENCES FACULTY(FacultyId),
    FOREIGN KEY (CollegeId) REFERENCES COLLEGE(CollegeId),
    FOREIGN KEY (TimeslotId) REFERENCES Timeslot(TimeslotId)
);
```

```sql
CREATE TABLE PROGRAM (
    ProgramId INT PRIMARY KEY,
    PName VARCHAR(255),
    DepartmentId INT,
    FOREIGN KEY (DepartmentId) REFERENCES DEPARTMENT(DepartmentId)
);

CREATE TABLE HAS (
    ProgramId INT,
    CourseId INT,
    FOREIGN KEY (ProgramId) REFERENCES PROGRAM(ProgramId),
    FOREIGN KEY (CourseId) REFERENCES COURSE(CourseId)
);

CREATE TABLE STAFF (
    StaffId INT PRIMARY KEY,
    SName VARCHAR(255),
    SRole VARCHAR(255),
    ProgramId INT,
    FOREIGN KEY (ProgramId) REFERENCES PROGRAM(ProgramId)
);

CREATE TABLE STUDENT (
    Rollno INT PRIMARY KEY,
    SName VARCHAR(255),
    DepartmentId INT,
    FOREIGN KEY (DepartmentId) REFERENCES DEPARTMENT(DepartmentId)
);

CREATE TABLE ENROLLS (
    Rollno INT,
    CourseId INT,
    DateEnrolled DATE,
    FOREIGN KEY (Rollno) REFERENCES STUDENT(Rollno),
    FOREIGN KEY (CourseId) REFERENCES COURSE(CourseId)
);

CREATE TABLE ASSIGNED (
    FacultyId INT,
    CourseId INT,
    DateAssigned DATE,
    FOREIGN KEY (FacultyId) REFERENCES FACULTY(FacultyId),
    FOREIGN KEY (CourseId) REFERENCES COURSE(CourseId)
);

CREATE TABLE ENROLLMENT (
    EnrollmentId INT PRIMARY KEY,
    EName varchar(255),
    Rollno INT,
    CourseId INT,
    ProgramId INT,
    FOREIGN KEY (Rollno) REFERENCES STUDENT(Rollno),
    FOREIGN KEY (CourseId) REFERENCES COURSE(CourseId),
    FOREIGN KEY (ProgramId) REFERENCES PROGRAM(ProgramId)
);
```

3.3 Insertion Queries

```sql
-- COLLEGE Table
INSERT INTO COLLEGE (CollegeId, CName, CLocation) VALUES
(1, 'College Edwrd', 'Location1'),
(2, 'College2', 'Location2'),
(3, 'College3', 'Location3'),
(4, 'College4', 'Location4'),
(5, 'College5', 'Location5'),
(6, 'College6', 'Location6'),
(7, 'College7', 'Location7'),
(8, 'College8', 'Location8');

-- DEPARTMENT Table
INSERT INTO DEPARTMENT (DepartmentId, DName, CollegeId) VALUES
(1, 'Engineering', 1),
(2, 'Business', 1),
(3, 'Arts', 2),
(4, 'Science', 2),
(5, 'Social Sciences', 3),
(6, 'History', 4),
(7, 'Chemistry', 5),
(8, 'Economics', 6),
(9, 'Political Science', 7),
(10, 'Psychology', 8);


-- FACULTY Table
INSERT INTO FACULTY (FacultyId, FName, DepartmentId) VALUES
(1, 'Faculty1', 1),
(2, 'Faculty2', 2),
(3, 'Faculty3', 3),
(4, 'Faculty4', 4),
(5, 'Faculty5', 5),
(6, 'Faculty6', 6),
(7, 'Faculty7', 7),
(8, 'Faculty8', 8),
(9, 'Faculty9', 9),
(10, 'Faculty10', 10);

-- ADMINISTRATOR Table
INSERT INTO ADMINISTRATOR (AdminId, AName, CollegeId) VALUES
(1, 'Admin1', 1),
(2, 'Admin2', 2),
(3, 'Admin3', 3),
(4, 'Admin4', 4),
(5, 'Admin5', 5),
(6, 'Admin6', 6),
(7, 'Admin7', 7),
(8, 'Admin8', 8);

-- TIMESLOT Table
INSERT INTO TIMESLOT (TimeslotId, StartTime, EndTime) VALUES
(1, '08:00:00', '10:00:00'),
(2, '10:00:00', '12:00:00'),
(3, '13:00:00', '15:00:00'),
(4, '15:00:00', '17:00:00'),
(5, '17:30:00', '19:00:00'),
(6, '19:30:00', '21:00:00'),
(7, '21:30:00', '23:00:00'),
(8, '23:30:00', '00:00:00');
```

```sql
-- COURSE Table
INSERT INTO COURSE (CourseId, CName, Code, DateOffered, CollegeId, FacultyId, TimeslotId, DepartmentId) VALUES
(1, 'Course1', 101, '2023-11-01', 1, 1, 1, 1),
(2, 'Course2', 102, '2023-11-15', 1, 2, 2, 2),
(3, 'Course3', 103, '2023-11-11', 2, 3, 3, 3),
(4, 'Course4', 104, '2023-11-12', 2, 4, 4, 4),
(5, 'Course5', 105, '2023-11-15', 3, 5, 1, 5),
(6, 'Course6', 106, '2023-11-20', 4, 6, 5, 6),
(7, 'Course7', 107, '2023-11-22', 5, 7, 6, 7),
(8, 'Course8', 108, '2023-11-25', 6, 8, 7, 8),
(9, 'Course9', 109, '2023-11-28', 7, 9, 8, 9),
(10, 'Course10', 110, '2023-11-30', 8, 10, 1, 10);

-- PROGRAM Table
INSERT INTO PROGRAM (ProgramId, PName, DepartmentId) VALUES
(1, 'Program1', 1),
(2, 'Program2', 2),
(3, 'Program3', 3),
(4, 'Program4', 4),
(5, 'Program5', 5),
(6, 'Program6', 6),
(7, 'Program7', 7),
(8, 'Program8', 8),
(9, 'Program9', 9),
(10, 'Program10', 10);


-- HAS Table
INSERT INTO HAS (ProgramId, CourseId) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10);

-- STAFF Table
INSERT INTO STAFF (StaffId, SName, SRole, ProgramId) VALUES
(1, 'Staff1', 'Role1', 1),
(2, 'Staff2', 'Role2', 2),
(3, 'Staff3', 'Role3', 3),
(4, 'Staff4', 'Role4', 4),
(5, 'Staff5', 'Role5', 5),
(6, 'Staff6', 'Role6', 6),
(7, 'Staff7', 'Role7', 7),
(8, 'Staff8', 'Role8', 8),
(9, 'Staff9', 'Role9', 9),
(10, 'Staff10', 'Role10', 10);
```

```sql
-- STUDENT Table
INSERT INTO STUDENT (Rollno, SName, DepartmentId) VALUES
(101, 'Student1', 1),
(102, 'Student2', 2),
(103, 'Student3', 3),
(104, 'Student4', 4),
(105, 'Student5', 5),
(106, 'Student6', 6),
(107, 'Student7', 7),
(108, 'Student8', 8),
(109, 'Student9', 9),
(110, 'Student10', 10);

-- ENROLLS Table
INSERT INTO ENROLLS (Rollno, CourseId, DateEnrolled) VALUES
(101, 6, '2023-11-02'),
(101, 7, '2023-11-03'),
(102, 8, '2023-11-15'),
(102, 9, '2023-11-16'),
(103, 10, '2023-11-11'),
(103, 6, '2023-11-12'),
(104, 7, '2023-11-12'),
(104, 8, '2023-11-13'),
(105, 9, '2023-11-15'),
(105, 10, '2023-11-16'),
(101, 1, '2023-11-02'),
(101, 1, '2023-11-03'),
(102, 2, '2023-11-15'),
(102, 2, '2023-11-16'),
(103, 3, '2023-11-11'),
(103, 3, '2023-11-12'),
(104, 4, '2023-11-12'),
(104, 4, '2023-11-13'),
(105, 5, '2023-11-15'),
(105, 5, '2023-11-16');



INSERT INTO ASSIGNED (FacultyId, CourseId, DateAssigned) VALUES
(1, 1, '2023-11-01'),
(2, 2, '2023-11-02'),
(3, 3, '2023-11-11'),
(4, 4, '2023-11-12'),
(5, 5, '2023-11-15'),
(6, 6, '2023-11-20'),
(7, 7, '2023-11-22'),
(8, 8, '2023-11-25'),
(9, 9, '2023-11-28'),
(10, 10, '2023-11-30'),
(1, 6, '2023-11-01'),
(2, 7, '2023-11-02'),
(3, 8, '2023-11-11'),
(4, 9, '2023-11-12'),
(5, 10, '2023-11-15');
```

```sql
-- ENROLLMENT Table
INSERT INTO ENROLLMENT (EnrollmentId, EName, Rollno, CourseId, ProgramId) VALUES
(1, 'Enrollment1',101, 1, 1),
(2, 'Enrollment2',102, 2, 2),
(3, 'Enrollment3',103, 3, 3),
(4, 'Enrollment4',104, 4, 4),
(5, 'Enrollment5',105, 5, 5),
(6, 'Enrollment6', 106, 6, 6),
(7, 'Enrollment7', 107, 7, 7),
(8, 'Enrollment8', 108, 8, 8),
(9, 'Enrollment9', 109, 9, 9),
(10, 'Enrollment10', 110, 10, 10);
```

3.4 QUERIES

```sql
--1
-- Weekly View
SELECT DISTINCT S.SName, E.CourseId
FROM STUDENT S
JOIN ENROLLS E ON S.Rollno = E.Rollno
JOIN COURSE C ON E.CourseId = C.CourseId
JOIN COLLEGE COL ON C.CollegeId = COL.CollegeId
WHERE COL.CName = 'College Edwrd'
AND DATEPART(WEEK, E.DateEnrolled) = DATEPART(WEEK, GETDATE());

-- Monthly View
SELECT DISTINCT S.SName, E.CourseId
FROM STUDENT S
JOIN ENROLLS E ON S.Rollno = E.Rollno
JOIN COURSE C ON E.CourseId = C.CourseId
JOIN COLLEGE COL ON C.CollegeId = COL.CollegeId
WHERE COL.CName = 'College Edwrd'
AND MONTH(E.DateEnrolled) = MONTH(GETDATE());

-- Daily View
SELECT DISTINCT S.SName, E.CourseId
FROM STUDENT S
JOIN ENROLLS E ON S.Rollno = E.Rollno
JOIN COURSE C ON E.CourseId = C.CourseId
JOIN COLLEGE COL ON C.CollegeId = COL.CollegeId
WHERE COL.CName = 'College Edwrd'
AND CAST(E.DateEnrolled AS DATE) = CAST(GETDATE() AS DATE);
```

	SName	CourseId
1	Student2	2

	SName	CourseId
1	Student1	1
2	Student2	2

	SName	CourseId

```sql
--2
-- Weekly View
SELECT DISTINCT C.CName AS CourseName, COL.CName AS CollegeName
FROM COURSE C
JOIN DEPARTMENT D ON C.DepartmentId = D.DepartmentId
JOIN COLLEGE COL ON C.CollegeId = COL.CollegeId
WHERE D.DName = 'Social Sciences'
AND DATEPART(WEEK, C.DateOffered) = DATEPART(WEEK, GETDATE());

-- Monthly View
SELECT DISTINCT C.CName AS CourseName, COL.CName AS CollegeName
FROM COURSE C
JOIN DEPARTMENT D ON C.DepartmentId = D.DepartmentId
JOIN COLLEGE COL ON C.CollegeId = COL.CollegeId
WHERE D.DName = 'Social Sciences'
AND MONTH(C.DateOffered) = MONTH(GETDATE());

-- Daily View
SELECT DISTINCT C.CName AS CourseName, COL.CName AS CollegeName
FROM COURSE C
JOIN DEPARTMENT D ON C.DepartmentId = D.DepartmentId
JOIN COLLEGE COL ON C.CollegeId = COL.CollegeId
WHERE D.DName = 'Social Sciences'
AND CAST(C.DateOffered AS DATE) = CAST(GETDATE() AS DATE);
```

	CourseName	CollegeName
1	Course5	College3

	CourseName	CollegeName
1	Course5	College3

	CourseName	CollegeName

```sql
--3
-- Weekly View
SELECT DISTINCT S.SName AS StudentName, E.CourseId
FROM STUDENT S
JOIN ENROLLS E ON S.Rollno = E.Rollno
JOIN COURSE C ON E.CourseId = C.CourseId
WHERE C.CName = 'Course5'
AND DATEPART(WEEK, E.DateEnrolled) = DATEPART(WEEK, GETDATE());

-- Monthly View
SELECT DISTINCT S.SName AS StudentName, E.CourseId
FROM STUDENT S
JOIN ENROLLS E ON S.Rollno = E.Rollno
JOIN COURSE C ON E.CourseId = C.CourseId
WHERE C.CName = 'Course1'
AND MONTH(E.DateEnrolled) = MONTH(GETDATE());

-- Daily View
SELECT DISTINCT S.SName AS StudentName, E.CourseId
FROM STUDENT S
JOIN ENROLLS E ON S.Rollno = E.Rollno
JOIN COURSE C ON E.CourseId = C.CourseId
WHERE C.CName = 'Course1'
AND CAST(E.DateEnrolled AS DATE) = CAST(GETDATE() AS DATE);
```

	StudentName	CourseId
1	Student5	5

	StudentName	CourseId
1	Student1	1

	StudentName	CourseId

```sql
--4
-- For Courses
SELECT DISTINCT C.CName AS CourseName, COL.CName AS CollegeName
FROM COURSE C
JOIN COLLEGE COL ON C.CollegeId = COL.CollegeId
WHERE C.DateOffered BETWEEN '2023-11-01' AND '2023-11-30';

-- For Faculty Assignments
SELECT DISTINCT F.FName AS FacultyName, D.DName AS DepartmentName, COL.CName AS CollegeName
FROM FACULTY F
JOIN ASSIGNED A ON F.FacultyId = A.FacultyId
JOIN COURSE C ON A.CourseId = C.CourseId
JOIN DEPARTMENT D ON C.DepartmentId = D.DepartmentId
JOIN COLLEGE COL ON C.CollegeId = COL.CollegeId
WHERE A.DateAssigned BETWEEN '2023-11-01' AND '2023-11-30';
```

	CourseName	CollegeName
1	Course1	College Edwrd
2	Course10	College8
3	Course2	College Edwrd
4	Course3	College2
5	Course4	College2
6	Course5	College3
7	Course6	College4
8	Course7	College5
9	Course8	College6
10	Course9	College7

	FacultyName	DepartmentName	CollegeName
1	Faculty1	Engineering	College Edwrd
2	Faculty1	History	College4
3	Faculty10	Psychology	College8
4	Faculty2	Business	College Edwrd
5	Faculty2	Chemistry	College5
6	Faculty3	Arts	College2
7	Faculty3	Economics	College6
8	Faculty4	Political Science	College7
9	Faculty4	Science	College2
10	Faculty5	Psychology	College8
11	Faculty5	Social Sciences	College3
12	Faculty6	History	College4
13	Faculty7	Chemistry	College5
14	Faculty8	Economics	College6
15	Faculty9	Political Science	College7

```sql
--5
-- Weekly View
SELECT DISTINCT F.FName AS FacultyName, D1.DName AS Department1, D2.DName AS Department2
FROM FACULTY F
JOIN ASSIGNED A1 ON F.FacultyId = A1.FacultyId
JOIN COURSE C1 ON A1.CourseId = C1.CourseId
JOIN DEPARTMENT D1 ON C1.DepartmentId = D1.DepartmentId

JOIN ASSIGNED A2 ON F.FacultyId = A2.FacultyId
JOIN COURSE C2 ON A2.CourseId = C2.CourseId
JOIN DEPARTMENT D2 ON C2.DepartmentId = D2.DepartmentId

WHERE D1.DepartmentId <> D2.DepartmentId
AND (
    A1.DateAssigned <= DATEADD(WEEK, 1, A2.DateAssigned)
    AND A2.DateAssigned <= DATEADD(WEEK, 1, A1.DateAssigned)
);
```

	FacultyName	Department1	Department2
1	Faculty1	Engineering	History
2	Faculty1	History	Engineering
3	Faculty2	Business	Chemistry
4	Faculty2	Chemistry	Business
5	Faculty3	Arts	Economics
6	Faculty3	Economics	Arts
7	Faculty4	Political Science	Science
8	Faculty4	Science	Political Science
9	Faculty5	Psychology	Social Sciences
10	Faculty5	Social Sciences	Psychology

```sql
-- Monthly View
SELECT DISTINCT
    F.FName AS FacultyName,
    D1.DName AS Department1,
    D2.DName AS Department2
FROM FACULTY F
JOIN ASSIGNED A1 ON F.FacultyId = A1.FacultyId
JOIN COURSE C1 ON A1.CourseId = C1.CourseId
JOIN DEPARTMENT D1 ON C1.DepartmentId = D1.DepartmentId

JOIN ASSIGNED A2 ON F.FacultyId = A2.FacultyId
JOIN COURSE C2 ON A2.CourseId = C2.CourseId
JOIN DEPARTMENT D2 ON C2.DepartmentId = D2.DepartmentId

WHERE D1.DepartmentId <> D2.DepartmentId
AND (
    A1.DateAssigned <= DATEADD(MONTH, 1, A2.DateAssigned)
    AND A2.DateAssigned <= DATEADD(MONTH, 1, A1.DateAssigned)
);
```

	FacultyName	Department1	Department2
1	Faculty1	Engineering	History
2	Faculty1	History	Engineering
3	Faculty2	Business	Chemistry
4	Faculty2	Chemistry	Business
5	Faculty3	Arts	Economics
6	Faculty3	Economics	Arts
7	Faculty4	Political Science	Science
8	Faculty4	Science	Political Science
9	Faculty5	Psychology	Social Sciences
10	Faculty5	Social Sciences	Psychology

```sql
-- Daily View
SELECT DISTINCT
    F.FName AS FacultyName,
    D1.DName AS Department1,
    D2.DName AS Department2
FROM FACULTY F
JOIN ASSIGNED A1 ON F.FacultyId = A1.FacultyId
JOIN COURSE C1 ON A1.CourseId = C1.CourseId
JOIN DEPARTMENT D1 ON C1.DepartmentId = D1.DepartmentId

JOIN ASSIGNED A2 ON F.FacultyId = A2.FacultyId
JOIN COURSE C2 ON A2.CourseId = C2.CourseId
JOIN DEPARTMENT D2 ON C2.DepartmentId = D2.DepartmentId

WHERE D1.DepartmentId <> D2.DepartmentId
AND (
    A1.DateAssigned <= DATEADD(DAY, 1, A2.DateAssigned)
    AND A2.DateAssigned <= DATEADD(DAY, 1, A1.DateAssigned)
);
```

	FacultyName	Department1	Department2
1	Faculty1	Engineering	History
2	Faculty1	History	Engineering
3	Faculty2	Business	Chemistry
4	Faculty2	Chemistry	Business
5	Faculty3	Arts	Economics
6	Faculty3	Economics	Arts
7	Faculty4	Political Science	Science
8	Faculty4	Science	Political Science
9	Faculty5	Psychology	Social Sciences
10	Faculty5	Social Sciences	Psychology

```sql
--6
SELECT
    F.FacultyId,
    F.FName AS FacultyName,
    D.DName AS DepartmentName,
    COUNT(A.CourseId) AS AssignmentCount
FROM FACULTY F
JOIN ASSIGNED A ON F.FacultyId = A.FacultyId
JOIN COURSE C ON A.CourseId = C.CourseId
JOIN DEPARTMENT D ON C.DepartmentId = D.DepartmentId
WHERE C.DateOffered >= DATEADD(WEEK, -1, GETDATE())
GROUP BY F.FacultyId, F.FName, D.DName;

-- Query for monthly assignments
SELECT
    F.FacultyId,
    F.FName AS FacultyName,
    D.DName AS DepartmentName,
    COUNT(A.CourseId) AS AssignmentCount
FROM FACULTY F
JOIN ASSIGNED A ON F.FacultyId = A.FacultyId
JOIN COURSE C ON A.CourseId = C.CourseId
JOIN DEPARTMENT D ON C.DepartmentId = D.DepartmentId
WHERE C.DateOffered >= DATEADD(MONTH, -1, GETDATE())
GROUP BY F.FacultyId, F.FName, D.DName;

-- Query for daily assignments
SELECT
    F.FacultyId,
    F.FName AS FacultyName,
    D.DName AS DepartmentName,
    COUNT(A.CourseId) AS AssignmentCount
FROM FACULTY F
JOIN ASSIGNED A ON F.FacultyId = A.FacultyId
JOIN COURSE C ON A.CourseId = C.CourseId
JOIN DEPARTMENT D ON C.DepartmentId = D.DepartmentId
WHERE C.DateOffered >= DATEADD(DAY, -1, GETDATE())
GROUP BY F.FacultyId, F.FName, D.DName;
```

	FacultyId	FacultyName	DepartmentName	AssignmentCount
1	3	Faculty3	Arts	1
2	2	Faculty2	Business	1
3	2	Faculty2	Chemistry	1
4	7	Faculty7	Chemistry	1
5	3	Faculty3	Economics	1
6	8	Faculty8	Economics	1
7	1	Faculty1	History	1
8	6	Faculty6	History	1
9	4	Faculty4	Political Science	1
10	9	Faculty9	Political Science	1
11	5	Faculty5	Psychology	1
12	10	Faculty10	Psychology	1
13	4	Faculty4	Science	1
14	5	Faculty5	Social Sciences	1

	FacultyId	FacultyName	DepartmentName	AssignmentCount
1	3	Faculty3	Arts	1
2	2	Faculty2	Business	1
3	2	Faculty2	Chemistry	1
4	7	Faculty7	Chemistry	1
5	3	Faculty3	Economics	1
6	8	Faculty8	Economics	1
7	1	Faculty1	Engineering	1
8	1	Faculty1	History	1
9	6	Faculty6	History	1
10	4	Faculty4	Political Science	1
11	9	Faculty9	Political Science	1
12	5	Faculty5	Psychology	1
13	10	Faculty10	Psychology	1
14	4	Faculty4	Science	1
15	5	Faculty5	Social Sciences	1

	FacultyId	FacultyName	DepartmentName	AssignmentCount
1	2	Faculty2	Business	1
2	2	Faculty2	Chemistry	1
3	7	Faculty7	Chemistry	1
4	3	Faculty3	Economics	1
5	8	Faculty8	Economics	1
6	1	Faculty1	History	1
7	6	Faculty6	History	1
8	4	Faculty4	Political Science	1
9	9	Faculty9	Political Science	1
10	5	Faculty5	Psychology	1
11	10	Faculty10	Psychology	1
12	4	Faculty4	Science	1
13	5	Faculty5	Social Sciences	1

```sql
--7
-- Query for faculty members with the most weekly assignments in specific departments
SELECT TOP 1 WITH TIES
    F.FacultyId,
    F.FName AS FacultyName,
    D.DName AS DepartmentName,
    COUNT(A.CourseId) AS AssignmentCount
FROM FACULTY F
JOIN ASSIGNED A ON F.FacultyId = A.FacultyId
JOIN COURSE C ON A.CourseId = C.CourseId
JOIN DEPARTMENT D ON C.DepartmentId = D.DepartmentId
WHERE C.DateOffered >= DATEADD(WEEK, -1, GETDATE())
GROUP BY F.FacultyId, F.FName, D.DName
ORDER BY AssignmentCount DESC;

-- Query for faculty members with the most monthly assignments in specific departments
SELECT TOP 1 WITH TIES
    F.FacultyId,
    F.FName AS FacultyName,
    D.DName AS DepartmentName,
    COUNT(A.CourseId) AS AssignmentCount
FROM FACULTY F
JOIN ASSIGNED A ON F.FacultyId = A.FacultyId
JOIN COURSE C ON A.CourseId = C.CourseId
JOIN DEPARTMENT D ON C.DepartmentId = D.DepartmentId
WHERE C.DateOffered >= DATEADD(MONTH, -1, GETDATE())
GROUP BY F.FacultyId, F.FName, D.DName
ORDER BY AssignmentCount DESC;

-- Query for faculty members with the most daily assignments in specific departments
SELECT TOP 1 WITH TIES
    F.FacultyId,
    F.FName AS FacultyName,
    D.DName AS DepartmentName,
    COUNT(A.CourseId) AS AssignmentCount
FROM FACULTY F
JOIN ASSIGNED A ON F.FacultyId = A.FacultyId
JOIN COURSE C ON A.CourseId = C.CourseId
JOIN DEPARTMENT D ON C.DepartmentId = D.DepartmentId
WHERE C.DateOffered >= DATEADD(DAY, -1, GETDATE())
GROUP BY F.FacultyId, F.FName, D.DName
ORDER BY AssignmentCount DESC;
```

	FacultyId	FacultyName	DepartmentName	AssignmentCount
1	3	Faculty3	Arts	1
2	2	Faculty2	Business	1
3	2	Faculty2	Chemistry	1
4	7	Faculty7	Chemistry	1
5	3	Faculty3	Economics	1
6	8	Faculty8	Economics	1
7	1	Faculty1	History	1
8	6	Faculty6	History	1
9	4	Faculty4	Political Science	1
10	9	Faculty9	Political Science	1
11	5	Faculty5	Psychology	1
12	10	Faculty10	Psychology	1
13	4	Faculty4	Science	1
14	5	Faculty5	Social Sciences	1

	FacultyId	FacultyName	DepartmentName	AssignmentCount
1	3	Faculty3	Arts	1
2	2	Faculty2	Business	1
3	2	Faculty2	Chemistry	1
4	7	Faculty7	Chemistry	1
5	3	Faculty3	Economics	1
6	8	Faculty8	Economics	1
7	1	Faculty1	Engineering	1
8	1	Faculty1	History	1
9	6	Faculty6	History	1
10	4	Faculty4	Political Science	1
11	9	Faculty9	Political Science	1
12	5	Faculty5	Psychology	1
13	10	Faculty10	Psychology	1
14	4	Faculty4	Science	1
15	5	Faculty5	Social Sciences	1

	FacultyId	FacultyName	DepartmentName	AssignmentCount
1	2	Faculty2	Business	1
2	2	Faculty2	Chemistry	1
3	7	Faculty7	Chemistry	1
4	3	Faculty3	Economics	1
5	8	Faculty8	Economics	1
6	1	Faculty1	History	1
7	6	Faculty6	History	1
8	4	Faculty4	Political Science	1
9	9	Faculty9	Political Science	1
10	5	Faculty5	Psychology	1
11	10	Faculty10	Psychology	1
12	4	Faculty4	Science	1
13	5	Faculty5	Social Sciences	1

```sql
--8
SELECT S.SName AS StudentName, C.CName AS CourseName, COUNT(E.CourseId) AS EnrollmentCount
FROM STUDENT S
JOIN ENROLLS E ON S.Rollno = E.Rollno
JOIN COURSE C ON E.CourseId = C.CourseId
GROUP BY S.SName, C.CName
HAVING COUNT(E.CourseId) > 1;
```

	StudentName	CourseName	EnrollmentCount
1	Student1	Course1	2
2	Student2	Course2	2
3	Student3	Course3	2
4	Student4	Course4	2
5	Student5	Course5	2

```sql
--9
SELECT S.SName AS StudentName, COUNT(DISTINCT E.CourseId) AS DifferentCourseCount
FROM STUDENT S
JOIN ENROLLS E ON S.Rollno = E.Rollno
GROUP BY S.SName
HAVING COUNT(DISTINCT E.CourseId) > 1;
```

	StudentName	DifferentCourseCount
1	Student1	3
2	Student2	3
3	Student3	3
4	Student4	3
5	Student5	3

Some Additional Queries for Updating Deletion and Insertion:

```sql
------------------------ADMINISTRATOR
-- Add an Administrator
INSERT INTO ADMINISTRATOR (AdminId, AName, CollegeId)
VALUES (11, 'NewAdmin', 4); -- Here we will replace college id

-- Update Administrator College
UPDATE ADMINISTRATOR
SET CollegeId = 3 -- Replace with the desired college
WHERE AdminId = 11; -- Replace with the desired Administrator's ID

-- Delete Administrator
DELETE FROM ADMINISTRATOR
WHERE AdminId = 11; -- Replace with the desired Administrator's ID

------------------------FACULTY MEMBER
-- Add a Faculty Member
INSERT INTO FACULTY (FacultyId, FName, DepartmentId)
VALUES (11, 'NewFaculty', 6); --  Here we will replace department id

-- Update Faculty Member Department
UPDATE FACULTY
SET DepartmentId = 4 -- Replace with the desired department
WHERE FacultyId = 11; -- Replace with the desired Faculty Member's ID

-- Delete Faculty Member
DELETE FROM FACULTY
WHERE FacultyId = 11; -- Replace with the desired Faculty Member's ID

------------------------STUDENT
-- Add a Student
INSERT INTO STUDENT (Rollno, SName, DepartmentId)
VALUES (106, 'NewStudent', 5);  --Here we will replace department id

-- Update Student Department
UPDATE STUDENT
SET DepartmentId = 3 -- Replace with the desired department
WHERE Rollno = 106; -- Replace with the desired Student's Rollno

-- Delete Student
DELETE FROM STUDENT
WHERE Rollno = 106; -- Replace with the desired Student's Rollno

------------------------STAFF
-- Add College Staff
INSERT INTO STAFF (StaffId, SName, SRole, ProgramId)
VALUES (11, 'NewStaff', 'Role6', 4); -- Here we will replace program id

-- Update College Staff's Program
UPDATE STAFF
SET ProgramId = 5 -- Replace with the desired program
WHERE StaffId = 11; -- Replace with the desired Staff's ID

-- Delete College Staff
DELETE FROM STAFF
WHERE StaffId = 11; -- Replace with the desired Staff's ID
```

Q2.

Insertion Anomaly:

Let's say you want to add a new student Haider, but he is yet to take a course meaning, he hasn't taken any course but, in the query, you must add a course so you cannot add Haider, without a course so this is an insertion anomaly.

Deletion Anomaly:

Assume you want to delete entry for Aleena but since she is the only student who is taking C2 course which will completely remove it from the table and that course could be some other student's as well so this is what deletion anomaly is that if you remove a data it may lead to deletion of another relevant data which will affect database.

Update Anomaly:

Let's say, you want to update the C2 course and change it to let's say C4 then it is compulsory to change both entries of C2. This is the update anomaly that if you only change 1 entry of C2 and not the other then it will create inconsistency in the tables.

Normalization

Original Table

Student_ID	Student_City	Student_Name	Course_Offered_ID	Course_Semester	Course_Offered_Year	Course Grade	Course ID	Course Name
1	Rawalpindi	Ali	1	Fall	2006	3.5	C1	Database
1	Rawalpindi	Ali	2	Fall	2006	3.3	C2	PF
2	Lahore	Aleena	3	SPRING	2007	3.1	C3	OOP
2	Lahore	Aleena	2	Fall	2006	3.4	C2	PF

1NF

Since there are repeating values, we will make two primary keys.

Student_ID	Course_Offered_ID	Student_City	Student_Name	Course_Semester	Course_Offered_Year	Course Grade	Course ID	Course Name
1	1	Rawalpindi	Ali	Fall	2006	3.5	C1	Database
1	2	Rawalpindi	Ali	Fall	2006	3.3	C2	PF
2	3	Lahore	Aleena	SPRING	2007	3.1	C3	OOP
2	2	Lahore	Aleena	Fall	2006	3.4	C2	PF

To counter the multivalued problem, we made "Course_Offered_ID" as key value along with Student_ID

2NF

Student_ID -> Student_City

Student_ID -> Student_Name

{Student_ID,Course_Offered_ID} -> Course Grade

Course_Offered_ID -> Course_Semester

Course_Offered_ID -> Course_Offered_Year

Course_Offered_ID -> Course_ID

Course_Offered_ID -> Course_Name

Student_ID	Student_City	Student_Name
1	Rawalpindi	Ali
2	Lahore	Aleena

Student_ID	Course_Offered_ID	Course Grade
1	1	3.5
1	2	3.3
2	3	3.1
2	2	3.4

Course_Offered_ID	Course_Semester	Course_Offered_Year	Course_ID	Course_Name
1	Fall	2006	C1	Database
2	Fall	2006	C2	PF
3	SPRING	2007	C3	OOP

3NF

Transitivity Rule

Course_Senester -> Course_Offered_Year

Course_ID -> Course_Name

Student_ID	Student_City	Student_Name
1	Rawalpindi	Ali
2	Lahore	Aleena

Student_ID	Course_Offered_ID	Course Grade
1	1	3.5
1	2	3.3
2	3	3.1
2	2	3.4

Course_Offered_ID	Course_Semester	Course_Offered_Year
1	Fall	2006
2	Fall	2006
3	SPRING	2007

Course_Offered_ID	Course_ID	Course_Name
1	C1	Database
2	C2	PF
3	C3	OOP