SQL Assignment

1. **Display StudentID, StudentName, and their Department.**

SELECT student_id, student_name, department_id from STUDENT;

2. **Display Department and Corresponding Student Count.**

SELECT department_name, department_intake from DEPARTMENT;

3. **Display Pending number of Seats department wise.**

Assuming 150 is the maximum number of seats that can be allocated for each department (MAX_SEAT = 150)

SELECT (150 - department_intake) AS ’Pending Seats’ FROM DEPARTMENT;

4. **Display Id, Name of Students who were born in the year 1997**

SELECT student_id, student_name FROM STUDENT where YEAR(date_of_birth) = ‘1997’;

5. **Display Id, Name of Students whose name starts with 'R'.**

SELECT student_id, student_name FROM STUDENT where student_name LIKE ‘R%’

6. **Display Id, Name of Students whose name starts with 'R' and ends with 'i' and the length is 4 characters.**

SELECT student_id, student_name FROM STUDENT where student_name LIKE ‘R _ _ i’;

7. **Filter the Students of "Electronics" Department.**

SELECT student_name FROM STUDENT S,DEPARTMENT D where S.department_id = D.department_id AND D.department_name = ‘Electronics’;

8. Fetch the List of Students who have enrolled in optional courses.

```
SELECT DISTINCT S.student_name FROM STUDENT S,
OPTIONALCOURSERELATION OCR where S.student_id = OCR.student_id;
```

9. Fetch the List of Students who havent enrolled for Optional Courses.

```
SELECT student_name FROM STUDENT where student_name NOT IN (SELECT
DISTINCT S.student_name FROM STUDENT S, OPTIONALCOURSERELATION
OCR where S.student_id = OCR.student_id);
```

10. Fetch the Departments which offer additional optional courses.

```
SELECT DISTINCT D.department_name FROM DEPARTMENT D,
DEPTOPTIONALCOURSERELATION DOCR where D.department_id =
DOCR.department_id
```

11. Fetch the Departments which don’t offer optional courses.

```
SELECT department_name FROM DEPARTMENT where department_id NOT IN
(SELECT DISTINCT department_id FROM DEPTOPTIONALCOURSERELATION);
```

12. Fetch the Optional Courses which are combinely provided by two or more departments.

```
SELECT OC.course_name, STRING_AGG(D.department_name,’-‘) FROM
OPTIONALCOURSES OC, DEPARTMENT D, DEPTOPTIONALCOURSERELATION
DOCR GROUP BY OC.course_name HAVING COUNT(OC.course_name) > 1 and
OC.course_id = DOCR.course_id and D.department_id = DOCR.department_id;
```

13. List the Optional Courses and their Corresponding Departments.

```
SELECT OC.course_name, D.department_name FROM DEPARTMENT D,
DEPTOPTIONALCOURSERELATION DOCR, OPTIONALCOURSES OC where
D.departmwnt_id = DOCR.department_id and DOCR.course_id = OC.course_id;
```

14.List the "General" Category Department as General Studies and display the corresponding optional courses provided.

```
SELECT OC.course_name AS ’General Studies’ FROM DEPARTMENT D,
DEPTOPTIONALCOURSERELATION DOCR, OPTIONALCOURSES OC where
```

```
D.departmwnt_id = DOCR.department_id and DOCR.course_id = OC.course_id
and D.department_name = ‘General’;
```

15.Students who were born in between December 1996 and May 1997 (inclusive).

```
SELECT student_name FROM STUDENT where YEAR(date_of_birth) BETWEEN
‘1996’ AND ‘1997’;
```

16. Students who have chosen the Optional Course in their own department.

```
SELECT S.student_name FROM OPTIONALCOURSRELATION OCR, STUDENT S,
DEPTOPTIONALCOURSERELATION DOCR where S.department_id =
DOCR.department_id and S.student_id = OCR.student_id and DOCR.course_id
= OCR.course_id;
```

17. Students who have chosen the Optional Course in other Departments.

```
SELECT S.student_name FROM OPTIONALCOURSRELATION OCR, STUDENT S,
DEPTOPTIONALCOURSERELATION DOCR where S.department_id <>
DOCR.department_id and S.student_id = OCR.student_id and DOCR.course_id
= OCR.course_id;
```

18.Student Count by Optional Course.

```
SELECT COUNT(student_id), OC.course_name FROM
OPTIONALCOURSERELATION OCR, OPTIONALCOURSES OC GROUP BY
OCR.course_id HAVING OC.course_id = OCR.course_id;
```

19.Which Department Offer the maximum Intake and which courses offer the least intake.

```
SELECT department_name FROM DEPARTMENT where department_intake =
MAX(department_intake);
```

20.Delete the Electrical Department and its associated Optional Courses.

DELETE FROM DEPARTMENT D, DEPTOPTIONALCOURSERELATION DOCR, OPTIONALCOURSES OC where D.department_id = DOCR.department_id and OC.course_id = DOCR.course_id and D.department_name = ‘Electrical’;

21.Update the Date of Birth of the Student Ravi as Feb 28, 1997

UPDATE STUDENT SET data_of_birth = ‘1997-02-28’ where student_name = ‘Ravi’;

22.Add the below optional courses for Bio Medical Engineering
a. Tissue Culture
b. Vaccine Tech
And Map them for any two Students.

a. INSERT INTO OPTIONALCOURSES (course_name) VALUES (‘Tissue Culture’); INSERT INTO DEPTOPTIONALCOURSERELATION (department_id,course_id) SELECT D.department_id, OC.course_id from DEPARTMENT D, OPTIONALCOURSES OC where course_name = ‘Tissue Culture’ and department_name = ‘Bio Medical Engineering’;

   INSERT INTO OPTIONALCOURSERELATION (student_id, course_id) SELECT S.student_id, OC.course_id from STUDENT S, OPTIONALCOURSES OC where S.student_id = ‘student1id’ and course_name = ‘Tissue Culture’;

b. INSERT INTO OPTIONALCOURSES (course_name) VALUES (‘Vaccine Tech’); INSERT INTO DEPTOPTIONALCOURSERELATION (department_id,course_id) SELECT D.department_id, OC.course_id from DEPARTMENT D, OPTIONALCOURSES OC where course_name = ‘Vaccine Tech’ and department_name = ‘Bio Medical Engineering’;

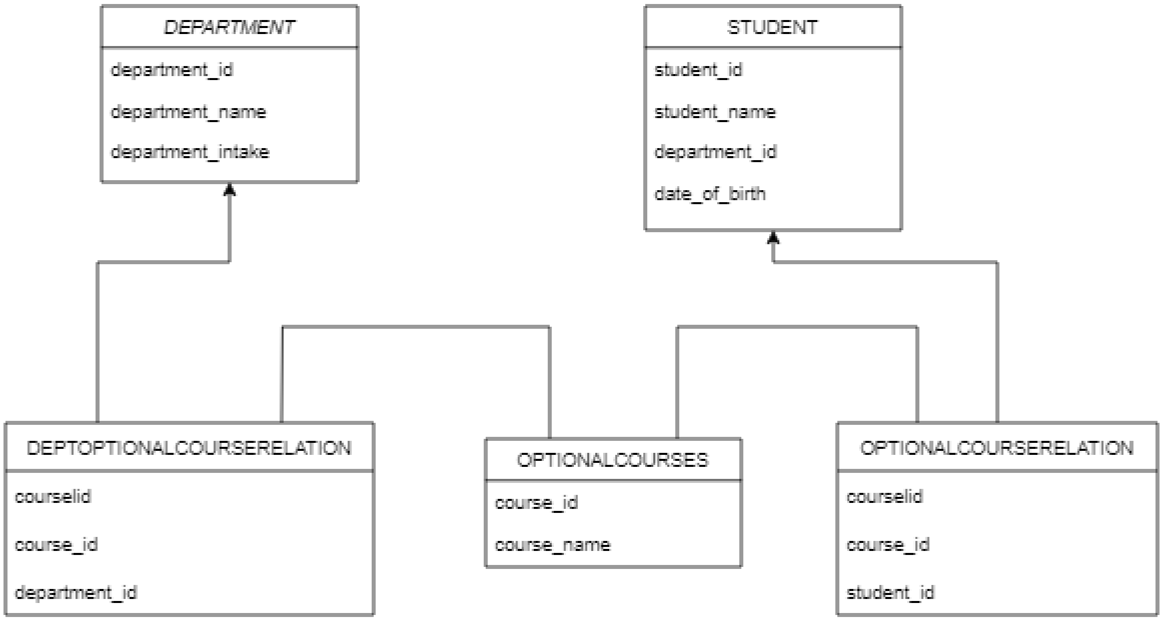
   INSERT INTO OPTIONALCOURSERELATION (student_id, course_id) SELECT S.student_id, OC.course_id from STUDENT S, OPTIONALCOURSES OC where S.student_id = ‘student1id’ and course_name = ‘Vaccine Tech’;

23. What else is needed in the database?

- We need a relation table for the Student and Department.

- We also need a table of the core courses provided by respective departments.

SCHEMA DIAGRAM



24.Explain the mappings available in the data.

- The Department table is mapped with the Optional Course table using department id and course id as foreign key.
- Department_id and course_id are primary keys of department and optional course tables respectively.
- Similarly, Student table is mapped with optional courses table using student_id and course_id as foreign key.
- Student_id is primary key of student table.

25. Explain the way to mitigate Foreign key constraint while modifying data.

- Foreign Key may restrict us to add data if it doesn't follow any of the Foreign key constraint that has been enabled.
- We have to change the datatype of the column specified in foreign key constraint to avoid while modifying data. (Avoid Mismatching datatypes)
- Sometime Foreign keys may link to non existing or deleted table it should be updated in foreign key constraint so then the foreign key is independent of the deleted irrelevant table. (Avoid dangling foreign keys)
- The constraint also stops modifying data through update statement.
- One way to mitigate this constraint is to modify the constraint or delete the foreign key constraint itself.
- For example: **ALTER TABLE <tablename> DROP FOREIGN KEY <fk_symbol>;**