**SQL Assignment**

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

SELECT student\_id, student\_name, department\_id from STUDENT;

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

SELECT department\_name, department\_intake from DEPARTMENT;

1. **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;

1. **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’;

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

SELECT student\_id, student\_name FROM STUDENT where student\_name LIKE ‘R%’

1. **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’;

1. **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’;

1. **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;

1. **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);

1. **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

1. **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);

1. **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;

1. **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;

1. **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’;

1. **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’;

1. **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;

1. **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;

1. **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;

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

SELECT department\_name FROM DEPARTMENT where department\_intake = MAX(department\_intake);

1. **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’;

1. **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’;

1. Add the below optional courses for Bio Medical Engineering

a. Tissue Culture

b. Vaccine Tech

And Map them for any two Students.

1. 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’;

1. INSERT INTO OPTIONALCOURSES (course\_name) VALUES (‘Vaccine Tech’);

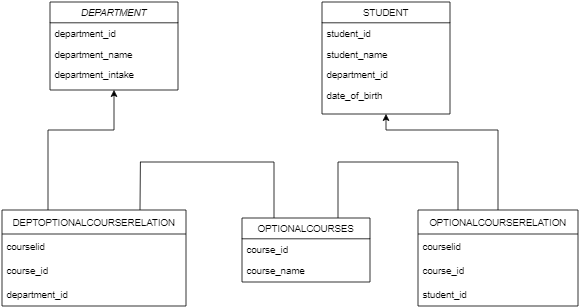
INSERT INTO DEPTOPTIONALCOURSERELATION (department\_id,course\_id) SELECT D.department\_id, OC.course\_id from DEPARTMENT D, OPTIONALCOURSES OC where course\_name = ‘Vaccine Tech’ and department\_name = ‘Bio Medical Engineering’;

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’;

1. 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



1. 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.

1. 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>;**