SQL Final Assessment

Table Definition

```
test=# \d DEPARTMENT
                                      Table "public.department"
      Column       |          Type          |                            Modifiers
-------------------+------------------------+-------------------------------------------------------------------
 department_id     | integer                | not null default nextval('department_department_id_seq'::regclass)
 department_name   | character varying(255) | not null
 department_intake | integer                | not null
Indexes:
    "department_pkey" PRIMARY KEY, btree (department_id)
Referenced by:
    TABLE "deptoptionalcourserelation" CONSTRAINT "deptoptionalcourserelation_department_id_fkey" FOREIGN KEY (department_id) REFERENCES departme
nt(department_id)
    TABLE "student" CONSTRAINT "student_department_id_fkey" FOREIGN KEY (department_id) REFERENCES department(department_id)
```

```
test=# \d OPTIONALCOURSES;
                                    Table "public.optionalcourses"
   Column    |          Type          |                              Modifiers
-------------+------------------------+-----------------------------------------------------------------------
 course_id   | integer                | not null default nextval('optionalcourses_course_id_seq'::regclass)
 course_name | character varying(255) |
Indexes:
    "optionalcourses_pkey" PRIMARY KEY, btree (course_id)
Referenced by:
    TABLE "deptoptionalcourserelation" CONSTRAINT "deptoptionalcourserelation_course_id_fkey" FOREIGN KEY (course_id) REFERENCES optionalcourses(
course_id)
    TABLE "optionalcourserelation" CONSTRAINT "optionalcourserelation_course_id_fkey" FOREIGN KEY (course_id) REFERENCES optionalcourses(course_i
d)
```

```
test=# \d STUDENT
                                     Table "public.student"
    Column     |          Type          |                           Modifiers
---------------+------------------------+---------------------------------------------------------------
 student_id    | integer                | not null default nextval('student_student_id_seq'::regclass)
 student_name  | character varying(255) | not null
 department_id | integer                | not null
 date_of_birth | date                   | not null default '2000-01-01'::date
Indexes:
    "student_pkey" PRIMARY KEY, btree (student_id)
Foreign-key constraints:
    "student_department_id_fkey" FOREIGN KEY (department_id) REFERENCES department(department_id)
Referenced by:
    TABLE "optionalcourserelation" CONSTRAINT "optionalcourserelation_student_id_fkey" FOREIGN KEY (student_id) REFERENCES student(student_id)
```

```
test=# \d DEPTOPTIONALCOURSERELATION
                              Table "public.deptoptionalcourserelation"
    Column     |  Type   |                                  Modifiers
---------------+---------+-----------------------------------------------------------------------------
 courserelid   | integer | not null default nextval('deptoptionalcourserelation_courserelid_seq'::regclass)
 department_id | integer | not null
 course_id     | integer | not null
Indexes:
    "deptoptionalcourserelation_pkey" PRIMARY KEY, btree (courserelid)
Foreign-key constraints:
    "deptoptionalcourserelation_course_id_fkey" FOREIGN KEY (course_id) REFERENCES optionalcourses(course_id)
    "deptoptionalcourserelation_department_id_fkey" FOREIGN KEY (department_id) REFERENCES department(department_id)
```

```
test=# \d OPTIONALCOURSERELATION
                            Table "public.optionalcourserelation"
   Column    |  Type   |                                Modifiers
-------------+---------+---------------------------------------------------------------------------
 courserelid | integer | not null default nextval('optionalcourserelation_courserelid_seq'::regclass)
 student_id  | integer | not null
 course_id   | integer | not null
Indexes:
    "optionalcourserelation_pkey" PRIMARY KEY, btree (courserelid)
Foreign-key constraints:
    "optionalcourserelation_course_id_fkey" FOREIGN KEY (course_id) REFERENCES optionalcourses(course_id)
    "optionalcourserelation_student_id_fkey" FOREIGN KEY (student_id) REFERENCES student(student_id)
```

Loading the data

Use the below command to load the data.

psql -Upostgres -f pgsql_training_dump.sql

SQL Final Assessment

Tasks

1. Display StudentID, StudentName and their Department.
2. Display Department and Corresponding Student Count.
3. Display Pending number of Seats department wise.
4. Display Id, Name of Students who were born in the year 1997.
5. Display Id, Name of Students whose name starts with 'R'.
6. Display Id, Name of Students whose name starts with 'R' and ends with 'i' and the length is 4 characters.
7. Filter the Students of "Electronics" Department.
8. Fetch the List of Students who have enrolled in optional courses.
9. Fetch the List of Students who havent enrolled for Optional Courses.
10. Fetch the Departments which offer additional optional courses.
11. Fetch the Departments which dont offer optional courses.
12. Fetch the Optional Courses which are combinely provided by two or more departments.
    [Hint : Use string_agg() function to accomplish this task.]
13. List the Optional Courses and their Corresponding Departments.
14. List the "General" Category Department as General Studies and display the corresponding optional courses provided.
15. Students who were born in between December 1996 and May 1997 (inclusive).
16. Students who have chosen the Optional Course in their own department.
17. Students who have chosen the Optional Course in other Departments.
18. Student Count by Optional Course
19. Which Department Offer the maximum Intake and which courses offer the least intake.
20. Delete the Electrical Department and its associated Optional Courses.
21. Update the Date of Birth of the Student Ravi as Feb 28, 1997.
22. Add the below optional courses for Bio Medical Engineering
    a. Tissue Culture
    b. Vaccine Tech

    And Map them for any two Students.
23. What else is needed in this database ? [Theoretical] . Prepare a Schema diagram
24. Explain the mappings available in the data.
25. Explain the way to mitigate Foreign key constraint while modifying data.

[Theoretical]