# Queries with OUTPUT
# Final Submission

**Sujal Prajapati - 202301478**
**Kalp Shah - 202301481**
**Vraj Parikh - 202301440**

## SQL Queries :

**Elective Management Queries :**

1. Insert a new elective course into the database

```
568 ∨  INSERT INTO Course (CourseID, CourseName, Branch, Credits, Lecture, Tutorial, Labs, Material)
569     VALUES ('CSE01', 'Data Science', 'CSE', 4, 3, TRUE, TRUE, 'Material content here');
570
```

Data Output   Messages   Notifications

```
INSERT 0 1

Query returned successfully in 107 msec.
```

2. Update the details of an existing elective course

```
571 ∨  UPDATE Course
572     SET CourseName = 'Advanced Data Science',
573         Credits = 5,
574         Material = 'Updated material content'
575     WHERE CourseID = 'CSE01';
576
```

Data Output   Messages   Notifications

```
UPDATE 1

Query returned successfully in 227 msec.
```

3. Delete an elective course from the database

```
577 ∨  DELETE FROM Course
578     WHERE CourseID = 'CSE01';
579
```

Data Output   Messages   Notifications

```
DELETE 1

Query returned successfully in 166 msec.
```

## 4.Retrieve a list of all elective courses

```sql
581  SELECT * FROM Course;
```

Data Output | Messages | Notifications

Showing rows: 1 to 54   Page No: 1   of 1

| | courseid [PK] character varying (5) | coursename character varying (100) | branch character varying (10) | credits integer | lecture integer | tutorial boolean | labs boolean | material text |
|---|---|---|---|---|---|---|---|---|
| 1 | GT01 | Graph Theory and Algorithms | ICT | 4 | 3 | true | false | Basic graph concepts, algorithms |
| 2 | AA02 | Approximation Algorithms | ICT | 3 | 3 | false | false | Approximation techniques |
| 3 | CC03 | Computational Complexity | ICT | 3 | 3 | true | false | Complexity classes |
| 4 | RA04 | Randomized Algorithms | ICT | 3 | 3 | false | false | Randomized computation |
| 5 | IC06 | Introduction to Cryptography | ICT | 3 | 3 | false | false | Encryption methods |
| 6 | CD11 | Compiler Design | ICT | 4 | 3 | true | true | Compiler construction |
| 7 | SE24 | Software Engineering | ICT | 4 | 3 | true | true | Software development |
| 8 | SN31 | System and Network Security | ICT | 4 | 3 | true | true | Security principles |
| 9 | NL22 | Natural Language Processing | ICT | 4 | 3 | true | false | NLP techniques |
| 10 | DM21 | Data Mining and Visualization | ICT | 4 | 3 | true | true | Data analysis |

## 5. Retrieve details of a specific elective course by course ID

```sql
583  SELECT * FROM Course
584  WHERE CourseID = 'GT01';
585
```

Data Output | Messages | Notifications

Showing rows: 1 to 1   Page No: 1

| | courseid [PK] character varying (5) | coursename character varying (100) | branch character varying (10) | credits integer | lecture integer | tutorial boolean | labs boolean | material text |
|---|---|---|---|---|---|---|---|---|
| 1 | GT01 | Graph Theory and Algorithms | ICT | 4 | 3 | true | false | Basic graph concepts, algorithms |

## 6. Find all elective courses offered by a specific department (Branch)

```sql
586  SELECT * FROM Course
587  WHERE Branch = 'MNC';
588
```

Data Output | Messages | Notifications

Showing rows: 1 to 12   Page No: 1   of 1

| | courseid [PK] character varying (5) | coursename character varying (100) | branch character varying (10) | credits integer | lecture integer | tutorial boolean | labs boolean | material text |
|---|---|---|---|---|---|---|---|---|
| 1 | QC05 | Quantum Computing | MNC | 4 | 3 | true | true | Quantum principles |
| 2 | AI37 | Introduction to AI | MNC | 4 | 3 | true | false | AI fundamentals |
| 3 | DL35 | Deep Learning | MNC | 4 | 3 | true | true | Neural networks |
| 4 | AM08 | Adversarial Machine Learning | MNC | 3 | 3 | false | false | Security in ML |
| 5 | MS09 | Machine Learning and Security | MNC | 3 | 3 | true | false | ML for security |
| 6 | RS36 | Recommendation Systems | MNC | 3 | 3 | false | false | Recommender algorithms |
| 7 | CN40 | Introduction to Complex Network | MNC | 3 | 3 | false | false | Network theory |
| 8 | IR39 | Introduction to Robotics | MNC | 4 | 3 | true | true | Robotics principles |
| 9 | CT19 | Control Theory | MNC | 4 | 3 | true | true | Control systems |
| 10 | OP25 | Optimization | MNC | 3 | 3 | false | false | Optimization methods |

## 7. Find all elective courses with a specific credit count

```sql
589  SELECT * FROM Course
590  WHERE Credits = 4;
591
```

Data Output | Messages | Notifications

Showing rows: 1 to 25   Page No: 1   of

| | courseid [PK] character varying (5) | coursename character varying (100) | branch character varying (10) | credits integer | lecture integer | tutorial boolean | labs boolean | material text |
|---|---|---|---|---|---|---|---|---|
| 1 | GT01 | Graph Theory and Algorithms | ICT | 4 | 3 | true | false | Basic graph concepts, algorithms |
| 2 | CD11 | Compiler Design | ICT | 4 | 3 | true | true | Compiler construction |
| 3 | SE24 | Software Engineering | ICT | 4 | 3 | true | true | Software development |
| 4 | SN31 | System and Network Security | ICT | 4 | 3 | true | true | Security principles |
| 5 | NL22 | Natural Language Processing | ICT | 4 | 3 | true | false | NLP techniques |
| 6 | DM21 | Data Mining and Visualization | ICT | 4 | 3 | true | true | Data analysis |
| 7 | QC05 | Quantum Computing | MNC | 4 | 3 | true | true | Quantum principles |
| 8 | AI37 | Introduction to AI | MNC | 4 | 3 | true | false | AI fundamentals |
| 9 | DL35 | Deep Learning | MNC | 4 | 3 | true | true | Neural networks |
| 10 | IR39 | Introduction to Robotics | MNC | 4 | 3 | true | true | Robotics principles |

**Student Enrollment Queries:**

1. Enroll a student in an elective course.

```
64 ∨  INSERT INTO Enrollment (StudentID, CourseID, GradeReceived)
65    VALUES ('202101001', 'AA02', NULL);
66
67
```

Data Output  Messages  Notifications

INSERT 0 1

Query returned successfully in 61 msec.

2. Remove a student from an elective course.

```
766
767 ∨  DELETE FROM Enrollment
768    WHERE StudentID = '202101001' AND CourseID = 'GT01';
```

Data Output  Messages  Notifications

DELETE 0

Query returned successfully in 349 msec.

3. Retrieve all courses a specific student has registered for.

```
770 ∨  SELECT c.CourseID, c.CourseName, e.GradeReceived
771    FROM Enrollment e
772    JOIN Course c ON e.CourseID = c.CourseID
773    WHERE e.StudentID = '202101002';
774
```

Data Output  Messages  Notifications

| | courseid character varying (5) 🔒 | coursename character varying (100) 🔒 | gradereceived character varying (2) 🔒 |
|---|---|---|---|
| 1 | GT01 | Graph Theory and Algorithms | AA |
| 2 | SN31 | System and Network Security | BC |
| 3 | DL35 | Deep Learning | BB |
| 4 | DS38 | Intro to Data Science | AB |

4. Find the number of students enrolled in a specific elective course.

```
775  SELECT c.CourseName, COUNT(e.StudentID) AS EnrollmentCount
776  FROM Enrollment e
777  JOIN Course c ON e.CourseID = c.CourseID
778  WHERE e.CourseID = 'GT01'
779  GROUP BY c.CourseName;
```

Data Output   Messages   Notifications

| | coursename<br>character varying (100) | enrollmentcount<br>bigint |
|---|---|---|
| 1 | Graph Theory and Algorithms | 9 |

5. Find all students enrolled in a specific elective course.

```
781  SELECT u.Name AS StudentName, s.StudentID
782  FROM Enrollment e
783  JOIN Student s ON e.StudentID = s.StudentID
784  JOIN Users u ON s.StudentID = u.UID
785  WHERE e.CourseID = 'GT01';
```

Data Output   Messages   Notifications

| | studentname<br>character varying (50) | studentid<br>character varying (10) |
|---|---|---|
| 1 | Diya Sharma | 202101002 |
| 2 | Ananya Reddy | 202104001 |
| 3 | Krish Malhotra | 202104002 |
| 4 | Ishaan Kumar | 202111001 |
| 5 | Aanya Verma | 202201001 |
| 6 | Aditi Nair | 202211001 |
| 7 | Advait Trivedi | 202301001 |
| 8 | Kabir Srinivasan | 202311001 |
| 9 | Pari Gupta | 202401001 |

6. Retrieve all elective courses available for a particular semester.

```sql
787  SELECT DISTINCT c.CourseID, c.CourseName
788  FROM Offered o
789  JOIN Course c ON o.CourseID = c.CourseID
790  WHERE o.Year = 2024 AND o.Semester = 1;
```

Data Output   Messages   Notifications

| | courseid [PK] character varying (5) | coursename character varying (100) |
|---|---|---|
| 1 | AA02 | Approximation Algorithms |
| 2 | AM08 | Adversarial Machine Learning |
| 3 | BC07 | Blockchain and Cryptocurrencies |
| 4 | DI12 | Digital Image Processing |
| 5 | GT01 | Graph Theory and Algorithms |

7. Retrieve students who have not enrolled in any elective course.

```sql
792  SELECT u.Name AS StudentName, s.StudentID
793  FROM Student s
794  JOIN Users u ON s.StudentID = u.UID
795  LEFT JOIN Enrollment e ON s.StudentID = e.StudentID
796  WHERE e.StudentID IS NULL;
```

Data Output   Messages   Notifications

| studentname character varying (50) | studentid character varying (10) |
|---|---|

**Faculty Related Queries:**

1. Assign an instructor to a specific elective course.

```
738 v  INSERT INTO Offered (CourseID, ProfID, Year, Semester, Intake)
739    VALUES ('AA02', 'MAJO', 2024, 1, 60);
740
```

Data Output   Messages   Notifications

```
INSERT 0 1

Query returned successfully in 63 msec.
```

2. Retrieve all elective courses taught by a specific instructor.

```
740
741 v  SELECT c.CourseID, c.CourseName, o.Year, o.Semester
742    FROM Offered o
743    JOIN Course c ON o.CourseID = c.CourseID
744    WHERE o.ProfID = 'MAJO';
```

Data Output   Messages   Notifications

| | courseid<br>character varying (5) | coursename<br>character varying (100) | year<br>integer | semester<br>integer |
|---|---|---|---|---|
| 1 | GT01 | Graph Theory and Algorithms | 2024 | 1 |
| 2 | GT01 | Graph Theory and Algorithms | 2023 | 1 |
| 3 | GT01 | Graph Theory and Algorithms | 2022 | 1 |
| 4 | GT01 | Graph Theory and Algorithms | 2021 | 1 |

3. Find the instructor assigned to a specific elective course.

```
746 v  SELECT u.Name AS ProfessorName
747    FROM Offered o
748    JOIN Professor p ON o.ProfID = p.ProfID
749    JOIN Users u ON p.ProfID = u.UID
750    WHERE o.CourseID = 'GT01' AND o.Year = 2024 AND o.Semester = 1;
```

Data Output   Messages   Notifications

| | professorname<br>character varying (50) |
|---|---|
| 1 | Manjunath V. Joshi |

4. Find all elective courses that do not have an assigned instructor.

```
752 ∨  SELECT c.CourseID, c.CourseName
753     FROM Course c
754     LEFT JOIN Offered o ON c.CourseID = o.CourseID
755     WHERE o.ProfID IS NULL;
756
757
758
```

Data Output   Messages   Notifications

| | courseid [PK] character varying (5) | coursename character varying (100) |
|---|---|---|
| 1 | OB53 | Organisational Behaviour |
| 2 | EP43 | Einstein Physics |
| 3 | CN40 | Introduction to Complex Network |
| 4 | NC23 | Natural Computing |
| 5 | AI47 | Analog IC Design |
| 6 | OP25 | Optimization |
| 7 | MC30 | Models of Computation |
| 8 | AP50 | Art: Ideas and Perspectives |
| 9 | NT42 | Computational Number Theory |
| 10 | SP28 | Software Project Management |
| 11 | MS09 | Machine Learning and Security |
| 12 | NS32 | No SQL Database |
| 13 | RS26 | Recommendation Systems |

5. Retrieve a list of instructors who have not been assigned any elective courses.

```
757 ∨  SELECT u.Name AS ProfessorName
758     FROM Professor p
759     JOIN Users u ON p.ProfID = u.UID
760     LEFT JOIN Offered o ON p.ProfID = o.ProfID
761     WHERE o.ProfID IS NULL;
```

Data Output   Messages   Notifications

| | professorname character varying (50) |
|---|---|
| 1 | Arpit Rana |
| 2 | Mukesh Tiwari |
| 3 | Amit Mankodi |
| 4 | Ratna Bharati Bhamidipati |
| 5 | Minal Bhise |
| 6 | Biswajit Mishra |
| 7 | Nabin Kumar Sahu |
| 8 | P S Kalyan Sasidhar |
| 9 | Jenson Joseph |
| 10 | Manoj Raut |
| 11 | Sunitha V |
| 12 | Tathagata Bandyopadhy... |

**Course Related Queries:**

1. Retrieve the total number of elective courses offered in a semester.

```
798 ∨  SELECT COUNT(DISTINCT CourseID) AS TotalElectiveCourses
799     FROM Offered
800     WHERE Year = 2024 AND Semester = 1;
```

Data Output   Messages   Notifications

| | totalelectivecourses<br>bigint |
|---|---|
| 1 | 5 |

2. Retrieve the total number of students enrolled in electives across all departments.

```
802 ∨  SELECT COUNT(DISTINCT StudentID) AS TotalStudentsEnrolled
803     FROM Enrollment;
```

Data Output   Messages   Notifications

| | totalstudentsenrolled<br>bigint |
|---|---|
| 1 | 27 |

3. Retrieve the average number of students per elective course.

```
805 ∨  SELECT ROUND(AVG(student_count), 0) AS AvgStudentsPerCourse
806     FROM (
807         SELECT CourseID, COUNT(StudentID) AS student_count
808         FROM Enrollment
809         GROUP BY CourseID
810     ) AS course_enrollments;
```

Data Output   Messages   Notifications

| | avgstudentspercourse<br>numeric |
|---|---|
| 1 | 3 |

4. Find students who have registered for the maximum number of electives allowed.

```sql
812 v  SELECT s.StudentID, u.Name, COUNT(e.CourseID) AS ElectivesTaken
813    FROM Enrollment e
814    JOIN Student s ON e.StudentID = s.StudentID
815    JOIN Users u ON s.StudentID = u.UID
816    GROUP BY s.StudentID, u.Name
817    ORDER BY ElectivesTaken DESC
818    LIMIT 10;
```

Data Output  Messages  Notifications

| | studentid<br>character varying (10) | name<br>character varying (50) | electivestaken<br>bigint |
|---|---|---|---|
| 1 | 202101001 | Aarav Patel | 4 |
| 2 | 202201001 | Aanya Verma | 4 |
| 3 | 202201002 | Arjun Joshi | 4 |
| 4 | 202301002 | Kiara Mehta | 4 |
| 5 | 202301001 | Advait Trivedi | 4 |
| 6 | 202101002 | Diya Sharma | 4 |
| 7 | 202203001 | Myra Bhatt | 3 |
| 8 | 202211001 | Aditi Nair | 3 |
| 9 | 202203002 | Reyansh Desai | 3 |
| 10 | 202303001 | Shaurya Rao | 3 |

5. List course with labs / tutorials.

```sql
820 v  SELECT
821        CourseID,
822        CourseName,
823        CASE WHEN Tutorial THEN 'Yes' ELSE 'No' END AS HasTutorial,
824        CASE WHEN Labs THEN 'Yes' ELSE 'No' END AS HasLab
825    FROM Course
826    WHERE Tutorial = TRUE OR Labs = TRUE;
```

Data Output  Messages  Notifications

| | courseid<br>[PK] character varying (5) | coursename<br>character varying (100) | hastutorial<br>text | haslab<br>text |
|---|---|---|---|---|
| 1 | GT01 | Graph Theory and Algorithms | Yes | No |
| 2 | CC03 | Computational Complexity | Yes | No |
| 3 | CD11 | Compiler Design | Yes | Yes |
| 4 | SE24 | Software Engineering | Yes | Yes |
| 5 | SN31 | System and Network Security | Yes | Yes |
| 6 | NL22 | Natural Language Processing | Yes | No |
| 7 | DM21 | Data Mining and Visualization | Yes | Yes |
| 8 | QC05 | Quantum Computing | Yes | Yes |

**Rating Related Queries :**

    1.  Top rated course with professor list in a particular semester and year

Query   Query History

```sql
554 ∨  SELECT
555         c.CourseName,
556         p.Name AS ProfessorName,
557         ROUND(AVG(r.Overall_Rating)::numeric, 2) AS AverageRating
558     FROM Rating r
559     JOIN Course c ON r.CourseID = c.CourseID
560     JOIN Professor pr ON r.ProfID = pr.ProfID
561     JOIN Users p ON pr.ProfID = p.UID
562     JOIN Offered o ON r.CourseID = o.CourseID AND r.ProfID = o.ProfID
563     WHERE o.Year = 2023 AND o.Semester = 1
564     GROUP BY c.CourseName, p.Name
565     ORDER BY AverageRating DESC
566     LIMIT 2;
```

Data Output   Messages   Notifications

| | coursename<br>character varying (100) | professorname<br>character varying (50) | averagerating<br>numeric |
|---|---|---|---|
| 1 | Graph Theory and Algorithms | Manjunath V. Joshi | 7.86 |
| 2 | Quantum Computing | Pratim Roy | 5.00 |

    2.  Most Difficult Course List

```sql
592 ∨  SELECT
593         c.CourseName,
594         ROUND(AVG(r.Course_Difficulty)::numeric, 2) AS AvgDifficulty
595     FROM Rating r
596     JOIN Course c ON r.CourseID = c.CourseID
597     GROUP BY c.CourseName
598     ORDER BY AvgDifficulty DESC
599     LIMIT 10;
600
```

Data Output   Messages   Notifications

| | coursename<br>character varying (100) | avgdifficulty<br>numeric |
|---|---|---|
| 1 | Quantum Computing | 9.00 |
| 2 | Introduction to Cryptography | 8.00 |
| 3 | Wireless System Design | 8.00 |
| 4 | Deep Learning | 8.00 |
| 5 | Operating Systems | 7.00 |
| 6 | Modern Algebra | 6.67 |

### 3. Most Useful Courses

```
601 ∨   SELECT c.CourseName, COUNT(p.CourseID) AS NumOfDependentCourses
602      FROM Prerequisite p
603      JOIN Course c ON p.Preq_CourseID = c.CourseID
604      GROUP BY c.CourseName
605      ORDER BY NumOfDependentCourses DESC
606      LIMIT 5;
```

Data Output  Messages  Notifications

| | coursename<br>character varying (100) | numofdependentcourses<br>bigint |
|---|---|---|
| 1 | Graph Theory and Algorithms | 4 |
| 2 | Introduction to AI | 3 |
| 3 | Modern Algebra | 3 |
| 4 | Digital Signal Processing | 2 |
| 5 | Intro to Data Science | 1 |

### 4. Course offering average grade above 8

```
SELECT
    c.CourseName,
    ROUND(AVG(
        CASE
            WHEN e.GradeReceived = 'AA' THEN 10
            WHEN e.GradeReceived = 'AB' THEN 9
            WHEN e.GradeReceived = 'BB' THEN 8
            WHEN e.GradeReceived = 'BC' THEN 7
            WHEN e.GradeReceived = 'CC' THEN 6
            WHEN e.GradeReceived = 'CD' THEN 5
            WHEN e.GradeReceived = 'DD' THEN 4
            WHEN e.GradeReceived = 'DE' THEN 3
            WHEN e.GradeReceived = 'F' THEN 0
            ELSE NULL
        END
    )::numeric, 2) AS AvgGrade
FROM Enrollment e
JOIN Course c ON e.CourseID = c.CourseID
GROUP BY c.CourseName
HAVING AVG(
    CASE
        WHEN e.GradeReceived = 'AA' THEN 10
        WHEN e.GradeReceived = 'AB' THEN 9
        WHEN e.GradeReceived = 'BB' THEN 8
        WHEN e.GradeReceived = 'BC' THEN 7
        WHEN e.GradeReceived = 'CC' THEN 6
        WHEN e.GradeReceived = 'CD' THEN 5
        WHEN e.GradeReceived = 'DD' THEN 4
```

```
635              WHEN e.GradeReceived = 'DD' THEN 4
636              WHEN e.GradeReceived = 'DE' THEN 3
637              WHEN e.GradeReceived = 'F' THEN 0
638            ELSE NULL
639          END
640      ) > 8
641    ORDER BY AvgGrade DESC;
```

Data Output   Messages   Notifications

| | coursename character varying (100) | avggrade numeric |
|---|---|---|
| 1 | Logic for Computer Science | 10.00 |
| 2 | Quantum Computing | 9.00 |
| 3 | Randomized Algorithms | 9.00 |
| 4 | Introduction to Robotics | 9.00 |
| 5 | Digital Signal Processing | 9.00 |
| 6 | Modern Algebra | 9.00 |
| 7 | Compiler Design | 8.75 |

5. All ratings given by a particular student

```
644
645  SELECT c.CourseName, u.Name AS ProfessorName,
646         r.Course_Difficulty, r.Tutorial_Lab_Difficulty,
647         r.Project_Difficulty, r.Overall_Rating
648  FROM Rating r
649  JOIN Course c ON r.CourseID = c.CourseID
650  JOIN Professor p ON r.ProfID = p.ProfID
651  JOIN Users u ON p.ProfID = u.UID
652  WHERE r.StudentID = '202101001'
653  ORDER BY r.Overall_Rating DESC;
654
655
656
657
658
659
```

Data Output   Messages   Notifications

Showing rows: 1 to 3

| | coursename character varying (100) | professorname character varying (50) | course_difficulty integer | tutorial_lab_difficulty integer | project_difficulty integer | overall_rating integer |
|---|---|---|---|---|---|---|
| 1 | Graph Theory and Algorithms | Manjunath V. Joshi | 6 | 5 | 7 | 8 |
| 2 | Modern Algebra | Gopinath Panda | 8 | 6 | [null] | 7 |
| 3 | Introduction to AI | Prasenjit Majumder | 7 | 6 | 8 | 7 |

## 6. All ratings of a particular course with its average rating

```sql
657  SELECT
658      c.CourseName,
659      u.Name AS ProfessorName,
660      r.Course_Difficulty,
661      r.Tutorial_Lab_Difficulty,
662      r.Project_Difficulty,
663      r.Overall_Rating
664  FROM Rating r
665  JOIN Course c ON r.CourseID = c.CourseID
666  JOIN Professor p ON r.ProfID = p.ProfID
667  JOIN Users u ON p.ProfID = u.UID
668  WHERE r.CourseID = 'GT01'
669
670  UNION ALL
671
672  -- Average rating for course GT01 (rounded to 2 decimal places)
673  SELECT
674      'Average for GT01' AS CourseName,
675      NULL AS ProfessorName,
676      NULL AS Course_Difficulty,
677      NULL AS Tutorial_Lab_Difficulty,
678      NULL AS Project_Difficulty,
679      ROUND(AVG(Overall_Rating)::numeric, 2) AS Overall_Rating
680  FROM Rating
681  WHERE CourseID = 'GT01'
682
683  ORDER BY ProfessorName NULLS LAST;
```

| | coursename<br>character varying | professorname<br>character varying | course_difficulty<br>integer | tutorial_lab_difficulty<br>integer | project_difficulty<br>integer | overall_rating<br>numeric |
|---|---|---|---|---|---|---|
| 1 | Graph Theory and Algorithms | Manjunath V. Joshi | 4 | 3 | 5 | 10 |
| 2 | Graph Theory and Algorithms | Manjunath V. Joshi | 7 | 6 | 8 | 7 |
| 3 | Graph Theory and Algorithms | Manjunath V. Joshi | 6 | 5 | 7 | 8 |
| 4 | Graph Theory and Algorithms | Manjunath V. Joshi | 8 | 7 | 9 | 6 |
| 5 | Graph Theory and Algorithms | Manjunath V. Joshi | 7 | 6 | 8 | 7 |
| 6 | Graph Theory and Algorithms | Manjunath V. Joshi | 5 | 4 | 6 | 9 |
| 7 | Graph Theory and Algorithms | Manjunath V. Joshi | 6 | 5 | 7 | 8 |
| 8 | Average for GT01 | [null] | [null] | [null] | [null] | 7.86 |

## 7. Three most loved professors

```
684 ∨  SELECT
685         u.Name AS ProfessorName,
686         ROUND(AVG(r.Overall_Rating)::numeric, 2) AS AvgRating
687     FROM Rating r
688     JOIN Professor p ON r.ProfID = p.ProfID
689     JOIN Users u ON p.ProfID = u.UID
690     GROUP BY u.Name
691     ORDER BY AvgRating DESC
692     LIMIT 3;
```

Data Output    Messages    Notifications

| | professorname<br>character varying (50) | avgrating<br>numeric |
|---|---|---|
| 1 | P M Jat | 9.00 |
| 2 | Prasenjit Majumder | 8.33 |
| 3 | Manjunath V. Joshi | 7.86 |

## 8. Number of students with a particular grade in a specified course

Query    Query History

```
694 ∨  SELECT
695         GradeReceived,
696         COUNT(*) AS NumberOfStudents
697     FROM Enrollment
698     WHERE CourseID = 'GT01' AND GradeReceived IS NOT NULL
699     GROUP BY GradeReceived
700     ORDER BY
701         CASE
702             WHEN GradeReceived = 'AA' THEN 1
703             WHEN GradeReceived = 'AB' THEN 2
704             WHEN GradeReceived = 'BB' THEN 3
705             WHEN GradeReceived = 'BC' THEN 4
706             WHEN GradeReceived = 'CC' THEN 5
707             WHEN GradeReceived = 'CD' THEN 6
708             WHEN GradeReceived = 'DD' THEN 7
709             WHEN GradeReceived = 'DE' THEN 8
710             WHEN GradeReceived = 'F' THEN 9
711             ELSE 10
712         END;
713
```

Data Output    Messages    Notifications

| | gradereceived<br>character varying (2) | numberofstudents<br>bigint |
|---|---|---|
| 1 | AA | 5 |
| 2 | AB | 2 |
| 3 | BB | 1 |
| 4 | F | 1 |

**SKILLS and PreRequisite Related Queries :**

1. Skills acquired from a particular course

```
714 v  SELECT s.Skill_name
715    FROM Required r
716    JOIN Skills s ON r.Skill_ID = s.Skill_ID
717    WHERE r.CourseID = 'GT01';
```

Data Output    Messages    Notifications

| | skill_name<br>character varying (40) 🔒 |
|---|---|
| 1 | Algorithm Design |
| 2 | Mathematical Modeling |

2. Prerequisite required for a particular course

```
718
719 v  SELECT c.CourseName
720    FROM Prerequisite p
721    JOIN Course c ON p.Preq_CourseID = c.CourseID
722    WHERE p.CourseID = 'AA02';
```

Data Output    Messages    Notifications

| | coursename<br>character varying (100) 🔒 |
|---|---|
| 1 | Graph Theory and Algorithms |

3. List Courses based on Particular Skill

```
724 v  SELECT c.CourseName
725    FROM Required r
726    JOIN Course c ON r.CourseID = c.CourseID
727    JOIN Skills s ON r.Skill_ID = s.Skill_ID
728    WHERE s.Skill_name = 'Algorithm Design';
```

Data Output    Messages    Notifications

| | coursename<br>character varying (100) 🔒 |
|---|---|
| 1 | Graph Theory and Algorithms |

**Specialization Related Queries :**

1. Professors with most specialization:

```sql
730  SELECT u.Name AS ProfessorName, COUNT(ps.Specialization) AS NumOfSpecializations
731  FROM Professor_Specialization ps
732  JOIN Professor p ON ps.ProfID = p.ProfID
733  JOIN Users u ON p.ProfID = u.UID
734  GROUP BY u.Name
735  ORDER BY NumOfSpecializations DESC
736  LIMIT 5;
737
738
```

Data Output   Messages   Notifications

| | professorname<br>character varying (50) | numofspecializations<br>bigint |
|---|---|---|
| 1 | Bharani Kollipara | 2 |
| 2 | Tapas Kumar Maiti | 2 |
| 3 | Vinay Palaparthy | 2 |
| 4 | Madhumita Mazumdar | 2 |
| 5 | Prosenjit Kundu | 2 |