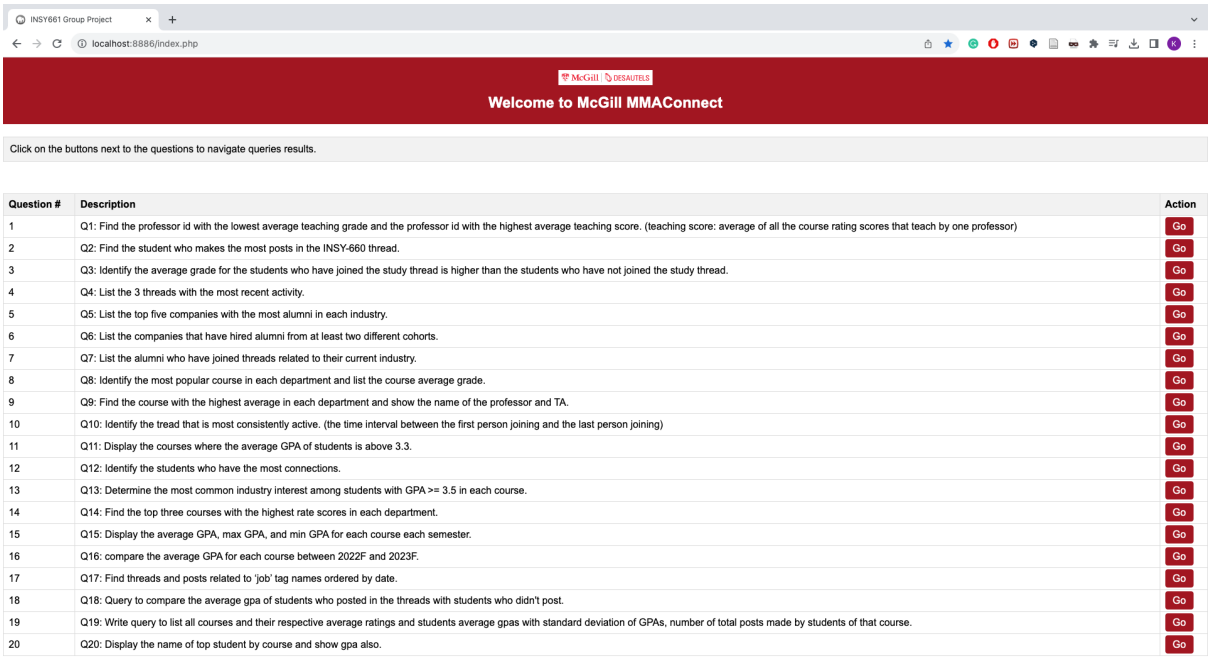
MMAConnect Sample Questions Queries And Screenshot Demo Output:

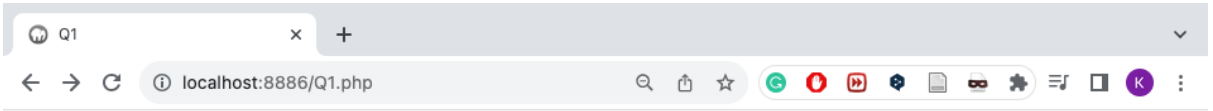


Question #	Description	Action
1	Q1: Find the professor id with the lowest average teaching grade and the professor id with the highest average teaching score. (teaching score: average of all the course rating scores that teach by one professor)	Go
2	Q2: Find the student who makes the most posts in the INSY-660 thread.	Go
3	Q3: Identify the average grade for the students who have joined the study thread is higher than the students who have not joined the study thread.	Go
4	Q4: List the 3 threads with the most recent activity.	Go
5	Q5: List the top five companies with the most alumni in each industry.	Go
6	Q6: List the companies that have hired alumni from at least two different cohorts.	Go
7	Q7: List the alumni who have joined threads related to their current industry.	Go
8	Q8: Identify the most popular course in each department and list the course average grade.	Go
9	Q9: Find the course with the highest average in each department and show the name of the professor and TA.	Go
10	Q10: Identify the tread that is most consistently active. (the time interval between the first person joining and the last person joining)	Go
11	Q11: Display the courses where the average GPA of students is above 3.3.	Go
12	Q12: Identify the students who have the most connections.	Go
13	Q13: Determine the most common industry interest among students with GPA >= 3.5 in each course.	Go
14	Q14: Find the top three courses with the highest rate scores in each department.	Go
15	Q15: Display the average GPA, max GPA, and min GPA for each course each semester.	Go
16	Q16: compare the average GPA for each course between 2022F and 2023F.	Go
17	Q17: Find threads and posts related to 'job' tag names ordered by date.	Go
18	Q18: Query to compare the average gpa of students who posted in the threads with students who didn't post.	Go
19	Q19: Write query to list all courses and their respective average ratings and students average gpas with standard deviation of GPAs, number of total posts made by students of that course.	Go
20	Q20: Display the name of top student by course and show gpa also.	Go

Q1：Find the professor id with the lowest average teaching grade and the professor id with the highest average teaching score. (teaching score: average of all the course rating scores that teach by one professor)

-lowest-
```
SELECT fact_id, AVG(rate_score)
FROM Rates r, Courses c, Teach t
WHERE r.dept_id=c.dept_id AND r.course_id=c.course_id AND t.dept_id=c.dept_id
AND t.course_id=c.course_id
GROUP BY fact_id
ORDER BY AVG(rate_score)
LIMIT 1;
```

-highest-
```
SELECT fact_id, AVG(rate_score)
FROM Rates r, Courses c, Teach t
WHERE r.dept_id=c.dept_id AND r.course_id=c.course_id AND t.dept_id=c.dept_id
AND t.course_id=c.course_id
GROUP BY fact_id
ORDER BY AVG(rate_score) DESC
LIMIT 1;
```



Q1： Find the professor id with the lowest average teaching grade and the professor id with the highest average teaching score. (teaching score: average of all the course rating scores that teach by one professor)

Lowest Teaching Score

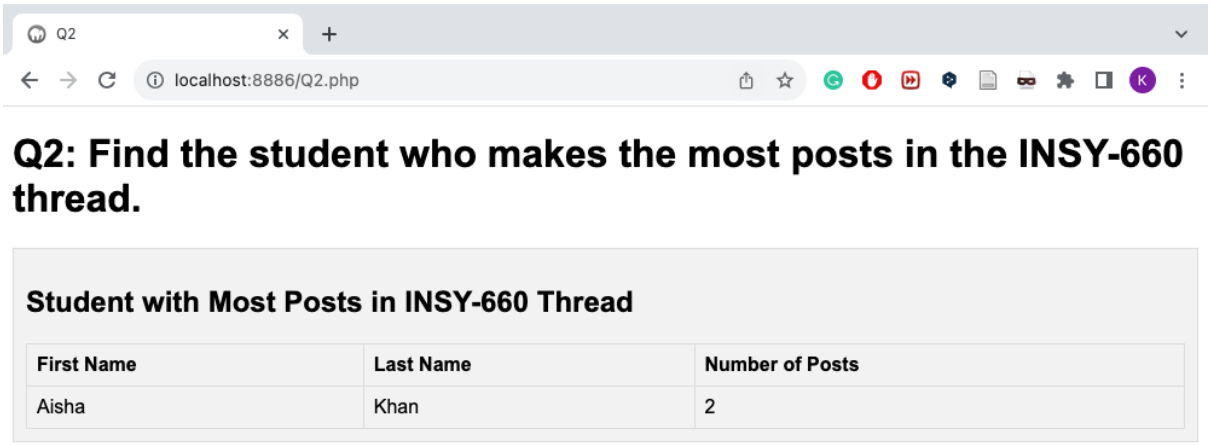
Faculty ID	Average Rate Score
4	6.71

Highest Teaching Score

Faculty ID	Average Rate Score
5	8.75

Q2: Find the student who makes the most posts in the INSY-660 thread.

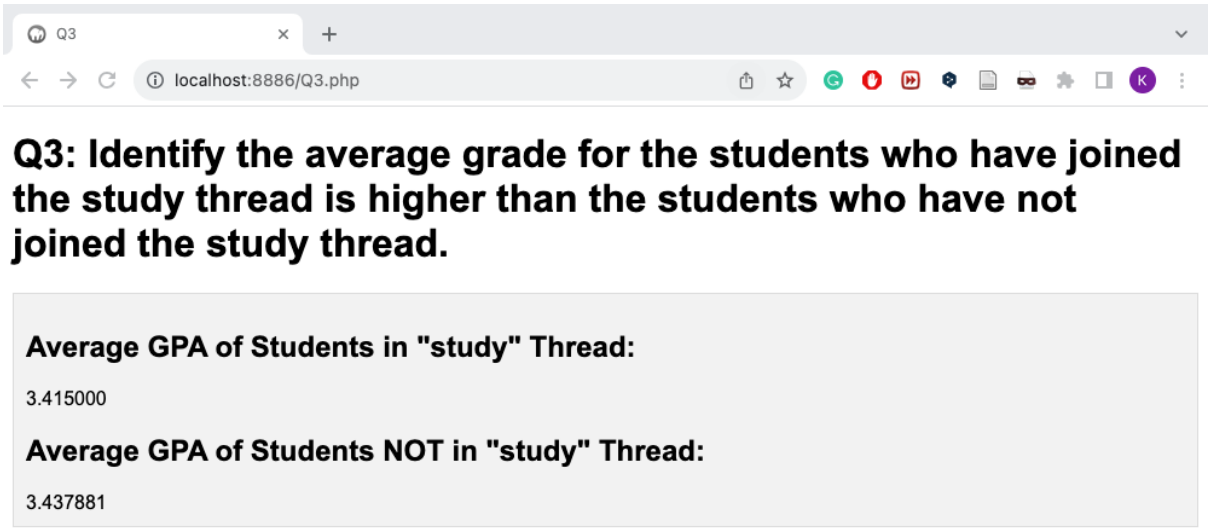
```
SELECT S.stud_fname, S.stud_lname, COUNT(*) AS num_post
FROM Threads T, Posts P, Students S
WHERE T.thread_id=P.thread_id AND S.stud_id=P.stud_id AND
T.thread_title='INSY-660'
GROUP BY P.stud_id
ORDER BY num_post DESC
LIMIT 1;
```



Q3: Identify the average grade for the students who have joined the study thread is higher than the students who have not joined the study thread.

```
SELECT AVG(stud_gpa) AS avg_gpa_in_study_thread
FROM Students ss, Stud_join sj, Threads t, Study sy
WHERE ss.stud_id=sj.stud_id AND sj.thread_id=t.thread_id AND
ss.stud_id=sy.stud_id AND t.thread_title='study';
```

```
SELECT AVG(stud_gpa) AS avg_gpa_notin_study_thread
FROM Study
WHERE stud_id NOT IN (SELECT DISTINCT ss.stud_id
FROM Students ss, Stud_join sj, Threads t, Study sy
WHERE ss.stud_id=sj.stud_id AND sj.thread_id=t.thread_id AND
ss.stud_id=sy.stud_id AND t.thread_title='study');
```



Q4. List the 3 threads with the most recent activity.

```
SELECT thread_title, MAX(post_timestamp) AS latest_activity
FROM Threads T JOIN Posts P ON P.thread_id=T.thread_id
GROUP BY T.thread_id
ORDER BY latest_activity DESC
LIMIT 3;
```

Q4. List the 3 threads with the most recent activity.

Thread Title	Latest Activity
general_chat	2023-08-19 19:37:00
INSY-660	2023-08-19 19:33:00
INSY-661	2023-08-17 19:37:00

Q5 List the top five companies with the most alumni in each industry

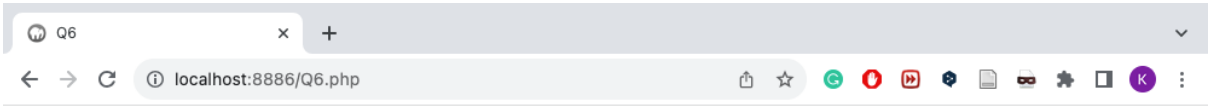
```
SELECT Industries.industry_name AS Industry_Name, Alumni.alum_companyname
AS Company_Name, COUNT(DISTINCT Alumni.alum_id) AS Alumni_Count
    FROM Industries
    JOIN Worked_in ON Industries.industry_id = Worked_in.industry_id
    JOIN Alumni ON Worked_in.alum_id = Alumni.alum_id
    GROUP BY Industries.industry_name, Alumni.alum_companyname
    HAVING Alumni_Count >= 2
    ORDER BY Industries.industry_name, Alumni_Count DESC
    LIMIT 5;
```

Q5 List the top five companies with the most alumni in each industry

Industry Name	Company Name	Alumni Count
Automotive	PQR Co	2
Education	RST Group	2
Energy	ABC Corp	2
Finance	LMN Ltd	2
Healthcare	XYZ Inc	2

Q6 List the companies that have hired alumni from at least two different cohorts

```
SELECT alum_companyname AS Company, COUNT(DISTINCT mmaCo_id) AS
Alumni_Hired
    FROM Alumni
    GROUP BY alum_companyname
    HAVING Alumni_Hired >= 2;
```

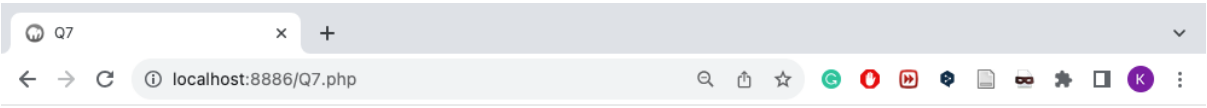


Q6 List the companies that have hired alumni from at least two different cohorts

Company	Alumni Hired
ABC Corp	4
LMN Ltd	4
PQR Co	4
RST Group	4
XYZ Inc	4

Q7: List the alumni who have joined threads related to their current industry

```
SELECT CONCAT(Alumni.alum_fname, ' ', Alumni.alum_lname) AS Alumni_Name,
        Alumni.alum_companyname AS Company_Name,
        Industries.industry_name AS Industry,
        Threads.thread_title AS Thread_Title
    FROM Alumni
    JOIN Worked_in ON Alumni.alum_id = Worked_in.alum_id
    JOIN Industries ON Worked_in.industry_id = Industries.industry_id
    JOIN Alum_join ON Alumni.alum_id = Alum_join.alum_id
    JOIN Threads ON Alum_join.thread_id = Threads.thread_id
    ORDER BY Alumni_Name;
```



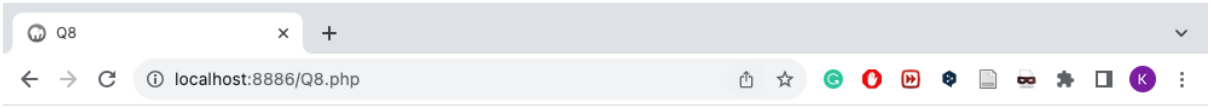
Q7: List the alumni who have joined threads related to their current industry.

Alumni Name	Company Name	Industry	Thread Title
Ava Hernandez	XYZ Inc	Telecommunications	study
Emma Garcia	RST Group	Education	Finance
Emma Garcia	RST Group	Education	Manufacturing
Emma Garcia	RST Group	Education	Entertainment
Emma Harris	ABC Corp	Finance	ORGB-660
Emma Martinez	RST Group	Retail	ORGB-661
Emma Martinez	PQR Co	Telecommunications	Technology
Emma Martinez	PQR Co	Telecommunications	Retail
Emma Martinez	PQR Co	Telecommunications	Telecommunications
Isabella Hernandez	XYZ Inc	Healthcare	Finance
Isabella Hernandez	XYZ Inc	Healthcare	Automotive
Isabella Hernandez	XYZ Inc	Healthcare	Entertainment
Isabella Rodriguez	PQR Co	Entertainment	Retail
Isabella Rodriguez	PQR Co	Entertainment	Telecommunications
Isabella Rodriguez	PQR Co	Entertainment	Energy
Isabella Wilson	LMN Ltd	Manufacturing	events
Jackson Lee	LMN Ltd	Healthcare	MGSC-660
James Garcia	RST Group	Technology	Healthcare
James Garcia	RST Group	Technology	Finance
James Garcia	RST Group	Technology	Manufacturing
James Lee	LMN Ltd	Finance	Automotive
James Lee	LMN Ltd	Finance	Retail
James Lee	LMN Ltd	Finance	Telecommunications
James Lopez	LMN Ltd	Telecommunications	Education
James Lopez	LMN Ltd	Telecommunications	Energy
James Taylor	ABC Corp	Entertainment	mtl_exploration
Liam Martin	XYZ Inc	Automotive	ORGB-661
Liam Miller	PQR Co	Automotive	ORGB-660
Liam Walker	RST Group	Education	events
Liam Walker	RST Group	Education	Healthcare
Noah Anderson	XYZ Inc	Manufacturing	Automotive
Noah Anderson	XYZ Inc	Manufacturing	Entertainment

James Lee	LMN Ltd	Finance	Automotive
James Lee	LMN Ltd	Finance	Retail
James Lee	LMN Ltd	Finance	Telecommunications
James Lopez	LMN Ltd	Telecommunications	Education
James Lopez	LMN Ltd	Telecommunications	Energy
James Taylor	ABC Corp	Entertainment	mtl_exploration
Liam Martin	XYZ Inc	Automotive	ORGB-661
Liam Miller	PQR Co	Automotive	ORGB-660
Liam Walker	RST Group	Education	events
Liam Walker	RST Group	Education	Healthcare
Noah Anderson	XYZ Inc	Manufacturing	Automotive
Noah Anderson	XYZ Inc	Manufacturing	Entertainment
Noah Anderson	XYZ Inc	Manufacturing	Telecommunications
Noah Davis	UVW Solutions	Education	general_chat
Noah Smith	LMN Ltd	Entertainment	study
Noah Smith	LMN Ltd	Entertainment	Automotive
Noah Smith	LMN Ltd	Entertainment	Entertainment
Oliver Adams	ABC Corp	Energy	Healthcare
Oliver Adams	ABC Corp	Energy	Education
Oliver Adams	ABC Corp	Energy	Energy
Oliver Garcia	PQR Co	Energy	INSY-660
Oliver Williams	ABC Corp	Retail	Healthcare
Oliver Williams	ABC Corp	Retail	Education
Oliver Williams	ABC Corp	Retail	Manufacturing
Olivia Brown	LMN Ltd	Finance	MGSC-660
Olivia Lewis	PQR Co	Retail	general_chat
Olivia Perez	XYZ Inc	Manufacturing	mtl_exploration
Olivia Perez	XYZ Inc	Manufacturing	Finance
Olivia Perez	XYZ Inc	Manufacturing	Manufacturing
Sophia Johnson	ABC Corp	Technology	INSY-660
Sophia Johnson	PQR Co	Automotive	Retail
Sophia Johnson	PQR Co	Automotive	Education
Sophia Johnson	PQR Co	Automotive	Energy
Sophia Nguyen	RST Group	Technology	INSY-661
William Williams	XYZ Inc	Healthcare	INSY-661

Q8 Identify the most popular course in each department and list the course average grade.

```
SELECT
    Departments.dept_name AS Department,
    Courses.course_name AS Course_Name,
    CONCAT(Courses.dept_id, Courses.course_id) AS Course_Number,
    COUNT(Study.stud_id) AS Enrollment_Count,
    AVG(Study.stud_gpa) AS Average_Grade
FROM Study
JOIN Courses ON Study.dept_id = Courses.dept_id AND Study.course_id =
Courses.course_id
JOIN Departments ON Study.dept_id = Departments.dept_id
JOIN (
    SELECT dept_id, course_id
    FROM Study
    GROUP BY dept_id, course_id
    HAVING COUNT(*) = (
        SELECT MAX(enrollment_count)
        FROM (
            SELECT dept_id, course_id, COUNT(*) AS enrollment_count
            FROM Study
            GROUP BY dept_id, course_id
        ) AS CoursePop
        WHERE CoursePop.dept_id = Study.dept_id
    )
) AS MaxEnrollment ON Study.dept_id = MaxEnrollment.dept_id AND Study.course_id
= MaxEnrollment.course_id
GROUP BY Department, Course_Number, Course_Name
ORDER BY Department;
```



Q8: Identify the most popular course in each department and list the course average grade.

Department	Course Name	Course Number	Enrollment Count	Average Grade
Artificial Intelligence	Deep Learning Fundamentals	AI603	9	3.74
Business Intelligence	Data Visualization	BI702	10	3.66
Data Base Management System	Database Administration	DBMS303	10	3.35
Finance	Investment Strategies	FIN503	9	3.44
Management	Business Ethics	MGMT403	10	3.57
Software development	Object-Oriented Programming	SWD203	15	3.40
Statistics	Introduction to Statistics	STATS101	30	3.23

Q9 Find the course with the highest average in each department and show the name of the professor and TA.

```
SELECT dept_id, course_id, sem_year, fact_fname, fact_lname
        FROM (
          SELECT s.dept_id, s.course_id, s.sem_year, s.average_grade,
              @rank := IF(@prev_dept = s.dept_id, @rank + 1, 1) AS 'highest',
              @prev_dept := s.dept_id,
              p.fact_fname, p.fact_lname
          FROM (
            SELECT dept_id, course_id, sem_year, AVG(stud_gpa) AS average_grade
            FROM Study
            GROUP BY dept_id, course_id, sem_year
            ORDER BY dept_id, average_grade DESC
          ) s
          LEFT JOIN (
            SELECT t.dept_id, t.course_id, t.sem_year, f.fact_fname, f.fact_lname
            FROM Teach t
            LEFT JOIN Faculties f USING(fact_id)
          ) p ON (p.dept_id = s.dept_id) AND (p.course_id = s.course_id) AND
(p.sem_year = s.sem_year),
          (SELECT @rank := 0, @prev_dept := NULL) r
        ) gg
        WHERE highest = 1;
```

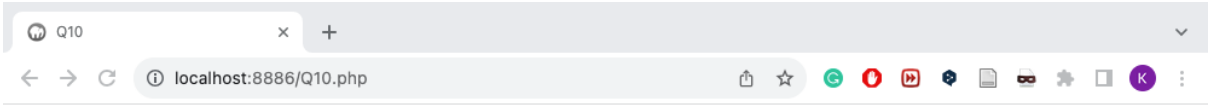
Q9: Find the course with the highest average in each department and show the name of the professor and TA.

Department ID	Course ID	Semester Year	Faculty First Name	Faculty Last Name
AI	601	2023F	Orion	Nightshade
STATS	102	2023F	Aurora	Frost
DBMS	303	2023F	Celeste	Silverleaf
BI	702	2023F	Sophia	Wildwood
SWD	201	2023F	Liam	Evergreen
FIN	502	2023F	Isabella	Ember
MGMT	402	2023F	Jackson	Stormrider

Q10: Identify the tread that is most consistently active. (the time interval between the first person joining and the last person joining)

```
SELECT thread_id, thread_title, time_interval FROM (
   SELECT thread_id, thread_title, TIMESTAMPDIFF(DAY, MIN(join_time),
MAX(join_time)) AS 'time_interval'
   FROM (
      SELECT thread_id, thread_title, fact_join_date_time AS 'join_time', fact_id AS
'member'
      FROM Threads
      LEFT JOIN Fact_join USING(thread_id)
      UNION ALL
      SELECT thread_id, thread_title, alum_join_date_time AS 'join_time', alum_id AS
'member'
      FROM Threads
      LEFT JOIN Alum_join USING(thread_id)
      UNION ALL
      SELECT thread_id, thread_title, stud_join_date_time AS 'join_time', stud_id AS
'member'
      FROM Threads
      LEFT JOIN Stud_join USING(thread_id)
   ) a
   GROUP BY thread_id, thread_title
) f
WHERE time_interval = (
   SELECT MAX(time_interval)
   FROM (
      SELECT thread_id, thread_title, TIMESTAMPDIFF(DAY, MIN(join_time),
MAX(join_time)) AS 'time_interval'
      FROM (
         SELECT thread_id, thread_title, fact_join_date_time AS 'join_time', fact_id AS
'member'
         FROM Threads
         LEFT JOIN Fact_join USING(thread_id)
         UNION ALL
         SELECT thread_id, thread_title, alum_join_date_time AS 'join_time', alum_id AS
'member'
         FROM Threads
         LEFT JOIN Alum_join USING(thread_id)
         UNION ALL
         SELECT thread_id, thread_title, stud_join_date_time AS 'join_time', stud_id AS
'member'
```

```
        FROM Threads
        LEFT JOIN Stud_join USING(thread_id)
      ) a
      GROUP BY thread_id, thread_title
   ) f2
);
```

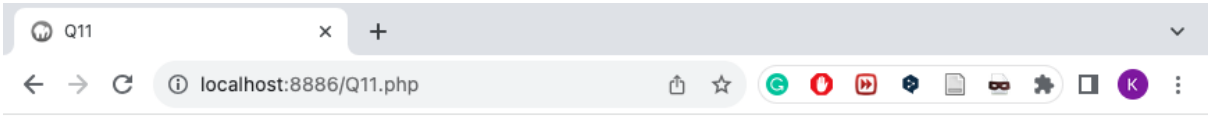


Q10: Identify the tread that is most consistently active. (the time interval between the first person joining and the last person joining)"

Thread ID	Thread Title	Join Time Interval (Days)
15	Education	27
16	Manufacturing	27
17	Entertainment	27
18	Telecommunications	27
19	Energy	27
11	Healthcare	27
12	Finance	27
13	Automotive	27
14	Retail	27

Q11 Display the courses where the average GPA of students is above 3.3.

```
SELECT c.dept_id, c.course_id, c.course_name, s.sem_year,
ROUND(AVG(s.stud_gpa), 2) AS 'average_grade'
    FROM Courses c
    LEFT JOIN Study s ON (c.dept_id = s.dept_id) AND (c.course_id = s.course_id)
    GROUP BY c.dept_id, c.course_id, c.course_name, s.sem_year
    HAVING average_grade >= 3.3
    ORDER BY average_grade DESC";
```

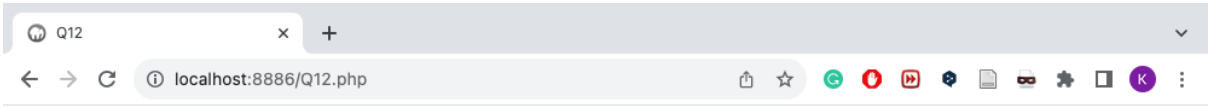


Q11: Display the courses where the average GPA of students is above 3.3.

Department ID	Course ID	Course Name	Semester Year	Average Grade
AI	601	Introduction to AI	2023F	3.80
FIN	502	Financial Analysis	2023F	3.80
AI	603	Deep Learning Fundamentals	2023F	3.74
STATS	102	Advanced Statistics	2023F	3.70
BI	702	Data Visualization	2023F	3.66
FIN	501	Finance Fundamentals	2023F	3.65
MGMT	402	Marketing Principles	2023F	3.60
BI	701	Business Intelligence Concepts	2023F	3.58
MGMT	403	Business Ethics	2023F	3.57
SWD	201	Introduction to Programming	2023F	3.57
DBMS	301	Database Design	2023F	3.46
AI	602	Machine Learning Basics	2023F	3.45
SWD	202	Software Development Fundamentals	2023F	3.45
FIN	503	Investment Strategies	2023F	3.44
SWD	203	Object-Oriented Programming	2023F	3.40
MGMT	401	Introduction to Business	2023F	3.36
DBMS	303	Database Administration	2023F	3.35
DBMS	302	SQL Fundamentals	2023F	3.31

Q12: Identify the students who have the most connections.

```
SELECT stud_id, stud_fname, stud_lname, number_of_connect FROM (
   SELECT stud_id, COUNT(DISTINCT fact_id) + COUNT(DISTINCT alum_id) AS
'number_of_connect'
   FROM Stud_join
   LEFT JOIN Alum_join USING(thread_id)
   LEFT JOIN Fact_join USING(thread_id)
   GROUP BY stud_id
) c
LEFT JOIN Students USING(stud_id)
WHERE number_of_connect = (
   SELECT MAX(number_of_connect)
   FROM (
      SELECT stud_id, COUNT(DISTINCT fact_id) + COUNT(DISTINCT alum_id) AS
'number_of_connect'
      FROM Stud_join
      LEFT JOIN Alum_join USING(thread_id)
      LEFT JOIN Fact_join USING(thread_id)
      GROUP BY stud_id
   ) c1
);
```



Q12: Identify the students who have the most connections.

Student ID	First Name	Last Name	Number of Connections
1	John	Doe	21
2	Aisha	Khan	21
3	Leonardo	Rossi	21
4	Mei	Chen	21
5	Alejandro	Gonzalez	21

Q13: Determine the most common industry interest among students with GPA >= 3.5 in each course.

```
SELECT
   industry_name,
   CONCAT(CAST(ROUND(proportion * 100, 2) AS CHAR), '%') AS
ProportionOfHighGPAStudents,
   ROUND(AvgGPA, 2) AS AvgGPA
FROM (
   SELECT
      Industries.industry_name,
      COUNT(DISTINCT h.stud_id) * 1.0 / (
         SELECT COUNT(DISTINCT Students.stud_id)
         FROM Students
         JOIN Study ON Students.stud_id = Study.stud_id
         WHERE Study.stud_gpa IS NOT NULL AND Study.stud_gpa >= 3.5
      ) AS proportion,
      AVG(h.max_gpa) AS AvgGPA
   FROM (
      SELECT
         Students.stud_id AS stud_id,
         Students.stud_fname,
         Students.stud_lname,
         ROUND(MAX(Study.stud_gpa), 2) AS max_gpa
      FROM
         Students
      JOIN
         Study ON Students.stud_id = Study.stud_id
      WHERE
         Study.stud_gpa IS NOT NULL AND Study.stud_gpa >= 3.5
      GROUP BY
         Students.stud_id,
         Students.stud_fname,
         Students.stud_lname
   ) AS h
   LEFT JOIN Interested_in ON Interested_in.stud_id = h.stud_id
   LEFT JOIN Industries ON Interested_in.industry_id = Industries.industry_id
   WHERE
      Industries.industry_name IS NOT NULL
   GROUP BY Industries.industry_name
) AS InnerQuery
ORDER BY
```

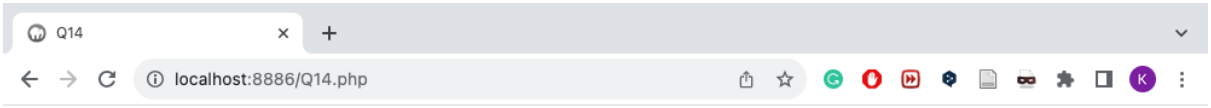
```
InnerQuery.proportion DESC,
InnerQuery.industry_name ASC,
InnerQuery.AvgGPA DESC;
```

Q13: Determine the most common industry interest among students with GPA >= 3.5 in each course.

Industry Name	Proportion of High GPA Students	Average GPA
Automotive	10.00%	3.61
Energy	10.00%	3.86
Entertainment	10.00%	3.72
Finance	7.50%	3.82
Manufacturing	7.50%	3.85
Retail	7.50%	3.72
Telecommunications	7.50%	3.73
Education	5.00%	3.70
Healthcare	5.00%	3.85
Technology	5.00%	3.68

Q14: Find the top three courses with the highest rate scores in each department.

```
SELECT
      d.dept_id,
      d.dept_name,
      r.course_id,
      ROUND(r.avg_rate_score, 2) AS rounded_avg_rate_score
   FROM (
      SELECT
         dept_id,
         course_id,
         AVG(rate_score) AS avg_rate_score,
         (
            SELECT COUNT(DISTINCT r1.course_id)
            FROM Rates r1
            WHERE r1.dept_id = r.dept_id AND AVG(r.rate_score) <= (
               SELECT AVG(r2.rate_score)
               FROM Rates r2
               WHERE r2.dept_id = r1.dept_id AND r2.course_id = r1.course_id
            )
         ) AS ranking
      FROM Rates r
      GROUP BY dept_id, course_id
   ) r
   JOIN Departments d ON r.dept_id = d.dept_id
   WHERE r.ranking <= 3
   ORDER BY d.dept_id, r.avg_rate_score DESC;
```



Q14: Find the top three courses with the highest rate scores in each department.

Department ID	Department Name	Course ID	Average Rating Score(0-10)
AI	Artificial Intelligence	602	8.00
AI	Artificial Intelligence	601	7.67
BI	Business Intelligence	701	8.25
DBMS	Data Base Management System	301	7.64
DBMS	Data Base Management System	302	7.63
FIN	Finance	501	8.75
MGMT	Management	401	7.00
MGMT	Management	402	6.17
STATS	Statistics	101	7.69
STATS	Statistics	102	6.50
SWD	Software development	202	8.13
SWD	Software development	201	7.53

Q15: Display the average GPA for each course each semester.

```
SELECT
    c.course_name,
    s.sem_year,
    ROUND(AVG(s.stud_gpa), 2) AS avg_gpa,
    MAX(s.stud_gpa) AS max_gpa,
    MIN(s.stud_gpa) AS min_gpa
FROM
    Study s
JOIN
    Courses c ON s.dept_id = c.dept_id AND s.course_id = c.course_id
GROUP BY
    c.course_name, s.sem_year
ORDER BY
    c.course_name, s.sem_year;
```

Q15: Display the average GPA, max GPA, and min GPA for each course each semester.

Course Name	Semester Year	Average GPA	Max GPA	Min GPA
Advanced Statistics	2023F	3.70	3.70	3.70
Business Ethics	2023F	3.57	3.90	3.20
Business Intelligence Concepts	2023F	3.58	3.75	3.40
Data Visualization	2023F	3.66	3.92	3.10
Database Administration	2023F	3.35	3.75	3.01
Database Design	2023F	3.46	3.95	2.89
Deep Learning Fundamentals	2023F	3.74	3.95	3.20
Finance Fundamentals	2023F	3.65	3.65	3.65
Financial Analysis	2023F	3.80	3.85	3.75
Introduction to AI	2023F	3.80	3.80	3.80
Introduction to Business	2023F	3.36	3.80	2.98
Introduction to Programming	2023F	3.57	3.95	3.20
Introduction to Statistics	2023F	3.23	3.88	2.74
Investment Strategies	2023F	3.44	3.75	3.15
Machine Learning Basics	2023F	3.45	3.60	3.30
Marketing Principles	2023F	3.60	3.60	3.60
Object-Oriented Programming	2023F	3.40	3.78	3.10
Probability Theory	2023F	3.27	3.65	2.90
Software Development Fundamentals	2023F	3.45	3.50	3.40
SQL Fundamentals	2023F	3.31	4.00	2.70

Q16: compare the average GPA for each course between 2022F and 2023F

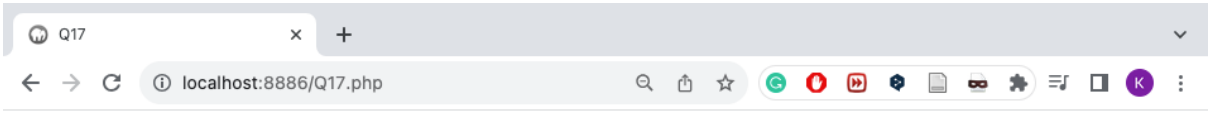
```
SELECT
    s.dept_id,
    c.course_id,
    c.course_name,
    COALESCE(ROUND(AVG(CASE WHEN o.sem_year = '2022F' THEN s.stud_gpa
END), 2), 0) AS avg_gpa_2022F,
    COALESCE(ROUND(AVG(CASE WHEN o.sem_year = '2023F' THEN s.stud_gpa
END), 2), 0) AS avg_gpa_2023F
FROM
    Study s
JOIN
    Courses c ON s.dept_id = c.dept_id AND s.course_id = c.course_id
JOIN
    Offered_by o ON s.dept_id = o.dept_id AND s.course_id = o.course_id
GROUP BY
    s.dept_id,
    c.course_id,
    c.course_name
HAVING
        avg_gpa_2022F > 0
        AND
        avg_gpa_2023F > 0
ORDER BY
    s.dept_id,
    c.course_id;
```

Q16: compare the average GPA for each course between 2022F and 2023F.

Department ID	Course ID	Course Name	Avg GPA 2022F	Avg GPA 2023F
AI	601	Introduction to AI	3.80	3.80
BI	701	Business Intelligence Concepts	3.58	3.58
DBMS	301	Database Design	3.46	3.46
DBMS	302	SQL Fundamentals	3.31	3.31
FIN	501	Finance Fundamentals	3.65	3.65
MGMT	401	Introduction to Business	3.36	3.36
STATS	101	Introduction to Statistics	3.23	3.23
STATS	102	Advanced Statistics	3.70	3.70
SWD	201	Introduction to Programming	3.57	3.57
SWD	202	Software Development Fundamentals	3.45	3.45

Q17: Find threads and posts related to “study” tag names ordered by date.

```
SELECT DISTINCT T.thread_id, T.thread_title, P.post_id, P.post_content,
P.post_timestamp
FROM Threads AS T
JOIN Categorize_by AS CB ON T.thread_id = CB.thread_id
JOIN Tags AS TG ON CB.tag_id = TG.tag_id
JOIN Posts AS P ON T.thread_id = P.thread_id
WHERE TG.tag_name LIKE '%study%'
ORDER BY P.post_timestamp ASC;
```



Q17: Find threads and posts related to study tag names ordered by date.

Thread ID	Thread Title	Post ID	Post Content	Post Timestamp
1	INSY-660	1	Hi guys, is anyone else confused about the final report deliverable for this or has asked the prof already? Cause she posted 2 conflicting requirements. Is the final report the same as the final presentation?	2023-08-19 19:31:00
1	INSY-660	2	there are two presentations. one is the final report and the other is presentaion for class	2023-08-19 19:31:00
1	INSY-660	3	wait, what? I was under the impression that the two are the same. I thought we have to upload the .pptx and present that in class (8+2 mins)	2023-08-19 19:32:00
1	INSY-660	4	final report is not presetnation but just 20 page ppt let me rephrase	2023-08-19 19:33:00

Q18: Query to compare the average gpa of students who posted in the threads with students who didn't post.

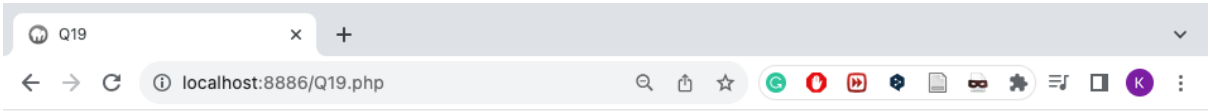
```
SELECT
   X.AvgGpa_StudsWithNoPost,
   Y.AvgGpa_StudsWithPost
FROM
   (SELECT AVG(ST.stud_gpa) as AvgGpa_StudsWithNoPost
   FROM Students AS S
   LEFT JOIN Posts AS P ON S.stud_id = P.stud_id
         LEFT JOIN Threads AS T ON P.thread_id = T.thread_id
         LEFT JOIN Study AS ST ON S.stud_id = ST.stud_id
   WHERE P.stud_id IS NULL) AS X,
   (SELECT AVG(ST.stud_gpa) as AvgGpa_StudsWithPost
   FROM Students AS S
   LEFT JOIN Posts AS P ON S.stud_id = P.stud_id
         LEFT JOIN Threads AS T ON P.thread_id = T.thread_id
         LEFT JOIN Study AS ST ON S.stud_id = ST.stud_id) AS Y;
```

Q18: Query to compare the average gpa of students who posted in the threads with students who didn't post.

Average GPA (Students without posts)	Average GPA (Students with posts)
3.42	3.42

Q19: Write query to list all courses and their respective average ratings and students average gpas with standard deviation of GPAs, number of total posts made by students of that course.

```
SELECT
    C.course_id,
    C.course_name,
    ROUND(AVG(R.rate_score), 2) AS average_rating,
    ROUND(AVG(ST.stud_gpa), 2) AS average_gpa,
    ROUND(STDDEV(ST.stud_gpa), 2) AS gpa_standard_deviation,
    COUNT(P.stud_id) AS total_posts
FROM Courses AS C
LEFT JOIN Rates AS R ON C.course_id = R.course_id
LEFT JOIN Study AS ST ON C.course_id = ST.course_id
LEFT JOIN Posts AS P ON ST.stud_id = P.stud_id
GROUP BY C.course_id, C.course_name
HAVING
    average_rating IS NOT NULL
AND average_gpa IS NOT NULL
AND gpa_standard_deviation IS NOT NULL
AND total_posts IS NOT NULL;
```

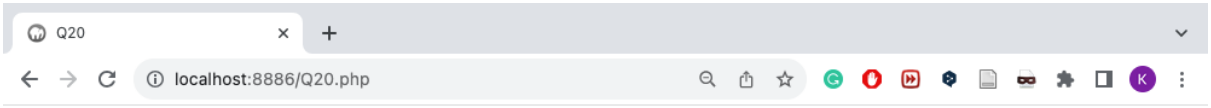


Q19: Write query to list all courses and their respective average ratings and students average gpas with standard deviation of GPAs, number of total posts made by students of that course.

Course ID	Course Name	Average Rating	Average GPA	GPA Standard Deviation	Total Posts
101	Introduction to Statistics	7.69	3.24	0.35	156
102	Advanced Statistics	6.50	3.70	0.00	0
201	Introduction to Programming	7.53	3.51	0.29	75
202	Software Development Fundamentals	8.13	3.45	0.05	0
301	Database Design	7.64	3.56	0.30	44
302	SQL Fundamentals	7.63	3.35	0.10	0
401	Introduction to Business	7.00	3.32	0.30	55
402	Marketing Principles	6.17	3.60	0.00	0
501	Finance Fundamentals	8.75	3.65	0.00	4
601	Introduction to AI	7.67	3.80	0.00	3
602	Machine Learning Basics	8.00	3.45	0.15	0
701	Business Intelligence Concepts	8.25	3.52	0.16	8

Q20: Display the name of top student by course and show gpa also.

```
SELECT
    C.course_id,
    C.course_name,
    ST.sem_year,
    S.stud_fname,
    S.stud_lname,
    ST.stud_gpa
FROM Courses AS C
JOIN Study AS ST ON C.course_id = ST.course_id
JOIN Students AS S ON ST.stud_id = S.stud_id
WHERE (ST.course_id, ST.stud_gpa) IN (
    SELECT course_id, MAX(stud_gpa)
    FROM Study
    GROUP BY course_id
)
ORDER BY C.course_id, ST.sem_year;
```



Q20: Display the name of top student by course and show gpa also.

Course ID	Course Name	Semester Year	Student First Name	Student Last Name	Student GPA
101	Introduction to Statistics	2023F	Rajesh	Patel	3.88
102	Advanced Statistics	2023F	Yasmine	Gonzalez	3.70
103	Probability Theory	2023F	Amina	Khan	3.65
201	Introduction to Programming	2023F	Leila	Rahman	3.95
202	Software Development Fundamentals	2023F	JeanLuc	Chen	3.50
203	Object-Oriented Programming	2023F	Ingrid	Johansson	3.78
203	Object-Oriented Programming	2023F	Daniel	Santos	3.78
301	Database Design	2023F	Aisha	Rossi	3.95
302	SQL Fundamentals	2023F	Leila	Said	3.45
303	Database Administration	2023F	Ayman	Tanaka	3.75
401	Introduction to Business	2023F	Amir	Hussein	3.80
402	Marketing Principles	2023F	Carlos	da Silva	3.60
403	Business Ethics	2023F	Luca	Ferrari	3.90
403	Business Ethics	2023F	Leandro	Rodriguez	3.90
501	Finance Fundamentals	2023F	Fatima	Ahmed	3.65
502	Financial Analysis	2023F	Ivan	Ahmed	3.85
503	Investment Strategies	2023F	Anna	Rahman	3.75
503	Investment Strategies	2023F	Yara	Chen	3.75
601	Introduction to AI	2023F	Alejandro	Gonzalez	3.80
602	Machine Learning Basics	2023F	Sofia	Garcia	3.60
603	Deep Learning Fundamentals	2023F	Sofia	Santos	3.95
701	Business Intelligence Concepts	2023F	Amir	Khan	3.75
702	Data Visualization	2023F	Noura	Ali	3.92
702	Data Visualization	2023F	Amir	Khan	3.92