```
117
118 •  select * from Students;
119 •  update students set gender = 'Male' where student_id = 2001 ;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
2001	Ella Green	2004-05-10	Male	ella.g@email.com	555-2001	404 Oak St	2023-09-01	1
2002	Frank White	2003-11-20	Male	frank.w@email.com	555-2002	505 Pine St	2023-09-01	1
2003	Grace Hall	2005-01-15	Female	grace.h@email.com	555-2003	606 Cedar St	2023-09-01	2
2004	Henry Black	2002-08-25	Male	henry.b@email.com	555-2004	707 Birch St	2023-09-01	3
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
118 •  select * from Students;
119 •  update students set gender = 'Male' where student_id = 2001 ;
120 •  delete from Students  where student_id = 2004 ;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
2001	Ella Green	2004-05-10	Male	ella.g@email.com	555-2001	404 Oak St	2023-09-01	1
2002	Frank White	2003-11-20	Male	frank.w@email.com	555-2002	505 Pine St	2023-09-01	1
2003	Grace Hall	2005-01-15	Female	grace.h@email.com	555-2003	606 Cedar St	2023-09-01	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
123
124     ## Q - 2
125
126 •   SELECT student_id, name
127     FROM Students
```

Result Grid | Filter Rows: | Edit:

student_id	name
2001	Ella Green
2002	Frank White
NULL	NULL

```
129
130 •   SELECT s.student_id, s.name,
131     SUM(g.marks_obtained) AS total_marks
132     FROM Students s
133     JOIN Grades g ON s.student_id = g.student_id
134     GROUP BY s.student_id, s.name
135     ORDER BY total_marks DESC
136     LIMIT 10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	name	total_marks
2001	Ella Green	175.00
2002	Frank White	45.00

```
LIMIT 10;

SELECT s.student_id, s.name,
(SUM(CASE WHEN a.status = 'Present' THEN 1 ELSE 0 END) * 100.0 / COUNT(a.attendance_id)) AS attendance_percentage
FROM Students s
JOIN Attendance a ON s.student_id = a.student_id
GROUP BY s.student_id, s.name
HAVING attendance_percentage < 75;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	name	attendance_percentage

Limit to 1000 rows

```
SELECT s.student_id, s.name
FROM Students s
JOIN Attendance a ON s.student_id = a.student_id
JOIN Grades g ON s.student_id = g.student_id
GROUP BY s.student_id, s.name
HAVING
    (SUM(CASE WHEN a.status = 'Present' THEN 1 ELSE 0 END) * 100.0 / COUNT(a.attendance_id)) < 50
    AND
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	name

```
SELECT s.student_id, s.name
FROM Students s
LEFT JOIN Grades g ON s.student_id = g.student_id
LEFT JOIN Attendance a ON s.student_id = a.student_id
GROUP BY s.student_id, s.name
HAVING
MAX(g.marks_obtained) > 90 OR
(SUM(CASE WHEN a.status = 'Present' THEN 1 ELSE 0 END) = COUNT(a.attendance_id)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	name
2001	Ella Green
2002	Frank White
2003	Grace Hall

```sql
SELECT student_id, name
FROM Students
ORDER BY name ASC;
```

Result Grid | Filter Rows: | Edit:

student_id	name
2001	Ella Green
2002	Frank White
2003	Grace Hall
NULL	NULL

```sql
SELECT d.department_name, COUNT(s.student_id) AS total_students
FROM Departments d
JOIN Students s ON d.department_id = s.department_id
GROUP BY d.department_name
ORDER BY total_students DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

department_name	total_students
Computer Science	2
Electrical Engineering	1

```sql
SELECT AVG(attendance_percentage) AS overall_avg_attendance
FROM (
SELECT (SUM(CASE WHEN a.status = 'Present' THEN 1 ELSE 0 END) * 100.0 / CO
FROM Attendance a
GROUP BY a.student_id
) AS student_attendance;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

overall_avg_attendance
NULL

```
193    ) AS student_attendance;
194
195
196    SELECT c.course_name, MAX(g.marks_obtained) AS highest_mark, MIN(g.marks_obtained) AS lowest_mark
197    FROM Courses c
198    JOIN Grades g ON c.course_id = g.course_id
199    GROUP BY c.course_name;
200
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

course_name	highest_mark	lowest_mark
Database Systems	95.00	45.00
Algorithms	80.00	80.00

```
200
201
202    SELECT d.department_name, COUNT(s.student_id) AS total_students
203    FROM Departments d
204    JOIN Students s ON d.department_id = s.department_id
205    GROUP BY d.department_name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

department_name	total_students
Computer Science	2
Electrical Engineering	1

```
225
226    ## Q - 7
227
228    SELECT s.name, s.email, d.department_name
229    FROM Students s
230    INNER JOIN Departments d ON s.department_id = d.department_id;
231
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	email	department_name
Ella Green	ella.g@email.com	Computer Science
Frank White	frank.w@email.com	Computer Science
Grace Hall	grace.h@email.com	Electrical Engineering

```
235     WHERE e.enrollment_id IS NULL;
236
237 •   SELECT c.course_name
238     FROM Faculty f
239     RIGHT JOIN Courses c ON f.faculty_id = c.faculty_id
240     WHERE f.faculty_id IS NULL;
241
242
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

course_name
Advanced AI

```
242
243 •   SELECT s.student_id, s.name
244     FROM Students s
245     LEFT JOIN Grades g ON s.student_id = g.student_id
246     WHERE g.grade_id IS NULL
247     UNION
248     SELECT s.student_id, s.name
249     FROM Students s
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	name
2003	Grace Hall

```
253
254 •   SELECT s.student_id, s.name
255     FROM Students s
256     LEFT JOIN Grades g ON s.student_id = g.student_id
257     WHERE g.grade_id IS NULL;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	name
2003	Grace Hall

```sql
SELECT s.student_id, s.name, g.marks_obtained
FROM Students s
JOIN Grades g ON s.student_id = g.student_id
WHERE g.marks_obtained > (SELECT AVG(marks_obtained) FROM Grades);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	name	marks_obtained
2001	Ella Green	95.00
2001	Ella Green	80.00

```sql
SELECT MONTH(attendance_date) AS attendance_month, COUNT(attendance_id) AS total_records
FROM Attendance
GROUP BY attendance_month
ORDER BY attendance_month;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

attendance_month	total_records

```sql
SELECT name, admission_date,
TIMESTAMPDIFF(YEAR, admission_date, CURDATE()) AS years_since_admission
FROM Students;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	admission_date	years_since_admission
Ella Green	2023-09-01	2
Frank White	2023-09-01	2
Grace Hall	2023-09-01	2

```sql
## Q - 10

SELECT faculty_id, UPPER(name) AS uppercase_name
FROM Faculty;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

faculty_id	uppercase_name
101	DR. ALICE SMITH
102	PROF. BOB JOHNSON
103	DR. CAROL LEE
104	DR. DAVID KIM

```sql
SELECT student_id, TRIM(name) AS trimmed_name
FROM Students;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	trimmed_name
2001	Ella Green
2002	Frank White
2003	Grace Hall

```sql
SELECT faculty_id, COALESCE(email, 'Email Not Provided') AS faculty_email
FROM Faculty;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

faculty_id	faculty_email
104	Email Not Provided
101	alice.s@univ.edu
102	bob.j@univ.edu
103	carol.l@univ.edu

```sql
SELECT s.student_id, s.name,
SUM(g.marks_obtained) AS total_marks,
RANK() OVER (ORDER BY SUM(g.marks_obtained) DESC) AS overall_rank
FROM Students s
JOIN Grades g ON s.student_id = g.student_id
GROUP BY s.student_id, s.name
ORDER BY overall_rank;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	name	total_marks	overall_rank
2001	Ella Green	175.00	1
2002	Frank White	45.00	2

```sql
SELECT
DATE_FORMAT(enrollment_date, '%Y-%m') AS enrollment_month,
COUNT(enrollment_id) AS students_in_month,
SUM(COUNT(enrollment_id)) OVER (ORDER BY DATE_FORMAT(enrollment_date, '%Y-%m')) AS running_total_enrolled
FROM Enrollments
GROUP BY enrollment_month
ORDER BY enrollment_month;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

enrollment_month	students_in_month	running_total_enrolled
2023-09	4	4

```sql
SELECT s.student_id, s.name, g.marks_obtained,
    CASE
        WHEN g.marks_obtained > 90 THEN 'Excellent'
        WHEN g.marks_obtained BETWEEN 75 AND 90 THEN 'Good'
        ELSE 'Needs Improvement'
    END AS performance_level
FROM Students s
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	name	marks_obtained	performance_level
2001	Ella Green	95.00	Excellent
2001	Ella Green	80.00	Good
2002	Frank White	45.00	Needs Improvement

Result 35