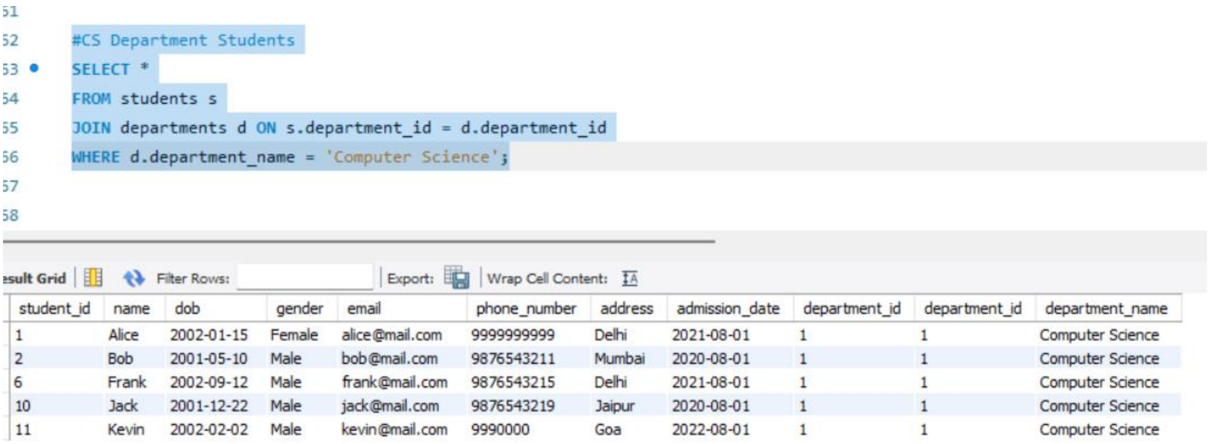
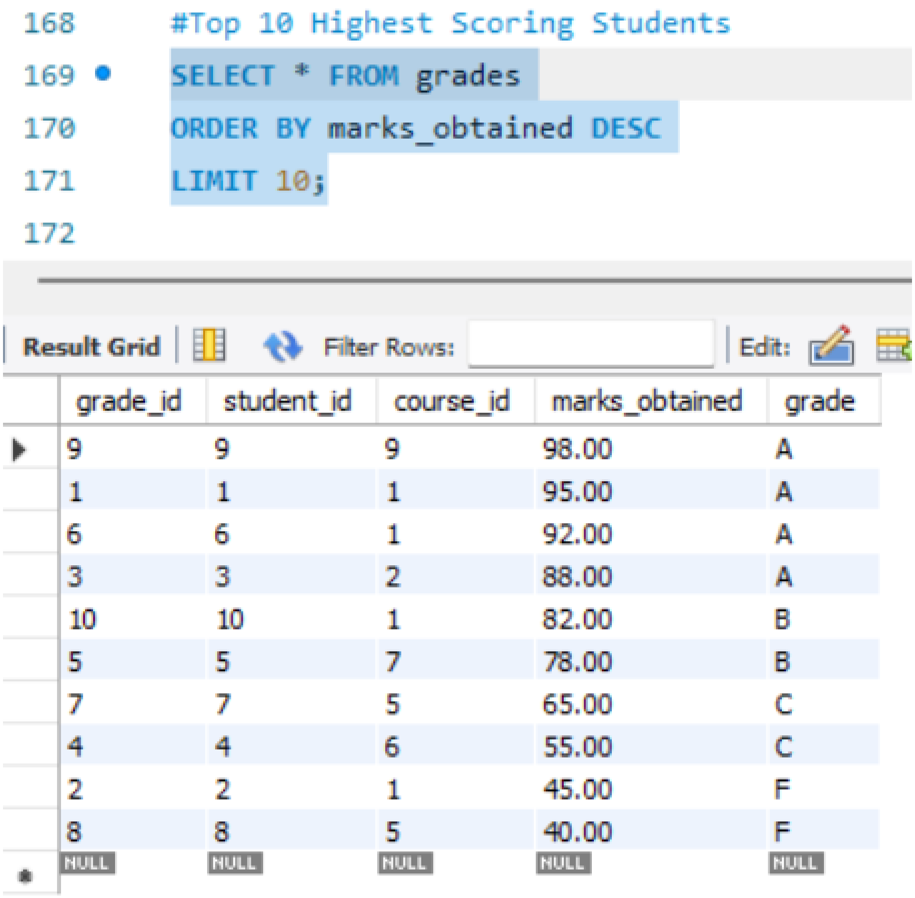


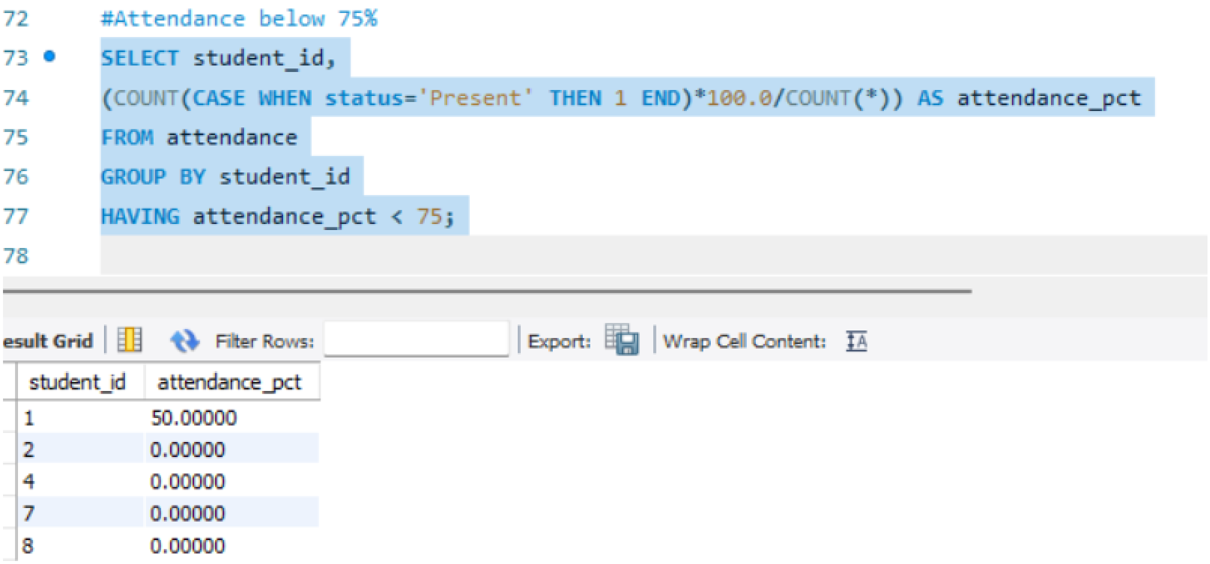
```sql
#CS Department Students
SELECT *
FROM students s
JOIN departments d ON s.department_id = d.department_id
WHERE d.department_name = 'Computer Science';
```

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id	department_id	department_name
1	Alice	2002-01-15	Female	alice@mail.com	9999999999	Delhi	2021-08-01	1	1	Computer Science
2	Bob	2001-05-10	Male	bob@mail.com	9876543211	Mumbai	2020-08-01	1	1	Computer Science
6	Frank	2002-09-12	Male	frank@mail.com	9876543215	Delhi	2021-08-01	1	1	Computer Science
10	Jack	2001-12-22	Male	jack@mail.com	9876543219	Jaipur	2020-08-01	1	1	Computer Science
11	Kevin	2002-02-02	Male	kevin@mail.com	9990000	Goa	2022-08-01	1	1	Computer Science



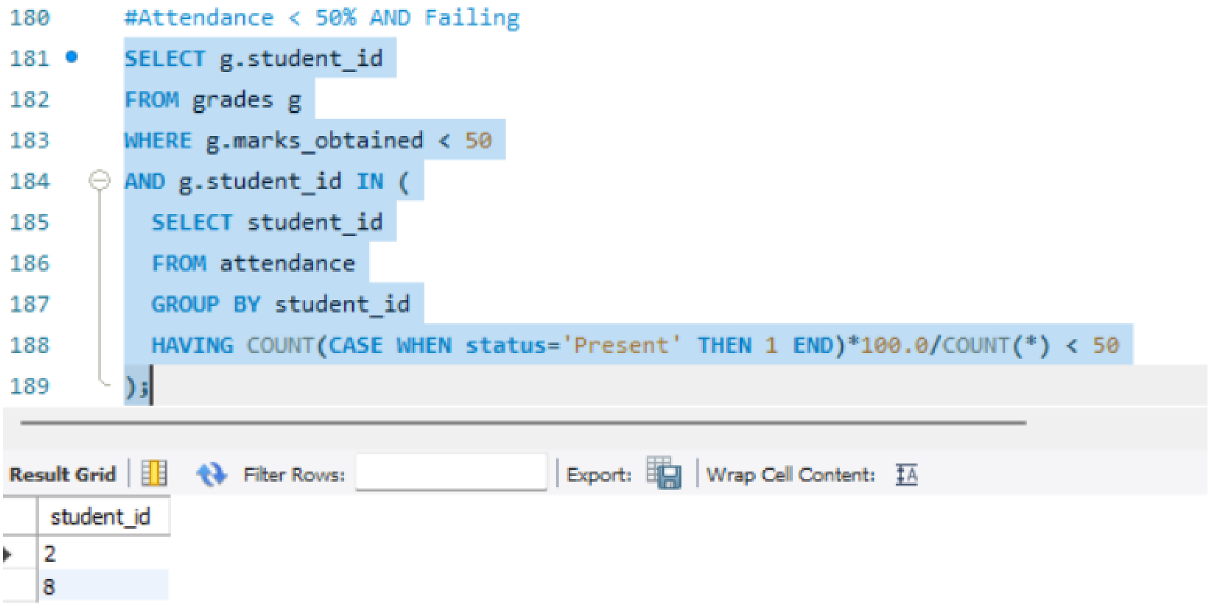
```sql
#Top 10 Highest Scoring Students
SELECT * FROM grades
ORDER BY marks_obtained DESC
LIMIT 10;
```

grade_id	student_id	course_id	marks_obtained	grade
9	9	9	98.00	A
1	1	1	95.00	A
6	6	1	92.00	A
3	3	2	88.00	A
10	10	1	82.00	B
5	5	7	78.00	B
7	7	5	65.00	C
4	4	6	55.00	C
2	2	1	45.00	F
8	8	5	40.00	F
NULL	NULL	NULL	NULL	NULL



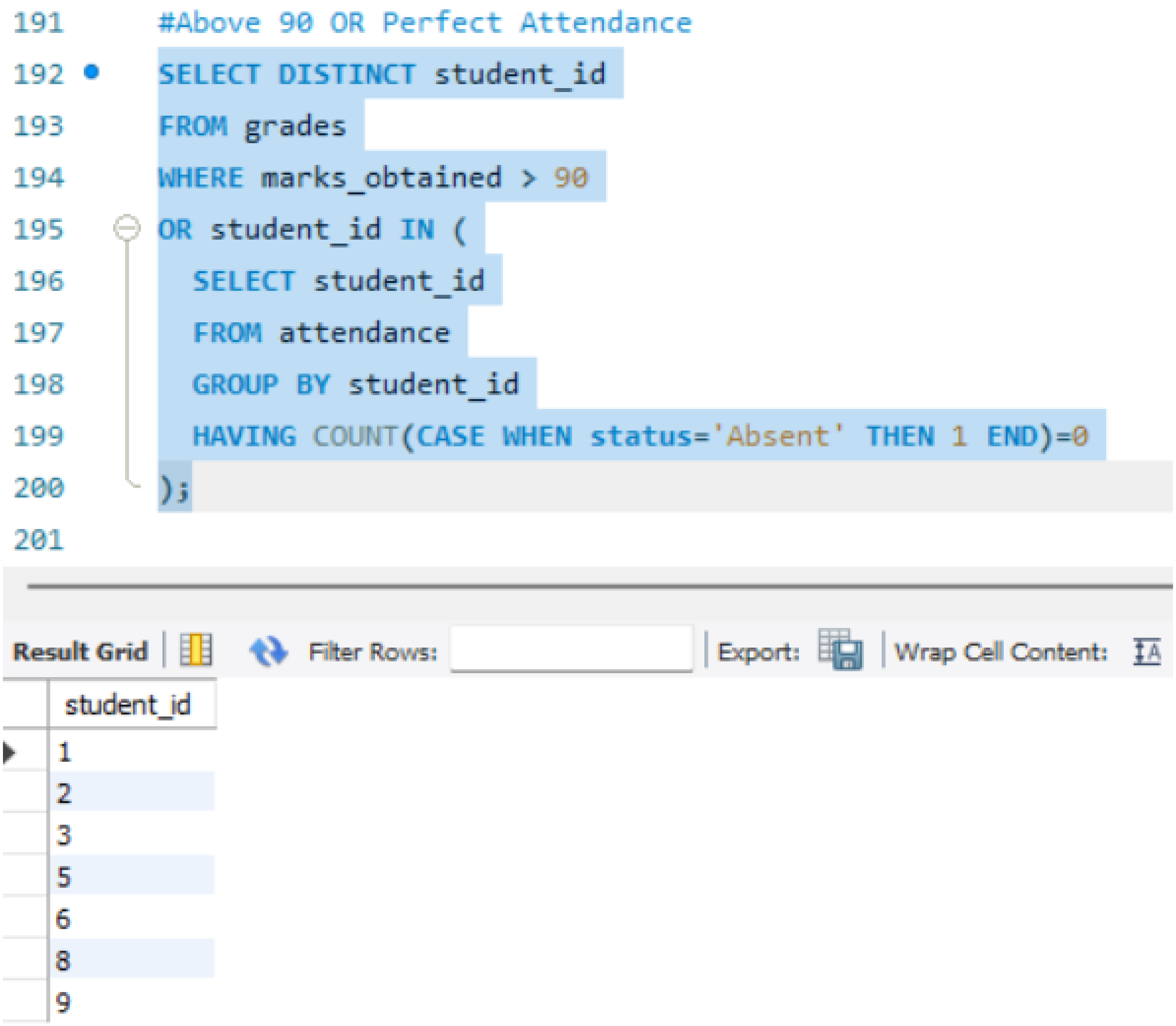
```sql
#Attendance below 75%
SELECT student_id,
(COUNT(CASE WHEN status='Present' THEN 1 END)*100.0/COUNT(*)) AS attendance_pct
FROM attendance
GROUP BY student_id
HAVING attendance_pct < 75;
```

student_id	attendance_pct
1	50.00000
2	0.00000
4	0.00000
7	0.00000
8	0.00000



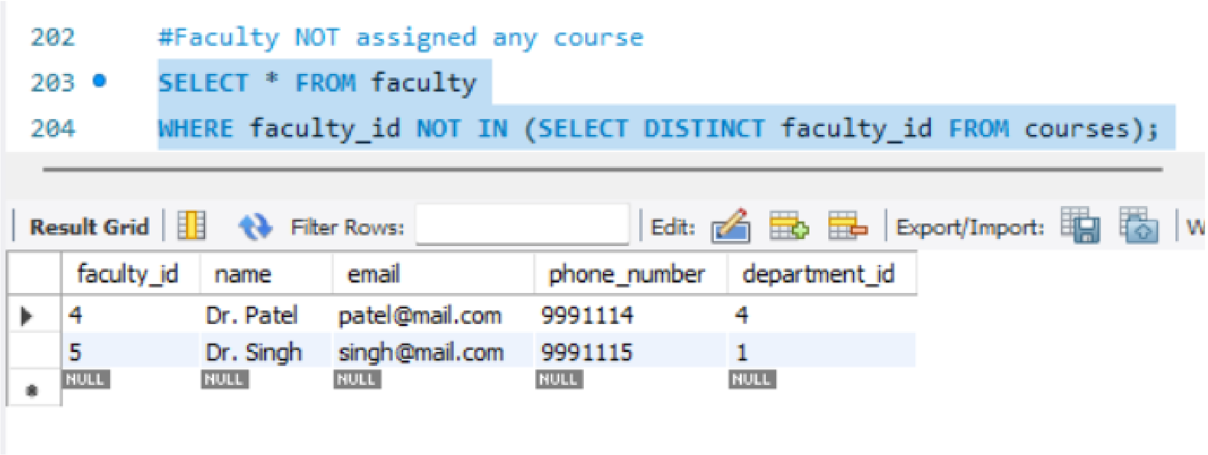
```sql
#Attendance < 50% AND Failing
SELECT g.student_id
FROM grades g
WHERE g.marks_obtained < 50
AND g.student_id IN (
  SELECT student_id
  FROM attendance
  GROUP BY student_id
  HAVING COUNT(CASE WHEN status='Present' THEN 1 END)*100.0/COUNT(*) < 50
);
```

student_id
2
8



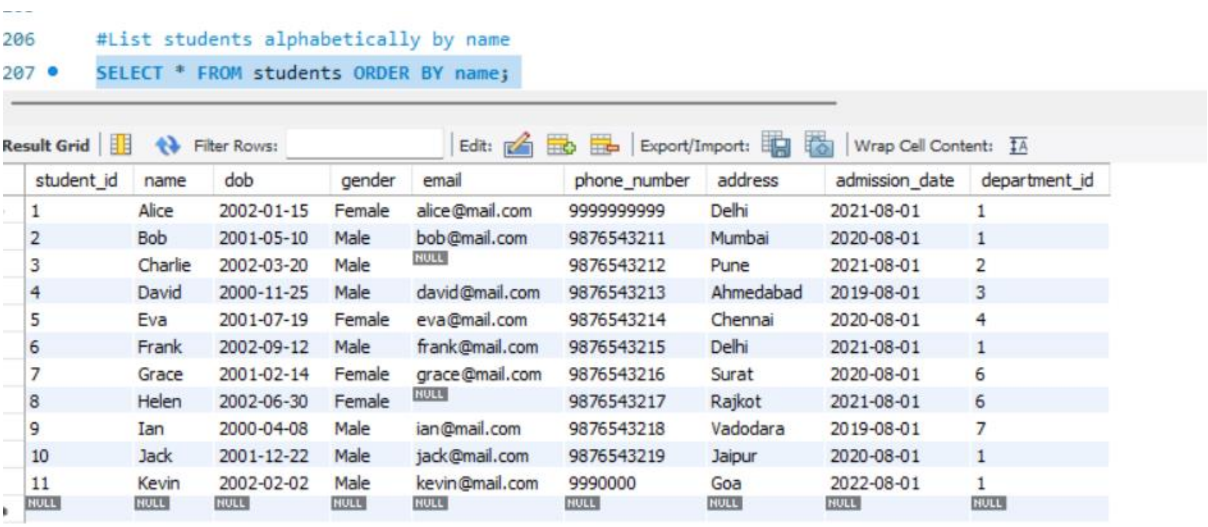
```
#Above 90 OR Perfect Attendance
SELECT DISTINCT student_id
FROM grades
WHERE marks_obtained > 90
OR student_id IN (
  SELECT student_id
  FROM attendance
  GROUP BY student_id
  HAVING COUNT(CASE WHEN status='Absent' THEN 1 END)=0
);
```

student_id
1
2
3
5
6
8
9



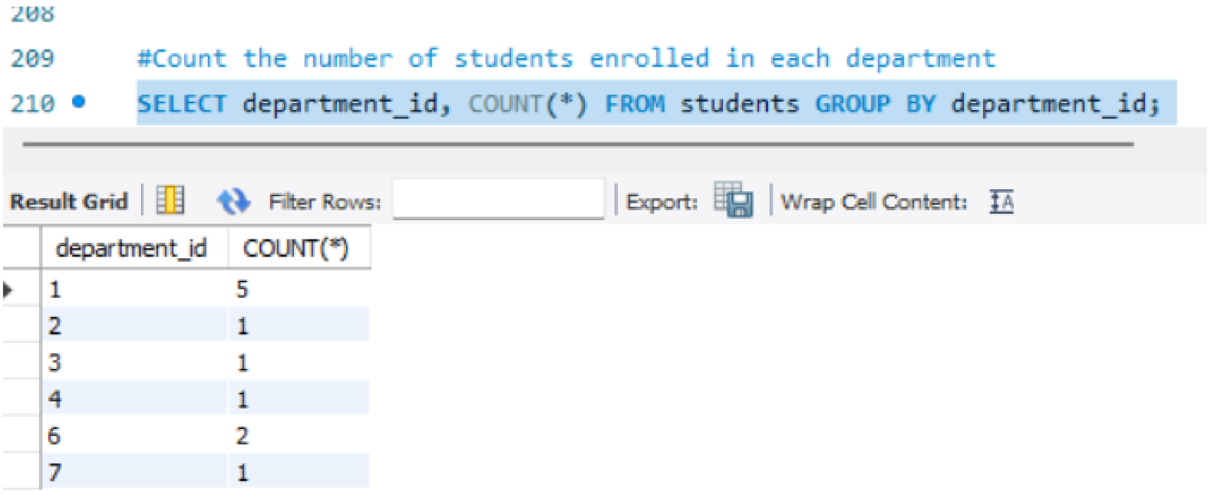
```
#Faculty NOT assigned any course
SELECT * FROM faculty
WHERE faculty_id NOT IN (SELECT DISTINCT faculty_id FROM courses);
```

faculty_id	name	email	phone_number	department_id
4	Dr. Patel	patel@mail.com	9991114	4
5	Dr. Singh	singh@mail.com	9991115	1
NULL	NULL	NULL	NULL	NULL



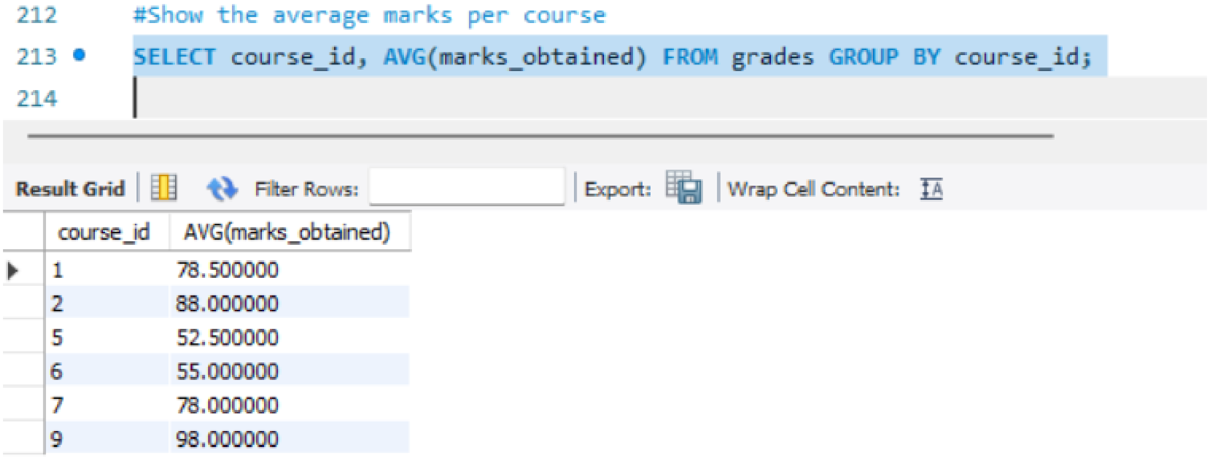
```
#List students alphabetically by name
SELECT * FROM students ORDER BY name;
```

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
1	Alice	2002-01-15	Female	alice@mail.com	9999999999	Delhi	2021-08-01	1
2	Bob	2001-05-10	Male	bob@mail.com	9876543211	Mumbai	2020-08-01	1
3	Charlie	2002-03-20	Male	NULL	9876543212	Pune	2021-08-01	2
4	David	2000-11-25	Male	david@mail.com	9876543213	Ahmedabad	2019-08-01	3
5	Eva	2001-07-19	Female	eva@mail.com	9876543214	Chennai	2020-08-01	4
6	Frank	2002-09-12	Male	frank@mail.com	9876543215	Delhi	2021-08-01	1
7	Grace	2001-02-14	Female	grace@mail.com	9876543216	Surat	2020-08-01	6
8	Helen	2002-06-30	Female	NULL	9876543217	Rajkot	2021-08-01	6
9	Ian	2000-04-08	Male	ian@mail.com	9876543218	Vadodara	2019-08-01	7
10	Jack	2001-12-22	Male	jack@mail.com	9876543219	Jaipur	2020-08-01	1
11	Kevin	2002-02-02	Male	kevin@mail.com	9990000	Goa	2022-08-01	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



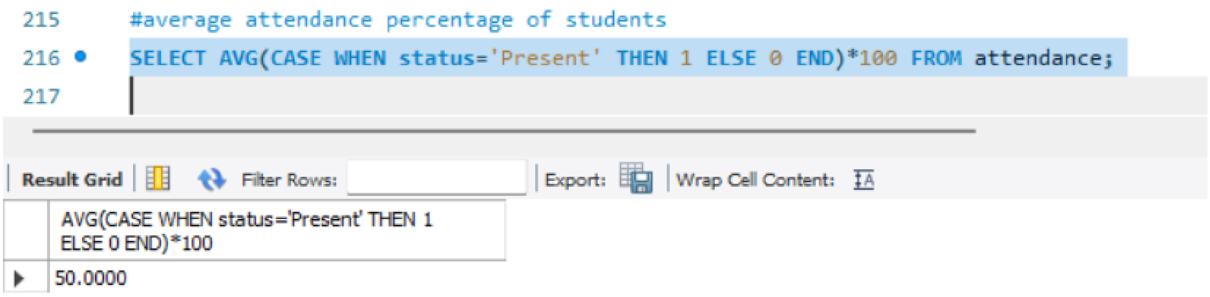
```
#Count the number of students enrolled in each department
SELECT department_id, COUNT(*) FROM students GROUP BY department_id;
```

department_id	COUNT(*)
1	5
2	1
3	1
4	1
6	2
7	1



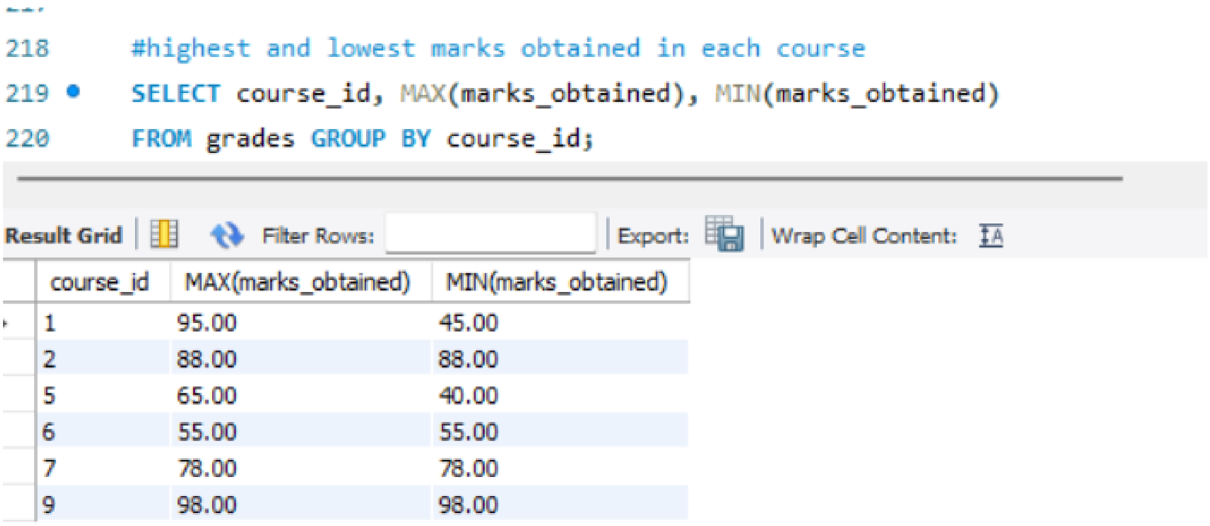
```
#Show the average marks per course
SELECT course_id, AVG(marks_obtained) FROM grades GROUP BY course_id;
```

course_id	AVG(marks_obtained)
1	78.500000
2	88.000000
5	52.500000
6	55.000000
7	78.000000
9	98.000000



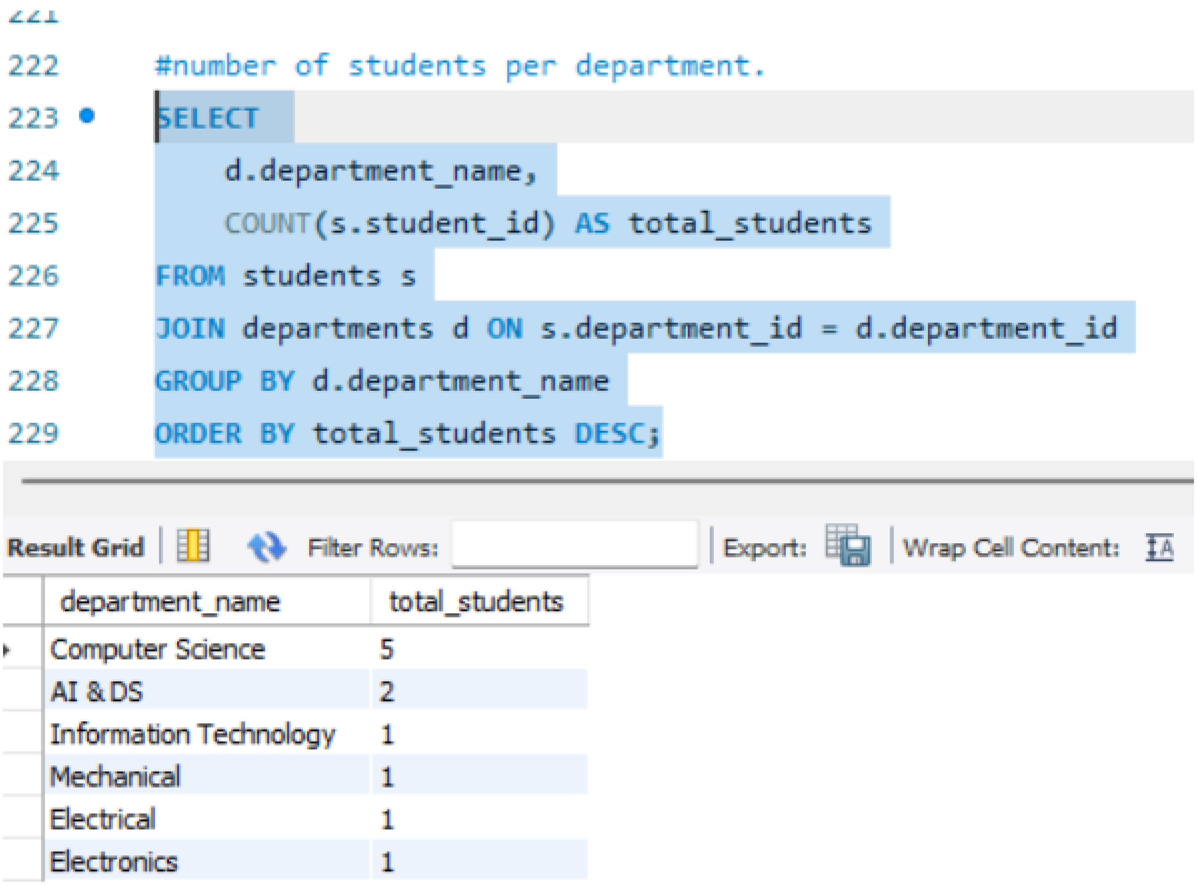
```sql
#average attendance percentage of students
SELECT AVG(CASE WHEN status='Present' THEN 1 ELSE 0 END)*100 FROM attendance;
```

AVG(CASE WHEN status='Present' THEN 1 ELSE 0 END)*100
50.0000



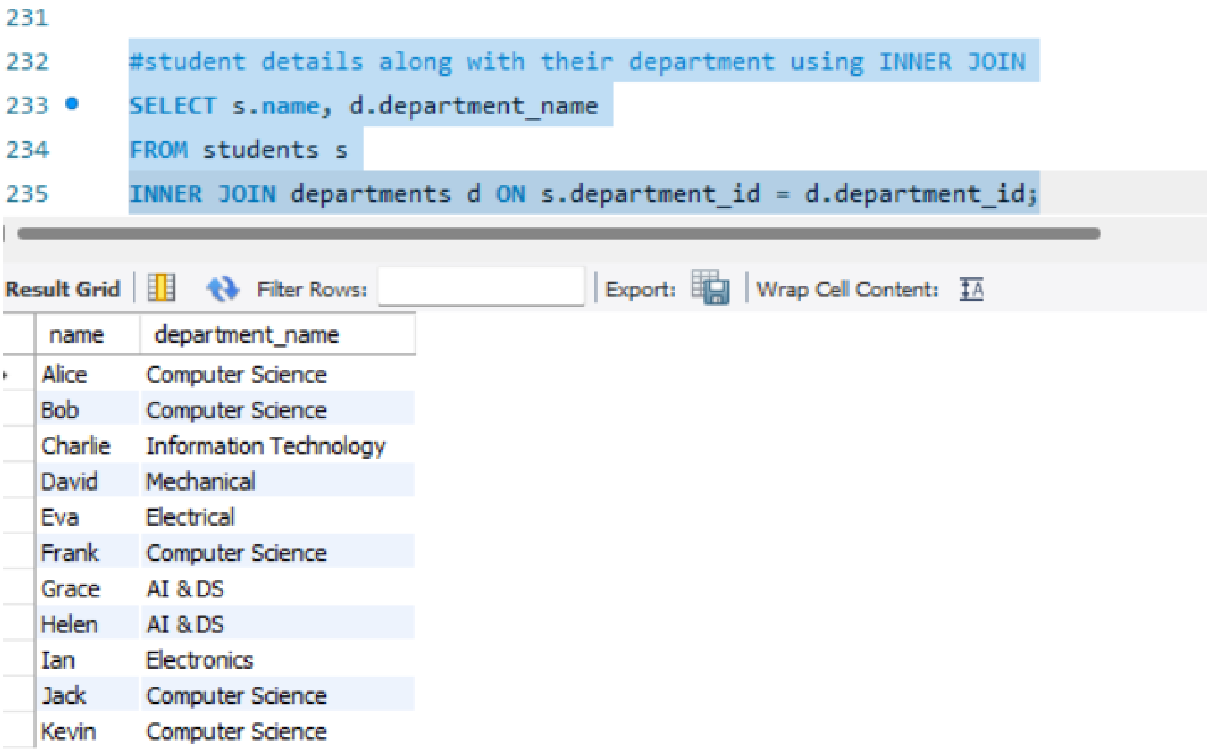
```sql
#highest and lowest marks obtained in each course
SELECT course_id, MAX(marks_obtained), MIN(marks_obtained)
FROM grades GROUP BY course_id;
```

course_id	MAX(marks_obtained)	MIN(marks_obtained)
1	95.00	45.00
2	88.00	88.00
5	65.00	40.00
6	55.00	55.00
7	78.00	78.00
9	98.00	98.00



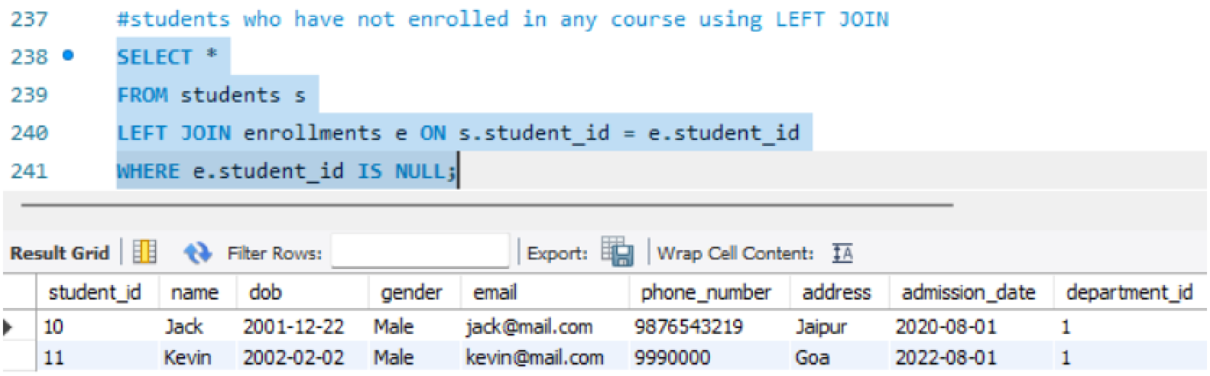
```sql
#number of students per department.
SELECT
    d.department_name,
    COUNT(s.student_id) AS total_students
FROM students s
JOIN departments d ON s.department_id = d.department_id
GROUP BY d.department_name
ORDER BY total_students DESC;
```

department_name	total_students
Computer Science	5
AI & DS	2
Information Technology	1
Mechanical	1
Electrical	1
Electronics	1



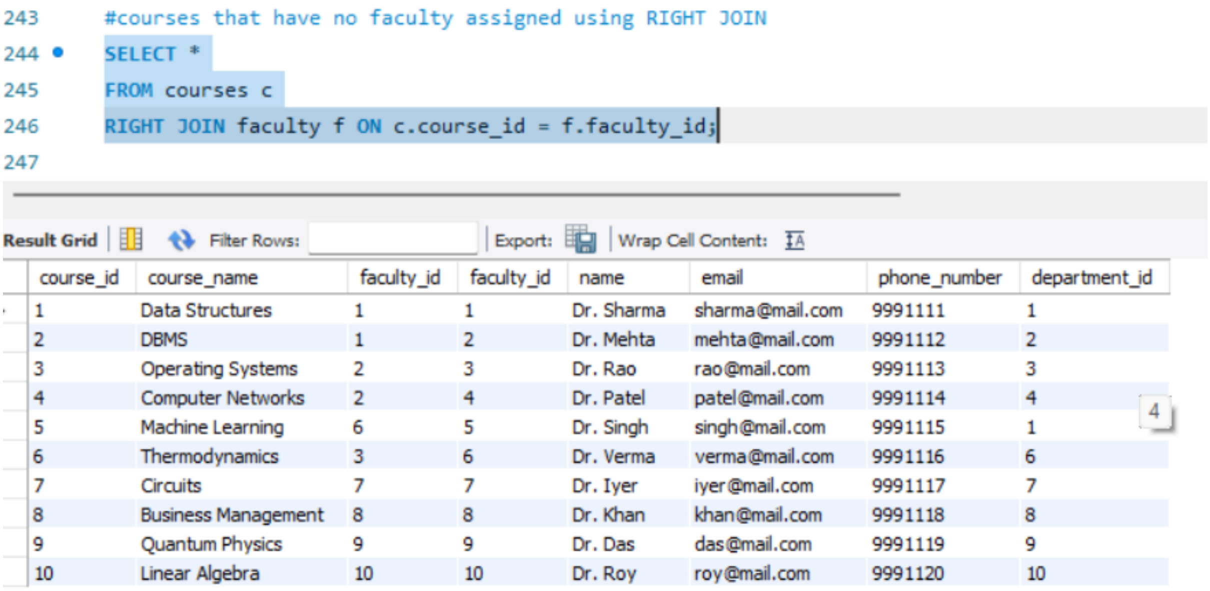
```sql
#student details along with their department using INNER JOIN
SELECT s.name, d.department_name
FROM students s
INNER JOIN departments d ON s.department_id = d.department_id;
```

name	department_name
Alice	Computer Science
Bob	Computer Science
Charlie	Information Technology
David	Mechanical
Eva	Electrical
Frank	Computer Science
Grace	AI & DS
Helen	AI & DS
Ian	Electronics
Jack	Computer Science
Kevin	Computer Science



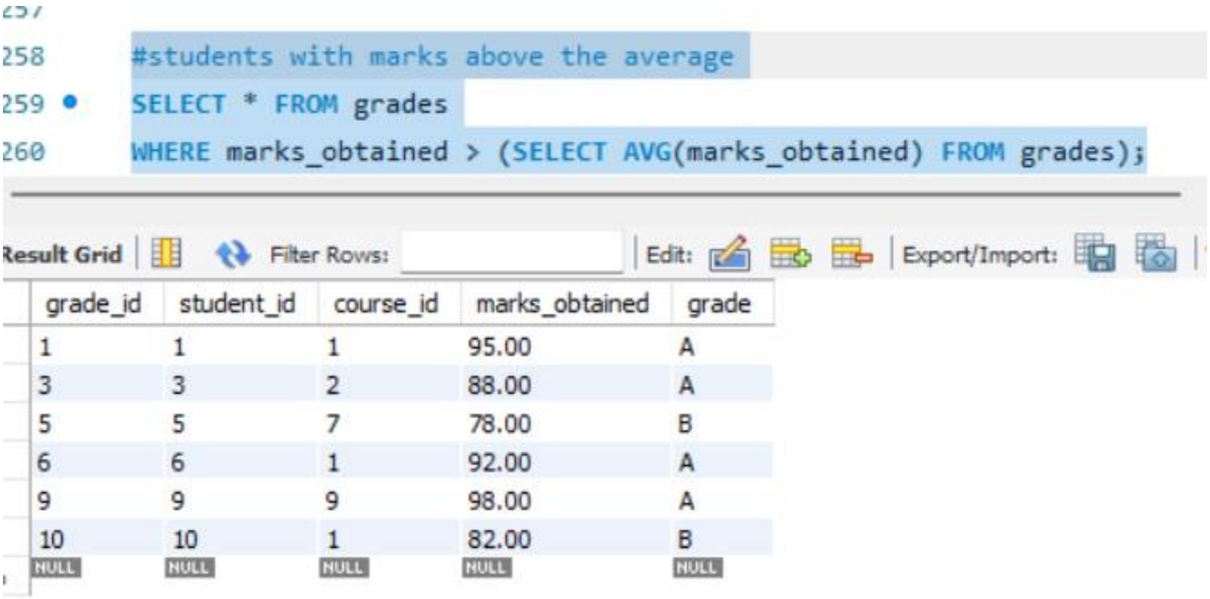
```sql
#students who have not enrolled in any course using LEFT JOIN
SELECT *
FROM students s
LEFT JOIN enrollments e ON s.student_id = e.student_id
WHERE e.student_id IS NULL;
```

student_id	name	dob	gender	email	phone_number	address	admission_date	department_id
10	Jack	2001-12-22	Male	jack@mail.com	9876543219	Jaipur	2020-08-01	1
11	Kevin	2002-02-02	Male	kevin@mail.com	9990000	Goa	2022-08-01	1



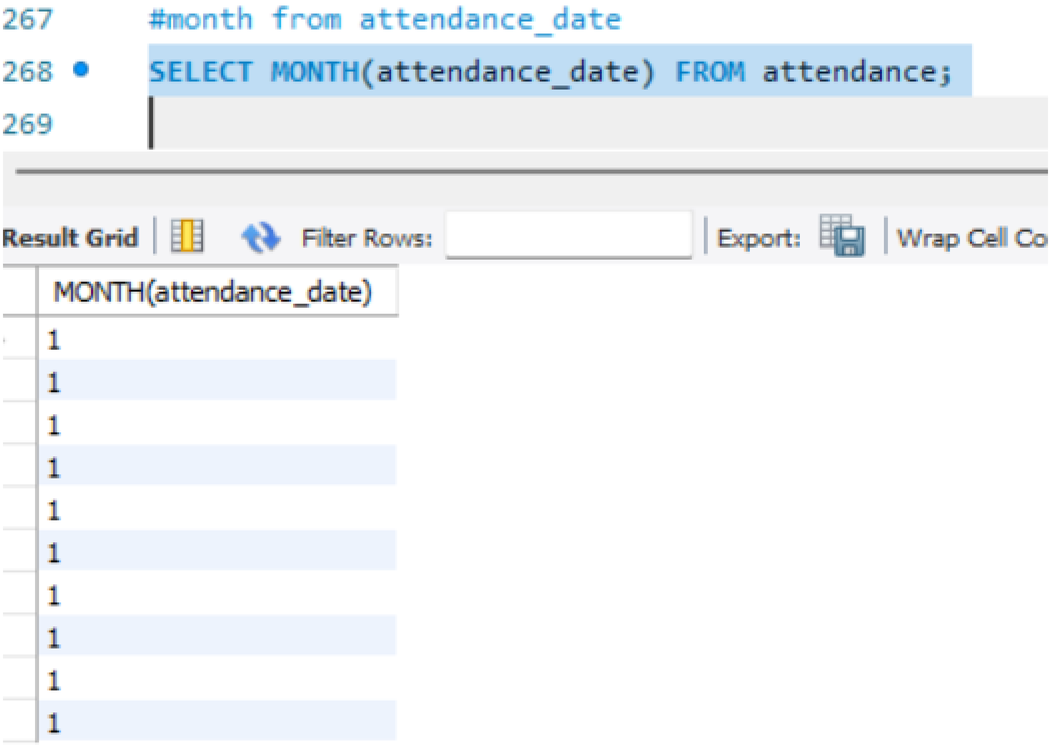
```sql
#courses that have no faculty assigned using RIGHT JOIN
SELECT *
FROM courses c
RIGHT JOIN faculty f ON c.course_id = f.faculty_id;
```

course_id	course_name	faculty_id	faculty_id	name	email	phone_number	department_id
1	Data Structures	1	1	Dr. Sharma	sharma@mail.com	9991111	1
2	DBMS	1	2	Dr. Mehta	mehta@mail.com	9991112	2
3	Operating Systems	2	3	Dr. Rao	rao@mail.com	9991113	3
4	Computer Networks	2	4	Dr. Patel	patel@mail.com	9991114	4
5	Machine Learning	6	5	Dr. Singh	singh@mail.com	9991115	1
6	Thermodynamics	3	6	Dr. Verma	verma@mail.com	9991116	6
7	Circuits	7	7	Dr. Iyer	iyer@mail.com	9991117	7
8	Business Management	8	8	Dr. Khan	khan@mail.com	9991118	8
9	Quantum Physics	9	9	Dr. Das	das@mail.com	9991119	9
10	Linear Algebra	10	10	Dr. Roy	roy@mail.com	9991120	10



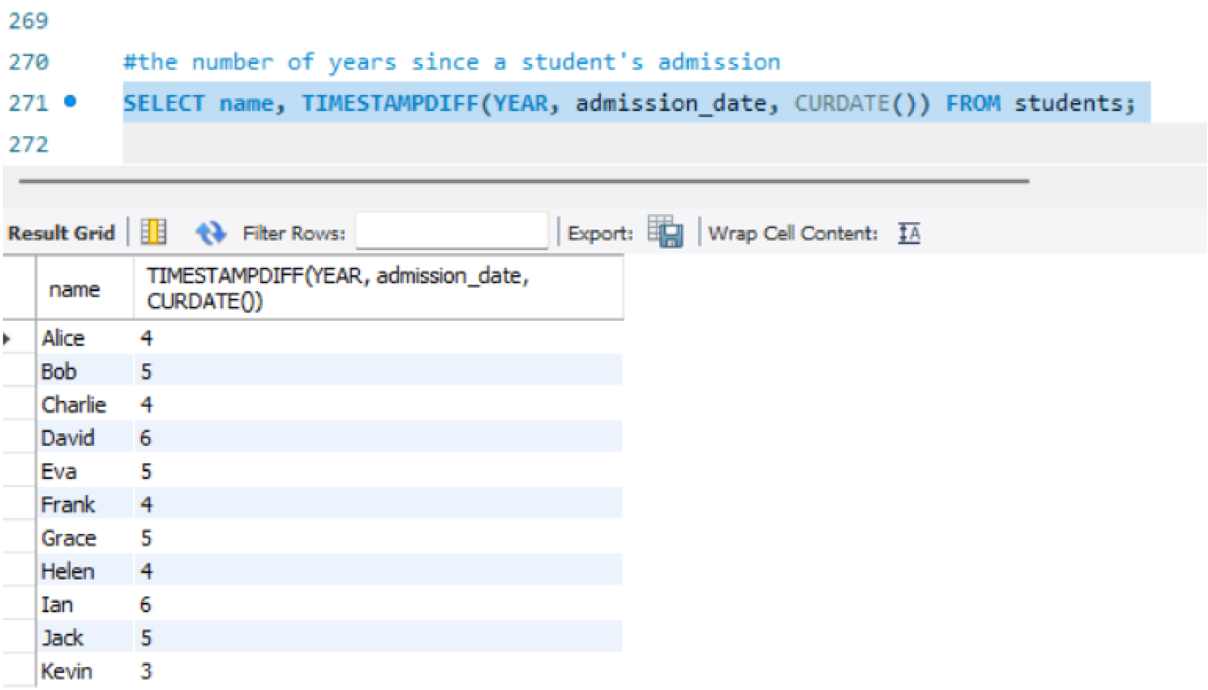
```
#students with marks above the average
SELECT * FROM grades
WHERE marks_obtained > (SELECT AVG(marks_obtained) FROM grades);
```

grade_id	student_id	course_id	marks_obtained	grade
1	1	1	95.00	A
3	3	2	88.00	A
5	5	7	78.00	B
6	6	1	92.00	A
9	9	9	98.00	A
10	10	1	82.00	B
NULL	NULL	NULL	NULL	NULL



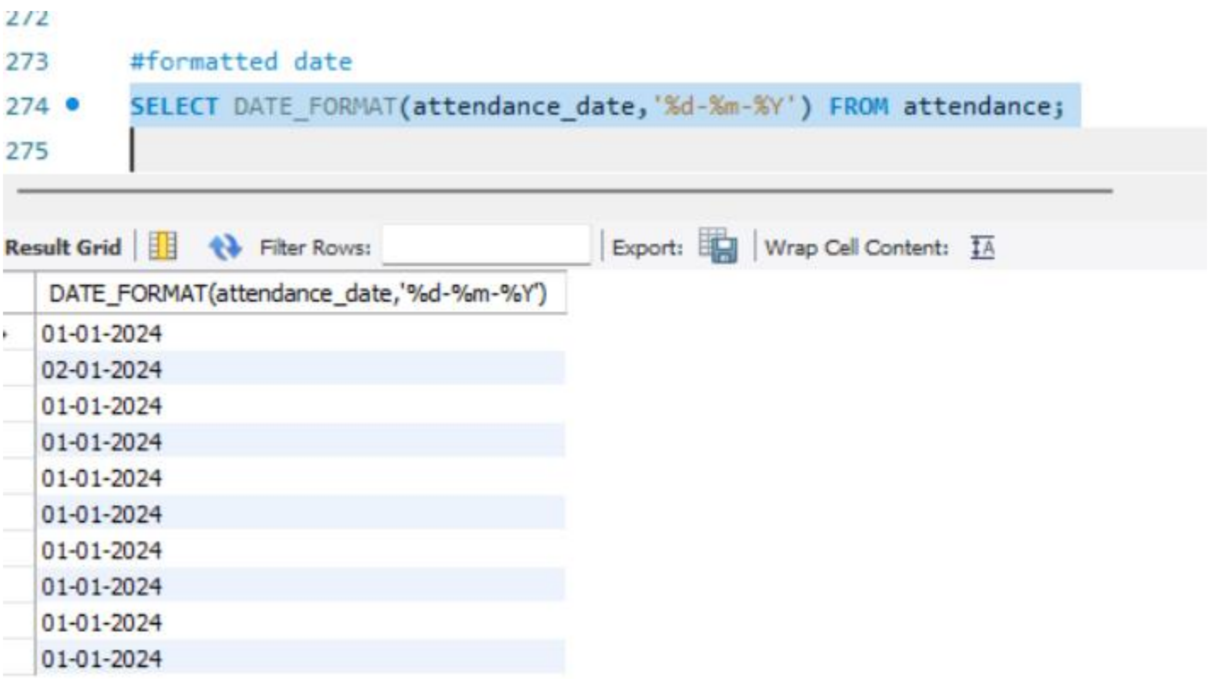
```
#month from attendance_date
SELECT MONTH(attendance_date) FROM attendance;
```

MONTH(attendance_date)
1
1
1
1
1
1
1
1
1
1



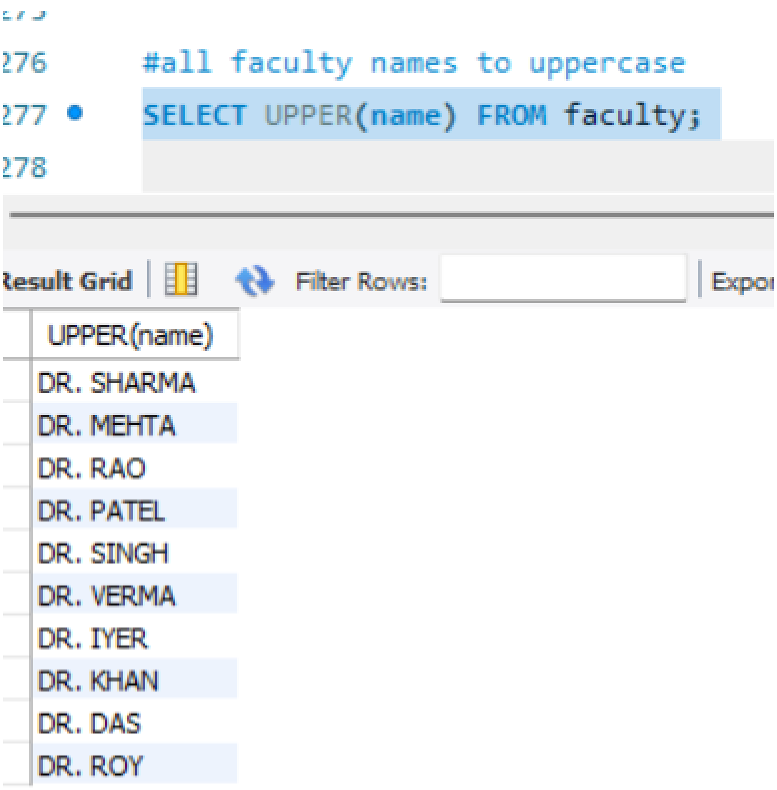
```sql

#the number of years since a student's admission
SELECT name, TIMESTAMPDIFF(YEAR, admission_date, CURDATE()) FROM students;

```

name	TIMESTAMPDIFF(YEAR, admission_date, CURDATE())
Alice	4
Bob	5
Charlie	4
David	6
Eva	5
Frank	4
Grace	5
Helen	4
Ian	6
Jack	5
Kevin	3



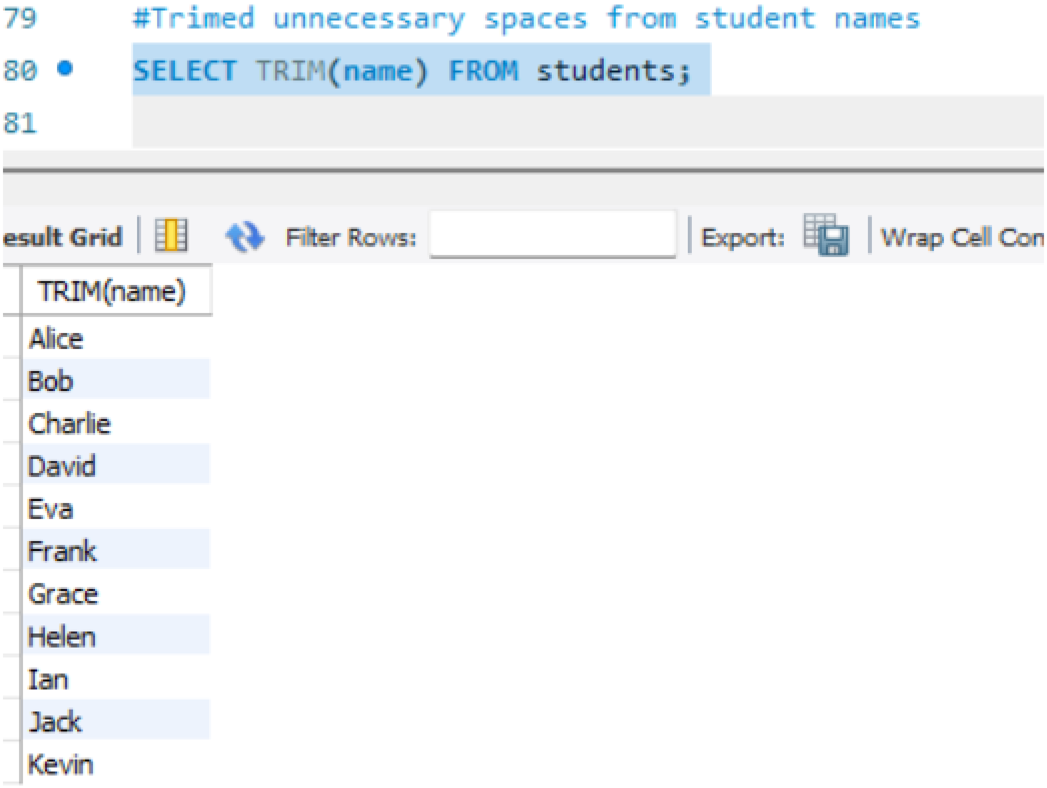
```sql

#formatted date
SELECT DATE_FORMAT(attendance_date,'%d-%m-%Y') FROM attendance;

```

DATE_FORMAT(attendance_date,'%d-%m-%Y')
01-01-2024
02-01-2024
01-01-2024
01-01-2024
01-01-2024
01-01-2024
01-01-2024
01-01-2024
01-01-2024
01-01-2024



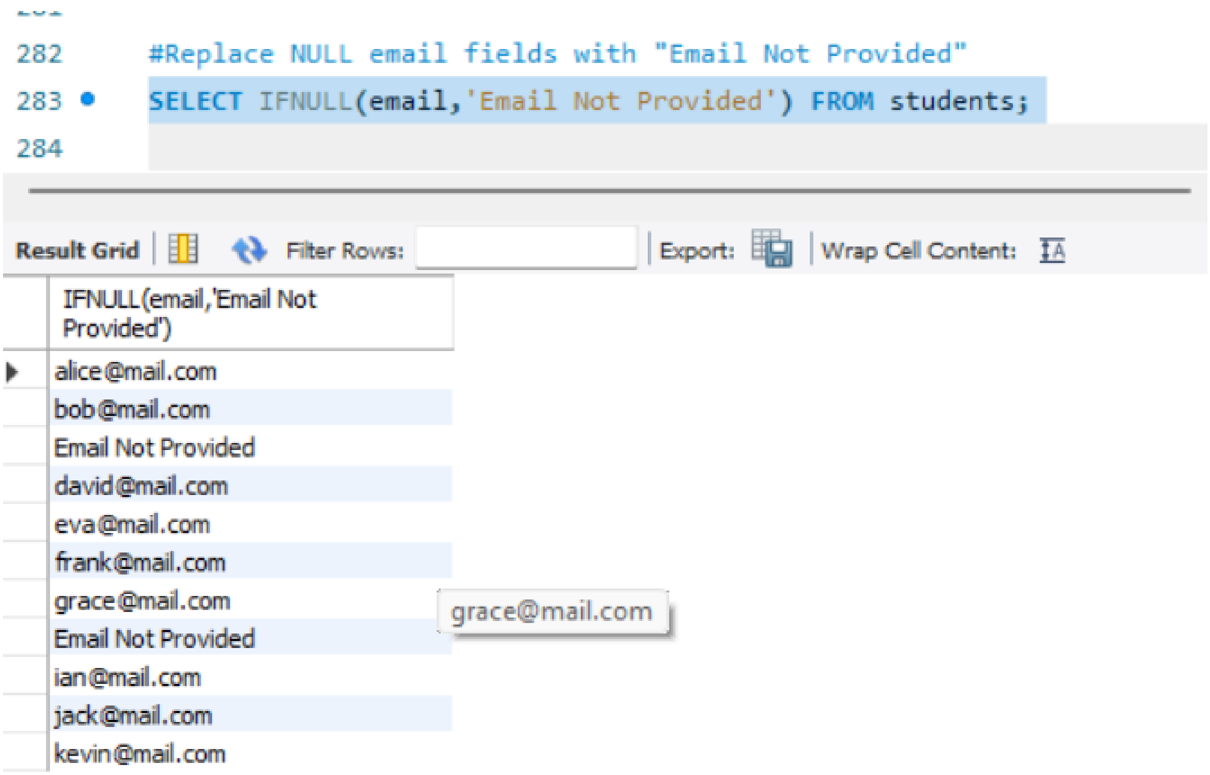
```
#all faculty names to uppercase
SELECT UPPER(name) FROM faculty;
```

UPPER(name)
DR. SHARMA
DR. MEHTA
DR. RAO
DR. PATEL
DR. SINGH
DR. VERMA
DR. IYER
DR. KHAN
DR. DAS
DR. ROY



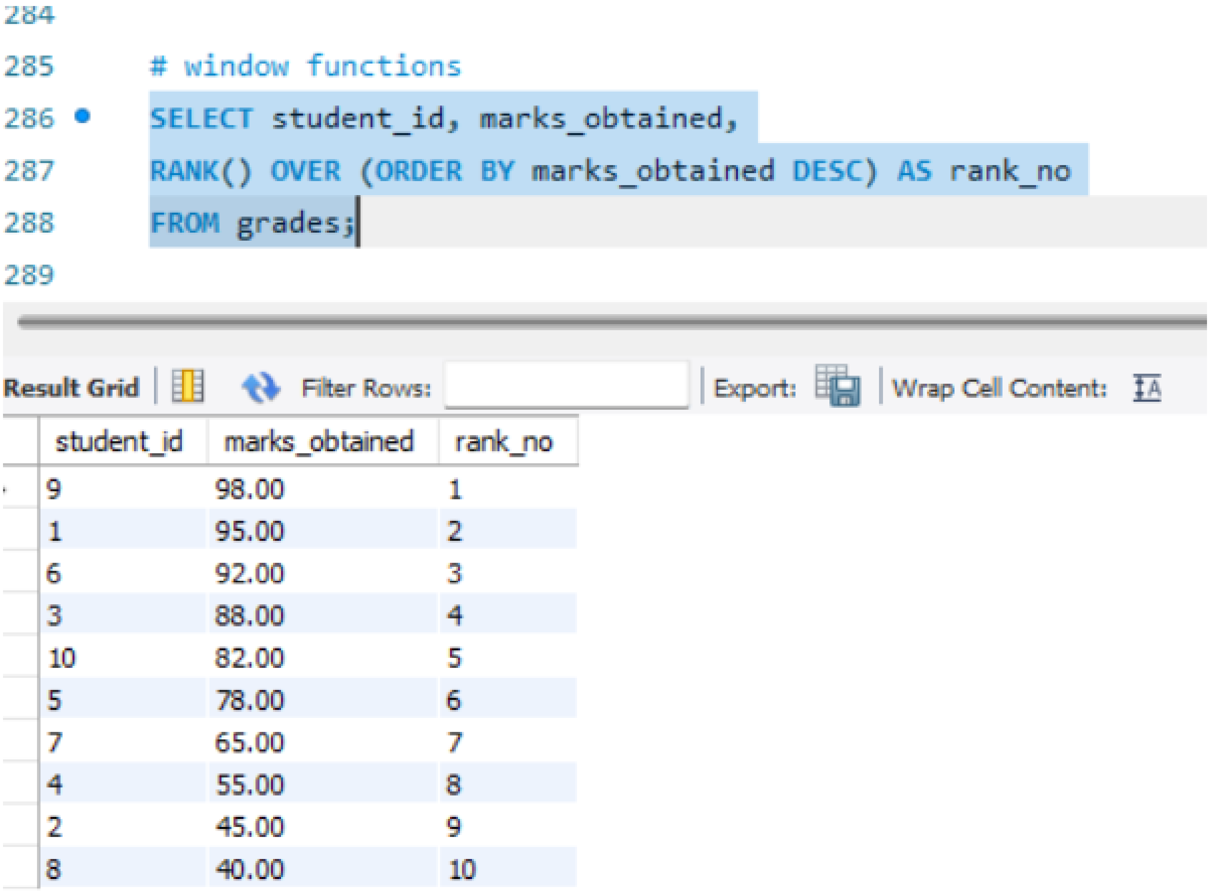
```
#Trimed unnecessary spaces from student names
SELECT TRIM(name) FROM students;
```

TRIM(name)
Alice
Bob
Charlie
David
Eva
Frank
Grace
Helen
Ian
Jack
Kevin



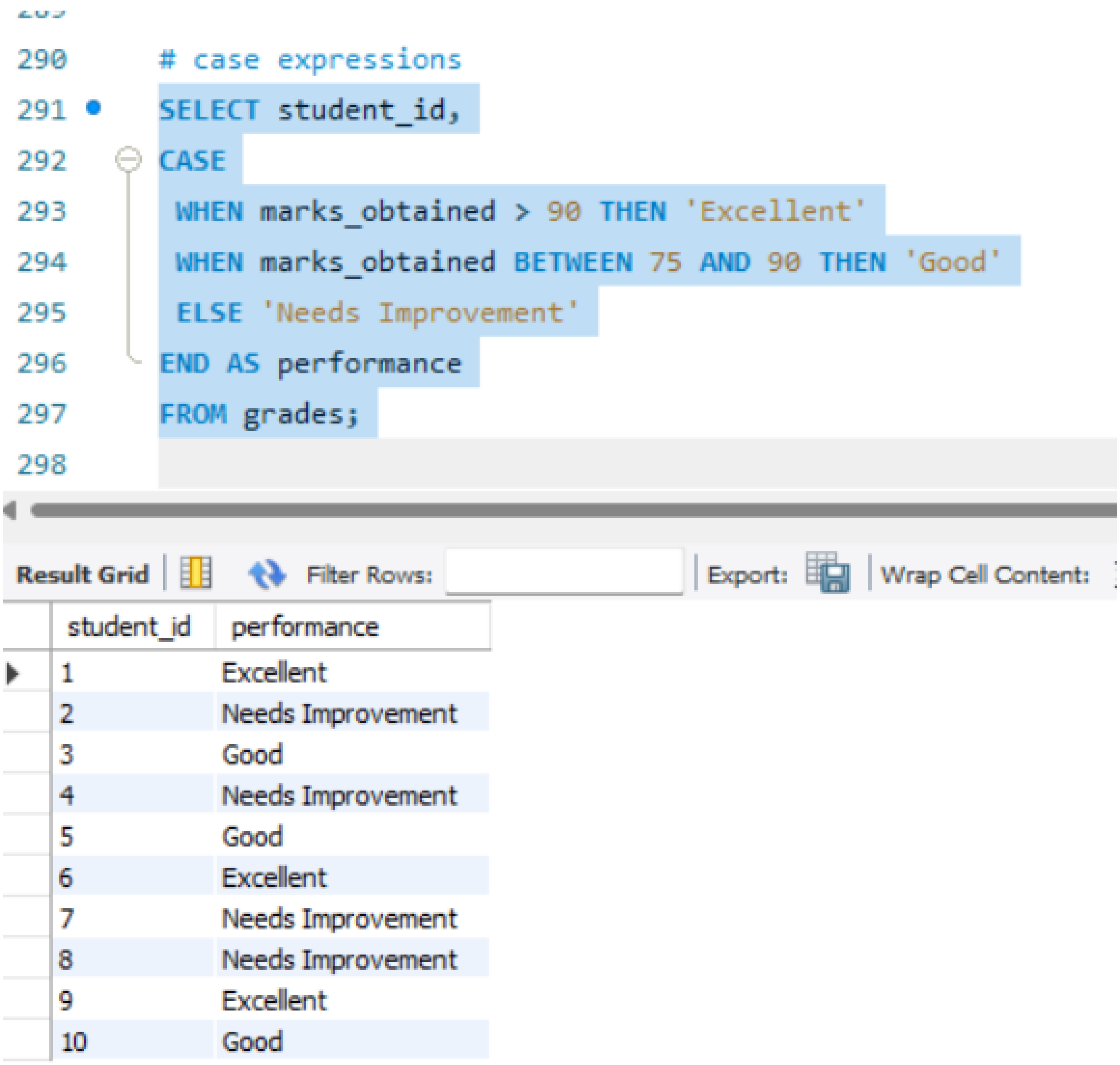
```
#Replace NULL email fields with "Email Not Provided"
SELECT IFNULL(email,'Email Not Provided') FROM students;
```

IFNULL(email,'Email Not Provided')
alice@mail.com
bob@mail.com
Email Not Provided
david@mail.com
eva@mail.com
frank@mail.com
grace@mail.com
Email Not Provided
ian@mail.com
jack@mail.com
kevin@mail.com



```
# window functions
SELECT student_id, marks_obtained,
RANK() OVER (ORDER BY marks_obtained DESC) AS rank_no
FROM grades;
```

student_id	marks_obtained	rank_no
9	98.00	1
1	95.00	2
6	92.00	3
3	88.00	4
10	82.00	5
5	78.00	6
7	65.00	7
4	55.00	8
2	45.00	9
8	40.00	10



```
# case expressions
SELECT student_id,
CASE
 WHEN marks_obtained > 90 THEN 'Excellent'
 WHEN marks_obtained BETWEEN 75 AND 90 THEN 'Good'
 ELSE 'Needs Improvement'
END AS performance
FROM grades;
```

student_id	performance
1	Excellent
2	Needs Improvement
3	Good
4	Needs Improvement
5	Good
6	Excellent
7	Needs Improvement
8	Needs Improvement
9	Excellent
10	Good