

Task-1: Find students who are older than 20 and have a GPA above the average GPA of all students

Code:

```
SELECT * FROM `students_info` WHERE age > 20 AND GPA > (SELECT AVG(GPA) from students_info)
```

Output:

student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
1	Eleven	21	3.80	Engineering	2021	10000.00	120	active
2	Dustin	22	3.90	Science	2020	9000.00	110	active
4	Mike	23	3.70	Science	2021	9500.00	115	inactive
6	Eddie	22	4.00	Arts	2019	8000.00	140	active
9	Steve	21	3.80	Science	2021	10500.00	120	active
12	Nancy	23	3.90	Business	2019	9500.00	135	active

Task-2: Find the top 5 students with the highest fees paid, ordered by GPA (in descending order) as a tiebreaker

Code:

```
SELECT student_name, fees_paid, GPA FROM `students_info` ORDER BY fees_paid AND GPA DESC LIMIT 3
```

Output:

student_name	fees_paid	GPA
Nancy	9500.00	3.90
Dustin	9000.00	3.90
Will	8500.00	3.40

Task-3: List students who belong to the "Engineering" department, have a GPA greater than 3.5, and are enrolled after 2020

Code:

```
SELECT * FROM `students_info` WHERE department="Engineering" AND GPA>3.5 AND year_of_admission > 2020
```

Output:

student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
1	Eleven	21	3.80	Engineering	2021	10000.00	120	active
10	Robin	20	3.60	Engineering	2022	11000.00	125	active

Task-4: Find students who are not active (i.e., enrollment_status = 'inactive') and have not paid any fees (fees_paid = 0)

Code:

```
SELECT * FROM `students_info` WHERE fees_paid=0.00 AND enrollment_status="inactive";
```

Output:

student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
------------	--------------	-----	-----	------------	-------------------	-----------	----------------	-------------------

Query results operations

Task-5: Calculate the total fees paid and average GPA for each department, but only for departments with more than 10 students

Code:

```
SELECT SUM(fees_paid) as total_sum , AVG(GPA) FROM `students_info` WHERE (SELECT COUNT(department) from `students_info`)>10
```

Output:

Extra options

total_sum	AVG(GPA)
104500.00	3.533333

☐ Show all | Number of rows: 25 ▼ Filter rows:

Query results operations