

In today's fast-paced and competitive educational environment, understanding the factors that influence student success is more important than ever. Just like the transport system in a bustling city like London must adapt to serve its residents, schools and educators must adapt to meet the needs of students. In this project, we will take a deep dive into a dataset containing rich details about various aspects of student life, such as hours studied, sleep patterns, attendance, and more, to uncover what truly impacts exam performance.

The dataset we'll be working with includes a wide range of factors influencing student performance. By analyzing this data, we'll be able to identify key drivers of success and provide insights that could help students, teachers, and policymakers make informed decisions. The table we'll use for this project is called `student_performance` and includes the following data:

Column	Definition	Data type
<code>attendance</code>	Percentage of classes attended	<code>float</code>
<code>extracurricular_activities</code>	Participation in extracurricular activities	<code>varchar</code> (Yes, No)
<code>sleep_hours</code>	Average number of hours of sleep per night	<code>float</code>
<code>tutoring_sessions</code>	Number of tutoring sessions attended per month	<code>integer</code>
<code>teacher_quality</code>	Quality of the teachers	<code>varchar</code> (Low, Medium, High)
<code>exam_score</code>	Final exam score	<code>float</code>

You will execute SQL queries to answer three questions, as listed in the instructions.

Projects Data    DataFrame as    `avg_exam_score_by_study_and_exti`

```
-- avg_exam_score_by_study_and_extracurricular
-- Edit the query below as needed
SELECT *
FROM student_performance
LIMIT 30;

-- avg_exam_score_by_study_and_extracurricular

-- This query calculates the average exam score of students who studied more than 10 hours
and participated in extracurricular activities.
SELECT Hours_Studied, AVG(Exam_Score) AS avg_exam_score
FROM student_performance
-- Filters the data to include only those who studied more than 10 hours and participated in
extracurricular activities.
WHERE Hours_Studied > 10
      AND Extracurricular_Activities = 'Yes'
-- Groups the results by the number of hours studied to calculate the average exam score for
each study group.
```

```
GROUP BY Hours_Studied
-- Orders the results by the number of hours studied in descending order.
ORDER BY Hours_Studied DESC;
```

hours_studied	avg_exam_score
0	43
1	39
2	38
3	37
4	36
5	35
6	34
7	33
8	32
9	31
10	30
11	29
12	28
13	27
14	26
15	25

30 rows



Projects Data DataFrame as avg\_exam\_score\_by\_hours\_studied.

```
-- avg_exam_score_by_hours_studied_range
-- Add solution code below
-- avg_exam_score_by_hours_studied_range

-- This query groups students by ranges of hours studied and calculates the average exam
score for each group.
SELECT
  CASE
    -- Categorizes hours studied into specific ranges: 1-5, 6-10, 11-15, and 16+ hours.
    WHEN Hours_Studied BETWEEN 1 AND 5 THEN '1-5 hours'
    WHEN Hours_Studied BETWEEN 6 AND 10 THEN '6-10 hours'
    WHEN Hours_Studied BETWEEN 11 AND 15 THEN '11-15 hours'
    ELSE '16+ hours'
  END AS hours_studied_range,
  -- Calculates the average exam score for each range of hours studied.
  AVG(Exam_Score) AS avg_exam_score
```

```
-- Specifies the source of the data.
FROM student_performance
-- Groups the data by the newly created study ranges.
GROUP BY hours_studied_range
-- Orders the result by average exam score in descending order to see which range performs best.
ORDER BY avg_exam_score DESC;
```

▼	hours_studied_range	▼	avg_exam_score
0	16+ hours		
1	11-15 hours		
2	6-10 hours		
3	1-5 hours		

4 rows [↓](#)

 Projects Data    DataFrame as   s

```
-- student_exam_ranking
-- Add solution code below
-- student_exam_ranking
-- This query ranks students based on their exam scores, including other factors like attendance, study hours, sleep hours, and tutoring sessions.
SELECT
    Attendance,
    Hours_Studied,
    Sleep_Hours,
    Tutoring_Sessions,
    -- Assigns a rank to each student based on their exam score, with the highest score ranked first.
    DENSE_RANK() OVER (ORDER BY Exam_Score DESC) AS exam_rank
-- Specifies the source of the data.
FROM student_performance
-- Orders the result by exam rank in ascending order so the highest-ranked students come first.
ORDER BY exam_rank ASC
-- Limits the result to the top 30 students.
LIMIT 30;
```

✓	attendance	✓	hours_studied	✓	sleep_hours	✓	tutoring_
0		98		27		6	
1		89		18		4	
2		90		14		8	
3		83		23		4	
4		96		28		4	
5		90		28		9	
6		83		16		8	
7		83		15		7	
8		74		21		6	