Data Analyst Test

Introduction

This test consists of two tasks. The first task is SQL-based questions and the second is chart analysis. You will be provided with six tables which reside in a MySQL DB. You are free to choose any SQL client, such as DBeaver, DataGrip, etc, to connect to the DB.

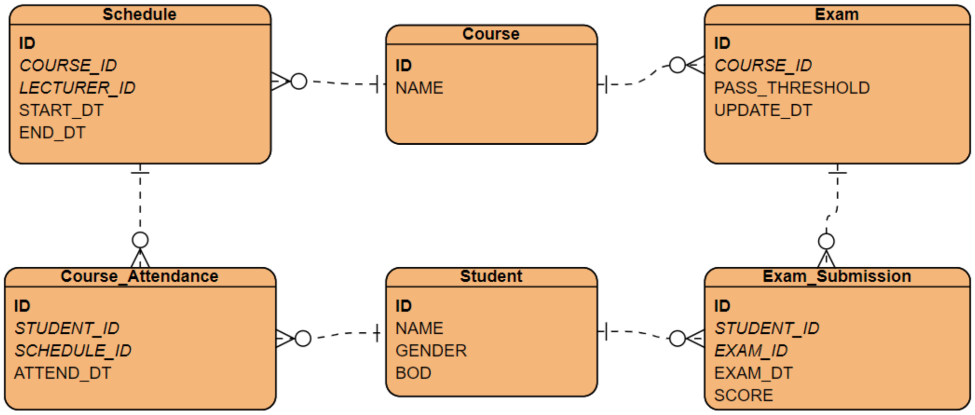
Database Connection (read-only access)

Host	34.101.51.177
DB Name	univ
DB User	candidate
DB Password	xDt905G%y23l!
Port	3306

Note : This is MySQL version 8.0

Table Summary

The tables give information about students, courses, and exams during the academic year of 2019/2020 in a university. The ERD is shown as follows:



The data is pretty straightforward. For students who take the exam and fail to get at least a minimum number of points to pass, they may take the exam once more, i.e. resit. If it is unsuccessful again, the student will completely fail the exam.

There are 13 weeks for each semester. For simplicity, we assume that there is no holiday throughout the semesters. Reference of days of week used in this data.
1: Sunday, 2:Monday, 3:Tuesday, 4:Wednesday, 5:Thursday, 6:Friday, 7:Saturday

Task 1

Answer the questions with SQL and copy the SQL in the green box (not screenshot), then present the answer within the table.

Note: If the time window is not mentioned in the question, it implies the whole period

1. **Who did not enroll in any courses in 2019 ?**

```
SELECT name
FROM student
LEFT JOIN enrollment
ON student.id=enrollment.student_id
WHERE  enroll_dt is null;
```

Student Name
Frank
James
Don
Benny
Pomela
Icarus
Angela

2. **Who had attendance below 75% and which course was it?**

```
WITH main AS
(SELECT
        ca.student_id,
        ca.schedule_id,
        s.course_id,
        co.name course_name,
        st.name,
        count(ca.id) as total_attendance,
        length(replace(s.course_days,',',''))*(week(s.end_dt)-week(s.start_dt))
total_course_day
FROM course_attendance ca
```

```
        LEFT JOIN schedule s on ca.schedule_id = s.id
        LEFT JOIN student st on ca.student_id = st.id
        LEFT JOIN course co on s.course_id = co.id
GROUP BY 1,2,3,4,5,7),

main2 AS
(SELECT name, course_name, (total_attendance/total_course_day*100) attd_pctg
FROM main)

SELECT name,course_name
from main2
where attd_pctg<75;
```

Student Name	Course Name
Sisca	Introduction to Applied Machine Learning
Sisca	Applied Data Science
Kiky	Statistic
Michael	Python Programming

3. **Who had 100% attendance but ever failed the exam ? Breakdown per course.**

```
WITH main AS
(SELECT
        ca.student_id,
        ca.schedule_id,
        s.course_id,
        co.name course_name,
        st.name,
        count(ca.id) as total_attendance,
        length(replace(s.course_days,',',''))*(week(s.end_dt)-week(s.start_dt))
total_course_day
FROM course_attendance ca
        LEFT JOIN schedule s on ca.schedule_id = s.id
        LEFT JOIN student st on ca.student_id = st.id
         LEFT JOIN course co on s.course_id = co.id
GROUP BY 1,2,3,4,5,7),
main2 AS
(SELECT
 student_id,
name,
course_name,
(total_attendance/total_course_day*100) attd_pctg
FROM main
),

main3 AS
(SELECT *
FROM main2
WHERE attd_pctg=100
)

SELECT name, course_name
FROM main3
WHERE student_id in
(SELECT distinct student_id
from exam_submission es
LEFT JOIN exam e on es.exam_id = e.id
WHERE es.score<e.pass_threshold );
```

Student Name	Course Name
Michael	Introduction to Applied Machine Learning
Sisca	Data Mining
Horatio	Introduction to Applied Machine Learning
Horatio	Data Mining

4. **Who had attendance below 75% but passed the exam ? Breakdown per course.**

```
WITH main AS
```

```
(SELECT
	ca.student_id,
	ca.schedule_id,
	s.course_id,
	co.name course_name,
	st.name,
	count(ca.id) as total_attendance,
	length(replace(s.course_days,',',''))*(week(s.end_dt)-week(s.start_dt))
total_course_day
FROM course_attendance ca
	LEFT JOIN schedule s on ca.schedule_id = s.id
	LEFT JOIN student st on ca.student_id = st.id
	LEFT JOIN course co on s.course_id = co.id
GROUP BY 1,2,3,4,5,7),

main2 AS
(SELECT student_id, name, course_name, (total_attendance/total_course_day*100)
attd_pctg
FROM main),

main3 AS
(SELECT *
from main2
where attd_pctg>75)

select name,student_id, course_name
from main3
WHERE student_id in(
select student_id
from exam_submission es
left join exam e on es.exam_id = e.id
WHERE es.score>e.pass_threshold)
```

Student Name	**Course Name**
Sisca	Data Mining
Horatio	Introduction to Applied Machine Learning
Horatio	Data Mining

Anthony	Data Mining
Anthony	Statistic
Anthony	Introduction to Applied Machine Learning
Michael	Introduction to Applied Machine Learning
Michael	Data Mining
Kiky	Natural Language Processing
Kiky	Data Mining
Grace	Applied Data Science

5. Who did resit and passed and what course was it ?

```
SELECT
        st.name AS student_name,
        co.name AS course_name,
        es.exam_id,
        es.score AS resit_score
FROM student st
JOIN exam_submission es
ON st.id = es.student_id
JOIN exam ex
ON es.exam_id = ex.id
JOIN course co ON ex.course_id = co.id
WHERE es.score >= ex.pass_threshold
   AND (
   SELECT COUNT(*)
   FROM exam_submission
   WHERE student_id = es.student_id AND exam_id = es.exam_id) > 1
   AND es.id IN
   (
      SELECT MAX(id)
      FROM exam_submission
      WHERE student_id = es.student_id AND exam_id = es.exam_id
      GROUP BY student_id, exam_id
   )
GROUP BY
   st.name, co.name, es.exam_id, es.score;
```

Student Name	Course Name
Sisca	Introduction to Applied Machine Learning

6. **Who completely failed the test and what course was it ?**

```
SELECT
        st.name AS student_name,
        co.name AS course_name,
        es.exam_id,
        MAX(es.score) AS highest_exam_score,
        ex.pass_threshold
FROM
        student st
JOIN
        exam_submission es ON st.id = es.student_id
JOIN
        exam ex ON es.exam_id = ex.id
JOIN
        course co ON ex.course_id = co.id
WHERE
        (SELECT COUNT(*)
FROM exam_submission WHERE student_id = es.student_id AND exam_id =
es.exam_id) > 1
GROUP BY
        st.name, co.name, es.exam_id, ex.pass_threshold
HAVING
        MAX(es.score) < ex.pass_threshold;
```

Student Name	Course Name
Michael	Python Programming

7. **Rank students based on the highest average score during the academic year 19/20.**
Note : for students who did resit, pick the maximum score. If the students have the same average scores, rank based on name alphabetically.

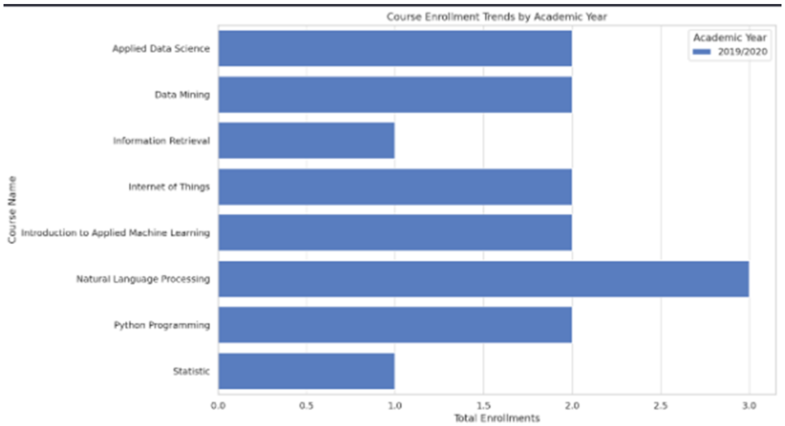
```
SELECT
        st.name AS student_name,
        AVG(best_score.max_score) AS average_score,
        RANK() OVER (ORDER BY AVG(best_score.max_score) DESC, st.name) AS
student_rank
FROM
        student st
JOIN
        (
        SELECT
        es.student_id,
        MAX(es.score) AS max_score
        FROM
        exam_submission es
        JOIN
        exam ex ON es.exam_id = ex.id
        JOIN
        schedule sch ON ex.course_id = sch.course_id
        JOIN
        enrollment enr ON sch.id = enr.schedule_id AND es.student_id =
enr.student_id
        WHERE
        enr.academic_year = '2019/2020'
        GROUP BY
        es.student_id, es.exam_id
        ) best_score ON st.id = best_score.student_id
GROUP BY
        st.name
ORDER BY
        average_score DESC,
        st.name;
```

Rank	Student Name	Average Score

1	Anthony	81.0
2	Kiky	81.0
3	Sisca	74.0
4	Grace	73.0
5	Horatio	59. 0
6	Michael	57.3

Task 2

As a data analyst, you are asked to showcase some analysis based on the given tables. Generate a chart with any metrics that you think are the most important for the university. Feel free to use any tools including Excel. Copy the charts below and elaborate your analysis.



Analysis of Course Distribution Over the Academic Year at a University

The visualization bar plot provides a detailed description of the distribution of student enrollment in various courses offered by the university during an academic year. Each bar in the plot represents the total number of student enrollments in a particular course, providing insight into which courses students were most interested in in that academic year.

Although the data only covers one academic year, this analysis still provides valuable insight into student interest in the courses offered. A course with a high number of enrollments indicates a high level of interest from students, which can be an indicator of the success or relevance of the course material.

Furthermore, this analysis also reveals enrollment patterns that may be influenced by factors such as course content, instructor reputation, or topic relevance to current industry and research trends.

The information obtained from this analysis is very useful for the university administration to understand student needs and interests. This allows universities to make strategic decisions in future course offerings, adapt educational resources, and develop curricula that are more aligned with student interests and needs.

This kind of analysis is important to ensure that universities continue to offer programs and courses that are relevant and engaging to students, ensuring the sustainability and growth of the institution.

Any questions for clarity, you can reach Tubagus via Whatsapp at 087887667574
Kindly read this doc thoroughly first before asking.
Thank you and good luck!