Practice Questions

Table: STUDENT

StudentID	FullName	Batch	AdvisorID
201	Areeba Khalid	2023	901
202	Fahad Aslam	2022	902
203	Sana Mehmood	2023	NULL
204	Hammad Tariq	2021	901
205	Rida Qureshi	2022	903

Table: ADVISOR

AdvisorID	Name	Dept
901	Dr. Salman	CS
902	Dr. Mehwish	SE
903	Dr. Zainab	CS
904	Dr. Naveed	EE

Table: COURSE_ENROLLMENT

EnrollmentID	StudentID	CourseCode	Grade
1	201	CS101	A
2	202	CS101	B
3	203	CS201	A
4	204	SE202	C
5	206	CS101	B

Question 1:

(a) Write a query to list the full names of students and the name of their advisor using `INNER JOIN`.

(b) Add a filter to show only students from the **CS department**.

(c) Modify the query to include students from **all departments**, but display “No Advisor” if a student has no assigned advisor.

Question 2:

(a) Write a query using a **LEFT JOIN** to list all students and the total number of courses they are enrolled in (if any).

(b) Include students who have not enrolled in any course.

(c) Sort the result by number of courses in descending order and use a meaningful alias for the column.

Question 3:

Write a query to show:

- Student full name
- Advisor name
- Course code they are enrolled in

Only show students who have both an advisor **and** at least one course enrollment.

Question 4:

(a) Identify course enrollments where the student **does not exist** in the STUDENT table.

(b) Modify the query to also show grades and student IDs.

(c) Why might such mismatches occur in real systems? How would you fix/prevent them?

Question 5:

Imagine a rule: **every advisor must be co-mentored by someone from the same department.**

(a) Write a **self join** query to list each advisor and their co-mentor from the same department (assume co-mentor has a different AdvisorID).

(b) Add a filter to **exclude** any advisor who has no available co-mentor.

(c) What change would you make to the schema to better support this rule structurally (hint: foreign key)?