1- What is the difference between GROUP BY and ORDER BY?

- GROUP BY:
  Groups rows that have the same values in specified columns into summary rows (like "total sales per product").
- ORDER BY:
  Sorts the result set of a query based on one or more columns or expressions, either ascending (ASC) or descending (DESC).

2- Why do we use HAVING instead of WHERE when filtering aggregate results?

- WHERE: Filters row *before* grouping/aggregation.
- HAVING: Filters groups *after* aggregation.

3- What are common beginner mistakes when writing aggregation queries?

- Forgetting to include all non-aggregated columns in the GROUP BY clause.
- Using WHERE instead of HAVING to filter aggregated values.
- Applying ORDER BY on non-existent or ambiguous columns after grouping.
- Confusing the order of SQL clauses (must be: SELECT → FROM → WHERE → GROUP BY → HAVING → ORDER BY).

4-When would you use COUNT(DISTINCT ...), AVG(...), and SUM(...) together?

- **When analyzing unique participation and performance:**
  - COUNT(DISTINCT StudentID) → number of unique students.
  - AVG(CompletionPercent) → average progress.
  - SUM(Price) → total revenue from enrollments.

5-How does GROUP BY affect query performance, and how can indexes help?

- GROUP BY requires sorting or hashing the data to group it, which can be resource-intensive on large tables.
- It can slow down performance if:
    - The table is large
    - There's no proper index on the grouping column(s)
- **Indexes improve performance by:**
    - Allowing faster retrieval and sorting of grouped values.
    - Supporting efficient lookups on columns used in GROUP BY or WHERE.