SQL Aggregation Practice Module

1. GROUP BY and ORDER BY in SQL

Two common statements in SQL that help with sorting your data are GROUP BY and ORDER BY.

The function of a GROUP BY statement is to group records with shared values. A GROUP BY statement is always used with an aggregate function in a query. Additionally, GROUP BY is useful for returning multiple desired results sorted by your specified group(s), rather than solely one column.

GROUP BY syntax

```
SELECT column_1, AGGREGATE_FUNCTION(column_2) FROM table GROUP BY column_1
```

The function of the ORDER BY statement is to sort results in ascending or descending order based on the column(s) you specify in the query. Depending on the data type stored by the column you specify after it, ORDER BY will organize them in alphabetical or numerical order.

ORDER BY syntax

```
SELECT column_1, column_2 FROM table ORDER BY column_1;
```

2. Uses of Filtering Clause (WHERE & HAVING)

The WHERE clause is used to filter individual rows before any grouping or aggregation takes place.

in contrast, the HAVING clause filters grouped records after aggregation. HAVING is often used alongside aggregate functions like SUM, COUNT, or AVG, allowing you to include only those groups that meet specific summary conditions.

3. Common Mistakes Beginners Make When Writing SQL Queries

When you're new to SQL, writing queries can feel like solving a tricky puzzle — and just like any puzzle, it's easy to make small mistakes that can leave you stuck and frustrated for hours.

- Writing a DELETE or UPDATE statement without specifying a WHERE condition.
- Assuming NULL behaves like a regular value.
- Mixing up INNER JOIN, LEFT JOIN, and RIGHT JOIN.
- Selecting non-aggregated columns without including them in GROUP BY.

4. When and Why to Use COUNT(DISTINCT), AVG, and SUM Together in SQL

Function	Description
COUNT **(**DISTINCT ...)	Counts the number of unique values in a column.
SUM(...)	Returns the total of numeric values in a column.
AVG(...)	Calculates the average of numeric values.

When to Use Them Together?

These three functions are often used together to gain a well-rounded understanding of a dataset—revealing how many unique items exist, the total amount of activity or value, and the average size or contribution of each item.

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

Using GROUP BY in SQL can impact performance because the database engine needs to scan data, sort or hash it by the grouping columns, and perform aggregations—tasks that become resource-heavy on large datasets.

However, performance can be significantly improved by creating indexes on the columns used in the GROUP BY clause.

Indexes help the database quickly locate and group relevant rows, reduce sorting overhead, and speed up overall execution. For best results, use covering indexes when possible and avoid selecting unnecessary columns.