1.INNER vs LEFT JOIN

INNER JOIN: Returns rows with a match in both tables by join condition.

LEFT JOIN: Returns all the rows from the left table and the matched rows from the right table; if no match, columns from the right table contain NULL.

2.FULL OUTER JOIN what is it?

A FULL OUTER JOIN returns all the rows from both tables, with common rows from each side where they exist. If there is no match, the unrepresented side is filled with NULL. It is practically a join of LEFT JOIN and RIGHT JOIN.

3. Are joins nestable?

Yes. Joins can be nested by joining the result of one join to another table (or join). This is usually done in queries with several relationships or step-by-step filtering.

4. How to join more than 2 tables?

By doing several joins in one query. You join the first two tables, then the result to the next table, and then the next result to the next table, etc., defining conditions for each join.

5. What is a CROSS JOIN?

A CROSS JOIN gives the Cartesian product of two tables — each row from the first table is combined with each row from the second table, with no join condition.

6. What is a NATURAL JOIN?

A NATURAL JOIN joins tables automatically based on all columns of the same name and of compatible data types. It omits the specification of the ON clause, but can be dangerous in case of unintended matching columns.

7. Is it possible to join tables without foreign key?

Yes. Foreign key is not required for joins — you can join on any columns with related data. The foreign key simply ensures referential integrity, but the SQL join is fine as long as there's a logical relationship.

8.What is a SELF JOIN

A SELF JOIN is where a table is joined against itself to match rows within the same table, frequently using table aliases to differentiate between the two occurrences.

9. Why does a Cartesian product happen?

A Cartesian product is when you join tables without an effective join condition (or use CROSS JOIN), causing each row from one table to be combined with every row from the other.

10 .How do I optimize joins?

- Use correct indexes on join columns.
- Join only necessary columns (don't use SELECT *).
- Filter rows in advance using WHERE or ON clauses.

- Use smaller intermediate result sets prior to joining big tables.
- Avoid unnecessary joins — eliminate unused tables.
- Make sure matching column data types to avoid conversions.