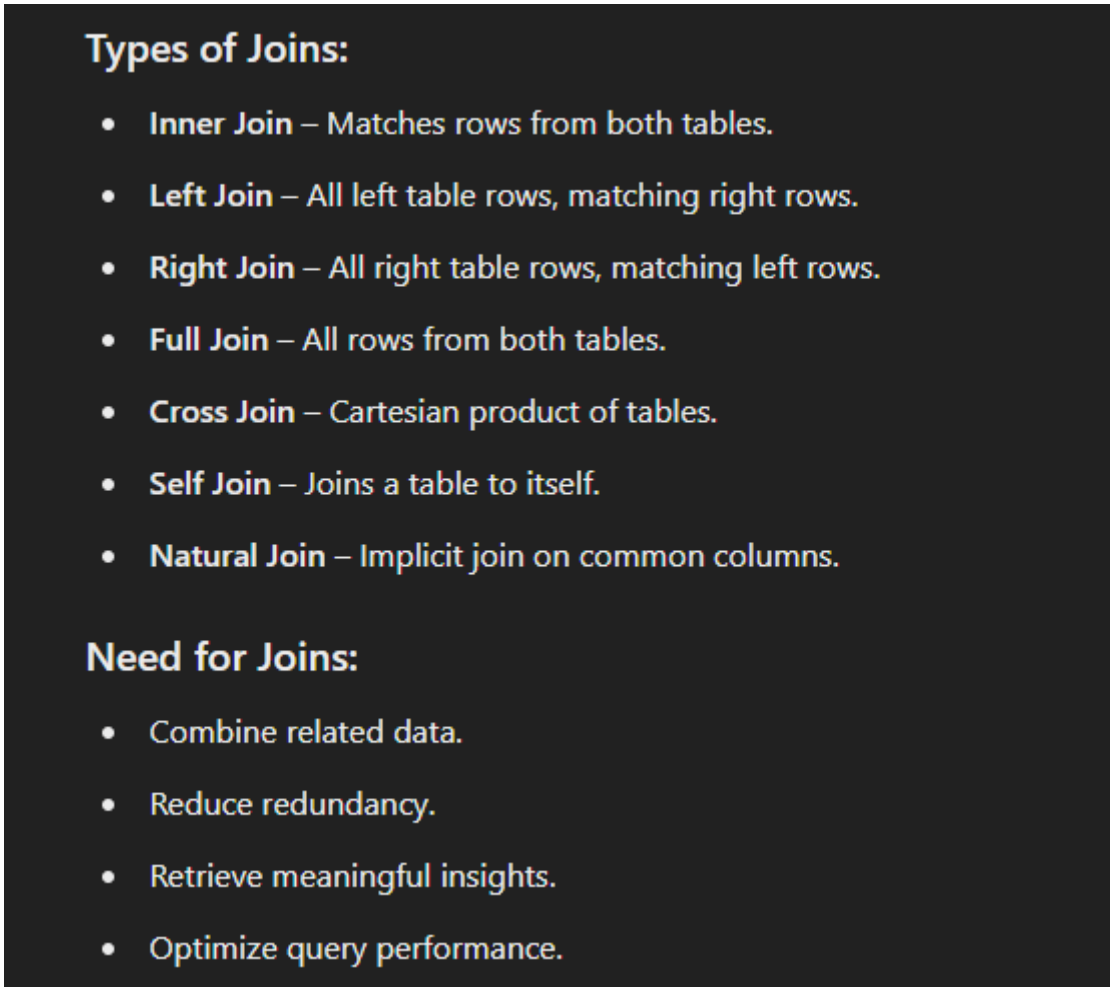
1. Discuss the various type of join operations ? Why are these join required.



2. Difference between the EXISTSand NOT EXISTS operators

EXISTS vs. NOT EXISTS

Operator	Description	Example
EXISTS	`TRUE` if subquery returns rows.	`SELECT * FROM Customers WHERE EXISTS (SELECT 1 FROM Orders WHERE Customers.id = Orders.customer_id);`
NOT EXISTS	`TRUE` if subquery returns no rows.	`SELECT * FROM Customers WHERE NOT EXISTS (SELECT 1 FROM Orders WHERE Customers.id = Orders.customer_id);`

3. Differentiate between SQL commands DROP TABLE and DROP VIEW.

DROP TABLE vs. DROP VIEW

DROP TABLE	DROP VIEW
Deletes a table and its data.	Deletes a view (virtual table).
Removes structure & records.	Removes only the view definition.
Permanently deletes data.	Views store no data.
`DROP TABLE table_name;`	`DROP VIEW view_name;`

4. What is the difference between WHERE and Having Clause ?

WHERE vs. HAVING

WHERE	HAVING
Filters rows before grouping.	Filters groups after aggregation.
Can't use with aggregate functions.	Uses aggregate functions.
`SELECT * FROM table WHERE condition;`	`SELECT column, COUNT(*) FROM table GROUP BY column HAVING condition;`

5. How are the nulls represented in database system?

- **NULL** represents missing or unknown values.
- **Not equal to zero or empty string.**
- **Stored as a special marker in databases.**
- **Handled using** `IS NULL` **or** `IS NOT NULL`**.**
- **Affects aggregate functions and comparisons.**

6. Describe the DDL, DML, DCL commands for the student's database, which contains student details: name, id, DOB, branch, DOJ, and course details: Course name, Course id, Stud Id, Faculty name, id, marks

DDL (Data Definition Language)

- **CREATE TABLE** – Defines tables for students and courses.

sql Copy Edit

```sql
CREATE TABLE Students (
    id INT PRIMARY KEY,
    name VARCHAR(50),
    DOB DATE,
    branch VARCHAR(50),
    DOJ DATE
);

CREATE TABLE Courses (
    course_id INT PRIMARY KEY,
    course_name VARCHAR(50),
    stud_id INT,
    faculty_name VARCHAR(50),
    faculty_id INT,
    marks INT,
    FOREIGN KEY (stud_id) REFERENCES Students(id)
);
```

- **ALTER TABLE** – Modifies table structure.

sql Copy Edit

```sql
ALTER TABLE Students ADD email VARCHAR(100);
```

- **DROP TABLE** – Deletes a table.

sql Copy Edit

```sql
DROP TABLE Students;
```

DML (Data Manipulation Language)

- **INSERT** – Adds data into tables.

```sql
INSERT INTO Students VALUES (1, 'John Doe', '2000-05-15', 'CSE', '2022-08-01');
```

- **UPDATE** – Modifies existing data.

```sql
UPDATE Students SET branch = 'ECE' WHERE id = 1;
```

- **DELETE** – Removes records.

```sql
DELETE FROM Students WHERE id = 1;
```

DCL (Data Control Language)

- **GRANT** – Provides access.

```sql
GRANT SELECT ON Students TO user1;
```

- **REVOKE** – Removes access.

```sql
REVOKE SELECT ON Students FROM user1;
```

7. What is the purpose of group by clause in the SELECT statement?

Purpose of GROUP BY Clause:

- Groups rows with the same values in specified columns.
- Used with aggregate functions (`COUNT`, `SUM`, `AVG`, `MAX`, `MIN`).
- Helps in summarizing data.

Example:

sql Copy Edit

```sql
SELECT branch, COUNT(*)
FROM Students
GROUP BY branch;
```

Output: (Counts students in each branch)

Branch	Count
CSE	10
ECE	8

8. Demonstrate various SQL Clauses like WHERE,DISTINCT,ORDER BY,GROUP BY AND HAVING

SQL Clauses Demonstration

1. WHERE Clause (Filters rows based on a condition)

sql Copy Edit

```sql
SELECT * FROM Students WHERE branch = 'CSE';
```

Retrieves students from the CSE branch.

2. DISTINCT Clause (Removes duplicate values)

sql Copy Edit

```sql
SELECT DISTINCT branch FROM Students;
```

3. ORDER BY Clause (Sorts result in ascending/descending order)

sql Copy Edit

```sql
SELECT * FROM Students ORDER BY name ASC;
```

Sorts students by name in ascending order.

4. GROUP BY Clause (Groups rows based on column values)

sql Copy Edit

```sql
SELECT branch, COUNT(*) FROM Students GROUP BY branch;
```

Counts students in each branch.

5. HAVING Clause (Filters groups after aggregation)

sql Copy Edit

```sql
SELECT branch, COUNT(*) FROM Students GROUP BY branch HAVING COUNT(*) > 5;
```