Basic PostgreSQL Aggregate Functions

COUNT(*)	Count all rows	SELECT COUNT(*) FROM users;
SUM(column)	Sums numeric values.	SELECT SUM(salary) FROM employees;
AVG(column)	Average value	SELECT AVG(score) FROM exams;
MAX(column)	Largest value	SELECT MAX(price) FROM products;
MIN(column)	Smallest value	SELECT MIN(price) FROM products;

PostgreSQL Constraints

1. PRIMARY KEY

Uniquely identifies each row; cannot be NULL.

2. UNIQUE

Ensures no two rows have the same value in that column.

3. NOT NULL

The column must always have a value.

4. CHECK

Allows only values that meet a specific condition.

Syntax: age INT CHECK (age > 0)

5. FOREIGN KEY

Ensures the value exists in another table (links tables safely).

6. DEFAULT

Automatically fills the column with a given value if none is provided.

PostgreSQL Sorting

1. Basic Sorting

```
SELECT * FROM table_name
ORDER BY column_name;
```

2. Sort Descending

```
ORDER BY column_name DESC;
```

3. Sort Ascending

```
ORDER BY column_name ASC;
```

Create a View

```
CREATE VIEW view_name AS
SELECT column1, column2
FROM table_name
WHERE condition;
```

Simple Function (no parameters)

```
CREATE FUNCTION func_name()
RETURNS datatype AS $$
BEGIN
   RETURN value;
END;
$$ LANGUAGE plpgsql;
```

Function with Parameters

```
CREATE FUNCTION func_name(param1 datatype, param2 datatype)
RETURNS datatype AS $$
BEGIN
   RETURN param1 + param2;
END;
$$ LANGUAGE plpgsql;
```

Q. Function to check wheater number is even or odd

```
CREATE OR REPLACE FUNCTION check_even_odd(num INT)
RETURNS TEXT AS $$
BEGIN
   IF num % 2 = 0 THEN
      RAISE NOTICE 'Even';
      RETURN 'Even';
   ELSE
      RAISE NOTICE 'Odd';
      RETURN 'Odd';
   END IF;
END;
$$ LANGUAGE plpgsql;
```

Q. Function to find greatest number between 3 numbers

```
CREATE OR REPLACE FUNCTION greatest_of_three(a INT, b INT, c INT)
RETURNS INT AS $$
BEGIN
   IF a >= b AND a >= c THEN
      RETURN a;
   ELSIF b >= a AND b >= c THEN
      RETURN b;
   ELSE
      RETURN c;
   END IF;
END;
$$ LANGUAGE plpgsql;
```

Q. Function to add two numbers

```
CREATE OR REPLACE FUNCTION add_two(a INT, b INT)
RETURNS INT AS $$
BEGIN
   RETURN a + b;
END;
$$ LANGUAGE plpgsql;
```

Q. Function to find greatest number between two numbers using ELSEIF

```
CREATE or replace FUNCTION elsif_demo () RETURNS void AS'

DECLARE

x integer := 72;

y integer := 72;

BEGIN

IF x > y THEN

RAISE NOTICE ''% is greater than %'',x, y;

ELSIF x < y THEN

RAISE NOTICE ''% is less than %'',x, y;

ELSE

RAISE NOTICE ''% is equal to %'',x, y;

END IF;

END;

$$ LANGUAGE 'plpgsql';
```