Important Inbuilt functions of SQL

1. COUNT()

Purpose: Returns the number of rows that match a specified criterion.

Query:

```
SELECT COUNT(*) FROM Employees WHERE Department = 'Sales';
```

Explanation: Counts the total number of employees in the 'Sales' department.

---

2. SUM()

Purpose: Returns the total sum of a numeric column.

Query:

```
SELECT SUM(Salary) FROM Employees WHERE Department = 'HR';
```

Explanation: Adds up the salary of all employees in the 'HR' department.

---

3. AVG()

Purpose: Returns the average value of a numeric column.

Query:

```
SELECT AVG(Salary) FROM Employees WHERE Department = 'IT';
```

Explanation: Finds the average salary of all employees in the 'IT' department.

---

4. MIN()

Purpose: Returns the smallest value in a column.

Query:

SELECT MIN(Salary) FROM Employees;

Explanation: Finds the employee with the minimum salary.

---

5. MAX()

Purpose: Returns the largest value in a column.

Query:

SELECT MAX(Salary) FROM Employees;

Explanation: Finds the employee with the highest salary.

---

6. LEN() or LENGTH()

Purpose: Returns the length of a string.

Query:

SELECT LENGTH(EmployeeName) FROM Employees;

Explanation: Returns the length of each employee's name.

---

7. UPPER()

Purpose: Converts a string to uppercase.

Query:

SELECT UPPER(EmployeeName) FROM Employees;

Explanation: Converts the employee names to uppercase.

---

8. LOWER()

Purpose: Converts a string to lowercase.

Query:

```
SELECT LOWER(EmployeeName) FROM Employees;
```

Explanation: Converts the employee names to lowercase.

---

9. ROUND()

Purpose: Rounds a numeric value to the specified number of decimal places.

Query:

```
SELECT ROUND(Salary, 2) FROM Employees;
```

Explanation: Rounds the salary of each employee to 2 decimal places.

---

10. NOW()

Purpose: Returns the current date and time.

Query:

```
SELECT NOW();
```

Explanation: Retrieves the current date and time from the database system.

---

11. GETDATE()

Purpose: Returns the current date and time in SQL Server (similar to NOW() in MySQL).

Query:

```
SELECT GETDATE();
```

Explanation: Retrieves the current date and time.

---

12. DATEDIFF()

Purpose: Returns the difference between two dates.

Query:

```
SELECT DATEDIFF(CURRENT_DATE, HireDate) FROM Employees;
```

Explanation: Calculates the number of days since an employee was hired.

---

13. DATEADD()

Purpose: Adds a specified time interval to a date.

Query:

```
SELECT DATEADD(YEAR, 1, HireDate) FROM Employees;
```

Explanation: Adds 1 year to the hire date of each employee.

---

14. COALESCE()

Purpose: Returns the first non-null expression.

Query:

```
SELECT COALESCE(MiddleName, 'N/A') FROM Employees;
```

Explanation: If the middle name is NULL, returns 'N/A' instead.

---

15. ISNULL()

Purpose: Replaces NULL with a specified replacement value.

Query:

SELECT ISNULL(MiddleName, 'Not Provided') FROM Employees;

Explanation: If MiddleName is NULL, it will return 'Not Provided'.

---

16. CONCAT()

Purpose: Concatenates two or more strings.

Query:

SELECT CONCAT(FirstName, ' ', LastName) FROM Employees;

Explanation: Combines the first name and last name of employees into one string.

---

17. TRIM()

Purpose: Removes leading and trailing spaces from a string.

Query:

SELECT TRIM(EmployeeName) FROM Employees;

Explanation: Removes any spaces from the beginning and end of employee names.

---

18. LTRIM()

Purpose: Removes leading spaces from a string.

Query:

SELECT LTRIM(EmployeeName) FROM Employees;

Explanation: Removes spaces from the left side of employee names.

---

19. RTRIM()

Purpose: Removes trailing spaces from a string.

Query:

```
SELECT RTRIM(EmployeeName) FROM Employees;
```

Explanation: Removes spaces from the right side of employee names.

---

20. SUBSTRING()

Purpose: Extracts a substring from a string.

Query:

```
SELECT SUBSTRING(EmployeeName, 1, 3) FROM Employees;
```

Explanation: Extracts the first 3 characters from each employee's name.

---

21. IN()

Purpose: Determines if a value matches any value in a list or a subquery.

Query:

```
SELECT * FROM Employees WHERE Department IN ('HR', 'IT');
```

Explanation: Selects all employees who work in either HR or IT departments.

---

22. BETWEEN()

Purpose: Filters records within a range.

Query:

```
SELECT * FROM Employees WHERE Salary BETWEEN 50000 AND 80000;
```

Explanation: Selects employees whose salaries are between 50,000 and 80,000.

---

23. LIKE()

Purpose: Searches for a pattern in a column.

Query:

```
SELECT * FROM Employees WHERE EmployeeName LIKE 'J%';
```

Explanation: Selects all employees whose names start with 'J'.

---

24. DISTINCT()

Purpose: Removes duplicate values from the result set.

Query:

```
SELECT DISTINCT Department FROM Employees;
```

Explanation: Returns a list of unique departments without duplicates.

---

25. CAST()

Purpose: Converts one data type to another.

Query:

```
SELECT CAST(Salary AS VARCHAR) FROM Employees;
```

Explanation: Converts the salary from numeric to string (varchar) format.

---

26. CONVERT()

Purpose: Similar to CAST(), but with more flexibility (in SQL Server).

Query:

SELECT CONVERT(VARCHAR, HireDate, 103) FROM Employees;

Explanation: Converts the hire date to a string format (DD/MM/YYYY).

---

27. NULLIF()

Purpose: Compares two expressions and returns NULL if they are equal.

Query:

SELECT NULLIF(Salary, 0) FROM Employees;

Explanation: If an employee's salary is 0, it returns NULL.

---

28. GROUP_CONCAT()

Purpose: Concatenates values from multiple rows into a single string (MySQL specific).

Query:

SELECT GROUP_CONCAT(EmployeeName) FROM Employees;

Explanation: Concatenates the names of all employees into a single string.

---

29. RANK()

Purpose: Assigns a rank to each row in a result set, with ties receiving the same rank.

Query:

SELECT EmployeeName, RANK() OVER (ORDER BY Salary DESC) AS Rank FROM Employees;

Explanation: Ranks employees based on their salary in descending order.

---

30. NTILE()

Purpose: Divides a result set into a specified number of buckets.

Query:

```
SELECT EmployeeName, NTILE(4) OVER (ORDER BY Salary DESC) AS Quartile FROM Employees;
```

Explanation: Divides employees into 4 quartiles based on their salary.