1, 1, 2, 3, 5, 8, 13, 21

DDL [Data Query Language]	DML (Data Manipulation Language)	DCL [Data Control Language]	TCL [Data Transation Language]	DQL [Data Query Language]
Deals with the Structure & Schema of database.	Manipulate data within Database	Control access to data in database	Manage transactions & ensure data consistency & Integrity.	-Select Query data from the database.
-Create	-Select	-Grant (Give users specific privilage)	-Commit	
-Alter	-Insert	-Revoke (Remove privilage from a users)	-Rollback	
-Drop	-Update		-Save point	
-Truncate	-Delete		-Set Transaction	

describe table_name; → Provide details of table.

Features	DBMS	RDBMS
Data Structure	Data can be stored in any structure (e.g. files, object)	Data is stored in tables with rows & columns (relation model).
Data Model	Can support hierarchical, network, or object model.	Strictly follows the relational model (table & relationship).
Data Integrity	Integrity constraints are minimal.	Enforce data Integrity through key, constraints & rules.
Normalization.	No inherient support for data normalization.	Support data normalization to eliminate redundency.
ACID compliance (Atomicity, Consistency, Isolation, Durability)	May not fully support ACID property.	Fully supports ACID properties for transaction management.

Query Language	May not have a standard query language.	Uses SQL (Structured Query Language) for querying & managing data.
Relationships	No or limited support for relationships b/w data.	Supports relationships b/w tables through foreign keys.
Examples	File system, XML-based DBMS, hierarchical DBMS.	MySQL, Oracle, PostgreSQL, SQL SERVER, SQLLITE.

1.> Concat() → concat(S1, S2, ...)
→ Combine two or more string into one..

2.> concat_ws() → concat_ws(seperator, S1, S2 ...)
→ combine two or more string with seperator.

3.> Substring() or substr() → substring(string, start, length)

Select & substring('HelloWorld', 2, 5) → elloW

4.> Length() → Returns the numbes of bytes in a string.

Select ('Hello'): → 5

5.> CHAR_LENGTH() or CHARACTE_LENGTH() →

Returns the numbes of a character in a string, counting mult-byte characters as individual characters.

Select CHAR_LENGTH('Hello') → 5

6.> TRIM() → TRIM([Removal_String] For string)

Remove leading and trailing space from a string.

Select TRIM(' Hello '); → Hello

Select TRIM(leading 'z' from 'zzzhelzozz') → helzozz

Select TRIM(trailing 'z' from 'zzhelzozz') → zzhelzo

Select TRIM('z' from 'zzhelzozz') → helzo

7.> LTRIM() → LTRIM(string)

Remove leading spaces from a string.

Select LTRIM(' Hello'); → Hello

8) RTRIM() → RTRIM(string)

Remove trailing space from a string.

Select RTRIM(' Hello '); → Hello

9.> UPPER() → Upper(string) → converts a string to uppercase.

10.> lower() → LOWER(string) → Convert a string to lowercase.

11.> REPLACE() → Replace(string, old_substring, new_substring)

Replace occurrences of a substring within a string with another substring.

Select Replace('Hello World', 'World', 'SQL'); → 'Hello SQL'

12.> INSTR() → instr(string, substring)

Returns the position of the first occurance of a substring in a string.
If the substring not found, it returns 0.
Select instr('Hello World', 'World'); → 7.

13.> LOCATE() → locate(substring, string)

locate() functions works similar to instr(), returning the position of the first occurence of a substring.

Select Locate('world', 'Hello World'); → 7.

14.> LEFT() → Left(string, N)

Returns the first N characters from a string.

SELECT LEFT('Hello World', 5) → Hello

15.> RIGHT() → Right(string, N)
Returns the last N characters from the string.
Select right('Hello World', 5) → World

16.> REVERSE() → reverse(string)
→ Returns reverse of a string
Select reverse('Hello'); → olleH

17.> ASCII() → ASCII(string)
→ Returns the ASCII value of the first character of a string.
SELECT ASCII('A'); → 65.

18.> CHARC) → CHAR (S1, S2, S3_ _ _ SN)

↳ Returns the character of a given ASCII code.

SELECT CHAR(65, 66, 67); → 'ABC'

19.> FORMATC) → Format (number, decimal places)

↳ ~~Returns~~ Format a number as a string, rounding it to a specified number of a decimal places.

Select format (1234.67895, 2); → 1234.68

20.> STRING_AGGC) → string_agg(expression, seperator)

↳ This function is used to concatenate values from multiple rows into a string

SELECT STRING_AGG (name, ',') as names from employees;

↳ 'John, Alice, Bob'