SQL Terminology and Definitions

Core Concepts
---
1. Database:
A structured collection of data stored electronically. Examples include MySQL, PostgreSQL, and SQL Server.

2. Table:
A collection of related data organized into rows and columns.
Example: A "Customers" table with columns like ID, Name, and Email.

3. Row (Record):
A single entry in a table representing one instance of data.

4. Column (Field):
A specific attribute or data point stored in a table.

5. Primary Key (PK):
A unique identifier for each row in a table (e.g., an "ID" column in a Customers table).

6. Foreign Key (FK):
A column in one table that links to the primary key of another table, establishing a relationship between tables.

7. Query:
A request to retrieve or manipulate data from a database.
Example: SELECT * FROM Customers;

SQL Commands
---
1. SELECT:
Retrieves data from one or more tables.
Example:
SELECT * FROM Customers;

2. INSERT:
Adds new records to a table.
Example:
INSERT INTO Customers (ID, Name, Email) VALUES (1, 'John Doe', 'john@example.com');

3. UPDATE:
Modifies existing records.
Example:
UPDATE Customers SET Email = 'newemail@example.com' WHERE ID = 1;

4. DELETE:
   Removes records from a table.
   Example:
   DELETE FROM Customers WHERE ID = 1;

5. WHERE:
   Filters records based on a specified condition.
   Example:
   SELECT * FROM Customers WHERE Name = 'John Doe';

6. ORDER BY:
   Sorts query results.
   Example:
   SELECT * FROM Customers ORDER BY Name ASC;

7. GROUP BY:
   Groups rows that have the same values in specified columns.
   Example:
   SELECT COUNT(ID), Country FROM Customers GROUP BY Country;

8. HAVING:
   Filters groups created by the GROUP BY clause.
   Example:
   SELECT COUNT(ID), Country FROM Customers GROUP BY Country HAVING COUNT(ID) > 5;

9. JOIN:
   Combines rows from two or more tables based on related columns.
   Example:
   SELECT Orders.OrderID, Customers.Name
   FROM Orders
   INNER JOIN Customers ON Orders.CustomerID = Customers.ID;

Additional Terminology & Concepts
---
1. Data Definition Language (DDL):
   A subset of SQL commands used to define and modify database structures.
   Examples: CREATE, ALTER, DROP.

2. Data Manipulation Language (DML):
   Commands used to manipulate data within tables.
   Examples: SELECT, INSERT, UPDATE, DELETE.

3. Data Control Language (DCL):
   Commands that control access to data within the database.
   Examples: GRANT, REVOKE.

4. Transaction:
A series of SQL commands executed as a single unit of work, ensuring that either all commands succeed or none do.

5. COMMIT:
Finalizes a transaction, saving all changes.
Example:
COMMIT;

6. ROLLBACK:
Reverts a transaction to its previous state if an error occurs.
Example:
ROLLBACK;

7. Constraint:
Rules enforced on data columns to ensure data integrity.
Examples: UNIQUE, NOT NULL, CHECK, DEFAULT.

8. Index:
A performance optimization structure that speeds up data retrieval by creating quick lookups for rows.

9. View:
A virtual table created by a query that provides a specific representation of the data.
Example:
CREATE VIEW CustomerView AS
SELECT Name, Email FROM Customers;

10. Stored Procedure:
A set of SQL statements stored in the database that can be executed on demand.

11. Trigger:
SQL code that automatically runs in response to specific events on a table (e.g., insertions, updates, or deletions).

12. Schema:
The organizational blueprint of the database, including tables, views, and relationships.

13. Data Types:
Define the kind of data a column can hold.
Examples: INTEGER, VARCHAR, DATE, BOOLEAN.

14. Aggregate Functions:

Functions that perform calculations on a set of values to return a single value.
Examples: COUNT(), SUM(), AVG(), MIN(), MAX().

15. DISTINCT:
A keyword used to return only unique values from a query.
Example:
SELECT DISTINCT Country FROM Customers;

16. LIMIT / OFFSET:
Clauses to restrict the number of rows returned by a query.
Example (MySQL):
SELECT * FROM Customers LIMIT 10 OFFSET 5;
(Note: SQL Server may use TOP instead.)

17. UNION / UNION ALL:
Operators that combine results from multiple SELECT queries.
- UNION removes duplicate rows, while UNION ALL includes all duplicates.
Example:
SELECT Name FROM Customers
UNION
SELECT Name FROM Employees;

18. Subquery (Nested Query):
A query embedded within another SQL query.
Example:
SELECT Name FROM Customers
WHERE ID IN (SELECT CustomerID FROM Orders WHERE OrderTotal > 100);

19. Alias:
A temporary name for a table or column to simplify queries, often using the AS keyword.
Example:
SELECT Name AS CustomerName FROM Customers;

20. NULL:
A marker used to indicate that a data value does not exist in the database.

21. LIKE:
An operator used for pattern matching in string comparisons.
Example:
SELECT * FROM Customers WHERE Name LIKE 'J%n';

22. IN:
Specifies multiple possible values for a column in a WHERE clause.
Example:
SELECT * FROM Customers WHERE Country IN ('USA', 'Canada', 'UK');

23. BETWEEN:
Filters data within a specific range.
Example:
SELECT * FROM Orders WHERE OrderTotal BETWEEN 50 AND 150;

24. CASE Statement:
Performs conditional logic within SQL queries.
Example:
```
SELECT Name,
      CASE
        WHEN Age >= 18 THEN 'Adult'
        ELSE 'Minor'
      END AS AgeGroup
FROM Customers;
```

25. COALESCE:
Returns the first non-null value in a list.
Example:
SELECT COALESCE(Phone, 'No phone provided') AS ContactPhone FROM Customers;

26. EXPLAIN:
A command used to analyze and understand the performance of a query by showing its execution plan.
Example:
EXPLAIN SELECT * FROM Customers WHERE Name = 'John Doe';

27. Transaction Isolation Levels:
Define how transactions interact with each other to maintain data integrity.
Examples: READ COMMITTED, REPEATABLE READ, SERIALIZABLE.

28. ACID Properties:
Fundamental principles ensuring reliable transaction processing:
- Atomicity: Each transaction is all or nothing.
- Consistency: Transactions lead the database from one valid state to another.
- Isolation: Transactions do not interfere with each other.
- Durability: Once a transaction is committed, it remains so.