

Basic Questions

1. What is DBMS?

A DBMS (Database Management System) is software that enables the creation, management, and manipulation of databases.

2. What is RDBMS?

RDBMS (Relational Database Management System) is a type of DBMS that organizes data into structured tables with relationships.

3. What is a primary key?

A primary key is a unique identifier for each record in a database table, ensuring no duplicate entries.

4. What is a foreign key?

A foreign key is a column that creates a relationship between two tables by referencing the primary key of another table.

5. What is normalization?

Normalization is the process of organizing data to reduce redundancy and improve data integrity.

6. What are the different types of normalization?

The different types of normalization include First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), and Boyce-Codd Normal Form (BCNF).

7. What is a transaction?

A transaction is a sequence of operations performed as a single logical unit of work that must be completed successfully or not at all.

8. What is ACID?

ACID (Atomicity, Consistency, Isolation, Durability) is a set of properties that ensure reliable transaction processing in a database.

9. What is SQL?

SQL (Structured Query Language) is a standard programming language used to manage and manipulate relational databases.

10. What is a join?

A join is an SQL operation that combines rows from two or more tables based on a related column.

11. What is indexing?

Indexing is a database optimization technique that improves the speed of data retrieval operations.

12. What is a view?

A view is a virtual table created by a query that selects data from one or more tables.

13. What is a stored procedure?

A stored procedure is a precompiled collection of SQL statements stored in the database for reuse.

14. What is a cursor?

A cursor is a database object used to retrieve, manipulate, and navigate through a result set one row at a time.

15. What is data integrity?

Data integrity refers to the accuracy and consistency of data within a database over its lifecycle.

DEFINITION

1. Open Source: Software with accessible source code for anyone to modify and distribute, promoting collaboration and transparency.

2. NOT NULL: A database constraint that ensures a column must always have a value and cannot contain NULL.

3. Primary Key: A column (or set of columns) that uniquely identifies each record in a table and cannot be NULL.

4. Foreign Key: A column in one table that references the primary key of another table, establishing a relationship between them.
5. Unique Key: A constraint ensuring all values in a column are unique, allowing NULL values (depending on the database).
6. GRANT: A command to provide specific privileges to a user in a database.
7. COMMIT: A command that saves all changes made during the current transaction permanently.
8. ROLLBACK: A command that undoes all changes made during the current transaction.
9. REVOKE: A command that removes previously granted privileges from a user in a database.
10. DDL (Data Definition Language): SQL commands used to define and modify database structures.
11. DML (Data Manipulation Language): SQL commands used to manipulate data within tables.
12. DCL (Data Control Language): Commands used to control user access and permissions in a database.
13. TCL (Transaction Control Language): Commands used to manage transactions in a database for data integrity.
14. DQL (Data Query Language): SQL commands used to query and retrieve data from a database.
15. DBMS (Database Management System): Software that enables the creation, management, and manipulation of databases.
16. RDBMS (Relational Database Management System): A type of DBMS that stores data in structured tables and maintains relationships between them.

- 17. INNER JOIN: Returns only the rows with matches in both tables.
- 18. LEFT JOIN: Returns all rows from the left table and matched rows from the right table.
- 19. RIGHT JOIN: Returns all rows from the right table and matched rows from the left table.
- 20. FULL JOIN: Returns rows when there is a match in either table.
- 21. CROSS JOIN: Returns the Cartesian product of two tables, combining every row from both.
- 22. SELF JOIN: Joins a table to itself to compare rows within the same table.
- 23. Procedural Language: Allows writing a sequence of instructions with control structures for data operations.
- 24. Sequential Language: Executes commands in a linear sequence without branching or looping.
- 25. SQL: Structured Query Language used for managing and manipulating relational databases.
- 26. PL/SQL: Procedural Language/Structured Query Language, an extension of SQL used in Oracle for procedural programming.
- 27. Clause: Components of a SQL statement that specify conditions or criteria for querying or manipulating data.
- 28. Cursor: A database object that allows row-by-row processing of query results, enabling detailed manipulation.

Syntax

1. CREATE TABLE

Defines a new table.

```
CREATE TABLE table_name (  
    column_name data_type [constraints]  
);
```

2. INSERT INTO

Inserts data into a table.

```
INSERT INTO table_name (column1, column2) VALUES (value1, value2);
```

3. SELECT

Retrieves data from a table.

```
SELECT column1, column2 FROM table_name WHERE condition;
```

4. UPDATE

Modifies existing data in a table.

```
UPDATE table_name SET column1 = value1 WHERE condition;
```

5. DELETE

Removes data from a table.

```
DELETE FROM table_name WHERE condition;
```

6. ALTER TABLE

Modifies the structure of an existing table.

```
ALTER TABLE table_name ADD column_name data_type;
```

7. DROP TABLE

Deletes a table and its data.

```
DROP TABLE table_name;
```

8. CREATE INDEX

Creates an index to improve query performance.

```
CREATE INDEX index_name ON table_name (column_name);
```

9. COMMIT

Saves all changes made in the current transaction.

```
COMMIT;
```

10. ROLLBACK

Reverts all changes made in the current transaction.

```
ROLLBACK;
```

Example

1. SELECT:

Retrieves data from a database.

Example:

```
SELECT * FROM employees;
```

2. INSERT:

Adds new data to a table.

Example:

```
INSERT INTO employees (employee_id, employee_name) VALUES (101, 'John Doe');
```

3. UPDATE:

Modifies existing data in a table.

Example: UPDATE employees SET salary = 5000 WHERE employee_id = 101;

4. DELETE:

Removes data from a table.

Example: DELETE FROM employees WHERE employee_id = 101;

5. CREATE TABLE:

Creates a new table in the database.

Example: CREATE TABLE employees (employee_id INT, employee_name VARCHAR(100));

6. ALTER TABLE:

Modifies an existing table structure.

Example: ALTER TABLE employees ADD department VARCHAR(50);

7. DROP TABLE:

Deletes a table and its data.

Example: DROP TABLE employees;

8. CREATE INDEX:

Creates an index to improve query performance.

Example: CREATE INDEX idx_name ON employees(employee_name);

9. JOIN:

Combines rows from two tables based on a related column.

Example: `SELECT * FROM employees JOIN departments ON employees.department_id = departments.department_id;`

10. GRANT:

Gives user permissions.

Example: `GRANT SELECT ON employees TO user_name;`

Codd's 12 rules (proposed by Edgar F. Codd) define the basic criteria that a system must meet to be considered a true relational database management system (RDBMS). Here's a summary of the rules:

1. Information Rule: All data is stored in tables (relations) as values within rows and columns.
2. Guaranteed Access Rule: Every piece of data is accessible through a combination of table name, primary key, and column name.
3. Systematic Treatment of NULL Values: NULL must be uniformly treated as missing or inapplicable information.
4. Dynamic Online Catalog: The metadata (data about the database structure) must be stored as a table and accessible via SQL.
5. Comprehensive Data Sublanguage Rule: The system must support a comprehensive language (like SQL) for data definition, manipulation, and transaction control.
6. View Updating Rule: Views (virtual tables) should be updatable like base tables.
7. High-Level Insert, Update, and Delete: The system must support operations on sets of rows (not just single rows).
8. Physical Data Independence: Changes in physical storage should not affect how data is accessed.

9. Logical Data Independence: Changes in table structures should not affect how data is accessed.

10. Integrity Independence: Integrity constraints should be independent of application programs and stored in the catalog.

11. Distribution Independence: The user should not be aware of whether the database is distributed across multiple locations.

12. Non-Subversion Rule: No low-level access (e.g., bypassing the RDBMS rules) should subvert the integrity of the database.

These rules outline the characteristics of a fully relational database system.