Q1. What is a database? Differentiate between SQL and NoSQL databases.

Ans: A database is a structured collection of data that is organized and stored in a computer system. It is designed to efficiently manage, store, and retrieve information. Databases can store various types of data, such as text, numbers, images, and more, and they provide mechanisms for defining the structure of the data and for processing and querying that data.

**SQL (Structured Query Language) Databases:**

* **Structure:** SQL databases are relational databases that use a structured schema with tables to organize and store data.
* **Data Model:** SQL databases use a predefined schema, where tables have predefined columns and data types, and relationships between tables are established using foreign keys.
* **Scalability:** Traditionally, scaling SQL databases vertically (adding more power to an existing machine) is common, but this has limitations.
* **Examples:** MySQL, PostgreSQL, Oracle, Microsoft SQL Server.

**NoSQL (Non-relational) Databases:**

* **Structure:** NoSQL databases do not use a fixed schema. They can store unstructured or semi-structured data, allowing for more flexibility in handling different types of data.
* **Data Model:** NoSQL databases use various data models such as document-oriented, key-value pairs, column-family, or graph formats.
* **Scalability:** NoSQL databases are often designed for horizontal scalability, allowing them to handle large amounts of data by adding more servers to a distributed database.
* **Examples:** MongoDB (document-oriented), Cassandra (column-family), Redis (key-value), Neo4j (graph).

Q2. What is DDL? Explain why CREATE, DROP, ALTER, and TRUNCATE are used with an example.

Ans: DDL, or Data Definition Language, is a subset of SQL (Structured Query Language) that deals with the structure and definition of the database objects. It includes commands for creating, altering, and deleting database objects such as tables, indexes, and schemas. Common DDL commands include CREATE, ALTER, DROP, and TRUNCATE.

**CREATE:** The CREATE command is used to create new database objects, such as tables, indexes, or views.

**Example:**

CREATE TABLE employees ( employee\_id INT PRIMARY KEY, first\_name VARCHAR(50), last\_name VARCHAR(50), salary DECIMAL(10, 2) );

**DROP:** The DROP command is used to remove existing database objects, such as tables, indexes, or views. **Example:** DROP TABLE employees;

**ALTER:** The ALTER command is used to modify the structure of an existing database object, such as adding or removing columns from a table.

**Example:** ALTER TABLE employees ADD COLUMN email VARCHAR(100);

1. **TRUNCATE:** The TRUNCATE command is used to remove all rows from a table while keeping the table structure for future use. It is faster than the DELETE command as it doesn't log individual row deletions. **Example:** TRUNCATE TABLE employees;

Q3. What is DML? Explain INSERT, UPDATE, and DELETE with an example.

Ans: DML, or Data Manipulation Language, is a subset of SQL (Structured Query Language) that deals with the manipulation of data stored in the database. It includes commands for inserting, updating, and deleting data within database objects, primarily tables.

**INSERT:** The INSERT command is used to add new rows of data into a table.

**Example:** INSERT INTO employees (employee\_id, first\_name, last\_name, salary) VALUES (1, 'John', 'Doe', 50000);

**UPDATE:** The UPDATE command is used to modify existing data in a table.

**Example:** UPDATE employees SET salary = 55000 WHERE employee\_id = 1;

**DELETE:** The DELETE command is used to remove rows from a table based on a specified condition.

**Example:** DELETE FROM employees WHERE employee\_id = 1;

Q4. What is DQL? Explain SELECT with an example.

Ans: DQL, or Data Query Language, is a subset of SQL (Structured Query Language) that deals with the retrieval of data from a database. The primary command used in DQL is SELECT, which allows users to query and fetch data from one or more tables in the database.

**Basic SELECT:** SELECT \* FROM employees;

This query retrieves all columns from the "employees" table for all records. The asterisk (\*) is a wildcard character that represents all columns.

**SELECT with Conditions:** SELECT first\_name, last\_name, salary FROM employees WHERE department = 'IT';

This query retrieves the first name, last name, and salary of employees from the "IT" department.

**SELECT with Aggregation:** SELECT department, AVG(salary) AS average\_salary FROM employees GROUP BY department;

This query calculates the average salary for each department and displays the results.

**SELECT with Sorting:** SELECT first\_name, last\_name, salary FROM employees ORDER BY salary DESC;

This query retrieves the first name, last name, and salary of employees, ordering the results by salary in descending order.

**SELECT with Joins:** SELECT employees.first\_name, employees.last\_name, departments.department\_name FROM employees INNER JOIN departments ON employees.department\_id = departments.department\_id;

Q5. Explain Primary Key and Foreign Key.

Ans: **Primary Key:** A primary key is a unique identifier for a record in a relational database table. It uniquely identifies each row/record in the table and ensures that there are no duplicate values. The primary key must contain unique values, and it cannot contain NULL (empty) values. Each table in a database can have only one primary key.

* **Characteristics of a Primary Key:**
  + Uniqueness: Each value in the primary key must be unique.
  + Non-null: The primary key cannot contain NULL values.
  + Fixed: It doesn't change over time or across records.

CREATE TABLE employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

salary DECIMAL(10, 2)

);

**Foreign Key:** A foreign key is a column or a set of columns in a relational database table that refers to the primary key of another table. It establishes a link or a relationship between two tables, enforcing referential integrity. The values in the foreign key column(s) must match the values in the primary key column of the referenced table.

* **Characteristics of a Foreign Key:**
  + It references the primary key of another table.
  + It ensures referential integrity.
  + Values in the foreign key must match values in the referenced primary key.

CREATE TABLE departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(50)

);

CREATE TABLE employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

salary DECIMAL(10, 2),

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES departments(department\_id)

);

Q6. Give the order of execution of SQL clauses in an SQL query.

Ans: The order of execution of SQL clauses in an SQL query generally follows a specific sequence. The typical order is as follows:

1. **SELECT:** This clause specifies the columns that you want to retrieve from the database.
2. **FROM:** The FROM clause indicates the table or tables from which the data will be retrieved.
3. **WHERE:** The WHERE clause is used to filter the rows based on a specified condition. It acts as a kind of "if" statement for the query.
4. **GROUP BY:** If grouping is required, the GROUP BY clause is used to group rows that have the same values in specified columns into summary rows.
5. **HAVING:** The HAVING clause is used to filter the results of the GROUP BY clause based on specified conditions.
6. **ORDER BY:** The ORDER BY clause is used to sort the result set based on one or more columns. It can be applied to both numerical and character columns.
7. **LIMIT/OFFSET or FETCH:** The LIMIT or OFFSET (for MySQL and PostgreSQL) or FETCH (for SQL Server and Oracle) clause is used to limit the number of rows returned or to skip a specific number of rows.

SELECT column1, column2

FROM table\_name

WHERE condition

GROUP BY column1

HAVING condition

ORDER BY column1 ASC

LIMIT 10;