**Basic DBMS Questions**

1. **What is a Database?**
   * **Answer**: A database is an organized collection of structured information, or data, typically stored electronically in a computer system. Databases are managed by Database Management Systems (DBMS) that allow users to store, retrieve, and manipulate the data efficiently.
   * **Example**: A library's catalog system that contains records of all the books, authors, and borrowers is an example of a database.
2. **What is DBMS?**
   * **Answer**: A Database Management System (DBMS) is software that interacts with end-users, applications, and the database itself to capture and analyse data. It allows users to create, read, update, and delete data in a database.
   * **Example**: MySQL, PostgreSQL, Oracle, and Microsoft SQL Server are popular DBMS.
3. **What is SQL?**
   * **Answer**: SQL (Structured Query Language) is a standard programming language used to communicate with and manipulate databases. SQL is used to perform tasks such as retrieving data, updating records, inserting new data, and deleting records.
   * **Example**: SELECT \* FROM employees retrieve all records from the "employees" table.
4. **What are the types of SQL commands?**
   * **Answer**: SQL commands are classified into the following types:
     + **DDL (Data Definition Language)**: Commands used to define or alter the structure of the database (e.g., CREATE, ALTER, DROP).
     + **DML (Data Manipulation Language)**: Commands used for manipulating data stored in the database (e.g., SELECT, INSERT, UPDATE, DELETE).
     + **DCL (Data Control Language)**: Commands used to control access to data in the database (e.g., GRANT, REVOKE).
     + **TCL (Transaction Control Language)**: Commands used to manage transactions in the database (e.g., COMMIT, ROLLBACK, SAVEPOINT).
5. **What is a Primary Key?**
   * **Answer**: A Primary Key is a field (or combination of fields) in a table that uniquely identifies each record in that table. It must contain unique values and cannot contain NULL values.
   * **Example**: In a table “student”, the "student\_id" can be a primary key as it uniquely identifies each student.
6. **What is a Foreign Key?**
   * **Answer**: A Foreign Key is a field (or a collection of fields) in one table that uniquely identifies a row of another table. It establishes a relationship between the two tables.
   * **Example**: In a "orders" table, a "customer\_id" can be a foreign key that references "id" in a "customers" table.
7. **What is Normalization?**
   * **Answer**: Normalization is the process of organizing the fields and tables of a relational database to minimize redundancy and dependency. It involves dividing large tables into smaller tables and defining relationships among them.
   * **Example**: Normalization reduces data redundancy by dividing customer details and order details into separate tables instead of keeping all data in a single table.
8. **What are the different Normal Forms?**
   * **Answer**: The most commonly used normal forms are:
     + **1NF (First Normal Form)**: Ensures that each column contains atomic (indivisible) values and each column contains values of a single type.
     + **2NF (Second Normal Form)**: Based on 1NF, ensures that each non-primary key attribute is fully functionally dependent on the primary key.
     + **3NF (Third Normal Form)**: Based on 2NF, ensures that all the attributes are functionally dependent only on the primary key.
   * **Example**: A table in 1NF would have each field as atomic, such as a "Name" field being split into "First Name" and "Last Name".
9. **What is a JOIN in SQL?**
   * **Answer**: A JOIN clause is used to combine rows from two or more tables, based on a related column between them.
   * **Types of JOINS**:
     + **INNER JOIN**: Returns records that have matching values in both tables.
     + **LEFT JOIN (LEFT OUTER JOIN)**: Returns all records from the left table, and the matched records from the right table. If there is no match, NULL values are returned for columns from the right table.
     + **RIGHT JOIN (RIGHT OUTER JOIN)**: Returns all records from the right table, and the matched records from the left table. If there is no match, NULL values are returned for columns from the left table.
     + **FULL JOIN (FULL OUTER JOIN)**: Returns all records when there is a match in either left or right table records.
   * **Example**:

SQL

Copy code

SELECT employees.name, departments.department\_name

FROM employees

INNER JOIN departments

ON employees.department\_id = departments.id;

1. **What is a View in SQL?**
   * **Answer**: A View is a virtual table that is based on the result set of an SQL query. It does not store data physically but provides a way to simplify complex queries by saving them as a single table that can be queried just like a regular table.
   * **Example**:

sql

Copy code

CREATE VIEW employee\_view AS

SELECT name, salary

FROM employees

WHERE department = 'Sales';

**Intermediate SQL Questions**

1. **How do you use the GROUP BY clause?**
   * **Answer**: The GROUP BY clause is used in with aggregate functions (like COUNT, MAX, MIN, SUM, AVG) to group the result set by one or more columns.
   * **Example**:

SQL

Copy code

SELECT department, COUNT (\*) AS num\_employees

FROM employees

GROUP BY department;

This query counts the number of employees in each department.

1. **What is the difference between WHERE and HAVING?**
   * **Answer**: The WHERE clause is used to filter records before any groupings are made. The HAVING clause is used to filter records after grouping has been done.
   * **Example**:

SQL

Copy code

SELECT department, COUNT (\*) AS num\_employees

FROM employees

WHERE salary > 50000

GROUP BY department

HAVING COUNT (\*) > 5;

This query first filters employees with a salary greater than 50,000 and then groups by department and filters departments with more than 5 employees.

1. **What are Subqueries and Correlated Subqueries?**
   * **Answer**:
     + **Subquery**: A query nested inside another query. It is executed once before the main query.
     + **Correlated Subquery**: A subquery that references columns from the outer query. It is executed once for every row processed by the outer query.
   * **Example of Subquery**:

sql

Copy code

SELECT name

FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);

* + - **Example of Correlated Subquery**:

sql

Copy code

SELECT e1.name, e1.salary

FROM employees e1

WHERE e1.salary > (SELECT AVG(e2.salary) FROM employees e2 WHERE e1.department = e2.department);

1. **What are Indexes in SQL?**
   * **Answer**: Indexes are used to speed up the retrieval of data from a table by creating pointers to the data. Indexes are created on columns that are frequently used in WHERE clauses, join conditions, or in sorting.
   * **Example**:

sql

Copy code

CREATE INDEX idx\_employee\_name ON employees(name);

This creates an index on the "name" column of the "employees" table to speed up searches on employee names.

1. **What is a Transaction in SQL, and what are its properties (ACID)?**
   * **Answer**: A transaction is a sequence of one or more SQL statements that are executed as a single unit of work. The properties of transactions are:
     + **Atomicity**: Ensures that all operations within a transaction are completed; if not, the transaction is aborted.
     + **Consistency**: Ensures that the database remains consistent before and after the transaction.
     + **Isolation**: Ensures that the execution of a transaction is isolated from other transactions.
     + **Durability**: Ensures that the changes made by a transaction are permanent, even in the case of a system failure.
   * **Example**:

sql

Copy code

BEGIN;

UPDATE accounts SET balance = balance - 100 WHERE account\_id = 1;

UPDATE accounts SET balance = balance + 100 WHERE account\_id = 2;

COMMIT;

1. **What is the difference between DELETE, TRUNCATE, and DROP?**
   * **Answer**:
     + **DELETE**: Removes rows from a table based on a condition. Can be rolled back (part of DML).
     + **TRUNCATE**: Removes all rows from a table without logging individual row deletions. Cannot be rolled back (part of DDL).
     + **DROP**: Deletes the table structure and all data within it. Cannot be rolled back (part of DDL).
   * **Example**:

sql

Copy code

DELETE FROM employees WHERE department = 'Sales'; -- Deletes specific rows

TRUNCATE TABLE employees; -- Deletes all rows in the table

DROP TABLE employees; -- Deletes the table itself

**Important SQL Queries for Practice**

1. **Find the second highest salary from an "employees" table**:

sql

Copy code

SELECT MAX(salary) AS second\_highest\_salary

FROM employees

WHERE salary < (SELECT MAX(salary) FROM employees);

1. **Get the count of employees in each department**:

sql

Copy code

SELECT department, COUNT(\*) AS num\_employees

FROM employees

GROUP BY department;

1. **Find employees whose names start with 'A'**:

sql

Copy code

SELECT \* FROM employees

WHERE name LIKE 'A%';

1. **Display the employee details who joined in the year 2020**:

sql

Copy code

SELECT \* FROM employees

WHERE YEAR(joining\_date) = 2020;

1. **Get the highest paid employee in each department**:

sql

Copy code

SELECT department, MAX(salary) AS max\_salary

FROM employees

GROUP BY department;

1. **List all employees who do not have any manager**:

sql

Copy code

SELECT \* FROM employees

WHERE manager\_id IS NULL;

1. **Write a query to fetch all employees who are also managers**:

sql

Copy code

SELECT DISTINCT e1.name

FROM employees e1

JOIN employees e2 ON e1.id = e2.manager\_id;

1. **Find the department with the highest number of employees**:

sql

Copy code

SELECT department

FROM employees

GROUP BY department

ORDER BY COUNT (\*) DESC

LIMIT 1;

These questions and examples cover basic to intermediate concepts in DBMS and SQL, preparing you well for an entry-level technical interview. Make sure to practice writing these queries and understand the underlying concepts to answer any variations you might encounter in an interview.