https://www.simplilearn.com/dbms-interview-questions-and-answers-article

https://www.softwaretestinghelp.com/database-interview-questions/

Basic SQL Questions

1. **Question:** What is SQL?
   - **Answer:** SQL (Structured Query Language) is a standardized language used to interact with relational databases. It allows users to manage and manipulate data stored in tables.
2. **Question:** What are the main types of SQL commands?
   - **Answer:** SQL commands include:
     - **DDL (Data Definition Language)**: CREATE, ALTER, DROP (to define or modify structures like tables).
     - **DML (Data Manipulation Language)**: INSERT, UPDATE, DELETE (to manipulate data in tables).
     - **DQL (Data Query Language)**: SELECT (to retrieve data).
     - **DCL (Data Control Language)**: GRANT, REVOKE (to control access permissions).
3. **Question:** Explain the difference between SQL and NoSQL databases.
   - **Answer:** SQL databases (e.g., MySQL, PostgreSQL) use structured tables with predefined schemas and SQL for querying. NoSQL databases (e.g., MongoDB, Redis) use flexible document, key-value, or graph-based models with no fixed schema.
4. **Question:** What is a primary key?
   - **Answer:** A primary key is a unique identifier for each record in a table. It ensures each row is uniquely identified and prevents duplicate and null values.
5. **Question:** How do you retrieve all columns from a table named `users`?
   - **Answer:** `SELECT * FROM users;`
6. **Question:** How do you filter records in SQL?
   - **Answer:** Use the `WHERE` clause. Example: `SELECT * FROM users WHERE age > 18;`
7. **Question:** What is the purpose of the `GROUP BY` clause?
   - **Answer:** The `GROUP BY` clause is used with aggregate functions (like COUNT, SUM) to group rows that have the same values into summary rows.
8. **Question:** How do you count the number of records in a table named `orders`?
   - **Answer:** `SELECT COUNT(*) FROM orders;`
9. **Question:** Explain the difference between `DELETE` and `TRUNCATE` commands.
   - **Answer:** `DELETE` is used to remove specific rows from a table based on a condition, while `TRUNCATE` removes all rows from a table and resets identity columns.
10. **Question:** What does `NULL` represent in SQL?
    - **Answer:** `NULL` represents a missing or undefined value in a table's column.

Intermediate SQL Questions

11. **Question:** How do you join two tables `orders` and `customers` using a common column `customer_id`?
    - **Answer:**

      ```
      sql
      ```

      Copy code

      `SELECT * FROM orders INNER JOIN customers ON orders.customer_id = customers.customer_id;`
12. **Question:** What is a foreign key?
    - **Answer:** A foreign key is a column or a set of columns in a table that uniquely identifies a record in another table. It establishes a link between the two tables.
13. **Question:** Explain the `HAVING` clause.
    - **Answer:** The `HAVING` clause is used in conjunction with `GROUP BY` to filter records returned by aggregate functions based on a condition.
14. **Question:** How do you create an index on a column named `email` in a table named `users`?
    - **Answer:** `CREATE INDEX idx_email ON users(email);`
15. **Question:** What is database normalization?
    - **Answer:** Database normalization is the process of organizing data in tables to reduce redundancy and dependency by dividing large tables into smaller tables and defining relationships between them.
16. **Question:** How do you handle transactions in SQL?
    - **Answer:** Use `BEGIN TRANSACTION` to start a transaction, `COMMIT` to save changes permanently, and `ROLLBACK` to undo changes if an error occurs.
17. **Question:** Explain the `UNION` operator.
    - **Answer:** The `UNION` operator is used to combine the result sets of two or more `SELECT` statements, removing duplicates.
18. **Question:** What are stored procedures in SQL?
    - **Answer:** Stored procedures are precompiled SQL statements stored in a database and executed by calling the procedure's name. They improve performance and maintainability.
19. **Question:** How do you grant SELECT permission on a table `employees` to a user named `user1`?
    - **Answer:** `GRANT SELECT ON employees TO user1;`
20. **Question:** What are triggers in SQL?
    - **Answer:** Triggers are special types of stored procedures that automatically execute when a certain event (e.g., `INSERT`, `UPDATE`, `DELETE`) occurs on a table.

Advanced SQL Questions

21. **Question:** How do you optimize a slow-running SQL query?

- **Answer:** Optimize by using indexes, avoiding `SELECT *`, optimizing joins, reducing data retrieval, and using appropriate database configuration settings.

22. **Question:** Explain the ACID properties in the context of transactions.
    - **Answer:** ACID stands for Atomicity, Consistency, Isolation, and Durability. These properties ensure that database transactions are reliable and maintain data integrity.
23. **Question:** What is a recursive SQL query?
    - **Answer:** A recursive SQL query is a query that refers to a subquery that repeatedly references the main query until the desired result is achieved, often used in hierarchical data structures.
24. **Question:** How do you handle pagination in SQL?
    - **Answer:** Use `LIMIT` and `OFFSET` clauses to retrieve a subset of rows, enabling efficient fetching of large datasets page by page.
25. **Question:** What are common types of SQL joins and when do you use each?
    - **Answer:** Common joins include `INNER JOIN` (returns rows when there is a match in both tables), `LEFT JOIN` (returns all rows from the left table and matching rows from the right table), `RIGHT JOIN` (returns all rows from the right table and matching rows from the left table), and `FULL OUTER JOIN` (returns all rows when there is a match in either table).
26. **Question:** How do you implement row-level security in SQL databases?
    - **Answer:** Use views, stored procedures, or security policies to restrict access to rows based on user roles or attributes.
27. **Question:** Explain the concept of materialized views.
    - **Answer:** Materialized views are precomputed views stored on disk as tables. They are updated periodically based on the underlying data changes and improve query performance for complex queries.
28. **Question:** How do you handle database schema migrations?
    - **Answer:** Use migration tools like `ALTER TABLE` statements, database migration frameworks (e.g., Flyway, Liquibase), and version control systems to manage and apply changes to database schemas.
29. **Question:** What are common techniques for optimizing SQL queries?
    - **Answer:** Techniques include using indexes, optimizing SQL query structure (e.g., avoiding correlated subqueries), using appropriate data types, caching query results, and tuning database configuration parameters.
30. **Question:** How do you manage concurrency issues in SQL databases?
    - **Answer:** Use techniques like row-level locking, optimistic concurrency control (e.g., using timestamps), and transaction isolation levels (e.g., `READ COMMITTED`, `REPEATABLE READ`) to manage and prevent concurrency issues such as race conditions and deadlocks.

Basic SQL Practice Questions

1. **Question:** Retrieve all columns from the `employees` table.
    - **Answer:**

    ```
    SELECT * FROM employees;
    ```

2. **Question:** Retrieve distinct values from the `department` column in the `employees` table.
   - **Answer:**
     ```
     SELECT DISTINCT department FROM employees;
     ```
3. **Question:** Insert a new record into the `employees` table with values for `name`, `age`, and `department`.
   - **Answer:**
     ```
     INSERT INTO employees (name, age, department) VALUES ('John Doe', 30, 'IT');
     ```
4. **Question:** Update the `salary` of all employees in the `IT` department by 10%.
   - **Answer:**
     ```
     UPDATE employees SET salary = salary * 1.1 WHERE department = 'IT';
     ```
5. **Question:** Delete all records from the `inactive_users` table.
   - **Answer:**
     ```
     DELETE FROM inactive_users;
     ```
6. **Question:** Retrieve the highest salary from the `employees` table.
   - **Answer:**
     ```
     SELECT MAX(salary) FROM employees;
     ```
7. **Question:** Calculate the average age of employees in the `HR` department.
   - **Answer:**
     ```
     SELECT AVG(age) FROM employees WHERE department = 'HR';
     ```
8. **Question:** Retrieve the number of employees in each department.
   - **Answer:**
     ```
     SELECT department, COUNT(*) AS num_employees FROM employees GROUP BY department;
     ```
9. **Question:** List the names and ages of employees whose age is greater than 40.
   - **Answer:**
     ```
     SELECT name, age FROM employees WHERE age > 40;
     ```
10. **Question:** Retrieve the top 5 highest-paid employees.
    - **Answer:**
      ```
      SELECT name, salary FROM employees ORDER BY salary DESC LIMIT 5;
      ```

Intermediate SQL Practice Questions

11. **Question:** Join the `employees` and `departments` tables to retrieve employees along with their department names.
    - **Answer:**
      ```
      SELECT e.name, d.department_name FROM employees e INNER JOIN departments d ON
      e.department_id = d.department_id;
      ```

12. **Question:** Calculate the total salary expense for each department.
    - **Answer:**

    ```
    SELECT department, SUM(salary) AS total_salary_expense FROM employees GROUP BY department;
    ```

13. **Question:** Retrieve employees who have joined in the last year (based on `join_date`).
    - **Answer:**

    ```
    SELECT name, join_date FROM employees WHERE join_date >= DATE_SUB(CURRENT_DATE(),
    INTERVAL 1 YEAR);
    ```

14. **Question:** Update the `manager_id` of employees who manage more than 10 employees to `1`.
    - **Answer:**

    ```
    UPDATE employees SET manager_id = 1 WHERE employee_id IN ( SELECT manager_id FROM
    employees GROUP BY manager_id HAVING COUNT(*) > 10 );
    ```

15. **Question:** Retrieve employees who have the same salary as their manager.
    - **Answer:**

    ```
    SELECT e.name AS employee_name, e.salary, m.name AS manager_name, m.salary AS
    manager_salary FROM employees e INNER JOIN employees m ON e.manager_id = m.employee_id
    WHERE e.salary = m.salary;
    ```

16. **Question:** List departments with more than 5 employees and their average salary.
    - **Answer:**

    ```
    SELECT department, COUNT(*) AS num_employees, AVG(salary) AS avg_salary FROM employees
    GROUP BY department HAVING COUNT(*) > 5;
    ```

17. **Question:** Retrieve employees who have not been assigned to any department.
    - **Answer:**

    ```
    SELECT name FROM employees WHERE department_id IS NULL;
    ```

18. **Question:** Calculate the total number of employees in the company.
    - **Answer:**

    ```
    SELECT COUNT(*) AS total_employees FROM employees;
    ```

19. **Question:** Find the employee(s) with the highest salary.
    - **Answer:**

    ```
    SELECT name, salary FROM employees WHERE salary = (SELECT MAX(salary) FROM employees);
    ```

20. **Question:** Retrieve employees who have a salary within the top 10% of the company.
    - **Answer:**

    ```
    SELECT name, salary FROM employees WHERE salary >= (SELECT PERCENTILE_CONT(0.9) WITHIN
    GROUP (ORDER BY salary) FROM employees);
    ```

Advanced SQL Practice Questions

21. **Question:** Calculate the 3-month moving average salary for each employee.

    - **Answer:**

    ```
    SELECT employee_id, salary, AVG(salary) OVER ( ORDER BY join_date ROWS BETWEEN 2
    PRECEDING AND CURRENT ROW ) AS moving_avg_salary FROM employees;
    ```

22. **Question:** List employees who have changed departments more than once.

    - **Answer:**

    ```
    SELECT employee_id, name, COUNT(DISTINCT department_id) AS num_departments FROM
    employee_history GROUP BY employee_id, name HAVING COUNT(DISTINCT department_id) > 1;
    ```

23. **Question:** Identify employees who are potential candidates for promotion (based on experience and performance).

    - **Answer:**

    ```
    SELECT name, experience_years, performance_rating FROM employees WHERE experience_years >
    5 AND performance_rating > 8;
    ```

24. **Question:** Calculate the cumulative sum of salaries for each department over time.

    - **Answer:**

    ```
    SELECT department, join_date, SUM(salary) OVER (PARTITION BY department ORDER BY
    join_date) AS cumulative_salary FROM employees;
    ```

25. **Question:** Retrieve employees who have never been absent (no records in the `absence` table).

    - **Answer:**

    ```
    SELECT e.name FROM employees e LEFT JOIN absence a ON e.employee_id = a.employee_id WHERE
    a.employee_id IS NULL;
    ```

26. **Question:** Calculate the median salary for each department.

    - **Answer:**

    ```
    SELECT department, PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY salary) AS median_salary
    FROM employees GROUP BY department;
    ```

27. **Question:** List employees who have a higher salary than their peers in the same department.

    - **Answer:**

    ```
    SELECT e1.name, e1.salary, e2.name AS peer_name, e2.salary AS peer_salary FROM employees
    e1 INNER JOIN employees e2 ON e1.department_id = e2.department_id WHERE e1.salary >
    e2.salary AND e1.employee_id != e2.employee_id;
    ```

28. **Question:** Implement a ranking system to rank employees based on their performance rating.

    - **Answer:**

    ```
    SELECT name, performance_rating, DENSE_RANK() OVER (ORDER BY performance_rating DESC) AS
    performance_rank FROM employees;
    ```

29. **Question:** Find employees who have stayed in the company for more than 10 years.

    - **Answer:**

```
SELECT name, join_date, DATEDIFF(CURRENT_DATE(), join_date) AS years_of_service FROM
employees WHERE DATEDIFF(CURRENT_DATE(), join_date) > 3650; -- 3650 days = 10 years
```

30. **Question:** Calculate the ratio of the highest to the lowest salary in each department.

    - **Answer:**

      ```
      SELECT department, MAX(salary) / MIN(salary) AS salary_ratio FROM employees GROUP BY
      department;
      ```