for 2 yr experieced full stack developer give me the list of queries which a interviewer expects from me that i can solve? give me list of queries and also their solutions

For a **2-year experienced full-stack developer**, interviewers typically expect proficiency in **SQL queries**, covering **basic CRUD operations, joins, aggregations, window functions, subqueries, and performance optimizations**.

Here's a **list of SQL queries** with **solutions** that you should prepare:

---

1. Basic CRUD Queries

1.1 Retrieve all records from a table

sql Copy Edit

```sql
SELECT * FROM Employees;
```

1.2 Insert a new record

sql Copy Edit

```sql
INSERT INTO Employees (id, name, department, salary)
VALUES (101, 'John Doe', 'IT', 60000);
```

1.3 Update a record

sql Copy Edit

```sql
UPDATE Employees
SET salary = 65000
WHERE id = 101;
```

1.4 Delete a record

sql Copy Edit

```sql
DELETE FROM Employees WHERE id = 101;
```

---

2. Filtering Data (WHERE, LIKE, IN, BETWEEN)

2.1 Get employees from IT or HR department

sql Copy Edit

```sql
SELECT * FROM Employees
WHERE department IN ('IT', 'HR');
```

2.2 Find employees whose name starts with 'J'

sql Copy Edit

```sql
SELECT * FROM Employees
WHERE name LIKE 'J%';
```

2.3 Get employees with salary between 50K and 80K

sql Copy Edit

```sql
SELECT * FROM Employees
WHERE salary BETWEEN 50000 AND 80000;
```

---

3. Sorting and Aggregation

3.1 Get the highest salary

```sql
SELECT MAX(salary) AS HighestSalary FROM Employees;
```

3.2 Get the total salary paid per department

```sql
SELECT department, SUM(salary) AS TotalSalary
FROM Employees
GROUP BY department;
```

3.3 Get the average salary of employees

```sql
SELECT AVG(salary) AS AvgSalary FROM Employees;
```

3.4 Count employees per department

```sql
SELECT department, COUNT(*) AS EmployeeCount
FROM Employees
GROUP BY department;
```

---

4. Joins

4.1 Get employee details along with their department details

```sql
SELECT e.id, e.name, e.salary, d.department_name
FROM Employees e
JOIN Departments d ON e.department_id = d.id;
```

4.2 Get employees who have no department assigned (LEFT JOIN)

```sql
SELECT e.id, e.name, e.salary, d.department_name
FROM Employees e
LEFT JOIN Departments d ON e.department_id = d.id
WHERE d.department_name IS NULL;
```

---

5. Subqueries

5.1 Get employees with salary greater than the average salary

```sql
SELECT * FROM Employees
WHERE salary > (SELECT AVG(salary) FROM Employees);
```

5.2 Get employees who belong to the 'IT' department using a subquery

```sql
SELECT * FROM Employees
WHERE department_id = (SELECT id FROM Departments WHERE department_name = 'IT');
```

---

6. Window Functions

6.1 Rank employees by salary within each department

```sql
```

```sql
SELECT id, name, department, salary,
       RANK() OVER (PARTITION BY department ORDER BY salary DESC) AS SalaryRank
FROM Employees;
```

6.2 Get the running total of salaries

sql Copy Edit

```sql
SELECT id, name, salary,
       SUM(salary) OVER (ORDER BY id) AS RunningTotal
FROM Employees;
```

---

7. Performance Optimization Queries

7.1 Use Index for faster search

sql Copy Edit

```sql
CREATE INDEX idx_employee_salary ON Employees(salary);
```

7.2 Find duplicate records in a table

sql Copy Edit

```sql
SELECT name, COUNT(*)
FROM Employees
GROUP BY name
HAVING COUNT(*) > 1;
```

7.3 Delete duplicate records but keep one

sql Copy Edit

```sql
DELETE FROM Employees
WHERE id NOT IN (
    SELECT MIN(id)
    FROM Employees
    GROUP BY name
);
```

---

8. Advanced Queries

8.1 Find the second-highest salary

sql Copy Edit

```sql
SELECT DISTINCT salary
FROM Employees
ORDER BY salary DESC
LIMIT 1 OFFSET 1;
```

8.2 Find employees who joined in the last 30 days

sql Copy Edit

```sql
SELECT * FROM Employees
WHERE join_date >= NOW() - INTERVAL 30 DAY;
```

8.3 Retrieve employees with same salary as someone else

sql Copy Edit

```sql
SELECT e1.*
FROM Employees e1
JOIN Employees e2 ON e1.salary = e2.salary AND e1.id <> e2.id;
```

Conclusion

These are some commonly asked **SQL queries** that a **2-year experienced full-stack developer** should be

These are some commonly asked **SQL queries** that a **2-year experienced full-stack developer** should be able to solve in an interview. You should also practice **query optimization, indexing, and real-world scenario-based queries**.

Would you like additional **practice questions** or **explanations** for any of these queries? 🚀

Compare with Claude

Ask anything

Search Reason