1. **Retrieve All Records from a Table**

```
sql
Copy code
SELECT * FROM table_name;
```

2. **Select Specific Columns**

```
sql
Copy code
SELECT column1, column2 FROM table_name;
```

3. **Filter Records Using WHERE**

```
sql
Copy code
SELECT * FROM table_name WHERE condition;
```

4. **Order Records by a Column**

```
sql
Copy code
SELECT * FROM table_name ORDER BY column_name [ASC|DESC];
```

5. **Find Unique Values Using DISTINCT**

```
sql
Copy code
SELECT DISTINCT column_name FROM table_name;
```

6. **Count the Number of Records**

```
sql
Copy code
SELECT COUNT(*) FROM table_name;
```

7. **Using GROUP BY with Aggregate Functions**
   - **Count the number of employees in each department:**

```
sql
Copy code
SELECT department, COUNT(*) FROM employees GROUP BY department;
```

8. **Aggregate Functions (SUM, AVG, MIN, MAX)**

```
sql
Copy code
SELECT SUM(column_name), AVG(column_name), MIN(column_name),
MAX(column_name) FROM table_name;
```

Intermediate Queries

9. **JOINs (Inner, Left, Right, Full)**
   - **Inner Join** (fetch matching records from both tables):

```
sql
Copy code
SELECT column1, column2 FROM table1 INNER JOIN table2 ON
table1.common_column = table2.common_column;
```

- **Left Join** (fetch all records from the left table and matching records from the right table):

```
sql
Copy code
SELECT column1, column2 FROM table1 LEFT JOIN table2 ON
table1.common_column = table2.common_column;
```

- **Right Join** (fetch all records from the right table and matching records from the left table):

```
sql
Copy code
SELECT column1, column2 FROM table1 RIGHT JOIN table2 ON
table1.common_column = table2.common_column;
```

- **Full Outer Join** (fetch all records when there is a match in either table):

```
sql
Copy code
SELECT column1, column2 FROM table1 FULL OUTER JOIN table2 ON
table1.common_column = table2.common_column;
```

10. **Find the Second Highest Salary**

```
sql
Copy code
SELECT MAX(salary) FROM employees WHERE salary < (SELECT MAX(salary)
FROM employees);
```

11. **Retrieve Employees with the Highest Salary in Each Department**

```
sql
Copy code
SELECT department, MAX(salary) FROM employees GROUP BY department;
```

12. **Subqueries**
    - **Using a subquery to filter results:**

```
sql
Copy code
SELECT * FROM employees WHERE salary > (SELECT AVG(salary) FROM
employees);
```

13. **Using HAVING with GROUP BY**
    - **Find departments with more than 5 employees:**

```
sql
Copy code
```

```
SELECT department, COUNT(*) FROM employees GROUP BY department
HAVING COUNT(*) > 5;
```

Advanced Queries

14. **Self Join**
    - **Find employees who report to the same manager:**

```
sql
Copy code
SELECT a.employee_name, b.employee_name AS manager_name
FROM employees a
INNER JOIN employees b ON a.manager_id = b.employee_id;
```

15. **Correlated Subquery**
    - **Find employees with a salary greater than the average salary of their department:**

```
sql
Copy code
SELECT employee_name, salary
FROM employees e
WHERE salary > (SELECT AVG(salary) FROM employees WHERE
department_id = e.department_id);
```

16. **Find Nth Highest Salary**
    - **Find the 3rd highest salary:**

```
sql
Copy code
SELECT salary FROM (
  SELECT salary, DENSE_RANK() OVER (ORDER BY salary DESC) AS
salary_rank
  FROM employees
) AS ranked_salaries
WHERE salary_rank = 3;
```

17. **Common Table Expressions (CTE)**
    - **With Recursive Query:**

```
sql
Copy code
WITH RECURSIVE cte AS (
  SELECT employee_id, employee_name, manager_id
  FROM employees
  WHERE manager_id IS NULL
  UNION ALL
  SELECT e.employee_id, e.employee_name, e.manager_id
  FROM employees e
  INNER JOIN cte ON e.manager_id = cte.employee_id
)
SELECT * FROM cte;
```

18. **Window Functions (ROW_NUMBER, RANK, DENSE_RANK)**
    - **Find the rank of employees by salary:**

```
sql
Copy code
SELECT employee_name, salary,
RANK() OVER (ORDER BY salary DESC) AS salary_rank
FROM employees;
```

19. **Case Statement**
    - **Assign a grade based on salary:**

```
sql
Copy code
SELECT employee_name, salary,
CASE
  WHEN salary > 100000 THEN 'A'
  WHEN salary BETWEEN 50000 AND 100000 THEN 'B'
  ELSE 'C'
END AS grade
FROM employees;
```

20. **Union and Union All**
    - **Combine results from two queries (removing duplicates with `UNION`):**

```
sql
Copy code
SELECT column1 FROM table1
UNION
SELECT column1 FROM table2;
```

    - **Use `UNION ALL` to retain duplicates:**

```
sql
Copy code
SELECT column1 FROM table1
UNION ALL
SELECT column1 FROM table2;
```

21. **Find Duplicate Records in a Table**

```
sql
Copy code
SELECT column_name, COUNT(*)
FROM table_name
GROUP BY column_name
HAVING COUNT(*) > 1;
```

22. **Delete Duplicate Records from a Table**

```
sql
Copy code
DELETE FROM table_name
WHERE id NOT IN (
  SELECT MIN(id)
  FROM table_name
  GROUP BY column_name
);
```

Miscellaneous

23. **Find Employees Who Have the Same Salary as Another Employee**

```
sql
Copy code
SELECT employee_name, salary
FROM employees e1
WHERE EXISTS (
  SELECT 1 FROM employees e2
  WHERE e1.salary = e2.salary AND e1.employee_id != e2.employee_id
);
```

24. **Update a Record Based on Condition**

```
sql
Copy code
UPDATE employees
SET salary = salary + 5000
WHERE employee_id = 101;
```

25. **Delete Records Based on Condition**

```
sql
Copy code
DELETE FROM employees WHERE employee_id = 101;
```

Scenario Based

1. Find Employees Who Earn More Than the Average Salary

Scenario: Retrieve the names of employees whose salary is greater than the average salary in the company.

```
sql
Copy code
SELECT employee_name, salary
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);
```

2. Retrieve the Second Highest Salary

Scenario: Find the second highest salary from the employees' table.

```
sql
Copy code
SELECT MAX(salary)
FROM employees
WHERE salary < (SELECT MAX(salary) FROM employees);
```

3. Find Employees Who Joined in the Last 6 Months

Scenario: Write a query to retrieve all employees who joined within the last six months.

```
sql
Copy code
SELECT employee_name, join_date
FROM employees
WHERE join_date >= DATEADD(MONTH, -6, GETDATE());
```

4. Find Duplicate Records in a Table

Scenario: You are tasked with finding duplicate records in a table based on a specific column, say `email`.

```
sql
Copy code
SELECT email, COUNT(*)
FROM employees
GROUP BY email
HAVING COUNT(*) > 1;
```

5. Delete Duplicate Records While Keeping One Instance

Scenario: Delete duplicate records from the table, but keep one instance of each duplicate record based on the `email` field.

```
sql
Copy code
DELETE FROM employees
WHERE employee_id NOT IN (
    SELECT MIN(employee_id)
    FROM employees
    GROUP BY email
);
```

6. Retrieve Employees Who Have the Same Salary

Scenario: Find all employees who have the same salary as at least one other employee.

```
sql
Copy code
SELECT employee_name, salary
FROM employees e1
WHERE EXISTS (
    SELECT 1 FROM employees e2
    WHERE e1.salary = e2.salary AND e1.employee_id != e2.employee_id
);
```

7. Find Top N Salaries in a Company

Scenario: Retrieve the top 3 salaries from the employees' table.

```
sql
Copy code
SELECT DISTINCT salary
```

```
FROM employees
ORDER BY salary DESC
LIMIT 3;
```

8. Find the Department with the Maximum Number of Employees

Scenario: Write a query to find the department that has the highest number of employees.

```
sql
Copy code
SELECT department_id, COUNT(*) AS employee_count
FROM employees
GROUP BY department_id
ORDER BY employee_count DESC
LIMIT 1;
```

9. Find Employees Who Report to the Same Manager

Scenario: Write a query to find employees who have the same manager.

```
sql
Copy code
SELECT e1.employee_name AS Employee, e2.employee_name AS Manager
FROM employees e1
JOIN employees e2 ON e1.manager_id = e2.employee_id
ORDER BY e2.employee_name;
```

10. Find Employees with No Assigned Manager

Scenario: List all employees who do not have a manager.

```
sql
Copy code
SELECT employee_name
FROM employees
WHERE manager_id IS NULL;
```

11. Find the Employee with the Highest Salary in Each Department

Scenario: Retrieve the employee(s) who has/have the highest salary in each department.

```
sql
Copy code
SELECT department_id, employee_name, salary
FROM employees e1
WHERE salary = (SELECT MAX(salary)
                FROM employees e2
                WHERE e1.department_id = e2.department_id);
```

12. Retrieve Employees Who Work in More Than One Department

Scenario: You are asked to find employees who are assigned to more than one department.

```
sql
```

```
Copy code
SELECT employee_id, COUNT(department_id) AS dept_count
FROM employee_departments
GROUP BY employee_id
HAVING dept_count > 1;
```

13. Find Consecutive Absences of Employees

Scenario: Identify employees who were absent for more than 3 consecutive days.

```
sql
Copy code
SELECT employee_id, absence_date
FROM (
    SELECT employee_id, absence_date,
           LAG(absence_date, 1) OVER (PARTITION BY employee_id ORDER BY
absence_date) AS prev_date
    FROM absences
) AS subquery
WHERE DATEDIFF(day, prev_date, absence_date) = 1;
```

14. Find the Percentage of Employees in Each Department

Scenario: Retrieve the percentage of employees working in each department relative to the total number of employees.

```
sql
Copy code
SELECT department_id,
       (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM employees)) AS percentage
FROM employees
GROUP BY department_id;
```

15. Retrieve All Employees and Their Managers

Scenario: Write a query to retrieve a list of employees along with the names of their managers.

```
sql
Copy code
SELECT e.employee_name AS Employee, m.employee_name AS Manager
FROM employees e
LEFT JOIN employees m ON e.manager_id = m.employee_id;
```

16. Find Employees Hired in Each Year

Scenario: Find the number of employees hired in each year.

```
sql
Copy code
SELECT YEAR(hire_date) AS year_hired, COUNT(*) AS employee_count
FROM employees
GROUP BY YEAR(hire_date)
ORDER BY year_hired;
```

17. List Employees Who Earn More Than Their Manager

Scenario: Write a query to find all employees whose salary is greater than their manager's salary.

```
sql
Copy code
SELECT e.employee_name, e.salary, m.employee_name AS manager_name, m.salary
AS manager_salary
FROM employees e
JOIN employees m ON e.manager_id = m.employee_id
WHERE e.salary > m.salary;
```

18. Display the 3rd Highest Salary in Each Department

Scenario: Retrieve the third highest salary in each department.

```
sql
Copy code
SELECT department_id, salary
FROM (
    SELECT department_id, salary,
           ROW_NUMBER() OVER (PARTITION BY department_id ORDER BY salary
DESC) AS salary_rank
    FROM employees
) AS ranked_salaries
WHERE salary_rank = 3;
```

19. Count the Number of Projects Each Employee Works On

Scenario: You are asked to find out how many projects each employee is assigned to.

```
sql
Copy code
SELECT employee_id, COUNT(project_id) AS project_count
FROM employee_projects
GROUP BY employee_id;
```

20. Find Departments Without Any Employees

Scenario: Identify departments that do not have any employees assigned.

```
sql
Copy code
SELECT department_name
FROM departments d
LEFT JOIN employees e ON d.department_id = e.department_id
WHERE e.employee_id IS NULL;
```