SQL Joins Solution

1.
   ```
   SELECT employee_name, role_name
   FROM employees
   JOIN roles ON employees.role_id = roles.role_id;
   ```

2.
   ```
   SELECT department_name, COUNT(*) AS employee_count
   FROM departments
   LEFT JOIN employees ON departments.department_id =
   employees.department_id
   GROUP BY department_name;
   ```

3.
   ```
   SELECT department_name, AVG(salary) AS avg_salary
   FROM departments
   LEFT JOIN employees ON departments.department_id =
   employees.department_id
   GROUP BY department_name;
   ```

4.
   ```
   SELECT employee_name
   FROM employees
   JOIN departments ON employees.department_id =
   departments.department_id
   WHERE department_name = 'Engineering';
   ```

5.
   ```
   SELECT employee_name, role_name
   FROM employees
   JOIN roles ON employees.role_id = roles.role_id
   WHERE department_id = (SELECT department_id FROM departments
   WHERE department_name = 'Sales');
   ```

6.
   ```
   SELECT department_name, AVG(salary) AS avg_salary
   FROM employees
   JOIN departments ON employees.department_id =
   departments.department_id
   GROUP BY department_name
   ORDER BY avg_salary DESC
   LIMIT 1;
   ```

7.
   ```
   SELECT role_name, COUNT(*) AS employee_count
   FROM employees
   JOIN roles ON employees.role_id = roles.role_id
   GROUP BY role_name;
   ```

8.
   ```
   SELECT employee_name, role_name
   FROM employees
   LEFT JOIN roles ON employees.role_id = roles.role_id;
   ```

9.
   ```
   SELECT role_name, COUNT(*) AS employee_count, AVG(salary) AS
   avg_salary
   FROM employees
   JOIN roles ON employees.role_id = roles.role_id
   GROUP BY role_name
   HAVING employee_count >= 2;
   ```

10.
    ```
    SELECT employee_name, role_name
    FROM employees
    JOIN roles ON employees.role_id = roles.role_id
    ORDER BY role_name, employee_name;
    ```