Here are some practical SQL practice questions covering beginner to intermediate topics, with a structure inspired by W3Schools tutorials.

Instructions:

*   For each question, write the SQL query that produces the desired result.
*   Assume you are working with a standard SQL database (like PostgreSQL, MySQL, or SQL Server).

Section 1: Basic Queries (SELECT)

Let's start with a simple table of employees.

Table: Employees

EmployeeID	FirstName	LastName	Department	Salary
1	John	Doe	HR	50000
2	Jane	Smith	IT	60000
3	Peter	Jones	Sales	55000
4	Mary	Williams	IT	62000
5	David	Brown	Sales	58000

Questions:

1. Write a query to select all columns for all employees.
2. Write a query to select only the `FirstName` and `LastName` of all employees.
3. Write a query to select the distinct (unique) departments from the `Employees` table.

Section 2: Filtering Data (WHERE)

Now, let's filter the data based on some conditions.

Questions:

4. Write a query to find all employees who work in the 'IT' department.
5. Write a query to find all employees who have a salary greater than 55000.
6. Write a query to find all employees in the 'Sales' department with a salary less than 60000.
7. Write a query to find all employees whose `FirstName` is 'Peter'.
8. Write a query to find all employees whose `LastName` starts with the letter 'S'.

Section 3: Sorting Data (ORDER BY)

Let's sort the results.

Questions:

9. Write a query to get all employees, sorted by their `Salary` in descending order.
10. Write a query to get all employees, sorted by `Department` alphabetically and

then by `Salary` in ascending order.

Section 4: Working with Tables (CREATE, INSERT, UPDATE, DELETE)

Let's create a new table and modify data.

Table: Products

ProductID	ProductName	SupplierID	CategoryID	Price
1	Chai	1	1	18.00
2	Chang	1	1	19.00
3	Aniseed Syrup	1	2	10.00

Questions:

11. Write a `CREATE TABLE` statement to create a `Customers` table with the following columns:
    * `CustomerID` (integer, primary key)
    * `CustomerName` (varchar)
    * `ContactName` (varchar)
    * `Country` (varchar)

12. Write an `INSERT INTO` statement to add the following customer to the `Customers` table:
    * `CustomerID`: 1
    * `CustomerName`: 'Alfred Futterkiste'
    * `ContactName`: 'Maria Anders'
    * `Country`: 'Germany'

13. Write an `UPDATE` statement to change the `Country` of the customer with `CustomerID` 1 to 'Mexico'.
14. Write a `DELETE` statement to remove the product with `ProductID` 3 from the `Products` table.

Section 5: Joins

Now, let's combine data from multiple tables.

Table: Orders

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

Table: Customers

CustomerID	CustomerName	Country
1	Alfred Futterkiste	Germany
2	Ana Trujillo	Mexico
3	Antonio Moreno	Mexico

Questions:

15. INNER JOIN: Write a query to list the `CustomerName` and `OrderID` for all orders.
16. LEFT JOIN: Write a query to list all `CustomerName`s and any `OrderID`s they might have. Include customers who have not placed any orders.
17. RIGHT JOIN: Write a query to list all `OrderID`s and the `CustomerName` associated with them. Include orders that may not have a corresponding customer. (This is less common in practice, but good to know).

Section 6: Advanced SQL (GROUP BY, HAVING, CASE)

Let's perform some more complex analysis.

Table: OrderDetails

OrderDetailID	OrderID	ProductID	Quantity
1	10308	1	12
2	10308	2	5
3	10309	3	10
4	10310	1	20

Questions:

18. GROUP BY: Write a query to find the total quantity of products sold for each `ProductID`.
19. HAVING: Write a query to find the `ProductID`s that have been ordered more than 15 times in total.
20. CASE: Write a query to categorize products based on their price from the `Products` table:
    *   'Expensive' if `Price` > 20
    *   'Moderate' if `Price` is between 10 and 20 (inclusive)
    *   'Cheap' if `Price` < 10

Section 7: Practical Scenarios

Let's apply what we've learned to some more practical problems.

Scenario: You are a data analyst for an e-commerce company. You have the `Employees`, `Customers`, `Orders`, `OrderDetails`, and `Products` tables.

Questions:

21. Find the name of the employee who sold the most products (in terms of quantity). You'll need to join multiple tables for this.
22. Find the top 3 customers who have spent the most money.
23. For each department, find the average salary.
24. Find the customers who have ordered the product 'Chai'.
25. MERGE (or equivalent): Imagine you have a new table `NewProducts` with the same structure as `Products`. Write a query to update the `Products` table with the data from `NewProducts`. If a product with the same `ProductID` exists, update its `Price`. If it doesn't exist, insert it as a new product. (The exact syntax for `MERGE` can vary between SQL dialects, so you can write a conceptual query or use `INSERT ... ON CONFLICT` for PostgreSQL or `MERGE` for SQL Server/Oracle).

This set of questions should give you a solid foundation in SQL. Good luck!