Here are some practice questions that focus on **joins** and **unions**, but with **a single table**. These exercises will help you practice how to use SELF JOIN (a type of join on the same table) and how to combine results from multiple queries using UNION while staying within one table.

**Question 1: Using SELF JOIN to Find Manager-Employee Pairs**

**Scenario:** You have a table Employees with the following structure:

| **EmployeeID** | **Name** | **ManagerID** |
| --- | --- | --- |
| 1 | Alice | NULL |
| 2 | Bob | 1 |
| 3 | Charlie | 1 |
| 4 | Dave | 2 |
| 5 | Eve | 2 |

**Task:** Write a query to find each employee along with their manager’s name. If an employee does not have a manager, show NULL for the manager’s name.

**Hint:** Use a SELF JOIN to join the table to itself. Join EmployeeID with ManagerID to match employees with their managers.

**Question 2: Using SELF JOIN to Find Pairs of Employees in the Same Department**

**Scenario:** You have a table Employees with the following columns:

| **EmployeeID** | **Name** | **DepartmentID** |
| --- | --- | --- |
| 1 | Alice | 101 |
| 2 | Bob | 102 |
| 3 | Charlie | 101 |
| 4 | Dave | 103 |
| 5 | Eve | 102 |

**Task:** Write a query to find pairs of employees who work in the same department. Ensure that no employee pairs with themselves.

**Hint:** Use a SELF JOIN where DepartmentID matches, and exclude pairs where EmployeeID is the same.

**Question 3: Using SELF JOIN to Find Employees with Same Job Title**

**Scenario:** You have a table Employees with the following columns:

| **EmployeeID** | **Name** | **JobTitle** |
| --- | --- | --- |
| 1 | Alice | Developer |
| 2 | Bob | Manager |
| 3 | Charlie | Developer |
| 4 | Dave | Tester |
| 5 | Eve | Developer |

**Task:** Write a query to find pairs of employees who have the same job title.

**Hint:** Use a SELF JOIN on the JobTitle column to match employees with the same job title.

**Question 4: Using UNION to Combine Employee Data**

**Scenario:** You have the table Employees with the following columns:

| **EmployeeID** | **Name** | **DepartmentID** | **JobTitle** |
| --- | --- | --- | --- |
| 1 | Alice | 101 | Developer |
| 2 | Bob | 102 | Manager |
| 3 | Charlie | 101 | Developer |
| 4 | Dave | 103 | Tester |
| 5 | Eve | 102 | Developer |

**Task:** Write a query to display all unique job titles from the Employees table and all unique department IDs. Combine the results using UNION.

**Hint:** Use two SELECT statements: one to select JobTitle and the other to select DepartmentID, and combine them with UNION.

**Question 5: Using UNION ALL to List All Employee Names and Job Titles**

**Scenario:** You have the table Employees with the following columns:

| **EmployeeID** | **Name** | **JobTitle** |
| --- | --- | --- |
| 1 | Alice | Developer |
| 2 | Bob | Manager |
| 3 | Charlie | Developer |
| 4 | Dave | Tester |
| 5 | Eve | Developer |

**Task:** Write a query to display the names of all employees along with their job titles. Use UNION ALL to combine employee names and job titles in a single result.

**Hint:** Use two SELECT statements: one to select the Name and the other to select the JobTitle, and combine them with UNION ALL.

**Question 6: Using UNION to Combine Different Sets of Employee Information**

**Scenario:** You have the table Employees with the following columns:

| **EmployeeID** | **Name** | **DepartmentID** | **JobTitle** |
| --- | --- | --- | --- |
| 1 | Alice | 101 | Developer |
| 2 | Bob | 102 | Manager |
| 3 | Charlie | 101 | Developer |
| 4 | Dave | 103 | Tester |
| 5 | Eve | 102 | Developer |

**Task:** Write a query to combine employee names and department IDs, and display only unique results. Use UNION.

**Hint:** Use two SELECT statements: one for Name and another for DepartmentID, and combine them with UNION.

**Question 7: Using SELF JOIN to Find Employees Reporting to the Same Manager**

**Scenario:** You have the table Employees with the following columns:

| **EmployeeID** | **Name** | **ManagerID** |
| --- | --- | --- |
| 1 | Alice | NULL |
| 2 | Bob | 1 |
| 3 | Charlie | 1 |
| 4 | Dave | 2 |
| 5 | Eve | 2 |

**Task:** Write a query to find employees who report to the same manager. The result should display pairs of employees reporting to the same manager.

**Hint:** Use a SELF JOIN on ManagerID and exclude pairs where EmployeeID is the same.

**Question 8: Using SELF JOIN to Find Employees in the Same Team**

**Scenario:** You have a table Employees with the following columns:

| **EmployeeID** | **Name** | **TeamID** |
| --- | --- | --- |
| 1 | Alice | A |
| 2 | Bob | B |
| 3 | Charlie | A |
| 4 | Dave | B |
| 5 | Eve | A |

**Task:** Write a query to find pairs of employees who belong to the same team.

**Hint:** Use a SELF JOIN on TeamID to match employees in the same team.

**Tips for Solving These Questions:**

* **SELF JOIN**: A SELF JOIN is used to join a table with itself. You'll need to alias the table twice (e.g., Employees E1 and Employees E2) to differentiate the two instances.
* **UNION vs. UNION ALL**: UNION combines results from multiple queries and removes duplicates, while UNION ALL includes all results, even duplicates.
* In SELF JOIN, you'll usually match one field (e.g., ManagerID) in both instances of the same table to retrieve relationships like manager-employee, team members, etc.