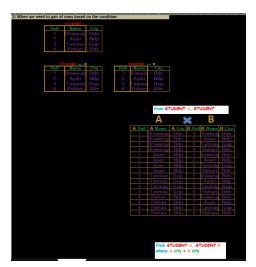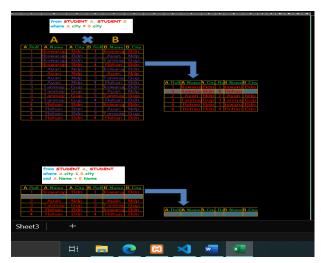Self Join is a join that is used to **join a table with itself**.
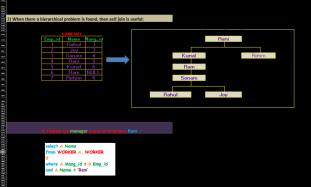
We can perform self join using **table aliases**. This allows us not to use the same table name twice with a single statement. If you don't use aliases it will **throw an error**.

(In Mysql, an **as** keyword can be used **to provide alias names to the table**)

**Table aliases** enable us to use the **temporary name** of the table that we are going to used in the query.







A **self join** is a SQL join that combines a table with itself. This can be useful for tasks such as:

- Finding hierarchical relationships within a table, such as the parent-child relationship in an employee table.
- Finding duplicate rows in a table.
- Calculating aggregate values for different groups of rows in a table.

To perform a self join in MySQL, you use the JOIN clause to join the table to itself, using a common column between the two tables. For example, the following SQL query performs a self join on the employees table to find all employees who have a manager:

SQL
```sql
SELECT e1.name AS employee_name, e2.name AS manager_name
FROM employees e1
JOIN employees e2 ON e1.manager_id = e2.id
WHERE e1.manager_id IS NOT NULL;
```

This query will return a table with two columns: employee_name and manager_name. The employee_name column will contain the name of the employee, and the manager_name column will contain the name of the employee's manager.

To perform a self join in MySQL using PHP, you can use the following steps:

1. Connect to the MySQL database using the mysqli_connect() function.
2. Write a SQL query to perform the self join.
3. Execute the query using the mysqli_query() function.
4. Fetch the results of the query using the mysqli_fetch_assoc() function.
5. Display the results of the query.

Here is an example of a complete PHP script that performs a self join on the employees table to find all employees who have a manager:

PHP
```php
<?php

// Connect to the MySQL database
$mysqli = new mysqli('localhost', 'username', 'password', 'database');

// Write the SQL query to perform the self join
$sql = 'SELECT e1.name AS employee_name, e2.name AS manager_name
FROM employees e1
JOIN employees e2 ON e1.manager_id = e2.id
WHERE e1.manager_id IS NOT NULL';

// Execute the query
$result = $mysqli->query($sql);
```

```php
// Fetch the results of the query
$rows = array();
while ($row = $result->fetch_assoc()) {
    $rows[] = $row;
}

// Close the database connection
$mysqli->close();

// Display the results of the query
echo '<table>';
echo '<tr><th>Employee Name</th><th>Manager Name</th></tr>';
foreach ($rows as $row) {
    echo '<tr>';
    echo '<td>' . $row['employee_name'] . '</td>';
    echo '<td>' . $row['manager_name'] . '</td>';
    echo '</tr>';
}
echo '</table>';

?>
```

This script will output a table with two columns: employee_name and manager_name. The employee_name column will contain the name of the employee, and the manager_name column will contain the name of the employee's manager.

Self joins can be a powerful tool for querying and analyzing data in a single table. They are often used in complex applications, such as social networks and enterprise systems.