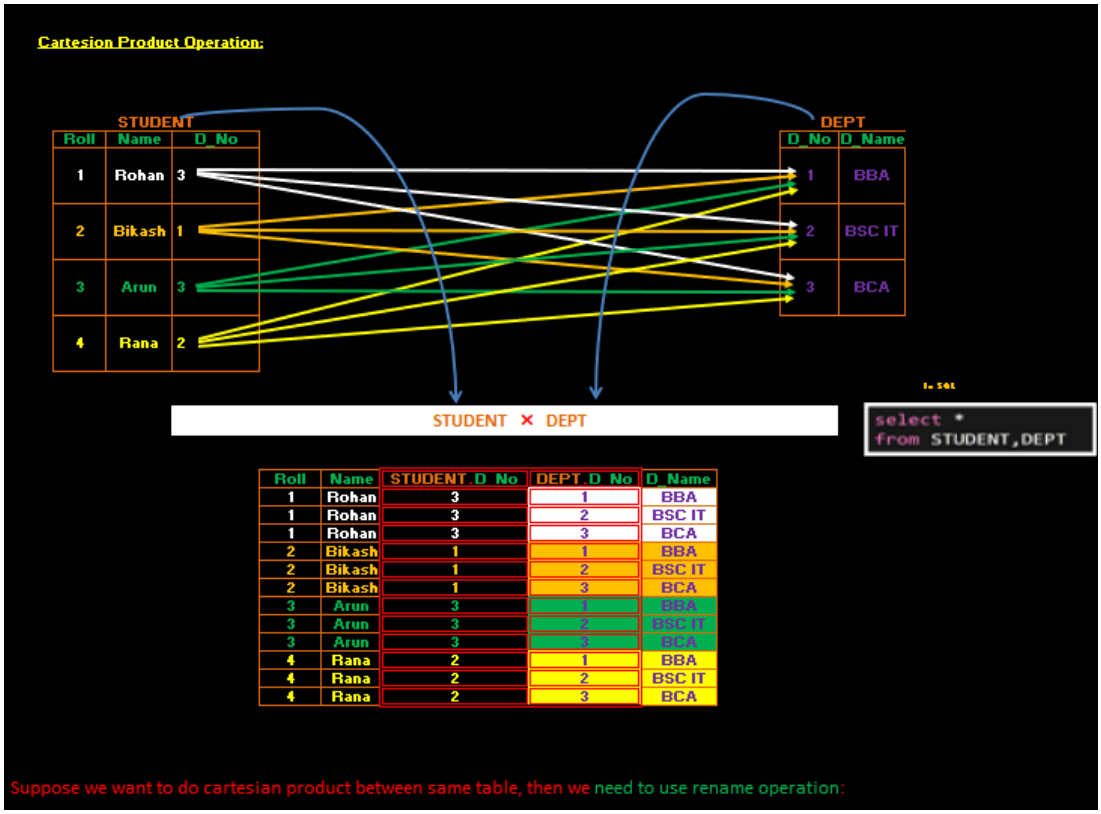

Cartesian Product Operation/Cross Product:

Cartesian product is used to combine each tuple/row in one table with each tuple/row in the other table.

It will take every tuple one by one from the left relation and will pair it up with all the tuples in the right relation.

It is also known as a cross product or cross join.

- The cartesian product operation is denoted by ×.
- Cartesian product is a binary operation means at a time we can apply the operation on two relations.
- The two relations on which we are performing the cartesian product operation, do not have the same type of tuples, which means Union compatibility (or Type compatibility) of the two relations is not necessary.

- The cardinality (number of tuples) of resulting relation from a Cross Product operation is equal to the number of tuples (say m) in the first relation multiplied by the number of tuples in the second relation (say n).

Cardinality = m*n

- The degree (number of attributes) of resulting relation from a Cross Product operation is equal to sum of attributes of both relations.

Degree = m+n

- In cartesian product, the same attribute name may appear in both relations, we need to devise a naming schema to distinguish between these attributes. We do so here by attaching to an attribute the name of the relation from which the attribute originally came.

Here we distinguish STUDENT.D_No from DEPT.D_No
This naming convention causes problem in some case, such as
1) when the Cartesian product of a relation with itself is desired.
2) If we use the result of a relational-algebra expression in a Cartesian product.
In this case we need to use rename operation.

- Generally, a cartesian product is never a meaningful operation when it performs alone. We use Cartesian Product followed by a Selection operation and comparison on the operators as shown below:

- In SQL, cartesian product (cross product) can be applied using cross join.

A cross join in MySQL is a type of join that returns all possible combinations of rows from two or more tables. It is also known as a Cartesian product. To perform a cross join in MySQL using PHP, you can use the following steps:

1. Create a new PHP file and connect to your MySQL database.
2. Write a SQL query to perform the cross join. The following is an example of a cross join query:

SQL
```
SELECT * FROM table1 CROSS JOIN table2;
```

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 cross join between two tables:

PHP
```
<?php

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

// Write the SQL query to perform the cross join
$sql = 'SELECT * FROM customers CROSS JOIN orders';

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

// Fetch the results of the query
```

```php
$rows = array();
while ($row = $result->fetch_assoc()) {
    $rows[] = $row;
}

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

// Display the results of the query
echo '<table>';
echo '<tr><th>Customer Name</th><th>Order ID</th></tr>';
foreach ($rows as $row) {
    echo '<tr>';
    echo '<td>' . $row['customer_name'] . '</td>';
    echo '<td>' . $row['order_id'] . '</td>';
    echo '</tr>';
}
echo '</table>';

?>
```

```php
<?php
  $conn=mysqli_connect("localhost","root","","tanmay");

  if($conn)
    echo "Connection Successfull";
  else
    die("Connection Unsuccessfull!!!");

  $sql="select * from WORKER A CROSS JOIN WORKER B";
  $result=mysqli_query($conn,$sql);

  if (mysqli_num_rows($result)>0){
    while($row=mysqli_fetch_assoc($result))
    {
      print_r($row);
      echo "<br>";
    }
  }
  mysqli_close($conn);

?>
```

When you run this script, it will output a table with all possible combinations of rows from the customers and orders tables. For example, if the customers table has 100 rows and the

orders table has 50 rows, the output table will have 5000 rows.

Cross joins can be useful for generating test data or for performing certain types of analysis. However, they can also produce very large result sets, so it is important to use them carefully.