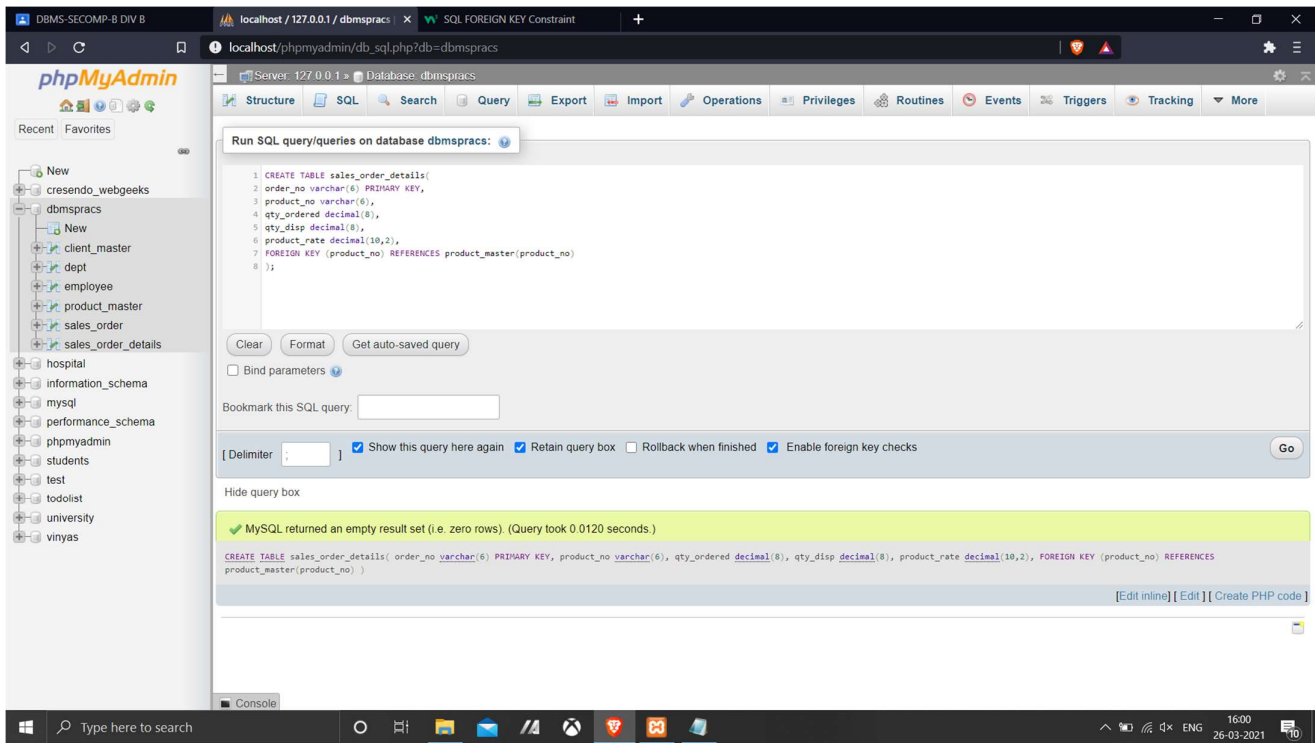


# DBMS Practical Implementation, Lab 6.

## 1. To Create Table sales\_order\_details:

```
CREATE TABLE sales_order_details(  
order_no varchar(6) PRIMARY KEY,  
product_no varchar(6),  
FOREIGN KEY (product_no) REFERENCES product_master(product_no),  
qty_ordered decimal(8),  
qty_disp decimal(8),  
product_rate decimal(10,2)  
);
```



## 2. Find the products which has been sold to client 'Ivan Sayross'

```
SELECT client_master.name ,sales_order_details.order_no,  
product_master.description, sales_order_details.qty_ordered  
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN  
client_master NATURAL JOIN product_master  
WHERE client_master.name = "Ivan Sayross"
```

The screenshot shows the phpMyAdmin web interface. The left sidebar displays a database structure tree with the 'dbmspracs' database selected. The main panel shows the 'Query' tab with a SQL query entered. The query results are displayed in a table with 2 rows and 4 columns: name, order\_no, description, and qty\_ordered. The results show two orders for 'Ivan Sayross': one for a 'Table Fan' (order\_no OD2351, qty\_ordered 50) and one for a 'Portable AC' (order\_no OD2354, qty\_ordered 50). The interface includes various navigation and tool options at the top and bottom.

Server: 127.0.0.1 / dbmspracs

Database: dbmspracs

Bookmark this SQL query: [ ]

[ Delimiter: ; ] ☒ Show this query here again ☒ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks

Hide query box

Showing rows 0 - 1 (2 total, Query took 0.0026 seconds.)

```
SELECT client_master.name ,sales_order_details.order_no, product_master.description, sales_order_details.qty_ordered FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN client_master NATURAL JOIN product_master WHERE client_master.name = "Ivan Sayross"
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

| name         | order_no | description | qty_ordered |
|--------------|----------|-------------|-------------|
| Ivan Sayross | OD2351   | Table Fan   | 50          |
| Ivan Sayross | OD2354   | Portable AC | 50          |

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Bookmark this SQL query

Label: [ ] ☐ Let every user access this bookmark

Console

### 3. Find out product and their quantities that is to be delivered.

```
SELECT product_master.product_no , product_master.description,  
sales_order_details.qty_ordered, product_master.new_price  
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN  
client_master NATURAL JOIN product_master  
WHERE client_master.name = "Ivan Sayross"
```

The screenshot shows the phpMyAdmin web interface. The left sidebar displays a database structure tree with the following databases: **dbmspracs** (expanded), **hospital**, **information\_schema**, **mysql**, **performance\_schema**, **phpmyadmin**, **students**, **test**, **todolist**, **university**, and **vinyas**. Under **dbmspracs**, the tables **client\_master**, **dept**, **employee**, **product\_master**, **sales\_order**, and **sales\_order\_details** are listed. The main panel shows the 'Query' tab selected. A green status bar indicates 'Showing rows 0 - 1 (2 total, Query took 0.0024 seconds.)'. The SQL query entered is: `SELECT product_master.product_no , product_master.description, sales_order_details.qty_ordered, product_master.new_price FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN client_master NATURAL JOIN product_master WHERE client_master.name = "Ivan Sayross"`. Below the query, there are controls for 'Show all', 'Number of rows' (set to 25), 'Filter rows' (a search box), and 'Sort by key' (set to None). The query results are displayed in a table with the following data:

| product_no | description | qty_ordered | new_price |
|------------|-------------|-------------|-----------|
| 1001       | Table Fan   | 50          | 6000.00   |
| 1003       | Portable AC | 50          | 4500.00   |

Below the table, there are controls for 'Show all', 'Number of rows' (set to 25), 'Filter rows' (a search box), and 'Sort by key' (set to None). The 'Query results operations' section includes links for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'. The 'Bookmark this SQL query' section has a 'Label' input field and a checkbox 'Let every user access this bookmark'. A 'Bookmark this SQL query' button is located at the bottom right of the results area. The bottom of the screen shows a Windows taskbar with the search bar, task view button, and several application icons. The system clock shows 16:44 on 27-03-2021.

#### 4. Find out the product number and description of Moving products.

```
SELECT product_master.product_no , product_master.description  
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN  
client_master NATURAL JOIN product_master  
WHERE sales_order.order_status="Moving";
```

The screenshot shows the phpMyAdmin web interface. The left sidebar displays a database structure with the following items: New, cresendo\_webgeeks, dbmspracs, New, client\_master, dept, employee, product\_master, sales\_order, sales\_order\_details, hospital, information\_schema, mysql, performance\_schema, phpmyadmin, students, test, todolist, university, and vinyas. The main panel shows a SQL query executed successfully, displaying 2 rows of data. The query is: `SELECT product_master.product_no , product_master.description, sales_order_details.qty_ordered, product_master.new_price FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN client_master NATURAL JOIN product_master WHERE sales_order.order_status="Moving"`. The results table has columns: product\_no, description, qty\_ordered, and new\_price. The data rows are: 1003 Portable AC (50, 4500.00) and 1011 floppies (4, 30.00). Below the results, there are options to show all rows, filter rows, and sort by key. There are also buttons for query results operations: Print, Copy to clipboard, Export, Display chart, and Create view. At the bottom, there is a section for bookmarking the query, with a label field and a checkbox for 'Let every user access this bookmark'. The console at the bottom shows the query execution details.

Showing query box

Showing rows 0 - 1 (2 total, Query took 0.0023 seconds)

```
SELECT product_master.product_no , product_master.description, sales_order_details.qty_ordered, product_master.new_price FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN client_master NATURAL JOIN product_master WHERE sales_order.order_status="Moving"
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

| product_no | description | qty_ordered | new_price |
|------------|-------------|-------------|-----------|
| 1003       | Portable AC | 50          | 4500.00   |
| 1011       | floppies    | 4           | 30.00     |

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label:  ☐ Let every user access this bookmark

Bookmark this SQL query

Console

5. Find out the names of clients who have purchased 'CD Drive'.

```
SELECT client_master.name ,sales_order_details.order_no,  
product_master.description, sales_order_details.qty_ordered,  
sales_order.order_status  
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN  
client_master NATURAL JOIN product_master  
WHERE product_master.description="CD Drive";
```

The screenshot shows the phpMyAdmin web interface in a browser. The left sidebar displays a database structure tree with the 'dbmspracs' database selected. The main panel shows a SQL query executed successfully, returning one row of data. The query is: `SELECT client_master.name ,sales_order_details.order_no, product_master.description, sales_order_details.qty_ordered, sales_order.order_status FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN client_master NATURAL JOIN product_master WHERE product_master.description="CD Drive"`. The result table has columns: name, order\_no, description, qty\_ordered, and order\_status. The single row shows: Jayesh Badwal, OD2353, CD Drive, 250, and Shipped. Below the table, there are options to bookmark the query and a console area at the bottom.

| name          | order_no | description | qty_ordered | order_status |
|---------------|----------|-------------|-------------|--------------|
| Jayesh Badwal | OD2353   | CD Drive    | 250         | Shipped      |

6. List the product\_no and order\_no of customers having quantity ordered less than 5 from sales\_order\_details table for the product 'floppies'.

```
SELECT client_master.name ,sales_order_details.order_no,  
product_master.product_no,  
sales_order_details.qty_ordered,product_master.description  
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN  
client_master NATURAL JOIN product_master  
WHERE product_master.description="floppies" and  
sales_order_details.qty_ordered<5;
```

The screenshot shows the phpMyAdmin web interface. The left sidebar displays a database structure with 'dbmspracs' selected. The main panel shows a SQL query executed successfully, returning one row of data. The query is: `SELECT client_master.name ,sales_order_details.order_no, product_master.product_no, sales_order_details.qty_ordered,product_master.description FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN client_master NATURAL JOIN product_master WHERE product_master.description="floppies" and sales_order_details.qty_ordered<5`. The result table has columns: name, order\_no, product\_no, qty\_ordered, and description. The single row shows: Charan Singh, OD2355, 1011, 4, floppies. Below the table, there are options to print, copy to clipboard, export, display chart, and create view. A bookmark section is also visible at the bottom.

Showing rows 0 - 0 (1 total, Query took 0.0036 seconds)

```
SELECT client_master.name ,sales_order_details.order_no, product_master.product_no, sales_order_details.qty_ordered,product_master.description FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN client_master NATURAL JOIN product_master WHERE product_master.description="floppies" and sales_order_details.qty_ordered<5
```

| name         | order_no | product_no | qty_ordered | description |
|--------------|----------|------------|-------------|-------------|
| Charan Singh | OD2355   | 1011       | 4           | floppies    |

Query results operations: Print, Copy to clipboard, Export, Display chart, Create view

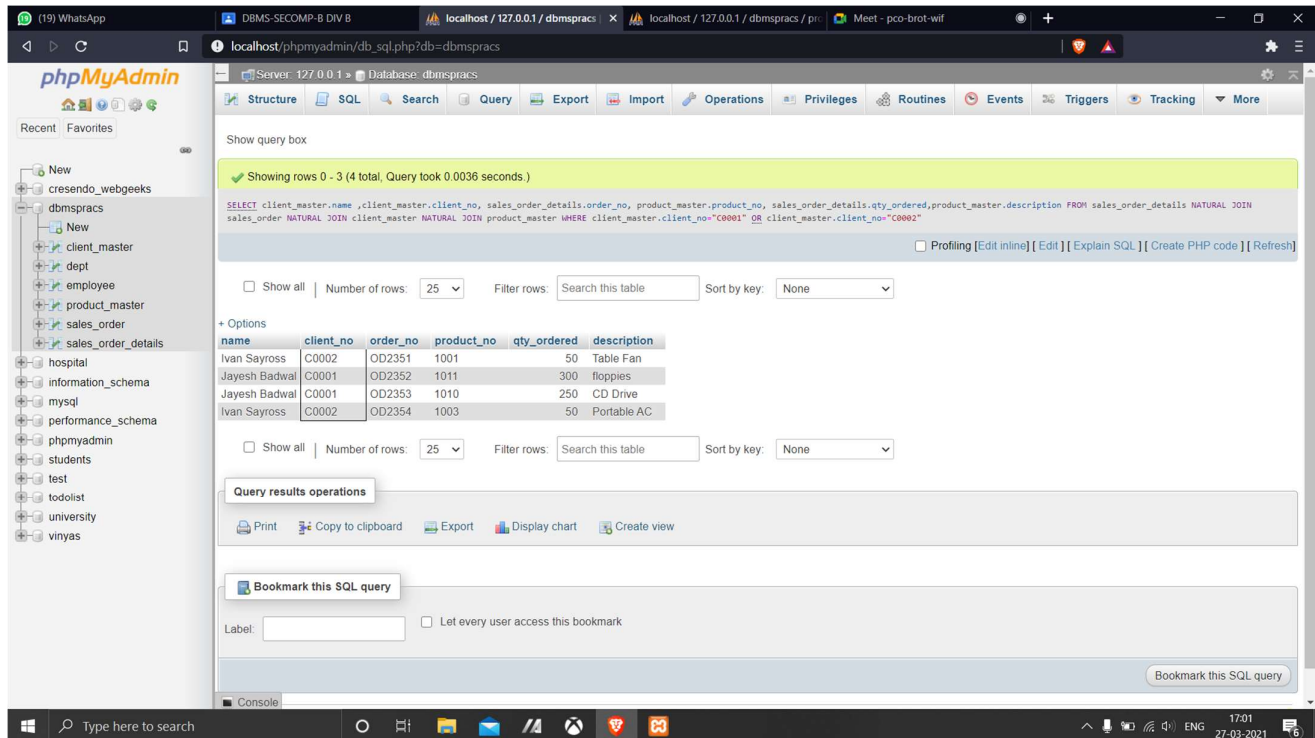
Bookmark this SQL query

Label:  ☐ Let every user access this bookmark

Bookmark this SQL query

7. Find the products and their quantities for the orders placed by client\_no 'C0001' and 'C0002'.

```
SELECT client_master.name ,client_master.client_no,  
sales_order_details.order_no, product_master.product_no,  
sales_order_details.qty_ordered,product_master.description  
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN  
client_master NATURAL JOIN product_master  
WHERE client_master.client_no="C0001" OR  
client_master.client_no="C0002";
```



The screenshot shows the phpMyAdmin interface with the following details:

- Query:** `SELECT client_master.name ,client_master.client_no, sales_order_details.order_no, product_master.product_no, sales_order_details.qty_ordered,product_master.description FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN client_master NATURAL JOIN product_master WHERE client_master.client_no="C0001" OR client_master.client_no="C0002"`
- Results:** Showing rows 0 - 3 (4 total, Query took 0.0036 seconds.)
- Table:**

| name          | client_no | order_no | product_no | qty_ordered | description |
|---------------|-----------|----------|------------|-------------|-------------|
| Ivan Sayross  | C0002     | OD2351   | 1001       | 50          | Table Fan   |
| Jayesh Badwal | C0001     | OD2352   | 1011       | 300         | floppies    |
| Jayesh Badwal | C0001     | OD2353   | 1010       | 250         | CD Drive    |
| Ivan Sayross  | C0002     | OD2354   | 1003       | 50          | Portable AC |

- Options:** Show all, Number of rows: 25, Filter rows: Search this table, Sort by key: None
- Query results operations:** Print, Copy to clipboard, Export, Display chart, Create view
- Bookmark this SQL query:** Label: [ ], Let every user access this bookmark [ ]

8. Find the description and total quantity sold for each products.

SELECT product\_master.description, SUM(sales\_order\_details.qty\_ordered) AS  
UNITS SOLD

FROM sales\_order\_details NATURAL JOIN sales\_order NATURAL JOIN  
client\_master NATURAL JOIN product\_master

GROUP BY product\_master.description

ORDER BY product\_master.description

The screenshot shows the phpMyAdmin web interface. The left sidebar displays a database structure tree with the following databases: cresendo\_webgeeks, dbmspracs, client\_master, dept, employee, product\_master, sales\_order, sales\_order\_details, hospital, information\_schema, mysql, performance\_schema, phpmysadmin, students, test, todoist, university, and vinyas. The main panel is titled 'Database: dbmspracs > Table: product\_master'. It shows a message: 'Showing rows 0 - 4 (5 total, Query took 0.0043 seconds) [description: CD DRIVE... - TABLE FAN...]'. Below this, the SQL query is displayed: 

```
SELECT product_master.description, SUM(sales_order_details.qty_ordered) FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN client_master NATURAL JOIN product_master GROUP BY product_master.description ORDER BY product_master.description
```

. The query results are shown in a table with two columns: 'description' and 'SUM(sales\_order\_details.qty\_ordered)'. The results are as follows:

| description | SUM(sales_order_details.qty_ordered) |
|-------------|--------------------------------------|
| CD Drive    | 250                                  |
| floppies    | 304                                  |
| Moving      | 4                                    |
| Portable AC | 50                                   |
| Table Fan   | 50                                   |

At the bottom of the interface, there is a 'Query results operations' section with buttons for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'. Below that is a 'Bookmark this SQL query' section with a checkbox labeled 'Let every user access this bookmark'.



## 9. Find the value of each product sold.

```
SELECT product_master.description,SUM(sales_order_details.qty_ordered) AS  
UNITS_SOLD  
,SUM(sales_order_details.qty_ordered)*sales_order_details.product_rate AS  
Value_Recieved  
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN  
client_master NATURAL JOIN product_master  
GROUP BY product_master.description  
ORDER BY product_master.description
```

The screenshot shows the phpMyAdmin web interface. The left sidebar displays a database structure with several databases, including 'dbmspracs'. The main panel shows the 'product\_master' table selected. A message at the top states: 'Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.' Below this, a green banner indicates 'Showing rows 0 - 4 (5 total, Query took 0.0026 seconds.) [description: CD DRIVE... - TABLE FAN...]'. The SQL query is displayed in a text area: `SELECT product_master.description, SUM(sales_order_details.qty_ordered) AS UNITS_SOLD FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN client_master NATURAL JOIN product_master GROUP BY product_master.description ORDER BY product_master.description`. Below the query, a table displays the results:

| description | UNITS_SOLD |
|-------------|------------|
| CD Drive    | 250        |
| floppies    | 304        |
| Moving      | 4          |
| Portable AC | 50         |
| Table Fan   | 50         |

Below the table, there are options for 'Show all', 'Number of rows' (set to 25), and a 'Filter rows' search box. At the bottom, there are links for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'. A 'Bookmark this SQL query' section is also visible, with a label input field and a checkbox 'Let every user access this bookmark'.

10. Find out the name of customers who have given the order of more than 10 qty.

```
SELECT client_master.client_no, client_master.name,  
SUM(sales_order_details.qty_ordered) AS Units_Ordered  
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN  
client_master NATURAL JOIN product_master  
GROUP BY client_master.client_no  
HAVING SUM(sales_order_details.qty_ordered)>10  
ORDER BY sales_order_details.qty_ordered DESC
```

The screenshot shows the phpMyAdmin web interface. The left sidebar displays a database structure tree with 'dbmspracs' selected. The main panel shows the 'client\_master' table. A SQL query is entered in the 'SQL' tab, and its results are displayed in a table. The query is: `SELECT client_master.client_no, client_master.name, SUM(sales_order_details.qty_ordered) AS Units_Ordered FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN client_master NATURAL JOIN product_master GROUP BY client_master.client_no HAVING SUM(sales_order_details.qty_ordered)>10 ORDER BY sales_order_details.qty_ordered DESC`. The results table has three columns: 'client\_no', 'name', and 'Units\_Ordered'. It contains two rows: 'C0001' for 'Jayesh Badwal' with 550 units, and 'C0002' for 'Ivan Sayross' with 100 units. The interface also includes a 'Query results operations' section with buttons for Print, Copy to clipboard, Export, Display chart, and Create view. A 'Bookmark this SQL query' section is at the bottom.

| client_no | name          | Units_Ordered |
|-----------|---------------|---------------|
| C0001     | Jayesh Badwal | 550           |
| C0002     | Ivan Sayross  | 100           |

## Postlab EXP 6.

Q1) Difference between inner join and outer join  
 As (i) Inner Join

(a) It returns the combined tuple between two or more tables.

(b) When any attributes are not common then it will return nothing.

(c) If tuples are more. Then INNER JOIN works faster than OUTER JOIN.

(d) It is used where we want detailed information about any specific attribute.

(e) Used clause INNER JOIN and JOIN.

(ii) Outer Join.

(a) It will return the combined tuple for a specified table even join condition will fail.

(b) It does not depend upon the common attributes. If the attributes is blank then here are applied NULL.

(c) Generally, the OUTER JOIN is slower than INNER JOIN. But except for some special cases.

(d) It is used when we want complete information.

(e) Used clause LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN etc.

12) EQUI JOIN & Non Equi Join examples.

(i) EQUI Join creates Join by using Join with ON and then providing the names of the column with their respective tables to check equal using equal sign (=).

Example:-

```
SELECT student.name, record.class, record.city  
FROM student, record  
WHERE student.city = record.city;
```

(ii) Non EQUI JOIN performs a JOIN using comparison operator other than equal (=) sign like  $\neq$ ,  $<$ ,  $>$ ,  $\leq$ ,  $\geq$  with condition.

Example:-

```
SELECT student.name, record.id, record.city  
FROM student, record  
WHERE student.id < record.id;
```





# Exercise:

Choose the correct `JOIN` clause to select all records from the two tables where there is a match in both tables.

```
SELECT *  
FROM Orders  
Inner Join Customer  
ON Orders.CustomerID=  
Customers.CustomerID;
```

[Submit Answer >](#)[Show Answer](#)



# Exercise:

Choose the correct `JOIN` clause to select all the records from the `Customers` table plus all the matches in the `Orders` table.

```
SELECT *  
FROM Orders  
Right Join Customers  
ON Orders.CustomerID=  
Customers.CustomerID;
```

[Submit Answer >](#)[Show Answer](#)



# Exercise:

List the number of customers in each country.

```
SELECT Count(CustomerID),  
Country  
FROM Customers  
Group by country;
```

Submit Answer >

Show Answer





# Exercise:

List the number of customers in each country, ordered by the country with the most customers first.

```
SELECT Count(CustomerID),  
Country  
FROM Customers  
Group by country  
ORDER BY  
COUNT(CustomerID) DESC;
```

[Submit Answer >](#)[Show Answer](#)





# Exercise:

Insert the missing parts in the `JOIN` clause to join the two tables `Orders` and `Customers`, using the `CustomerID` field in both tables as the relationship between the two tables.

Correct!

Next >

Next Exercise >

Show Answer

