



UITs

UNIVERSITY OF INFORMATION
TECHNOLOGY AND SCIENCES

DBMS LAB REPORT

Course Code: CSE0612216S

Submitted To

Ms. Tania Akter Setu

Assistant Professor

Submitted By

Helal uddin Patwary

ID: 0432410005101086

Section: 3B2

Batch: 55

Computer Science and Engineering

DBMS LAB Assessment Query with Results

Created Table Query:

```
CREATE TABLE salesman (  
  salesman_id INTEGER PRIMARY KEY,  
  name TEXT NOT NULL,  
  city TEXT NOT NULL,  
  commission decimal(3,2)  
);  
INSERT INTO salesman VALUES (5001, 'James Hoog', 'New York', 0.15);  
INSERT INTO salesman VALUES (5002, 'Nail Knite', 'Paris', 0.13);  
INSERT INTO salesman VALUES (5005, 'Pit Alex', 'London', 0.11);  
INSERT INTO salesman VALUES (5006, 'Mc Lyon', 'Paris', 0.14);  
INSERT INTO salesman VALUES (5007, 'Paul Adam', 'Rome', 0.13);  
INSERT INTO salesman VALUES (5003, 'Lauson Hen', 'San Jose', 0.12);
```

```
CREATE TABLE customer (  
  customer_id INTEGER PRIMARY KEY,  
  cust_name TEXT NOT NULL,  
  city TEXT NOT NULL,  
  grade INTEGER,  
  salesman_id INTEGER  
);  
INSERT INTO customer VALUES (3002, 'Nick Rimando', 'New York', 100, 5001);  
INSERT INTO customer VALUES (3007, 'Brad Davis', 'New York', 200, 5001);  
INSERT INTO customer VALUES (3005, 'Graham Zusi', 'California', 200, 5002);  
INSERT INTO customer VALUES (3008, 'Julian Green', 'London', 300, 5002);  
INSERT INTO customer VALUES (3004, 'Fabian Johnson', 'Paris', 300, 5006);  
INSERT INTO customer VALUES (3009, 'Geoff Cameron', 'Berlin', 100, 5003);  
INSERT INTO customer VALUES (3003, 'Jozy Altidor', 'Moscow', 200, 5007);  
INSERT INTO customer VALUES (3001, 'Brad Guzan', 'London', NULL, 5005);
```

```

CREATE TABLE orders (
ord_no INTEGER PRIMARY KEY,
purch_amt INTEGER,
ord_date date,
customer_id INTEGER,

salesman_id INTEGER
);
INSERT INTO orders VALUES (70001, 150.5, '2012-10-05', 3005, 5002);
INSERT INTO orders VALUES (70009, 270.65, '2012-09-10', 3001, 5005);
INSERT INTO orders VALUES (70002, 65.26, '2012-10-05', 3002, 5001);
INSERT INTO orders VALUES (70004, 110.5, '2012-08-17', 3009, 5003);
INSERT INTO orders VALUES (70007, 948.5, '2012-09-10', 3005, 5002);
INSERT INTO orders VALUES (70005, 2400.6, '2012-07-27', 3007, 5001);
INSERT INTO orders VALUES (70008, 5760, '2012-09-10', 3002, 5001);
INSERT INTO orders VALUES (70010, 1983.43, '2012-10-10', 3004, 5006);
INSERT INTO orders VALUES (70003, 2480.4, '2012-10-10', 3009, 5003);
INSERT INTO orders VALUES (70012, 250.45, '2012-06-27', 3008, 5002);
INSERT INTO orders VALUES (70011, 75.29, '2012-08-17', 3003, 5007);
INSERT INTO orders VALUES (70013, 3045.6, '2012-04-25', 3002, 5001);

```

Questions Query with Results:

Q1. From the following tables write a SQL query to find the salesperson and customer who belong to the same city. Return Salesman, cust_name and city.

```

SELECT s.name AS Salesman, c.cust_name, c.city
FROM salesman s
JOIN customer c ON s.salesman_id = c.salesman_id
WHERE s.city = c.city;

```

The screenshot shows the phpMyAdmin interface with the 'lab_2' database selected. A SQL query is executed, showing 6 rows. The query is:

```
SELECT salesman.name AS salesman, customer.cust_name, customer.city FROM salesman JOIN customer ON customer.city = salesman.city;
```

The result table has the following data:

salesman	cust_name	city
Pit Alex	Brad Guzan	London
James Hoog	Nick Rimando	New York
Nail Knite	Fabian Johnson	Paris
Mc Lyon	Fabian Johnson	Paris
James Hoog	Brad Davis	New York
Pit Alex	Julian Green	London

Q2. From the following tables write a SQL query to find those orders where order amount exists between 500 and 2000. Return ord_no, purch_amt, cust_name, city.

```
SELECT o.ord_no, o.purch_amt, c.cust_name, c.city
FROM orders o
JOIN customer c ON o.customer_id = c.customer_id
WHERE o.purch_amt BETWEEN 500 AND 2000;
```

The screenshot shows the phpMyAdmin interface with the 'lab_2' database selected. A SQL query is executed, showing 2 rows. The query is:

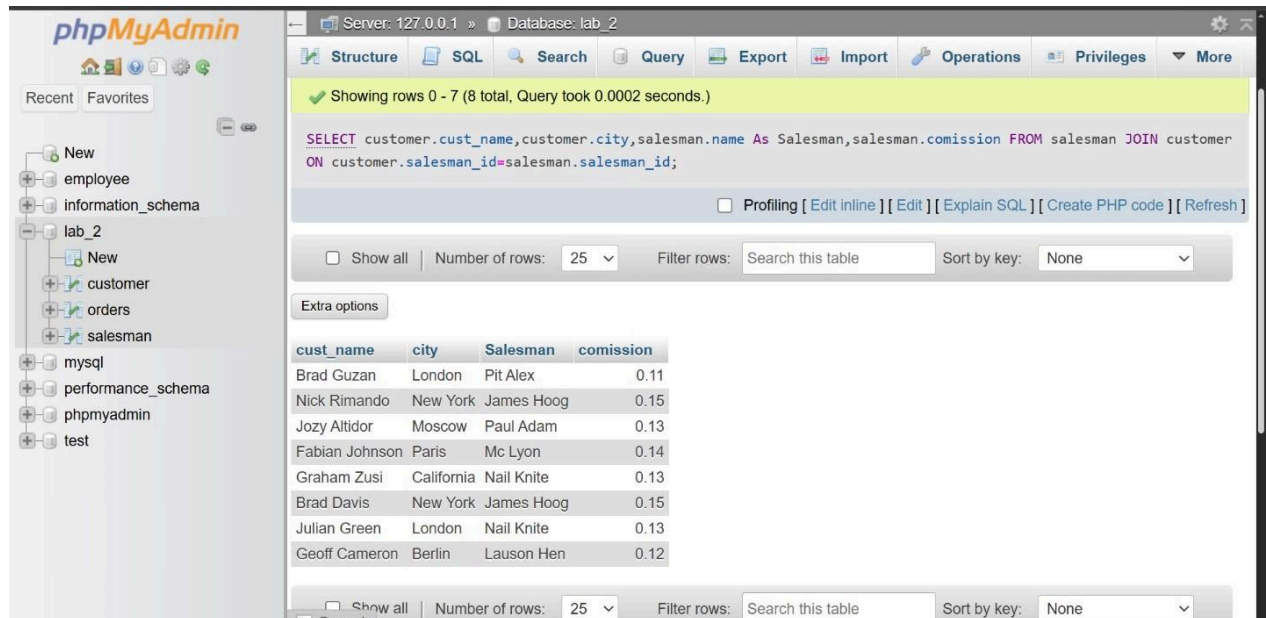
```
SELECT orders.ord_no, orders.purch_amt, customer.cust_name, customer.city FROM orders JOIN customer ON orders.customer_id = customer.customer_id WHERE orders.purch_amt BETWEEN 500 AND 2000;
```

The result table has the following data:

ord_no	purch_amt	cust_name	city
70007	948.5	Graham Zusi	California
70010	1983.43	Fabian Johnson	Paris

Q3. From the following tables write a SQL query to find the salesperson(s) and the customer(s) he handle. Return Customer Name, city, Salesman, commission.

```
SELECT c.cust_name, c.city, s.name AS Salesman, s.commission
FROM customer c
JOIN salesman s ON c.salesman_id = s.salesman_id;
```



The screenshot shows the phpMyAdmin interface with the 'lab_2' database selected. A SQL query is entered in the 'Query' tab, and the results are displayed in a table. The query is: `SELECT customer.cust_name, customer.city, salesman.name As Salesman, salesman.commission FROM salesman JOIN customer ON customer.salesman_id=salesman.salesman_id;` The results table has 8 rows and 4 columns: cust_name, city, Salesman, and comission (note the typo in the column name). The data is as follows:

cust_name	city	Salesman	comission
Brad Guzan	London	Pit Alex	0.11
Nick Rimando	New York	James Hoog	0.15
Jozy Altidor	Moscow	Paul Adam	0.13
Fabian Johnson	Paris	Mc Lyon	0.14
Graham Zusi	California	Nail Knite	0.13
Brad Davis	New York	James Hoog	0.15
Julian Green	London	Nail Knite	0.13
Geoff Cameron	Berlin	Lauson Hen	0.12

Q4. From the following tables write a SQL query to find those salespersons who received a commission from the company more than 12%. Return Customer Name, customer city, Salesman, commission

```
SELECT c.cust_name, c.city AS customer_city, s.name AS Salesman, s.commission
FROM customer c
JOIN salesman s ON c.salesman_id = s.salesman_id
WHERE s.commission > 0.12;
```

phpMyAdmin

Server: 127.0.0.1 » Database: lab_2

Structure SQL Search Query Export Import Operations Privileges More

Showing rows 0 - 5 (6 total, Query took 0.0002 seconds.)

```
SELECT customer.cust_name, customer.city, salesman.name AS salesman, salesman.comission FROM salesman JOIN customer ON customer.salesman_id=salesman.salesman_id WHERE salesman.comission>.12;
```

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

Extra options

cust_name	city	salesman	comission
Nick Rimando	New York	James Hoog	0.15
Jozy Altidor	Moscow	Paul Adam	0.13
Fabian Johnson	Paris	Mc Lyon	0.14
Graham Zusi	California	Nail Knite	0.13
Brad Davis	New York	James Hoog	0.15
Julian Green	London	Nail Knite	0.13

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

Console

Q5. From the following tables write a SQL query to find those salespersons do not live in the same city where their customers live and received a commission from the company more than 12%. Return Customer Name, customer city, Salesman, salesman city, commission.

```
SELECT c.cust_name, c.city AS customer_city, s.name AS Salesman, s.city AS
salesman_city, s.commission
FROM customer c
JOIN salesman s ON c.salesman_id = s.salesman_id
WHERE s.city != c.city AND s.commission > 0.12;
```

phpMyAdmin

Server: 127.0.0.1 » Database: lab_2

Structure SQL Search Query Export Import Operations Privileges More

Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)

```
SELECT customer.cust_name, customer.city, salesman.name AS salesman, salesman.city, salesman.comission FROM salesman JOIN customer ON customer.salesman_id=salesman.salesman_id WHERE customer.city!=salesman.city;
```

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

Extra options

cust_name	city	salesman	city	comission
Jozy Altidor	Moscow	Paul Adam	Rome	0.13
Graham Zusi	California	Nail Knite	Paris	0.13
Julian Green	London	Nail Knite	Paris	0.13
Geoff Cameron	Berlin	Lauson Hen	San Jose	0.12

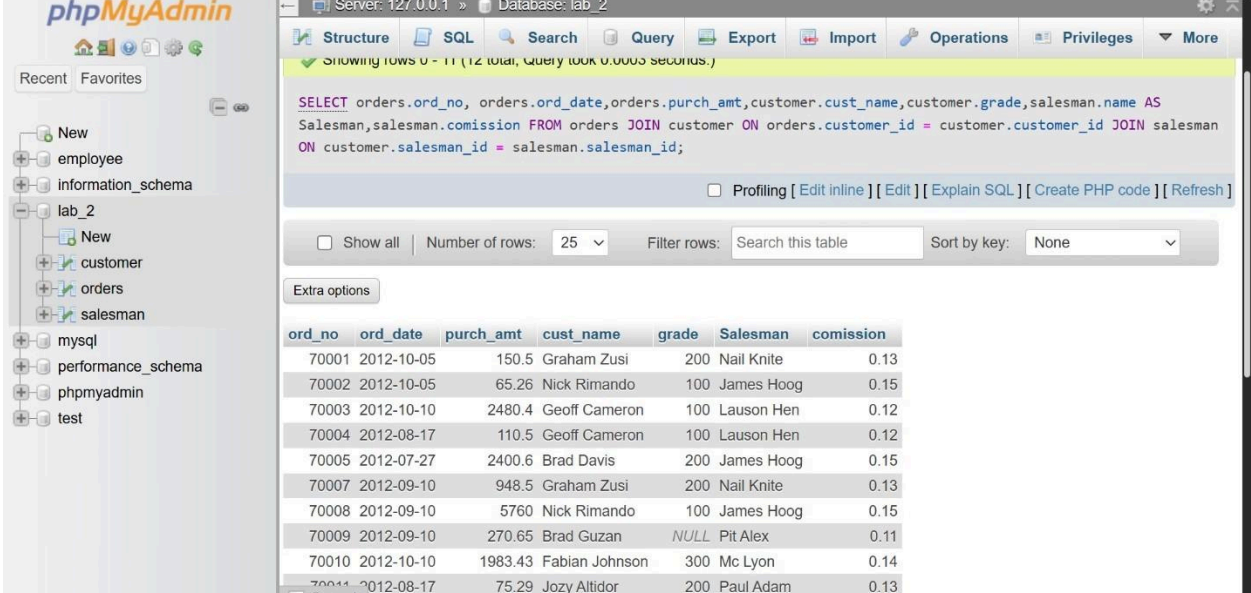
Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Console

Q6. From the following tables write a SQL query to find the details of an order. Return ord_no,ord_date, purch_amt, Customer Name, grade, Salesman, commission.

```
SELECT o.order_no,o.ord_date, o.purchase_amt, c.cust_name,c.grade,s.name
AS salesman, s.commission
FROM orders o
JOIN
    customer c ON o.customer_id = c.customer_id
JOIN
    salesman s ON o.salesman_id = s.salesman_id;
```



The screenshot shows the phpMyAdmin interface for a database named 'lab_2'. The SQL query entered in the 'Query' tab is:

```
SELECT orders.ord_no, orders.ord_date, orders.purch_amt, customer.cust_name, customer.grade, salesman.name AS
Salesman, salesman.commission FROM orders JOIN customer ON orders.customer_id = customer.customer_id JOIN salesman
ON customer.salesman_id = salesman.salesman_id;
```

The results are displayed in a table with the following columns: ord_no, ord_date, purch_amt, cust_name, grade, Salesman, and comission. The table contains 11 rows of data.

ord_no	ord_date	purch_amt	cust_name	grade	Salesman	comission
70001	2012-10-05	150.5	Graham Zusi	200	Nail Knite	0.13
70002	2012-10-05	65.26	Nick Rimando	100	James Hoog	0.15
70003	2012-10-10	2480.4	Geoff Cameron	100	Lauson Hen	0.12
70004	2012-08-17	110.5	Geoff Cameron	100	Lauson Hen	0.12
70005	2012-07-27	2400.6	Brad Davis	200	James Hoog	0.15
70007	2012-09-10	948.5	Graham Zusi	200	Nail Knite	0.13
70008	2012-09-10	5760	Nick Rimando	100	James Hoog	0.15
70009	2012-09-10	270.65	Brad Guzan	NULL	Pit Alex	0.11
70010	2012-10-10	1983.43	Fabian Johnson	300	Mc Lyon	0.14
70011	2012-08-17	75.29	Jozy Altidor	200	Paul Adam	0.13

Q7. From the following tables write a SQL query to display the cust_name, customer city, grade, Salesman, salesman city. The result should be ordered by ascending on customer_id.

```
SELECT
    c.cust_name,
    c.city AS customer_city,
    c.grade,
    s.name AS salesman,
    s.city AS salesman_city
FROM
    customer c
JOIN
    salesman s ON c.salesman_id = s.salesman_id
ORDER BY
    c.customer_id ASC;
```


The screenshot shows the phpMyAdmin interface for a database named 'lab_2'. The left sidebar displays a tree view of the database structure, including tables like 'customer', 'orders', and 'salesman'. The main panel shows a SQL query that has been executed, displaying the results in a table format. The query is:
`SELECT customer.cust_name, customer.city AS customer_city, customer.grade, salesman.name AS Salesman, salesman.city AS salesman_city FROM customer JOIN salesman ON customer.salesman_id = salesman.salesman_id ORDER BY customer.customer_id ASC;`
 The results table has 5 columns: cust_name, customer_city, grade, Salesman, and salesman_city. It contains 10 rows of data, ordered by customer_id in ascending order. The interface also shows options for filtering, sorting, and exporting the data.

cust_name	customer_city	grade	Salesman	salesman_city
Brad Guzan	London	NULL	Pit Alex	London
Nick Rimando	New York	100	James Hoog	New York
Jozy Altidor	Moscow	200	Paul Adam	Rome
Fabian Johnson	Paris	300	Mc Lyon	Paris
Graham Zusi	California	200	Nail Knite	Paris
Brad Davis	New York	200	James Hoog	New York
Julian Green	London	300	Nail Knite	Paris
Geoff Cameron	Berlin	100	Lauson Hen	San Jose

Q8. From the following tables write a SQL query to find those customers whose grade less than 300. Return cust_name, customer city, grade, Salesman, saleman city. The result should be ordered by ascending customer_id

```
SELECT
    c.cust_name,
    c.city AS customer_city,
    c.grade,
    s.name AS salesman,
    s.city AS salesman_city
FROM
    customer c
JOIN
    salesman s ON c.salesman_id = s.salesman_id
WHERE
    c.grade < 300
ORDER BY
    c.customer_id ASC;
```


The screenshot shows the phpMyAdmin interface for a database named 'lab_2'. The left sidebar displays the database structure, including tables like 'customer', 'orders', and 'salesman'. The main panel shows a SQL query that has been executed, displaying 5 rows of data. The query is as follows:

```
SELECT customer.cust_name, customer.city, customer.grade, salesman.name AS Salesman, salesman.city FROM customer
JOIN salesman ON salesman.salesman_id=customer.salesman_id WHERE customer.grade<300 ORDER BY
customer.customer_id ASC;
```

The results table shows the following data:

cust_name	city	grade	Salesman	city
Nick Rimando	New York	100	James Hoog	New York
Jozy Altidor	Moscow	200	Paul Adam	Rome
Graham Zusi	California	200	Nail Knite	Paris
Brad Davis	New York	200	James Hoog	New York
Geoff Cameron	Berlin	100	Lauson Hen	San Jose

Q9. Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to find that either any of the existing customers have placed no order or placed one or more orders.

```
SELECT
    c.cust_name,
    c.city,
    o.order_no,
    o.ord_date,
    o.purchase_amt
FROM
    customer c
LEFT JOIN
    orders o ON c.customer_id = o.customer_id
ORDER BY
    o.ord_date ASC;
```

The screenshot shows the phpMyAdmin interface with the following details:

- Server:** 127.0.0.1
- Database:** lab_2
- Navigation Panel:**
 - Recent: New
 - Favorites: employee, information_schema, lab_2, customer, orders, salesman, mysql, performance_schema, phpmyadmin, test
- SQL Query:**

```
SELECT customer.cust_name AS Customer_name, customer.city, orders.ord_no, orders.ord_date, orders.purch_amt FROM customer LEFT JOIN orders ON orders.customer_id=customer.customer_id ORDER BY `orders`.`ord_date` ASC;
```
- Query Results:**

Customer_name	city	ord_no	ord_date	1	purch_amt
Nick Rimando	New York	70013	2012-04-25		3045.6
Julian Green	London	70012	2012-06-27		250.45
Brad Davis	New York	70005	2012-07-27		2400.6
Geoff Cameron	Berlin	70004	2012-08-17		110.5
Jozy Altidor	Moscow	70011	2012-08-17		75.29
Graham Zusi	California	70007	2012-09-10		948.5
Nick Rimando	New York	70008	2012-09-10		5760
Brad Guzan	London	70009	2012-09-10		270.65
Graham Zusi	California	70001	2012-10-05		150.5
Nick Rimando	New York	70002	2012-10-05		65.26
ron	Berlin	70003	2012-10-10		2480.4

Q10. Write a SQL statement to make a report with customer name, city, order number, order date, order amount salesman name and commission to find that either any of the existing customers have placed no order or placed one or more orders by their salesman or by own.

```
SELECT
    c.cust_name,
    c.city,
    o.order_no,
    o.ord_date,
    o.purchase_amt,
    s.name AS salesman,
    s.commission
FROM
    customer c
LEFT JOIN
    orders o ON c.customer_id = o.customer_id
LEFT JOIN
    salesman s ON c.salesman_id = s.salesman_id
ORDER BY
    o.ord_date ASC;
```

The screenshot shows the phpMyAdmin interface with a SQL query executed. The query is a LEFT JOIN between the customer, salesman, and orders tables. The results table displays columns: cust_name, city, ord_no, ord_date, purch_amt, name, and comission. The results are sorted by salesman name in ascending order.

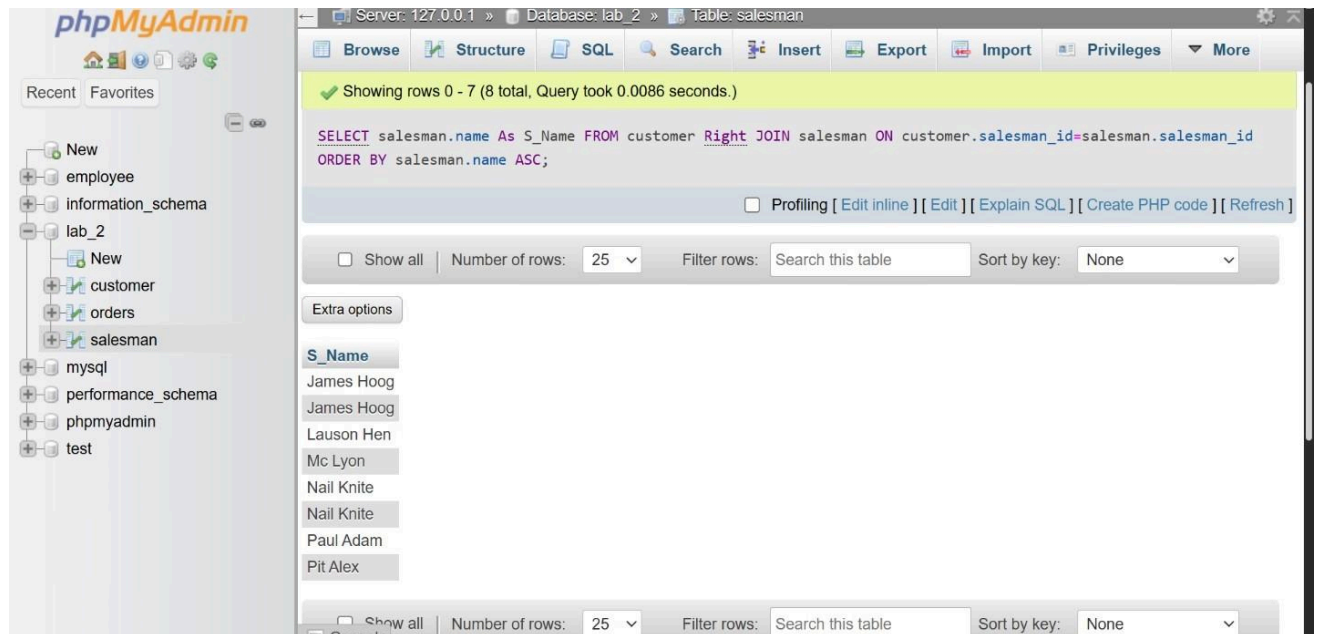
cust_name	city	ord_no	ord_date	purch_amt	name	comission
Graham Zusi	California	70001	2012-10-05	150.5	Nail Knite	0.13
Nick Rimando	New York	70002	2012-10-05	65.26	James Hoog	0.15
Geoff Cameron	Berlin	70003	2012-10-10	2480.4	Lauson Hen	0.12
Geoff Cameron	Berlin	70004	2012-08-17	110.5	Lauson Hen	0.12
Brad Davis	New York	70005	2012-07-27	2400.6	James Hoog	0.15
Graham Zusi	California	70007	2012-09-10	948.5	Nail Knite	0.13
Nick Rimando	New York	70008	2012-09-10	5760	James Hoog	0.15
Brad Guzan	London	70009	2012-09-10	270.65	Pit Alex	0.11
Fabian Johnson	Paris	70010	2012-10-10	1983.43	Mc Lyon	0.14
Jozy Altidor	Moscow	70011	2012-08-17	75.29	Paul Adam	0.13
Julian Green	London	70012	2012-06-27	250.45	Nail Knite	0.13
Console	New York	70013	2012-04-25	3045.6	James Hoog	0.15

Q11. Write a SQL statement to make a list in ascending order for the salesmen who works either for one or more customer or not yet join under any of the customers.

```

SELECT
    s.salesman_id,
    s.name,
    s.city,
    s.commission
FROM
    salesman s
LEFT JOIN
    customer c ON s.salesman_id = c.salesman_id
ORDER BY
    s.salesman_id ASC;

```



Q12. Write a SQL statement to make a list in ascending order for the salesmen who works either for one or more customer or not yet join under any of the customers.

```
SELECT
    s.name AS salesman,
    c.cust_name,
    c.city,
    c.grade,
    o.order_no,
    o.ord_date,
    o.purchase_amt
FROM
    salesman s
LEFT JOIN
    customer c ON s.salesman_id = c.salesman_id
LEFT JOIN
    orders o ON c.customer_id = o.customer_id;
```

The screenshot shows the phpMyAdmin interface with the following details:

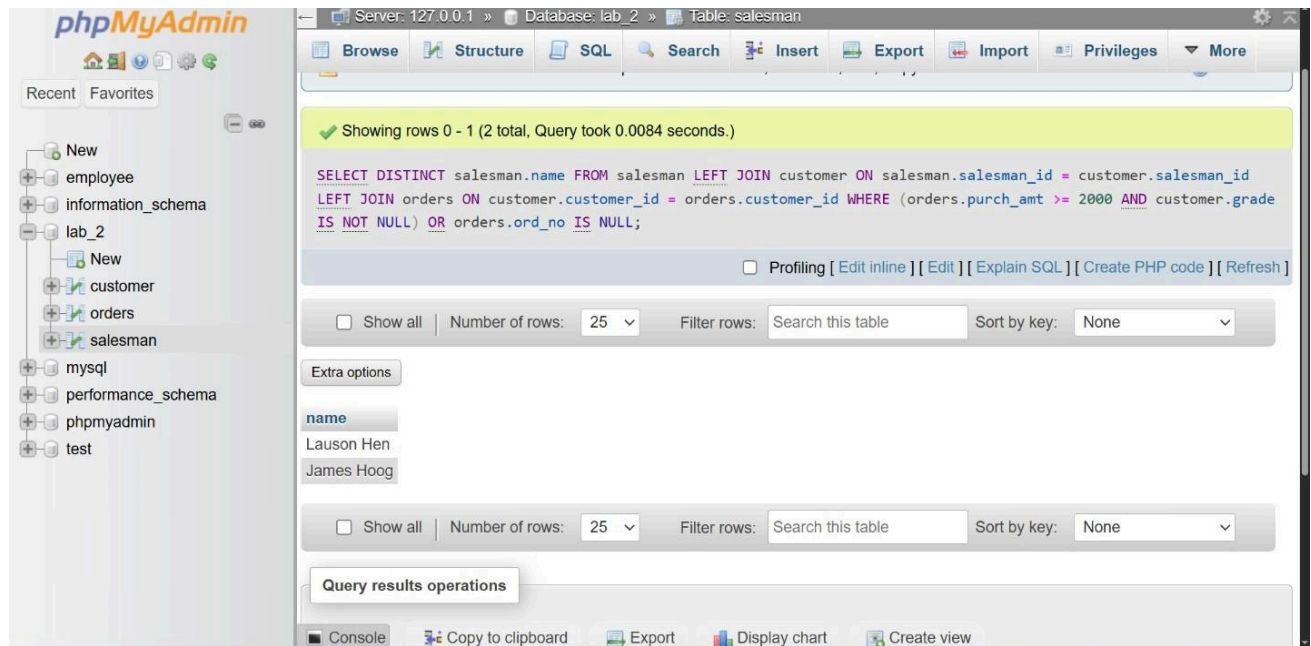
- Server:** 127.0.0.1 » **Database:** lab_2 » **Table:** salesman
- Navigation Panel (Left):**
 - Recent: New, employee, information_schema, lab_2
 - lab_2: New, customer, orders, salesman
 - mysql, performance_schema, phpmyadmin, test
- SQL Query:**

```
SELECT salesman.name AS Salespersons, customer.cust_name, customer.city, orders.ord_no, orders.ord_date, orders.purch_amt FROM salesman LEFT JOIN customer ON salesman.salesman_id=customer.salesman_id LEFT JOIN orders ON customer.customer_id=orders.customer_id;
```
- Query Options:**
 - ☐ 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
- Results Table:**

Salespersons	cust_name	city	ord_no	ord_date	purch_amt
Nail Knite	Graham Zusi	California	70001	2012-10-05	150.5
James Hoog	Nick Rimando	New York	70002	2012-10-05	65.26
Lauson Hen	Geoff Cameron	Berlin	70003	2012-10-10	2480.4
Lauson Hen	Geoff Cameron	Berlin	70004	2012-08-17	110.5
James Hoog	Brad Davis	New York	70005	2012-07-27	2400.6
Nail Knite	Graham Zusi	California	70007	2012-09-10	948.5
James Hoog	Nick Rimando	New York	70008	2012-09-10	5760
Pit Alex	Brad Guzan	London	70009	2012-09-10	270.65
Mc Lyon	Fabian Johnson	Paris	70010	2012-10-10	1983.43
Console	Jozy Altidor	Moscow	70011	2012-08-17	75.29

Q13. Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customer. The customer may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier.

```
SELECT DISTINCT
    s.name AS salesman
FROM
    salesman s
LEFT JOIN
    customer c ON s.salesman_id = c.salesman_id
LEFT JOIN
    orders o ON c.customer_id = o.customer_id
WHERE
    (o.purchase_amt >= 2000 AND c.grade IS NOT NULL)
    OR c.customer_id IS NULL;
```



Q14. Write a SQL statement to make a report with customer name, city, order no. order date, purchase amount for those customers from the existing list who placed one or more orders or which order(s) have been placed by the customer who is not on the list.

```
SELECT
    c.cust_name,
    c.city,
    o.order_no,
    o.ord_date,
    o.purchase_amt
FROM
    orders o
LEFT JOIN
    customer c ON o.customer_id = c.customer_id;
```


The screenshot shows the phpMyAdmin interface. On the left is the database navigation tree with 'lab_2' selected. The main panel displays a SQL query and its results. The query is:

```
SELECT c.cust_name, c.city, o.ord_no, o.ord_date, o.purch_amt FROM orders o LEFT JOIN customer c ON o.customer_id = c.customer_id WHERE c.customer_id IS NOT NULL UNION SELECT NULL AS cust_name, NULL AS city, o.ord_no, o.ord_date, o.purch_amt FROM orders o LEFT JOIN customer c ON o.customer_id = c.customer_id WHERE c.customer_id IS NULL;
```

Below the query, there are controls for 'Show all', 'Number of rows' (set to 25), 'Filter rows' (Search this table), and 'Sort by key' (None). The results are displayed in a table with the following data:

cust_name	city	ord_no	ord_date	purch_amt
Graham Zusi	California	70001	2012-10-05	150.5
Nick Rimando	New York	70002	2012-10-05	65.26
Geoff Cameron	Berlin	70003	2012-10-10	2480.4
Geoff Cameron	Berlin	70004	2012-08-17	110.5
Brad Davis	New York	70005	2012-07-27	2400.6
Graham Zusi	California	70007	2012-09-10	948.5
Nick Rimando	New York	70008	2012-09-10	5760
Brad Guzan	London	70009	2012-09-10	270.65
Fabian Johnson	Paris	70010	2012-10-10	1983.43
James Milner	Moscow	70011	2012-08-17	75.29

Q15. Write a SQL statement to make a report with customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who is neither in the list nor have a grade.

```
SELECT
    c.cust_name,
    c.city,
    o.order_no,
    o.ord_date,
    o.purchase_amt
FROM
    orders o
LEFT JOIN
    customer c ON o.customer_id = c.customer_id
WHERE
    (c.grade IS NOT NULL)
    OR (c.customer_id IS NULL OR c.grade IS NULL);
```


phpMyAdmin

Recent

Favorites

New

employee

information_schema

lab_2

- New
 - customer
 - orders
 - salesman

mysql

performance_schema

phpmyadmin

test

Server: 127.0.0.1 » Database: lab_2

StructureSQLSearchQueryExportImportOperationsPrivilegesMore

Showing rows 0 - 11 (12 total, Query took 0.0088 seconds.)

SELECT c.cust_name, c.city, o.ord_no, o.ord_date, o.purch_amt FROM orders o JOIN customer c ON o.customer_id = c.customer_id WHERE c.grade IS NOT NULL UNION SELECT NULL AS cust_name, NULL AS city, o.ord_no, o.ord_date, o.purch_amt FROM orders o LEFT JOIN customer c ON o.customer_id = c.customer_id WHERE c.customer_id IS NULL OR c.grade IS NULL;

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

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

cust_name	city	ord_no	ord_date	purch_amt
Graham Zusi	California	70001	2012-10-05	150.5
Nick Rimando	New York	70002	2012-10-05	65.26
Geoff Cameron	Berlin	70003	2012-10-10	2480.4
Geoff Cameron	Berlin	70004	2012-08-17	110.5
Brad Davis	New York	70005	2012-07-27	2400.6
Graham Zusi	California	70007	2012-09-10	948.5
Nick Rimando	New York	70008	2012-09-10	5760
Fabian Johnson	Paris	70010	2012-10-10	1983.43
Jamy Altidore	Moscow	70011	2012-08-17	75.29

Console