Institute of Engineering & Technology

Devi Ahilya Vishwavidyalaya, Indore (M.P)

Department of Computer Science & Engineering

DATABASE MANAGEMENT SYSTEM(CER4C4)

Lab Assignment-6

Submitted To:

Mrs. Jyoti Haveliya Mam

CS-Dept

IET-DAVV

Submitted By:

Tanishq Chauhan (21C4184)

CS "B" 2nd Year

Lab Assignment-6

Create three tables salesman, customer and order with following data respectively:

```
mysql> select * from salesman;
+-------------+------------+----------+------------+
| salesman_id | name       | city     | commission |
+-------------+------------+----------+------------+
|        5001 | James Hoog | New York |       0.15 |
|        5002 | Nail Knite | Paris    |       0.13 |
|        5005 | Pit Alex   | London   |       0.11 |
|        5006 | Mc Lyon    | Paris    |       0.14 |
|        5007 | Paul Adam  | Rome     |       0.13 |
|        5003 | Lauson Hen | San Jose |       0.12 |
+-------------+------------+----------+------------+
6 rows in set (0.00 sec)
```

```
mysql> select * from customer;
+-------------+---------------+------------+-------+-------------+
| customer_id | cust_name     | city       | grade | salesman_id |
+-------------+---------------+------------+-------+-------------+
|        3002 | Nick Rimando  | New York   |   100 |        5001 |
|        3007 | Brad Davis    | New York   |   200 |        5001 |
|        3005 | Graham Zusi   | California |   200 |        5002 |
|        3008 | Julian Green  | London     |   300 |        5002 |
|        3004 | Fabian Johnson| Paris      |   300 |        5006 |
|        3009 | Geoff Cameron | Berlin     |   100 |        5003 |
|        3003 | Jozy Altidor  | Moscow     |   200 |        5007 |
|        3001 | Brad Guzan    | London     |   100 |        5005 |
+-------------+---------------+------------+-------+-------------+
8 rows in set (0.00 sec)
```

```
mysql> select * from ord;
+--------+-----------+------------+-------------+-------------+
| ord_no | purch_amt | ord_date   | customer_id | salesman_id |
+--------+-----------+------------+-------------+-------------+
|  70001 |    150.50 | 2012-10-05 |        3005 |        5002 |
|  70009 |    270.65 | 2012-09-10 |        3001 |        5005 |
|  70002 |     65.29 | 2012-10-05 |        3002 |        5001 |
|  70004 |    110.50 | 2012-08-17 |        3009 |        5003 |
|  70007 |    948.50 | 2012-09-10 |        3005 |        5002 |
|  70005 |   2400.60 | 2012-07-27 |        3007 |        5001 |
|  70008 |   5760.00 | 2012-09-10 |        3002 |        5001 |
|  70010 |   1983.43 | 2012-10-10 |        3004 |        5006 |
|  70003 |   2480.40 | 2012-10-10 |        3009 |        5003 |
|  70012 |    250.45 | 2012-06-27 |        3008 |        5002 |
|  70011 |     75.29 | 2012-08-17 |        3003 |        5007 |
|  70013 |   3045.60 | 2012-04-25 |        3002 |        5001 |
+--------+-----------+------------+-------------+-------------+
12 rows in set (0.00 sec)
```

Now solve the following query:-

1. **WAQ to find the salesperson and customer who belongs to same city. Return Salesman, cust_name and city.**

Query:- select salesman.name as “Salesman”,

- ➔ Customer.cust_name, customer.city
- ➔ From salesman, customer
- ➔ Where salesman.city = customer.city;

```
mysql> select salesman.name as "Salesman",
    -> customer.cust_name, customer.city
    -> from salesman,customer
    -> where salesman.city = customer.city;
+------------+----------------+----------+
| Salesman   | cust_name      | city     |
+------------+----------------+----------+
| Pit Alex   | Brad Guzan     | London   |
| James Hoog | Nick Rimando   | New York |
| Mc Lyon    | Fabian Johnson | Paris    |
| Nail Knite | Fabian Johnson | Paris    |
| James Hoog | Brad Davis     | New York |
| Pit Alex   | Julian Green   | London   |
+------------+----------------+----------+
6 rows in set (0.01 sec)
```

2. WAQ to find those orders where order amount exists between 500 and 2000. Return ord_no, purch_amt, cust_name, city.

Query:- select ord.ord_no, ord.purch-amt,

- ➔ customer.cust_name, customer.city
- ➔ from ord, customer
- ➔ where ord.customer_id = customer.customer_id
- ➔ AND ord.purch_amt BETWEEN 500 AND 2000;

```
mysql> select ord.ord_no, ord.purch_amt,
    -> customer.cust_name, customer.city
    -> from ord, customer
    -> where ord.customer_id = customer.customer_id
    -> AND ord.purch_amt BETWEEN 500 AND 2000;
+--------+-----------+---------------+------------+
| ord_no | purch_amt | cust_name     | city       |
+--------+-----------+---------------+------------+
|  70007 |    948.50 | Graham Zusi   | California |
|  70010 |   1983.43 | Fabian Johnson | Paris      |
+--------+-----------+---------------+------------+
2 rows in set (0.00 sec)
```

3. **WAQ to find the salesperson(s) and the customer(s) he handle. Return Customer Name, city, Salesman, commission.**

Query:- select c.cust_name as “Customer Name”

- ➔ c.city, s.name as “Salesman”, s.commission
- ➔ from customer c
- ➔ INNER JOIN salesman s
- ➔ ON c.salesman_id = s.salesman_id;

```
mysql> select c.cust_name as "Customer Name",
    -> c.city, s.name as "Salesman", s.commission
    -> from customer c
    -> INNER JOIN salesman s
    -> ON c.salesman_id = s.salesman_id;
+---------------+------------+------------+------------+
| Customer Name | city       | Salesman   | commission |
+---------------+------------+------------+------------+
| Nick Rimando  | New York   | James Hoog |       0.15 |
| Brad Davis    | New York   | James Hoog |       0.15 |
| Graham Zusi   | California | Nail Knite |       0.13 |
| Julian Green  | London     | Nail Knite |       0.13 |
| Geoff Cameron | Berlin     | Lauson Hen |       0.12 |
| Brad Guzan    | London     | Pit Alex   |       0.11 |
| Fabian Johnson| Paris      | Mc Lyon    |       0.14 |
| Jozy Altidor  | Moscow     | Paul Adam  |       0.13 |
+---------------+------------+------------+------------+
8 rows in set (0.00 sec)
```

4. **WAQ to find those salespersons who received a commission from the company more than 12%. Return Customer Name, customer city, Salesman, commission.**

Query:- select c.cust_name as “Customer Name”,

- ➔ c.city as “Customer City”,
- ➔ s.name as “Salesman”, s.commission
- ➔ from customer c
- ➔ INNER JOIN salesman s
- ➔ ON c.salesman_id = s.salesman_id
- ➔ WHERE s.commission > .12;

```
mysql> select c.cust_name as "Customer Name",
    -> c.city as "Customer City",
    -> s.name as "Salesman", s.commission
    -> from customer c
    -> INNER JOIN salesman s
    -> ON c.salesman_id = s.salesman_id
    -> WHERE s.commission > .12;
+---------------+---------------+------------+------------+
| Customer Name | Customer City | Salesman   | commission |
+---------------+---------------+------------+------------+
| Nick Rimando  | New York      | James Hoog |       0.15 |
| Brad Davis    | New York      | James Hoog |       0.15 |
| Graham Zusi   | California    | Nail Knite |       0.13 |
| Julian Green  | London        | Nail Knite |       0.13 |
| Fabian Johnson| Paris         | Mc Lyon    |       0.14 |
| Jozy Altidor  | Moscow        | Paul Adam  |       0.13 |
+---------------+---------------+------------+------------+
6 rows in set (0.00 sec)
```

5. **WAQ to find those salespersons who 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.**

Query:- select c.cust_name as “Customer Name”,

- ➔ c.city as “Customer City”,
- ➔ s.name as “Salesman”, s.city as “Salesman City”,
- ➔ s.commission from customer c
- ➔ INNER JOIN salesman s
- ➔ ON c.salesman_id = s.salesman_id
- ➔ WHERE s.commission > .12
- ➔ AND c.city <> s.city;

```
mysql> select c.cust_name as "Customer Name",
    -> c.city as "Customer City",
    -> s.name as "Salesman", s.city as "Salesman City",
    -> s.commission from customer c
    -> INNER JOIN salesman s
    -> ON c.salesman_id = s.salesman_id
    -> WHERE s.commission > .12
    -> AND c.city <> s.city;
+---------------+---------------+------------+---------------+------------+
| Customer Name | Customer City | Salesman   | Salesman City | commission |
+---------------+---------------+------------+---------------+------------+
| Graham Zusi   | California    | Nail Knite | Paris         |       0.13 |
| Julian Green  | London        | Nail Knite | Paris         |       0.13 |
| Jozy Altidor  | Moscow        | Paul Adam  | Rome          |       0.13 |
+---------------+---------------+------------+---------------+------------+
3 rows in set (0.00 sec)
```

6. **WAQ to find the details of an order. Return ord_no, ord_date, purch_amt, Customer Name, grade, Salesman, commission.**

Query:- select o.ord_no, o.ord_date, o.purch_amt,

- ➔ c.cust_name as “Customer Name”, c.grade,
- ➔ s.name as “Salesman”, s.commission
- ➔ from ord o
- ➔ INNER JOIN customer c
- ➔ ON o.customer_id = c.customer_id
- ➔ INNER JOIN salesman s
- ➔ ON o.salesman_id = s.salesman_id;

```
mysql> select o.ord_no, o.ord_date, o.purch_amt,
    -> c.cust_name as "Customer Name", c.grade,
    -> s.name as "Salesman", s.commission
    -> from ord o
    -> INNER JOIN customer c
    -> ON o.customer_id = c.customer_id
    -> INNER JOIN salesman s
    -> ON o.salesman_id = s.salesman_id;
+--------+------------+-----------+---------------+-------+------------+------------+
| ord_no | ord_date   | purch_amt | Customer Name | grade | Salesman   | commission |
+--------+------------+-----------+---------------+-------+------------+------------+
|  70002 | 2012-10-05 |     65.29 | Nick Rimando  |   100 | James Hoog |       0.15 |
|  70005 | 2012-07-27 |   2400.60 | Brad Davis    |   200 | James Hoog |       0.15 |
|  70008 | 2012-09-10 |   5760.00 | Nick Rimando  |   100 | James Hoog |       0.15 |
|  70013 | 2012-04-25 |   3045.60 | Nick Rimando  |   100 | James Hoog |       0.15 |
|  70001 | 2012-10-05 |    150.50 | Graham Zusi   |   200 | Nail Knite |       0.13 |
|  70007 | 2012-09-10 |    948.50 | Graham Zusi   |   200 | Nail Knite |       0.13 |
|  70012 | 2012-06-27 |    250.45 | Julian Green  |   300 | Nail Knite |       0.13 |
|  70003 | 2012-10-10 |   2480.40 | Geoff Cameron |   100 | Lauson Hen |       0.12 |
|  70004 | 2012-08-17 |    110.50 | Geoff Cameron |   100 | Lauson Hen |       0.12 |
|  70009 | 2012-09-10 |    270.65 | Brad Guzan    |   100 | 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 |
+--------+------------+-----------+---------------+-------+------------+------------+
12 rows in set (0.00 sec)
```

7. **Write a statement to make a join on the tables salesman, customer and orders in such a form that the same column of each table will appear once and only the relational rows will come.**

Query:- select * from ord

➔ NATURAL JOIN customer
➔ NATURAL JOIN salesman;

```
mysql> select * from ord
    -> NATURAL JOIN customer
    -> NATURAL JOIN salesman;
+------------+---------+-------------+--------+-----------+------------+---------------+-------+-----------+------------+
| salesman_id | city    | customer_id | ord_no | purch_amt | ord_date   | cust_name     | grade | name      | commission |
+------------+---------+-------------+--------+-----------+------------+---------------+-------+-----------+------------+
|       5005 | London  |        3001 |  70009 |    270.65 | 2012-09-10 | Brad Guzan    |   100 | Pit Alex  |       0.11 |
|       5001 | New York |       3002 |  70002 |     65.29 | 2012-10-05 | Nick Rimando  |   100 | James Hoog |      0.15 |
|       5001 | New York |       3002 |  70008 |   5760.00 | 2012-09-10 | Nick Rimando  |   100 | James Hoog |      0.15 |
|       5001 | New York |       3002 |  70013 |   3045.60 | 2012-04-25 | Nick Rimando  |   100 | James Hoog |      0.15 |
|       5006 | Paris   |        3004 |  70010 |   1983.43 | 2012-10-10 | Fabian Johnson |  300 | Mc Lyon   |       0.14 |
|       5001 | New York |       3007 |  70005 |   2400.60 | 2012-07-27 | Brad Davis    |   200 | James Hoog |      0.15 |
+------------+---------+-------------+--------+-----------+------------+---------------+-------+-----------+------------+
6 rows in set (0.00 sec)
```

8. **WAQ to display the cust_name, customer city, grade, Salesman, salesman city. The result should be ordered by ascending on customer_id.**

Query:- select c.cust_name, c.city as “Customer City”,

➔c.grade, s.name as “Salesman”, s.city as “Salesman City”

➔ from customer c

➔ LEFT JOIN salesman s

➔ ON c.salesman_id = s.salesman_id

➔ ORDER BY c.customer_id;

```
mysql> select c.cust_name, c.city as "Customer City",
    -> c.grade, s.name as "Salesman", s.city as "Salesman City"
    -> from customer c
    -> LEFT JOIN salesman s
    -> ON c.salesman_id = s.salesman_id
    -> ORDER BY c.customer_id;
+----------------+---------------+-------+------------+---------------+
| cust_name      | Customer City | grade | Salesman   | Salesman City |
+----------------+---------------+-------+------------+---------------+
| Brad Guzan     | London        |   100 | 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      |
+----------------+---------------+-------+------------+---------------+
8 rows in set (0.00 sec)
```

9. WAQ 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.

Query:- select c.cust_name, c.city as “Customer City”,

➔c.grade, s.name as “Salesman”, s.city as “Salesman city”

➔ from customer c

➔ LEFT OUTER JOIN salesman s

➔ ON c.salesman_id = s.salesman_id

➔ WHERE c.grade < 300

➔ ORDER BY c.customer_id;

```
mysql> select c.cust_name, c.city as "Customer City",
    -> c.grade, s.name as "Salesman", s.city as "Salesman City"
    -> from customer c
    -> LEFT OUTER JOIN salesman s
    -> ON c.salesman_id = s.salesman_id
    -> WHERE c.grade < 300
    -> ORDER BY c.customer_id;
+---------------+---------------+-------+------------+---------------+
| cust_name     | Customer City | grade | Salesman   | Salesman City |
+---------------+---------------+-------+------------+---------------+
| Brad Guzan    | London        |   100 | Pit Alex   | London        |
| 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      |
+---------------+---------------+-------+------------+---------------+
6 rows in set (0.00 sec)
```

10. Write a 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.

Query:- select c.cust_name as “Customer Name”, c.city,

➔ o.ord_no as “Order Number”, o.ord_date as “Order Date”,

➔ o.purch_amt as “Order Amount”

➔ from customer c

➔ LEFT OUTER JOIN ord o

➔ ON c.customer_id = o.customer_id

➔ ORDER BY o.ord_date;

```
mysql> select c.cust_name as "Customer Name", c.city,
    -> o.ord_no as "Order Number", o.ord_date as "Order Date",
    -> o.purch_amt as "Order Amount"
    -> from customer c
    -> LEFT OUTER JOIN ord o
    -> ON c.customer_id = o.customer_id
    -> ORDER BY o.ord_date;
+---------------+------------+--------------+------------+--------------+
| Customer Name | city       | Order Number | Order Date | Order Amount |
+---------------+------------+--------------+------------+--------------+
| Nick Rimando  | New York   |        70013 | 2012-04-25 |      3045.60 |
| Julian Green  | London     |        70012 | 2012-06-27 |       250.45 |
| Brad Davis    | New York   |        70005 | 2012-07-27 |      2400.60 |
| Jozy Altidor  | Moscow     |        70011 | 2012-08-17 |        75.29 |
| Geoff Cameron | Berlin     |        70004 | 2012-08-17 |       110.50 |
| Brad Guzan    | London     |        70009 | 2012-09-10 |       270.65 |
| Nick Rimando  | New York   |        70008 | 2012-09-10 |      5760.00 |
| Graham Zusi   | California |        70007 | 2012-09-10 |       948.50 |
| Nick Rimando  | New York   |        70002 | 2012-10-05 |        65.29 |
| Graham Zusi   | California |        70001 | 2012-10-05 |       150.50 |
| Fabian Johnson| Paris      |        70010 | 2012-10-10 |      1983.43 |
| Geoff Cameron | Berlin     |        70003 | 2012-10-10 |      2480.40 |
+---------------+------------+--------------+------------+--------------+
12 rows in set (0.00 sec)
```

11. Write a 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.

Query:- (select c.cust_name as "Customer Name", c.city, o.ord_no as "Order No.", o.ord_date as "Order Date", o.purch_amt as "Purchase Amount" from customer c

➔ LEFT OUTER JOIN ord o ON c.customer_id = o.customer_id)

➔ UNION

➔ (select c.cust_name as "Customer Name", c.city, o.ord_no as "Order No.", o.ord_date as "Order Date", o.purch_amt as "Purchase Amount" from customer c

➔ RIGHT OUTER JOIN ord o ON c.customer_id = o.customer_id);

```
mysql> (select c.cust_name as "Customer Name", c.city, o.ord_no as "Order No.", o.ord_date as "Order Date", o.purch_amt as "Purchase Amount" from customer c
    -> LEFT OUTER JOIN ord o ON c.customer_id = o.customer_id)
    -> UNION
    -> (select c.cust_name as "Customer Name", c.city, o.ord_no as "Order No.", o.ord_date as "Order Date", o.purch_amt as "Purchase Amount" from customer c
    -> RIGHT OUTER JOIN ord o ON c.customer_id = o.customer_id);
+---------------+-----------+----------+-----------+---------------+
| Customer Name | city      | Order No. | Order Date | Purchase Amount |
+---------------+-----------+----------+-----------+---------------+
| Brad Guzan    | London    |    70009 | 2012-09-10 |        270.65 |
| Nick Rimando  | New York  |    70002 | 2012-10-05 |         65.29 |
| Nick Rimando  | New York  |    70008 | 2012-09-10 |       5760.00 |
| Nick Rimando  | New York  |    70013 | 2012-04-25 |       3045.60 |
| Jozy Altidor  | Moscow    |    70011 | 2012-08-17 |         75.29 |
| Fabian Johnson | Paris    |    70010 | 2012-10-10 |       1983.43 |
| Graham Zusi   | California |    70001 | 2012-10-05 |        150.50 |
| Graham Zusi   | California |    70007 | 2012-09-10 |        948.50 |
| Brad Davis    | New York  |    70005 | 2012-07-27 |       2400.60 |
| Julian Green  | London    |    70012 | 2012-06-27 |        250.45 |
| Geoff Cameron | Berlin    |    70003 | 2012-10-10 |       2480.40 |
| Geoff Cameron | Berlin    |    70004 | 2012-08-17 |        110.50 |
+---------------+-----------+----------+-----------+---------------+
12 rows in set (0.01 sec)
```