### Assignment-20

### Where Clause

## Kaviya C

**1.** From the following table, write a SQL query to find the details of the customers who have a gradevalue above 100. Return customer\_id, cust\_name, city, grade, and salesman\_id.

## Sample table: customer

| salesman_id |        | cust_name      |   |            |   |     |   |
|-------------|--------|----------------|---|------------|---|-----|---|
|             |        |                |   |            |   |     |   |
| F 0 0 1     | 3002   | Nick Rimando   |   | New York   |   | 100 | 1 |
| 5001        | 3007   | Brad Davis     | I | New York   | 1 | 200 | 1 |
| 5001        | 3005   | Graham Zusi    | I | California | 1 | 200 | 1 |
| 5002        | 3008 I | Julian Green   | ı | London     | ı | 300 | 1 |
| 5002        | ·      |                |   |            |   |     | • |
| 5006        | 3004   | Fabian Johnson |   | Paris      |   | 300 |   |
|             | 3009   | Geoff Cameron  | ١ | Berlin     | 1 | 100 | 1 |
| 5003        | 3003   | Jozy Altidor   |   | Moscow     | 1 | 200 | 1 |
| 5007        | 3001   | Brad Guzan     | ı | London     | 1 |     | 1 |
| 5005        | J001   | Didd Guzan     | ı | HOHAOH     | I |     | I |

```
173
174
            create database SalesCust;
    175 •
            use SalesCust;
    176
    177 •
             create table Customer
    178
    179
            customer id int,
    180
            cust_name varchar(20),
            city varchar(15),
    181
    182
            grade int,
    183
            salesman_id int
    184
    185
    186 •
            desc customer;
            Select * from customer;
    187 •
    188
            insert into customer values(3002, 'Nick Rimando', 'New York',100,5001);
    189 •
            insert into customer values(3007, 'Brad Davis', 'New York', 200, 5001);
    190 •
            insert into customer values(3005, 'Graham Zusi', 'California ',200,5002);
    191 •
            insert into customer values(3008, 'Julian Green', 'London ',300,5002);
    192 •
            insert into customer values(3004, 'Fabian Johnson', 'Paris', 300, 5006);
    193 •
    194 •
            insert into customer values(3009, 'Geoff Cameron', 'Berlin', 100, 5003);
            insert into customer values(3003, 'Jozy Altidor', 'Moscow', 200, 5007);
    195 •
            insert into customer values(3001, 'Brad Guzan', 'London',0,5005);
    196
```

```
mysql> use salescust;
Database changed
mysql> show tables;
 Tables in salescust
 customer
 row in set (0.12 sec)
mysql> desc customer;
                             | Null | Key | Default | Extra
              Type
 customer_id
                int
                              YES
                                            NULL
                varchar(20)
                              YES
                                            NULL
 cust_name
                              YES
 city
                varchar(15)
                                            NULL
                int
                               YES
                                            NULL
 grade
                              YES
 salesman_id
                                            NULL
 rows in set (0.00 sec)
```

| nysql> select * | * from customer;<br> | <b></b>    | +     | ++          |  |  |
|-----------------|----------------------|------------|-------|-------------|--|--|
| 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     | 0     | 5005        |  |  |
| +               |                      |            |       |             |  |  |

# Query: ANS:

| mysql> select * from customer where grade>100; |   |   |   |  |  |  |
|--|---|---|---|--|--|--|
| customer_id                                    | cust_name   | city<br>  | grade                                   | salesman_id                                  |  |  |
| 3007<br>3005<br>3008<br>3004<br>3003           | Brad Davis<br>Graham Zusi<br>Julian Green<br>Fabian Johnson<br>Jozy Altidor | New York<br>California<br>London<br>Paris<br>Moscow | 200  <br>200  <br>300  <br>300  <br>200 | 5001  <br>5002  <br>5002  <br>5006  <br>5007 |  |  |
| 5 rows in set (0.00 sec)                       |   |   |   |  |  |  |

**2.** From the following table, write a SQL query to find all the customers in 'New York' city who have a grade value above 100. Return customer\_id, cust\_name, city, grade, and salesman\_id.

## Sample table: customer

| custome:<br>salesman | _    | cust_name      |     | city       | gra | ade |   |
|----------------------|------|----------------|-----|------------|-----|-----|---|
|                      | +    |                | -+- |            | -+  |     | + |
| <br>                 | 3002 | Nick Rimando   | I   | New York   | I   | 100 | I |
| 5001                 | 3007 | Brad Davis     | I   | New York   |     | 200 | I |
| 5002                 | 3005 | Graham Zusi    | I   | California |     | 200 | I |
| 5002                 | 3008 | Julian Green   | 1   | London     | 1   | 300 | I |
| 5002                 | 3004 | Fabian Johnson |     | Paris      | l   | 300 | l |

```
3009 | Geoff Cameron | Berlin | 100 |
5003
3003 | Jozy Altidor | Moscow | 200 |
5007
3001 | Brad Guzan | London | |
```

Query:

**3.** From the following table, write a SQL query to find the customers who belong to either the city 'New York' or have a grade above 100. Return customer\_id, cust\_name, city, grade, and salesman\_id.

### Sample table: 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 I
5002
      3004 | Fabian Johnson | Paris |
                                   300 |
5006
      3009 | Geoff Cameron | Berlin | 100 |
5003
      3003 | Jozy Altidor | Moscow |
                                   200 |
5007
      3001 | Brad Guzan | London |
5005
```

```
nysql> select * from customer where city='New York'
   -> OR
   -> grade>100;
                              city
 customer_id | cust_name
                                             grade | salesman_id
                                New York
        3002
               Nick Rimando
                                                              5001
        3007
               Brad Davis
                                New York
                                                200
                                                              5001
                                California
        3005
               Graham Zusi
                                                200
                                                              5002
               Julian Green
                                London
                                                              5002
        3008
                                                300
               Fabian Johnson
                                Paris
        3004
                                                 300
                                                              5006
        3003
             | Jozy Altidor
                                                200
                                                              5007
                               Moscow
 rows in set (0.00 sec)
```

**4.** From the following table, write a SQL query to find the customers who belong to either the city 'New York' or not have a grade above 100. Return customer\_id, cust\_name, city, grade, and salesman\_id.

Sample table: customer

| salesman | n_id | cust_name      |    |            |   |     |   |
|----------|------|----------------|----|------------|---|-----|---|
|          |      |                | Т. |            |   |     |   |
| 5001     | 3002 | Nick Rimando   |    | New York   |   | 100 |   |
|          | 3007 | Brad Davis     | I  | New York   | 1 | 200 | 1 |
| 5001     | 3005 | Graham Zusi    | I  | California | 1 | 200 |   |
| 5002     | 3008 | Julian Green   | I  | London     |   | 300 | I |
| 5002     | 3004 | Fabian Johnson | I  | Paris      | 1 | 300 | 1 |
| 5006     | 3009 | Geoff Cameron  | I  | Berlin     | 1 | 100 | 1 |
| 5003     | 3003 | Jozy Altidor   | I  | Moscow     | I | 200 |   |
| 5007     | 3001 | Brad Guzan     | ı  | London     |   |     |   |
| 5005     | ·    |                |    |            |   |     |   |

```
mysql> select * from customer where city='New York'
   -> OR NOT
   -> grade>100;
 customer_id | cust_name | city | grade | salesman_id
        3002 | Nick Rimando | New York |
                                           100
                                                        5001
        3007 | Brad Davis | New York |
                                           200
                                                        5001
        3009 | Geoff Cameron | Berlin
                                           100
                                                        5003
        3001 | Brad Guzan | London
                                             0
                                                        5005
 rows in set (0.00 sec)
```

**5.** From the following table, write a SQL query to find those customers who belong to neither the 'New York' city nor their grade value exceeds 100. Return customer\_id, cust\_name, city, grade, and salesman\_id.

### Sample table: customer

| salesmar | n_id   | cust_name      |     |            |    |     |    |
|----------|--------|----------------|-----|------------|----|-----|----|
|          | +-     |                | -+- |            | -+ |     | -+ |
| 5001     | 3002   | Nick Rimando   |     | New York   |    | 100 |    |
|          | 3007   | Brad Davis     |     | New York   | 1  | 200 |    |
| 5001     | 3005   | Graham Zusi    |     | California | 1  | 200 |    |
| 5002     | 3008   | Julian Green   | ı   | London     | ı  | 300 | 1  |
| 5002     | 3000   | oditan orcen   | ı   | попаоп     | ı  | 300 | ı  |
| F006     | 3004   | Fabian Johnson |     | Paris      |    | 300 | 1  |
| 5006     | 3009   | Geoff Cameron  |     | Berlin     | 1  | 100 | 1  |
| 5003     | 2002 1 | Togr. 71+idom  |     | Moggan     | 1  | 200 | 1  |
| 5007     | 3003   | Jozy Altidor   | I   | MOSCOW     | 1  | 200 | I  |
| 5005     | 3001   | Brad Guzan     |     | London     | 1  |     |    |
| 5005     |        |                |     |            |    |     |    |

**6.** From the following table, write a SQL query to find details of all order excluding combination of ord\_date equal to '2012-09-10' and salesman\_id higher than 5005 or purch\_amt greater than 1000. Return ord\_no, purch\_amt, ord\_date, customer\_id and salesman\_id.

### Sample table : orders

| ord_no | purch_amt | ord_date   | customer_id | salesman_id |
|--------|-----------|------------|-------------|-------------|
| 70001  | 150.5     | 2012-10-05 | 3005        | 5002        |
| 70009  | 270.65    | 2012-09-10 | 3001        | 5005        |
| 70002  | 65.26     | 2012-10-05 | 3002        | 5001        |
| 70004  | 110.5     | 2012-08-17 | 3009        | 5003        |
| 70007  | 948.5     | 2012-09-10 | 3005        | 5002        |
| 70005  | 2400.6    | 2012-07-27 | 3007        | 5001        |
| 70008  | 5760      | 2012-09-10 | 3002        | 5001        |
| 70010  | 1983.43   | 2012-10-10 | 3004        | 5006        |
| 70003  | 2480.4    | 2012-10-10 | 3009        | 5003        |
| 70012  | 250.45    | 2012-06-27 | 3008        | 5002        |
| 70011  | 75.29     | 2012-08-17 | 3003        | 5007        |
| 70013  | 3045.6    | 2012-04-25 | 3002        | 5001        |

```
create table Orders
 212 •
       \Theta (
 213
         ord no int,
 214
         purch amt double,
 215
 216
         ord date date,
         customer id int,
 217
         salesman id int
 218
 219
 220
         insert into orders values(7001,150.5,'2012-10-05', 3005,5002);
 221 •
          insert into orders values(7009,270.65,'2012-09-10', 3001,5005);
 222 •
          insert into orders values(7002,65.26,'2012-10-05', 3002,5001);
 223 •
         insert into orders values(7004,110.5,'2012-08-17', 3009,5003);
 224 •
 225 •
          insert into orders values(7007,948.5,'2012-09-10', 3005,5002);
         insert into orders values(7005,2400.6,'2012-07-27', 3075,5001);
 226
 227 •
          insert into orders values(7008,5760,'2012-09-10', 3002,5001);
          insert into orders values(7010,1983.43,'2012-10-10', 3004,5006);
 228
 229 •
         insert into orders values(7003,2480.4,'2012-10-10', 3009,5003);
 230 •
         insert into orders values(7012,250.45,'2012-06-27', 3008,5002);
         insert into orders values(7011,75.29,'2012-08-17', 3003,5007);
 231 •
 232 •
         insert into orders values(7013,3045.6,'2012-04-25', 3002,5001);
mysql> select * from orders;
                                   customer_id | salesman_id |
 ord_no | purch_amt | ord_date
               150.5 | 2012-10-05 |
                                              3005
    7009
               270.65 | 2012-09-10
                                             3001
                                                             5005
    7002
               65.26
                        2012-10-05
                                              3002
                                                             5001
               110.5
                        2012-08-17
                                              3009
    7004
                                                             5003
    7007
               948.5
                        2012-09-10
                                              3005
                                                             5002
    7005
               2400.6
                        2012-07-27
                                              3075
                                                              5001
    7008
                 5760
                        2012-09-10
                                              3002
                                                              5001
              1983.43
                        2012-10-10
                                              3004
                                                              5006
    7010
               2480.4
    7003
                        2012-10-10
                                              3009
                                                              5003
    7012
               250.45
                        2012-06-27
                                              3008
                                                             5002
                      2012-06-27
2012-08-17
                75.29
                                                             5007
    7011
                                              3003
               3045.6 | 2012-04-25 |
                                              3002
                                                             5001
    7013
12 rows in set (0.00 sec)
```

## Query: ANS:

```
mysql> select * from orders
    -> where not((ord_date='2012-09-10' AND salesman_id >5005)
    -> OR
    -> purch_amt >1000.00);
 ord_no | purch_amt | ord_date | customer_id | salesman_id
              150.5 | 2012-10-05 |
270.65 | 2012-09-10 |
    7001
                                             3005
                                                            5002
    7009
                                             3001
                                                            5005
    7002
               65.26 | 2012-10-05 |
                                             3002
                                                            5001
    7004
               110.5 | 2012-08-17 |
                                             3009
                                                            5003
    7007
               948.5 | 2012-09-10 |
                                             3005
                                                            5002
              250.45
    7012
                        2012-06-27
                                             3008
                                                            5002
                                                            5007
    7011
               75.29 | 2012-08-17 |
                                             3003
  rows in set (0.07 sec)
```

**7.** From the following table, write a SQL query to find the details of those salespeople whose commissions range from 0.10 to 0.12. Return salesman\_id, name, city, and commission.

### Sample table: salesman

| salesman_id |                              | -        |      |
|-------------|------------------------------|----------|------|
| 5001        | James Hoog  <br>  Nail Knite | New York | 0.15 |
| 5005        | Pit Alex                     | London   | 0.11 |
| 5006        | Mc Lyon                      | Paris    | 0.14 |
| 5007        | Paul Adam                    | Rome     | 0.13 |
| 5003        | l Lauson Hen l               | San Jose | 0.12 |

```
196
197 •
        create table salesman
198
        salesman id int,
199
200
        name varchar(15),
201
        city varchar(12),
        commission float
202
203
204
205 •
        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);
207 •
208 •
        insert into salesman values(5006, 'Mc Lyon', 'Paris', 0.14);
        insert into salesman values(5003, 'Lauson Hen', 'San Jose', 0.12);
209 •
        insert into salesman values(5007, 'Paul Adam', 'Rome', 0.13);
210 •
211
```

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

## **Query:**

#### Ans:

**8.** From the following table, write a SQL query to find details of all order where purchase amount less than 200 or excluding combination of order date greater

than or equal to '2012-02-10' and customer ID less than 3009. Return ord\_no, purch\_amt, ord\_date, customer\_id and salesman\_id.

Query:

```
mysql> select * from orders WHERE
   -> (purch amt<200
   -> OR NOT
   -> (ord_date>='2012-02-10'
    -> AND
      customer_id<3009
   -> )
   -> )
 ord_no | purch_amt | ord_date
                                    | customer_id | salesman_id
               150.5 | 2012-10-05 |
65.26 | 2012-10-05 |
   7001
                                              3005
                                                             5002
   7002
                                              3002
                                                             5001
   7004
               110.5 | 2012-08-17
                                              3009
                                                             5003
   7005
              2400.6
                        2012-07-27
                                              3075
                                                             5001
   7003
              2480.4
                        2012-10-10
                                              3009
                                                             5003
   7011
               75.29
                        2012-08-17
                                              3003
                                                             5007
 rows in set (0.00 sec)
```

**9.** From the following table, write a SQL query to find all orders subject to following conditions. Exclude combination of order date equal to '2012-08-17' or customer ID higher than 3005 and purchase amount less than 1000.

### Sample table: orders

| ord_no | purch_amt | ord_date   | customer_id | salesman_id |
|--------|-----------|------------|-------------|-------------|
|        |           |            |             |             |
| 70001  | 150.5     | 2012-10-05 | 3005        | 5002        |
| 70009  | 270.65    | 2012-09-10 | 3001        | 5005        |
| 70002  | 65.26     | 2012-10-05 | 3002        | 5001        |
| 70004  | 110.5     | 2012-08-17 | 3009        | 5003        |
| 70007  | 948.5     | 2012-09-10 | 3005        | 5002        |
| 70005  | 2400.6    | 2012-07-27 | 3007        | 5001        |
| 70008  | 5760      | 2012-09-10 | 3002        | 5001        |
| 70010  | 1983.43   | 2012-10-10 | 3004        | 5006        |
| 70003  | 2480.4    | 2012-10-10 | 3009        | 5003        |
| 70012  | 250.45    | 2012-06-27 | 3008        | 5002        |
| 70011  | 75.29     | 2012-08-17 | 3003        | 5007        |
| 70013  | 3045.6    | 2012-04-25 | 3002        | 5001        |

```
mysql> select * from orders where
    -> NOT
    -> ((ord_date='2012-08-17'
    -> OR
    -> customer_id>3005)
    -> AND
    -> purch_amt<1000);</pre>
  ord_no | purch_amt | ord_date | customer_id | salesman_id
               150.5 | 2012-10-05 |
270.65 | 2012-09-10 |
65.26 | 2012-10-05 |
    7001
                                                3005
                                                                5002
    7009
                                                3001
                                                                5005
                                                                5001
    7002
                                                3002
                948.5 | 2012-09-10 |
                                                3005
    7007
                                                                5002
    7005
               2400.6 | 2012-07-27 |
                                                3075
                                                                5001
    7008
                  5760 | 2012-09-10 |
                                                3002
                                                                5001
              1983.43
                         2012-10-10
                                                3004
                                                                5006
    7010
               2480.4
                         2012-10-10
                                                3009
    7003
                                                                5003
    7013
               3045.6 | 2012-04-25
                                                3002
                                                                5001
9 rows in set (0.00 sec)
```

**10.** Write a SQL query to display order number, purchase amount, achieved, the unachieved percentage for those order which exceeds the 50% of the target value of 6000.

### Sample table: orders

| ord_no | purch_amt | ord_date   | customer_id | salesman_id |
|--------|-----------|------------|-------------|-------------|
|        |           |            |             |             |
| 70001  | 150.5     | 2012-10-05 | 3005        | 5002        |
| 70009  | 270.65    | 2012-09-10 | 3001        | 5005        |
| 70002  | 65.26     | 2012-10-05 | 3002        | 5001        |
| 70004  | 110.5     | 2012-08-17 | 3009        | 5003        |
| 70007  | 948.5     | 2012-09-10 | 3005        | 5002        |
| 70005  | 2400.6    | 2012-07-27 | 3007        | 5001        |
| 70008  | 5760      | 2012-09-10 | 3002        | 5001        |
| 70010  | 1983.43   | 2012-10-10 | 3004        | 5006        |
| 70003  | 2480.4    | 2012-10-10 | 3009        | 5003        |
| 70012  | 250.45    | 2012-06-27 | 3008        | 5002        |
| 70011  | 75.29     | 2012-08-17 | 3003        | 5007        |
| 70013  | 3045.6    | 2012-04-25 | 3002        | 5001        |

**11.** From the following table, write a SQL query to find the details of all employees whose last name is 'Dosni' or 'Mardy'. Return emp\_idno, emp\_fname, emp\_lname, and emp\_dept.

Sample table: emp\_details

| PT |
|----|
|    |
| 57 |
| 63 |
| 57 |
| 63 |
| 47 |
| 47 |
| 57 |
| 47 |
| 57 |
| 27 |
| 63 |
| 27 |
| 57 |
|    |

```
create table emp details
8 •
9
       (
       EMP IDNO int,
0
1
       EMP FNAME varchar(15),
       EMP_LNAME varchar(10),
2
3
       EMP DEPT int
4
5
       insert into emp details values(127323, 'Michale', 'Robbin',57);
7 •
       insert into emp details values(526689, 'Carlos', 'Snares',63);
        insert into emp details values(843795, 'Enric', 'Dosio',57);
         insert into emp_details values(328717, 'John', 'Snares',63);
9 •
          insert into emp_details values(44527, 'Josheph', 'Dosni',47);
           insert into emp_details values(659831, 'Zanifer', 'Emily',47);
1 •
            insert into emp details values(847674, 'Kuleswar', 'Sitaraman', 57);
             insert into emp_details values(748681, 'Henrey', 'Gabriel',47);
3 •
              insert into emp details values(55935, 'Alex', 'Manuel',57);
               insert into emp_details values(539569, 'George', 'Mardy', 27);
5 •
                insert into emp_details values(733843, 'Mario', 'Saule',63);
7 •
                 insert into emp details values(631548, 'Alan', 'Snappy', 27);
8 •
                  insert into emp_details values(839139, 'Maria', 'Foster',57);
a
mysql> select * from emp_details;
  EMP IDNO | EMP FNAME | EMP LNAME | EMP DEPT
    127323
               Michale
                             Robbin
                                                   57
    526689
               Carlos
                             Snares
                                                  63
    843795
               Enric
                             Dosio
                                                  57
     328717
               John
                             Snares
                                                  63
      44527
               Josheph
                             Dosni
                                                  47
    659831
               Zanifer
                             Emily
                                                  47
    847674
               Kuleswar
                             Sitaraman
                                                  57
     748681
               Henrey
                             Gabriel
                                                  47
                                                  57
      55935
               Alex
                             Manuel
    539569
                                                  27
               George
                             Mardy
     733843
               Mario
                             Saule
                                                  63
     631548
               Alan
                                                   27
                             Snappy
    839139
               Maria
                             Foster
                                                  57
```

## **Query:**

13 rows in set (0.00 sec)

**12.** From the following table, write a SQL query to find the employees who works at depart 47 or 63. Return emp\_idno, emp\_fname, emp\_lname, and emp\_dept.

Sample table: emp\_details

| EMP_IDNO E | EMP_FNAME | EMP_LNAME | EMP_DEPT |
|------------|-----------|-----------|----------|
|            |           |           |          |
| 127323     | Michale   | Robbin    | 57       |
| 526689     | Carlos    | Snares    | 63       |
| 843795     | Enric     | Dosio     | 57       |
| 328717     | Jhon      | Snares    | 63       |
| 444527     | Joseph    | Dosni     | 47       |
| 659831     | Zanifer   | Emily     | 47       |
| 847674     | Kuleswar  | Sitaraman | 57       |
| 748681     | Henrey    | Gabriel   | 47       |
| 555935     | Alex      | Manuel    | 57       |
| 539569     | George    | Mardy     | 27       |
| 733843     | Mario     | Saule     | 63       |
| 631548     | Alan      | Snappy    | 27       |
| 839139     | Maria     | Foster    | 57       |

```
mysql> select * from emp_details where
    -> emp_dept=47
    -> OR
    -> emp_dept=63;
 EMP_IDNO | EMP_FNAME | EMP_LNAME | EMP_DEPT
   526689 | Carlos
328717 | John
                        Snares
                                             63
                         Snares
                                             63
    44527
             Josheph
                         Dosni
                                             47
    659831 | Zanifer
                                             47
                         Emily
                         Gabriel
    748681 | Henrey
                                             47
    733843 | Mario
                                             63
                        Saule
 rows in set (0.00 sec)
```