MANVI JALAWADIYA
L_018

PRACTICAL NO. 1

AIM: DDL operations on Relational Schema

```
create table salesman(
   -> salesman_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
   -> name VARCHAR(100) NOT NULL,
   -> city VARCHAR(100) NOT NULL,
   -> commision DECIMAL(10,2)
   -> );
```

```
mysql> desc salesman;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| salesman_id | int(11)       | NO   | PRI | NULL    | auto_increment |
| name        | varchar(100)  | NO   |     | NULL    |                |
| city        | varchar(100)  | NO   |     | NULL    |                |
| commision   | decimal(10,2) | YES  |     | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

```
create table customer(
   -> customer_id INT AUTO_INCREMENT PRIMARY KEY,
   -> customer_name VARCHAR(100) NOT NULL,
   -> city VARCHAR(100) NOT NULL,
   -> grade INT,
   -> salesman_id INT,
   -> FOREIGN KEY(salesman_id) REFERENCES salesman(salesman_id)
   -> );
```

```
mysql> desc customer;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| customer_id  | int(11)      | NO   | PRI | NULL    | auto_increment |
| customer_name| varchar(100) | NO   |     | NULL    |                |
| city         | varchar(100) | NO   |     | NULL    |                |
| grade        | int(11)      | YES  |     | NULL    |                |
| salesman_id  | int(11)      | YES  | MUL | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

```
create table orders(
   -> order_no INT AUTO_INCREMENT PRIMARY KEY,
   -> purch_amt DECIMAL(10,2) NOT NULL,
   -> order_date DATE NOT NULL,
```

```
-> customer_id INT,
-> salesman_id INT,
-> FOREIGN KEY(customer_id) REFERENCES customer(customer_id),
-> FOREIGN KEY(salesman_id) REFERENCES salesman(salesman_id)
-> );
```

```
mysql> desc orders;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| order_no    | int(11)       | NO   | PRI | NULL    | auto_increment |
| purch_amt   | decimal(10,2) | NO   |     | NULL    |                |
| order_date  | date          | NO   |     | NULL    |                |
| customer_id | int(11)       | YES  | MUL | NULL    |                |
| salesman_id | int(11)       | YES  | MUL | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)
```

Values of salesman

```
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(5003, 'Lauson Hen', '', 0.12);
insert into salesman values(5007, 'Paul Adam', 'Rome', 0.13);
```

```
+-------------+------------+----------+-----------+
| salesman_id | name       | city     | commision |
+-------------+------------+----------+-----------+
|        5001 | James Hoog | New York |      0.15 |
|        5002 | Nail knite | Paris    |      0.13 |
|        5003 | MC Lyon    | Paris    |      0.14 |
|        5005 | Pit Alex   | London   |      0.11 |
|        5006 | Lauson Hen | New York |      0.12 |
|        5007 | Paul Adam  | Rome     |      0.13 |
+-------------+------------+----------+-----------+
6 rows in set (0.00 sec)
```

Values of customer

```
insert into customer values(3001, 'Brad Guzan', 'London', NULL,5003);
insert into customer values(3004, 'Fabian Johns', 'Paris', 300,5006);
insert into customer values(3007, 'Brad davis', 'New York', 200,5001);
insert into customer values(3009, 'Geoff camero', 'Berlin', 100,5003);
insert into customer values(3008, 'Julian Green', 'London', 300,5002);
insert into customer values(3003, 'Jozy Altidor', 'Moncow', 200,5007);
```

salesman_id	name	city	commision
5001	James Hoog	New York	0.15
5002	Nail knite	Paris	0.13
5003	Lauson Hen		0.12
5005	Pit Alex	London	0.11
5006	MC Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13

6 rows in set (0.00 sec)

Values of orders

```
insert into orders values(70001, 150.5, '2016-10-05',3005,5002);
insert into orders values(70009, 270.65, '2016-09-10',3001,5003);
insert into orders values(70002, 65.26, '2016-10-15',3002,5001);
insert into orders values(70004, 110.5, '2016-08-17',3009,5003);
insert into orders values(70007, 948.5, '2016-09-10',3005,5002);
insert into orders values(70005, 2400.6, '2016-07-27',3007,5001);
insert into orders values(700010, 1983.43, '2016-10-10',3004,5006);
insert into orders values(70003, 2480.4, '2016-10-10',3009,5003);
insert into orders values(700012, 250.45, '2016-06-27',3008,5002);
insert into orders values(700011, 75.29, '2016-08-17',3003,5007);
```

customer_id	customer_name	city	grade	salesman_id
3001	Brad Guzan	London	NULL	5003
3002	Nick rimando	New York	100	5001
3003	Jozy Altidor	Moncow	200	5007
3004	Fabian Johns	Paris	300	5006
3005	Graham Zusi	California	200	5002
3007	Brad davis	New York	200	5001
3008	Julian Green	London	300	5002
3009	Geoff camero	Berlin	100	5003

1. Display name and commission for all the salesmen.

select name, commision FROM salesman;

```
mysql> select name, commision FROM salesman;
+------------+-----------+
| name       | commision |
+------------+-----------+
| James Hoog |      0.15 |
| Nail knite |      0.13 |
| Lauson Hen |      0.12 |
| Pit Alex   |      0.11 |
| MC Lyon    |      0.14 |
| Paul Adam  |      0.13 |
+------------+-----------+
6 rows in set (0.00 sec)
```

2. Retrieve salesman id of all salesmen from orders table without any repeats.

select DISTINCT salesman_id FROM orders;

```
+-------------+
| salesman_id |
+-------------+
|        5001 |
|        5002 |
|        5003 |
|        5006 |
|        5007 |
+-------------+
5 rows in set (0.00 sec)
```

3. Display names and city of salesman, who belongs to the city of Paris.

select name,city FROM salesman WHERE city='Paris';

```
+------------+-------+
| name       | city  |
+------------+-------+
| Nail knite | Paris |
| MC Lyon    | Paris |
+------------+-------+
2 rows in set (0.00 sec)
```

4. Display all the information for those customers with a grade of 200.

select * from customer WHERE grade=200;

```
+-------------+---------------+------------+-------+-------------+
| customer_id | customer_name | city       | grade | salesman_id |
+-------------+---------------+------------+-------+-------------+
|        3003 | Jozy Altidor  | Moncow     |   200 |        5007 |
|        3005 | Graham Zusi   | California |   200 |        5002 |
|        3007 | Brad davis    | New York   |   200 |        5001 |
+-------------+---------------+------------+-------+-------------+
```

5. Display the order number, order date and the purchase amount for order(s) which will be delivered by the salesman with ID 5001.

select order_no,order_date, purch_amt FROM orders WHERE salesman_id="5001";

order_no	order_date	purch_amt
70002	2016-10-15	65.26
70005	2016-07-27	2400.60

2 rows in set (0.00 sec)

6. Display all the customers, who are either belongs to the city New York or not had a grade above 100.

select * from customer where city='New York' OR grade<=100;

customer_id	customer_name	city	grade	salesman_id
3002	Nick rimando	New York	100	5001
3007	Brad davis	New York	200	5001
3009	Geoff camero	Berlin	100	5003

7. Find those salesmen with all information who gets the commission within a range of 0.12 and 0.14.

select * from salesman WHERE commision BETWEEN 0.12 AND 0.14;

salesman_id	name	city	commision
5002	Nail knite	Paris	0.13
5003	Lauson Hen		0.12
5006	MC Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13

4 rows in set (0.00 sec)

8. Find all those customers with all information whose names are ending with the letter 'n'.

select * from salesman WHERE name LIKE '%n';

salesman_id	name	city	commision
5003	Lauson Hen		0.12
5006	MC Lyon	Paris	0.14

2 rows in set (0.00 sec)

9. Find those salesmen with all information whose name containing the 1st character is 'N' and the 4th character is 'l' and rests may be any character.

select * from salesman WHERE name LIKE 'N_i%';

```
+-------------+-----------+-------+-----------+
| salesman_id | name      | city  | commision |
+-------------+-----------+-------+-----------+
|        5002 | Nail knite | Paris |      0.13 |
+-------------+-----------+-------+-----------+
1 row in set (0.00 sec)
```

10. Find that customer with all information who does not get any grade except NULL.

select * from customer WHERE grade is NULL;

```
+-------------+---------------+--------+-------+-------------+
| customer_id | customer_name | city   | grade | salesman_id |
+-------------+---------------+--------+-------+-------------+
|        3001 | Brad Guzan    | London |  NULL |        5003 |
+-------------+---------------+--------+-------+-------------+
1 row in set (0.00 sec)
```

11. Find the total purchase amount of all orders.

select SUM(purch_amt) AS total_purchase FROM orders;

```
+----------------+
| total_purchase |
+----------------+
|       14495.58 |
+----------------+
1 row in set (0.01 sec)
```

12. Find the number of salesman currently listing for all of their customers.

select salesman_id, COUNT(customer_id) AS total_customers FROM customer GROUP BY salesman_id;

```
+-------------+-----------------+
| salesman_id | total_customers |
+-------------+-----------------+
|        5001 |               2 |
|        5002 |               2 |
|        5003 |               1 |
|        5005 |               1 |
|        5006 |               1 |
|        5007 |               1 |
+-------------+-----------------+
6 rows in set (0.01 sec)
```

13. Find the highest grade for each of the cities of the customers.

select city, Max(grade) As highest_grade FROM customer GROUP BY city;

city	highest_grade
Londan	100
New York	200
Moncow	200
Paris	300
California	200
London	300
Berlin	100

7 rows in set (0.01 sec)

14. Find the highest purchase amount ordered by each customer with their ID and highest purchase amount.

select customer_id, Max(purch_amt) AS highest_purchase FROM orders GROUP BY customer_id;

customer_id	highest_purchase
3001	270.65
3002	5760.00
3003	75.29
3004	1983.43
3005	948.50
3007	2400.60
3008	250.45
3009	2480.40

8 rows in set (0.00 sec)

15. Find the highest purchase amount ordered by each customer on a particular date with their ID, order date and highest purchase amount.

select customer_id, order_date, Max(purch_amt) AS highest_purchase FROM orders GROUP BY customer_id, order_date;

customer_id	order_date	highest_purchase
3005	2016-10-05	150.50
3002	2016-10-05	65.26
3009	2016-10-10	2480.40
3009	2016-08-17	110.50
3007	2016-07-27	2400.60
3005	2016-09-10	948.50
3002	2016-09-10	5760.00
3001	2016-09-10	270.65
3004	2016-10-10	1983.43
3003	2016-08-17	75.29
3008	2016-06-27	250.45

11 rows in set (0.00 sec)

16. Find the highest purchase amount on a date '2012-08-17' for each salesman with their ID.
select salesman_id, MAX(purch_amt) AS highest_purchase FROM orders WHERE order_date = 2012-08-17 GROUP BY salesman_id;

```
mysql> select salesman_id, MAX(purch_amt) AS highest_purch
 WHERE order_date = 2012-08-17 GROUP BY salesman_id;
Empty set, 1 warning (0.01 sec)
```

17. Find the highest purchase amount with their customer ID and order date, for only those customers who have the highest purchase amount in a day is more than 2000.
select customer_id, order_date, MAX(purch_amt) AS highest_purchase FROM orders GROUP BY customer_id, order_date HAVING MAX(purch_amt)>2000;

```
+-------------+------------+------------------+
| customer_id | order_date | highest_purchase |
+-------------+------------+------------------+
|        3009 | 2016-10-10 |          2480.40 |
|        3007 | 2016-07-27 |          2400.60 |
|        3002 | 2016-09-10 |          5760.00 |
+-------------+------------+------------------+
3 rows in set (0.01 sec)
```

18. Write a SQL statement that counts all orders for a date August 17th, 2012.
select COUNT(*) AS total_orders FROM orders WHERE order_date = 2012-08-17;

```
+--------------+
| total_orders |
+--------------+
|            0 |
+--------------+
1 row in set, 1 warning (0.00 sec)
```