Rehmah Ahmed Batki
MSC DSAI L010

Practical - 1

6-12-24

DDL operations on Relational Schema

Write-up:-

- Codd's 12 rules
- RDBMS vs DBMS
- Types of attributes
- Types of Keys
- ERD
- Constraints
- DDL

Design the following schema and execute the following queries on it:

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
5003	Lauson Hen		0.12
5007	Paul Adam	Rome	0.13

customer

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

order

order_no	purch_amt	order_date	customer_id	salesman_id
70001	150.5	2016-10-05	3005	5002
70009	270.65	2016-09-10	3001	
70002	65.26	2016-10-05	3002	5001
70004	110.5	2016-08-17	3009	
70007	948.5	2016-09-10	3005	5002
70005	2400.6	2016-07-27	3007	5001
70008	5760	2016-09-10	3002	5001
70010	1983.43	2016-10-10	3004	5006
70003	2480.4	2016-10-10	3009	
70012	250.45	2016-06-27	3008	5002
70011	75.29	2016-08-17	3003	5007

```
mysql> create database rehmah;
Query OK, 1 row affected (0.05 sec)

mysql> use rehmah
Database changed
mysql> create table salesman(
    -> salesman_id int,
    -> name varchar(255),
    -> city varchar(255),
    -> commission float,
    -> PRIMARY KEY(salesman_id)
    -> );
Query OK, 0 rows affected (0.07 sec)

mysql> insert into salesman values(5001, 'James Hooq', 'New York', 0.15);
Query OK, 1 row affected (0.01 sec)

mysql> insert into salesman values(5002, 'Nail Knite', 'Paris', 0.13);
Query OK, 1 row affected (0.00 sec)

mysql> insert into salesman values(5005, 'Pit Alex', 'London', 0.11);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into salesman values(5006, 'Mc Lyon', 'Paris', 0.14);
Query OK, 1 row affected (0.00 sec)

mysql> insert into salesman values(5003, 'Lauson Hen', NULL, 0.12);
Query OK, 1 row affected (0.01 sec)

mysql> insert into salesman values(5007, 'Paul Adam', 'Rome', 0.13);
Query OK, 1 row affected (0.00 sec)

mysql> select * from salesman;
+-------------+------------+----------+------------+
| salesman_id | name       | city     | commission |
+-------------+------------+----------+------------+
|        5001 | James Hooq | New York |       0.15 |
|        5002 | Nail Knite | Paris    |       0.13 |
|        5003 | Lauson Hen | NULL     |       0.12 |
|        5005 | Pit Alex   | London   |       0.11 |
|        5006 | Mc Lyon    | Paris    |       0.14 |
|        5007 | Paul Adam  | Rome     |       0.13 |
+-------------+------------+----------+------------+
6 rows in set (0.01 sec)

mysql> create table customer(
    -> customer_id int,
    -> customer_name varchar(255),
    -> city varchar(255),
    -> grade int,
    -> salesman_id int,
    -> PRIMARY KEY(customer_id),
    -> FOREIGN KEY(salesman_id) REFERENCES salesman(salesman_id)
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> insert into customer values(3002, 'Nick Rimando', 'New York', 100,
5001);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3005, 'Graham Fuel', 'California', 200,
5002);
Query OK, 1 row affected (0.00 sec)

mysql> insert into customer values(3001, 'Brad Guran', 'London', NULL,
NULL);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3004, 'Fabian Johns', 'Paris', 300,
5006);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3007, 'Brad Davis', 'New York', 200,
5001);
Query OK, 1 row affected (0.00 sec)

mysql> insert into customer values(3009, 'Geoff Camero', 'Berlin', 100,
NULL);
Query OK, 1 row affected (0.00 sec)

mysql> insert into customer values(3008, 'Julian Green', 'London', 300,
5002);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3003, 'Joxy Altidor', 'Moncow', 200,
5007);
Query OK, 1 row affected (0.00 sec)

mysql> select * from customer;
```

```
+------------+--------------+-----------+-------+------------+
| customer_id | customer_name | city      | grade | salesman_id |
+------------+--------------+-----------+-------+------------+
|       3001 | Brad Guran    | London    |  NULL |       NULL |
|       3002 | Nick Rimando  | New York  |   100 |       5001 |
|       3003 | Joxy Altidor  | Moncow    |   200 |       5007 |
|       3004 | Fabian Johns  | Paris     |   300 |       5006 |
|       3005 | Graham Fuel   | California |   200 |       5002 |
|       3007 | Brad Davis    | New York  |   200 |       5001 |
|       3008 | Julian Green  | London    |   300 |       5002 |
|       3009 | Geoff Camero  | Berlin    |   100 |       NULL |
+------------+--------------+-----------+-------+------------+
8 rows in set (0.00 sec)

mysql> create table `order`(
    -> order_no int,
    -> purch_amt float,
    -> order_date date,
    -> customer_id int,
    -> salesman_id int,
    -> primary key(order_no),
    -> foreign key(customer_id) references customer(customer_id),
    -> foreign key(salesman_id) references salesman(salesman_id)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> insert into `order` values(70001, 150.5, '2016-10-05', 3005, 5002);
Query OK, 1 row affected (0.01 sec)

mysql> insert into `order` values(70009, 270.65, '2016-09-10', 3001,
NULL);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into `order` values(70002, 65.26, '2016-10-05', 3002, 5001);
Query OK, 1 row affected (0.01 sec)

mysql> insert into `order` values(70004, 110.5, '2016-08-17', 3009, NULL);
Query OK, 1 row affected (0.01 sec)

mysql> insert into `order` values(70007, 948.5, '2016-09-10', 3005, 5002);
Query OK, 1 row affected (0.00 sec)

mysql> insert into `order` values(70005, 2400.6, '2016-07-27', 3007, 5001);
Query OK, 1 row affected (0.00 sec)

mysql> insert into `order` values(70008, 5760, '2016-09-10', 3002, 5001);
Query OK, 1 row affected (0.01 sec)

mysql> insert into `order` values(70010, 1983.43, '2016-10-10', 3004,
5006);
Query OK, 1 row affected (0.01 sec)

mysql> insert into `order` values(70003, 2480.4, '2016-10-10', 3009,
NULL);
Query OK, 1 row affected (0.00 sec)

mysql> insert into `order` values(70012, 250.45, '2016-06-27', 3008, 5002);
Query OK, 1 row affected (0.00 sec)

mysql> insert into `order` values(70011, 75.29, '2016-08-17', 3003, 5007);
Query OK, 1 row affected (0.00 sec)

mysql> select * from `order`;
+----------+-----------+------------+-------------+-------------+
| order_no | purch_amt | order_date | customer_id | salesman_id |
+----------+-----------+------------+-------------+-------------+
```

70001	150.5	2016-10-05	3005	5002
70002	65.26	2016-10-05	3002	5001
70003	2480.4	2016-10-10	3009	NULL
70004	110.5	2016-08-17	3009	NULL
70005	2400.6	2016-07-27	3007	5001
70007	948.5	2016-09-10	3005	5002
70008	5760	2016-09-10	3002	5001
70009	270.65	2016-09-10	3001	NULL
70010	1983.43	2016-10-10	3004	5006
70011	75.29	2016-08-17	3003	5007
70012	250.45	2016-06-27	3008	5002

11 rows in set (0.00 sec)

1. Display name and commission for all the salesmen.

```
mysql> SELECT name, commission from salesman;
```

name	commission
James Hooq	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.

```
mysql> SELECT DISTINCT salesman_id FROM `order`;
```

```
+-------------+
| salesman_id |
+-------------+
|        NULL |
|        5001 |
|        5002 |
|        5006 |
|        5007 |
+-------------+
5 rows in set (0.01 sec)
```

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

```
mysql> SELECT name, city FROM salesman WHERE city = 'Paris';
+------------+-------+
| name       | city  |
+------------+-------+
| Nail Knite | Paris |
| Mc Lyon    | Paris |
+------------+-------+
```

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

```
mysql> SELECT * FROM customer WHERE grade = 200;
+-------------+---------------+------------+-------+-------------+
| customer_id | customer_name | city       | grade | salesman_id |
+-------------+---------------+------------+-------+-------------+
|        3003 | Joxy Altidor  | Moncow     |   200 |        5007 |
|        3005 | Graham Fuel   | California |   200 |        5002 |
|        3007 | Brad Davis    | New York   |   200 |        5001 |
+-------------+---------------+------------+-------+-------------+
3 rows in set (0.00 sec)
```

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

```
mysql> SELECT order_no, order_date, purch_amt FROM `order` WHERE
salesman_id = 5001;
+----------+------------+-----------+
| order_no | order_date | purch_amt |
+----------+------------+-----------+
|    70002 | 2016-10-05 |     65.26 |
|    70005 | 2016-07-27 |    2400.6 |
|    70008 | 2016-09-10 |      5760 |
+----------+------------+-----------+
3 rows in set (0.00 sec)
```

6. Show the winner of the 1971 prize for Literature.
7. Show all the details of the winners with first name Louis.
8. Show all the winners in Physics for 1970 together with the winner of Economics for 1971.
9. Show all the winners of Nobel prize in the year 1970 except the subject Physiology and Economics.
10. Find all the details of the Nobel winners for the subject not started with the letter 'P' and arranged the list as the most recent comes first, then by name in order.
11. Find the name and price of the cheapest item(s).

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

```
mysql> 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 |        NULL |
+-------------+---------------+----------+-------+-------------+
3 rows in set (0.00 sec)
```

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

```
mysql> SELECT * FROM salesman WHERE commission BETWEEN 0.12
AND 0.14;
+-------------+-----------+-------+------------+
| salesman_id | name      | city  | commission |
+-------------+-----------+-------+------------+
|        5002 | Nail Knite | Paris |       0.13 |
|        5007 | Paul Adam  | Rome  |       0.13 |
+-------------+-----------+-------+------------+
2 rows in set (0.00 sec)
```

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

```
mysql> SELECT * FROM customer WHERE customer_name LIKE '%n';
+-------------+---------------+--------+-------+-------------+
| customer_id | customer_name | city   | grade | salesman_id |
+-------------+---------------+--------+-------+-------------+
|        3001 | Brad Guran    | London |  NULL |        NULL |
|        3008 | Julian Green  | London |   300 |        5002 |
+-------------+---------------+--------+-------+-------------+
2 rows in set (0.00 sec)
```

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

```
mysql> SELECT * FROM salesman WHERE name LIKE 'N__I%';
+------------+-----------+-------+------------+
| salesman_id | name       | city  | commission |
+------------+-----------+-------+------------+
|       5002 | Nail Knite | Paris |       0.13 |
+------------+-----------+-------+------------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT * FROM customer WHERE grade IS NULL;
+------------+--------------+--------+-------+------------+
| customer_id | customer_name | city   | grade | salesman_id |
+------------+--------------+--------+-------+------------+
|       3001 | Brad Guran    | London |  NULL |       NULL |
+------------+--------------+--------+-------+------------+
1 row in set (0.00 sec)
```

17. Find the total purchase amount of all orders.

```
mysql> SELECT SUM(purch_amt) AS total_purchase_amount FROM
`order`;
+---------------------+
| total_purchase_amount |
+---------------------+
|    14495.580047607422 |
+---------------------+
1 row in set (0.01 sec)
```

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

```
mysql> SELECT COUNT(DISTINCT salesman_id) AS
number_of_salesmen FROM customer;
+-------------------+
| number_of_salesmen |
+-------------------+
|                 4 |
+-------------------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT city, MAX(grade) AS highest_grade FROM customer
GROUP BY city;
+------------+---------------+
| city       | highest_grade |
+------------+---------------+
| London     |           300 |
| New York   |           200 |
| Moncow     |           200 |
| Paris      |           300 |
| California |           200 |
| Berlin     |           100 |
+------------+---------------+
6 rows in set (0.00 sec)
```

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

```
mysql> SELECT customer_id, MAX(purch_amt) AS
highest_purchase_amount FROM `order` GROUP BY customer_id;
+-------------+-----------------------+
| customer_id | highest_purchase_amount |
+-------------+-----------------------+
|        3001 |                270.65 |
|        3002 |                  5760 |
|        3003 |                 75.29 |
|        3004 |               1983.43 |
|        3005 |                 948.5 |
|        3007 |                2400.6 |
|        3008 |                250.45 |
|        3009 |                2480.4 |
+-------------+-----------------------+
8 rows in set (0.00 sec)
```

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

```
mysql> SELECT customer_id, order_date, MAX(purch_amt) AS
highest_purchase_amount
FROM `order` GROUP BY customer_id, order_date;
+-------------+------------+-----------------------+
| customer_id | order_date | highest_purchase_amount |
+-------------+------------+-----------------------+
|        3005 | 2016-10-05 |                 150.5 |
|        3002 | 2016-10-05 |                 65.26 |
|        3009 | 2016-10-10 |                2480.4 |
|        3009 | 2016-08-17 |                 110.5 |
|        3007 | 2016-07-27 |                2400.6 |
|        3005 | 2016-09-10 |                 948.5 |
|        3002 | 2016-09-10 |                  5760 |
|        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)
```

22. Find the highest purchase amount on a date '2012-08-17' for each salesman with their ID.

```
mysql> SELECT salesman_id, MAX(purch_amt) AS
highest_purchase_amount FROM `order` WHERE order_date =
'2012-08-17' GROUP BY salesman_id;
Empty set (0.00 sec)
```

```
mysql> SELECT salesman_id, MAX(purch_amt) AS
highest_purchase_amount FROM `order` WHERE order_date =
'2016-08-17' GROUP BY salesman_id;
+------------+-----------------------+
| salesman_id | highest_purchase_amount |
+------------+-----------------------+
|       NULL |                 110.5 |
|       5007 |                 75.29 |
+------------+-----------------------+
2 rows in set (0.00 sec)
```

23. 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.

```
mysql> SELECT customer_id, order_date, MAX(purch_amt) AS
highest_purchase_amount
    -> FROM `order`
    -> GROUP BY customer_id, order_date
```

```
   -> HAVING MAX(purch_amt) > 2000;
+-------------+------------+-----------------------+
| customer_id | order_date | highest_purchase_amount |
+-------------+------------+-----------------------+
|        3009 | 2016-10-10 |                2480.4 |
|        3007 | 2016-07-27 |                2400.6 |
|        3002 | 2016-09-10 |                  5760 |
+-------------+------------+-----------------------+
3 rows in set (0.00 sec)
```

24. Write a SQL statement that counts all orders for a date August 17th, 2012.

```
mysql> SELECT COUNT(*) AS total_orders
   -> FROM `order`
   -> WHERE order_date = '2012-08-17';
+--------------+
| total_orders |
+--------------+
|            0 |
+--------------+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) AS total_orders
   -> FROM `order`
   -> WHERE order_date = '2016-08-17';
+--------------+
| total_orders |
+--------------+
|            2 |
+--------------+
1 row in set (0.00 sec)
```