NAME: AAMINAKHAN ROLLNO:L033 (Msc DSAI)

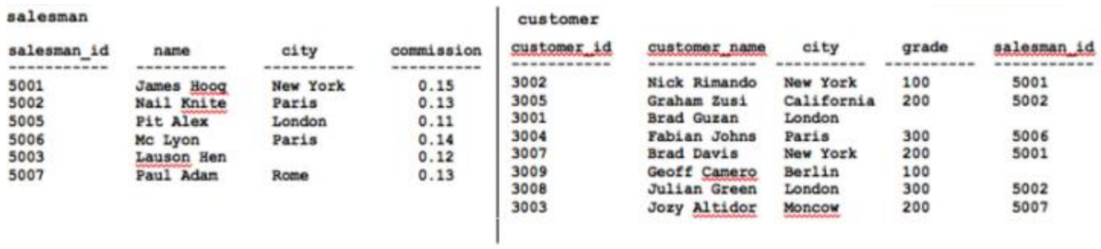
Practical-1 : DDL operations on Relational Schema

Date:-06/12/2024 **Submission Date:- 3/02/2024**

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:

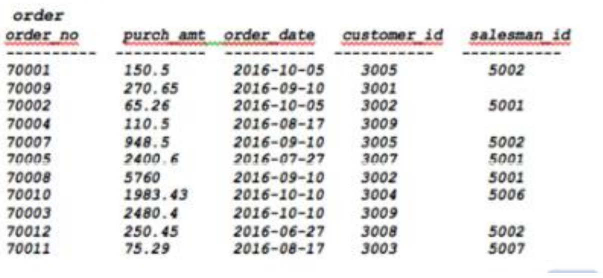


salesman

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

Salesman Table:

```
create table salesman(saleman_id int primary key, name varchar(50), city
varchar(50),
   -> commision decimal(5,2));
Query OK, 0 rows affected (4.37 sec)
insert into salesman(saleman_id, name, city, commision) values
   -> (5001, 'James Hong', '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', Null, 0.12),
   -> (5007, 'Paul Aden', 'Rome', 0.13);
Query OK, 6 rows affected (2.10 sec)
select * from salesman;
```

```
+------------+------------+----------+-----------+
| saleman_id | name       | city     | commision |
+------------+------------+----------+-----------+
|       5001 | James Hong | 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 Aden  | Rome     |      0.13 |
+------------+------------+----------+-----------+
6 rows in set (0.27 sec)
```

Customer Table:

```
create table customer(customer_id int primary key, customer_name
varchar(50),
    -> city varchar(50),
    -> grade int,
    -> saleman_id int, foreign key (saleman_id) references
salesman(saleman_id));
Query OK, 0 rows affected (36.78 sec)

insert into customer (customer_id, customer_name, city, grade, saleman_id)
values
    -> (3002, 'Nick Rimando', 'New York', 100, 5001),
    -> (3005, 'Graham Zusi', 'California', 200, 5002),
    -> (3001, 'Brad Guzan', 'London', NULL, 5005),
    -> (3004, 'Fabian Johns', 'Paris', 300, 5006),
    -> (3007, 'Brad Davis', 'New York', 200, 5001),
    -> (3009, 'Geoff Camero', 'Berlin', 100, NULL),
    -> (3008, 'Julian Green', 'London', 300, 5002),
    -> (3003, 'Jozy Altidor', 'Moncow', 200, 5007);
Query OK, 8 rows affected (1.31 sec)

select * from customer;
+-------------+---------------+------------+-------+------------+
| customer_id | customer_name | city       | grade | saleman_id |
+-------------+---------------+------------+-------+------------+
|        3001 | Brad Guzan    | London     |  NULL |       5005 |
|        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 |       NULL |
+-------------+---------------+------------+-------+------------+
8 rows in set (0.00 sec)
```

Order Table:

```
CREATE TABLE `order` (
   ->     order_no INT PRIMARY KEY,
   ->     purch_amt DECIMAL(10, 2),
   ->     order_date DATE,
   ->     customer_id INT,
   ->     saleman_id INT,
   ->     FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
   ->     FOREIGN KEY (saleman_id) REFERENCES salesman(saleman_id));
Query OK, 0 rows affected (53.09 sec)
INSERT INTO `order` (order_no, purch_amt, order_date, customer_id,
saleman_id)
   -> VALUES
   -> (70001, 150.50, '2016-10-05', 3005, 5002),
   -> (70009, 270.65, '2016-09-10', 3001, NULL),
   -> (70002, 65.26, '2016-10-05', 3002, 5001),
   -> (70004, 110.50, '2016-08-17', 3009, NULL),
   -> (70007, 948.50, '2016-09-10', 3005, 5002),
   -> (70005, 2400.60, '2016-07-27', 3007, 5001),
   -> (70008, 5760.00, '2016-09-10', 3002, 5001),
   -> (70010, 1983.43, '2016-10-10', 3004, 5006),
   -> (70003, 2480.40, '2016-10-10', 3009, NULL),
   -> (70012, 250.45, '2016-06-27', 3008, 5002),
   -> (70011, 75.29, '2016-08-17', 3003, 5007);
Query OK, 11 rows affected (0.87 sec)

select * from `order`;
+----------+-----------+------------+-------------+------------+
| order_no | purch_amt | order_date | customer_id | saleman_id |
+----------+-----------+------------+-------------+------------+
|    70001 |    150.50 | 2016-10-05 |        3005 |       5002 |
|    70002 |     65.26 | 2016-10-05 |        3002 |       5001 |
|    70003 |   2480.40 | 2016-10-10 |        3009 |       NULL |
|    70004 |    110.50 | 2016-08-17 |        3009 |       NULL |
|    70005 |   2400.60 | 2016-07-27 |        3007 |       5001 |
|    70007 |    948.50 | 2016-09-10 |        3005 |       5002 |
|    70008 |   5760.00 | 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 (0.61 sec)
```

Display name and commission for all the salesmen.

```
SELECT name, commision from salesman;
+------------+-----------+
| name       | commision |
+------------+-----------+
| James Hong |      0.15 |
| Nail Knite |      0.13 |
| Lauson Hen |      0.12 |
| Pit Alex   |      0.11 |
| Mc Lyon    |      0.14 |
| Paul Aden  |      0.13 |
+------------+-----------+
6 rows in set (0.00 sec)
```

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

```
SELECT DISTINCT saleman_id
    -> FROM `order`;
+--------+
|      NULL |
|      5001 |
|      5002 |
|      5006 |
|      5007 |
+------------+
5 rows in set (0.20 sec)
```

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 |
+------------+-------+
2 rows in set (0.51 sec)
```

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

```
SELECT *
    -> FROM customer
    -> WHERE grade = 200;
+-------------+---------------+------------+-------+------------+
| customer_id | customer_name | city       | grade | saleman_id |
+-------------+---------------+------------+-------+------------+
|        3003 | Jozy Altidor  | Moncow     |   200 |       5007 |
|        3005 | Graham Zusi   | California |   200 |       5002 |
|        3007 | Brad Davis    | New York   |   200 |       5001 |
+-------------+---------------+------------+-------+------------+
3 rows in set (0.08 sec)
```

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 `order`
    -> WHERE saleman_id = 5001;
+----------+------------+-----------+
| order_no | order_date | purch_amt |
+----------+------------+-----------+
|    70002 | 2016-10-05 |     65.26 |
|    70005 | 2016-07-27 |   2400.60 |
|    70008 | 2016-09-10 |   5760.00 |
+----------+------------+-----------+
3 rows in set (0.11 sec)
```

Show the winner of the 1971 prize for Literature.
Showall the details of the winners with first name Louis.

Show all the winners in Physics for 1970 together with the winner of Economics for 1971.
Show all the winners of Nobel prize in the year 1970 except the subject Physiology and Economics.
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.
Find the name and price of the cheapest item(s).

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 | saleman_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 (0.15 sec)
```

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;
+------------+------------+-------+-----------+
| saleman_id | name       | city  | commision |
+------------+------------+-------+-----------+
|       5002 | Nail Knite | Paris |      0.13 |
|       5003 | Lauson Hen | NULL  |      0.12 |
|       5006 | Mc Lyon    | Paris |      0.14 |
|       5007 | Paul Aden  | Rome  |      0.13 |
+------------+------------+-------+-----------+
4 rows in set (0.37 sec)
```

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

```
SELECT *
   -> FROM customer
   -> WHERE customer_name LIKE '%n';
+-------------+---------------+--------+-------+------------+
| customer_id | customer_name | city   | grade | saleman_id |
+-------------+---------------+--------+-------+------------+
|        3001 | Brad Guzan    | London |  NULL |       5005 |
|        3008 | Julian Green  | London |   300 |       5002 |
+-------------+---------------+--------+-------+------------+
2 rows in set (0.49 sec)
```

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__l%';
+------------+------------+-------+-----------+
| saleman_id | name       | city  | commision |
+------------+------------+-------+-----------+
|       5002 | Nail Knite | Paris |      0.13 |
+------------+------------+-------+-----------+
1 row in set (0.00 sec)
```

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 | saleman_id |
+-------------+---------------+--------+-------+------------+
|        3001 | Brad Guzan    | London |  NULL |       5005 |
+-------------+---------------+--------+-------+------------+
1 row in set (0.00 sec)
```

Find the total purchase amount of all orders.

```
SELECT SUM(purch_amt) AS total_purchase_amount
   -> FROM `order`;
+-----------------------+
| total_purchase_amount |
+----------------------+
|           14495.58 |
+----------------------+
1 row in set (0.15 sec)
```

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

```
SELECT saleman_id, COUNT(DISTINCT customer_id) AS total_customers
   -> FROM `order`
   -> GROUP BY saleman_id;
+------------+-----------------+
| saleman_id | total_customers |
+------------+-----------------+
|      NULL |             2 |
|      5001 |             2 |
|      5002 |             2 |
|      5006 |             1 |
|      5007 |             1 |
+------------+-----------------+
5 rows in set (0.57 sec)
```

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 |
+------------+---------------+
| London     |         300 |
| New York   |         200 |
| Moncow     |         200 |
| Paris      |         300 |
| California |         200 |
| Berlin     |         100 |
+------------+---------------+
6 rows in set (0.82 sec)
```

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_amount
   -> FROM `order`
   -> GROUP BY customer_id;
+-------------+------------------------+
| customer_id | highest_purchase_amount |
+-------------+------------------------+
|        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 (0.33 sec)
```

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_amount
   -> FROM `order`
   -> GROUP BY customer_id, order_date;
+-------------+------------+------------------------+
| customer_id | order_date | highest_purchase_amount |
+-------------+------------+------------------------+
|        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)
```

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

```
SELECT saleman_id, MAX(purch_amt) AS highest_purchase_amount
   -> FROM `order`
   -> WHERE order_date = '2012-08-17'
   -> GROUP BY saleman_id;
Empty set (0.22 sec)
```

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_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 |                 2480.40 |
|        3007 | 2016-07-27 |                 2400.60 |
|        3002 | 2016-09-10 |                 5760.00 |
+-------------+------------+-------------------------+
3 rows in set (0.00 sec)
```

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

```
SELECT COUNT(*) AS total_orders
   -> FROM `order`
   -> WHERE order_date = '2012-08-17';
+--------------+
| total_orders |
+--------------+
|            0 |
+--------------+
1 row in set (0.09 sec)
```