

# **IT252 Assignment-6**

NAME: SUYASH CHINTAWAR

ROLL NO.:191IT109

TOPIC: SQL VIEWS

## Q1.(b) Solutions

- 1) Write a query to create a view that shows for each order the salesman and customer by name.

```
mysql> create view ques1_1 as select o.ord_no Order_no,c.customer_id,c.cust_name Customer,
s.salesman_id,s.name Salesman from orders o,salesman s,customer c where o.customer_id=c.c
ustomer_id and s.salesman_id=o.salesman_id;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from ques1_1;
```

Order_no	customer_id	Customer	salesman_id	Salesman
70002	3002	Nick Rimando	5001	James Hoog
70005	3007	Brad Davis	5001	James Hoog
70008	3002	Nick Rimando	5001	James Hoog
70013	3002	Nick Rimando	5001	James Hoog
70001	3005	Graham Zusi	5002	Nail Knite
70007	3005	Graham Zusi	5002	Nail Knite
70012	3008	Julian Green	5002	Nail Knite
70003	3009	Geoff Cameron	5003	Lauson Hen
70004	3009	Geoff Cameron	5003	Lauson Hen
70009	3001	Brad Guzan	5005	Pit Alex
70010	3004	Fabian Johnson	5006	Mc Lyon
70011	3003	Jozy Altidor	5007	Paul Adam

12 rows in set (0.00 sec)

- 2) Write a query to create a view that finds the salesman who has the customer with the highest order of a day.

```
mysql> create view ques1_2 as select o.ord_date,o.ord_no,o.purch_amt,s.salesman_id,s.name,
c.cust_name from orders o,customer c,salesman s where o.customer_id=c.customer_id and s.sa
lesman_id=o.salesman_id and o.purch_amt=any(select max(purch_amt) from orders group by ord
_date);
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select * from ques1_2;
```

ord_date	ord_no	purch_amt	salesman_id	name	cust_name
2012-07-27	70005	2400	5001	James Hoog	Brad Davis
2012-09-10	70008	5760	5001	James Hoog	Nick Rimando
2012-04-25	70013	3045	5001	James Hoog	Nick Rimando
2012-10-05	70001	150	5002	Nail Knite	Graham Zusi
2012-06-27	70012	250	5002	Nail Knite	Julian Green
2012-10-10	70003	2480	5003	Lauson Hen	Geoff Cameron
2012-08-17	70004	110	5003	Lauson Hen	Geoff Cameron

7 rows in set (0.00 sec)

3) Write a query to create a view to getting a count of how many customers we have at each level of a grade.

```
mysql> create view ques1_3 as select grade,count(*) count from customer group by grade;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from ques1_3;
+-----+-----+
| grade | count |
+-----+-----+
| NULL  | 1     |
| 100   | 2     |
| 200   | 3     |
| 300   | 2     |
+-----+-----+
4 rows in set (0.00 sec)
```

4) Write a query to find the salesmen of the city New York who achieved the commission more than 13%.

```
mysql> create view ques1_4 as select * from salesman where city='New York' and commision>0.13;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from ques1_4;
+-----+-----+-----+-----+
| salesman_id | name      | city    | commision |
+-----+-----+-----+-----+
| 5001        | James Hoog | New York | 0.15      |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

NOTE: Ques 2 on next page

## Q2(b) Solutions

- 1) Create a view called TNS containing title-name-stars triples, where the movie (title) was reviewed by a reviewer (name) and received the rating (stars). Then referencing only view TNS and table Movie, write a SQL query that returns the latest year of any movie reviewed by Chris Jackson. You may assume movie names are unique.

```
mysql> create or replace view TNS as select m.title, re.name, ra.stars from Movie m,
Reviewer re, Rating ra where m.MID=ra.MID and re.rID=ra.rID;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select * from TNS;
```

title	name	stars
Gone with the Wind	Sarah Martinez	4
Gone with the Wind	Sarah Martinez	2
Snow White	Daniel Lewis	4
Raiders of the Lost Ark	Brittany Harris	2
Raiders of the Lost Ark	Brittany Harris	4
The Sound of Music	Brittany Harris	2
Gone with the Wind	Mike Anderson	3
Raiders of the Lost Ark	Chris Jackson	4
E.T.	Chris Jackson	2
The Sound of Music	Chris Jackson	3
Snow White	Elizabeth Thomas	5
Avatar	Elizabeth Thomas	3
Avatar	James Cameron	5
E.T.	Ashley White	3

```
14 rows in set (0.00 sec)
```

```
mysql> select max(year) year from Movie where title=any(select title from TNS wher
e name='Chris Jackson');
```

year
1982

```
1 row in set (0.00 sec)
```

- 2) Referencing view TNS from Exercise 1 and no other tables, create a view RatingStats containing each movie title that has at least one rating, the number of ratings it received, and its average rating. Then referencing view RatingStats and no other tables, write a SQL query to find the title of the highest-average-rating movie with at least three ratings.

```
mysql> create or replace view RatingStats as select title,count(*) rating_count, avg(stars) avg_rating from TNS group by title having count(*)>=1;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select * from RatingStats;
```

title	rating_count	avg_rating
Gone with the Wind	3	3.0000
Snow White	2	4.5000
Raiders of the Lost Ark	3	3.3333
The Sound of Music	2	2.5000
E.T.	2	2.5000
Avatar	2	4.0000

```
6 rows in set (0.00 sec)
```

```
mysql> select title from RatingStats where avg_rating=(select max(avg_rating) from RatingStats where rating_count>=3);
```

title
Raiders of the Lost Ark

```
1 row in set (0.01 sec)
```

- 3) Create a view Favorites containing rID-mID pairs, where the reviewer with rID gave the movie with mID the highest rating he or she gave any movie. Then referencing only view Favorites and tables Movie and Reviewer, write a SQL query to return reviewer-reviewer-movie triples where the two (different) reviewers have the movie as their favorite. Return each pair once, i.e., don't return a pair and its inverse.

```
mysql> create or replace view Favorites as select rid,mid from Rating where (rid,stars)=any(select rid,max(stars) from Rating group by rid);
Query OK, 0 rows affected (0.02 sec)

mysql> select * from Favorites;
+-----+-----+
| rid  | mid  |
+-----+-----+
| 201  | 101  |
| 202  | 106  |
| 203  | 108  |
| 204  | 101  |
| 205  | 108  |
| 206  | 106  |
| 207  | 107  |
| 208  | 104  |
+-----+-----+
8 rows in set (0.00 sec)

mysql> select a.name Reviewer1,b.name Reviewer2,a.title Movie from (select name,title,f.rid,f.mid from Favorites f,Movie m,Reviewer r where f.rid=r.rid and m.mid=f.mid) a,(select name,title,f.rid,f.mid from Favorites f,Movie m,Reviewer r where f.rid=r.rid and m.mid=f.mid) b where a.rid<>b.rid and a.mid=b.mid and a.rid<b.rid;
+-----+-----+-----+
| Reviewer1 | Reviewer2 | Movie |
+-----+-----+-----+
| Sarah Martinez | Mike Anderson | Gone with the Wind |
| Brittany Harris | Chris Jackson | Raiders of the Lost Ark |
| Daniel Lewis | Elizabeth Thomas | Snow White |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

THANK YOU