**Assignment No. 27.2**

1. Who is the customer who spent the most on rental movies? Return his/her customer id, first name and the amount spent.

mysql> **create** **view** rental\_paying\_customer\_1 AS (**select** customer.customer\_id, customer.first\_name, **sum**(payment.amount) **as** totalpay **from** customer, payment, rental **where** rental.customer\_id = payment.customer\_id **and** payment.customer\_id = customer.customer\_id **group** **by** rental.customer\_id);

Query OK, 0 rows affected (0.07 sec)

mysql> **select** customer\_id, first\_name, totalpay **from** rental\_paying\_customer\_1 **where** totalpay = (**select** **max**(totalpay) **from** rental\_paying\_customer\_1);

+-------------+------------+----------+

| customer\_id | first\_name | totalpay |

+-------------+------------+----------+

| 526 | KARL | 9969.75 |

+-------------+------------+----------+

1 row in set (1.34 sec)

1. Give an interesting query of your own that is not already in the assignment. The query should involve at least two joins, HAVING clause and aggregation operation. Give the English explanation and the answer.

THE QUERRY: I wanted to know what the inventory of films in which the top 10 actors have acted. The top 10 actors were selected based on the number of films they featured. Essentially I wanted to know whether the films of top actors are sold in large numbers.

Since this information is scattered across in multiple tables like, inventory, film, film\_actor. So I wanted to create views like popularactor with the following querry

mysql> create view popularactor as (select actor\_id, count(film\_id) as number from film\_actor groupby actor\_id;

Then I would create a view of the top 10 actors and their number of films

mysql> create view top10actors as (select \* from popularactor order by number desc limit 10);

Query OK, 0 rows affected (0.06 sec)

Then I would like to map the films which included the top 10 actors by creating a view like the following querry

mysql> create view filmstop10actors as (select film\_id from film\_actor where film\_actor.actor\_id IN (SELECT actor\_id from top10actors));

Query OK, 0 rows affected (0.08 sec)

Finally I would like to use the inner join two times the group by function to get the stock of films in which the top 10 actors have acted as follows

mysql> select inventory.film\_id, count(inventory.inventory\_id) as stock from inventory inner join film\_actor on inventory.film\_id = film\_actor.film\_id inner join filmstop10actors on inventory.film\_id = filmstop10actors.film\_id group by inventory.film\_id order by count(inventory.inventory\_id) desc;

adding the having clause with more than 100 pieces of the films in inventory as follows

mysql> select inventory.film\_id, count(inventory.inventory\_id) as stock from inventory inner join film\_actor on inventory.film\_id = film\_actor.film\_id inner join filmstop10actors on inventory.film\_id = filmstop10actors.film\_id group by inventory.film\_id having count(inventory.inventory\_id) > 100 order by count(inventory.inventory\_id) desc;

+---------+-------+

| film\_id | stock |

+---------+-------+

| 892 | 168 |

| 606 | 156 |

| 833 | 140 |

| 508 | 135 |

| 458 | 120 |

| 785 | 120 |

| 966 | 120 |

| 162 | 112 |

| 311 | 112 |

| 239 | 112 |

+---------+-------+

10 rows in set (0.03 sec)