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

use sakila;

SELECT c1.customer\_id,c1.first\_name,sum(p1.amount) as Amount\_spent

FROM customer c1,payment p1

WHERE c1.customer\_id=p1.customer\_id

GROUP BY c1.customer\_id,c1.first\_name

HAVING sum(p1.amount)=(SELECT max(amt) FROM (SELECT customer\_id,sum(amount) amt

FROM payment GROUP BY customer\_id)as t);

Output:

customer\_id first\_name Amount\_spent

|  |  |  |
| --- | --- | --- |
| 526 | KARL | 221.55 |
|  |  |  |
|  |  |  |

**2. 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 sql code below displays **top 5 films with the max rental\_rate and max replacement\_cost** with the first\_name and last\_name of the actors.

>Two **JOIN**s are used to connect table ‘actor’ to table ‘film’ through table ‘film\_actor’.

>**Having** clause and **Max** operators are used to filter the max rental\_rate.

>**Order by** used to sort the replacement\_cost in the descending order limited to 5.

use sakila;

SELECT a1.first\_name,a1.last\_name,f2.title,f2.rental\_rate,f2.replacement\_cost

FROM actor a1

JOIN film\_actor f1 ON a1.actor\_id=f1.actor\_id

JOIN film f2 ON f1.film\_id=f2.film\_id

HAVING f2.rental\_rate=(SELECT MAX(rental\_rate) FROM film AS t)

ORDER BY replacement\_cost DESC LIMIT 5;

Output:

first\_name last\_name title rental\_rate replacement\_cost

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BURT | DUKAKIS | BLINDNESS GUN | 4.99 | 29.99 |
| CHRIS | BRIDGES | BLINDNESS GUN | 4.99 | 29.99 |
| ADAM | HOPPER | BLINDNESS GUN | 4.99 | 29.99 |
| CARY | MCCONAUGHEY | DESPERATE TRAINSPOTTING | 4.99 | 29.99 |
| RENEE | TRACY | DESPERATE TRAINSPOTTING | 4.99 | 29.99 |