**1st Answer**

SELECT c.first\_name, c.last\_name, f.title, DATEDIFF(r.return\_date,r.rental\_date)/7 AS rental\_time

FROM rental AS r

INNER JOIN inventory AS i

ON r.inventory\_id = i.inventory\_id

INNER JOIN film AS f

ON i.film\_id = f.film\_id

INNER JOIN customer AS c

ON r.customer\_id = c.customer\_id

HAVING rental\_time = (SELECT MAX(DATEDIFF(return\_date,rental\_date))/7 FROM rental)

**2nd Answer**

**Favouring aggregation over joins**

From [SQL statement - “join” vs “group by and having”](https://stackoverflow.com/questions/477006/sql-statement-join-vs-group-by-and-having/477013#477013):

First query:

SELECT userid

FROM userrole

WHERE roleid IN (1, 2, 3)

GROUP by userid

HAVING COUNT(1) = 3

Query time: 0.312 s

Second query:

SELECT t1.userid

FROM userrole t1

JOIN userrole t2 ON t1.userid = t2.userid AND t2.roleid = 2

JOIN userrole t3 ON t2.userid = t3.userid AND t3.roleid = 3

AND t1.roleid = 1

Query time: 0.016 s

The join version I proposed is **faster than the aggregate version**