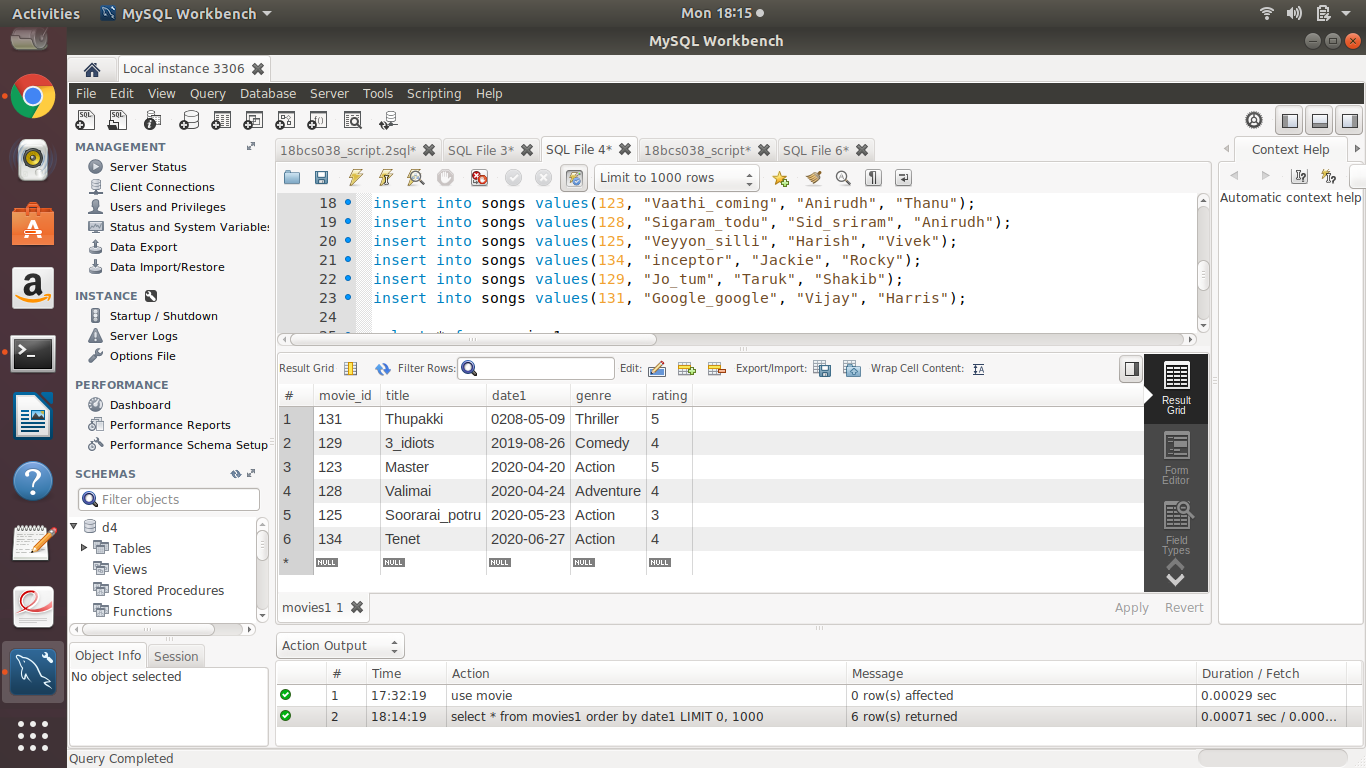
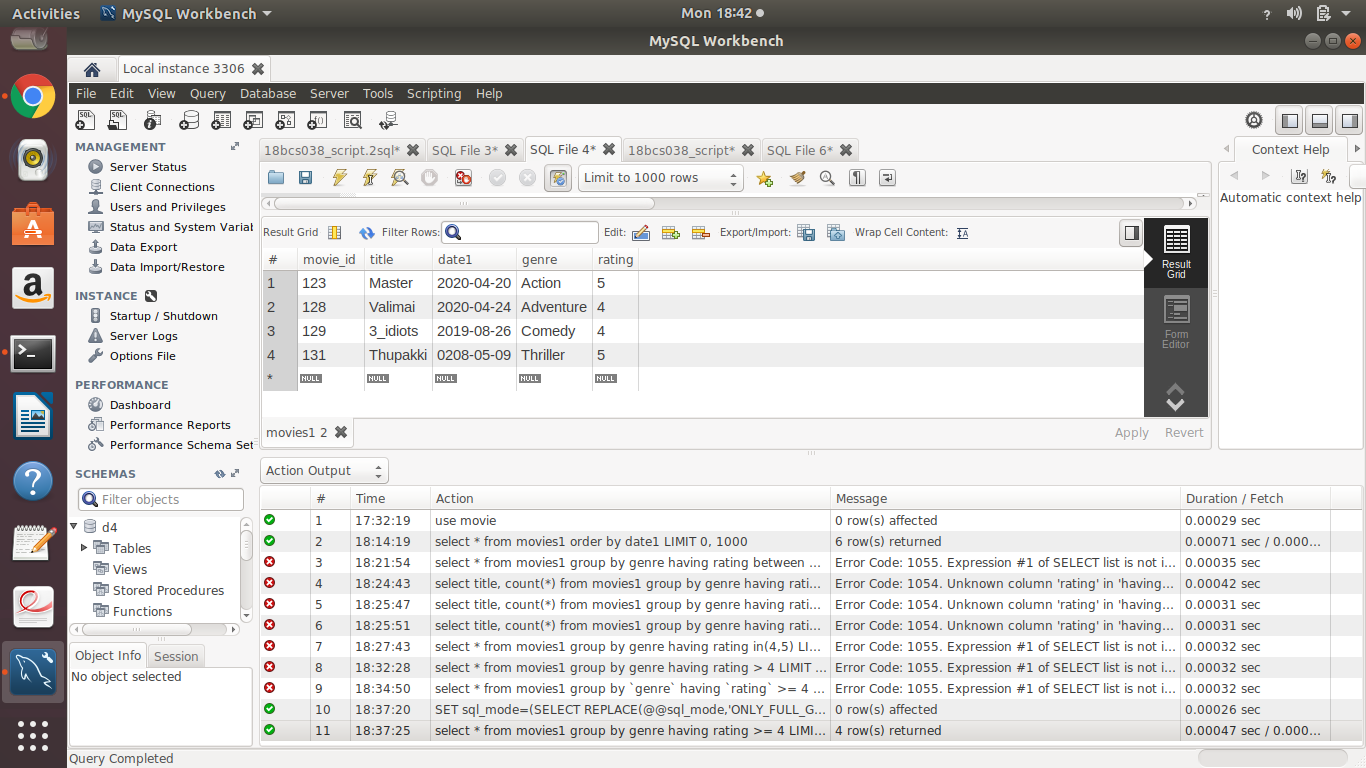
1. Order by clause.

Ans. select \* from movies1 order by date1;



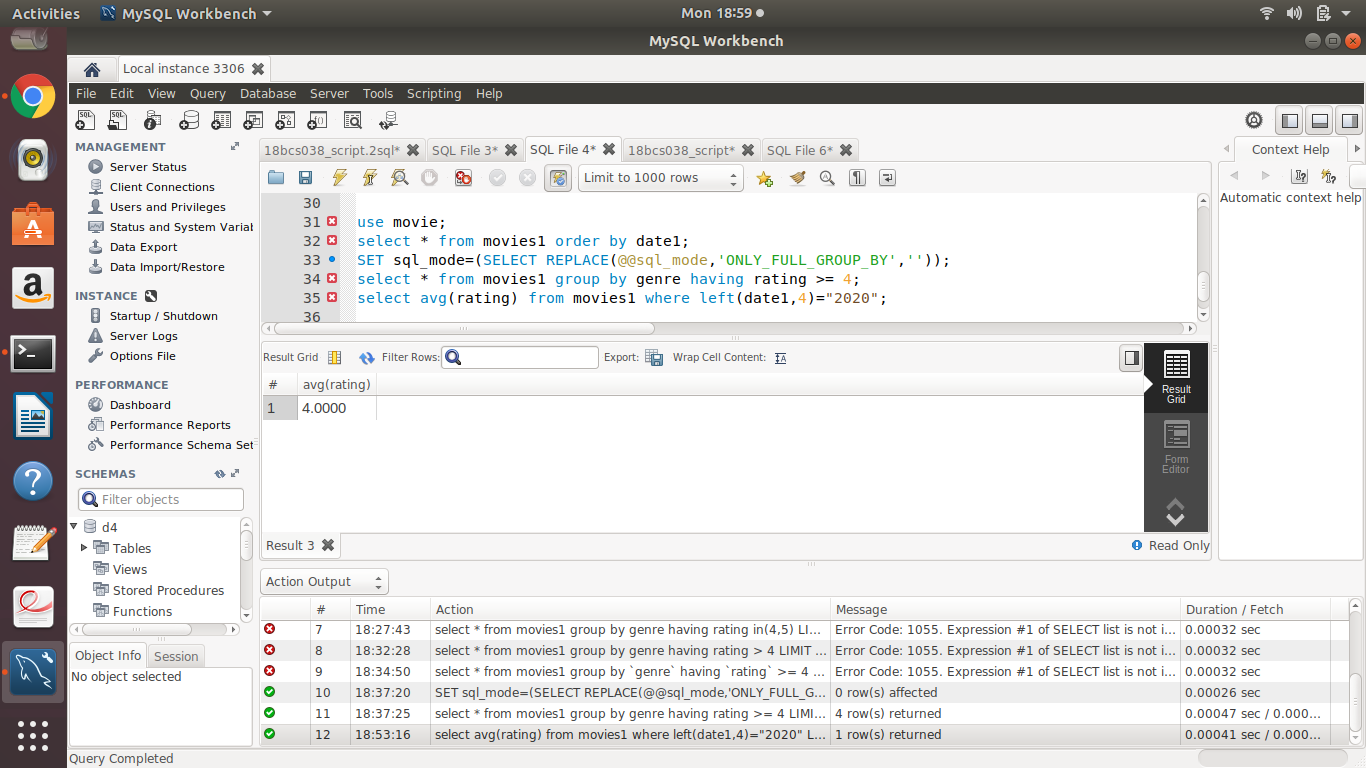
1. Group by and having.

Ans. select \* from movies1 group by genre having rating >= 4;



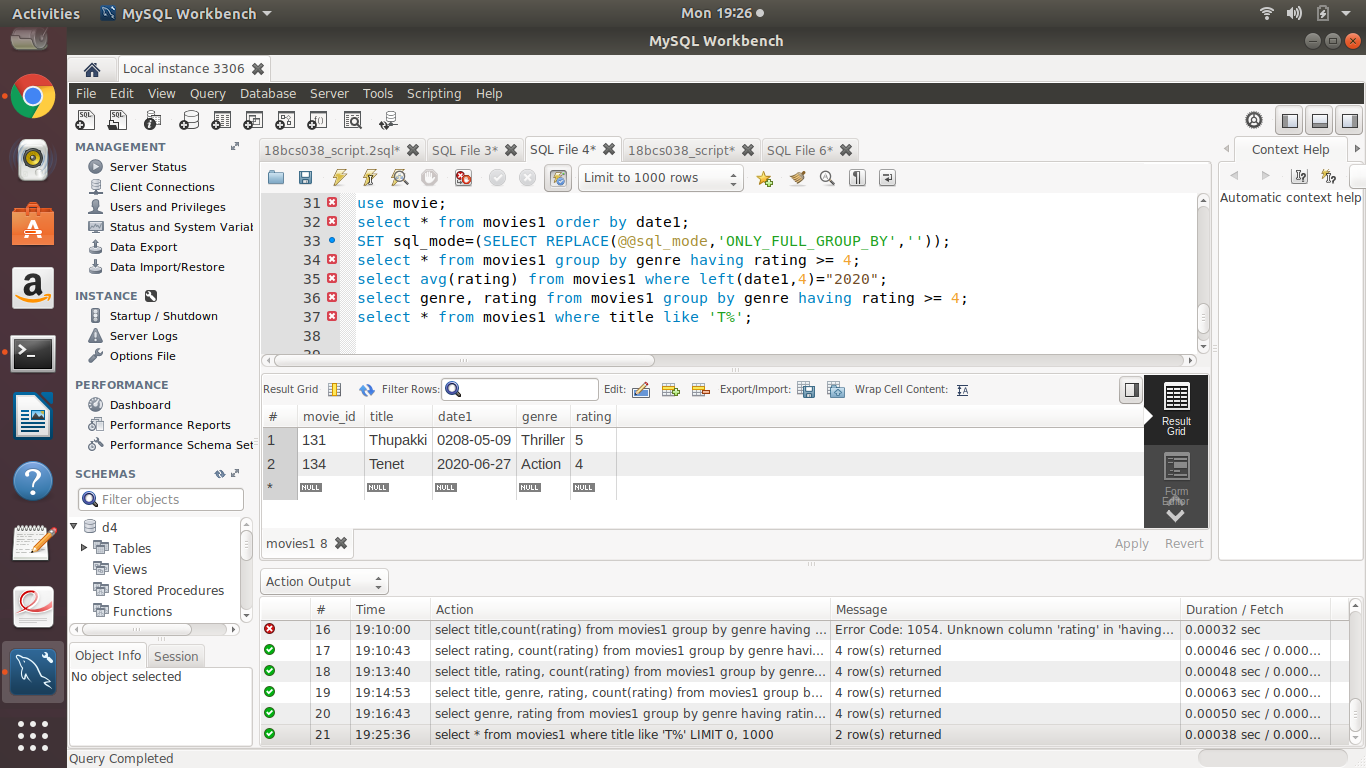
1. Aggregate functions.

Ans. select avg(rating) from movies1 where left(date1,4)="2020";



1. Logical operators especially with LIKE.

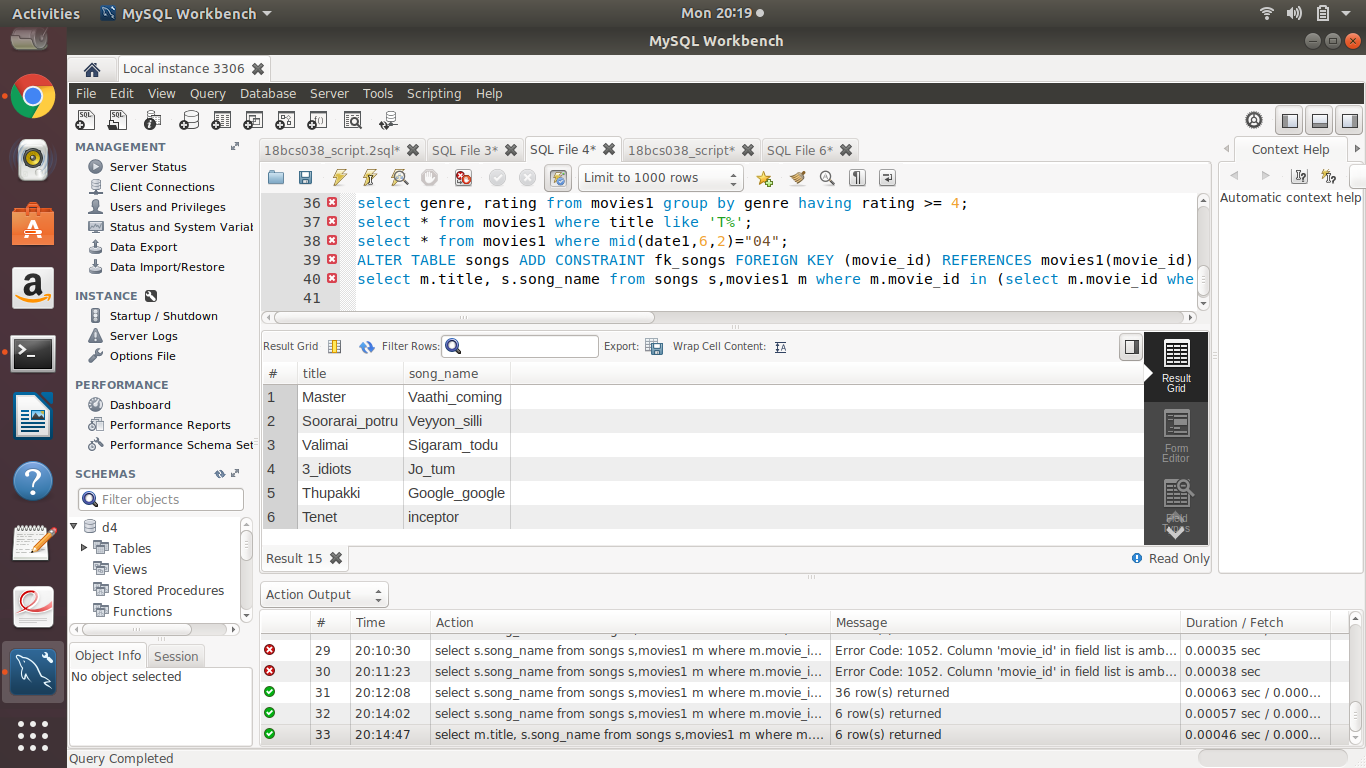
Ans. select \* from movies1 where title like 'T%';



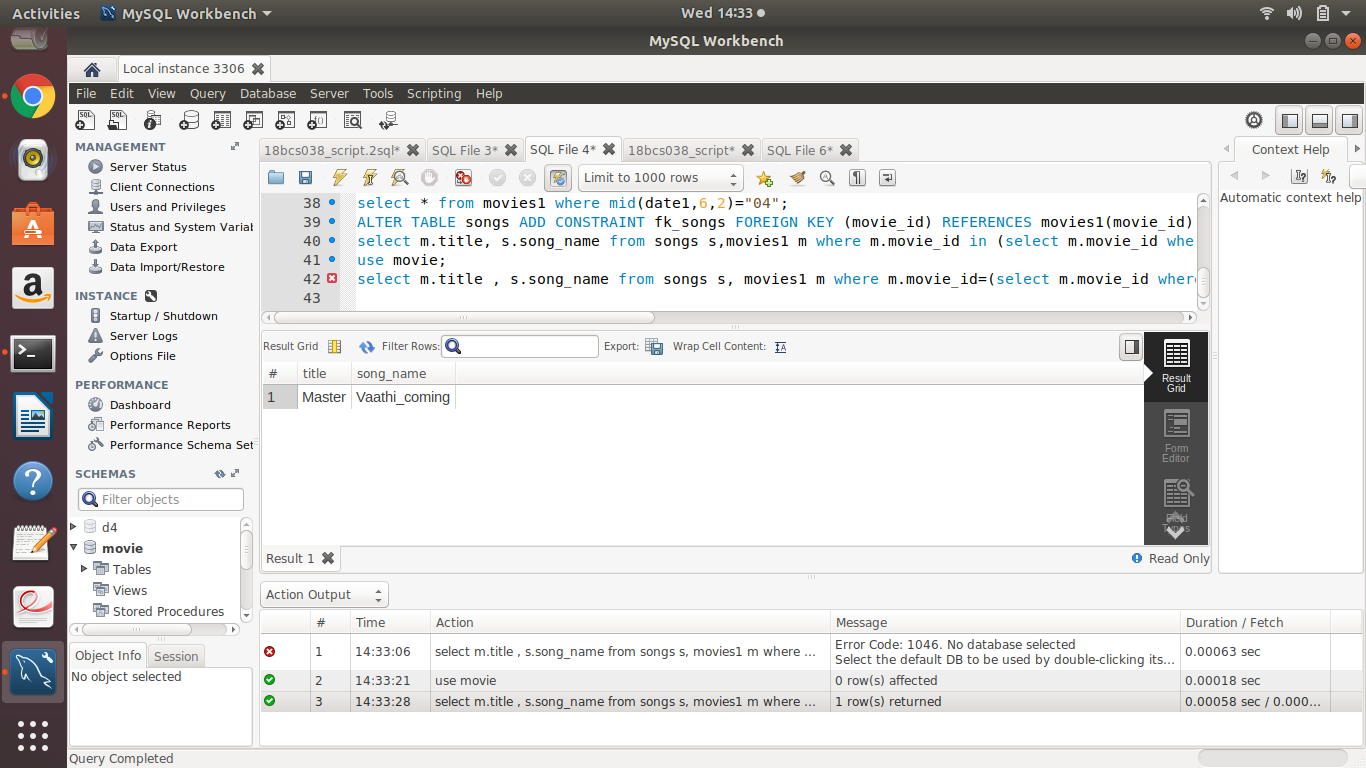
1. At least 4 nested queries specific to your Database, out of which at least 2 should have multiple subquery.

Ans.

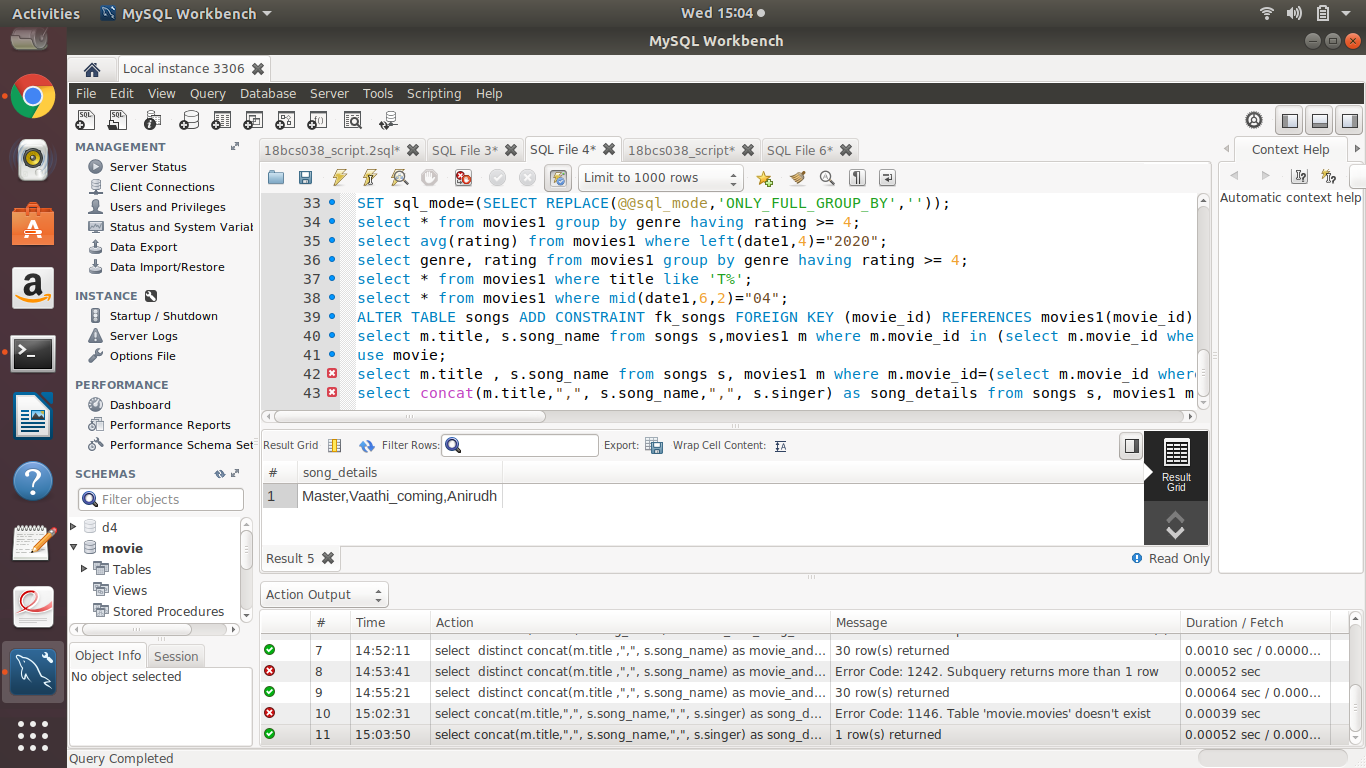
A. select m.title, s.song\_name from songs s,movies1 m where m.movie\_id in (select m.movie\_id where m.movie\_id=s.movie\_id);



B. select m.title , s.song\_name from songs s, movies1 m where m.movie\_id=(select m.movie\_id where m.movie\_id=s.movie\_id and m.title="Master");



C. select concat(m.title,",", s.song\_name,",", s.singer) as song\_details from songs s, movies1 m where m.movie\_id in( select m.movie\_id where m.movie\_id=s.movie\_id and m.title in(select m.title where m.title="Master"));



D. select concat(m.title,",", s.song\_name,",", s.singer,",",composer) as artist\_involved\_in\_action\_movies from songs s, movies1 m where m.movie\_id in(select m.movie\_id where m.movie\_id=s.movie\_id and m.genre in(select m.genre where m.genre="Action"));

