**Find the names and addresses of all employees**

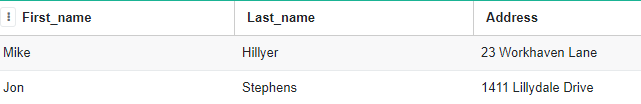
select emp.first\_name, emp.last\_name, addr.address

from employee emp

left join addr

on emp.addr\_id = addr.addr\_id;

employee.first\_name, employee.last\_name, addr.address employee ⟕ employee.addr\_id = addr.addr\_id addr



**Find the sum of transaction amounts of each employee in the year 2020**

select emp.first\_name, emp.last\_name, sum(trans.amount)

from employee emp

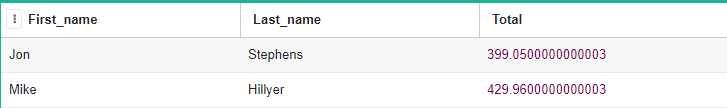
left join payment trans

on emp.employee\_id = trans.employee\_id

WHERE strftime('%Y',trans.payment\_date) == '2020'

group by emp.first\_name, emp.last\_name;

γ employee.first\_name, employee.last\_nameemployee.first\_name, employee.last\_name, sum(amount) σ year(payment.payment\_date) == 2005 (employee ⟕ employee.employee\_id = payment.employee\_id payment)



**Find all movies along with the number of actors listed in that movie, where the number of listed actors is > 7**

select mv.title, count(\*) number\_of\_actors

from movie mv

inner join movie\_actor mv\_act

on mv.movie\_id = mv\_act.movie\_id

group by mv.title having count(\*) > 7

order by number\_of\_actors desc;

τnum\_actors desc (σ num\_actors > 7 (γ movie.title σmovie.title, num\_actors<- count(\*) (movie ⟕movie.movie\_id = movie\_actor.movie\_id movie\_actor)))



**Find the highest paying customers, i.e. customers who have spent more than $125. Display their name along with the total amount paid**

select cust.first\_name, cust.last\_name, sum(trans.amount) 'Total Amount Paid'

from payment trans

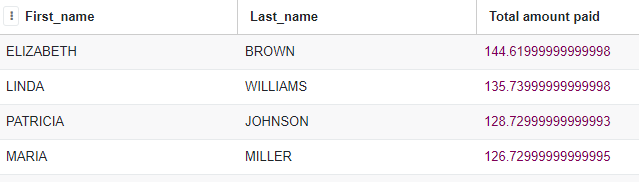
join buyer cust

on trans.buyer\_id = cust.buyer\_id

group by cust.first\_name, cust.last\_name HAVING SUM(trans.amount) > 125

order by sum(trans.amount) DESC;

τ Amount desc (σ Amount > 125 γ buyer.first\_name, buyer.last\_name σbuyer.first\_name, buyer.last\_name, Amount <- sum(payment.amount) (payment ⟕ payment.buyer\_id = buyer.buyer\_id buyer))



**Find all action movies**

select movie\_id, title

from movie

where movie\_id in (

select movie\_id

from movie\_category

where category\_id in (

select category\_id

from category

where name = 'Action'

)

);

ρ M (movie) ; ρ MC (movie\_category) ; ρ A (category)

π M.movie\_id, M.title σ C.name = ‘Action’ ^ C.category\_id = MC.category\_id ^ MC.movie\_id = M.movie\_id ’ (M x MC x C)



**Find First name and last name of the actors in the movie ‘Deer Virginian’**

select first\_name, last\_name

from actor

where actor\_id in (

select actor\_id

from movie\_actor

where movie\_id in (

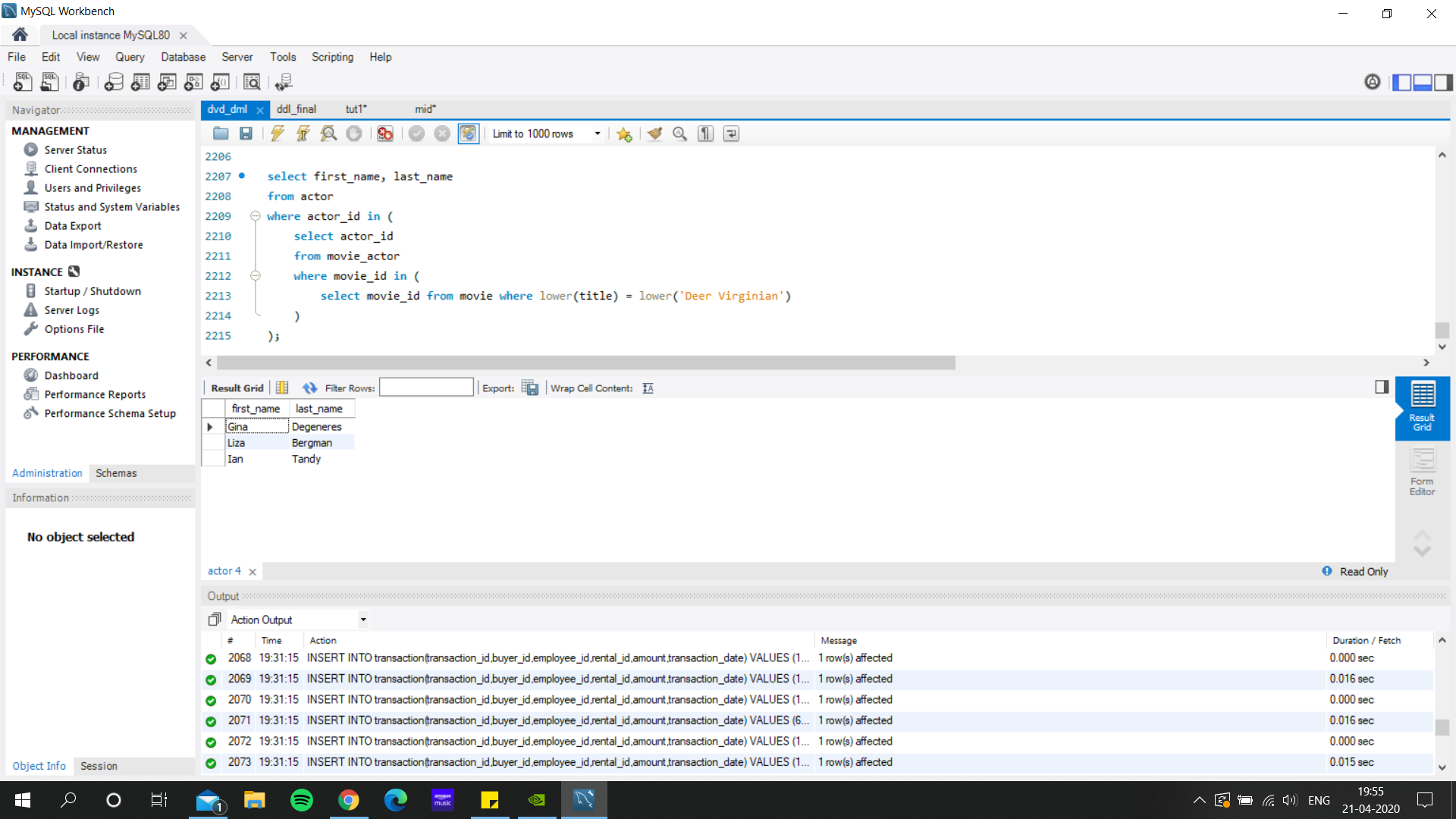
select movie\_id from movie where lower(title) = lower('Deer Virginian')

)

);

ρ M (movie) ; ρ MA (movie\_actor) ; ρ A (actor)

π A.first\_name, A.last\_name σ M.title = ‘Deer Virginian’ ^ M.movie\_id = MA.movie\_id ^ MA.actor\_id = A.actor\_id ’ (M x MA x A)

****

**Find the movie title and its release year having the actor “Alec Wayne”**

select title, release\_year

from movie

where movie\_id in (

select movie\_id

from movie\_actor

where actor\_id in (

select actor\_id

from actor

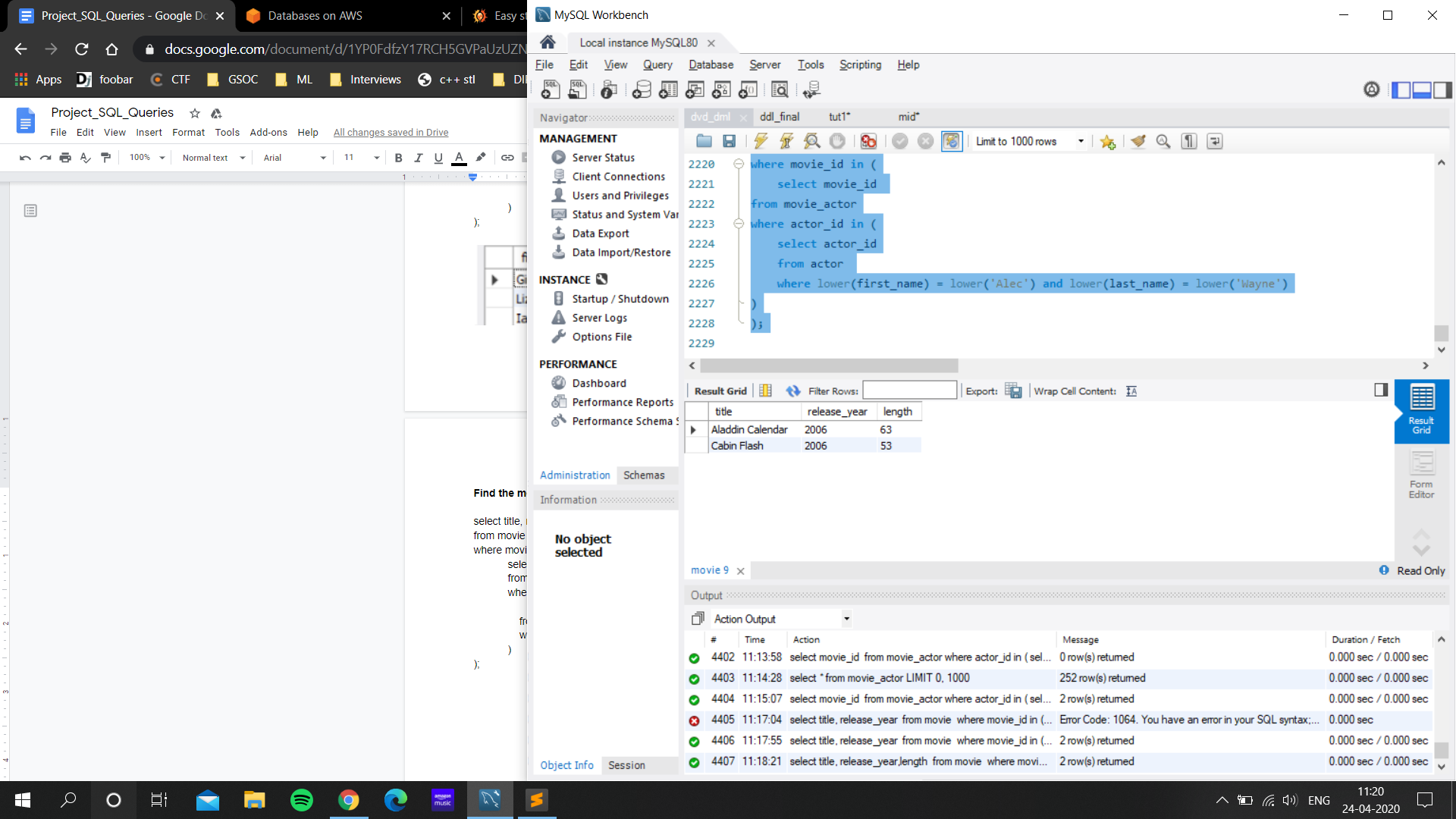
where lower(first\_name) = lower('Alec') and lower(last\_name) = lower('Wayne')

)

);

ρ M (movie) ; ρ MA (movie\_actor)

π M.title, M.release\_year σ M.movie\_id = MA.movie\_id ^ MA.first\_name = ‘Alec’ ^MA.last\_name = ‘Wayne’ (M x MA)



**Find the movie having length lesser than 60 mins i.e. 1 hour and view them in decreasing order.**

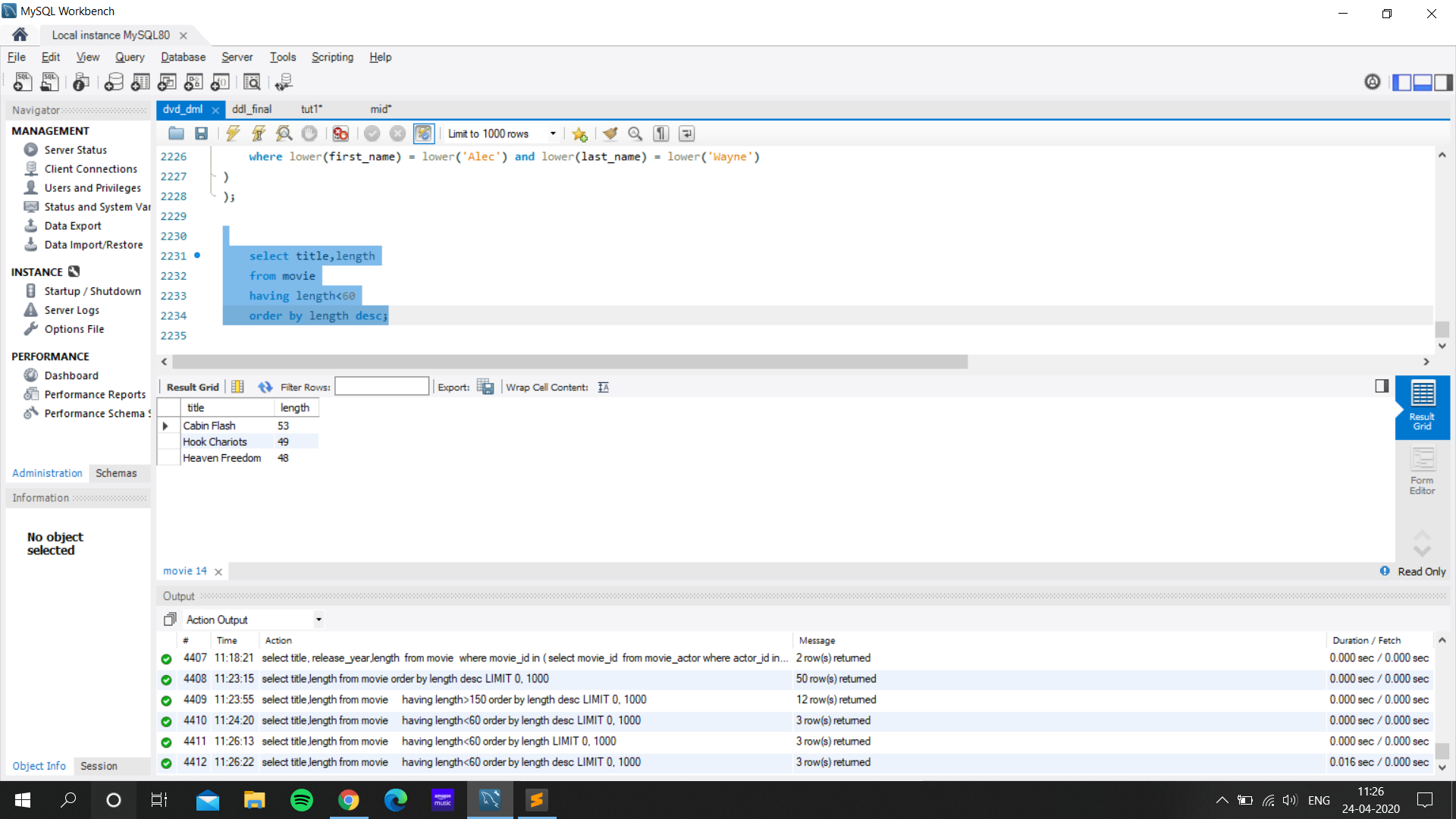
select title,length

from movie

Where length<60

order by length desc;

τ length desc π title, length σ length < 60 movie

****

**Find the first name, last name, and emails of the customers who bought movies on 24-05-2005**

select first\_name,last\_name,email

from buyer

where buyer\_id in (

select buyer\_id

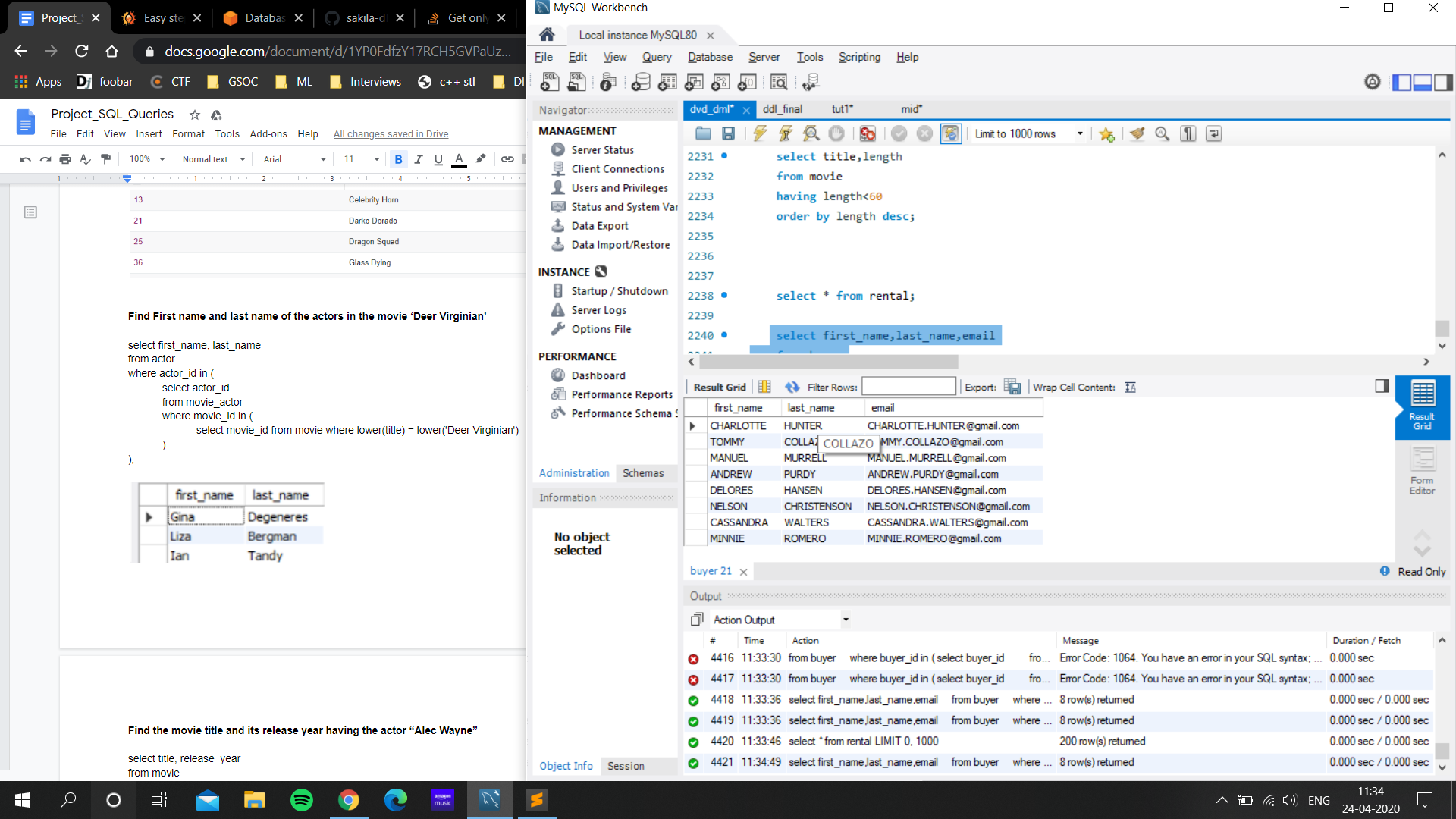
from rental

where date(rental\_date) = "2005-05-24"

);

ρ B1 (buyer) ; ρ R (Rental)

π B1.first\_name, B1.last\_name, B1.email σ R.buyer\_id = B1.buyer\_id ^ date(R.rental\_date) = ‘2005-05-24’ (B1 x R)

****

**Find the name of movies and number of times it has been rented till date**

select mv.title, count(\*) rented\_times

from movie mv

inner join rental r

on mv.movie\_id = r.movie\_id

group by r.movie\_id having count(\*) >0

order by rented\_times desc;

τ rented desc (σ rented > 0 γ movie.titleσ movie.title, rented <- count (\*) (movie ⨝movie.movie\_id = rental.movie\_id rental))

**Find movies with the same category of a person**

select \* from movie where movie\_id in (

select movie\_id from movie\_category where category\_id in (

select category\_id from movie\_category where movie\_id in(

select movie\_id

from rental

where buyer\_id in(

select buyer\_id from buyer where lower(email)= lower('MARIA.MILLER@gmail.com')

))

)

);

ρ B1 (buyer) ; ρ R (Rental) ; ρ M (movie) ; ρ MC (movie\_category)

π M.title (σ B1.email = ‘maria.miller@gmail.com’ ^ B1.buyer\_id = R.buyer\_id ^ R.movie\_id = MC.movie\_id ^ MC.movie\_id = M.movie\_id (B1 x R x M x MC)

**-- a query to find all orders of user id 1**

select title, movie\_id from movie where movie\_id in(

select movie\_id from rental where buyer\_id = 1

);

**-- a query to find all active orders for user id1**

select movie\_id, title, return\_date from movie, rental where movie.id = rental.id and buyr.id = 1 and return\_date>today;

**Display each movie name with its category name**

select mov.title, cat.name from movie mov left join movie\_category mcat on mov.movie\_id = mcat.movie\_id left join category cat on mcat.category\_id = cat.category\_id;

---------------------------- FOR ML DEMO QUERIES (Don’t change)----------------------------------

select mov.title, cat.name from movie mov left join movie\_category mcat on mov.movie\_id = mcat.movie\_id left join category cat on mcat.category\_id = cat.category\_id;

select \* from buyer;

select title from movie where movie\_id in

(select movie\_id

from rental

where buyer\_id in(

select buyer\_id from buyer where lower(email)= lower('MARIA.MILLER@gmail.com')

)

);

select \* from rental;