In the lecture he used the Exists function in the query:

SELECT first\_name, last\_name

From customer AS c

WHERE EXISTS

(SELECT \* FROM payment as p

WHERE p.customer\_id = c.customer\_id

AND amount > 11)

He did not explain it in the lecture but this query is called a correlated subquery. This query is the same as the query:

SELECT first\_name, last\_name

FROM customer

WHERE customer.customer\_id IN

(SELECT payment.customer\_id

FROM payment

WHERE payment.amount > 11)

It finds all the customer names who have made at least one payment with amount > 11. In the lectures he said that a sub query is executed first, so then how does the first query work? I googled it and the first query is called a correlated subquery where what happens for a correlated sub query is that the query is executed again for every row in the outer query table. A correlated sub query uses the outer table in the sub query, this turns it from a normal sub query into a correlated sub query. A normal sub query is just executed once first but a correlated sub query is executed for every row in the outer queries table. The EXIST then returns true when there is one row in the inner query that matches and that row from the outer query is added to the output result.

Here is a good youtube video on Correlated Subqueries called “Understanding Correlated Queries in SQL” at URL: <https://www.youtube.com/watch?v=SM9cDMxAeK4>

In my head what I thought the first query, the correlated query was doing was this query:

SELECT first\_name, last\_name

From customer

WHERE EXISTS

(SELECT \*

FROM payment INNER JOIN customer

ON payment.customer\_id = customer.customer\_id

WHERE amount > 11)

I though it would just look for payments wit amount greater than 11 and then if one exists just return the whole table. But it doesn’t do that. The correlated subquery returns 8 rows. The IN subquery returns 8 rows. The third query returns 599 rows.