**11-9 notes**

Advanced SQL Queries

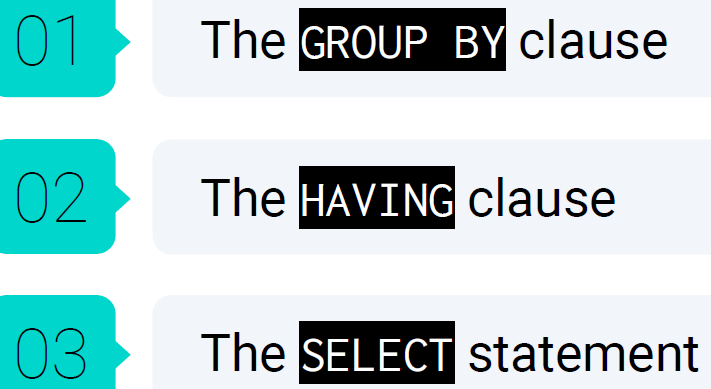
Goals:

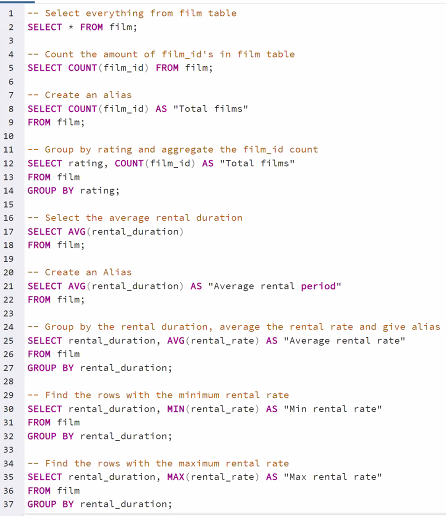
* Create aggregate queries.
* Create subqueries to further explore data.
* Create views and run subqueries off of them.

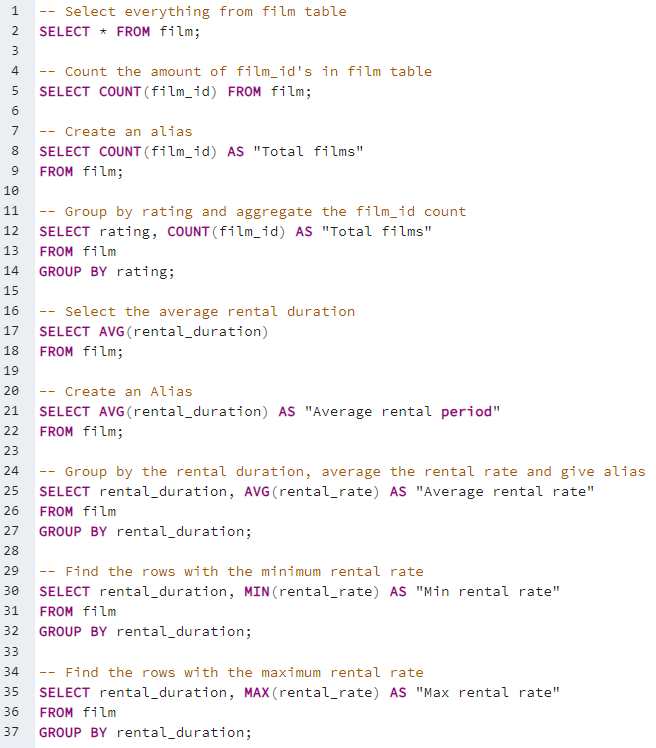
Aggregate functions allow you to perform a calculation on a set of values to return a single value.

* AVG Calculates the average of a set of values
* COUNT Counts the rows in a specific table or view
* MIN Returns the minimum value in a set of values
* MAX Returns the maximum value in a set of values
* SUM Calculates the sum of a set of values

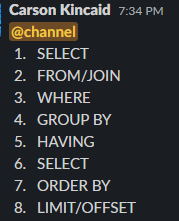
Aggregate Functions are often used with:



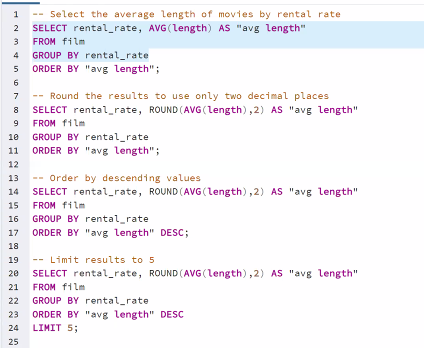




Order of SQL steps, per Carson:

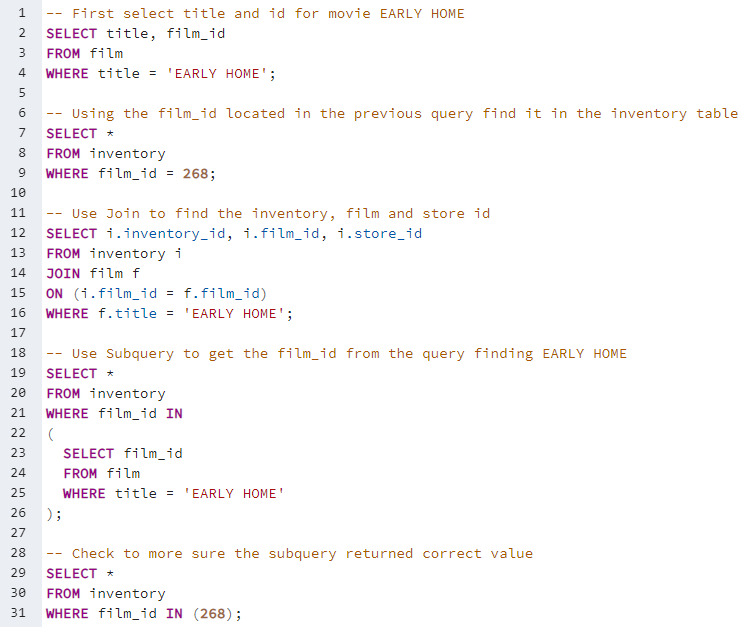


**Ordering by Aggregates solution:**



Getting the top five means using Order By and then Limit x(number you want to limit to).

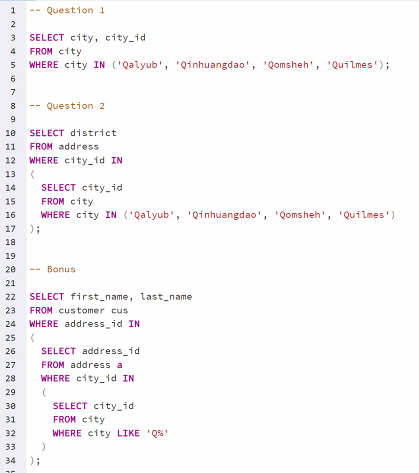
**Subqueries Solution:**



^ Line 31 IN is functioning like an equal sign.

*Joins are more efficient for processing time, but sometimes subqueries are necessary when there’s no common column name*

**Stu Subqueries solution:**



^ Line 32 uses % as a wild character. It’ll pick up any records that start with Q.

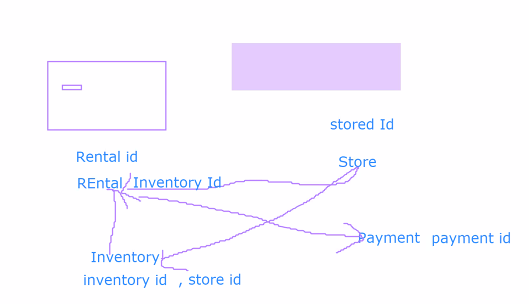
SQL will work from the innermost query, then the middle queries, then the outermost query.

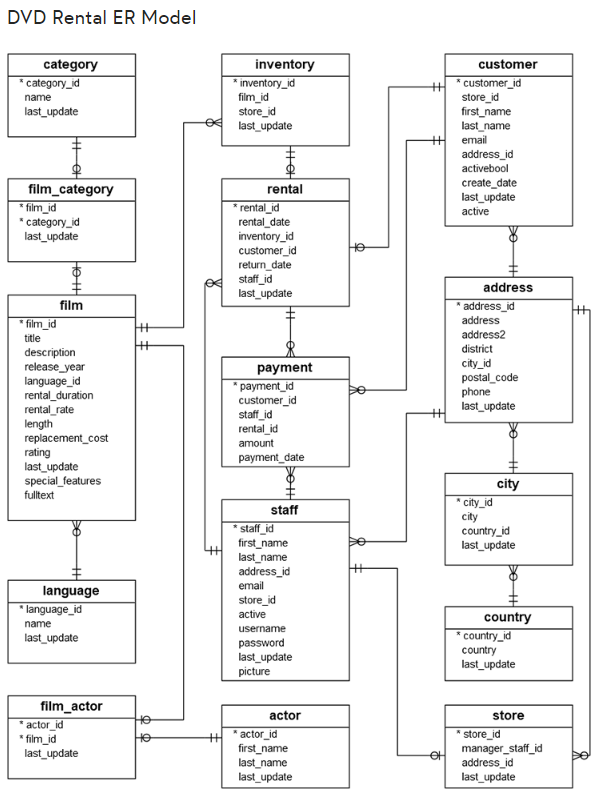
Views are a visual table (object) that you can make from tables or other views.

To view details of a table, R click on table, select Scripts, and Select. This code will generate the results for you.

Postgres cannot create an image showing how tables are linked on which columns like Access can. You need to know how they’re linked before you dig in to the tables to merge them to a single place. You can always look at the column headers to see what they have in common. Entity Relationship Diagram <https://www.postgresqltutorial.com/postgresql-getting-started/postgresql-sample-database/>

* One to many, many to many, many to one
  + Crows foot notation for the symbols on the lines in the diagram

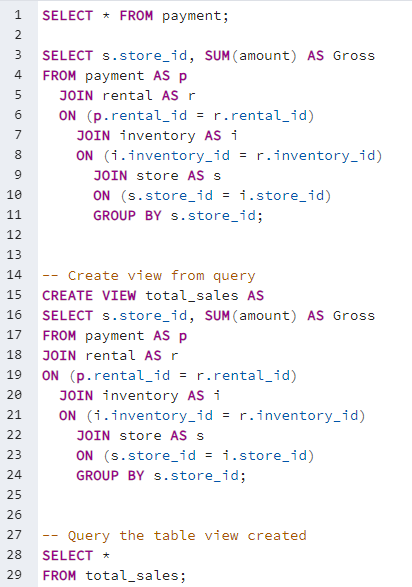




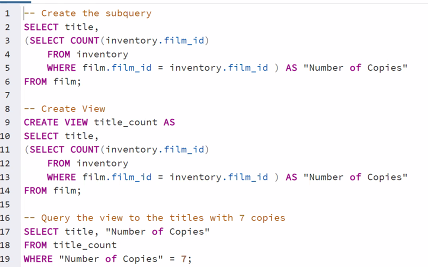
You can save a common query as a view to be able to refresh the details easier.

A view works like a table (it’s in the Left navigation pane).

* Except it pulls the most up to date data each time you run it.



**Room full of Queries solution:**



^ Line 5- You can write a query and have those results entered to the table as a column.

So lines 2-6 are a nested query. Lines 2 and 6 select the title, and lines 3-5 selects the number of copies in inventory.

Alternatively, you can use the join below to get the same result as lines 2-6.

* Joins are clean and easier to understand.



**Stu\_Mine\_the\_Subquery Solution:**

