W07 Paper

Implicit groups are when you perform an aggregation in a query without a group by statement. It looks at all the rows in the query and will perform the aggregation for each option in the SELECT statement. Explicit groups are when you specifically tell the SQL to query by a specific group, whether it be a category or a factor.

When you don’t count by a distinct value in a query, it’ll return each instance a certain value appears. For example, if there are 5 observations of the actor\_Id of 1, and you use distinct, it’ll only return only a single instance of that id. However, if you did a basic count function, it would return all 5 cases of the actor\_id.

Expressions are powerful when used in an aggregation because you can cut down the amount of code that you use. It’ll iterate through each row with whatever aggregation is being used. It makes it so that you don’t have to write extra bits of code outside of the aggregation or in a separate query.

The only time null values are counted are when you are counting the number of rows, or observations, within a query. However, when not counting rows, SQL will skip over the null values, and it won’t be part of any aggregation. This is useful because you don’t need to worry about null values affecting the results of aggregations that are used.

The main difference between single and multiple column grouping is the number of columns that are in the GROUP BY statement. If you’re only wanting a single column to be grouped, then you just put that column in the statement. When doing multiple columns in the statement, you’ll put whatever column you want to group by first, and then any columns that follow. The query will follow the order you set for the statement.