Grouping Data

* Overview:
  + In order to use aggregate functions most effectively, you will probably need to group the results
    - GROUP BY indicates the dimensions by which you want to group your data (e.g. a category to sort into subgroups.
    - A GROUP BY clause works on the rows returned by the query by summarizing identical rows into a single / distinct group and returns a single row with the summary for each group, by using appropriate aggregate functions in the SELCT list, like COUNT(), SUM(), MIN(), MAX(), AVG(), ect.
    - Aggregates are typically over measurement fields: sales, miles, heights, ect.
    - GROUP BY is over the dimensions that are being measured
  + In addition to GROUP BY, you can filter the resulting aggregation with adding a HAVING clause
    - HAVING is how you can filter measueres you have aggregated (e.g. where a SUM is over a certain value)
    - The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions
    - A HAVING clause in SQL specifies that a SQL SELECT statement should only return rows where aggregate values meet the specified conditions
* GROUP BY and HAVING
  + Using aggregate functions might not give you all the information you are looking for, as they return only a single value from one column
  + What would we do if we wanted to know how many films are in each genre?
    - We could try to use the COUNT() function to answer rthis question
      * But first, we’d have to figure out what generes exist in the “genre” column, using DISTINCT

SELECT DISTINCT genre

FROM films;



* + - Or, we could use the COUNT() together with a WHERE clause for each genre:

SELECT COUNT(\*)

FROM films;

127

SELECT COUNT(\*)

FROM films;

WHERE genre = 'Sci-Fi';

39

SELECT COUNT(\*)

FROM films;

WHERE genre = 'Animation';

92

* + - But this quickly turns into a very tedious process.
    - The ebst way to see the number of films per genre in our table is by using the GROUP BY clause
      * GROUP BY gather alike rows together into one summary row
      * For example, to view the number of films by genre, we would type:

SELECT genre, COUNT(\*)

FROM films;

GROUP BY genre;

* + - * Which would return:

