SQL in a nutshell (2)

Another common database task is aggregation. That is, reducing your data by breaking it into chunks and summarizing each chunk.

This is done in SQL using the `GROUP BY` command. This command breaks your data into groups and applies a function from your `SELECT` statement to each group.

For example, if you wanted to count the number of flights from each of two origin destinations, you could use the query

```
SELECT COUNT(*) FROM flights
GROUP BY origin;
```

`GROUP BY origin` tells SQL that you want the output to have a row for each unique value of the `origin` column. The `SELECT` statement selects the values you want to populate each of the columns. Here, we want to `COUNT()` every row in each of the groups.

It's possible to `GROUP BY` more than one column. When you do this, the resulting table has a row for every combination of the unique values in each column. The following query counts the number of flights from SEA and PDX to every destination airport:

```
SELECT origin, dest, COUNT(*) FROM flights
GROUP BY origin, dest;
```

The output will have a row for every combination of the values in `origin` and `dest` (i.e. a row listing each origin and destination that a flight flew to). There will also be a column with the `COUNT()` of all the rows in each group.

Remember, a more in depth look at SQL can be found **here**.

What information would this query get? Remember the `flights` table holds information about flights that departed PDX and SEA in 2014 and 2015. Note that `AVG()` function gets the average value of a column!