# Group Summary Statistics: Takeaways ⤴

## Syntax

- Computing summary statistics by a unique value in a row:

```
SELECT SUM(Employed)

FROM recent_grads

GROUP BY Major_category;
```

- Filtering results after aggregation:

```
SELECT Major_category, AVG(Employed) / AVG(Total) AS share_employed

FROM recent_grads

GROUP BY Major_category

HAVING share_employed > 0.8;
```

- Rounding a column to two decimal places:

```
SELECT Major_category, ROUND(ShareWomen, 2) AS rounded_share_women

FROM recent_grads;
```

- Converting, known as casting, a column to a float type:

```
SELECT CAST(Women as Float) / CAST(Total as Float)

FROM recent_grads;
```

- Using if/then logic in SQL:

```
CASE

WHEN <condition_1> THEN <value_1>

WHEN <condition_2> THEN <value_2>

ELSE <value_3>

END AS <new_column_name>
```

- High-level structure of a query:

```
SELECT column(s)

  FROM some_table

 WHERE some_condition

 GROUP BY column(s)

HAVING some_condition

 ORDER BY column(s)

 LIMIT some_limit;
```

- The order by which SQL runs the high-level structure above:
  - `FROM`
  - `WHERE`
  - `GROUPBY`
  - `HAVING`
  - `SELECT`
  - `ORDERBY`
  - `LIMIT`

## Concepts

- The `GROUPBY` clause allows you to compute summary statistics by group.
- The `HAVING` clause filters on the virtual column that `GROUPBY` generates.
- `WHERE` filters results before the aggregation, whereas `HAVING` filters after aggregation.
- The `ROUND` function rounds the results to desired decimal places.
- The `CAST` function in SQL converts data from one data type to another. For example, we can use the `CAST` function to convert numeric data into character string data.

## Resources

[Core functions of SQLite](#)