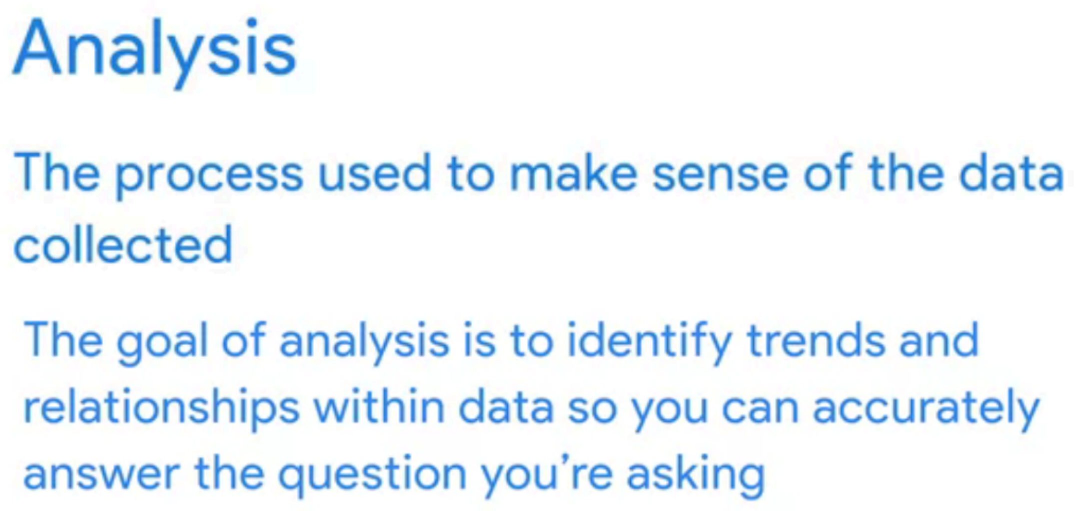
Analyze Data to Answer Questions

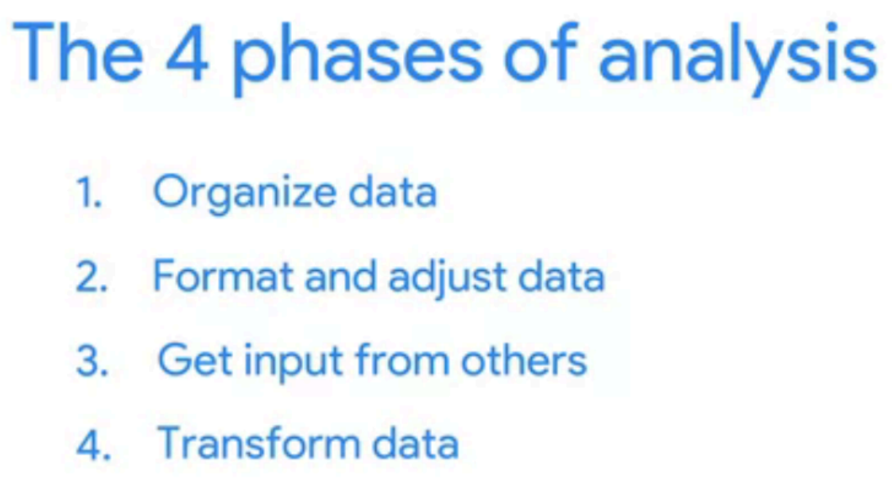
★**Organize data to begin analysis**



Analysis

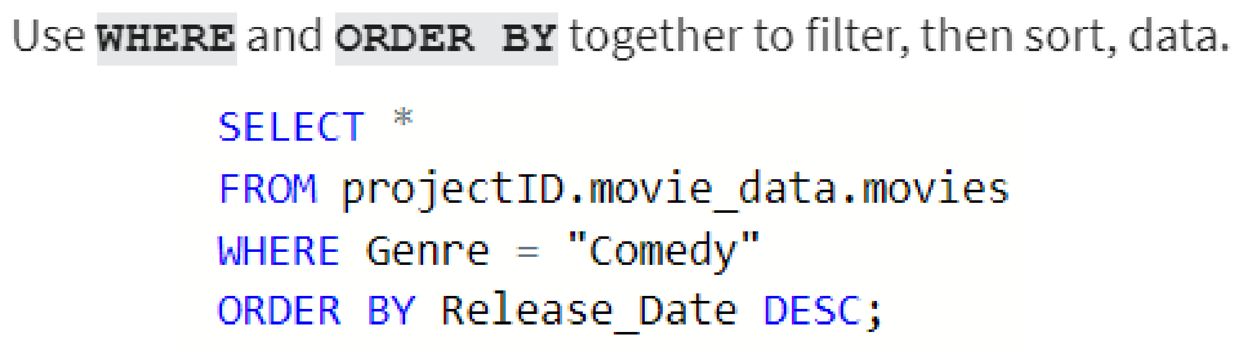
The process used to make sense of the data collected

The goal of analysis is to identify trends and relationships within data so you can accurately answer the question you're asking



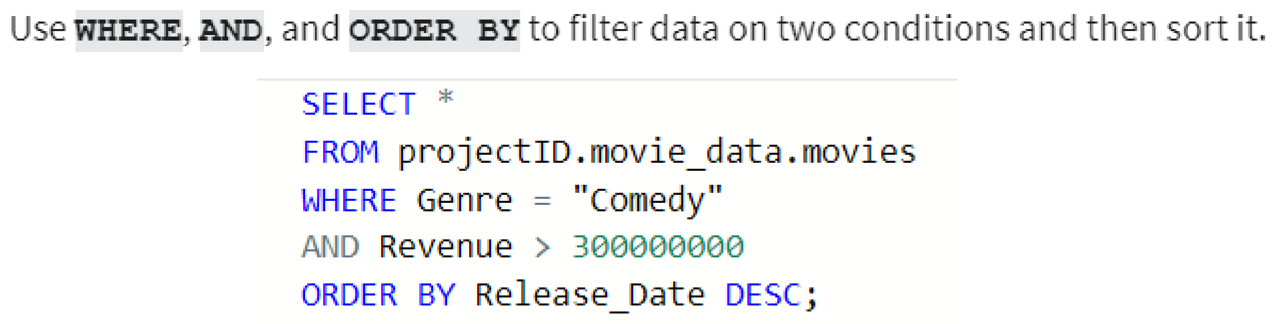
The 4 phases of analysis

1. Organize data
2. Format and adjust data
3. Get input from others
4. Transform data



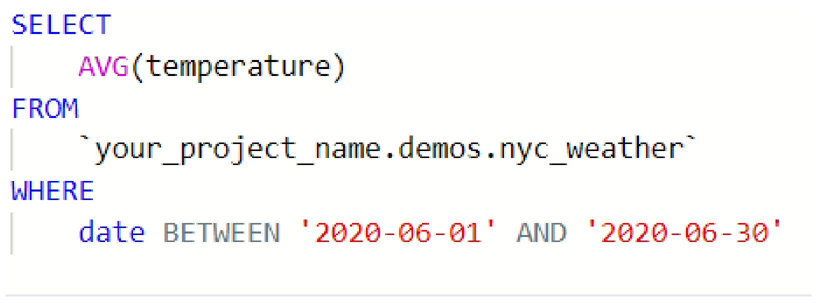
Use `WHERE` and `ORDER BY` together to filter, then sort, data.

```
SELECT *
FROM projectID.movie_data.movies
WHERE Genre = "Comedy"
ORDER BY Release_Date DESC;
```



Use `WHERE`, `AND`, and `ORDER BY` to filter data on two conditions and then sort it.

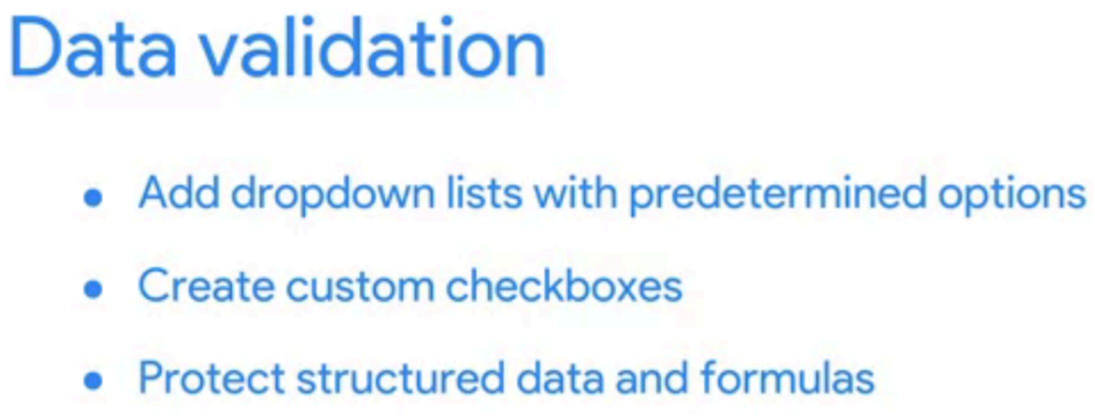
```
SELECT *
FROM projectID.movie_data.movies
WHERE Genre = "Comedy"
AND Revenue > 300000000
ORDER BY Release_Date DESC;
```



```
SELECT
    AVG(temperature)
FROM
    `your_project_name.demos.nyc_weather`
WHERE
    date BETWEEN '2020-06-01' AND '2020-06-30'
```

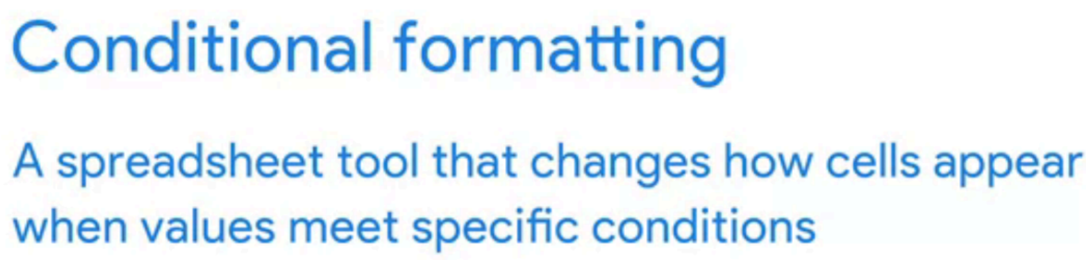
★ **Format and adjust data**

- ○ ***Features Inside Spreadsheet(Google Sheet):***



Data validation

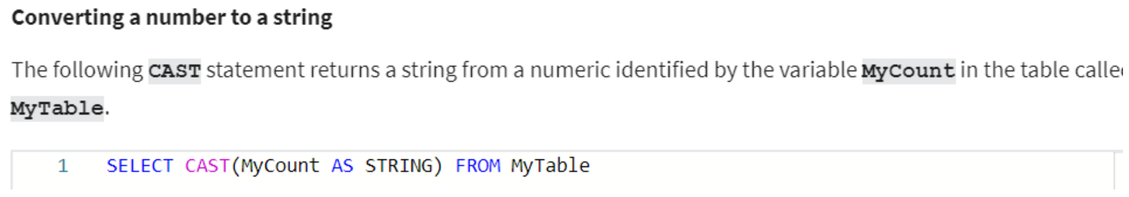
- Add dropdown lists with predetermined options
- Create custom checkboxes
- Protect structured data and formulas



Conditional formatting

A spreadsheet tool that changes how cells appear when values meet specific conditions

- ***Features in SQL:***

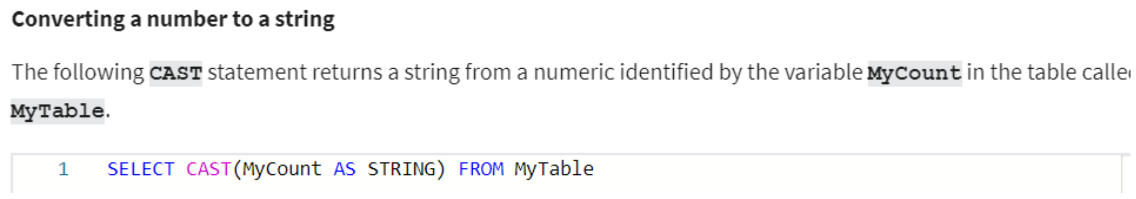


Converting a number to a string

The following `CAST` statement returns a string from a numeric identified by the variable `MyCount` in the table calle

`MyTable`.

```sql
SELECT CAST(MyCount AS STRING) FROM MyTable
```

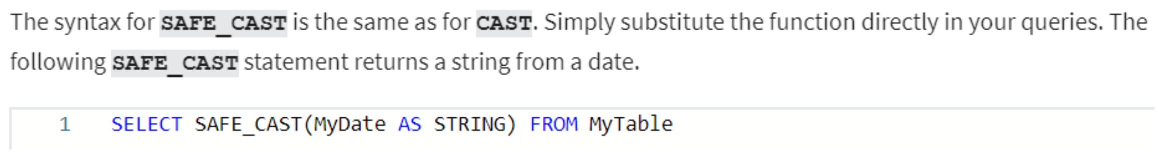


Converting a number to a string

The following `CAST` statement returns a string from a numeric identified by the variable `MyCount` in the table calle

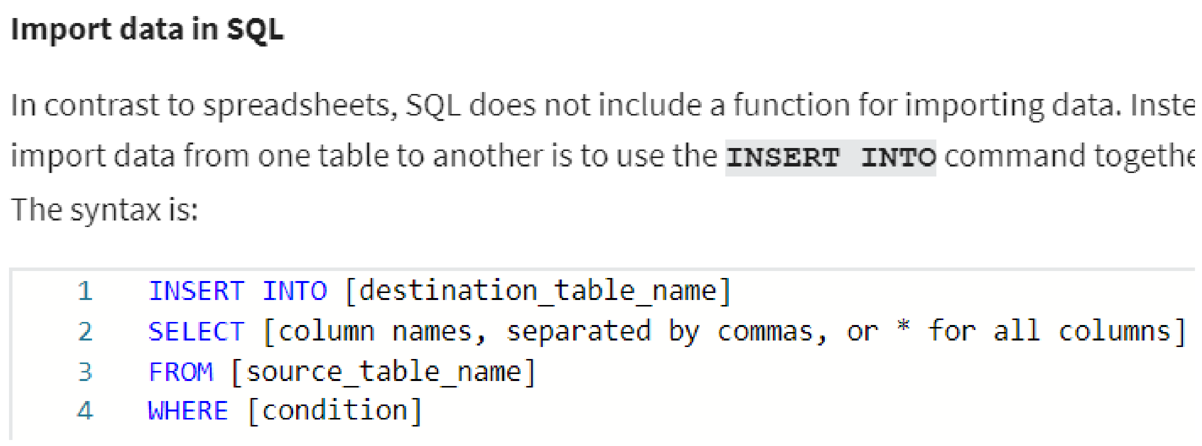
`MyTable`.

```sql
SELECT CAST(MyCount AS STRING) FROM MyTable
```



The syntax for `SAFE_CAST` is the same as for `CAST`. Simply substitute the function directly in your queries. The following `SAFE_CAST` statement returns a string from a date.

```sql
SELECT SAFE_CAST(MyDate AS STRING) FROM MyTable
```



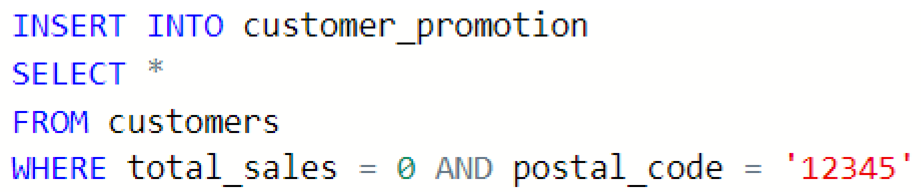
Import data in SQL

In contrast to spreadsheets, SQL does not include a function for importing data. Inste

import data from one table to another is to use the `INSERT INTO` command togethe

The syntax is:

```sql
INSERT INTO [destination_table_name]
SELECT [column names, separated by commas, or * for all columns]
FROM [source_table_name]
WHERE [condition]
```



```sql
INSERT INTO customer_promotion
SELECT *
FROM customers
WHERE total_sales = 0 AND postal_code = '12345'
```

Combine data in SQL

In SQL, use the **`CONCAT`** function to join strings tog
to improve the readability of reports (such as comk
customer list). Or, you might combine data to gene
syntax:

```
SELECT CONCAT(field1, " ", field2)
FROM [table_name]
```

```
SELECT CONCAT(field1, " ", field2) AS alias
FROM [table_name]
```

Concatenate strings with SQL

Review the table below as a summary of the **`CONCAT`** function and its variations in SQL.

Function/ operator	**Use**	**Example**	**Result**
`CONCAT`	Concatenate strings to create new text strings	`CONCAT('Google', '.com')`	`Google.com`
`CONCAT_WS`	Concatenate two or more strings together with a separator between each string	`CONCAT_WS(' . ', 'www', 'google', 'com')`	`www.google.com`
`||`	Concatenate two or more strings together with the `||` operator	`'Google' || '.com'`	`Google.com`

★ **Aggregate data for analysis**

Data aggregation

The process of gathering data from multiple sources in order to combine it into a single summarized collection

★ ***Data Aggregation in Spreadsheet(Google Sheet)***

The VALUE function converts the text string to a numerical value.

VLOOKUP

A spreadsheet function that vertically searches for a certain value in a column to return a corresponding piece of information

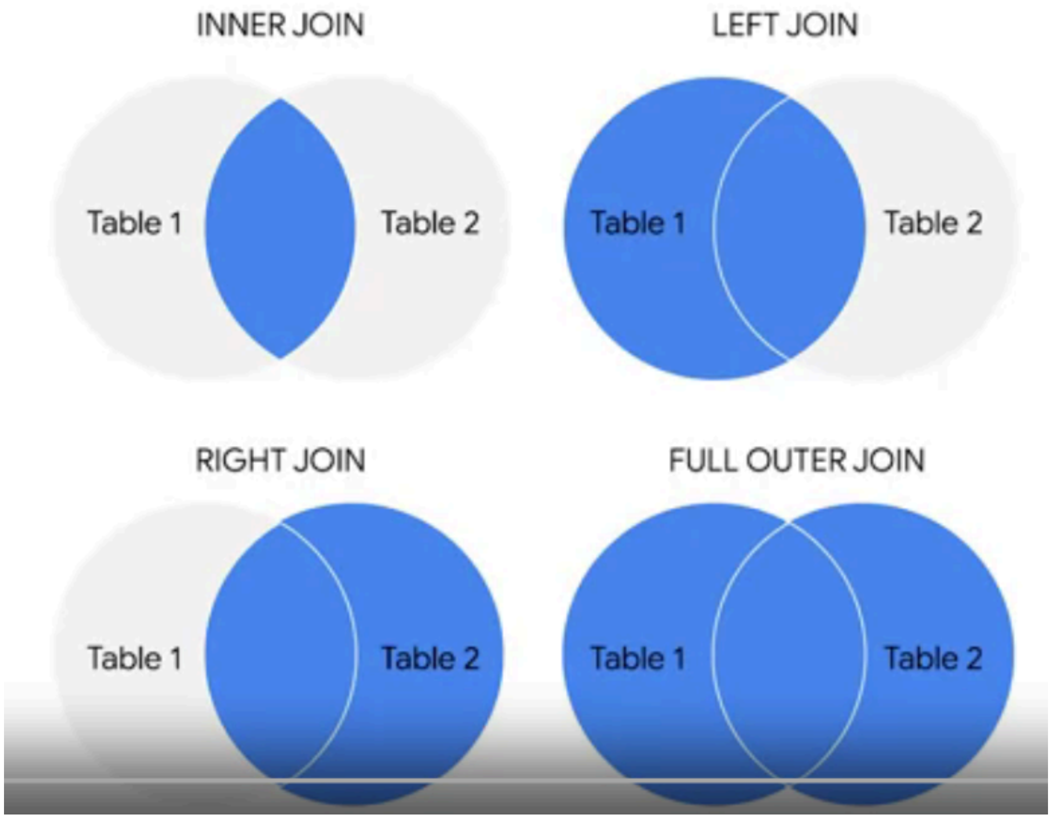
`VLOOKUP`'s syntax is:

```
VLOOKUP(search_key, range, index, is_sorted)
```

★ ***Data Aggregation in SQL***

JOIN

A SQL clause that is used to combine rows from two or more tables based on a related column



- **`INNER JOIN`:** a function that returns records with matching values in both tables
- **`LEFT JOIN`:** a function that returns all the records from the left table (first mentioned) and only the matching records from the right table (second mentioned)
- **`RIGHT JOIN`:** a function that returns all records from the right table (second mentioned) and only the matching records from the left table (first mentioned).
- **`OUTER JOIN`:** a function that combines the **`RIGHT JOIN`** and **`LEFT JOIN`** to return all matching records in both tables.

```
SELECT
   -- table columns from tables are inserted here
   table_name1.column_name
   table_name2.column_name
FROM
   table_name1
JOIN
   table_name2
ON table_name1.column_name = table_name2.column_name
```

```sql
SELECT
    employees.name AS employee_name,
    employees.role AS employee_role,
    departments.name AS department_name
FROM
    employee_data.employees
LEFT JOIN
    employee_data.departments ON
    employees.department_id = departments.department_id
```

Clauses like HAVING and CASE, paired with subqueries, will help you build more and more complex queries

Use a subquery in a `SELECT` statement

Use a subquery in a `FROM` statement

```sql
SELECT employee_id
FROM employees
WHERE department_id IN (SELECT department_i
FROM departments
WHERE location_id = 1000)
```

```sql
SELECT price
FROM sales
WHERE price = (SELECT MAX (salary)
FROM sales)
```

★ **Perform data calculations**

- ***Common functions in Spreadsheet(Google Sheet)***

```
=COUNTIF(range,“value”)
```

```
=SUMIF(range,criteria/condition,[sum_range])
```

```
=AVERAGEIF(range, criteria, [sum_range])
```

```
fx | =MAXIFS(D2:D21,B2:B21,"NY",E2:E21,"<400")
```

```
=sumproduct(array1, [array2]....)
```

Pivot tables

Let you view data in multiple ways to find insights and trends

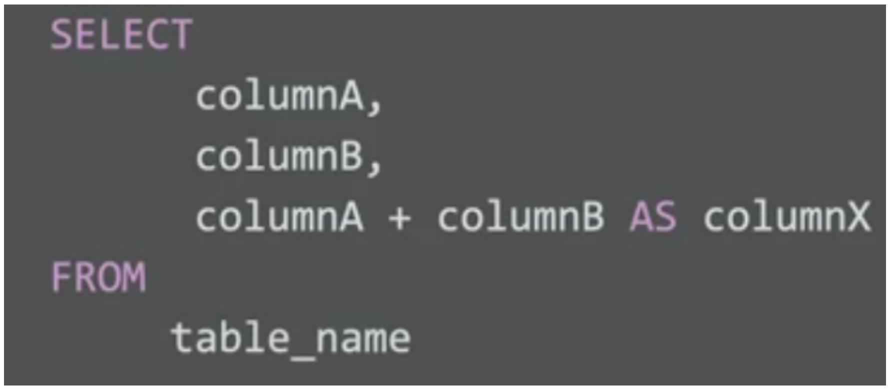
Calculated field

A new field within a pivot table that carries out certain calculations based on the values of other fields

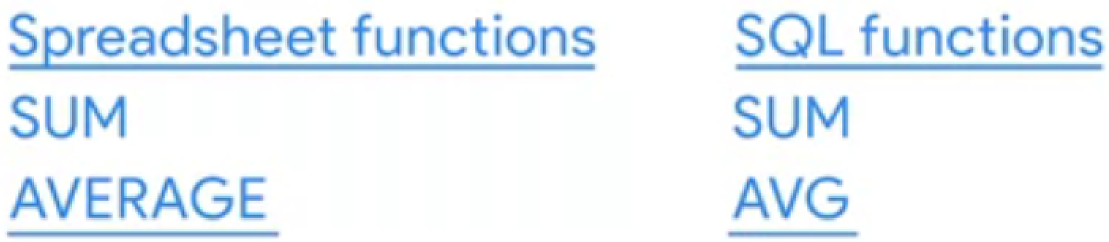
Data Validation

Applying a Filter to Clean Data

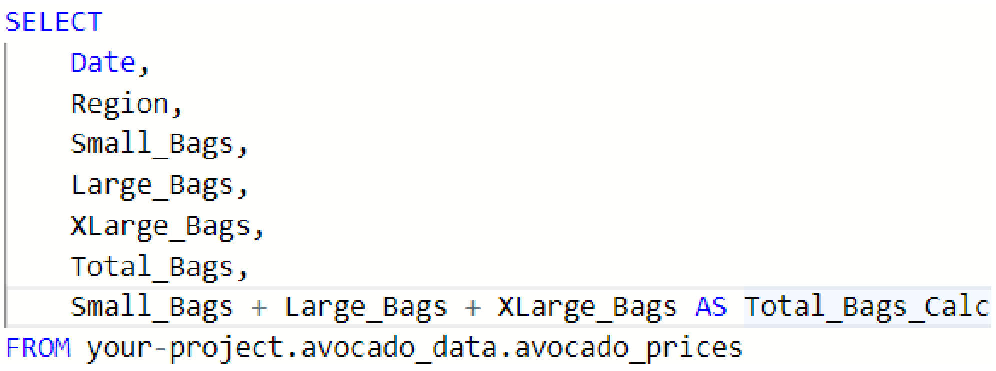
- ○ ***Common Calculation formulas/functions in SQL***



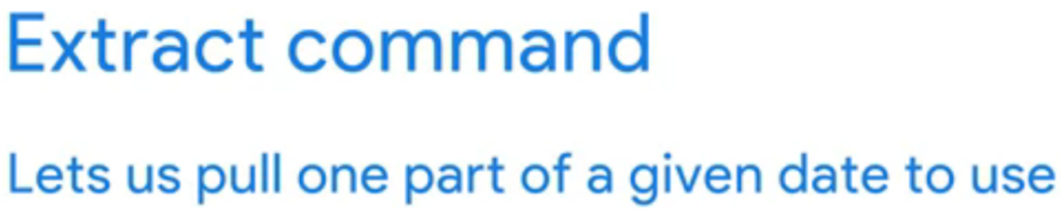
```
SELECT
        columnA,
        columnB,
        columnA + columnB AS columnX
FROM
       table_name
```



Spreadsheet functions	SQL functions
SUM	SUM
AVERAGE	AVG

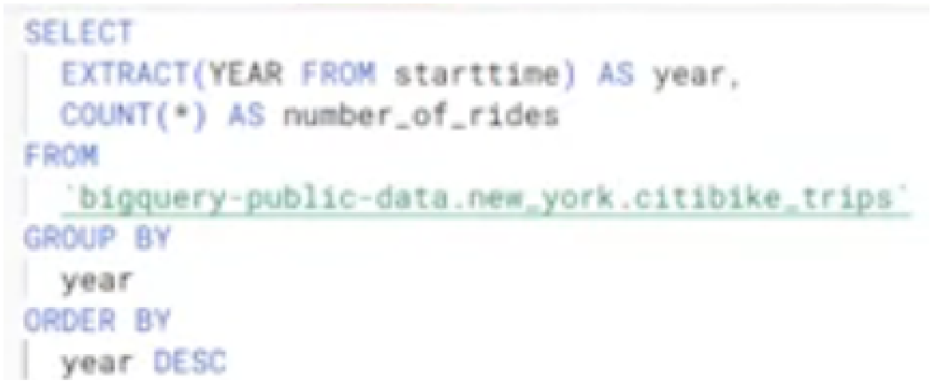


```
SELECT
    Date,
    Region,
    Small_Bags,
    Large_Bags,
    XLarge_Bags,
    Total_Bags,
    Small_Bags + Large_Bags + XLarge_Bags AS Total_Bags_Calc
FROM your-project.avocado_data.avocado_prices
```



Extract command

Lets us pull one part of a given date to use

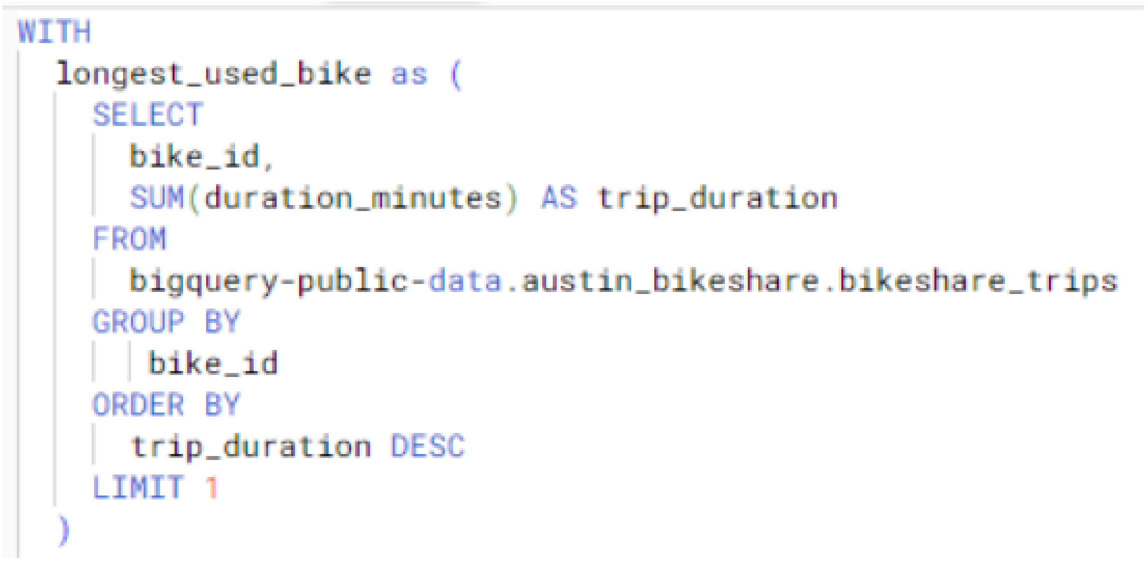


```
SELECT
  EXTRACT(YEAR FROM starttime) AS year,
  COUNT(*) AS number_of_rides
FROM
  `bigquery-public-data.new_york.citibike_trips`
GROUP BY
  year
ORDER BY
  year DESC
```

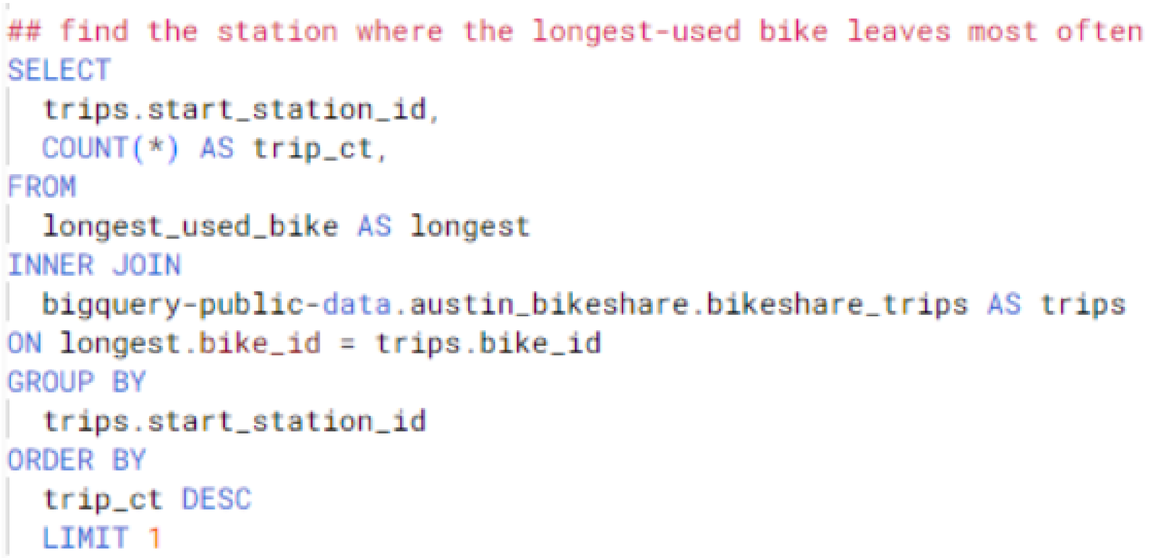
Temporary table

A database table that is created and exists temporarily on a database server

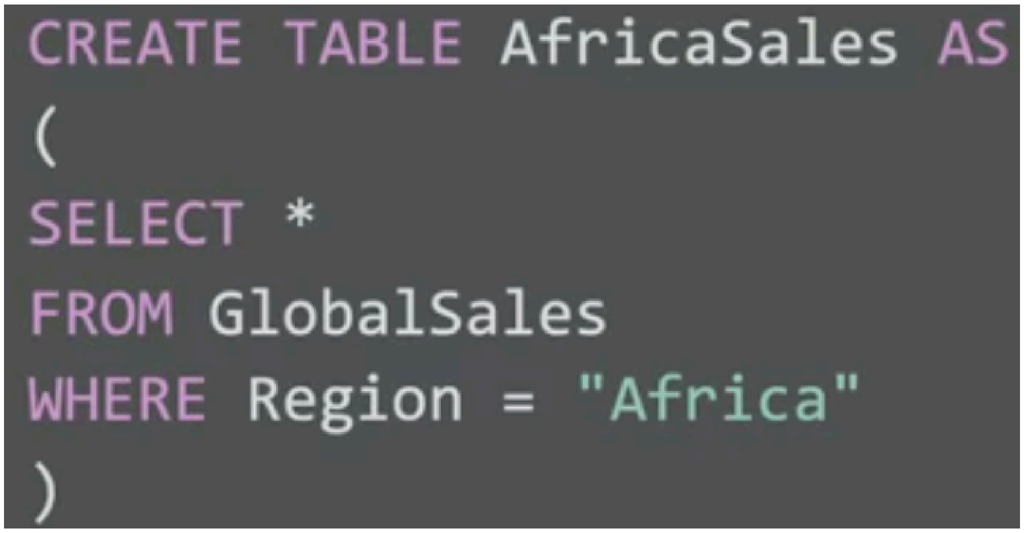
The WITH clause is a type of temporary table that you can query from multiple times



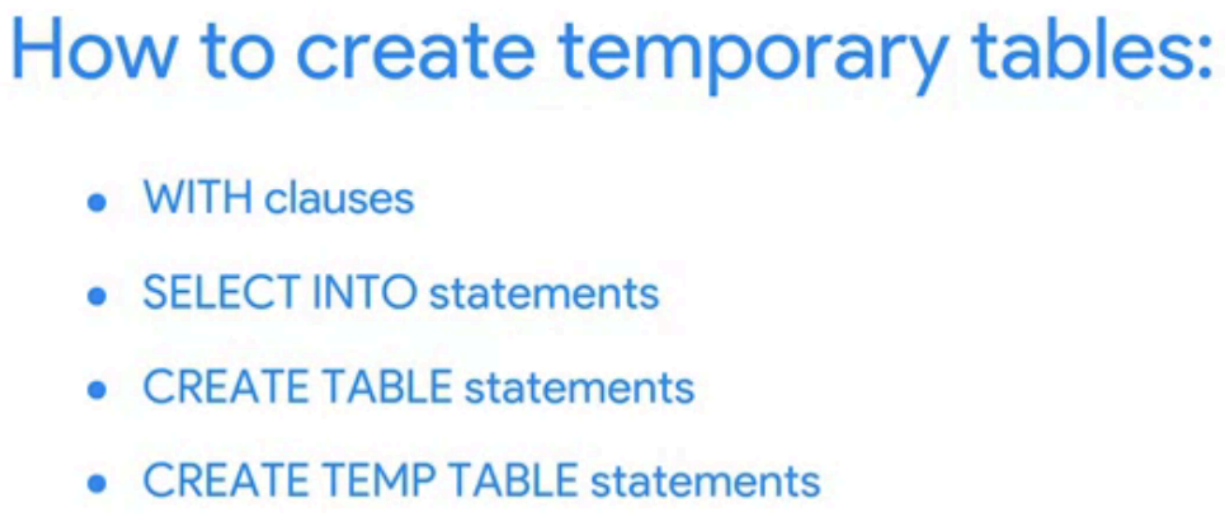
```sql
WITH
  longest_used_bike as (
    SELECT
      bike_id,
      SUM(duration_minutes) AS trip_duration
    FROM
      bigquery-public-data.austin_bikeshare.bikeshare_trips
    GROUP BY
      bike_id
    ORDER BY
      trip_duration DESC
    LIMIT 1
  )
```



```sql
## find the station where the longest-used bike leaves most often
SELECT
  trips.start_station_id,
  COUNT(*) AS trip_ct,
FROM
  longest_used_bike AS longest
INNER JOIN
  bigquery-public-data.austin_bikeshare.bikeshare_trips AS trips
ON longest.bike_id = trips.bike_id
GROUP BY
  trips.start_station_id
ORDER BY
  trip_ct DESC
  LIMIT 1
```

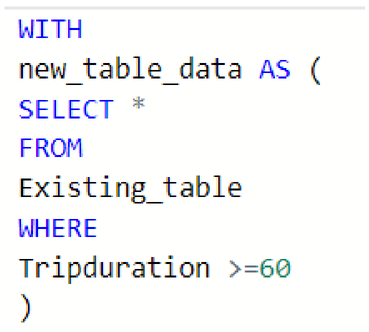


```
CREATE TABLE AfricaSales AS
(
SELECT *
FROM GlobalSales
WHERE Region = "Africa"
)
```

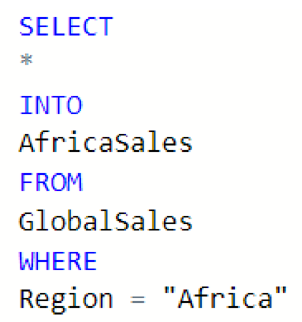


How to create temporary tables:

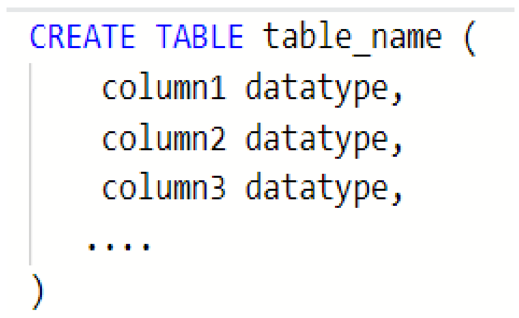
- WITH clauses
- SELECT INTO statements
- CREATE TABLE statements
- CREATE TEMP TABLE statements



```
WITH
new_table_data AS (
SELECT *
FROM
Existing_table
WHERE
Tripduration >=60
)
```



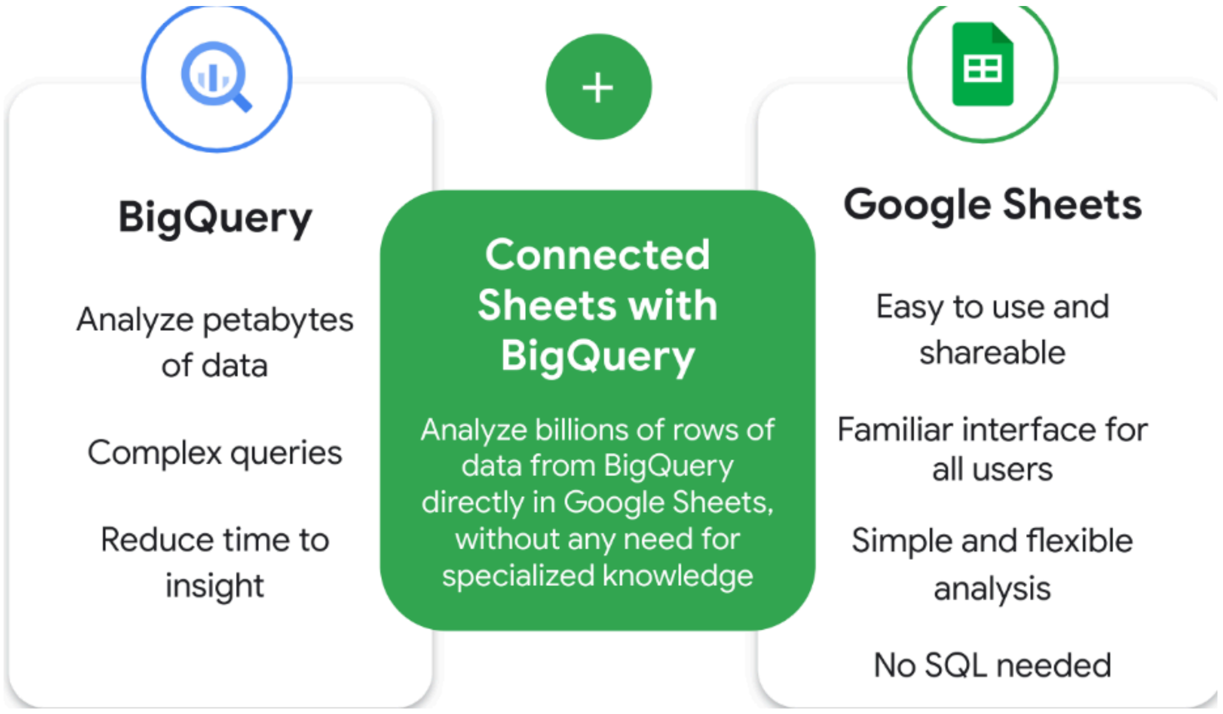
```
SELECT
*
INTO
AfricaSales
FROM
GlobalSales
WHERE
Region = "Africa"
```



```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
)
```

Link to Intermediate SQL Guide:

https://d3c33hcgiwev3.cloudfront.net/BsaAoIwwQLKGgKCMMOCyFw_d522c0a682164c5dbaa4e2b507f01df1_Your-Intermediate-Guide-to-SQL.pdf?Expires=1712620800&Signature=Bxq7fqqDVKhtOgDMSF96xExUHfwzMO1suW-0FKo-umNcniu29Bh8B~Q2Btbg6gCMpyrnJxgSWoFBUD~pymhDX9Zr-KvWHDXXRLR18tv3rt~zetwEihALEgl3AmyKsVwt0z3wSDvwkF7qJcW7SFEIG~ihX0sRJLcs6RmO91HxZm4_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A


BigQuery
Analyze petabytes of data
Complex queries
Reduce time to insight
+
Connected Sheets with BigQuery
Analyze billions of rows of data from BigQuery directly in Google Sheets, without any need for specialized knowledge
Google Sheets
Easy to use and shareable
Familiar interface for all users
Simple and flexible analysis
No SQL needed