**Google Data Analytics**

## **Analyze Data to Answer Questions**

**Analysis:**

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

Database **organisation** enables analysts to make decisions about which data is relevant to pull for a specific analysis. It also helps them decide which data types and variables are appropriate

**Two ways of organising:**

**Sorting** is when you arrange data into a meaningful order to make it easier to understand, analyse, and visualise

**Filtering** is used when you are only interested in seeing data that meets a specific criteria, and hiding the rest

**Data Validation:**

Allows you to control what can and can't be entered in your worksheet

* Add a drop-down list to a column with predetermined options
* Create custom checkboxes
* Protect structured data and formulas

(**Note:** This is a function in spreadsheet as opposed to the process we talked about earlier)

We could use **Data Validation with Conditional Formatting** to create custom tools for our spreadsheets.

In SQL we could use CAST() or SAFE\_CAST() typecast values.

**Strings in SQL:**

| **Function** | **Usage** | **Example** |
| --- | --- | --- |
| CONCAT | A function that adds strings together to create new text strings that can be used as unique keys | CONCAT (‘Google’, ‘.com’); |
| CONCAT\_WS | A function that adds two or more strings together with a separator | CONCAT\_WS (‘ . ’, ‘www’, ‘google’, ‘com’)  \*The separator (being the period) gets input before and after Google when you run the SQL function |
| CONCAT with + | Adds two or more strings together using the + operator | ‘Google’ + ‘.com’ |

**Data Aggregation**

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

**Subqueries in SQL:**

There are a few rules that subqueries must follow:

* Subqueries must be enclosed within parentheses
* A subquery can have only one column specified in the SELECT clause. But if you want a subquery to compare multiple columns, those columns must be selected in the main query.
* Subqueries that return more than one row can only be used with multiple value operators, such as the IN operator which allows you to specify multiple values in a WHERE clause.
* A subquery can’t be nested in a SET command. The SET command is used with UPDATE to specify which columns (and values) are to be updated in a table.

**Types of Data Validation:**

1. **Data type:**

* **Purpose:** Check that the data matches the data type defined for a field.
* **Example:** Data values for school grades 1-12 must be a numeric data type.
* **Limitations:** The data value 13 would pass the data type validation but would be an unacceptable value. For this case, data range validation is also needed

1. **Data range:**

* **Purpose**: Check that the data falls within an acceptable range of values defined for the field.
* **Example**: Data values for school grades should be values between 1 and 12.
* **Limitations**: The data value 11.5 would be in the data range and would also pass as a numeric data type. But, it would be unacceptable because there aren't half grades. For this case, data constraint validation is also needed.

1. **Data constraints:**

* **Purpose**: Check that the data meets certain conditions or criteria for a field. This includes the type of data entered as well as other attributes of the field, such as number of characters.
* **Example**: Content constraint: Data values for school grades 1-12 must be whole numbers.
* **Limitations**: The data value 13 is a whole number and would pass the content constraint validation. But, it would be unacceptable since 13 isn’t a recognized school grade. For this case, data range validation is also needed.

1. **Data consistency:**

* **Purpose**: Check that the data makes sense in the context of other related data.
* **Example**: Data values for product shipping dates can’t be earlier than product production dates.
* **Limitations**: Data might be consistent but still incorrect or inaccurate. A shipping date could be later than a production date and still be wrong.

1. **Data structure:**

* **Purpose**: Check that the data follows or conforms to a set structure.
* **Example**: Web pages must follow a prescribed structure to be displayed properly.
* **Limitations**: A data structure might be correct with the data still incorrect or inaccurate. Content on a web page could be displayed properly and still contain the wrong information.

1. **Code validation:**

* **Purpose:** Check that the application code systematically performs any of the previously mentioned validations during user data input.
* **Example:** Common problems discovered during code validation include: more than one data type allowed, data range checking not done, or ending of text strings not well defined.
* **Limitations:** Code validation might not validate all possible variations with data input.

**Temporary Tables in SQL:**

* They are automatically deleted from the database when you end your SQL session.
* They can be used as a holding area for storing values if you are making a series of calculations. This is sometimes referred to as **pre-processing** of the data.
* They can collect the results of multiple, separate queries. This is sometimes referred to as data **staging**. Staging is useful if you need to perform a query on the collected data or merge the collected data.
* They can store a filtered subset of the database. You don’t need to select and filter the data each time you work with it. In addition, using fewer SQL commands helps to keep your data clean.

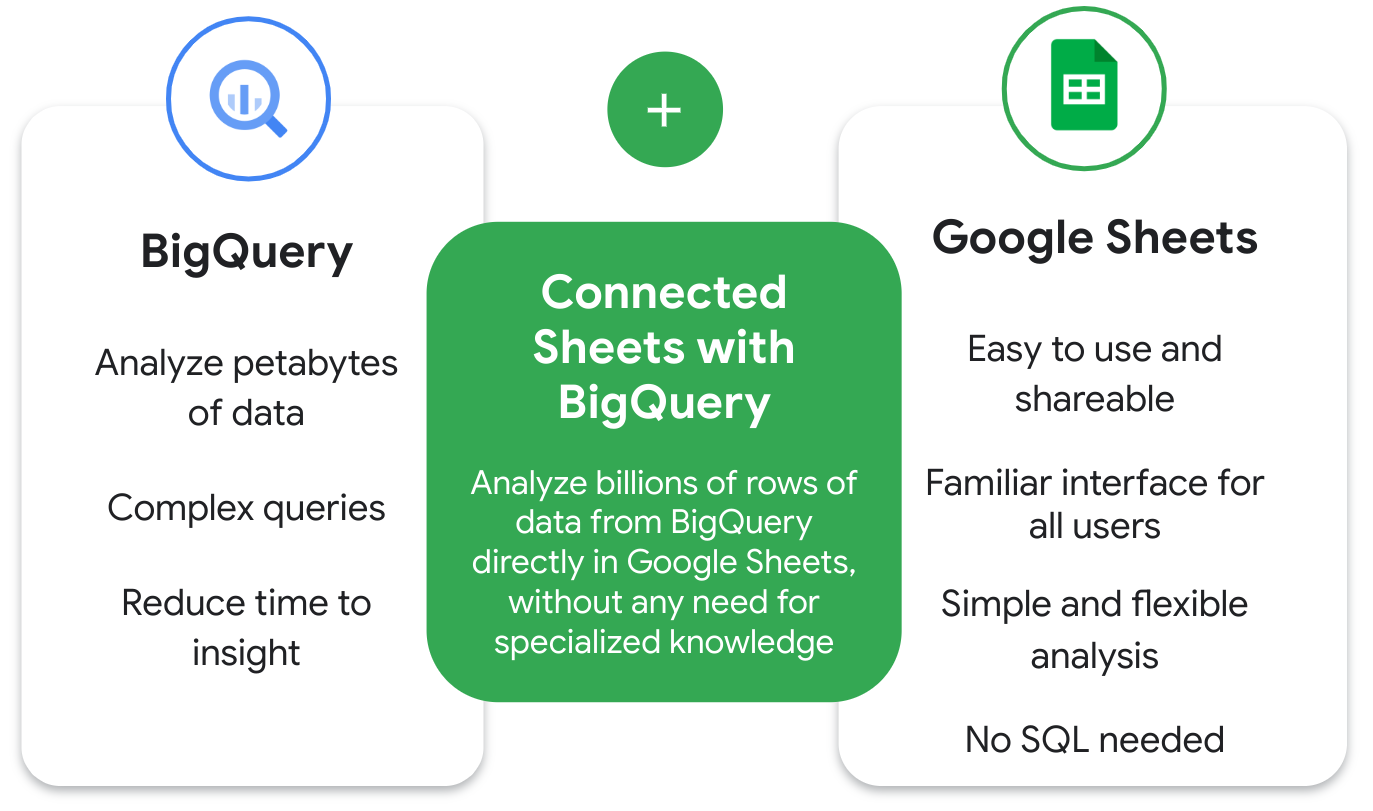
**How to create temporary tables:**

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

**Connected Sheets with BigQuery:**

A tool that allows data professionals to use basic spreadsheet functions to analyze large datasets housed in BigQuery.

**Connected Sheets** integrates both BigQuery and Google Sheets, allowing the user to analyze billions of rows of data in Sheets without any need for specialized knowledge, such as SQL.



A few example use cases of Connected Sheets include:

* **Business planning:** A user can build and prepare datasets, and then find insights from the data. For example, a data analyst can analyze sales data to determine which products sell better in different locations.
* **Customer service:** A user can find out which stores have the most complaints per 10,000 customers.
* **Sales:** A user can create internal finance and sales reports. After completing, they can share revenue reports with sales reps.
* **Logistics, fulfillment, and delivery:** A user can run real-time inventory management and intelligent analytics tools.

**Pros:**

## Collaborate with teammates and stakeholders

## Do more with familiar tools

## Easily visualize data

## Up to date data

## Less data integrity and security risk

**Cons:**

## Limited free pricing tier

## Data must be housed in BigQuery

## Query will fail with large results