4. Process Data from Dirty to Clean

Spreadsheets :

Field : - a single piece of information from a row or column of a spreadsheet.

Field length : - a tool for determining how many characters can be keyed into a field.

Data validation : - a tool for checking the accuracy and quality of data before adding or importing it.

Null and zero difference: -

Null - represents value does not exists
Zero - numeric zero value

Common data cleaning pitfalls -

- Not checking for spelling errors
- Forgetting to document errors
- Not checking for misfielded values.
- Overlooking missing values
- Looking at the subset of the data, not the whole picture.
- Losing track of business objectives.
- Not fixing the source of the error.
- Not analyzing the system prior to data cleaning.
- Not backing up your data before data cleaning
- Not accounting for data cleaning in your deadlines.

Transposing - Converting data from current long format (more rows than columns) to wide format (more columns than rows) this action is called transposing.

Spreadsheet formatting -

- **Conditional formatting** : -is a spreadsheet tool that changes how cells appear when values meet specific conditions.

E.g :- google sheets --- format-> conditional formatting

- **Remove duplicates** : tool that automatically searches and eliminates duplicates from the entries.
- **Split** : is a tool that divides the text around a specified character and put each fragment into a new separate cell.

E.g:- Data-> split text to columns ----google sheets

Spreadsheet functions -

put = (equal to) sign before functions

COUNTIF() :- if we want to count how many times given value occurs in given range

Syntax : COUNTIF(range, "value")
E.g:- =COUNTIF(I2:I72, "<100")

If there are multiple arguments then use countifs()
=COUNTIFS(B2:B21, "NY", C2:C21, "1")

- **Function with multiple constraint has 'S' at the end like sumifs(), countifs(),**

LEN() : - this function tells you the length of the text string by counting.
LEFT() - this function gives you number characters from the left side of the string.
Syntax : -LEFT(range, number of characters)
E.g:- =LEFT(a2,5)
RIGHT() :- gives number of characters from right side.
Syntax : - right(range, number of characters)
MID() : -is a function that gives you a segment from the middle of the text string.
Syntax : - =MID(range, reference starting point, number of middle characters.

Find () : - locate specific character in string.

Concatenate : - a function that joins multiple text strings into a single string.
Syntax = CONCATENATE(string 1, string 2)

e.g: concatenate(c2,d2) then drag all column length for all column values.

Trim : - a function that removes leading , trailing and repeated spaces in data.

Syntax : - =trim(range)

COUNTA - A function that counts the total number of values within a specified range.

Convert():- used to convert data from one type to another.

E.g:- =convert(b2,"f","c")

In this example b2 is the source cell which we want to convert from fahrenheit to celsius

Concat : -

can be used to combine strings from multiple tables in order to create a new string

Concat() vs concatenate() :

Google Sheet CONCAT strings work exactly the same as CONCATENATE except you can only use two text strings instead of three or more. Any of the below examples that only use two arguments could also use the CONCAT function. CONCAT in Google Sheets is less powerful, so it may be best to stick with CONCATENATE.

E.g:- =concat(a2,b2)

Output of this first name and last name will not have any space between.

E.g: - =concatenate(a2," ",b2)

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
E.g : - ‘Google’ + ‘.com’
=SUMIF(range, criteria, sum_range)

The first range is where the function will search for the condition that you have set. The criteria is the condition you are applying and the sum_range is the range of cells that will be included in the calculation.

E.g: -
=sumif(B3:B50,"=1",C3:C50)

- **Averageif**() : -Just like the previous two functions, the AVERAGEIF function will average the values in an array based on a given criteria. The syntax is =AVERAGEIF(range, criteria, [sum_range]).

The inputs to this function, range, criteria, and sum_range, work in exactly the same manner as in the SUMIF function. Again, the sum_range is optional.

=AVERAGEIF(B2:B21, "NY", D2:D21)

- Maxifs() : - The MAXIFS function is slightly different from the other three functions. The easiest way to observe the difference is to examine the syntax: =MAXIFS(max_range, range1, criteria1, [range2], [criteria2], ...).

The first argument, max_range, is the array over which you are finding the maximum. The second argument (range1) is the array you are checking. The third argument (criteria1) is the value that you are checking for. The inputs in the square brackets are for optional additional constraints.

Use this function to find the maximum sales from any salesperson in New York. Type the following: =MAXIFS(D2:D21, B2:B21, "NY").

example, to find the maximum sales in New York where the Max Item Cost is below $400, type the following into the function bar: =MAXIFS(D2:D21, B2:B21, "NY", E2:E21, "<400")

The first three inputs are the same as above, but now you've added the additional constraint that Max Item Value must be less than $400. The array **E2:E21** is the Max Item array and its cells are checked against the criteria <400. The function returns the following, which is the maximum sales of any New York salesperson who did not sell any single item over (or equal to) $400.

Pivot Tables -

- Pivot table is a summarization tool that is used in data processing.
- Pivot tables sort, reorganize, group , count, total and average data stored in the database.

To add pivot table in google sheets
Insert --> pivot table
Then add the rows which you want to see

Pivot tables make it possible to view data in multiple ways in order to identify insights and trends. They can help you quickly make sense of larger data sets by comparing metrics, performing calculations, and generating reports.

A pivot table has four basic parts: rows, columns, values, and filters

VLOOKUP : - stands for vertical lookup.

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

Syntax

=VLOOKUP(data to look up, 'where to look' !Range, column, false)
E.g:
=VLOOKUP(A2,'sheet 2'!A1:B31,2,false)

Schema : - a way of describing how something is organized.
Foreign key - a field within a table that is the primary key in another table.
Data mapping : -the process of matching fields from one data source to another.

- **Sort sheet : -** all data in spreadsheet is sorted by ranking of specified sorted column - data across the row is kept together.
- **Sort range :-** data across the row will not be kept together.
  Nothing else on the spreadsheet is rearranged besides the specified cells in a column.

SORT function
Sorts the rows of a given array or range by the values in one or more columns.
Sample Usage
SORT(A2:B26, 1, TRUE)
SORT({1, 2; 3, 4; 5, 6}, 2, FALSE)
SORT(A2:B26, C2:C26, TRUE)
Syntax
SORT(range, sort_column, is_ascending, [sort_column2, is_ascending2, ...])

- Filter function :

  FILTER(A2:B26, A2:A26 > 5, D2:D26 < 10)
  FILTER(A2:C5, {TRUE; TRUE; FALSE; TRUE})
  FILTER(A2:B10, NOT(ISBLANK(A2:A10)))
  Syntax
  FILTER(range, condition1, [condition2, ...])

Sumproduct - A function that multiplies arrays and returns the sum of those products. Syntax: =sumproduct(array1, array2….)

Array is a collection of values in a cell. Array is a range.
=sumproduct(B3:B7,C3:C7)

Course 5 - Analyze data to answer questions

- **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 are asking.
- 4 phases of analysis :
    1. Organize data
    2. Format and adjust data
    3. Get input from others
    4. Transform data
- **Outliers** are data points that are very different from similarly collected data and might not be reliable values.

Link to go to the big query sandbox account : -
https://console.cloud.google.com/bigquery

Other option is search in google :- big query sandbox console

Safe cast function : -
Using the **CAST** function in a query that fails returns an error in Big Query. To avoid errors in the event of a failed query, use the **SAFE_CAST** function instead. The **SAFE_CAST** function returns a value of Null instead of an error when a query fails.

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.

E.g: select safe_cast(mydate as string) from mytable

- **VLOOKUP ()- vertical lookup**
    A function that searches for a certain value in a column to return a corresponding piece of information.
    VLOOKUP() only returns the first match it finds.
    Vlookup can only return data from the right , it can't look left.

Always use $ to lock the rows and columns.
Two common reasons to use VLOOKUP are:

- Populating data in a spreadsheet
- Merging data from one spreadsheet with data in another

E.g:

=vlookup(A2,'Employee Rates'!$A$2:$B$5,2,false)

Here, there are 2 sheets in the spreadsheet

1. Employee hours :contains employee number and hour worked.
2. Employee rates : contains employee number and rate of pay.

In vlookup A2 is the value we are searching for in the other sheet. And 'employee rate' is the other sheet and the exclamation mark (!) comes after the sheet name. then the sheet range we want to search for begins with $ sign to lock the formula so it will not change when we are copying the formula. Then the value we want from the other sheet is a column number here, its column 2 that is the value we want to return, then the last thing we specified is false because we want an exact match if we give true then it will give only a close match. But we want a close match ,so only using false.

Helpful VLOOKUP reminders

- TRUE means an approximate match, FALSE means an exact match on the search key. If the data used for the search key is sorted, TRUE can be used.
- You want the column that matches the search key in a VLOOKUP formula to be on the left side of the data. VLOOKUP only looks at data to the right after a match is found. In other words, the index for VLOOKUP indicates columns to the right only. This may require you to move columns around before you use VLOOKUP.

- After you have populated data with the VLOOKUP formula, you may copy and paste the data as values only to remove the formulas so you can manipulate the data again.

Value() -

A function that converts a text string that represents a number to a numeric value.

E.g: - =value(a2)

INA() -

You can use the **IFNA** function to replace the #N/A error with something more descriptive, like “Does not exist.”

=IFNA(#NA,"Does not exists")

Joins -

- Join is a sql clause that is used to combine rows from two or more tables based on related columns.
- In the query table mention first is the left and table mention second is the right table.
- Join combines table using primary or foreign keys
- Two tables can be joined if the **primary key** for one table is included in the other table as a **foreign key**.

Common joins -

Inner : A function that returns records with matching values in both tables. We use join then sql by default uses Inner join.

E.g:

```
SELECT e.name as employee_name,
      e.role as employee_role,
      d.name as department_name
       FROM employee_data.employee as e
       inner join employee_data.department as d
       on
       e.department_id = d.department_id
```

here we can use 'as' aliases for tables or not depending on what software supports.

Full Outer : a function that combines left and right join to return all matching records in both tables. This means it will return all records in both tables.

It can be used as full join or full outer join.

Left : a function that will return all the records from the left table and the matching records from the right table.

Left outer join and left join both are correct syntax.

Right : a function that will return all the records from the right and only matching records from the left.

Right join is rarely used.

E.g:

1.

```
select *
from TableA
Left join
TableB
On keyA = keyB
```

2.

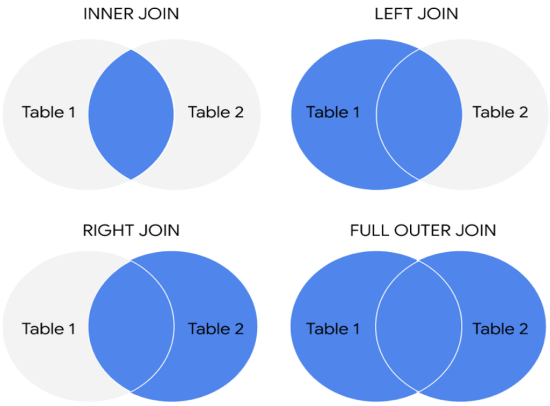
```
Select *
From TableB
Right join
TableA
On keyA = keyB
```

These both queries will return the same output.

Union Join - will stack tables on top of each other resulting in new rows.

Cross Join - would result in a table with all possible combinations of your tables' rows together. This can result in enormous tables and should be used with caution.

Cross Joins will likely only be used when your tables contain single values that you want to join together without a common dimension.



Case : - returns records with your condition by allowing you to include an if/then statement in your query.

E.g:

```
SELECT OrderID, Quantity,
CASE
  WHEN Quantity > 30 THEN 'The quantity is greater than 30'
  WHEN Quantity = 30 THEN 'The quantity is 30'
  ELSE 'The quantity is under 30'
END AS QuantityText
FROM OrderDetails;
```

If : Return "YES" if the condition is TRUE, or "NO" if the condition is FALSE:

```
SELECT IF(500<1000, "YES", "NO");
```

Subqueries :

- subqueries : query nestled within query. Usually, you will find subqueries nested in the SELECT, FROM, and/or WHERE clauses.
- Subquery can't be nestled within the set command of UPDATE.
- 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.

SQL Operators:

- Select columnA,
  ColumnB,
  ColumnA+columnB as ColumnC
  from Tabletemp

- Group by : this command groups rows that have the same values from the table into summary rows.
- Group by comes at the end of the query.

Extract : extract command lets us pull one part of a given date to use.
To use in select command type:
Extract(year from starttime) as year

```
select
extract(year from date) as year,
extract(month from date) as month,
productId,
round(max(UnitPrice),2) as unitprice,
sum(Quantity) as unitsold
from
sales.sales_info
group by year, month, productId
order by year, month, productId
```

Single operator in calculations

```
SELECT station_name,
ridership_2013,
ridership_2014,
ridership_2014 - ridership_2013 as change_2014_raw
FROM `bigquery-public-data.new_york_subway.subway_ridership_2013_present`
```

Modulo operator(%) in sql returns the remainder of division calculation.

Import Data into Spreadsheet -

- If you're using Google Sheets, you'll first need to import the data files (csv) into your spreadsheet . Open Sheets and navigate to the File menu, then select Import from the dropdown list.
- Select the first file and upload it to the spreadsheet. Choose Replace spreadsheet to insert it into the current sheet.

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.

Temporary tables, or temp tables, store subsets of data from standard data tables for a certain period of time. When you end your SQL database session, they are automatically deleted. Temp tables allow you to run calculations in temporary data tables without needing to make modifications to the primary tables in your database.

- 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.
- Temporary tables can be created using different clauses. In BigQuery, the **WITH** clause can be used to create a temporary table. The general syntax for this method is as follows:

```
With new_temp_table as
( select *
From existing table
Where trip_duration >=60
)
```

- The following method isn't supported in BigQuery, but most other versions of SQL databases support it, including SQL Server and mySQL. Using **SELECT** and **INTO**, you can create a temporary table based on conditions defined by a **WHERE** clause to locate the information you need for the temporary table. The general syntax for this method is as follows:

```
Select *
Into
Temp_table
From globalSales
Where region ='Africa'
```

- BigQuery uses **CREATE TEMP TABLE** instead of **CREATE TABLE**, but the general syntax is the same.

There are also other ways to create a temp table. Instead of using the **WITH** clause, you can use the **SELECT INTO** or the **CREATE TABLE** clauses.

The **SELECT INTO** clause copies data from one table into a new table, but doesn't add the new table to the database. It's useful if you want to make a copy of a table with a specific condition.

The **CREATE TABLE** clause is a good option when several people need to access the same temp table. This statement adds the table into the database.

Example of creating temporary table :

```
with
longest_used_bike as
    (select bikeid,
          sum(duration_minutes) as trip_duration
        from `bigquery-public-data.austin_bikeshare.bikeshare_trips`
        group by bikeid
        order by trip_duration desc
        limit 1
    )
```

```
## find station at which longest used bikes leaves most often
select trips.start_station_id,
       count(*) as trip_count
from longest_used_bike longest
full join
   `bigquery-public-data.austin_bikeshare.bikeshare_trips` trips
  on trips.bikeid = longest.bikeid
group by trips.start_station_id
order by trip_count desc
limit 1
```

Steps for Data Cleaning :

- Use the Trim() function to remove extra spaces.
- Use tools to remove duplicates.
- Check the data types
- Check sorting orders.