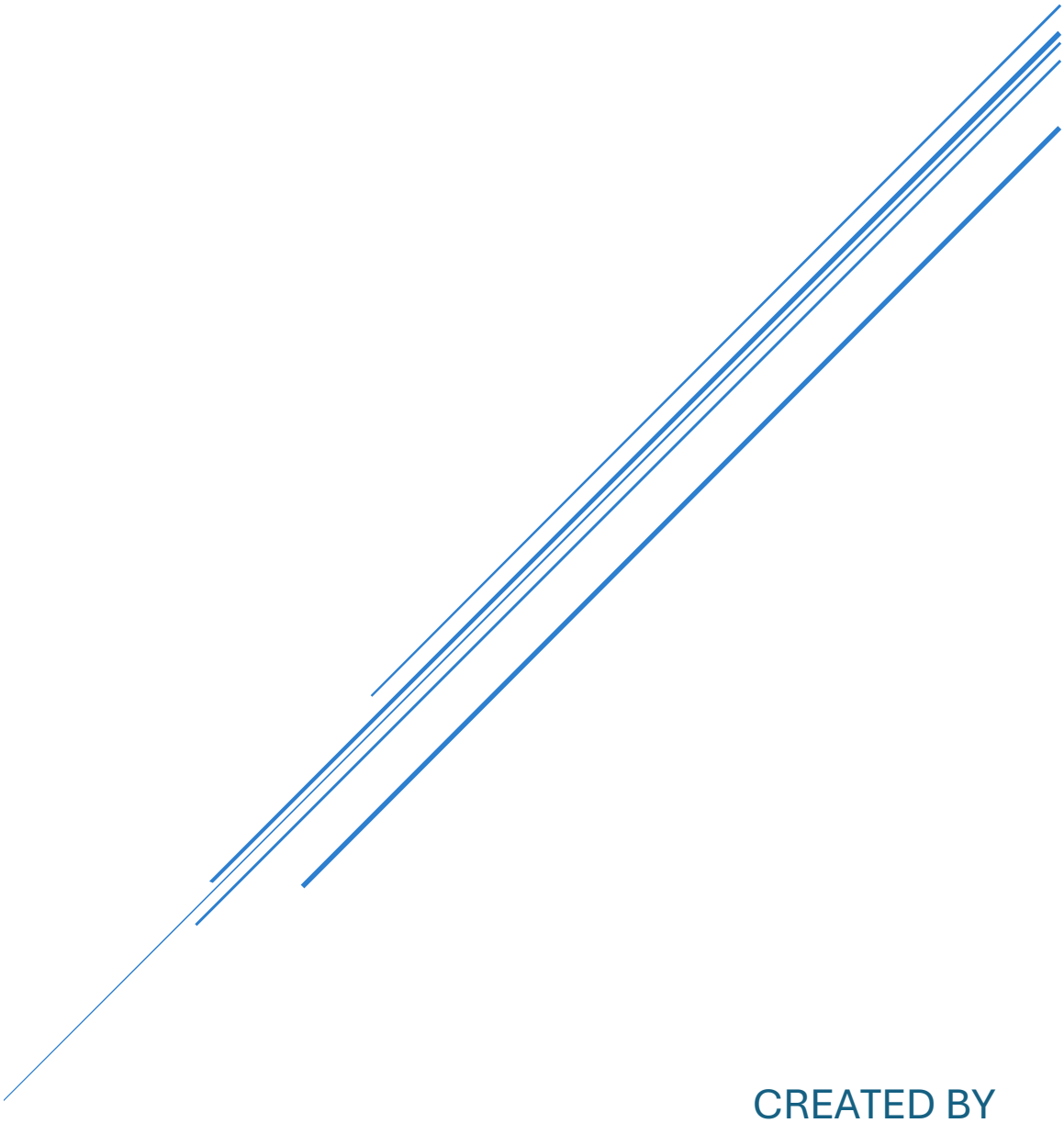
DATA ANALYSIS

SQL, EXCEL, TABLEAU, POWERBI, Python



CREATED BY
YERRA KEERTHANA

SQL QUERY CATEGORIZED FOR ACCOUNTING AND TAXATION TASKS:

Category	Query Description	SQL Query Example
Basic Data Retrieval	Retrieve all transactions	SELECT * FROM transactions;
	Retrieve transactions for a specific date	SELECT * FROM transactions WHERE transaction_date = '2024-12-01';
	Retrieve transactions for a specific customer	SELECT * FROM transactions WHERE customer_id = 101;
Summarization	Calculate total revenue	SELECT SUM(amount) AS total_revenue FROM transactions WHERE type = 'income';
	Calculate total expenses	SELECT SUM(amount) AS total_expenses FROM transactions WHERE type = 'expense';
	Calculate monthly revenue and expenses	SELECT MONTH(transaction_date) AS month, SUM(amount) AS total FROM transactions GROUP BY MONTH(transaction_date);
Tax Calculations	Calculate sales tax for a period	SELECT SUM(amount) * 0.07 AS sales_tax FROM transactions WHERE type = 'sale' AND transaction_date >= '2024-01-01';
	Identify transactions with missing tax amounts	SELECT * FROM transactions WHERE tax_amount IS NULL;
	Calculate VAT (Value Added Tax)	SELECT SUM(amount) * 0.2 AS vat FROM transactions WHERE type = 'sale';
Customer Reports	List of customers with outstanding balances	SELECT customer_id, SUM(balance) AS outstanding_balance FROM customers WHERE balance > 0 GROUP BY customer_id;

Category	Query Description	SQL Query Example
	Retrieve customers with high transaction volumes	SELECT customer_id, COUNT(*) AS transactions FROM transactions GROUP BY customer_id HAVING COUNT(*) > 100;
	Calculate total spending per customer	SELECT customer_id, SUM(amount) AS total_spent FROM transactions GROUP BY customer_id;
Vendor Reports	List all payments to vendors	SELECT * FROM payments WHERE payee_type = 'vendor';
	Calculate total payment to a specific vendor	SELECT vendor_id, SUM(amount) AS total_paid FROM payments WHERE vendor_id = 202;
Aging Reports	Retrieve aging report for receivables	SELECT customer_id, invoice_id, DATEDIFF(NOW(), due_date) AS days_due FROM receivables WHERE paid = 0;
	Retrieve aging report for payables	SELECT vendor_id, invoice_id, DATEDIFF(NOW(), due_date) AS days_due FROM payables WHERE paid = 0;
Account Reconciliation	Compare income vs expenses	SELECT (SUM(CASE WHEN type = 'income' THEN amount ELSE 0 END) - SUM(CASE WHEN type = 'expense' THEN amount ELSE 0 END)) AS net_income FROM transactions;
	Identify unreconciled transactions	SELECT * FROM transactions WHERE reconciled = 0;
	Match transactions to bank statements	SELECT * FROM transactions t LEFT JOIN bank_statements b ON t.transaction_id = b.transaction_id WHERE b.transaction_id IS NULL;

Category	Query Description	SQL Query Example
Audit and Compliance	Find duplicate invoices	SELECT invoice_number, COUNT(*) FROM invoices GROUP BY invoice_number HAVING COUNT(*) > 1;
	Retrieve transactions outside normal business hours	SELECT * FROM transactions WHERE HOUR(transaction_time) NOT BETWEEN 9 AND 18;
	Retrieve transactions exceeding a specific limit	SELECT * FROM transactions WHERE amount > 10000;
Advanced Analytics	Identify top 10 customers by revenue	SELECT customer_id, SUM(amount) AS total_revenue FROM transactions GROUP BY customer_id ORDER BY total_revenue DESC LIMIT 10;
	Calculate revenue growth year over year	SELECT YEAR(transaction_date) AS year, SUM(amount) AS revenue FROM transactions GROUP BY YEAR(transaction_date);

Query Type	Example Query	Explanation
SELECT	SELECT * FROM sales;	Retrieves all columns from the sales table.
WHERE Clause	SELECT * FROM sales WHERE region = 'North';	Filters sales data for the North region.
JOIN	SELECT a.id, a.name, b.amount FROM customers a JOIN sales b ON a.id = b.customer_id;	Joins customers and sales tables.
GROUP BY	SELECT region, SUM(amount) FROM sales GROUP BY region;	Groups sales data by region and calculates totals.

Query Type	Example Query	Explanation
HAVING	SELECT region, SUM(amount) FROM sales GROUP BY region HAVING SUM(amount) > 1000;	Filters grouped results.
Subqueries	SELECT * FROM sales WHERE amount > (SELECT AVG(amount) FROM sales);	Uses a subquery to filter sales.
Window Functions	SELECT id, amount, AVG(amount) OVER (PARTITION BY region) FROM sales;	Applies a window function to sales data.

	Query Description	SQL Query Example
Joins	Combine customer and transaction data	SELECT c.customer_id, c.name, t.amount FROM customers c INNER JOIN transactions t ON c.customer_id = t.customer_id;
	Retrieve unpaid invoices with vendor details	SELECT i.invoice_id, i.amount, v.name FROM invoices i LEFT JOIN vendors v ON i.vendor_id = v.vendor_id WHERE i.status = 'unpaid';
	List customers with no transactions	SELECT c.customer_id, c.name FROM customers c LEFT JOIN transactions t ON c.customer_id = t.customer_id WHERE t.transaction_id IS NULL;
	Match invoices with payments	SELECT i.invoice_id, i.amount, p.payment_id, p.amount_paid FROM invoices i FULL OUTER JOIN payments p ON i.invoice_id = p.invoice_id;
Views	Create a view for taxable income	CREATE VIEW taxable_income AS SELECT customer_id, SUM(amount) AS income FROM transactions WHERE type = 'income' GROUP BY customer_id;

	View overdue invoices	CREATE VIEW overdue_invoices AS SELECT invoice_id, due_date, DATEDIFF(NOW(), due_date) AS days_overdue FROM invoices WHERE paid = 0 AND due_date < NOW();
	Drop a view	DROP VIEW taxable_income;
Triggers	Automatically record tax calculations on insert	sql CREATE TRIGGER tax_calculation AFTER INSERT ON transactions FOR EACH ROW BEGIN UPDATE transactions SET tax_amount = NEW.amount * 0.1 WHERE id = NEW.id; END;
	Prevent duplicate invoice numbers	sql CREATE TRIGGER prevent_duplicates BEFORE INSERT ON invoices FOR EACH ROW BEGIN IF EXISTS (SELECT 1 FROM invoices WHERE invoice_number = NEW.invoice_number) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Duplicate invoice'; END IF; END;
	Log updates to payment status	sql CREATE TRIGGER log_payment_updates AFTER UPDATE ON payments FOR EACH ROW BEGIN INSERT INTO payment_logs (payment_id, status, updated_at) VALUES (NEW.payment_id, NEW.status, NOW()); END;
Functions	Calculate monthly revenue	sql CREATE FUNCTION monthly_revenue(month INT, year INT) RETURNS DECIMAL(10,2) BEGIN RETURN (SELECT SUM(amount) FROM transactions WHERE MONTH(transaction_date) = month AND YEAR(transaction_date) = year AND type = 'income'); END;

	Calculate total tax for a customer	sql CREATE FUNCTION customer_tax(customer_id INT) RETURNS DECIMAL(10,2) BEGIN RETURN (SELECT SUM(tax_amount) FROM transactions WHERE customer_id = customer_id); END;
	Drop a function	DROP FUNCTION customer_tax;
Procedures	Generate customer account statement	sql CREATE PROCEDURE customer_statement(IN customer_id INT) BEGIN SELECT transaction_id, amount, type, transaction_date FROM transactions WHERE customer_id = customer_id ORDER BY transaction_date; END;
	Update payment status for overdue invoices	sql CREATE PROCEDURE update_overdue_status() BEGIN UPDATE invoices SET status = 'overdue' WHERE paid = 0 AND due_date < NOW(); END;
	Call a procedure	CALL customer_statement(101);
ACID Compliance	Implement a transaction for payment and balance update	sql START TRANSACTION; UPDATE accounts SET balance = balance - 500 WHERE account_id = 1; INSERT INTO payments (account_id, amount) VALUES (1, 500); COMMIT;
	Rollback on error	sql START TRANSACTION; BEGIN TRY UPDATE accounts SET balance = balance - 500 WHERE account_id = 1; INSERT INTO payments (account_id, amount) VALUES (1, 500); COMMIT; END TRY BEGIN CATCH ROLLBACK; END CATCH;
Normalization	Split customer data into 3NF tables	1NF: CREATE TABLE customers (customer_id INT, name VARCHAR(255), phone VARCHAR(15));

		2NF: Separate addresses: CREATE TABLE customer_addresses (address_id INT, customer_id INT, address_line VARCHAR(255), city VARCHAR(100));
		3NF: Add ZIP mapping: CREATE TABLE zip_codes (zip_id INT, zip_code VARCHAR(10), city VARCHAR(100));
	De-normalize for performance	SELECT c.name, t.amount, t.transaction_date, a.city FROM customers c JOIN transactions t ON c.customer_id = t.customer_id JOIN customer_addresses a ON c.customer_id = a.customer_id;

EXCEL FORMULA

Category	Formula/Function	Syntax	Example	Description
Taxation & Accounting	**SUM**	=SUM(range)	=SUM(A1:A10)	Adds all the numbers in the specified range.
	SUMIF	=SUMIF(range, criteria, [sum_range])	=SUMIF(B1:B10, "Taxable", A1:A10)	Adds values based on a condition (e.g., tax-related).
	TAX CALCULATION	Custom formula: =income * tax_rate	=A1 * 0.20	Custom formula for calculating taxes (income * tax rate).

Category	Formula/Function	Syntax	Example	Description
	IF for Tax Brackets	=IF(income > 50000, income * 0.30, income * 0.20)	=IF(A1 > 50000, A1 * 0.30, A1 * 0.20)	Use IF to apply tax brackets (e.g., 30% if income > 50,000, 20% otherwise).
	PMT (Loan Calculation)	=PMT(rate, nper, pv, [fv], [type])	=PMT(5%/12, 60, 10000)	Calculates the monthly payment for a loan (PMT) based on rate, term, and present value (pv).
	FV (Future Value)	=FV(rate, nper, pmt, [pv], [type])	=FV(0.05/12, 120, -200, 10000)	Calculates the future value of an investment or loan (e.g., for tax savings or compound interest).
	CUMIPMT (Cumulative Interest)	=CUMIPMT(rate, nper, pv, start_period, end_period, type)	=CUMIPMT(5%/12, 60, 10000, 1, 12, 0)	Calculates cumulative interest over a specific range of periods.
	XNPV (Net Present Value with dates)	=XNPV(rate, values, dates)	=XNPV(0.05, A1:A10, B1:B10)	Calculates the net present value of cash flows with non-

Category	Formula/Function	Syntax	Example	Description
				periodic dates (useful for taxation and investment scenarios).
Accounting	**Net Income Calculation**	=revenue - expenses	=A1 - A2	Basic net income formula for accounting purposes.
	Balance Sheet Formula	=Assets = Liabilities + Equity	=A1 = B1 + C1	Ensures the balance sheet follows the accounting equation (Assets = Liabilities + Equity).
Statistics	**AVERAGE**	=AVERAGE(range)	=AVERAGE(A1:A10)	Calculates the mean of the data in the specified range.
	STDEV	=STDEV(range)	=STDEV(A1:A10)	Calculates the standard deviation of a data set.
	MEDIAN	=MEDIAN(range)	=MEDIAN(A1:A10)	Returns the median of the data in the range.

Category	Formula/Function	Syntax	Example	Description
	MODE	=MODE(range)	=MODE(A1:A10)	Returns the most frequently occurring value in a range.
	VAR	=VAR(range)	=VAR(A1:A10)	Calculates the variance of a data set.
	CORREL (Correlation)	=CORREL(range1, range2)	=CORREL(A1:A10, B1:B10)	Calculates the correlation coefficient between two data sets.
	PERCENTILE	=PERCENTILE(range, k)	=PERCENTILE(A1:A10, 0.5)	Returns the k-th percentile (e.g., median is the 50th percentile).
	RANK	=RANK(number, range, [order])	=RANK(A1, A1:A10)	Returns the rank of a number within a range.
	Z-SCORE	=STANDARDIZE(x, mean, standard_dev)	=STANDARDIZE(A1, AVERAGE(A1:A10), STDEV(A1:A10))	Calculates the Z-score for a given value.

Category	Formula/Function	Syntax	Example	Description
Pivot Tables & Dashboards	**Creating a Pivot Table**	Insert > PivotTable	N/A	Steps to create a PivotTable: Select data, choose PivotTable, select row/column values, and define aggregations.
	Grouping in Pivot Table	Right-click on field > "Group"	N/A	Group dates, numbers, or other fields into ranges in a PivotTable (e.g., by month, year).
	Calculated Fields in Pivot Table	Insert > Calculated Field	N/A	Create custom calculations in PivotTables by adding a calculated field (e.g., Profit = Revenue - Cost).
	Pivot Table Filter	Insert > Filter	N/A	Add filters to PivotTable to slice data by categories (e.g., by Tax

Category	Formula/Function	Syntax	Example	Description
				Year or Region).
	SUMIFS in Pivot Table	=SUMIFS(sum_range, criteria_range1, criteria1, ...)	=SUMIFS(A1:A10, B1:B10, ">10")	Use SUMIFS within a PivotTable to aggregate data based on multiple conditions.
	Pivot Table Drill-Down	Double-click on a value	N/A	Drill down into data to view detailed records behind a PivotTable summary.
	Create Dashboards	Combine multiple PivotTables, Charts, and Slicers	N/A	Design dashboards using PivotTables, charts, and interactive slicers for data exploration.
Pivot Table Tricks	**Pivot Table Slicer**	Insert > Slicer	N/A	Use slicers for interactive filtering in PivotTables and dashboards.

Category	Formula/Function	Syntax	Example	Description
	Pivot Table Timeline	Insert > Timeline	N/A	Use timeline for date-based filtering in PivotTables (e.g., by year, quarter).
	Pivot Table Show Values As	Right-click value > "Show Values As" > "% of Total"	N/A	Display values as percentages, running totals, or other variations.
	Pivot Table Sorting	Right-click value > "Sort"	N/A	Sort PivotTable data in ascending or descending order.
	Dynamic Pivot Table with Formula	Use formula to reference Pivot Table	=GETPIVOTDATA("Total Sales", A1)	Use GETPIVOTDATA to dynamically pull data from PivotTables for custom reporting or further analysis.
	Refresh Pivot Table	Right-click > "Refresh"	N/A	Refresh PivotTables to update them

Category	Formula/Function	Syntax	Example	Description
				with the latest data after changes to the source data.

Here's a table with **keyboard shortcuts** for **advanced usage** of **Computer**, **Excel**, and **Word** to boost productivity and streamline your tasks.

General Computer Shortcuts:

Action	Shortcut	Description
Open Task Manager	Ctrl + Shift + Esc	Opens Task Manager to view running programs and processes.
Switch between open apps	Alt + Tab	Switch between open applications.
Open File Explorer	Windows + E	Opens File Explorer to browse files and folders.
Minimize all windows	Windows + D	Minimizes all windows and shows the desktop.
Take a screenshot	Windows + Print Screen	Takes a screenshot and saves it automatically to the Pictures folder.
Lock computer	Windows + L	Locks your computer screen.
Open Settings	Windows + I	Opens Windows Settings.
Open Run Dialog	Windows + R	Opens the Run dialog box to run commands.
Open Start Menu	Windows	Opens the Start Menu.
Snap window to left/right	Windows + Left/Right Arrow	Snap the active window to the left or right side of the screen.

Action	Shortcut	Description
Open Action Center	Windows + A	Opens the Action Center to view notifications and quick settings.

Excel Keyboard Shortcuts:

Action	Shortcut	Description
Open a new workbook	Ctrl + N	Creates a new blank workbook.
Open an existing workbook	Ctrl + O	Opens an existing workbook.
Save workbook	Ctrl + S	Saves the current workbook.
Copy selection	Ctrl + C	Copies the selected cells to the clipboard.
Cut selection	Ctrl + X	Cuts the selected cells.
Paste selection	Ctrl + V	Pastes the copied or cut cells.
Undo action	Ctrl + Z	Undoes the previous action.
Redo action	Ctrl + Y	Redoes the previously undone action.
Select entire column	Ctrl + Space	Selects the entire column of the active cell.
Select entire row	Shift + Space	Selects the entire row of the active cell.
Add a new worksheet	Shift + F11	Inserts a new worksheet in the current workbook.
Insert a function	Shift + F3	Opens the Insert Function dialog box to insert a function.
Open the Go To dialog box	Ctrl + G or F5	Opens the Go To dialog to navigate to a specific cell or range.
Freeze panes	Alt + W, F, F	Freezes the selected row or column.

Action	Shortcut	Description
Open the Format Cells dialog	Ctrl + 1	Opens the Format Cells dialog to modify cell formatting.
Quick Sum (AutoSum)	Alt + =	Automatically inserts the SUM function for the selected range.
Show or hide the Ribbon	Ctrl + F1	Hides or shows the Ribbon.
Switch between worksheets	Ctrl + Page Up/Page Down	Switches between worksheets in the current workbook.
Move to next worksheet	Ctrl + Tab	Switches to the next worksheet in the workbook.
Open the Find dialog	Ctrl + F	Opens the Find dialog to search within the workbook.
Open the Replace dialog	Ctrl + H	Opens the Replace dialog to search and replace text.

Word Keyboard Shortcuts

Action	Shortcut	Description
Open a new document	Ctrl + N	Creates a new blank document.
Open an existing document	Ctrl + O	Opens an existing document.
Save document	Ctrl + S	Saves the current document.
Print document	Ctrl + P	Opens the Print dialog to print the document.
Bold selected text	Ctrl + B	Bolds the selected text.
Italicize selected text	Ctrl + I	Italicizes the selected text.
Underline selected text	Ctrl + U	Underlines the selected text.
Copy selected text	Ctrl + C	Copies the selected text to the clipboard.
Cut selected text	Ctrl + X	Cuts the selected text.
Paste copied/cut text	Ctrl + V	Pastes the copied or cut text.
Undo action	Ctrl + Z	Undoes the previous action.
Redo action	Ctrl + Y	Redoes the previously undone action.
Select all text	Ctrl + A	Selects all text in the document.
Open Find and Replace	Ctrl + H	Opens the Find and Replace dialog to search and replace text.
Bold/Italic/Underline toggle	Ctrl + Shift + B/I/U	Toggles bold, italic, or underline formatting on or off.
Align text to the left	Ctrl + L	Aligns the selected text to the left.
Center-align text	Ctrl + E	Centers the selected text.
Right-align text	Ctrl + R	Right-aligns the selected text.
Increase font size	Ctrl + Shift + >	Increases the font size of the selected text.

Action	Shortcut	Description
Decrease font size	Ctrl + Shift + <	Decreases the font size of the selected text.
Insert hyperlink	Ctrl + K	Opens the Insert Hyperlink dialog to insert a link.
Open the Styles pane	Alt + Ctrl + Shift + S	Opens the Styles pane for formatting styles in the document.
Insert a table	Alt + N, T	Inserts a table into the document.
Show/hide formatting marks	Ctrl + Shift + 8	Toggles visibility of non-printing characters (spaces, tabs, etc.).
Insert page break	Ctrl + Enter	Inserts a page break at the cursor position.
Move to next comment	Ctrl + Alt + N	Moves to the next comment in a document (useful in reviewing).

Advanced Tips:

- **Excel**: Use Ctrl + Shift + L to quickly apply filters to your data.
- **Word**: Use Ctrl + Alt + M to insert a comment when reviewing a document.
- **Windows**: Use Windows + X to open the Quick Link menu, which gives you quick access to various system tools like Device Manager, PowerShell, etc.

CHEAT SHEET FOR CREATING DASHBOARDS IN TABLEAU, INCLUDING STEPS, COMMON FORMULAS, AND FUNCTIONS IN A TABLE FORMAT:

Step/Task	Description/Action
1. Connect to Data	- Open Tableau and select the data source (Excel, SQL, etc.)
2. Data Preparation	- Clean, filter, or transform data as needed using the Data Source tab

Step/Task	Description/Action
3. Create a Worksheet	- Drag dimensions (e.g., Category, Date) and measures (e.g., Sales, Revenue) into Rows/Columns shelf
4. Build Views	- Choose chart type (Bar, Line, Pie, etc.) and drop fields onto Marks shelf (Color, Size, Shape, Label)
5. Create Calculations	- Create new calculated fields (Analysis > Create Calculated Field)
Function/Formula / **Description** **SUM()** - Adds up values in a field (e.g., SUM(Sales)) **AVG()** - Calculates the average of a field (e.g., AVG(Profit)) **COUNT()** - Counts the number of records (e.g., COUNT(Customer ID)) **COUNTD()** - Counts distinct records (e.g., COUNTD(Product)) **IF** - Conditional logic (e.g., IF [Sales] > 1000 THEN 'High' ELSE 'Low' END) **CASE** - Similar to IF but with multiple conditions (e.g., CASE [Region] WHEN 'East' THEN 'East Coast' ELSE 'Other' END)	- Combine multiple sheets onto a single dashboard (Dashboard > New Dashboard)

Step/Task		Description/Action
DATEPART()	- Extracts a part of a date (e.g., DATEPART('month', [Order Date]))	
DATENAME()	- Extracts the name of a date part (e.g., DATENAME('month', [Order Date]))	
DATEDIFF()	- Calculates the difference between two dates (e.g., DATEDIFF('day', [Order Date], TODAY()))	
ZN()	- Converts NULL values to 0 (e.g., ZN([Sales]))	
WINDOW_SUM()	- Calculates the sum across a window of data (e.g., WINDOW_SUM([Sales]))	
LOOKUP()	- Accesses data in previous or next rows (e.g., LOOKUP([Sales], -1) to get the previous row's value)	
RANK()	- Ranks data based on a field (e.g., RANK([Sales]))	
RUNNING_SUM()	- Calculates a running total (e.g., RUNNING_SUM(SUM([Sales])))	
ROUND()	- Rounds a number to the nearest integer or specified decimal (e.g., ROUND([Profit], 2))	
DATE()	- Converts a string or number to a date (e.g., DATE('2024-01-01'))	
6. Build Dashboard		
7. Add Interactivity		- Add filters, highlight actions, and URL actions to make dashboards interactive

Step/Task	Description/Action
8. Format Dashboard	- Adjust size, colors, labels, and fonts for a clean layout (Format > Borders, Fonts, etc.)
9. Publish	- Publish to Tableau Server, Tableau Online, or export as image/PDF

Common Tableau Formulas & Functions

Dashboard Tips & Tricks

Tip/Trick	Description
Use Containers	- Group elements in horizontal or vertical containers to keep the layout organized.
Use Dashboard Actions	- Add filter actions, highlight actions, or URL actions to create interactivity.
Optimize Performance	- Minimize the number of filters, use extracts for large datasets, and avoid complex calculations on the fly.
Responsive Design	- Adjust dashboard size and layout to fit different devices (Desktop, Tablet, Mobile).
Use Color Wisely	- Avoid overuse of colors; use color to highlight important data trends or patterns.

Power BI Steps for Creating Dashboards

Step/Task	Description/Action
1. Connect to Data	- Open Power BI Desktop and select the data source (Excel, SQL, etc.)

Step/Task	Description/Action
2. Data Transformation	- Use Power Query Editor to clean, transform, and load data (Transform Data button)
3. Create Reports	- Drag fields to the report canvas; select chart types (Bar, Line, Pie, etc.) from Visualizations pane
4. Add Filters	- Apply filters to visuals or create slicers for interactivity (Filters pane or Visualizations pane)
5. Create Measures	- Create new DAX measures for advanced calculations (Modeling tab > New Measure)
6. Build Dashboard	- Combine multiple report elements into a single dashboard (Pin visuals to Dashboard in Power BI Service)
7. Add Interactivity	- Use slicers, drillthrough, and tooltips to create interactive reports
8. Format Report	- Format the visuals using the Formatting pane (adjust colors, fonts, labels, etc.)
9. Publish	- Publish to Power BI Service to share dashboards with others or export reports (Publish button)

Common Power BI DAX Formulas & Functions

Function/Formula	Description
SUM()	- Adds up values in a column (e.g., SUM(Sales))
AVERAGE()	- Calculates the average of a column (e.g., AVERAGE(Profit))
COUNT()	- Counts the number of rows (e.g., COUNT(Customer ID))
DISTINCTCOUNT()	- Counts the number of distinct values in a column (e.g., DISTINCTCOUNT(Product))
IF()	- Conditional logic (e.g., IF(Sales > 1000, "High", "Low"))
SWITCH()	- Multiple conditions in a single formula (e.g., SWITCH([Region], "East", "East Coast", "Other"))

Function/Formula	Description
DATEADD()	- Shifts a date by a specified number of intervals (e.g., DATEADD([Date], 1, MONTH))
DATEDIFF()	- Calculates the difference between two dates (e.g., DATEDIFF([Order Date], TODAY(), DAY))
TODAY()	- Returns the current date (e.g., TODAY())
CALCULATE()	- Changes the context of a calculation (e.g., CALCULATE(SUM(Sales), [Region] = "East"))
FILTER()	- Filters data based on specified conditions (e.g., FILTER(Sales, Sales > 1000))
ALL()	- Removes filters from columns (e.g., ALL(Sales))
YEAR()	- Extracts the year from a date (e.g., YEAR([Order Date]))
MONTH()	- Extracts the month from a date (e.g., MONTH([Order Date]))
RANKX()	- Ranks values in a column (e.g., RANKX(ALL(Sales), SUM(Sales)))
RELATED()	- Retrieves a related value from another table (e.g., RELATED(Products[Category]))
LOOKUPVALUE()	- Looks up a value from a table based on conditions (e.g., LOOKUPVALUE(Products[Price], Products[Name], "Laptop"))

Power BI Tips & Tricks

Tip/Trick	Description
Use Measures for Flexibility	- Instead of using calculated columns, use measures for dynamic calculations that adjust based on filters.
Use Conditional Formatting	- Apply conditional formatting to highlight specific data points (e.g., color scales, data bars).
Drillthrough	- Enable drillthrough functionality to allow users to right-click on a visual and see more detailed data.

Tip/Trick	Description
Create Hierarchies	- Create hierarchies (e.g., Year > Quarter > Month > Day) for easier drill-downs.
Use Bookmarks	- Capture and save different states of a report (e.g., filters or visuals) to navigate easily.
Use Slicers	- Add slicers for easy filtering across multiple visuals simultaneously.
Optimize Performance	- Use aggregated tables, limit the amount of data, and avoid complex row-level filters on large datasets.
Pin Visuals to Dashboard	- Pin frequently used visuals to your Power BI Service dashboard for easy access.

COMPREHENSIVE GUIDE TO DATA ANALYSIS, COVERING THE STEPS, TOOLS, FUNCTIONS, AND THE IMPLEMENTATION PROCESS:

Data Analysis Steps

Step	Description
1. Define the Problem	- Clearly identify the problem or objective. Define the goals of the analysis to understand what you need.
2. Collect Data	- Gather relevant data from various sources (databases, APIs, spreadsheets, etc.). Ensure data quality.
3. Data Cleaning	- Clean the data by handling missing values, removing duplicates, and correcting errors.
4. Data Exploration	- Perform exploratory data analysis (EDA) to understand data distribution, relationships, and patterns.
5. Data Transformation	- Transform data (e.g., normalize, aggregate, or join datasets) to prepare for analysis or modeling.

Step	Description
6. Apply Statistical Analysis	- Conduct statistical tests, calculate summary statistics, and find correlations to understand relationships.
7. Build Models	- If needed, apply machine learning models or algorithms (e.g., regression, classification) for predictions.
8. Interpret Results	- Analyze and interpret the results to gain insights. Translate these findings into actionable decisions.
9. Communicate Findings	- Visualize results using charts, graphs, or dashboards and present findings to stakeholders.
10. Take Action	- Implement the recommendations based on the analysis to improve business processes or solve the problem.

Data Analysis Tools

Tool	Description
Excel	- Widely used for small to medium datasets, offering powerful features like pivot tables, functions, and charts.
SQL	- Essential for querying databases to extract and manipulate data using structured queries.
Python	- A versatile programming language for data analysis with libraries like Pandas, NumPy, Matplotlib, and Scikit-learn.
R	- A programming language focused on statistical analysis and visualizations (e.g., ggplot2, dplyr).
Power BI	- Business intelligence tool for creating interactive reports and dashboards using various data sources.
Tableau	- Data visualization tool for creating interactive and shareable dashboards with strong visual analytics.
Google Analytics	- Used for analyzing website traffic and user behavior, providing insights into web performance.

Tool	Description
AWS	- Cloud-based tools for large-scale data storage, processing (e.g., AWS S3, AWS Redshift), and analysis.
Google BigQuery	- Data warehouse for performing large-scale data analysis using SQL on Google Cloud.
Hadoop/Spark	- Frameworks for processing large datasets (Big Data) across distributed systems.

Common Data Analysis Functions

Function	Description
SUM()	- Calculates the total of a column or range (e.g., SUM(Sales)).
AVERAGE()	- Calculates the mean of a set of values (e.g., AVERAGE(Profit)).
COUNT()	- Counts the number of entries in a column (e.g., COUNT(Orders)).
MEDIAN()	- Returns the middle value in a dataset (e.g., MEDIAN(Income)).
MODE()	- Returns the most frequent value in a dataset (e.g., MODE(Age)).
CORREL()	- Calculates the correlation between two variables (e.g., CORREL(Sales, Advertising Spend)).
VLOOKUP()	- Searches for a value in the first column of a table and returns a corresponding value from another column.
IF()	- Conditional logic for evaluating if a condition is true or false (e.g., IF([Sales]>1000, "High", "Low")).
GROUPBY()	- Groups data based on one or more columns (e.g., GROUPBY(Product, SUM(Sales))).
PIVOT()	- Transforms unique values from one column into multiple columns (e.g., pivoting regions into individual columns).
REGEX()	- Applies regular expressions for pattern matching (e.g., extracting emails or phone numbers from text).

Function	Description
LEFT() / RIGHT()	- Extracts characters from the left or right side of a string (e.g., LEFT(Name, 3) for the first 3 letters).
DATE()	- Converts a string or number to a date (e.g., DATE(2024, 12, 31)).

Data Analysis Implementation Process

1. **Data Collection & Access**
    - Ensure data is accurate, timely, and relevant. Use appropriate tools (e.g., APIs, databases, spreadsheets).
2. **Data Cleaning & Transformation**
    - Handle missing values (e.g., impute, delete, or flag them). Remove duplicates, correct data types, and normalize data.
3. **Exploratory Data Analysis (EDA)**
    - Perform initial analysis to visualize distributions, trends, and identify outliers using graphs (histograms, box plots).
    - Use descriptive statistics like mean, median, standard deviation.
4. **Modeling and Analysis**
    - For predictive analysis, apply statistical models or machine learning algorithms (e.g., regression, clustering, classification).
    - Choose the right model based on the objective and validate using techniques like cross-validation.
5. **Data Visualization**
    - Create meaningful charts (e.g., bar, line, pie, scatter plots) to represent findings.
    - Use Power BI, Tableau, or Excel for creating dashboards and reports that summarize key insights.
6. **Interpretation of Results**
    - Assess the results from statistical tests or models. Look for significant patterns, relationships, or anomalies in the data.

- Draw insights and translate these findings into actionable business recommendations.

7. **Reporting & Presentation**
    - Communicate the findings clearly through visualizations and reports.
    - Tailor reports to the audience (e.g., business executives, technical teams).
8. **Take Action & Iterate**
    - Implement recommendations based on the analysis. Measure the impact of changes.
    - Iterate the analysis process as needed based on feedback or new data.

Python cheat sheet with commonly used code snippets and functions for various tasks:

9. **1. Basic Python Syntax**

Task	Code
Print to Console	print("Hello, World!")
Variable Assignment	x = 10
Commenting	# This is a comment
Multi-line Comment	''' This is a multi-line comment '''
Input from User	name = input("Enter your name: ")
Check Data Type	type(x)
Type Casting	int("10"), float("10.5"), str(100)

10. **2. Data Structures**

Task	Code
List (Array)	my_list = [1, 2, 3, 4, 5]
Access List Item	my_list[0]
List Slicing	my_list[1:4]
Add Item to List	my_list.append(6)
Remove Item from List	my_list.remove(3)
Tuple	my_tuple = (1, 2, 3, 4)
Set	my_set = {1, 2, 3, 4}
Dictionary (HashMap)	my_dict = {"key1": "value1", "key2": "value2"}
Access Dictionary Value	my_dict["key1"]

Task	Code
Add Key-Value Pair	my_dict["key3"] = "value3"

11. 3. Control Flow

Task	Code
If Statement	if x > 10: print("x is greater than 10")
If-Else Statement	if x > 10: print("x is greater than 10") else: print("x is less than or equal to 10")
Elif Statement	if x > 10: print("x is greater") elif x == 10: print("x is 10") else: print("x is smaller")
For Loop	for i in range(5): print(i)
While Loop	while x < 10: x += 1
Break	for i in range(5): if i == 3: break
Continue	for i in range(5): if i == 3: continue

12. 4. Functions

Task	Code
Define Function	def my_function(): print("Hello from function!")
Function with Parameters	def greet(name): print(f"Hello, {name}!")
Return Value from Function	def add(a, b): return a + b
Lambda Function	add = lambda a, b: a + b

13. 5. String Manipulation

Task	Code
Concatenate Strings	full_name = "John" + " " + "Doe"
String Length	len("Hello")
Convert to Upper Case	"hello".upper()
Convert to Lower Case	"HELLO".lower()
Substring	"Hello, World!"[7:12]
Find Substring	"Hello, World!".find("World")
Replace Substring	"Hello, World!".replace("World", "Python")
Split String	"Hello, World!".split(",")

14. 6. File Handling

Task	Code
Open a File	file = open("example.txt", "r")
Read File	content = file.read()
Read Line by Line	lines = file.readlines()
Write to a File	file = open("example.txt", "w"); file.write("Hello, World!")
Close a File	file.close()

15. 7. List Comprehension

Task	Code
Basic List Comprehension	[x**2 for x in range(5)]
List Comprehension with Condition	[x for x in range(10) if x % 2 == 0]

16. 8. Error Handling

Task	Code
Try-Except Block	try: x = 10 / 0 except ZeroDivisionError: print("Cannot divide by zero")
Finally Block	try: x = 10 / 0 except ZeroDivisionError: print("Error!") finally: print("This runs always")

17. 9. Working with Libraries

Task	Code
Importing a Library	import math
Using a Library Function	math.sqrt(16)
Install a Library (using pip)	pip install pandas
Import Specific Function	from math import sqrt

18. 10. NumPy for Numerical Operations

Task	Code
Import NumPy	import numpy as np
Create NumPy Array	arr = np.array([1, 2, 3, 4, 5])
Array Reshaping	arr.reshape(5, 1)
Array Operations	arr + 10, arr * 2
Array Slicing	arr[1:4]
Array Statistics	np.mean(arr), np.median(arr), np.std(arr)

19. 11. Pandas for Data Handling

Task	Code
Import Pandas	import pandas as pd
Create DataFrame	df = pd.DataFrame({"Name": ["Alice", "Bob"], "Age": [25, 30]})
Read CSV File	df = pd.read_csv("data.csv")
View Data	df.head()
Basic Statistics	df.describe()
Filter Data	df[df["Age"] > 25]
Group By	df.groupby("Age").mean()

20. 12. Matplotlib for Plotting

Task	Code
Import Matplotlib	import matplotlib.pyplot as plt
Simple Plot	plt.plot([1, 2, 3], [4, 5, 6]); plt.show()
Bar Plot	plt.bar([1, 2, 3], [4, 5, 6]); plt.show()
Histogram	plt.hist([1, 2, 2, 3, 4, 5]); plt.show()
Scatter Plot	plt.scatter([1, 2, 3], [4, 5, 6]); plt.show()