**Explanation: Authoring Reports in Power BI**

**Authoring reports** refers to the process of designing, creating, and customizing reports that present data insights through visualizations. In Power BI, report authoring is one of the most important tasks, allowing users to turn raw data into meaningful, interactive visuals for decision-making.

**What is Report Authoring?**

**Report authoring** is the activity of creating structured, interactive reports by selecting appropriate data sources, applying necessary transformations, creating data models, and finally designing data visualizations. These reports help users to analyze trends, patterns, and business performance visually.

In simple terms — **it is the process of turning data into a story using charts, graphs, and visuals**.

**Key Steps in Authoring Reports in Power BI**

**1️. Connect to Data Sources**

* Link Power BI to Excel, CSV, SQL, cloud services, or web APIs.
* Use **Import Mode** or **DirectQuery** based on need.

**2️. Data Transformation & Cleaning**

* Use **Power Query Editor** to:
  + Remove unnecessary columns.
  + Handle missing data.
  + Change data types.
  + Create calculated columns/measures.
  + Merge multiple sources.

**3️. Build a Data Model**

* Define relationships between tables.
* Use **star** or **snowflake schema**.
* Add DAX measures, hierarchies, and calculated tables.

**4️. Create Visualizations**

* Use visuals like:
  + Bar, line, pie charts
  + Tables, matrices, cards, KPIs
  + Maps for location data
* Drag and drop fields onto the canvas.

**5️. Add Interactivity**

* Use:
  + **Slicers** for filtering
  + **Drill-down** for detail views
  + **Cross-filtering** between visuals
  + **Bookmarks & buttons** for navigation

**6️. Publish & Share Reports**

* Publish to **Power BI Service** (cloud) or **Report Server** (on-premises).
* Share via web, Teams, or SharePoint.
* Set up **scheduled refreshes** for live data updates.

**Importance of Report Authoring:**

* Helps in converting raw data into actionable insights.
* Allows users to monitor business performance through KPIs.
* Supports better, faster decision-making through visual analysis.
* Makes data analysis accessible to non-technical users via interactive visuals.

**Theory: List, Crosstab, and Chart Reports**

In business intelligence reporting, data is often presented in various formats to simplify analysis and support decision-making. The most commonly used report formats are **List Reports**, **Crosstab Reports**, and **Chart Reports**. Each of these serves a specific purpose and suits different types of data visualization and analysis needs.

**1️. List Reports**

**Definition:**

A **List Report** is a type of report that displays detailed data records in a simple, tabular format, listing each data item or row sequentially. It is similar to a spreadsheet or a database table view.

**Characteristics:**

* Displays data in a **row-by-row format**.
* Shows **detailed information for each record**.
* Can include multiple columns, with headers and data values.
* Can support sorting, filtering, and grouping.
* Often used for transactional or operational data reporting.

**Example Use Cases:**

* List of all customer orders.
* Employee directory with names, IDs, and contact details.
* Inventory list showing product codes, names, and stock levels.

**Advantages:**

* Simple to design and easy to understand.
* Useful for displaying raw, unaggregated data.
* Can be exported directly to Excel or CSV for further analysis.

**2️. Crosstab Reports**

**Definition:**

A **Crosstab Report** (also called a **Matrix Report** or **Pivot Table**) summarizes and displays data in a **two-dimensional grid format**, where one set of data fields forms the rows, another set forms the columns, and aggregated values are displayed at the intersections.

**Characteristics:**

* Used for **summarizing large volumes of data**.
* Allows displaying data dimensions on both **rows and columns**.
* Uses aggregation functions like SUM, COUNT, AVG, etc.
* Supports drill-down capabilities to view detailed records.

**Example Use Cases:**

* Sales summary by Region (rows) and Product Category (columns).
* Student marks distribution by Subject (columns) and Class (rows).
* Expense report categorized by Department and Month.

**Advantages:**

* Makes it easy to **compare values across multiple dimensions**.
* Provides a **concise summary of complex data**.
* Highly interactive when combined with filters and drill-down features.

**3️. Chart Reports**

**Definition:**

A **Chart Report** represents data visually using graphical elements such as bars, lines, pies, and areas. These reports are designed to make it easier to identify patterns, trends, and outliers within data.

**Characteristics:**

* Uses **visual elements like charts and graphs** to present data.
* Can display aggregated values, percentages, or trends over time.
* Supports interactivity like tooltips, legends, and drill-down in tools like Power BI.

**Common Chart Types:**

* Bar and Column Charts
* Pie and Donut Charts
* Line and Area Charts
* Tree Maps and Scatter Plots
* Maps (for geographical data)

**Example Use Cases:**

* Sales trends over months using a line chart.
* Market share distribution through a pie chart.
* Product performance comparison via a bar chart.

**Advantages:**

* **Visually appealing and easy to interpret**.
* Helps quickly identify trends, comparisons, and relationships.
* Makes data-driven storytelling more effective.

**Theory: Grouping and Summarizing Data**

**What is Grouping and Summarizing Data?**

**Grouping** and **Summarizing** are two essential operations in data analysis used to **organize data into meaningful categories and calculate aggregate values** such as totals, averages, counts, or percentages.

They help convert detailed raw data into summarized insights for better understanding, decision-making, and reporting.

**1️. Grouping Data**

**Definition:**

**Grouping** is the process of organizing data into distinct categories based on the values of one or more fields (columns). It groups together rows that have the same value in the specified column(s).

**Characteristics:**

* Groups data based on categories like region, product type, department, etc.
* Reduces large datasets into a smaller number of logical groups.
* Commonly used with summarizing functions to show grouped totals.

**Example:**

|  |  |
| --- | --- |
| **Region** | **Sales Amount** |
| East | 50,000 |
| West | 40,000 |
| North | 60,000 |

Here, data is **grouped by Region**.

**Purpose:**

* Organize large data into understandable chunks.
* Facilitate comparison between different groups.
* Identify patterns, trends, and outliers in data.

**2️. Summarizing Data**

**Definition:**

**Summarizing** is the process of applying **aggregate functions** to grouped data to compute meaningful totals or statistics like:

* **SUM** (total sales)
* **AVERAGE** (average salary)
* **COUNT** (number of orders)
* **MAX / MIN** (highest / lowest sales)
* **PERCENTAGE** (contribution of a group)

**Characteristics:**

* Converts detailed data into key metrics.
* Can be performed on grouped or ungrouped data.
* Often displayed in reports, dashboards, and summary tables.

**Example:**

If you group sales data by region, summarizing might involve:

* **Total sales per region**
* **Number of transactions per region**
* **Average sales per order in each region**

**Importance of Grouping and Summarizing:**

* Simplifies large, complex datasets.
* Supports comparative and trend analysis.
* Helps decision-makers quickly understand business performance.
* Forms the basis for **dashboards, scorecards, and KPI reports**.

**3️. How It’s Done in Tools like Power BI:**

* **Group By** option in Power Query.
* **Group and Summarize** in Data Visualizations (e.g., Matrix visual or Table visual with aggregation).
* **DAX functions** like SUM(), AVERAGE(), COUNTROWS(), MAX(), MIN() used for summarizing.
* Interactive tools like slicers or filters to dynamically group and summarize data.

**Theory: Filters with Single Level and Filters on Multiple Pages**

**What is a Filter in Reporting?**

A **filter** in reporting is a feature that allows you to **restrict, refine, or control the data displayed in a report or visualization based on specific criteria**. Filters help focus on particular data points, ranges, or categories by including or excluding records as needed.

**1️. Filter with Single Level (Visual Level Filters)**

**Definition:**

A **Single Level Filter** (also known as **Visual Level Filter**) is applied to **only one specific visualization or chart** on a report page. It affects the data shown in that individual visual without impacting other visuals on the same page.

**Characteristics:**

* Applied directly to individual visuals like charts, tables, or cards.
* Restricts data for that visual only.
* Multiple single-level filters can exist independently on a report page.
* Can use conditions like equals, contains, greater than, etc.

**Example:**

In a sales report, if you apply a filter to a bar chart to show data for only **‘2024’**, that filter will affect **only that bar chart**. Other visuals showing 2023 or other years remain unaffected.

**Use Cases:**

* Focusing a visual on a specific region, product category, or time period.
* Comparing different filtered visuals side-by-side.
* Isolating key metrics without disturbing the rest of the report.

**2️. Filter on Multiple Pages (Report Level Filters)**

**Definition:**

A **Multi-Page Filter** (also called **Report Level Filter**) is a filter that is applied to **the entire report and affects all pages and visuals within that report file**. It ensures consistency by applying the same data criteria across multiple report pages.

**Characteristics:**

* Applied once and impacts all pages and visuals in the report.
* Ensures uniform data context throughout the entire report.
* Saves time by eliminating the need to apply the same filter multiple times.
* Can be modified at any time, instantly updating the entire report.

**Example:**

If you apply a report-level filter to show only data for **‘India’** in a global sales report, all pages (like Sales by Year, Customer Segments, Profit Analysis) will display data related to India only.

**Use Cases:**

* Focusing an entire report on a specific region, department, or timeframe.
* Creating country-wise or branch-wise reports with consistent filters.
* Generating role-based reports (e.g., showing only data relevant to a sales manager’s region).

**Difference Table:**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Single Level Filter** | **Multi-Page Filter** |
| **Scope** | One visual only | Entire report (all pages & visuals) |
| **Application** | Applied directly to individual visuals | Applied at the report-level settings |
| **Usage** | Compare or isolate specific visuals | Maintain consistency across pages |
| **Example** | Filter one table to show 2023 data | Filter entire report for region = ‘South’ |

**Theory: Sorting in Power BI**

*(Sort by Column, Sort on Visual Graph, Sort Multiple Columns)*

**What is Sorting in Power BI?**

**Sorting** in Power BI refers to the process of **arranging data in a specific order** — ascending, descending, or based on custom logic — so that patterns, trends, and important insights become easier to identify in visuals, tables, and reports.

**1️. Use Sort by Column**

**Definition:**

The **Sort by Column** feature in Power BI allows you to **define a custom sort order for a column** based on the values of another column. This is especially useful when the default alphabetical or numerical order isn’t meaningful for the business context.

**Example:**

If you have a column for **Month Names** (January, February, March, …), it would sort alphabetically by default.  
To sort it by actual month order, you can use a **Month Number column (1–12)** and apply **Sort by Column** on the Month Name.

**Steps:**

* Select the column you want to sort.
* Go to **Column Tools > Sort by Column**.
* Choose the column to define the sort order.

**Use Cases:**

* Sorting month names, weekdays, financial quarters.
* Custom ranking or priority order.
* Handling hierarchical or non-standard sort logic.

**2️. Sort on Visual Graph**

**Definition:**

**Sort on Visual Graph** refers to the ability to **sort the data shown in a specific visual (like a bar chart, table, matrix, or card) directly from within the visual interface** — either ascending or descending.

**Features:**

* Sorting can be based on **category (dimension)** or **value (measure)**.
* Done by clicking the **ellipses (⋯)** or the **sort icon** on the visual.
* Changes the order of data presentation without modifying underlying data.

**Example:**

In a **Sales by Country** bar chart:

* Sort countries alphabetically.
* Or sort by descending sales figures.

**Use Cases:**

* Quickly identify top or bottom performers.
* Make visuals more readable and insightful.
* Control how data trends are presented in dashboards.

**3️. Sort Multiple Columns**

**Definition:**

**Sorting Multiple Columns** is applied in Power BI’s **Table or Matrix visuals**, where you can arrange data by **multiple columns sequentially** — defining which field to sort first, then next, and so on (similar to Excel multi-level sorting).

**Features:**

* The first column defines the primary order.
* The second column sorts within the primary column groups.
* Can be set using the **Sort Ascending/Descending** options on each column header.

**Example:**

In a **Sales Data Table**:

1. First, sort by **Region** (A-Z).
2. Then, sort within each region by **Sales Amount** (High to Low).

**Use Cases:**

* Organizing data hierarchically (Region → Product → Sales).
* Providing structured, drill-down style views in reports.
* Creating clean, logically ordered tables for reporting.

**Summary Table:**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Use Case** |
| **Sort by Column** | Sort one column based on the values of another column | Sorting month names by month no. |
| **Sort on Visual Graph** | Sort a visual directly by category or value | Sorting bar chart by sales |
| **Sort Multiple Columns** | Sort a table/matrix by multiple fields sequentially | Sort region first, then sales |

**Theory: Using CALCULATE Function with ALL and ALLEXCEPT**

**What is CALCULATE in DAX?**

The **CALCULATE function** in DAX is used to **modify the context in which a calculation is performed**. It evaluates an expression in a context that is modified by the specified filters.

**Syntax:**

CALCULATE(<expression>, <filter1>, <filter2>, ...)

**1️. Use of ALL Function**

**Definition:**

The **ALL function** removes all filters from a specified column or table. When used inside a CALCULATE, it **ignores existing filters and evaluates the expression in an unfiltered context**.

**Syntax:**

ALL(<column/table>)

**Example:**

If you want to calculate **Total Sales ignoring any filters applied on the 'Region' column**:

Total Sales All Region = CALCULATE(SUM(Sales[Amount]), ALL(Sales[Region]))

**Use Cases:**

* To calculate **grand totals** or **overall averages** unaffected by slicers or visuals.
* To create **percentage of total** calculations.
* To remove filters temporarily for comparative analysis.

**2️. Use of ALLEXCEPT Function**

**Definition:**

The **ALLEXCEPT function** removes all context filters in the table **except those specified for certain columns**. It’s useful when you want to **preserve filters on some columns but ignore others**.

**Syntax:**

ALLEXCEPT(<table>, <column1>, <column2>, ...)

**Example:**

If you want to calculate **Total Sales for each product while ignoring all other filters except for Product Name**:

Total Sales by Product = CALCULATE(SUM(Sales[Amount]), ALLEXCEPT(Sales, Sales[Product]))

**Use Cases:**

* Calculating **subtotals** within categories.
* Keeping certain filters while clearing others.
* Useful in dashboards for drilldown or cross-filtering control.

**Difference Table:**

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Use Case** |
| ALL | Removes all filters from a table/column | Calculate grand totals, % of total |
| ALLEXCEPT | Removes all filters except those on specified columns | Subtotals by category, context control |

**Theory: Matrix in Power BI**

**What is a Matrix in Power BI?**

A **Matrix** is a powerful visual element in Power BI that displays data in a **tabular format with dynamic rows and columns**, allowing users to group, summarize, and explore hierarchical or pivoted data — similar to a pivot table in Excel.

It enables **drill-down** functionality and can show data relationships across multiple dimensions.

**Key Features of Matrix**

|  |  |
| --- | --- |
| **Feature** | **Description** |
| **Rows and Columns** | Organize data into dynamic, hierarchical groups using fields for rows and columns. |
| **Values (Measures)** | Display numerical summaries like totals, averages, counts, etc. |
| **Subtotals and Grand Totals** | Automatically calculate subtotals for groups and overall totals for the entire data set. |
| **Drill Down / Drill Through** | Allow users to navigate deeper into hierarchical data levels interactively. |
| **Conditional Formatting** | Apply colors, icons, or data bars based on cell values to enhance readability. |
| **Stepped Layout** | Option to show hierarchical indentation within rows. |
| **Word Wrap** | Enable/disable text wrapping within cells for better visibility. |

**Matrix vs Table**

|  |  |
| --- | --- |
| **Matrix** | **Table** |
| Can display data hierarchically. | Flat data structure (one level). |
| Supports rows and columns groups. | Only columns — no groupings. |
| Allows drill-down functionality. | No drill-down option. |
| Can pivot data like a pivot table. | Static tabular format. |

**Use Cases of Matrix in Power BI**

* **Sales Performance by Region and Year.**
* **Product sales analysis across categories and sub-categories.**
* **Financial reporting (Income statement, Balance sheet format).**
* **Employee count by department and job role.**
* **Year-over-year comparisons within business metrics.**

**Example**

**Matrix Structure:**

|  |  |  |
| --- | --- | --- |
| **Region** | **2023 Sales** | **2024 Sales** |
| North | ₹5,00,000 | ₹6,50,000 |
| South | ₹3,00,000 | ₹4,20,000 |
| **Total** | ₹8,00,000 | ₹10,70,000 |

* **Rows:** Region
* **Columns:** Year
* **Values:** Sales Amount
* **Total:** Automatically calculated

**Theory: Creating Two Tables and Joining with a Crosstab in Power BI**

**Objective**

To combine and compare information from two separate tables using a **common field (Emp ID)** and display the combined result in a **Crosstab (Matrix)** visual.

**Steps to Create and Join Two Tables**

**1️. Create Two Tables**

|  |  |
| --- | --- |
| **Table 1: Employee Salary Data** |  |
| **Emp ID** | **Name** |
| 101 | Rahul Sharma |
| 102 | Sneha Gupta |

|  |  |
| --- | --- |
| **Table 2: Employee Sales Data** |  |
| **Emp ID** | **Address** |
| 101 | Mumbai |
| 102 | Delhi |

You can manually enter these tables in **Power BI Desktop → Home → Enter Data** or load them from Excel/CSV.

**2️. Create a Relationship**

To join both tables:

* Go to **Model View**.
* Drag the **Emp ID** from Table 1 to **Emp ID** in Table 2.
* This creates a **one-to-one or one-to-many relationship** based on data.

**Cardinality:** One-to-one  
**Cross-filter direction:** Single or Both (prefer Both for full interaction)

**3️. Build a Crosstab (Matrix Visual)**

* Insert a **Matrix visual** from the Visualizations pane.
* From **Table 1**, drag **Name** to **Rows**.
* From **Table 1**, drag **Average Salary Per Year** to **Values**.
* From **Table 2**, drag **Sales Per Year** to **Values**.

This will align the two measures (salary and sales) side by side for each employee.

**Example: Matrix Output**

|  |  |  |
| --- | --- | --- |
| **Name** | **Average Salary Per Year** | **Sales Per Year** |
| Rahul Sharma | ₹6,50,000 | ₹15,00,000 |
| Sneha Gupta | ₹7,20,000 | ₹18,00,000 |

The **Emp ID relationship** ensures the data matches correctly for each employee across both tables.

**Benefits**

* Simplifies combining multiple data sources through relationships.
* Allows unified reporting with values from different tables.
* Maintains data model integrity using primary key (Emp ID).
* Supports drill-down and conditional formatting in Matrix visual.

The **purpose** of creating two tables (e.g., Employee Salary Data and Employee Sales Data) and joining them in a **Crosstab (Matrix visual)** by **Emp ID** in Power BI is to:

1. **Combine Data from Different Sources:**  
   It allows you to combine related data from different tables (e.g., employee salary and sales data) into a unified report, which provides a more comprehensive view of the data.
2. **Facilitate Comparative Analysis:**  
   By using a Crosstab or Matrix visual, you can easily compare metrics (e.g., average salary and sales per year) side by side for each employee, which can reveal insights such as which employees have high sales but low salaries, or vice versa.
3. **Enhance Reporting and Visualization:**  
   Matrix visuals allow you to group and summarize data efficiently. This enables clear, structured reporting where complex data relationships are simplified and made easier to understand.
4. **Improve Data Interaction:**  
   With the ability to drill down into data hierarchies and apply interactive filters, users can explore the combined data at different levels, giving more flexibility in reporting.
5. **Ensure Data Integrity through Relationships:**  
   By linking the two tables using a common identifier (Emp ID), you ensure that the data is joined correctly, and you avoid issues related to misaligned or missing data.
6. **Support Better Decision Making:**  
   Combining salary and sales data into a single view allows managers and decision-makers to analyze employee performance more effectively, identify trends, and make data-driven decisions based on both financial and sales metrics.

In summary, this approach streamlines reporting and analysis by combining multiple related datasets into a single view, allowing for better comparison, insight generation, and decision-making.

**Step-by-Step Guide to Create the Report**

**Step 1: Open Power BI Desktop**

1. Launch Power BI Desktop on your machine.

**Step 2: Import Dataset**

1. **Click on** the **Home** tab.
2. Select **Get Data** → **Excel** (or any other dataset type you’re using).
3. Browse for your file (e.g., Employee\_Salary.xlsx) and click **Open**.
4. Select the relevant table(s) from the dataset and click **Load**.

**Step 3: Creating Visualizations**

1. In the **Fields** pane, drag and drop the columns into a new **Table** visualization on the canvas.
   * Example Columns (from your dataset):
     + EMP ID
     + Name
     + Average Salary
     + Total Sales

**Step 4: Apply Formatting to Table (Font, Colors, Column Headings)**

1. Click on the **Table** visualization to select it.
2. In the **Visualizations** pane, go to the **Format** section (paint roller icon).

**Font and Color Adjustments:**

* **Column Headings**:
  + Expand the **Column Headers** section.
  + Turn **Text Size** to 12-14 for readability.
  + Set **Font Color** and **Background Color** to your choice (e.g., white text on a dark blue background).
* **Values (Data)**:
  + Expand the **Values** section.
  + Set **Text Size** to 10-12 for data readability.
  + Change **Font Color** if desired (e.g., black).
  + Adjust **Background Color** for rows (e.g., light gray) for better contrast.

**Step 5: Apply Row Banding**

1. Under **Format** > **Grid** section, enable **Alternating Row Colors**.
2. Choose a light and dark color for banding (e.g., light gray and white for alternating rows).
3. Set the **Row Padding** to 2-5 for more comfortable spacing.

**Step 6: Add Page Header**

1. To add a **Page Header**, go to the **Insert** tab in the ribbon and click **Text Box**.
2. Type the title of the report, such as "Employee Sales and Salary Report".
3. Format the text (e.g., bold, larger font size like 20-24px) and center-align it.
4. Position the header at the top of the report canvas.

**Step 7: Add Page Footer**

1. To add a **Page Footer**, go to the **Insert** tab and click **Text Box** again.
2. Type your footer content, such as "Report Generated on [Date]" or "Page 1 of X".
3. Format the footer text similarly, and align it to the bottom center or right of the report.
4. Position the footer at the bottom of the report canvas.

**Step 8: Final Adjustments**

1. Adjust the size of the table and page layout for better alignment and balance.
2. Go to the **View** tab and choose **Actual Size** to see the report as it will appear when exported.

**Step 9: Export Report to PDF**

1. Once satisfied with your report layout, go to the **File** tab at the top-left.
2. Click on **Export** → **Export to PDF**.
3. Choose a file location, name the file, and click **Save**.
4. Power BI will generate and save the report as a PDF.

**Step 10: Final Review**

* Open the exported PDF to ensure everything appears as expected.
* Ensure that your page header and footer are in place and the data is well-presented with proper formatting, row banding, and font adjustments.

**What’s a Matrix Visual?**

It’s like a pivot table in Excel — it lets you group your data by one or more fields (rows), and display aggregates (like sum, average) for other fields (values).

**Steps to Create a Matrix Visual Report in Power BI**

**1️. Insert Matrix Visual**

* Open your report page in **Power BI Desktop**.
* From the **Visualizations pane** (on the right), click on the **Matrix visual** icon — it looks like a small grid.

**2️. Add Department as Row Group**

* In the **Fields pane** (on the right), find the Department column from your dataset.
* Drag Department into the **Rows** field well under the Matrix visual.

Now — the Matrix will list **each department as a separate group heading**.

**3️. Add Average Salary and Total Sales as Values**

* Find Average Salary in your **Fields pane**.
* Drag Average Salary into the **Values** field well in the Matrix visual.

Notice it automatically aggregates (usually as *Sum* initially).

**4️. Change Aggregation to Average for Salary**

* In the **Values** section (inside the Matrix visual), click the small dropdown arrow next to Average Salary.
* Select **Average**.

This will compute the **average salary for each department group**.

**5️. Add Total Sales as Sum**

* Drag Total Sales into the **Values** section of the Matrix visual.
* It will default to *Sum*, which is what you want for total sales.

**6️. (Optional) Add Year to Columns**

* If you want to split these summaries by year, drag Year from your Fields pane into the **Columns** field well in the Matrix.

Your matrix will now display **Department as rows**, **Year as columns**, and **aggregated values per department per year**.

**7️. Turn On Subtotals (Group Footer)**

* Click on the Matrix visual.
* In the **Visualizations pane**, click on the **Format (paint roller)** icon.
* Expand the **Subtotals** section.
  + Turn **on** "Per Row Level" for Department
  + Choose background color and font style for subtotals as needed.

This displays **group footers with total values for each department**.

**8️. Style Matrix Headers and Values**

* In the **Format pane (paint roller icon)**:
  + **Row headers:** adjust font size, color, background
  + **Column headers:** adjust font, size, color
  + **Values:** turn on *Row banding* for alternate row colors
  + Set bold headers and light grey alternate rows for readability.

**9️. Add Page Header & Footer**

* From the **Insert tab**:
  + Click **Text box**
  + At the top of your report page → type **Employee Performance Report (2022-2023)**  
    (Style: Bold, 20pt, dark blue, center aligned)
  + At the bottom → type **Confidential Report for Internal Use**  
    (Style: Italic, 10pt, grey)