Accounts receivable credit

### 1. Clustered Bar Chart: Sum of Customer Balance by Profile Class Name

* **Purpose**: This visualization will show the distribution of customer balances across different customer profiles. It helps in identifying which customer profiles (e.g., small business, large business, etc.) hold the highest receivable amounts.
* **Insights**: You can use this to prioritize collections or assess credit risks by profile class.

**X-axis**: Profile Class Name  
**Y-axis**: Sum of Customer Balance

**2. Scatter Plot: Original Transaction Balance vs. Outstanding Balance**

* **Purpose**: This will show the relationship between the original balance and the outstanding balance for transactions.
* **Insights**: Identify large transactions with high outstanding balances, and analyze trends in payment delays.

### X-axis: Original Transaction Balance Y-axis: Outstanding Balance 3. Pie Chart: Distribution of Outstanding Balance by Unpaid Reason

* **Purpose**: This chart will help you understand the most common reasons for unpaid balances.
* **Insights**: This can provide insights into recurring reasons for overdue payments, helping you address systemic issues (e.g., disputes, service quality issues).

**Values**: Sum of Outstanding Balance  
**Legend**: Unpaid Reason  
  
including blank/ excluding blank values

**4. Line Chart: Count of Transactions Over Time**

* **Purpose**: This will track the volume of transactions over time using the Transaction Date.
* **Insights**: See trends in transaction activity over time and identify periods with higher credit activity.

**X-axis**: Transaction Date  
**Y-axis**: Count of Transactions

Account receivable debit

### 1. Clustered Bar Chart: Sum of Customer Balance by Profile Class Name

* **Purpose**: This visualization will show the distribution of customer balances across different customer profiles. It helps in identifying which customer profiles (e.g., small business, large business, etc.) hold the highest receivable amounts.
* **Insights**: You can use this to prioritize collections or assess credit risks by profile class.

**X-axis**: Profile Class Name  
**Y-axis**: Sum of Customer Balance

**2. Scatter Plot: Original Transaction Balance vs. Outstanding Balance**

* **Purpose**: This will show the relationship between the original balance and the outstanding balance for transactions.
* **Insights**: Identify large transactions with high outstanding balances, and analyze trends in payment delays.

### X-axis: Original Transaction Balance Y-axis: Outstanding Balance 3. Pie Chart: Distribution of Outstanding Balance by Unpaid Reason

* **Purpose**: This chart will help you understand the most common reasons for unpaid balances.
* **Insights**: This can provide insights into recurring reasons for overdue payments, helping you address systemic issues (e.g., disputes, service quality issues).

**Values**: Sum of Outstanding Balance  
**Legend**: Unpaid Reason  
  
including blank/ excluding blank values

**4. Line Chart: Count of Transactions Over Time**

* **Purpose**: This will track the volume of transactions over time using the Transaction Date.
* **Insights**: See trends in transaction activity over time and identify periods with higher credit activity.

**X-axis**: Transaction Date  
**Y-axis**: Count of Transactions

### All Expenses Sum of Reimbursable Amount by Expense Type (Pie Chart)

* **How**: Drag **Expense Type** to the **Legend** and **Reimbursable Amount** to the **Values**.
* **Why**: This visual helps to understand the distribution of expenses by type, showing which expense types contribute the most to the total reimbursable amount.

**2. Total Reimbursable Amount Over Time (Line Chart)**

* **How**: Drag **Expense Report Date** to the **Axis** and **Reimbursable Amount** to the **Values**.
* **Why**: This visual tracks how the total reimbursable amount changes over time, highlighting trends and seasonality in expenses.

**3. Reimbursable Amount by Cost Center Code (Clustered Bar Chart)**

* **How**: Drag **Cost Center Code** to the **Axis** and **Reimbursable Amount** to the **Values**.
* **Why**: This visual shows how expenses are distributed across different cost centers, helping to identify areas with higher expenses.

**4. Count of Expense Reports by Report Status (Stacked Column Chart)**

* **How**: Drag **Report Status** to the **Axis** and **Report Number** to the **Values**.
* **Why**: This visual provides insights into the status distribution of expense reports, showing how many are pending, approved, or rejected.

**5. Average Reimbursable Amount by Employee (Treemap) top 10,**

* **How**: Drag **Employee Name** to the **Group** and **Reimbursable Amount** to the **Values**.
* **Why**: This visual displays the average reimbursable amount by employee, highlighting who submits the highest or lowest expenses.

**6. Reimbursable Amount vs. Trip Distance (Scatter Plot)**

* **How**: Drag **Trip Distance** to the **X-axis** and **Reimbursable Amount** to the **Y-axis**.
* **Why**: This visual helps to explore if there is a relationship between trip distance and the reimbursable amount, revealing if higher distances result in higher expenses.

Daily account balance

### Visualizations

1. **Daily Ledger Balances (Line Chart)**

**Purpose:** To visualize how ledger balances change over time.

**Steps:**

* + Insert a **Line Chart** visual.
  + Drag **Balance date** to the **Axis**.
  + Drag **Last Night's Ledger**, **Today's Ledger**, **Start Of Day Ledger** to the **Values**.
  + Format the chart to clearly distinguish between the different ledger values over time.

1. **Cleared Balances Comparison (Clustered Column Chart)**

**Purpose:** To compare cleared balances over time.

**Steps:**

* + Insert a **Clustered Column Chart** visual.
  + Drag **Balance date** to the **Axis**.
  + Drag **Last Night's Cleared** and **Today's Cleared** to the **Values**.
  + Format the chart for clarity.

1. **Daily Changes in Ledger Balances (Stacked Area Chart)**

**Purpose:** To show the distribution and changes in ledger balances over time.

**Steps:**

* + Insert a **Stacked Area Chart** visual.
  + Drag **Balance date** to the **Axis**.
  + Drag **Last Night's Ledger**, **Today's Ledger**, **Start Of Day Ledger** to the **Values**.
  + Format the chart to display the changes effectively.

1. **Balance Changes Overview (Table)**

**Purpose:** To view a detailed table of balances.

**Steps:**

* + Insert a **Table** visual.
  + Drag **Account Identifier**, **Balance date**, **Last Night's Ledger**, **Today's Ledger**, **Last Night's Cleared**, **Today's Cleared**, **Start Of Day Ledger**, **Start Of Day Cleared** to the **Values**.

**KPIs**

1. **Total Cleared Amount**
   * Measure to calculate the total cleared amount over the period.

DAX

Copy code

Total\_Cleared\_Amount = SUM('Table'[Last Night's Cleared]) + SUM('Table'[Today's Cleared])

1. **Total Ledger Balance**
   * Measure to calculate the total ledger balance at the end of the day.

DAX

Copy code

Total\_Ledger\_Balance = SUM('Table'[Today's Ledger])

1. **Net Change in Ledger**
   * Measure to calculate the net change in ledger balance over the period.

DAX

Copy code

Net\_Change = SUM('Table'[Today's Ledger]) - SUM('Table'[Last Night's Ledger])

**Slicers**

1. **Date Slicer**
   * Allows users to filter data based on specific date ranges.
   * Drag **Balance date** to the **Slicer**.
2. **Account Identifier Slicer**
   * Filters data by specific account identifiers.
   * Drag **Account Identifier** to the **Slicer**.

**1. Treemap: Sum of # of Requests by Department Name (top10)**

* **Purpose:** To visualize the distribution of help desk requests across different departments.
* **How to Create:**
  + **Insert Treemap Visual**
  + **Values:** # of Requests (Summarize by sum)
  + **Group By:** Department Name
  + **Tooltips:** You might add additional details like Category Name or Job Name for more context.

**2. Line Chart: Count of Average Time to Resolve (Days) by Category Name(top 5)**

* **Purpose:** To track how the average time to resolve tickets varies by category over time.
* **How to Create:**
  + **Insert Line Chart Visual**
  + **X-axis:** Creation Date (Ensure it's set to a date hierarchy level like month or quarter)
  + **Y-axis:** Average Time to Resolve (Days) (Aggregate by average)
  + **Legend:** Category Name
  + **Tooltips:** Include Department Name, Job Name

**3. Clustered Bar Chart: Sum of # of Requests by Agent Assignment Number**

* **Purpose:** To evaluate the workload of each agent based on the number of requests they handle.
* **How to Create:**
  + **Insert Clustered Bar Chart Visual**
  + **X-axis:** Agent Assignment Number
  + **Y-axis:** # of Requests (Summarize by sum)
  + **Tooltips:** Add Average Time to Resolve (Days), Department Name for additional insights.

**4. Stacked Column Chart: Sum of # of Requests and Count of Queue Name by Status**

* **Purpose:** To understand the volume and distribution of requests by their status (Open, Closed, etc.).
* **How to Create:**
  + **Insert Stacked Column Chart Visual**
  + **X-axis:** Status
  + **Y-axis:**
    - Add # of Requests (Sum)
    - Add Queue Name (Count)
  + **Legend:** Use Queue Name or another relevant dimension.

**5. Clustered Column Chart: Count of Department Name by Average Time to Resolve (Days) (top5)**

* **Purpose:** To compare the average resolution time across different departments.
* **How to Create:**
  + **Insert Clustered Column Chart Visual**
  + **X-axis:** Department Name
  + **Y-axis:** Average Time to Resolve (Days) (Aggregate by average)
  + **Tooltips:** Add # of Requests, Category Name

**Helping desk**

**1. Treemap: Sum of # of Requests by Department Name**

* **Purpose:** To visualize the distribution of help desk requests across different departments.
* **How to Create:**
  + **Insert Treemap Visual**
  + **Values:** # of Requests (Summarize by sum)
  + **Group By:** Department Name
  + **Tooltips:** You might add additional details like Category Name or Job Name for more context.

**2. Line Chart: Count of Average Time to Resolve (Days) by Category Name**

* **Purpose:** To track how the average time to resolve tickets varies by category over time.
* **How to Create:**
  + **Insert Line Chart Visual**
  + **X-axis:** Creation Date (Ensure it's set to a date hierarchy level like month or quarter)
  + **Y-axis:** Average Time to Resolve (Days) (Aggregate by average)
  + **Legend:** Category Name
  + **Tooltips:** Include Department Name, Job Name

**3. Clustered Bar Chart: Sum of # of Requests by Agent Assignment Number**

* **Purpose:** To evaluate the workload of each agent based on the number of requests they handle.
* **How to Create:**
  + **Insert Clustered Bar Chart Visual**
  + **X-axis:** Agent Assignment Number
  + **Y-axis:** # of Requests (Summarize by sum)
  + **Tooltips:** Add Average Time to Resolve (Days), Department Name for additional insights.

**4. Stacked Column Chart: Sum of # of Requests and Count of Queue Name by Status**

* **Purpose:** To understand the volume and distribution of requests by their status (Open, Closed, etc.).
* **How to Create:**
  + **Insert Stacked Column Chart Visual**
  + **X-axis:** Status
  + **Y-axis:**
    - Add # of Requests (Sum)
    - Add Queue Name (Count)
  + **Legend:** Use Queue Name or another relevant dimension.

**5. Clustered Column Chart: Count of Department Name by Average Time to Resolve (Days)**

* **Purpose:** To compare the average resolution time across different departments.
* **How to Create:**
  + **Insert Clustered Column Chart Visual**
  + **X-axis:** Department Name
  + **Y-axis:** Average Time to Resolve (Days) (Aggregate by average)
  + **Tooltips:** Add # of Requests, Category Name

These visuals and KPIs will help you to analyze various aspects of help desk data effectively, allowing you to understand request distribution, resolution times, and agent performance.

### Invoice extratcec 1. Total Amount by Hold Reason (Clustered Bar Chart)

**Steps:**

1. **Load Data**: Import the "Invoice\_Extract 29 July.xlsx" file into Power BI.
2. **Create Clustered Bar Chart**:
   * Go to the “Visualizations” pane and select the “Clustered Bar Chart” visual.
   * Drag Hold Reason to the **Axis** field.
   * Drag Amount to the **Values** field.
3. **Add Slicers**:
   * Add slicers for Invoice Type, Supplier Site, and Currency.
   * To add a slicer, go to the “Visualizations” pane, select “Slicer”, and drag the desired field into the slicer.

**2. bar chart, assifned to vs total amount**

**4. Invoice Amount Over Time (Line Chart)**

**Steps:**

1. **Create Line Chart**:
   * Go to the “Visualizations” pane and select the “Line Chart” visual.
   * Drag Due Date (or another time-based field) to the **Axis** field.
   * Drag Amount to the **Values** field.
2. **Add Slicers**:
   * Add slicers for Invoice Type, Supplier Site, and Currency.

**5. Top 10 Invoices by Amount (Table or Bar Chart)**

**Steps:**

1. **Create Table**:
   * Go to the “Visualizations” pane and select the “Table” visual.
   * Drag Invoice Number and Amount to the **Values** field.
   * Sort the table by Amount in descending order.
   * Use the top N filter to show only the top 10.
2. **Create Bar Chart** (alternative):
   * Go to the “Visualizations” pane and select the “Bar Chart” visual.
   * Drag Invoice Number to the **Axis** field.
   * Drag Amount to the **Values** field.
   * Sort by Amount in descending order and use the top N filter to show the top 10.

Slicers  
invoice type

Payment term

Hold reason

Kpis

Total amount

Count of supplier site

Distinct Count of total assigned to

MAILBOX

Visulas,

Line charts  
sum of accounts payabbe mailbox by month  
count of blocking earliest unread\_1 by month

Pie charts  
sum of crditors creations mailbox,sum of creatitors manua mailbox,sumof creditors mailbox and asum of account payble mailbox

Clusrered column chart

Sum of Creditors Manual Mailbox, Sum of Creditors Mailbox and Sum of Creditors Creations Mailbox by Month

Kpis

Sum of creditors creations mailbox

Sum of ceditors manula mailbox

Sum of creditors mailbox

Sum of accounts pable mailbox

Sliers,  
date

Reenue monitoring committee file  
Sum of RevisedBudget £ by Cost Centres top 5 and bottom 5,  
  
cluster col chart  
Sum of Full Year Variance by Cost Centres top 5 and bottom 5

Scatter plot

#### **Scatter Plot: % Variance vs. Full Year Variance**

* **Purpose**: Explore the relationship between percentage variance and full year variance.
* **X-axis**: % Variance, non summarize
* **Y-axis**: Full Year Variance, non summarize
* **Legend**: Non

Kpis

Sum of revised budget

Sum of totl full year projection

Distinct count of cost senters

For the Customer Aging by Bucket List Report table with the following columns:

* **Customer Name**
* **CAN**
* **Bill To Site**
* **Collector**
* **Trans No.**
* **Instalment Number**
* **Due Date**
* **Aging Bucket**
* **Original Balance**
* **Collected**
* **Amount Remaining**
* **Profile Class**

Here’s how you can visualize this data, and what KPIs and slicers you might consider:

### ****1. Visualizations****

#### **1.1. Stacked Bar Chart: Balance by Aging Bucket**

* **Purpose**: To see the distribution of outstanding balances across different aging buckets.
* **X-axis**: Aging Bucket (e.g., 0-30 days, 31-60 days, 61-90 days, 90+ days)
* **Y-axis**: Original Balance
* **Legend**: None (Stacked bars show the total balance per bucket)
* **How**:
  1. Insert a Stacked Bar Chart.
  2. Set the X-axis to Aging Bucket.
  3. Set the Y-axis to Original Balance.

#### **1.2. Clustered Column Chart: Amount Remaining by Customer**

* **Purpose**: To compare the amount remaining for different customers.
* **X-axis**: Customer Name
* **Y-axis**: Amount Remaining
* **Legend**: None
* **How**:
  1. Insert a Clustered Column Chart.
  2. Set the X-axis to Customer Name.
  3. Set the Y-axis to Amount Remaining.

#### **1.3. Line Chart: Collected vs. Original Balance Over Time**

* **Purpose**: To track how collected amounts and original balances change over time.
* **X-axis**: Due Date
* **Y-axis**: Values
* **Legend**: Collected, Original Balance
* **How**:
  1. Insert a Line Chart.
  2. Set the X-axis to Due Date.
  3. Set the Y-axis to Collected and Original Balance.
  4. Add Collected and Original Balance to the Legend.

#### **1.4. Pie Chart: Distribution of Original Balance by Aging Bucket**

* **Purpose**: To show the proportion of the original balance in each aging bucket.
* **Values**: Original Balance
* **Legend**: Aging Bucket
* **How**:
  1. Insert a Pie Chart.
  2. Set the Values to Original Balance.
  3. Set the Legend to Aging Bucket.

#### **1.5. Treemap: Balance by Profile Class**

* **Purpose**: To visualize the distribution of outstanding balances by profile class.
* **Group**: Profile Class
* **Values**: Original Balance
* **How**:
  1. Insert a Treemap.
  2. Set the Group to Profile Class.
  3. Set the Values to Original Balance.

### ****2. KPIs****

#### **2.1. Total Outstanding Balance**

* **Purpose**: To measure the total outstanding balance across all aging buckets.
* **KPI Measure**:

DAX

Copy code

Total\_Outstanding\_Balance = SUM('Customer Aging by Bucket List Report'[Original Balance])

#### **2.2. Total Collected Amount**

* **Purpose**: To measure the total amount collected across all transactions.
* **KPI Measure**:

DAX

Copy code

Total\_Collected\_Amount = SUM('Customer Aging by Bucket List Report'[Collected])

#### **2.3. Total Amount Remaining**

* **Purpose**: To measure the total amount remaining to be collected.
* **KPI Measure**:

DAX

Copy code

Total\_Amount\_Remaining = SUM('Customer Aging by Bucket List Report'[Amount Remaining])

### ****3. Slicers****

#### **3.1. Aging Bucket Slicer**

* **Purpose**: To filter visualizations by specific aging buckets.
* **Slicer Field**: Aging Bucket

#### **3.2. Customer Slicer**

* **Purpose**: To filter visualizations by specific customers.
* **Slicer Field**: Customer Name

#### **3.3. Profile Class Slicer**

* **Purpose**: To filter visualizations by profile class.
* **Slicer Field**: Profile Class