Sales Performance Analysis Report Documentation

1. Overview

Report Name: Sales Performance Analysis Report
Module: Sales
Report Type: Script Report
Purpose: To provide actionable insights into sales performance by analyzing data across multiple DocTypes, facilitating informed decision-making for sales management.

2. Objectives

- Analyse sales performance across products, salespersons, and time periods.
- Calculate key metrics:
  - Total revenue by product category.
  - Sales representative performance comparison.
  - Order fulfilment rate.
  - Payment collection efficiency.
- Implement dynamic filters for customizable data views.
- Provide visual representations through charts and sortable tabular data.

3. Data Sources

The report aggregates data from the following DocTypes:

- **Sales Order**
- **Sales Invoice**
- **Customer**
- **Item**
- **Sales Person**
- **Delivery Note**
- **Payment Entry Reference**

4. Data Relationships

- **Sales Order** is linked to **Sales Invoice** via the sales_order field.
- **Sales Invoice** items are associated with **Item** via the item_code field.
- **Sales Order** is connected to **Delivery Note** through the against_sales_order field in **Delivery Note Item**.
- **Sales Order** includes **Sales Team** entries, linking to **Sales Person**.
- **Payment Entry Reference** connects payments to **Sales Invoice** via the reference_name field.

5. Filters Implemented

- **Date Range:** Allows selection of specific time periods with presets for quarter, month, and year.
- **Customer Group/Type:** Filters data based on customer classifications.
- **Item Group/Category:** Enables analysis by product categories.

6. Key Metrics Calculated

- **Total Revenue by Product Category:** Sum of net amounts from sales invoices grouped by item group.
- **Sales Representative Performance Comparison:** Aggregated sales figures per sales person.
- **Order Fulfilment Rate:** Ratio of delivered orders to total sales orders, expressed as a percentage.
- **Payment Collection Efficiency:** Ratio of payments received to total invoiced amounts, expressed as a percentage.

7. Technical Implementation

7.1. Report Files

- **Python File:** sales_performance_analysis_report.py
- **JavaScript File:** sales_performance_analysis_report.js

7.2. SQL Query Optimization

- Utilized LEFT JOIN statements to combine data across DocTypes efficiently.
- Applied WHERE clauses to filter data based on user-selected filters.
- Indexed relevant fields to enhance query performance.

7.3. Error Handling

- Implemented try-except blocks to catch and log exceptions during data retrieval.
- Validated the existence of necessary fields and tables before executing queries.

7.4. Permissions

- Restricted report access to users with the "Sales Manager" role.
- Ensured compliance with ERPNext's role-based permission system.

8. Visualization

- **Chart:** Implemented a bar chart displaying total revenue per product category.
- **Tabular View:** Provided a sortable table with columns for sales person, item group, total revenue, order fulfilment rate, and payment collection efficiency.

9. Performance Considerations

- Optimized SQL queries to reduce execution time.
- Limited data retrieval to necessary fields.
- Implemented pagination for large datasets to improve load times.

10. Benefits to Sales Management

- Enables identification of top-performing products and salespersons.
- Assists in recognizing areas with low order fulfilment or payment collection rates.
- Facilitates strategic planning through data-driven insights.

11. Screenshots

Note: Include the following screenshots in your documentation.

- **Prepared Report Details:** Showcasing filters applied and report generation status.
- **Report Output View:** Displaying the generated report with calculated metrics and visualizations.

12. Conclusion

- The "Sales Performance Analysis Report" effectively consolidates critical sales data, providing comprehensive insights that empower sales management to make informed decisions. Its implementation adheres to ERPNext's best practices, ensuring scalability and maintainability.ss

NOTE:

- ➢ According to screenshot that I have given in that one screenshot shows the output of report (show report) in that there is a column named Payment Collection Efficiency in that it values greater than 100% indicate overpayments, advance payments, or cleared past dues beyond the current invoiced amount.
    - To fix this issue I have write the code for that which is to add in .py file of report. The code is:

```
efficiency = 0
if invoiced_amount and payment_received:
    efficiency = (payment_received / invoiced_amount) * 100
    efficiency = min(efficiency, 100)
```