#### **1. Objective**

The goal is to retrieve and compare scheme data for a specified time period (start and end date) with relevant sales data, including comparisons across two financial years. If no dates are provided, the system defaults to the current financial year and compares sales data across the last two financial years, providing detailed sales figures for all months of the current year.

#### **2. Key Functional Requirements**

##### **2.1 Date Handling & Default Values**

* If startDate and endDate are not provided:
  + The system will set the default to the **current financial year**:
    - If the current month is April or later, the default period is from **April 1st** of the current year to **March 31st** of the next year.
    - If the current month is before April, the default period is from **April 1st** of the previous year to **March 31st** of the current year.

##### **2.2 Fetch Eligible Schemes for the Date Range**

* **Data Source**: tbl\_scheme\_master & tbl\_scheme\_category
  + The system fetches schemes that are eligible within the selected date range (startDate, endDate).
  + A scheme is considered eligible if its **start date** and **end date** overlap with the selected period.
  + **Filters Used:**
    - **start\_date <= selected startDate**
    - **end\_date >= selected endDate**
  + **Data Shown**:
    - **Scheme Details:** scheme\_name, scheme\_type, scheme\_brief, criteria, start\_date, end\_date, period, state\_ids, scheme\_category\_id, scheme\_category\_name, account\_type\_id, product\_type, sales\_category
    - **From tbl\_scheme\_master and tbl\_scheme\_category tables**

##### **2.3 Account Filters**

* **Account Type Filter**:
  + Filters schemes based on the selected account\_type\_id. If no account type is provided, all account types are considered.
  + **Data Source**: tbl\_scheme\_master
  + **Filter Used**: account\_type\_id = selected account\_type\_id

##### **2.4 Sales Category and Product Type Filters**

* **Sales Category Filter**:
  + Filters the schemes by the provided sales\_category (could be multiple).
  + **Data Source**: tbl\_scheme\_master
  + **Filter Used**: sales\_category (if provided)
* **Product Type Filter**:
  + Filters the schemes by product\_type (could be multiple).
  + **Data Source**: tbl\_scheme\_master
  + **Filter Used**: product\_type

##### **2.5 State Filters**

* **State Filter**:
  + Filters based on state\_ids. If not provided, the system fetches default state IDs from the state\_tbl where working\_status = 1.
  + **Data Source**: state\_tbl
  + **Filter Used**: state\_ids

##### **2.6 Sales Data Fetching for Two Financial Years**

* **Data Source**: dispatch\_item (sales data)
  + The system fetches and aggregates sales data for **two financial years** (the current and the previous year) within the provided date range, for the selected states (state\_ids).
  + **Filters Used**:
    - **state\_id IN (selected state\_ids)**
    - **date >= startDate - 2 years** and **date <= endDate**
  + **Data Shown**:
    - **Financial Year** (financial\_year)
    - **Sales Achievement** (sales\_achievement)

##### **2.7 Sales Comparison for Two Financial Years**

* The system compares **sales data for the last two financial years**, by aggregating monthly sales figures for each year and generating the corresponding comparison for the current financial year (cy\_data) and the previous year (ly\_data).

##### **2.8 Monthly Sales Breakdown**

* For each month of the **current financial year**, the system provides monthly sales comparisons across the two financial years (current and previous). If no startDate and endDate are provided, the system defaults to comparing all months of the **current financial year**.

#### **3. Non-Functional Requirements**

##### **3.1 Performance Optimization**

* Use optimized queries and indexing to handle large datasets efficiently.
* Ensure queries handle the aggregation of sales data and scheme results without timeouts.

##### **3.2 Data Accuracy**

* Ensure that the retrieved data matches the stored scheme and sales information, flagging discrepancies if necessary.

##### **3.3 Scalability**

* The system must be capable of handling a growing number of schemes and sales data over time.

#### **4. DB Structure Analysis**

##### **4.1 Tables Involved**

* **tbl\_scheme\_master**: Contains scheme details such as scheme name, start and end dates, account type, sales category, and product type.
  + Key Fields: scheme\_name, start\_date, end\_date, account\_type\_id, sales\_category, product\_type, state\_ids
* **tbl\_scheme\_category**: Contains information about the scheme category (used for joining with tbl\_scheme\_master).
  + Key Fields: scheme\_category\_id, scheme\_category\_name
* **dispatch\_item**: Contains sales data for accounts.
  + Key Fields: state\_id, date, quantity (sales amount), sales\_achievement
* **state\_tbl**: Contains state information, used to fetch default state\_ids if not provided.
  + Key Fields: state\_id, working\_status

#### **5. Optimizations**

##### **5.1 Precompute Sales Data**

* Precompute and store sales data by month and financial year to optimize performance when querying large datasets.

##### **5.2 Indexes and Query Optimization**

* Create indexes on fields like account\_type\_id, sales\_category, state\_id, and product\_type to speed up querying and filtering operations.

#### **6. Process Flow**

1. **User Inputs**:
   1. **Required Inputs**: startDate, endDate, account\_type\_id, sales\_category, product\_type, category\_id, state\_ids (optional).
2. **Validate and Process Filters**:
   1. Convert sales\_category, category\_id, and product\_type into arrays if provided as comma-separated strings.
   2. Fetch default state\_ids if not provided.
3. **Determine Date Range**:
   1. If startDate and endDate are not provided, the system defaults to the current financial year. If the current month is after March, the system will consider the current financial year from April 1st to March 31st of the following year. Otherwise, it will consider the previous financial year.
4. **Fetch Eligible Scheme Data**:
   1. Query the tbl\_scheme\_master table to fetch eligible schemes based on the selected date range and the provided filters (account\_type\_id, sales\_category, product\_type, and state\_ids).
5. **Fetch Sales Data for Two Financial Years**:
   1. Fetch the sales data for the last two financial years using the dispatch\_item table, aggregated by financial year.
6. **Sales Comparison**:
   1. Compare sales data for the last two financial years and generate a monthly breakdown of sales achievements.
7. **Generate Report**:
   1. The final report will include:
      1. **Scheme Results**: Details of schemes for the current financial year and previous year.
      2. **Sales Data for Two Financial Years**: Sales achievements for the past two years.
      3. **Monthly Sales Breakdown**: Breakdown of monthly sales achievements for the current and previous financial years.

#### **7. Example API Response**

The API response will include:

* **schemeResultsCurrentYear**: List of eligible schemes for the current financial year.
* **schemeResultsPreviousYear**: List of eligible schemes for the previous financial year.
* **salesDataFY**: Sales data for the last two financial years.
* **detail**: A detailed report including monthly comparisons of sales achievements and schemes.

json

{  
 "error": false,  
 "msg": "Ok",  
 "detail": [  
 {  
 "month": "January",  
 "sale\_data": [  
 { "fy": "2023-2024", "sale": "50036.73" },  
 { "fy": "2022-2023", "sale": "44772.48" }  
 ],  
 "cy\_data": [  
 {  
 "scheme\_name": "Scheme A",  
 "scheme\_brief": "Scheme A Brief",  
 "period": "April 2023 - March 2024",  
 "start\_date": "2023-04-01",  
 "end\_date": "2024-03-31",  
 "state\_ids": "1,2,3",  
 "sales\_category": "Normal",  
 "product\_type": "TMTONLY",  
 "category\_id": "2",  
 "scheme\_category": "Category A"  
 }  
 ],  
 "ly\_data": [  
 {  
 "scheme\_name": "Scheme B",  
 "scheme\_brief": "Scheme B Brief",  
 "period": "April 2022 - March 2023",  
 "start\_date": "2022-04-01",  
 "end\_date": "2023-03-31",  
 "state\_ids": "1,3",  
 "sales\_category": "Discount",  
 "product\_type": "ALL",  
 "category\_id": "3",  
 "scheme\_category": "Category B"  
 }  
 ]  
 },  
 ...  
 ]  
}

#### **8. Conclusion**

The system efficiently handles the retrieval of scheme details and sales data for the specified date range. It supports flexible filtering based on account type, sales category, product type, and states, and automatically defaults to the current financial year when no dates are provided. The generated report compares sales data across two financial years and provides monthly breakdowns for easy analysis.