**Data Warehouse and Reporting Specification**

# **1. Reporting Requirements Analysis**

Conduct a thorough review of the reporting needs to ensure the business receives exactly the data and insights it requires. This includes understanding the context of each report, the end users, and the decisions they will support.

## **1.1. GAP Analysis**

Compare the validated list of fields against the existing database schema. Identify all missing fields and determine the best approach for adding them to the data warehouse — whether through schema changes, ETL logic, or derived calculations. Document any required upstream changes in source systems.

# **2. ETL Development**

The goal is to update and extend the ETL processes to extract, transform, and load all required data into the data warehouse, ensuring it is accurate, complete, and up to date.

## **2.1. Source System Extraction**

Identify all source systems, such as the core banking platform and CRM. Develop extraction processes — SQL queries, API calls, or flat file imports — that retrieve the required data fields without unnecessary overhead.

## **2.2. Transformation Logic**

Implement transformations to normalize the data that was extracted, map transaction types to standardized codes. Ensure that all date and currency formats are standardized across the warehouse.

**2.3. Data Loading**

Integrate the transformed data into the warehouse’s fact and dimension tables, updating tables such as Fact\_Transactions and Dim\_Accounts. Implement an incremental load strategy to handle daily updates efficiently without reloading historical data unnecessarily.

# **3. SQL Reporting Layer**

Build database views or stored procedures that consolidate and present the required data in a format ready for reporting and visualization.

## **3.1. Account Portfolio Overview**

The first report concerns Account Portfolio Overview. The report must be in tabular format, and it must contain the following fields.

* Account Number
* Customer Number
* Customer Last Name
* Customer First Name
* Type of Account
* Account Status

## **3.2. Account Transaction History View**

The second report displays detailed records of all transactions per account. This report also must be in tabular format. It must contain the following fields.

* Account Number
* Transaction Number
* Transaction Date
* Transaction Type
* Transaction Amount
* Is Reversed

**3.3. Customer Contact Information Overview**

The report must be in tabular format, and it must contain the following fields:

* Customer Number
* Customer Last Name
* Customer First Name
* Address
* Country
* National Postal Code
* Is Primary Contact

# **4. Power BI / Visualization Layer**

Create user-friendly dashboards in Power BI that present the data from the SQL layer in a clear, interactive, and actionable way.

**4.1. Data Connectivity**

Set up a secure connection between Power BI and the data warehouse. Where appropriate, use parameterized queries to enable dynamic filtering at the source, minimizing data load times.

**4.2. Report Layout and Navigation**

Develop a dedicated **Account Portfolio Overview** tab that displays account data with filters for account status and type. Create a **Transaction History** tab with slicers for date ranges, transaction types, and other relevant criteria. Ensure navigation between tabs is intuitive.

**4.3. KPIs and Summary Metrics**

Add calculated metrics such as total transaction amounts per period, average transaction values per account, and counts of reversed transactions. Use clear visualizations — cards, line charts, bar charts — to make trends and outliers easy to spot.

**5. Quality Assurance and Testing**

Verify that all new data, reports, and dashboards are correct, performant, and aligned with business needs.

**5.1. Data Validation**

Compare the output from the reports with source system records to ensure accuracy. Perform random sampling across different accounts and transactions to validate both summary and detail-level data.

**5.2. Performance Testing**

Run stress tests on SQL queries and Power BI dashboards to ensure they can handle expected data volumes without degradation. Fine-tune indexing, joins, and filtering strategies as needed to maintain responsiveness.

**6. Documentation and Handover**

Deliver comprehensive documentation so that both technical teams and business users can maintain and use the solution effectively.

**6.1. Technical Documentation**

Update the entity relationship diagrams to reflect schema changes. Document ETL job flows, including schedules and dependencies. Provide detailed definitions of SQL views, including business logic applied in each.

**6.2. User Documentation**

Create a Power BI user guide with screenshots and instructions on using filters, slicers, and navigation. Include a business glossary explaining each data field and KPI to ensure consistent interpretation across the organization.