Data and prognosis of activity
700
600
500
400
300
200
100
0
120
180
250
390
200
280
400
410
360
470
600
610
440
520
Detailed information of changing business activity of subdivisions of main company
Subdivision 1
Subdivision 2
Subdivision 3
Best subdivision value
Value of all subdivisions
Product placement in different regions
Main company
Subdivisions
Others
Competitors

Finpro Bank

Data Architecture, Implementation and Recommendations

Outline

- Introduction
- Data acquisition and cleaning
- Database design and setup
- Evaluation and interpretation
- Data integration and security
- Data reporting
- General handover documentation
- Insights and recommendations
- Summary
- Appendix

Introduction

Purpose of the presentation

The purpose of this presentation is to provide a comprehensive overview of FinPro Bank’s data management strategy. We will explore how an end-to-end data pipeline enhances decision-making, customer satisfaction, and fraud detection. This session will highlight the tools and technologies used, from data preparation in Excel to business intelligence insights in Tableau, demonstrating how FinPro Bank optimizes its data operations for strategic growth.

Summary of work done on the project

In this project, we have built a seamless data management system to drive informed decision-making at FinPro Bank. Our work began with cleaning and preparing raw data in Excel to ensure accuracy and usability. We then organized the data within a MySQL relational database, allowing for structured storage and retrieval. To enable deeper analysis, we transferred key datasets into a Postgres data warehouse, ensuring data integrity and performance. Finally, we transformed the data into actionable insights by creating strategic dashboards in Tableau, enabling stakeholders to track key performance indicators, customer trends, and fraud patterns in real time.

Data acquisition and cleaning

Data cleaning documentation (1 of 3)

- Customer Data Sheet provides detailed information about individual customers, including their names, contact details, demographics, account details, and more.
- Transaction Data Sheet records the transactional history of customers, encompassing details like transaction dates, types, amounts, and locations.
- Account Type Sheet categorizes different account types offered by the bank.
- Countries Sheet lists various countries and their corresponding codes and continents.
- US_Cities Sheet provides a list of US cities and their associated codes.

Data cleaning documentation (2 of 3)

Data Import & Initial Review

- Imported raw data from multiple sources into Excel.
- Conducted an initial scan to identify inconsistencies, missing values, and potential outliers.

Handling Missing Values

- Used filtering and conditional formatting to detect missing data.
- Filled missing numerical values using the mean or median where appropriate.
- For categorical data, used the mode or forward-fill/backward-fill techniques to maintain consistency.
- Removed records with excessive missing values if they could not be reasonably imputed.

Removing Duplicates

- Used the **Remove Duplicates** feature to eliminate redundant records.
- Verified duplicate entries by checking unique identifiers such as customer ID or transaction ID before deletion.

Data cleaning documentation (3 of 3)

```
= DATEDIF([DateOfBirth], TODAY(), "Y")
```

```
= TEXT([TransactionDate], "MMMM")
```

```
= IF(A2>=10000, "High Balance", IF(A2>=5000, "Medium Balance", "Low Balance"))
```

Analysis and visualization using MS Excel (1 of 2)

1. Customer Metrics

- **Customer Retention Rate** = (Customers at End of Period – New Customers) / Customers at Start of Period × 100
  Measures how well the bank retains existing customers.
- **Customer Lifetime Value (CLV)** = Average Transaction Value × Purchase Frequency × Customer Lifespan
  Estimates the total revenue expected from a customer over their banking relationship.

2. Fraud Detection KPIs

- **Suspicious Transaction Percentage** = (Flagged Transactions / Total Transactions) × 100
  Monitors anomalies in transaction behavior.
- **Unusual Login Activity** = Count of logins from new locations or devices.
  Identifies potential account breaches.

Analysis and visualization using MS Excel (2 of 2)

Regular Data Cleaning

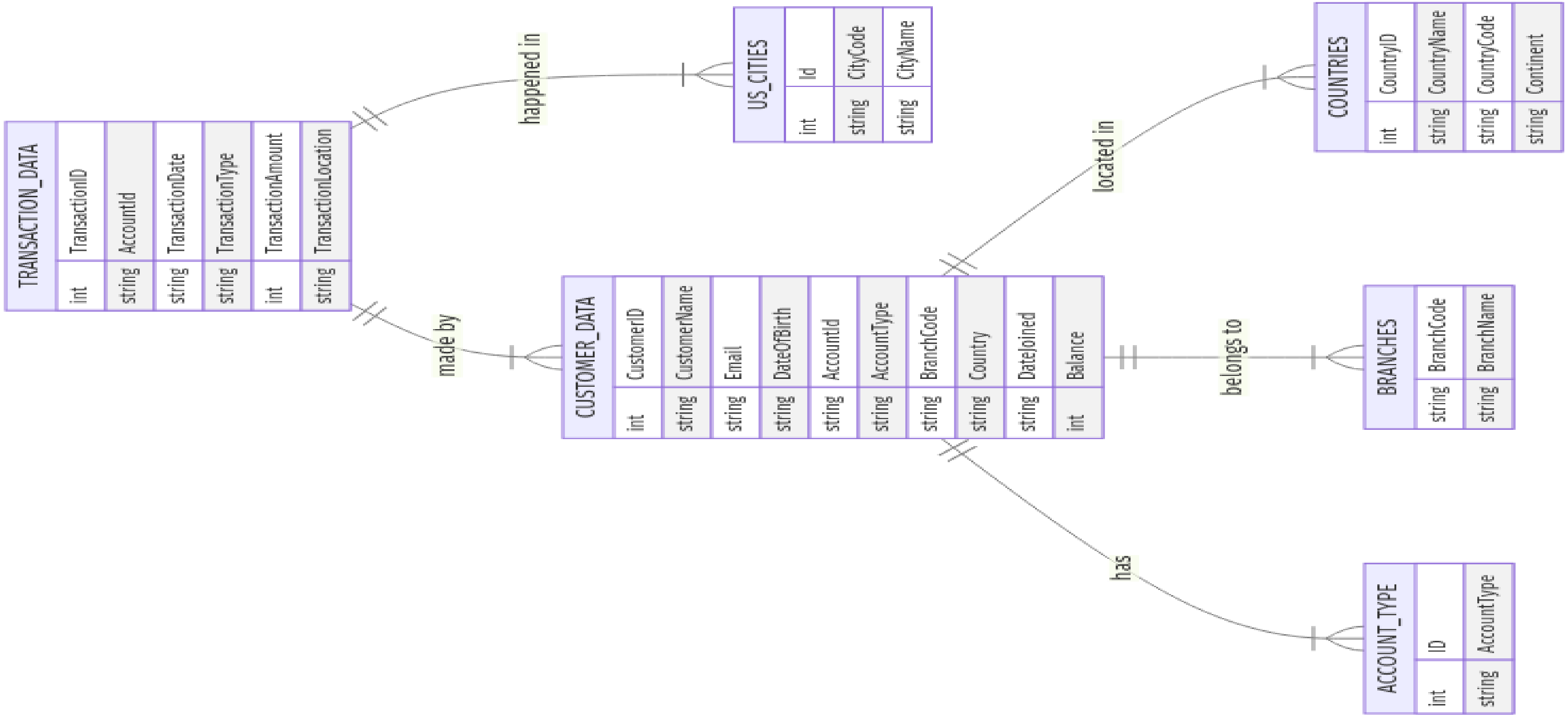
- Check for missing values and apply the same imputation techniques used previously (e.g., mean/median for numerical values, mode for categorical values).
- Remove duplicates using Excel’s **Remove Duplicates** tool.
- Identify outliers using statistical functions such as **STDEV, QUARTILE, or IQR method**.

Update Relational Database (MySQL)

- Append new transaction and customer data to the MySQL database.
- Validate data integrity using SQL queries (e.g., checking for null values or inconsistencies in primary keys).
- Optimize queries to improve performance and retrieval times.

Database design and setup

Entity relationship diagram (ERD) (1 of 3)



TRANSACTION_DATA	
int	TransactionID
string	AccountId
string	TransactionDate
string	TransactionType
int	TransactionAmount
string	TransactionLocation

US_CITIES	
int	Id
string	CityCode
string	CityName

CUSTOMER_DATA	
int	CustomerID
string	CustomerName
string	Email
string	DateOfBirth
string	AccountId
string	AccountType
string	BranchCode
string	Country
string	DateJoined
int	Balance

COUNTRIES	
int	CountryID
string	CountryName
string	CountryCode
string	Continent

BRANCHES	
string	BranchCode
string	BranchName

ACCOUNT_TYPE	
int	ID
string	AccountType

Entity relationship diagram (ERD) (2 of 3)

The finpro database appears to be designed for a financial institution, likely a **banking system** that manages customer accounts, transactions, branches, and locations.

ACCOUNT_TYPE (Stores different types of accounts)

- **ID** (int) → Unique identifier for the account type
- **AccountType** (varchar, Unique) → Describes the type of account (e.g., Savings, Checking)

Entity relationship diagram (ERD) (3 of 3)

Customer-Centric Approach:

- The **customer_data** table is at the center, linking **account type, branches, and countries**.
- This makes it easy to **track customer behavior** and financial history.

Transaction Tracking for Security & Compliance:

- Every **transaction** is logged with **amount, type, and location**.
- Helps in detecting **fraudulent activities** (e.g., sudden withdrawals in different cities).

Geographical Distribution:

- Countries and cities allow for **international operations**.
- Useful for **regional reporting and financial regulations compliance**.

Branch-Based Banking:

- Each customer is linked to a **branch**, ensuring **better customer service**.
- Helps in **branch-level revenue tracking and performance analysis**.

Data Integrity & Relationships:

- Foreign key constraints ensure **data consistency**.
- Unique constraints prevent **duplicate accounts, branches, and locations**.

Table structure design

ACCOUNT_TYPE (Stores different types of accounts)

Primary Key: AccountType

BRANCHES (Stores branch details)

Primary Key: BranchCode

COUNTRIES (Stores country details)

Primary Key: CountryName

CUSTOMER_DATA (Stores customer information)

Primary Key: AccountId

TRANSACTION_DATA (Stores transaction details)

Foreign Key: AccountId

OLAP schema (1 of 3)

The database follows a **Snowflake Schema** rather than a **Star Schema**.

- A **central fact table** connects to **normalized dimension tables**, which may be further broken down.
- **Dimension tables are normalized** to remove redundancy.
- **Advantage:** Saves storage space, avoids duplicate data, and ensures better data integrity.
- **Disadvantage:** Queries require more joins, making them slightly slower.

OLAP schema (2 of 3)

Transaction Data (Fact Table)

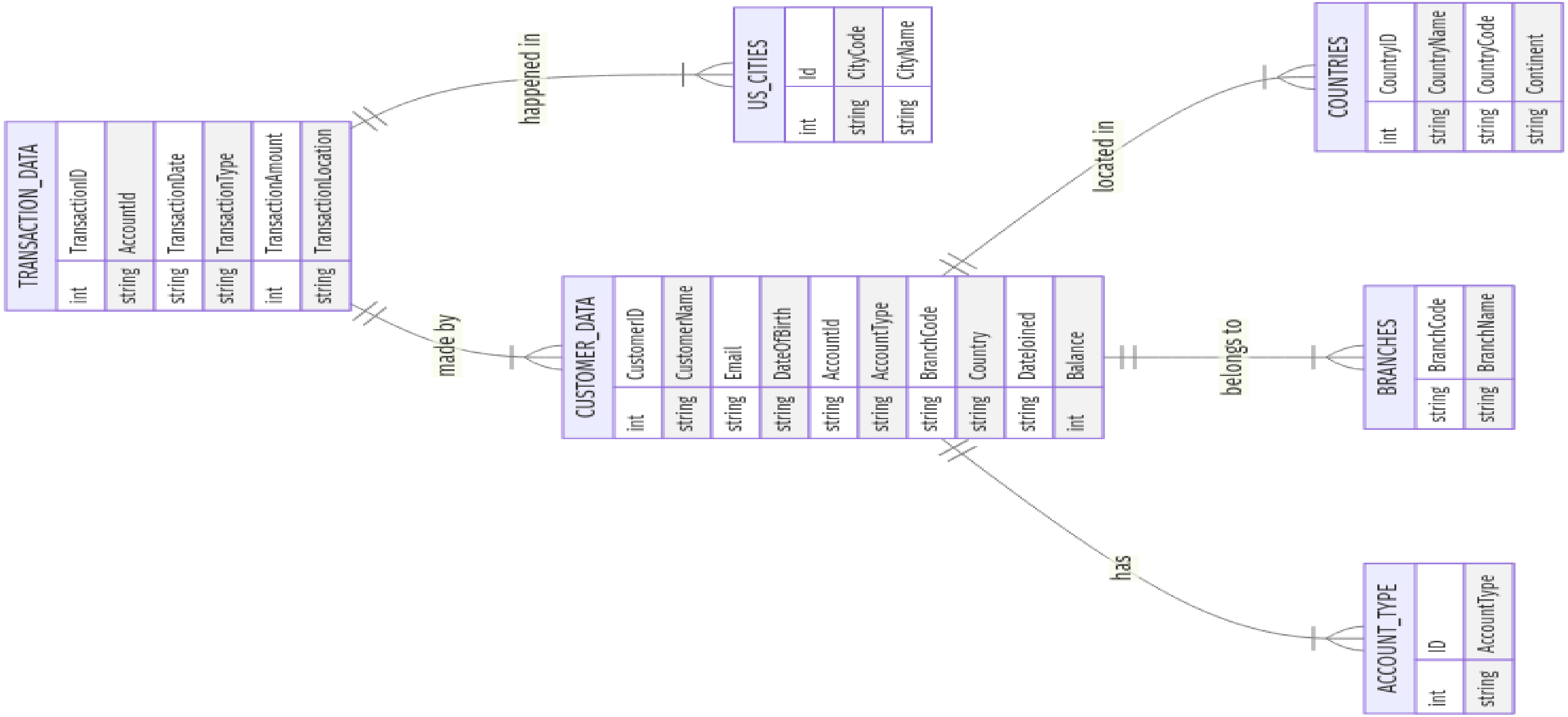
Role:

- This is the **central table** that records all financial transactions.
- Stores **measurable values** (e.g., transaction amount, type, and date).
- Connects to **multiple dimension tables** for detailed analysis.

Dimension Tables

- Customer Dimension
- Account Type Dimension
- Branch Dimension
- Geographical Dimension
- Transaction Location Dimension

OLAP schema (3 of 3)



Evaluation and interpretation

Documenting SQL queries and scripts (1 of 4)

- CREATE TABLE transaction_data (TransactionID INT AUTO_INCREMENT PRIMARY KEY,   AccountId VARCHAR(100),   TransactionDate DATETIME,   TransactionType VARCHAR(50),   TransactionAmount DECIMAL(10,2),   TransactionLocation VARCHAR(100),   FOREIGN KEY (AccountId) REFERENCES customer_data(AccountId),   FOREIGN KEY (TransactionLocation) REFERENCES us_cities(CityName));
- CREATE TABLE transaction_summary (MonthYear VARCHAR(7), -- Example: '2024-01'   AccountId VARCHAR(100),   TotalTransactions INT,   TotalAmount DECIMAL(12,2),   AverageTransaction DECIMAL(10,2),   PRIMARY KEY (MonthYear, AccountId));

Documenting SQL queries and scripts (2 of 4)

- INSERT INTO customer_data (CustomerName, Email, DateOfBirth, AccountId, AccountType, BranchCode, Country, DateJoined, Balance)VALUES ('John Doe', 'johndoe@email.com', '1990-05-20', 'ACC98765', 'Savings', 'BR001', 'USA', NOW(), 5000.00);

- UPDATE customer_data SET Balance = Balance - 300.00 WHERE AccountId = 'ACC98765';

Documenting SQL queries and scripts (3 of 4)

- SELECT DATE_FORMAT(TransactionDate, '%Y-%m') AS Month, AccountId,   SUM(TransactionAmount) AS TotalAmount, COUNT(TransactionID) AS TotalTransactionsFROM transaction_dataGROUP BY ROLLUP (Month, AccountId);

- SELECT a.AccountType,   c.Country,   SUM(t.TransactionAmount) AS TotalRevenueFROM transaction_data tJOIN customer_data c ON t.AccountId = c.AccountIdJOIN account_type a ON c.AccountType = a.AccountTypeGROUP BY CUBE (a.AccountType, c.Country);

Documenting SQL queries and scripts (4 of 4)

- SELECT AccountId, CustomerName, BalanceFROM customer_dataORDER BY Balance DESC;

- SELECT   c.CustomerName,   SUM(CASE WHEN t.TransactionType = 'Deposit' THEN 1 ELSE 0 END) AS TotalDeposits,   SUM(CASE WHEN t.TransactionType = 'Withdrawal' THEN 1 ELSE 0 END) AS TotalWithdrawalsFROM transaction_data tJOIN customer_data c ON t.AccountId = c.AccountIdGROUP BY c.CustomerNameORDER BY TotalDeposits DESC;

Inferences (1 of 2)

The **OLTP analysis queries** provide critical insights into **customer behavior, account balances, and transaction patterns**.

OLTP Analysis Query 1: Closing Balance Per Account

```
SELECT AccountId, CustomerName, Balance
FROM customer_data
ORDER BY Balance DESC;
```

Purpose:

- Retrieves the **current balance per account**.
- Identifies customers with **high or low balances** for **risk assessment and personalized offers**.

Inferences (2 of 2)

Monthly Transaction Summary

```
SELECT     DATE_FORMAT(TransactionDate, '%Y-%m') AS Month,
AccountId,    SUM(TransactionAmount) AS TotalAmount,
COUNT(TransactionID) AS TotalTransactionsFROM
transaction_dataGROUP BY ROLLUP (Month, AccountId);
```

Purpose:

- Summarizes **total transactions per account per month**.
- Uses **ROLLUP** to compute **subtotals at multiple levels** (monthly totals and individual customer totals).

Data integration and security

Data integration recommendations (1 of 3)

Customer Data and Transactions (Customer Data Sheet & Transaction Data Sheet)

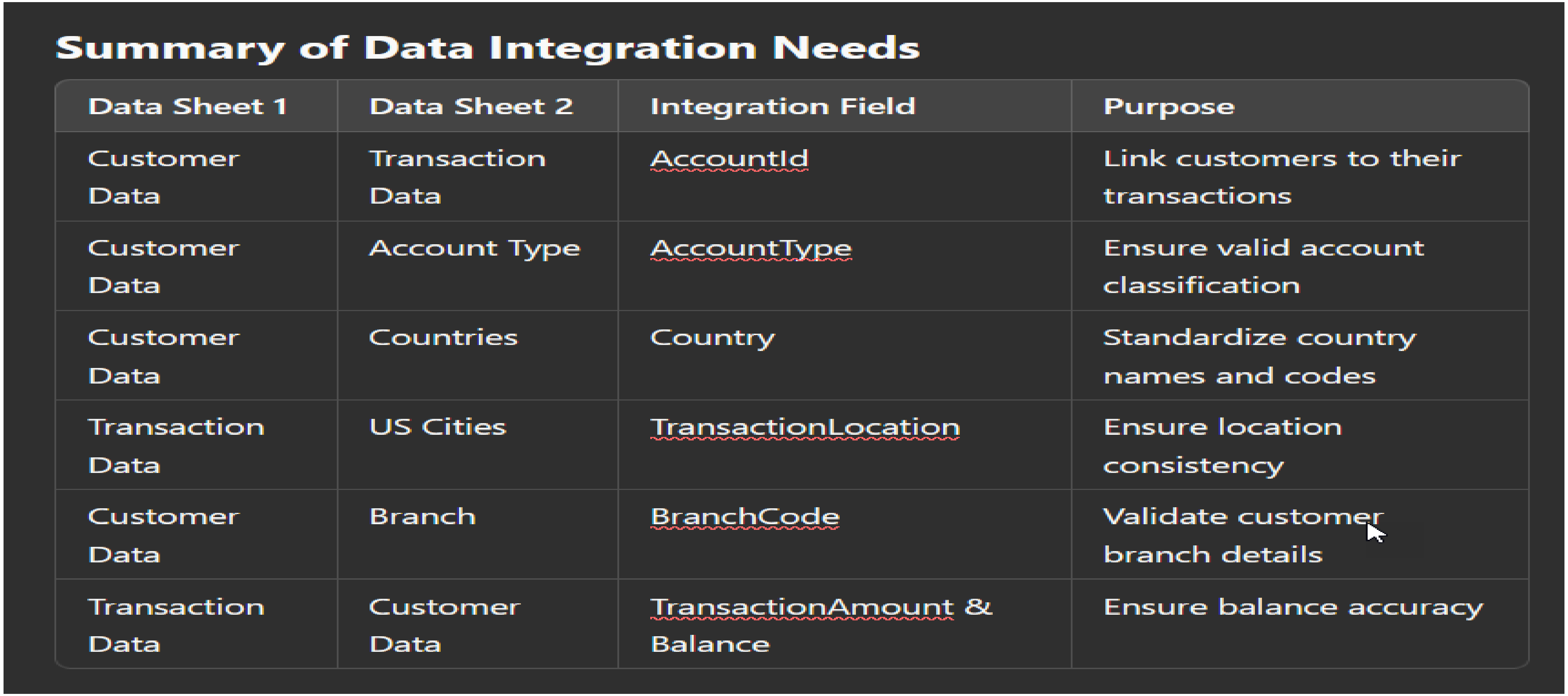
Integration Requirement:

- The **AccountId** field in the **Customer Data Sheet** links to the **Transaction Data Sheet** to track customer transactions.
- **TransactionDate** should align with **DateJoined** to prevent transactions before the customer joined.

Justification:

- Ensures accurate transaction history per customer.
- Validates transactions by linking them to active customers only.
- Helps in fraud detection by ensuring accounts aren't used before activation.

Data integration recommendations (2 of 3)



Summary of Data Integration Needs

Data Sheet 1	Data Sheet 2	Integration Field	Purpose
Customer Data	Transaction Data	AccountId	Link customers to their transactions
Customer Data	Account Type	AccountType	Ensure valid account classification
Customer Data	Countries	Country	Standardize country names and codes
Transaction Data	US Cities	TransactionLocation	Ensure location consistency
Customer Data	Branch	BranchCode	Validate customer branch details
Transaction Data	Customer Data	TransactionAmount & Balance	Ensure balance accuracy

Data integration recommendations (3 of 3)

ETL (Extract, Transform, Load)

Implementation Steps:

1. **Extract**: Pull data from multiple sources (Excel sheets, databases, APIs).
2. **Transform**:
    1. Standardize country names (e.g., "US" → "United States").
    2. Clean invalid emails and format dates correctly.
    3. Map transaction locations to standardized city codes.
3. **Load**: Insert the cleaned and structured data into the **Snowflake Schema** database.

Recommended Tools:

Informatica PowerCenter, **Talend**, **Apache NiFi**, or **AWS Glue** for ETL processing

Security documentation (1 of 3)

Role-Based Access Control (RBAC) is a security model that **restricts system access based on user roles** within an organization. Each role is assigned specific permissions that define **what actions can be performed** and **which data can be accessed**.

For FinPro Bank, the fin_manager **(Finance Manager) role** is responsible for **overseeing financial operations, transaction approvals, and financial reporting**. Given the sensitivity of financial data, RBAC ensures that the fin_manager can only access the information necessary for their responsibilities while **preventing unauthorized access to critical banking systems**.

Security documentation (2 of 3)

Best Practices for Column-Level Encryption

1. Identify Sensitive Data for Encryption
2. Choose an Encryption Method
3. Implement Column-Level Encryption Using AES-256
4. Apply Role-Based Restrictions to Encrypted Data
5. Audit & Monitor Encrypted Data Access

Security documentation (3 of 3)

Step-by-Step Procedure for Assigning Table-Specific Privileges

Step 1: Create the fin_manager role

Step 2: Assign the Role to Users

Step 3: Grant Read-Only Access to Financial Data

Step 4: Grant Transaction Approval Privileges

Step 5: Restrict Access to Personal Customer Data

Step 6: Prevent Unauthorized Modifications

Step 7: Enable Logging & Auditing

Step 8: Verify Privileges

Data reporting

Report specifications (1 of 6)

1. Customer Data Sheet

- **Description**: Contains details about individual bank customers, including personal information, account details, and demographic data.
- **Relevance**: Used to analyze customer demographics, account types, and banking behavior. It helps in customer segmentation, retention analysis, and fraud detection.

2. Transaction Data Sheet

- **Description**: Records all banking transactions, including deposits, withdrawals, and transfers, with details on transaction amounts, dates, and locations.
- **Relevance**: Essential for financial reporting, trend analysis, fraud detection, and identifying high-value transactions.

Report specifications (2 of 6)

3. Account Type Sheet

- **Description**: Provides a list of different types of bank accounts available, such as Savings and Checking accounts.
- **Relevance**: Used to classify and analyze customer accounts, track product popularity, and assess account-specific trends.

4. Countries Sheet

- **Description**: Lists countries where the bank operates, including country names, codes, and their respective continents.
- **Relevance**: Helps standardize country data for consistency in reporting, regional analysis, and compliance tracking.

Report specifications (3 of 6)

5. US Cities Sheet

- **Description**: Contains a list of US cities and their corresponding city codes used in bank records.
- **Relevance**: Supports regional transaction analysis, customer location tracking, and branch performance assessments within the US.

6. Branch Sheet

- **Description**: Contains details of bank branches, including their unique branch codes and names.
- **Relevance**: Used to analyze branch-specific transactions, performance, and customer distribution.

Report specifications (4 of 6)

a) Date Range Filter

b) Transaction Type Filter

Report specifications (5 of 6)

c) Account Type Filter

d) Branch Filter

Report specifications (6 of 6)

e) Country Filter

f) High-Value Transactions Filter

Performance analysis guidelines (1 of 3)

Comprehensive List of Metrics/KPIs for Performance Analysis

1. Financial Performance Metrics
2. Customer Experience & Retention Metrics
3. Operational Efficiency Metrics
4. Security & Compliance Metrics
5. Digital Banking & IT Performance Metrics
6. Risk Management & Fraud Prevention Metrics
7. Employee Performance & Workforce Metrics

Performance analysis guidelines (2 of 3)

Dashboard Design

- **Layout**: Multi-section design with financial, operational, customer, and security metrics.
- **Visuals**: Interactive charts, KPI scorecards, and trend analysis graphs.
- **Filters**: Date range, branch selection, account type, and transaction type.
- **Data Sources**: Transaction logs, customer data, financial reports, and security logs.

Key Insights Generated

- **Profitability Trends**: Identifies revenue drivers by analyzing **Net Interest Margin (NIM), ROA, and ROE**.
- **Customer Satisfaction & Retention**: Pinpoints retention risks via **NPS scores, complaint analysis, and churn rate**.

Performance analysis guidelines (3 of 3)

Maintaining and Updating FinPro Bank’s Performance Dashboard

Step 1: Refresh Data Regularly

Step 2: Add or Remove KPIs Based on Business Needs

Step 3: Optimize Dashboard Performance

Step 4: Ensure Data Accuracy & Security Compliance

General handover documentation

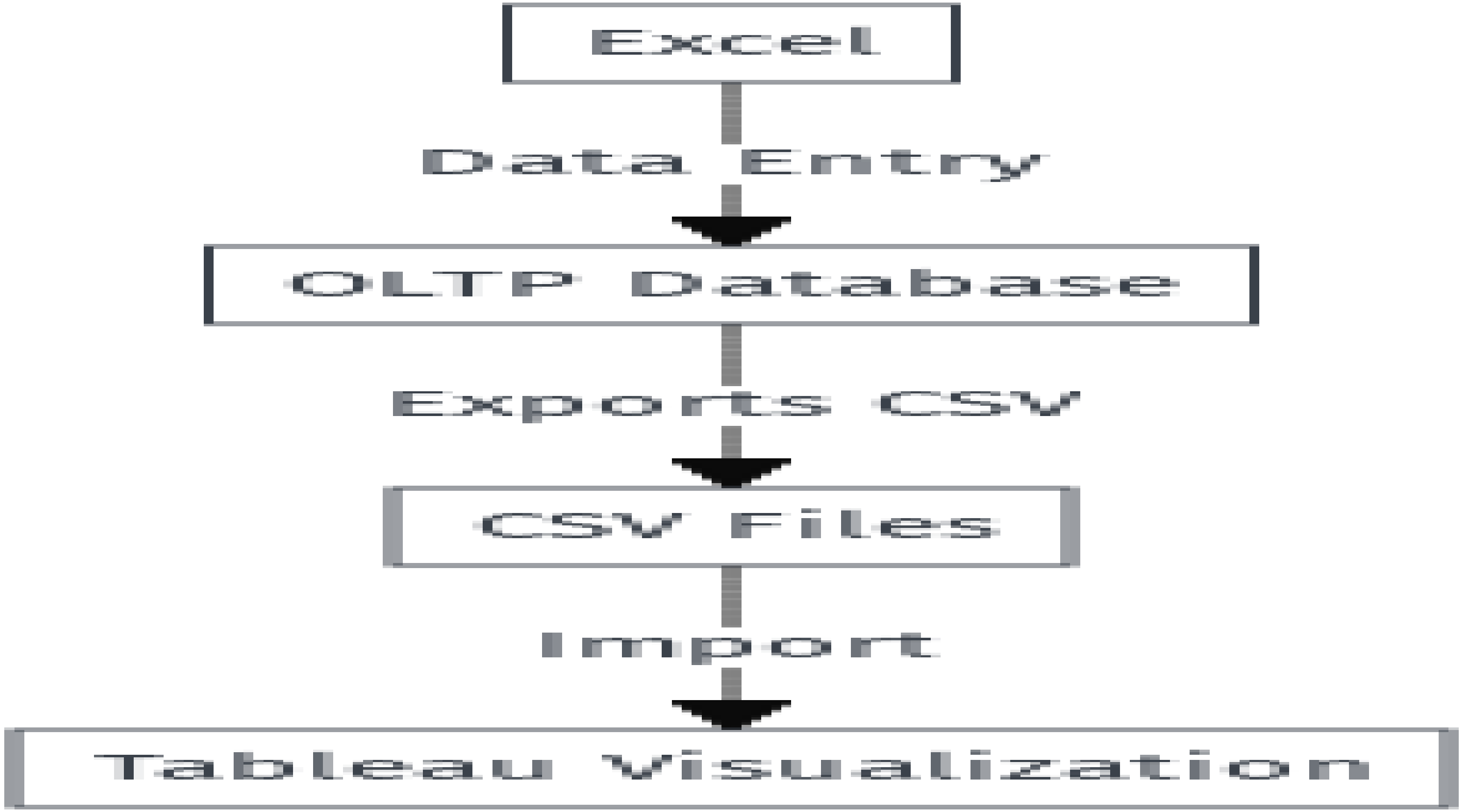
Role and key responsibilities

As a **Data Manager** at FinPro Bank, my primary role was to design, implement, and manage the bank's data architecture, ensuring seamless data integration, processing, and reporting. My responsibilities covered the entire data pipeline, from raw data acquisition to the creation of insightful business intelligence dashboards.

Key Responsibilities and Contributions

1. Data Acquisition and Cleaning
2. Database Design and Management
3. Data Warehousing and Optimization
4. Business Intelligence and Reporting
5. Security and Compliance
6. Handover Documentation and Training

System architecture diagram



Key challenges and resolutions

Data Inconsistencies Across Multiple Sources

Issue:

- Different data sources had inconsistent formats, such as varying country codes ("US" vs. "United States"), date formats, and missing account types.
- Transaction records contained incorrect or duplicate customer information.

Resolution:

Standardized data formats using **Excel functions** (e.g., **TEXT, FIND & REPLACE, and IFERROR**).
Used **SQL queries** in MySQL to clean and normalize data before storing it in the database.
Implemented **data validation rules** to enforce consistency during data entry.

Insights and recommendations

Insights and recommendations

1. Data Quality is the Foundation of Reliable Analytics

Insight:

- Poor data quality leads to incorrect analysis, misleading reports, and flawed decision-making. Issues such as missing values, inconsistent formats, and duplicates can significantly impact the bank's operations.

2. Optimized Data Processing Improves System Performance

Insight:

- Large volumes of transactional data can slow down queries and dashboards, leading to delays in generating insights for bank management.

3. Effective Documentation and Knowledge Sharing Ensure Long-Term Success

Insight:

- A well-documented system is crucial for future updates, troubleshooting, and onboarding new team members. Without proper documentation, resolving issues or making enhancements becomes time-consuming and prone to errors.

Summary

Appendix

Appendix

- Include any relevant additional charts, worksheets, or tables that you may have created during the analysis phase
- You can provide links to public tableau reports and dashboards you have prepared