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

- Handover the Data Management System
- Explain Data Architecture and workflows
- Present the documentations and activities performed in Data Architecture
- Provide recommendations for current data eco system
- Demonstrate data reporting capabilities
    - Showcase the data reporting tools
    - Highlight the key metrics and performance reporting used by management

Summary of work done on the project

Data Acquisition and cleaning

- Acquiring data from desperate data sources
- Cleaning and transform data using excel and SQL

Database Design and Setup

- Database configuration
- Data Modeling with ERD

Data governance and Quality control

- Implementation of RBAC
- Encryption and Decryption of sensitive data

Data Analysis

- Generating insights from the data

Data Reporting

- Automated reporting
- Dashboards development

Data acquisition and cleaning

Data acquisition documentation (1 of 1)

- Total of 6 data sheets with information relevant to bank operations
  - Customer Data Sheet (xlsx) - provides detailed information about individual customers, including their names, contact details, demographics, account details.
  - Transaction Data Sheet (xlsx) - records transactions of customers with transaction dates, types, amounts and locations.
  - Account Type (csv) - categorizes account types offered by the bank.
  - Countries (csv) - lists countries and their codes and continents.
  - US_Cities (csv) - consists a list of US cities and their corresponding code.
  - Branches (csv) - lists the branches with their code and names.

Data cleaning documentation (1 of 2)

- Customer data sheet
  - Removing the NULL values from balance field.
  - Create calculated field for Age of customer and define age group
  - Formatting date field as MM/DD/YYYY
  - Create field to categorize according to the balance
- Transactions data sheet
  - Create new field 'Transaction month' from transaction date
  - Standardize the location field.

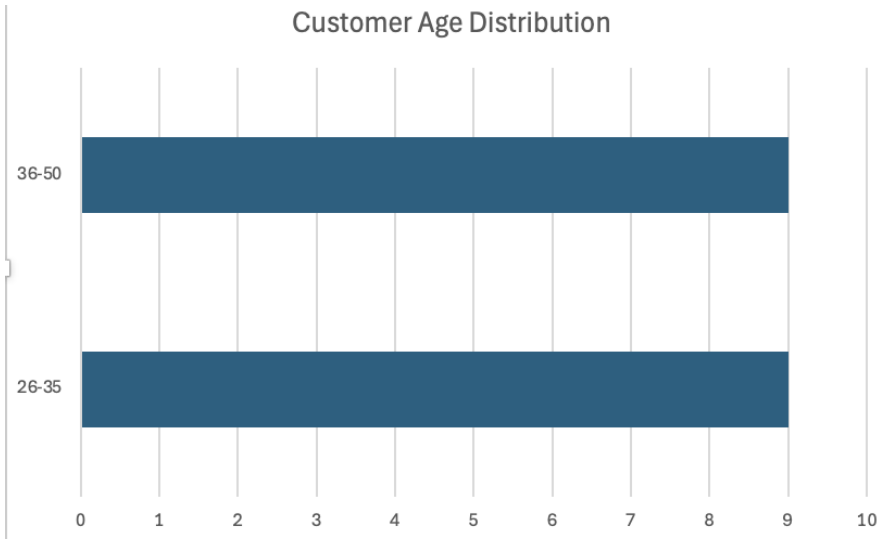
Data cleaning documentation (2 of 2)

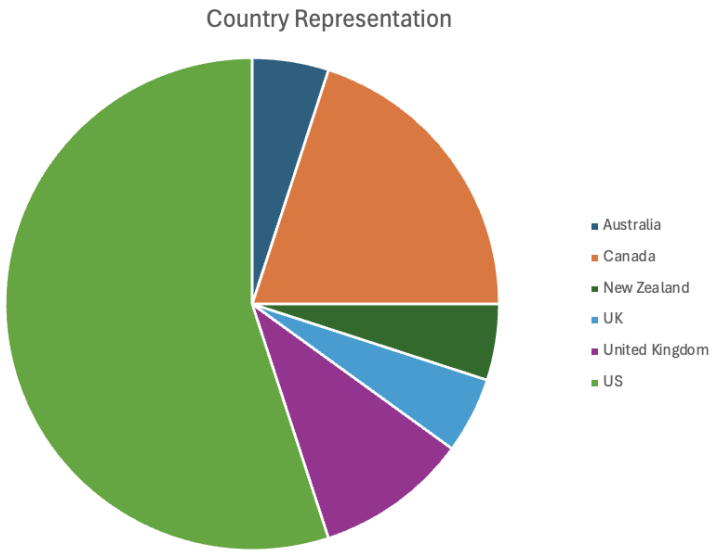
- Formulas and Macros used to perform data cleaning
  - =IFS(L2 >=10000, "High Balance", L2 >= 5000, "Medium Balance", L2 < 5000, "Low Balance")
  - =TEXT(C2,"MMMM")
  - =IFS(E3 <=25, "18-25", E3 <=35, "26-35", E3<=50, "36-50", E3>=51, "51+")
  - =AVERAGE(L2:L21)

Analysis and visualization using MS Excel (1 of 2)

Key metrics and KPI used in analysis

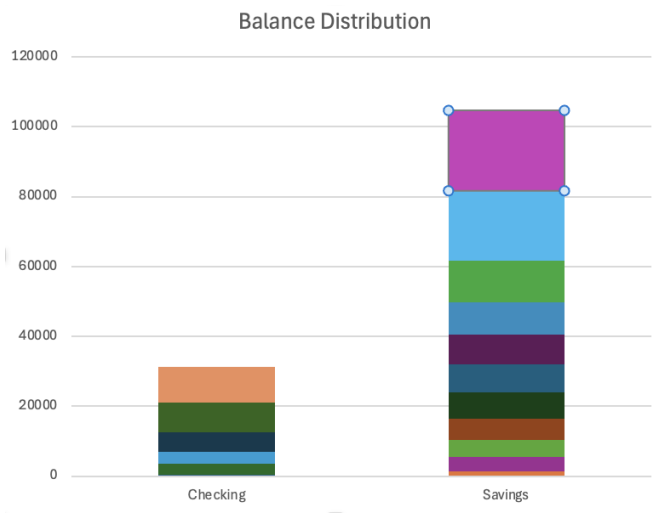
- **Country Representation:** shows the geographic distribution of customers and key metric to be used in finding concentrated customer base for marketing and expanding strategies.
- **Customer Age Distribution:** can help the bank understand the target audience and tailor services and products accordingly.

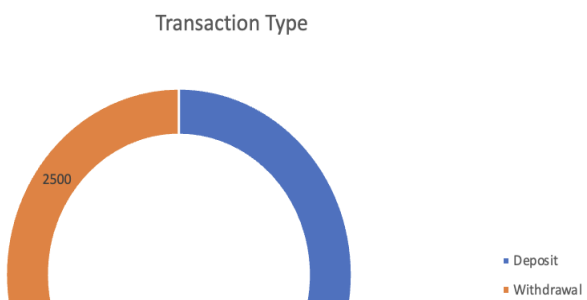


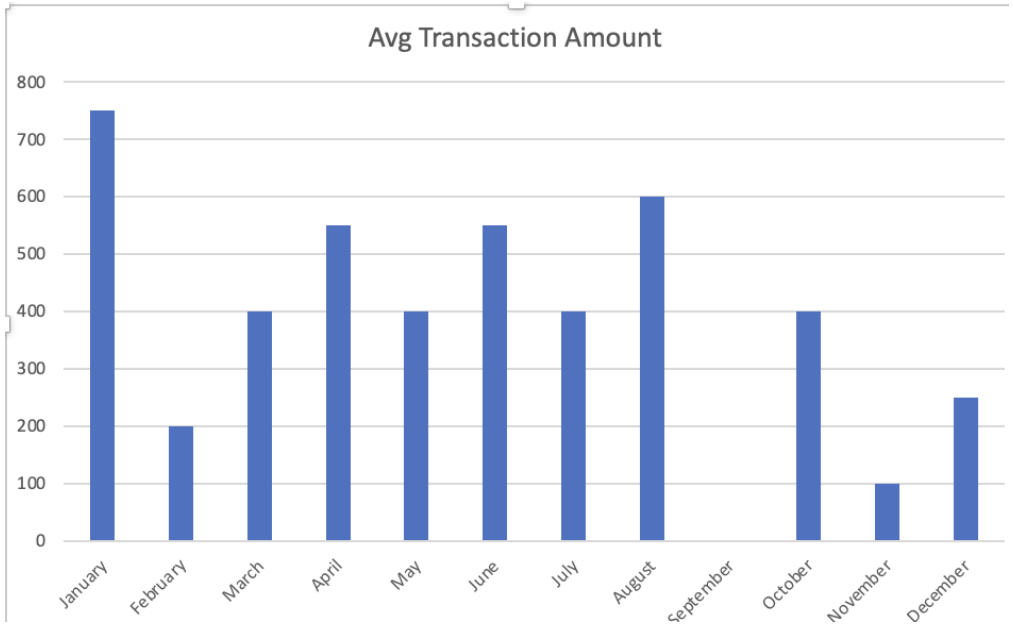


Analysis and visualization using MS Excel (2 of 2)

- **Balance Distribution:** helps in assessing the financial status of the customer base and can indicate which groups hold most of the bank's assets.
- **Monthly Transaction Trend:** assist in understanding periods of high or low activity and making data-driven decisions.
- **Transaction Type Representation:** to understand customer behavior and the most frequent types of transactions.







Database design and setup

Entity relationship diagram (ERD) (1 of 3)

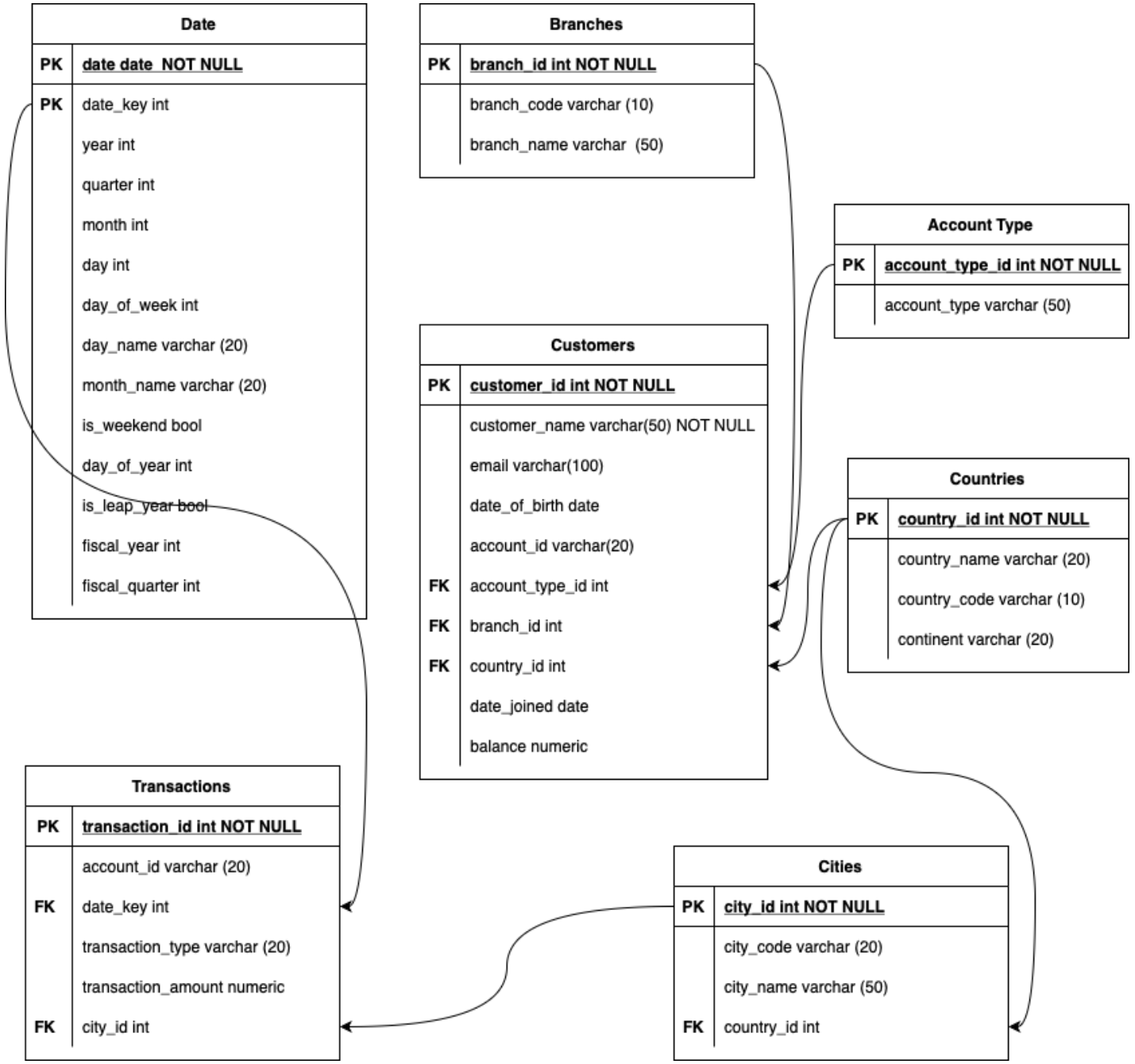
Overview of relationships between entities in database.

Six Dimension Tables

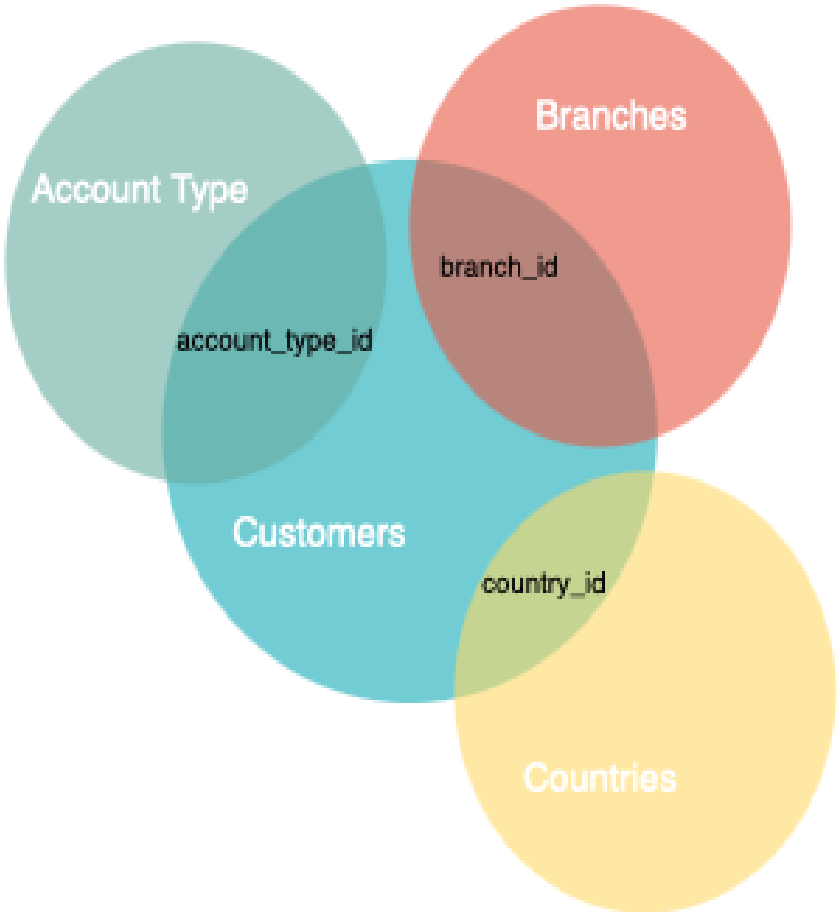
(Customers, Date , Branches, Cities, Countries, Account Type)

One Fact Table

(Transactions)



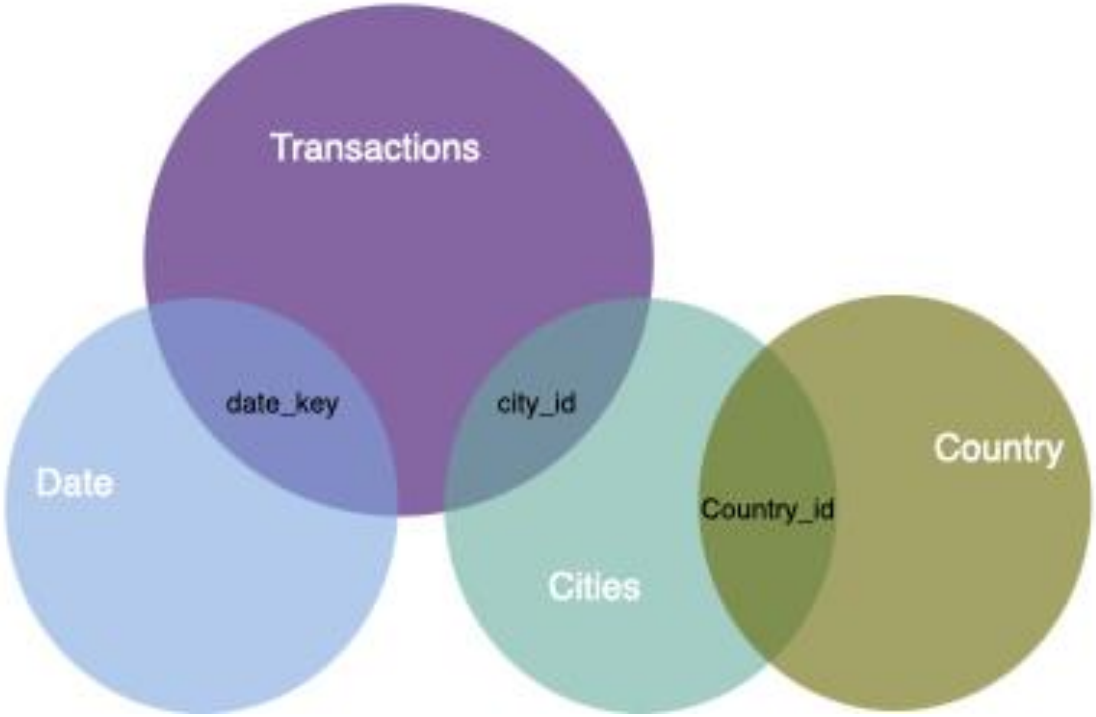
Entity relationship diagram (ERD) (2 of 3)



Relationships and Business Logic

- **Customers to Branches** - Connects customers to their associated bank branch
- **Customers to Countries** - Links customers to their country, supporting nationality-based analysis
- **Customers to Account Type** - Associates each customer with an account type, useful for product-specific insights

Entity relationship diagram (ERD) (3 of 3)



Relationships and Business Logic

- **Transactions to Date** - Enables time-based analysis of transactions
- **Transactions to Cities**- Tracks the transaction locations
- **Cities to Countries** - Defines which country each city belongs to, enabling geographic roll-ups.

Table structure design (1 of 5)

Customers Dimension Table

Column Name	Data Type	Constraint
customer_id	INT	Primary Key (PK), Not NULL, Auto Increment
customer_name	VARCHAR(50)	NOT NULL
Email	VARCHAR(100)	UNIQUE
date_of_birth	DATE	
account_id	VARCHAR(20)	NOT NULL
account_type_id	INT	Foreign Key (FK) → AccountType(account_type_id)
branch_id	INT	Foreign Key (FK) → Branches(branch_id)
country_id	INT	Foreign Key (FK) → Countries(country_id)
balance	NUMERIC	DEFAULT 0.00
date_joined	DATE	

Table structure design (2 of 5)

Branches Dimension Table

Column Name	Data Type	Constraint
branch_id	INT	Primary Key (PK), Not NULL
branch_code	VARCHAR(10)	
branch_name	VARCHAR(50)	

Account Type Dimension Table

Column Name	Data Type	Constraint
account_type_id	INT	Primary Key (PK), Not NULL
account_type	VARCHAR(50)	

Table structure design (3 of 5)

Cities Dimension Table

Column Name	Data Type	Constraint
ciry_id	INT	Primary Key (PK), Not NULL
city_code	VARCHAR(20)	
city_name	VARCHAR(50)	
country_id	INT	Foreign Key (FK) → Countries(country_id)

Countries Dimension Table

Column Name	Data Type	Constraint
country_id	INT	Primary Key (PK), Not NULL
country_name	VARCHAR(20)	
country_code	VARCHAR(10)	
continent	VARCHAR(20)	

Table structure design (4 of 5)

Date Dimension Table

Column Name	Data Type	Constraint
date	DATE	Primary Key (PK), Not NULL
date_key	INT	Primary Key (PK), Not NULL
year	INT	
quarter	INT	
month	INT	
day	INT	
day_of_week	INT	
day_name	VARCHAR(20)	
month_name	VARCHAR(20)	
is_weekend	BOOL	
day_of_year	INT	
is_leap_year	BOOL	
fiscal_year	INT	
fiscal_quarter	INT	

Table structure design (5 of 5)

Transactions Fact Table

Column Name	Data Type	Constraint
transaction_id	INT	Primary Key (PK), Not NULL, Auto Increment
account_id	VARCHAR(20)	NOT NULL
date_key	INT	Foreign Key (FK) → Date(date_key)
transaction_type	VARCHAR(20)	
transaction_amount	NUMERIC	
city_id	INT	Foreign Key (FK) → Cities (city_id)

OLAP schema (1 of 3)

Chosen Schema: Star Schema

The star schema has been chosen for the design of the data warehouse due to its simple and efficient structure for analytical queries.

- **Simplified Design**: The star schema is easier to understand and implement
- **Faster Queries**: The denormalized structure of the star schema excels at performing complex aggregations
- **Effective for Reporting:** It is suited for reporting and summarize data quickly.

OLAP schema (2 of 3)

Dimension and Fact Tables

Fact Table

- Transactions (attributes: transaction_id, account_id, date_key, transaction_type, transaction_amount, city_id)

Dimension Tables

- Customers (attributes: customer_id, name, email, date_of_birth, account_id, account_type_id, branch_id, country_id, balance, date_joined)
- Date (attributes: date,date_key, year, quarter, month, day, day_of_week, day_name, month_name, is_weekend, day_of_year, is_leap_year, fiscal_year, fiscal_quarter)
- Branches (attributes: branch_id, branch_code, branch_name)
- Cities (attributes: city_id, city_code, city_name, country_id)
- Countries (attributes: country_id, country_code, country_name, continent)
- Account Type (attributes: account_type_id, account_type)

OLAP schema (3 of 3)

ERD Relationships and reporting capabilities

Customers dimension, linked through account_id and country_id, allows for detailed reporting on customer demographics and behaviors

Date dimension's connection via date_key facilitates temporal analysis and enables reporting on daily, monthly, quarterly or yearly performance

Branch_id in Customers to Branches, the schema enables branch-specific reporting

Cities and Countries dimensions support location-based analysis

Account_type_id relationship show cases their contribution to revenue which can target promotions

Transactions fact table serves as the core analytical data source. With its quantitative data and relations with dimension tables allow aggregations and metrics

Dimension-to-dimension relationships add depth to cross-dimensions reporting

Evaluation and interpretation

Documenting SQL queries and scripts (1 of 8)

Categorization of SQL Scripts

- **OLTP Operations**: to handle large volume of transactions such as insert, update, delete and typically for day to day operational requirement.
  - OLTP SQL commands for table creation(CREATE), selection(SELECT), insertion data (INSERT), updates (UPDATE), and deletions (DELETE)

- **OLAP Operations**: to handle the analytical operations and reporting requirements. It supports complex queries and data analysis processes like aggregations, joins, and multi-dimensional analysis.
  - OLAP SQL commands for analytics involves AGGREGATE_FUNCTION with GROUP BY, JOIN, GROUPING SETS, ROLL UP, QUBE, MATERIALIZED QUERY TABLE (MQT)

Documenting SQL queries and scripts (2 of 8)

OLTP Operations

INSERT : inserting the new transactional data to the system.

```
INSERT INTO `transaction_data` VALUES
(201,'ACCT-10002345','01/05/2022','Deposit',500,'New York');
```

```
SELECT * FROM `transaction_data`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

TransactionID	AccountId	TransactionDate	TransactionType	TransactionAmount	TransactionLocation
201	ACCT-10002345	01/05/2022	Deposit	500	New York

UPDATE: update the existing data.

```
UPDATE `customer_data`
SET Balance= 2000 WHERE CustomerID= 102 ;
```

CustomerID	CustomerName	AccountId	AccountType	Country	Balance
102	Bob Smith	ACCT-10005678	Checking	United States	2000

Documenting SQL queries and scripts (3 of 8)

OLTP Operations

CREATE: Create database object such as table.

```
CREATE TABLE `account_type` (
  `ID` int DEFAULT NULL, `AccountType` varchar(50) DEFAULT NULL,
  UNIQUE KEY `AccountType_UNIQUE` (`AccountType`)
);
```

			ID	AccountType
Edit	Copy	Delete	1	Savings
Edit	Copy	Delete	2	Current
Edit	Copy	Delete	3	Overdraft
Edit	Copy	Delete	4	Checking

Delete: delete the existing data from table.

```
DELETE FROM `transaction_data` WHERE TransactionID= 201;
```

TransactionID	AccountId	TransactionDate	TransactionType	TransactionAmount	T
202	ACCT-10005678	02/15/2022	Withdrawal	200	N
203	ACCT-10007891	03/15/2022	Deposit	300	L

Documenting SQL queries and scripts (4 of 8)

OLAP Operations

AGGREGATION (SUM, AVG, COUNT, MIN, MAX)

Purpose: Summarizes data using aggregate functions

GROUPING SETS

Purpose: Computes multiple grouping combinations in a single query.

Query | Query History

```sql
SELECT customerid, accountid, SUM(balance) FROM finprodimcustomers
GROUP BY
Grouping sets (
(customerid, accountid),
(customerid),
(accountid),
()
);
```

Data Output | Messages | Notifications

	customerid integer	accountid text	sum bigint
1	[null]	[null]	134400
2	116	ACCT-10005478	-500
3	120	ACCT-10003489	9200
4	108	ACCT-10006789	3500
5	101	ACCT-10002345	12000
6	105	ACCT-10003456	-1000
7	114	ACCT-10006543	10000
8	119	ACCT-10007621	500
9	109	ACCT-10009876	8000
10	102	ACCT-10005678	3000
11	103	ACCT-10007891	8500

Documenting SQL queries and scripts (5 of 8)

OLAP Operations

JOINs (INNER JOIN, LEFT JOIN, etc.)

Purpose: Combines data from multiple tables.

ROLLUP

Purpose: Hierarchical aggregation (e.g., sub-totals and grand totals)

CUBE

Purpose: Computes all possible aggregations for specified columns

Query | Query History

```sql
SELECT dc.branchcode, dc.accounttype, SUM(ft.transactionamount) AS total_transaction
FROM finprofacttransactions ft
JOIN finprodimcustomers dc ON ft.accountid = dc.accountid
GROUP BY ROLLUP(dc.branchcode, dc.accounttype);
```

Data Output | Messages | Notifications

	branchcode text	accounttype text	total_transaction bigint
1	[null]	[null]	7200
2	BR002	Checking	200
3	BR019	Checking	350
4	BR016	Savings	500
5	BR015	Savings	-200
6	BR001	Savings	500
7	BR013	Savings	250
8	BR009	Savings	600
9	BR003	Savings	300
10	BR004	Savings	-400
11	BR014	Checking	1000

Documenting SQL queries and scripts (6 of 8)

OLAP Operations

Materialized Query Table (MQT)

Purpose: Stores precomputed query results for fast retrieval.

Query Query History

```sql
CREATE MATERIALIZED VIEW Total_transaction_per_branch (branchcode, total_transaction)
AS SELECT dc.branchcode, SUM(ft.transactionamount)
FROM finprofacttransactions ft
JOIN finprodimcustomers dc ON ft.accountid = dc.accountid
GROUP BY dc.branchcode;
```

Data Output Messages Notifications

```
SELECT 20

Query returned successfully in 560 msec.
```

Documenting SQL queries and scripts (7 of 8)

OLTP – Opening & Closing Balance

This query calculates the closing balance per account based on the sum of transactions.

Analysis:

- Opening balance is the account's initial balance
- Total TransactionAmount is deducted from balance
- The WHERE cd.AccountId = td.AccountId ensures that only accounts with transactions are included
- GROUP BY cd.AccountId aggregates data per account

```
select cd.AccountId, BranchCode, balance openingBalance, balance -
SUM(TransactionAmount) closingBalance from customer_data cd, transaction_data td
where cd.AccountId = td.AccountId group by cd.AccountId;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

←T→	AccountId	BranchCode	openingBalance	closingBalance
Edit Copy Delete	ACCT-10005678	BR002	3000	2800
Edit Copy Delete	ACCT-10007891	BR003	8500	8200
Edit Copy Delete	ACCT-10001234	BR004	5000	5400
Edit Copy Delete	ACCT-10003456	BR005	-1000	-1450
Edit Copy Delete	ACCT-10004567	BR006	7500	7400
Edit Copy Delete	ACCT-10008901	BR007	20000	19250
Edit Copy Delete	ACCT-10006789	BR008	3500	3300
Edit Copy Delete	ACCT-10009876	BR009	8000	7400
Edit Copy Delete	ACCT-10005432	BR010	0	0
Edit Copy Delete	ACCT-10007654	BR011	6000	5600
Edit Copy Delete	ACCT-10004321	BR012	5500	5400
Edit Copy Delete	ACCT-10002134	BR013	4000	3750
Edit Copy Delete	ACCT-10006543	BR014	10000	9000
Edit Copy Delete	ACCT-10003210	BR015	1500	1700
Edit Copy Delete	ACCT-10005478	BR016	-500	-1000
Edit Copy Delete	ACCT-10008765	BR017	23000	22200
Edit Copy Delete	ACCT-10002301	BR018	8700	8000
Console it Copy Delete	ACCT-10007621	BR019	500	150

Documenting SQL queries and scripts (8 of 8)

OLTP – Deposit & Withdrawal Count

This query counts the number of deposit and withdrawal transactions per customer.

Analysis:

- The CASE statement classifies transaction types
- The SUM() function aggregates these counts per customer
- WHERE cd.AccountId = td.AccountId joins both tables based on AccountId.
- GROUP BY cd.customerid ensures results are aggregated per customer.

Server: labs-mysql-little-mammoth-nightly.3306 » Database: finpro » Table: customer_data

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

```
select cd.customerid, sum(case when td.transactiontype = 'Deposit' then 1 else 0 end) Deposit_Count, sum(case when td.transactiontype
= 'Withdrawal' then 1 else 0 end) Withdrawal_Count from customer_data cd, transaction_data td where cd.AccountId = td.AccountId group
by cd.customerid;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

←T→	customerid	Deposit_Count	Withdrawal_Count
Edit Copy Delete	101	1	0
Edit Copy Delete	102	0	1
Edit Copy Delete	103	1	0
Edit Copy Delete	104	0	1
Edit Copy Delete	105	1	0
Edit Copy Delete	106	0	1
Edit Copy Delete	107	1	0
Edit Copy Delete	108	0	1
Edit Copy Delete	109	1	0
Edit Copy Delete	110	0	1
Edit Copy Delete	111	1	0
Edit Copy Delete	112	0	1
Edit Copy Delete	113	1	0
Edit Copy Delete	114	1	0
Edit Copy Delete	115	0	1
Edit Copy Delete	116	1	0
Edit Copy Delete	117	1	0
Edit Copy Delete	118	0	1

Console

Inferences (1 of 2)

OLTP analysis provides real-time financial insights

- Potential fraud detection: If a customer has an unusually high number of small deposits followed by withdrawals, it may indicate money laundering or suspicious activity
- Identifying inactive accounts for reactivation strategies
- Detect high volume deposit or withdrawal customers for targeted financial products
- Tracking deposit and withdrawal patterns for risk assessment
- OLTP analysis helps in fraud detection, overdraft prevention, and activity monitoring
- Tracking deposit and withdrawal patterns for risk assessment

Inferences (2 of 2)

OLAP Analysis provides strategic insights for long-term business

- Aggregated insights: using multidimensional data and analyze trends over time
- Forecasting, trend analysis, and customer segmentation
- Focus more on historical trends across different dimensions such as region, product type, or customer category
- Identifying seasonal transaction trends (e.g., increased withdrawals during holidays)
- Branche performance according to region

Data integration and security

Data integration recommendations (1 of 2)

Areas Requiring Data Integration & Their Importance

- Data Acquisition from different data sources
  - Collecting data from different banking systems and unifying the data will help to improve the data quality and business insight. Unified data helps reduce data silos and streamline data processing.

- Transaction processing
  - Reat time transaction data processing from mutiple touchpoints (ATM, Branch, mobile) can help the bank to track transactions effectively, provide customer with real-time financial services and better monitor fraudulent transactions.

- Regulatory Compliance
  - Data encryption and RBAC (Role Based Access Control) ensures the compliance with regulations such as **GDPR** and **PCI DSS,** safeguarding customer trust.

Data integration recommendations (2 of 2)

Recommended Data Integration Patterns

ETL (Extract, Transform, Load) for Historical Data Processing

- Batch processing of transaction history, customer profiles, and compliance reports, ensuring consistency, accuracy and data quality for reporting and compliance.

Event-Driven Architecture

- Event trigger data integration enables Fraud detection, instant notifications, and transaction monitoring ensuring real-time risk management and customer alerts.

API Integration

- API integration supports data accuracy and security ensuring seamless interoperability.

Security documentation (1 of 3)

Role-Based Access Control (RBAC) for fin_manager

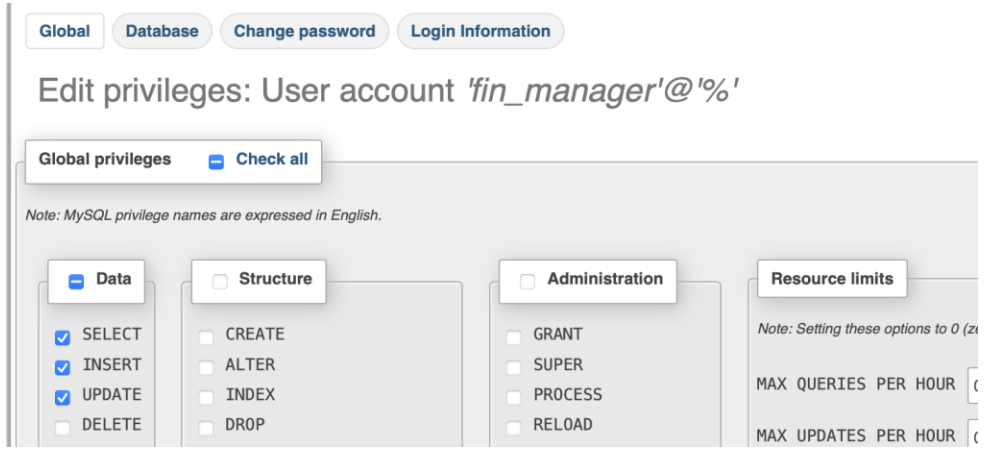
Overview of RBAC

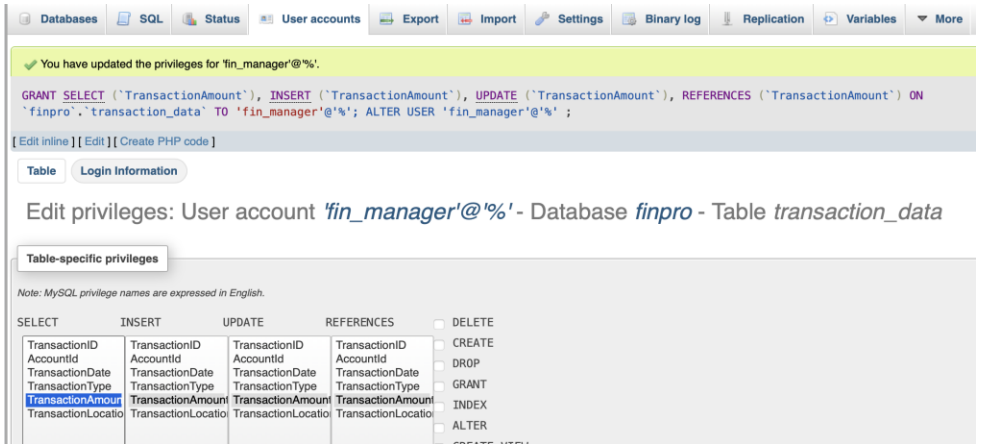
RBAC is a security model that restricts access to data and system resources based on user roles. The least privilege is given to the user who will be assigned role according to operational requirements.

The fin_manager role is intended for financial managers who need access to transaction data but should have restricted access to unauthorized changes or exposure of sensitive customer information.

Access Permissions for fin_manager

- Allows reading (SELECT), adding new transactions (INSERT), and modifying transaction records (UPDATE) on most tables except customer-sensitive tables.
- Allows update only the TransactionAmount column in the transaction_data table.
- Does not include DELETE, preventing accidental or unauthorized data removal.





Security documentation (2 of 3)

Guidelines for Implementing Column-Level Encryption

- Determine the columns contain sensitive information that requires encryption.
- Use AES symmetric encryption algorithm with a secure passphrase (key)
- To secure the encryption key, hash the passphrase using SHA-2 (512-bit)
- Convert the column data type to VARBINARY to store encrypted values
- Use AES_DECRYPT and CAST to retrieve original values
- Only authorized user will have access to the passphrase in order to decode the data.

```
rows in set (0.00 sec)

sql>    SELECT * FROM customer_data  LIMIT 5;
-----------+---------------+-------------------------+-------------+-------------------------
-----+-------------+------------+---------------+------------+---------+
CustomerID | CustomerName  | Email                   | DateOfBirth | AccountId
     | AccountType | BranchCode | Country       | DateJoined | Balance |
-----------+---------------+-------------------------+-------------+-------------------------
-----+-------------+------------+---------------+------------+---------+
       101 | Alice Johnson | alice.j@example.com     | 5/21/1990   | 0xB1DAD7C709CA614A88FAA7F
FE11 | Savings     | BR001      | United States | 1/15/2020  |   12000 |
       102 | Bob Smith     | bob.smith@xyz.com       | 8/15/1985   | 0x77D495F1E677CC7EA2B5A9B
31AB | Checking    | BR002      | United States | 11/30/2019 |    3000 |
       103 | Cathy Davis   | cathy.davis@example.com | 11/3/1992   | 0x2A9FD7C1B8386E0CE8A1D9D
2C0A | Savings     | BR003      | United States | 6/1/2021   |    8500 |
       104 | David Lee     | david.l@xyz.com         | 2/28/1978   | 0xBCB8483D8B8BA8A24295FCA
CF6E | Savings     | BR004      | United States | 3/20/2021  |    5000 |
       105 | Eva Turner    | No Email Provided       | 4/7/1989    | 0xFB8720C1D60DF1690EC7EB2
EB55 | Checking    | BR005      | United States | 7/19/2020  |   -1000 |
-----------+---------------+-------------------------+-------------+-------------------------
-----+-------------+------------+---------------+------------+---------+
rows in set (0.01 sec)

sql>
```

```
+-----------------------------------------------------+
| cast(AES_DECRYPT(AccountId , @key_str) as char(255)) |
+-----------------------------------------------------+
| ACCT-10009876                                       |
| ACCT-10007891                                       |
| ACCT-10002134                                       |
| ACCT-10006789                                       |
| ACCT-10008765                                       |
| ACCT-10003489                                       |
| ACCT-10005432                                       |
| ACCT-10007621                                       |
| ACCT-10004567                                       |
| ACCT-10005678                                       |
| ACCT-10004321                                       |
| ACCT-10008901                                       |
| ACCT-10007654                                       |
| ACCT-10005478                                       |
| ACCT-10002345                                       |
| ACCT-10006543                                       |
| ACCT-10001234                                       |
| ACCT-10002301                                       |
| ACCT-10003210                                       |
| ACCT-10003456                                       |
+-----------------------------------------------------+
20 rows in set (0.00 sec)
```

Tool iframe

Security documentation (3 of 3)

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

Checking privileges for the role.

SHOW GRANTS FOR fin_manager;

Granting privileges for the role

GRANT SELECT, INSERT ON finpro.transaction_data TO fin_manager;

Granting table-specific privileges for the role

GRANT UPDATE (TransactionAmount) ON finpro.transaction_data TO fin_manager;

Revoking privilege for the role

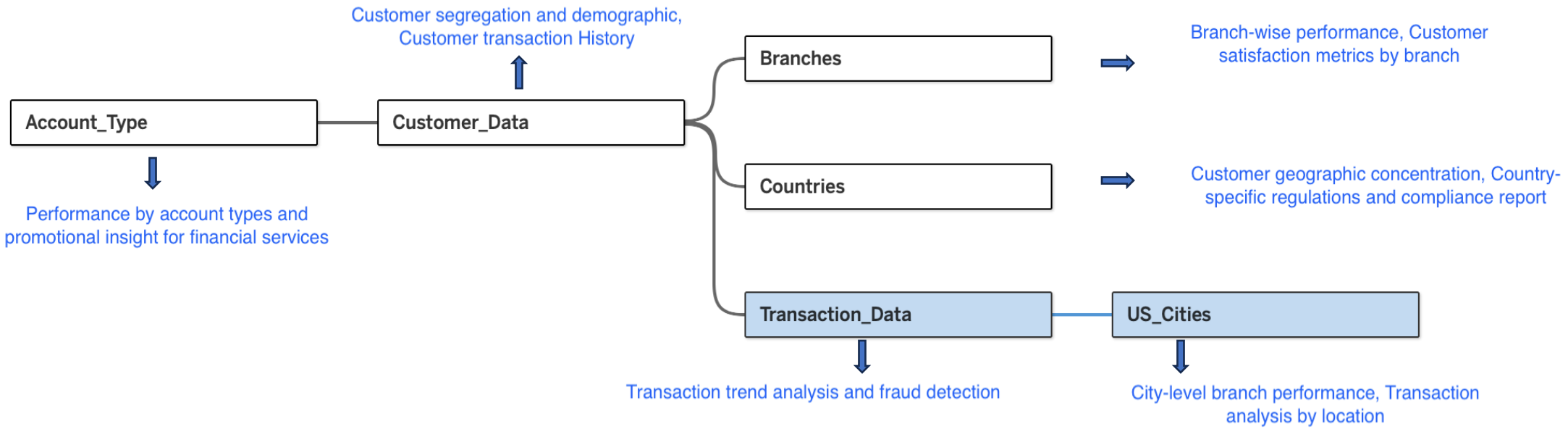
REVOKE DELETE ON finpro.transaction_data FROM fin_manager;

Data reporting

Report specifications (1 of 6)

Descriptions of each source and its relevance to the report

- Customer Data Sheet (xlsx) , Transaction Data Sheet (xlsx), Account Type (csv), Countries (csv), US_Cities (csv), Branches (csv)



Report specifications (2 of 6)

Key Filters and Calculated Fields for Reporting

Calculated Fields

- Closing Balance Calculated field is used for trend analysis, financial forecasting

  Formula: Closing Balance=Opening Balance+Deposits−Withdrawals

- Average Age of Customer helps in customer segmentation for marketing and detecting target age groups for financial products.

Key Filters (For Data Drill-Down & Segmentation)

- Closing Balance (Filter by Value) identify customer with high balance for enhanced services and low balances for potential retention strategies.
- Customer by Country filters customers based on their residence showcases concentration of customer base and enables regulatory reporting.
- Transaction Count filters on the number of transactions identifies activity of customers for operational requirement. It helps in fraud detection (unusually high transaction volume).

Report specifications (3 of 6)

Customer-wise closing balance report

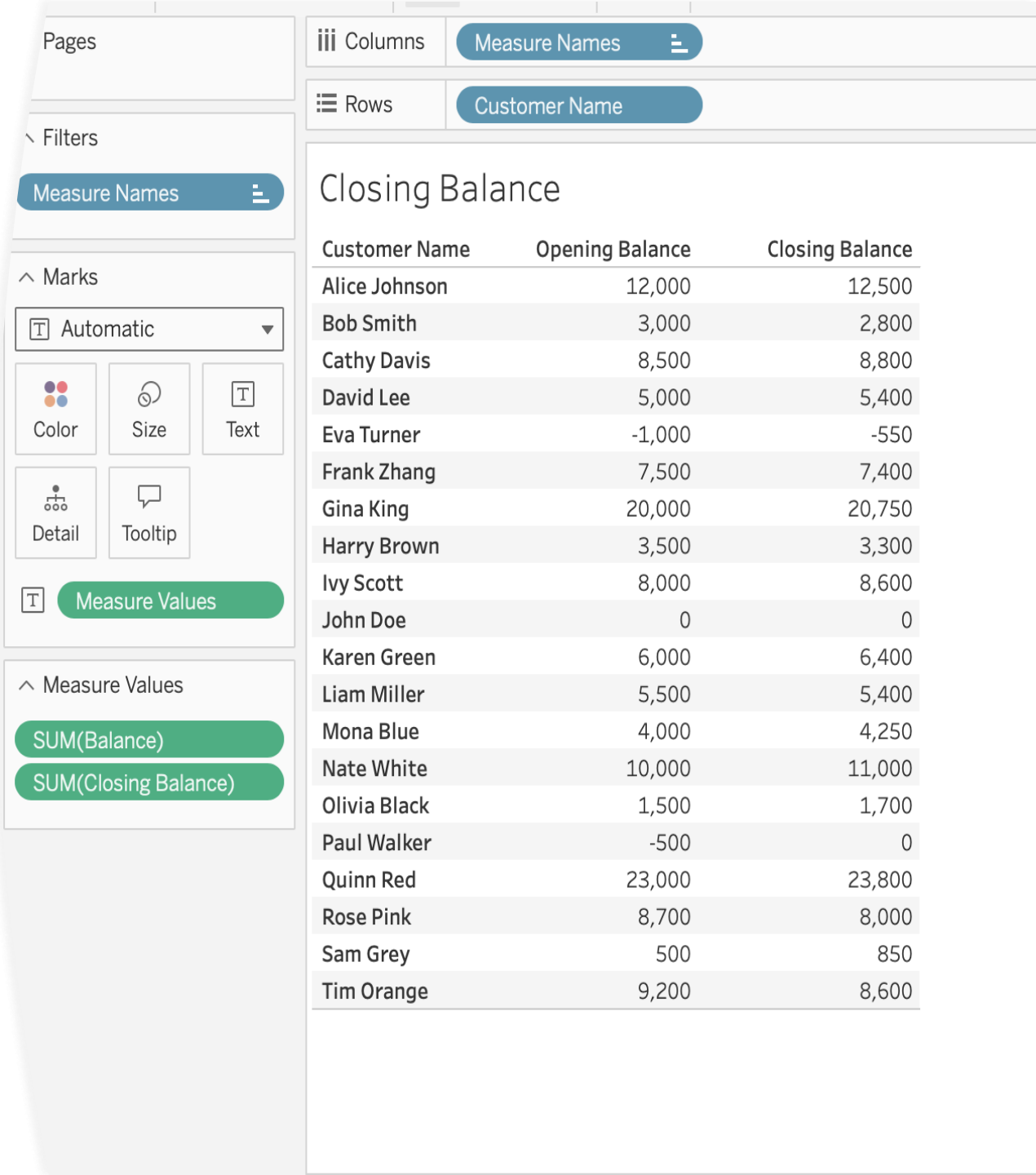
Creating the report

- Data Used: Customer name, Balance as opening balance and calculated field of closing balance based on the balance against transaction amount.
- Filters & Measures : Names, Balance, Closing Balance

Purpose of the report and its relevance

Report showcases insights into each customer's account balances over time.

- It is crucial for Financial Forecasting
- Customer Segmentation – Identifies high-balance customers for premium services and low-balance customers for retention efforts
- Fraud Monitoring – detects transaction patterns



Closing Balance

Customer Name	Opening Balance	Closing Balance
Alice Johnson	12,000	12,500
Bob Smith	3,000	2,800
Cathy Davis	8,500	8,800
David Lee	5,000	5,400
Eva Turner	-1,000	-550
Frank Zhang	7,500	7,400
Gina King	20,000	20,750
Harry Brown	3,500	3,300
Ivy Scott	8,000	8,600
John Doe	0	0
Karen Green	6,000	6,400
Liam Miller	5,500	5,400
Mona Blue	4,000	4,250
Nate White	10,000	11,000
Olivia Black	1,500	1,700
Paul Walker	-500	0
Quinn Red	23,000	23,800
Rose Pink	8,700	8,000
Sam Grey	500	850
Tim Orange	9,200	8,600

Report specifications (4 of 6)

Branch-wise closing balance report

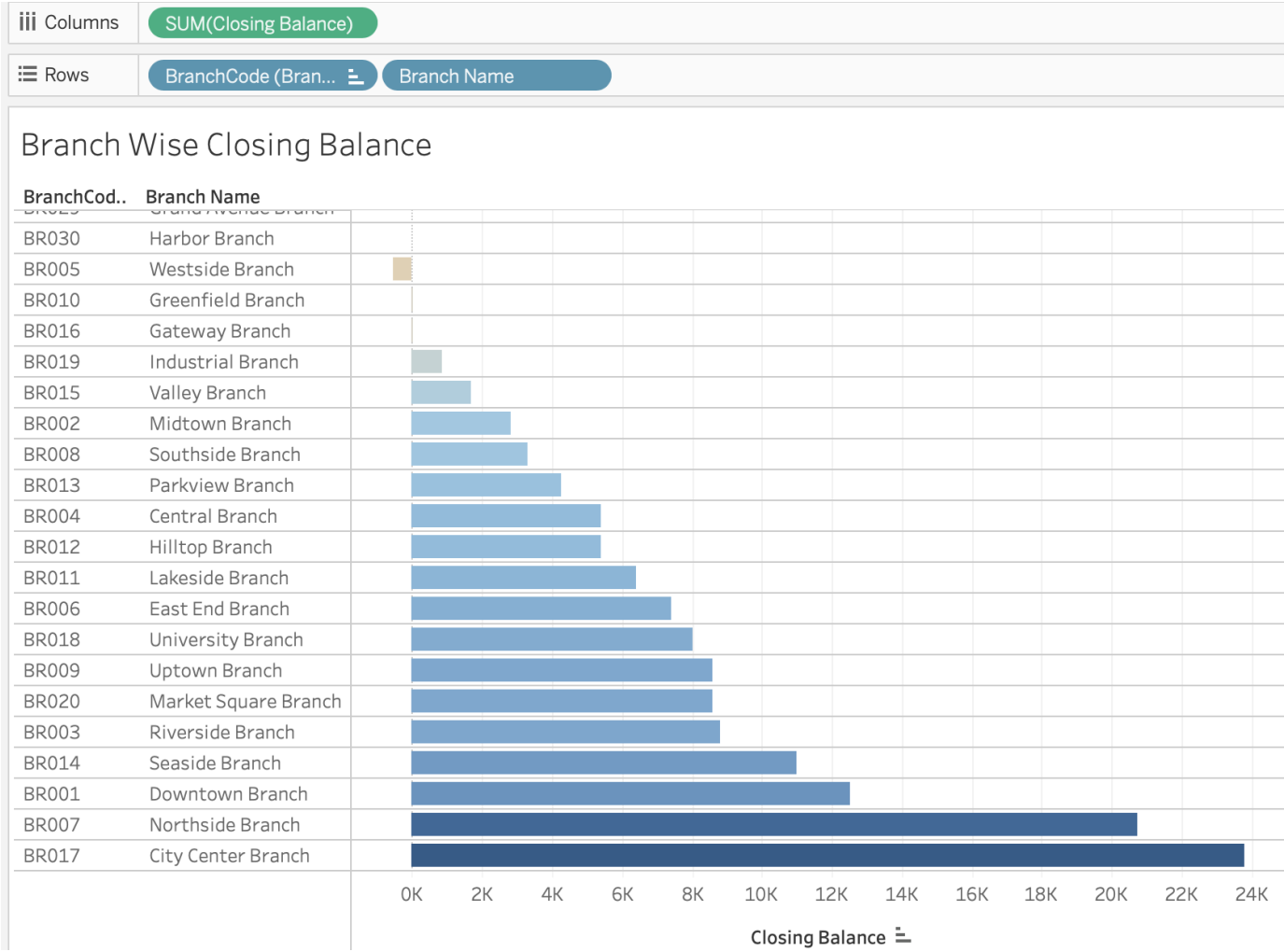
Creating the report

- Data Used: Branch code, Branch Name, Closing balance
- Filters & Measures : Branch Names, Closing Balance

Purpose of the report and its relevance

Report provides balances across different bank branches.

- Branch Performance Analysis – Identify performing branches
- Financial Planning & Resource Allocation – Helps optimize staffing, and operational strategies
- Customer Segmentation by Location – Supports targeted marketing and customer service strategies.



Report specifications (5 of 6)

Country-wise transaction count report

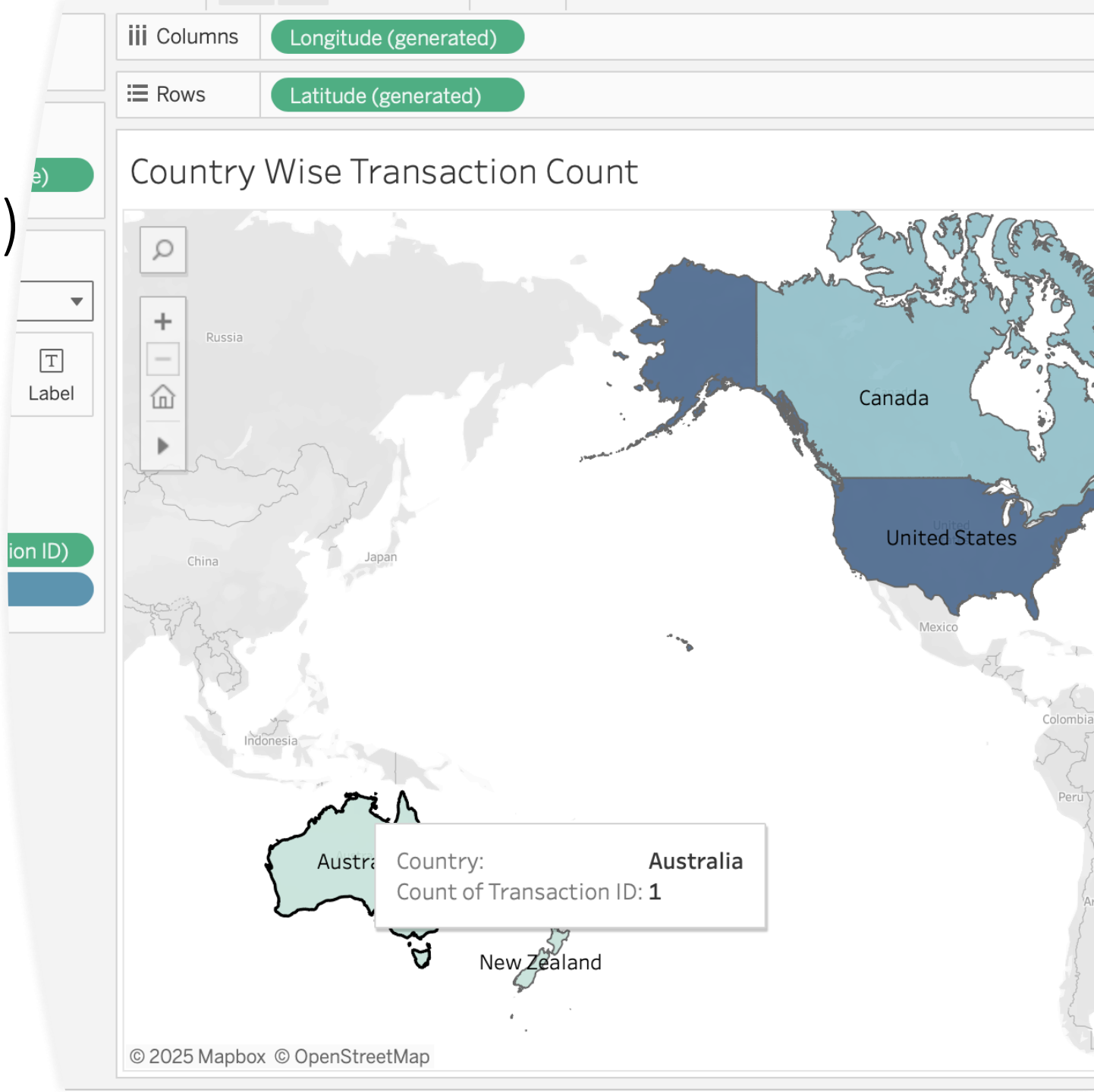
Creating the report

- Data Used: Longitude, Latitude, Region, Transaction Count and Transaction Type.
- Filters & Measures : Transaction Type

Purpose of the report and its relevance

Report provides insights into transaction activity across different countries.

- Identifying High-Activity Regions
- Fraud Detection
- Regulatory and Compliance Reporting – Ensures adherence to regulations by analyzing cross-border transaction



Report specifications (6 of 6)

Customer-wise closing balance report

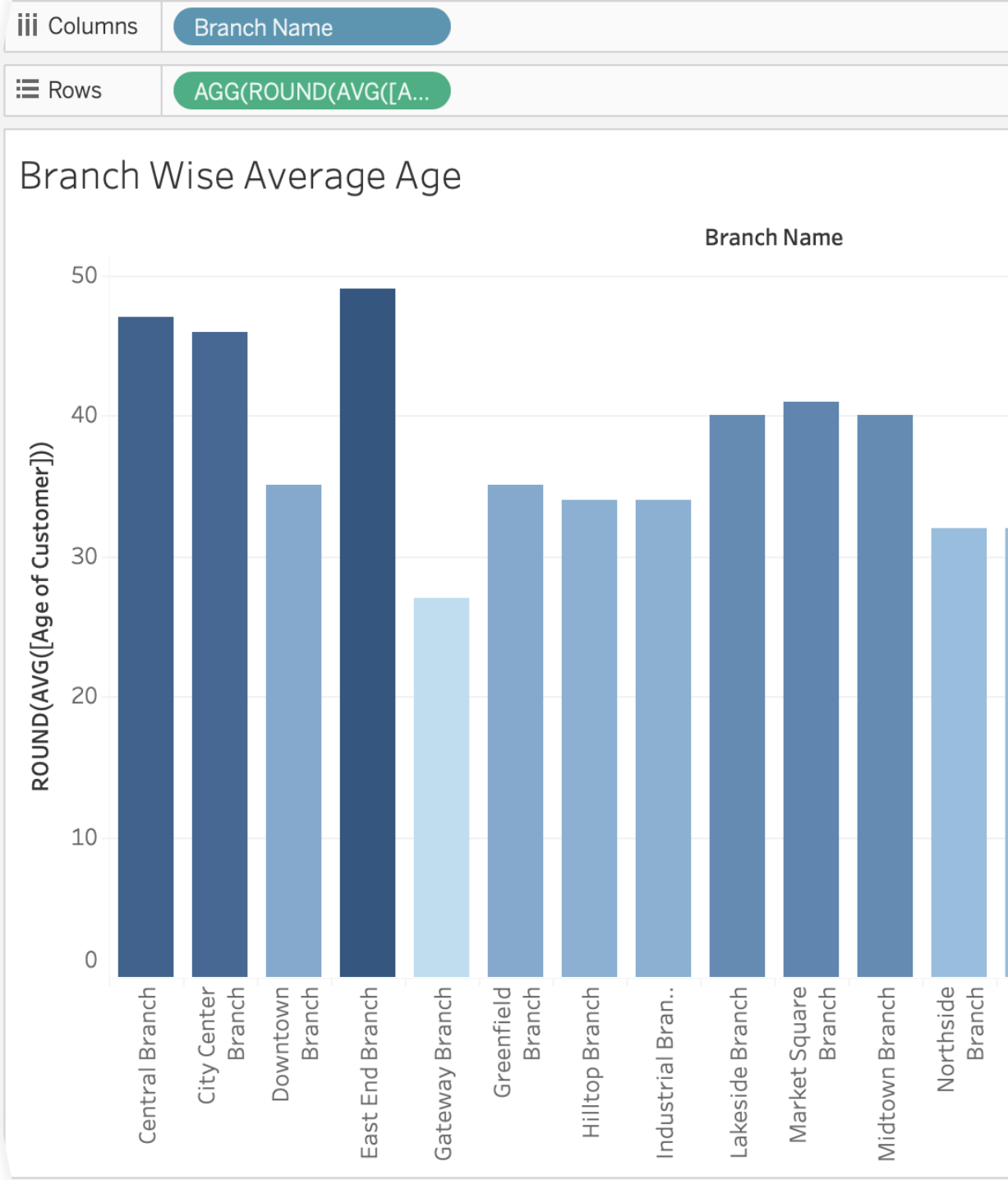
Creating the report

- Data Used: Branch Name, Age of customer
- Filters & Measures : Average Age of Cusotmer

Purpose of the report and its relevance

Report provides insights into the age distribution of customers.

- Customer Segmentation – grouping customers into different age categories
- Targeted Financial Products – models suitable banking products based on customer age
- Credit Scoring – Supports analyzing age-related spending and saving behaviors.



Performance analysis guidelines (1 of 4)

KPI of performance analysis

Closing Balance

- Formula: Closing Balance=Opening Balance+Deposits−Withdrawals
- Role: Used for trend analysis and financial forecasting to evaluate account balances over time.

Branch Wise Closing Balance

- Formula: Branch Wise Closing Balance = ∑(Closing Balance per Branch)
- Role: Identifies high-performing and low-performing branches. Supports planning on branch operations and customer engagement.

Country Wise Transactions Count

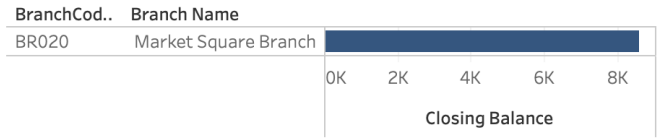
- Formula: Country Wise Transactions Count=∑(Transaction counts : Deposits + Withdrawals per country)
- Role: Regions with high financial activity. Monitoring customer engagement by location.

Branch Wise Average Age

- Formula: Branch-Wise Average Age= $\frac{\sum(\text{Ages of Customers in the Branch})}{\text{Total Customers in the Branch}}$
- Role: Indicate the customer demographic for personalized financial services. Supports workforce planning.

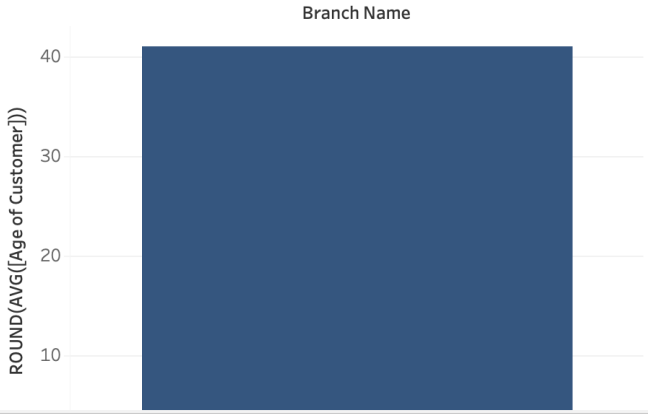
Performance analysis guidelines (2 of 4)

Customer Name	Opening Balance	Closing Balance
Gina King	20,000	20,750
Harry Brown	3,500	3,300
Ivy Scott	8,000	8,600
John Doe	0	0
Karen Green	6,000	6,400
Liam Miller	5,500	5,400
Mona Blue	4,000	4,250
Nate White	10,000	11,000
Olivia Black	1,500	1,700
Paul Walker	-500	0
Quinn Red	23,000	23,800
Rose Pink	8,700	8,000
Sam Grey	500	850
Tim Orange	9,200	8,600



Country Wise Transaction Count

Branch Wise Average Age



Dashboard offers data-driven insights for strategic decision-making, helping optimize branch performance, customer engagement, and financial forecasting.
With interactive filtering, data can be drilled down to spotlight key insights.

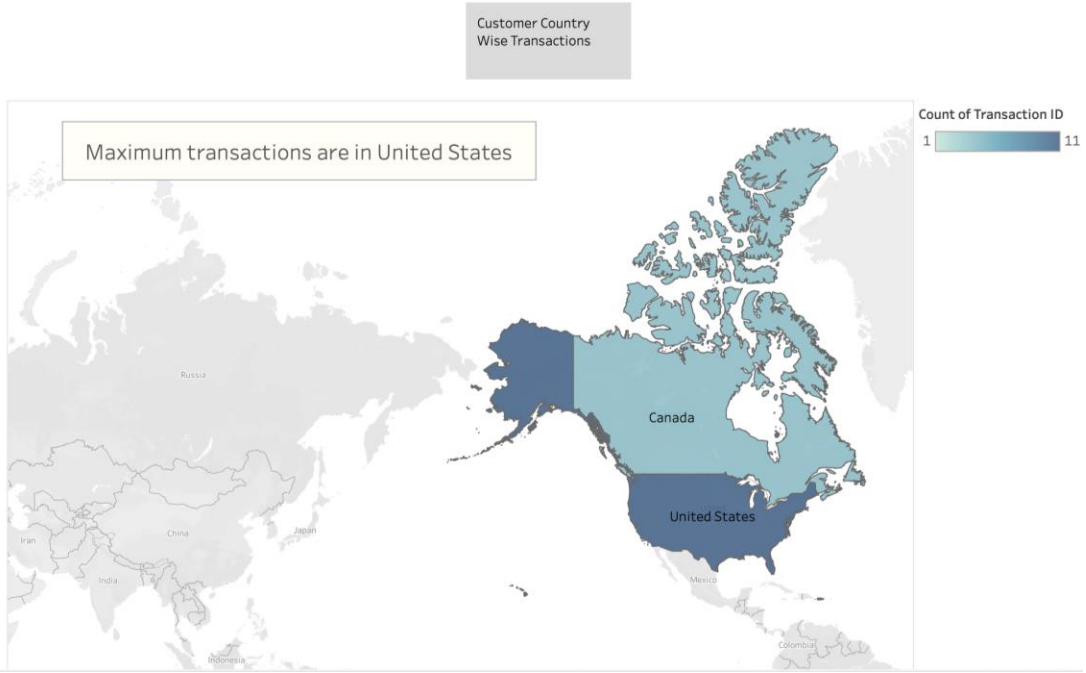
Performance analysis guidelines (3 of 4)

Story can highlight the key takeaway and present a clear narrative rather than just isolated charts.

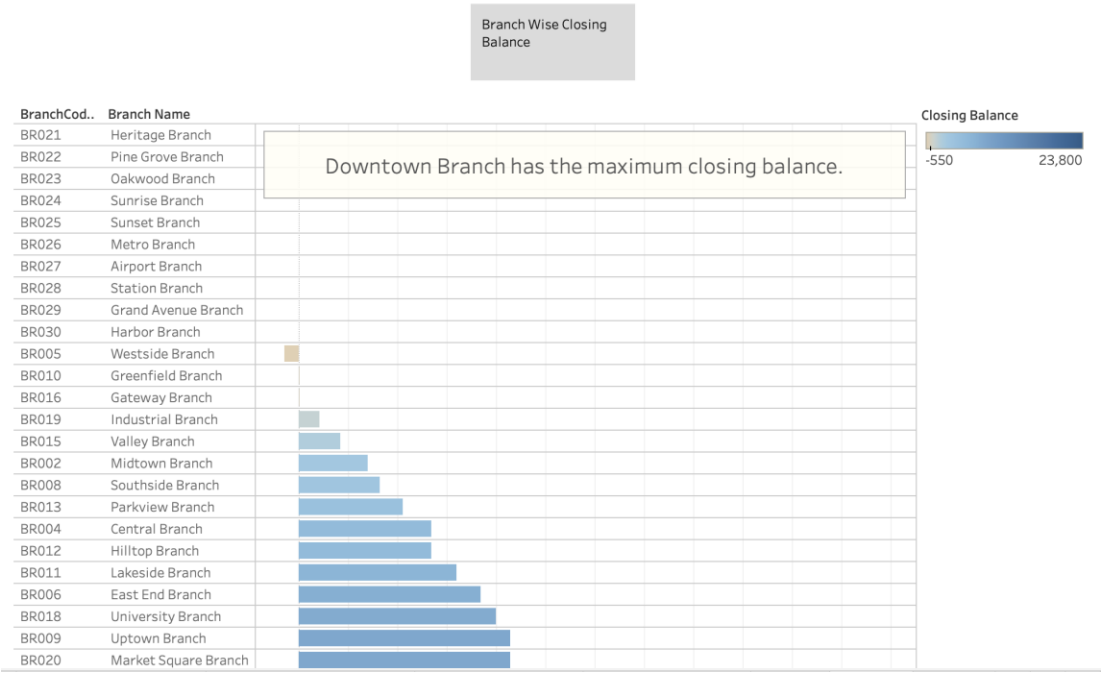
In Country Wise Analysis story, it exhibits United States as location where maximum transactions occurred.

In Branch Wise Analysis story, it displays the downtown branch as maximum closing balance branch.

Country Wise Analysis



Branch Wise Analysis



Performance analysis guidelines (4 of 4)

Maintaining and updating dashboards

- Automate scheduled data refresh (daily, hourly)
- Identify new KPIs based on business needs
- Use aggregated tables instead of raw data
- Monitor performance using built-in tools
- Gather feedback from stakeholders
- Improve dashboard layout and usability
- Maintain documentation for updates and best practices

General handover documentation

Role and key responsibilities

Role Overview

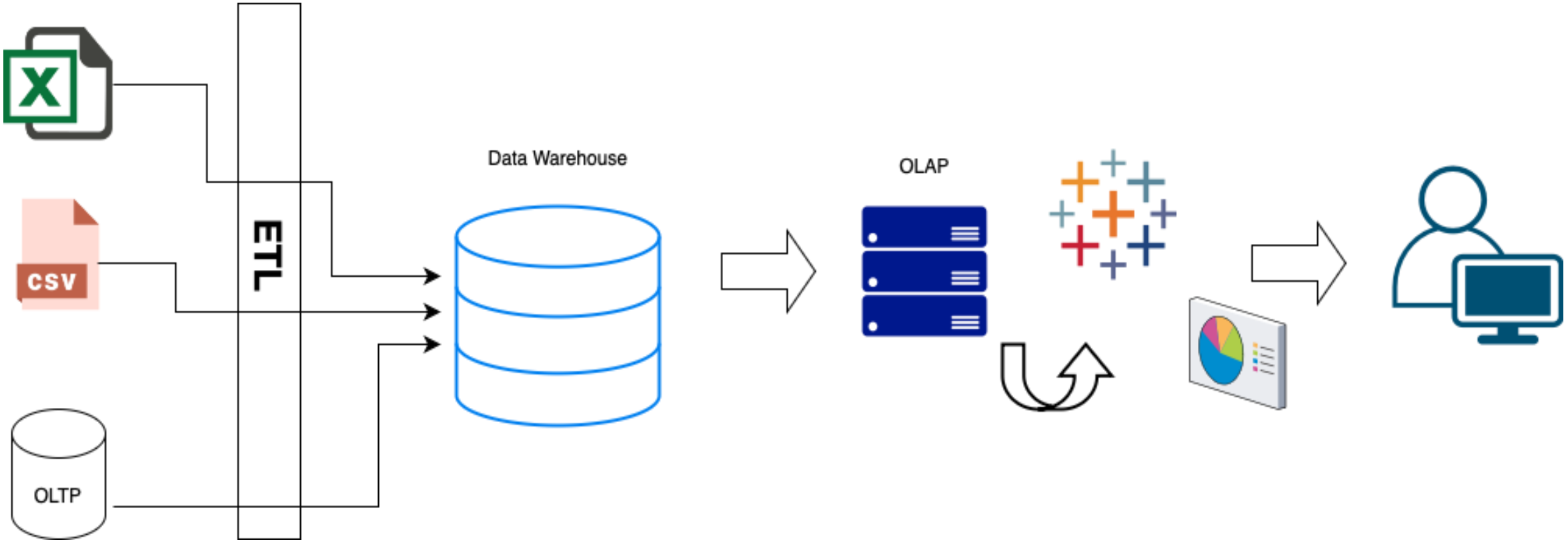
- Carry out data integration, governance, security, and reporting frameworks, ensuring financial data integrity and compliance.

Key Responsibilities:

- Data Integration & Quality: Streamline multi-source data, data cleaning and enrichment
- Database modeling : Designing ERD, table structure and database schema with SQL CLI and workbench
- Security & Privacy: Implement encryption, RBAC, and regulatory compliance protocols, in MySQL and phpMyAdmin
- Dashboard & Reporting Management: Create dashboard and reporting system with branch performance insights & KPI using Excel, IBM Cognos, Tableau
- Performance Optimization: Enhance query efficiency, automation, and real-time analytics

System architecture diagram

High-level overview of system architecture



Key challenges and resolutions

Data Quality Issue: Inconsistent, missing, or duplicate data from Excel

Resolution:

- Implement data validation in excel during ETL.
- Standardize data formats before loading.

OLAP Query Performance: Slow response times for complex analytical queries.

Resolution:

- Use pre-aggregate data
- Use materialized views
- Optimize cube design for multi-dimensional analysis.

Data Security & Access Control: Unauthorized access to sensitive financial or customer data.

Resolution:

- Implement role-based access control (RBAC)
- Use encryption for data storage.
- Restrict access to raw transactional data at the OLTP level.

Tableau Dashboard Performance Issues: Slow dashboard loading

Resolution:

- Reduce the number of filters and calculations in Tableau.

Insights and recommendations

Insights and recommendations

Data Governance and Compliance

- Maintain audit logs to ensure accountability.
- Regularly review data retention policies to comply with regulations and avoid unnecessary storage costs.

Data Quality Control

- Implement data deduplication and normalization techniques to reduce redundancies.

Data Security and Privacy

- Implement multi-factor authentication (MFA) and strong password policies to secure user access.

Summary

Ensuring thorough handover of the Data Management System, presentation covers data integration and cleaning, data structure, ETL workflows, and integration across systems.

With the documentations, Data architecture blueprints for future reference and recommended actions, it will ensure that the Data Management System remains scalable, secure, and efficient.

Appendix

Appendix

Link for Tableau Dashboard

Tableau Dashboard