Case Study For Zulo Bank

EXECUTIVE SUMMARY

This case study provides a comprehensive overview of the process involved in database and data warehouse modeling for Zulo Bank, aimed at beginner data engineers.

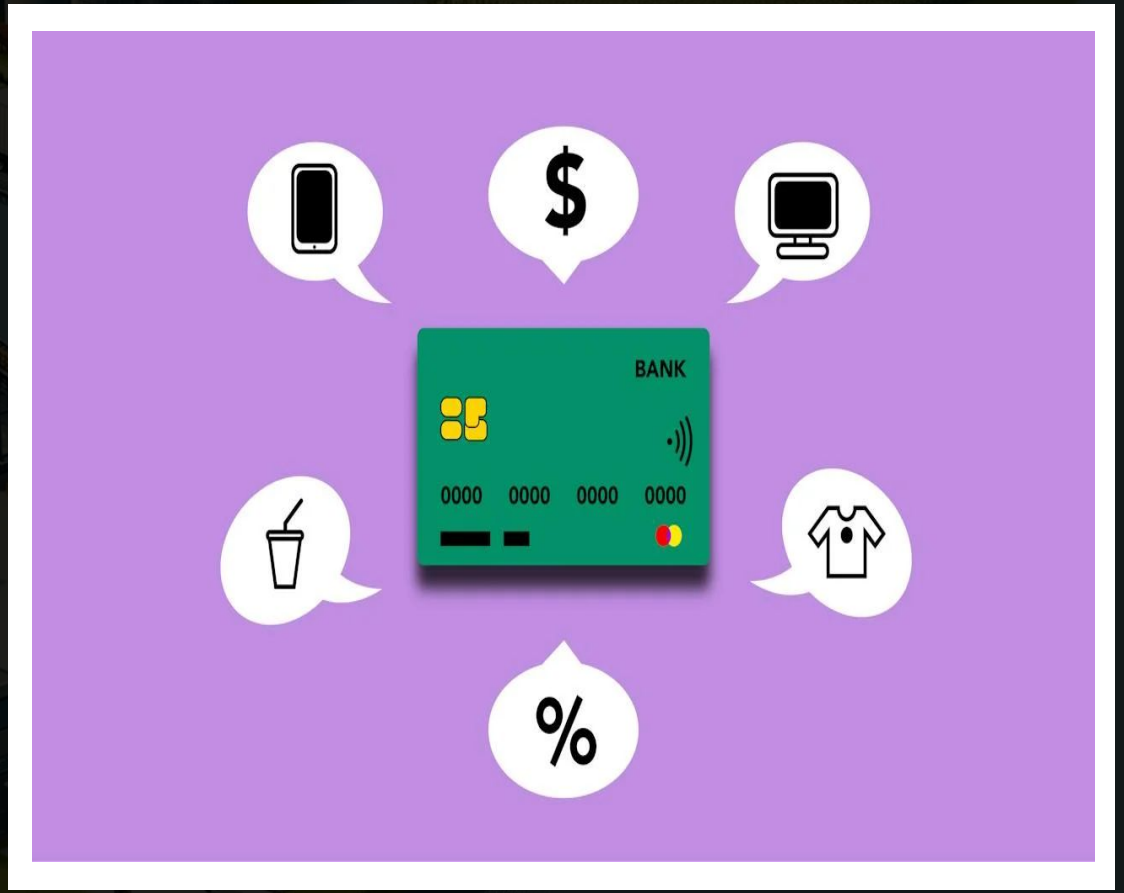
It covers the foundational aspects of database design, including gathering business requirements, understanding and applying key database principles such as primary and foreign keys, normalization, and denormalization, as well as designing schemas.

The study transitions into data warehouse modeling, focusing on schema transformation, with an emphasis on the practical application of these concepts to solve real-world banking data management challenges.

BUSINESS PROBLEM STATEMENT

Zulo Bank seeks to overhaul its data management system to enhance efficiency, improve data quality, and enable advanced analytics capabilities. The current system is plagued with issues related to data redundancy, inconsistency, and accessibility, hindering real-time decision-making and monthly reporting.

Data engineers are tasked with designing a new system that resolves these issues, ensuring seamless data integration, reporting, and analytics.



TECH STACK

A. **Python:**
Used for data processing, analysis, and synthetic data generation. Python's extensive libraries (Pandas) facilitate efficient data manipulation for testing and development.

B. **SQL (Structured Query Languages)**

C. **Pgadmin (Postgres Server)**

PROJECT SCOPE

1. **Database Design:**
   Construct a normalized relational database schema for operational use, covering entities such as customers, accounts, transactions, and loans. The design process includes identifying primary and foreign keys, implementing normalization up to the third normal form (3NF), and ensuring data integrity and efficiency.

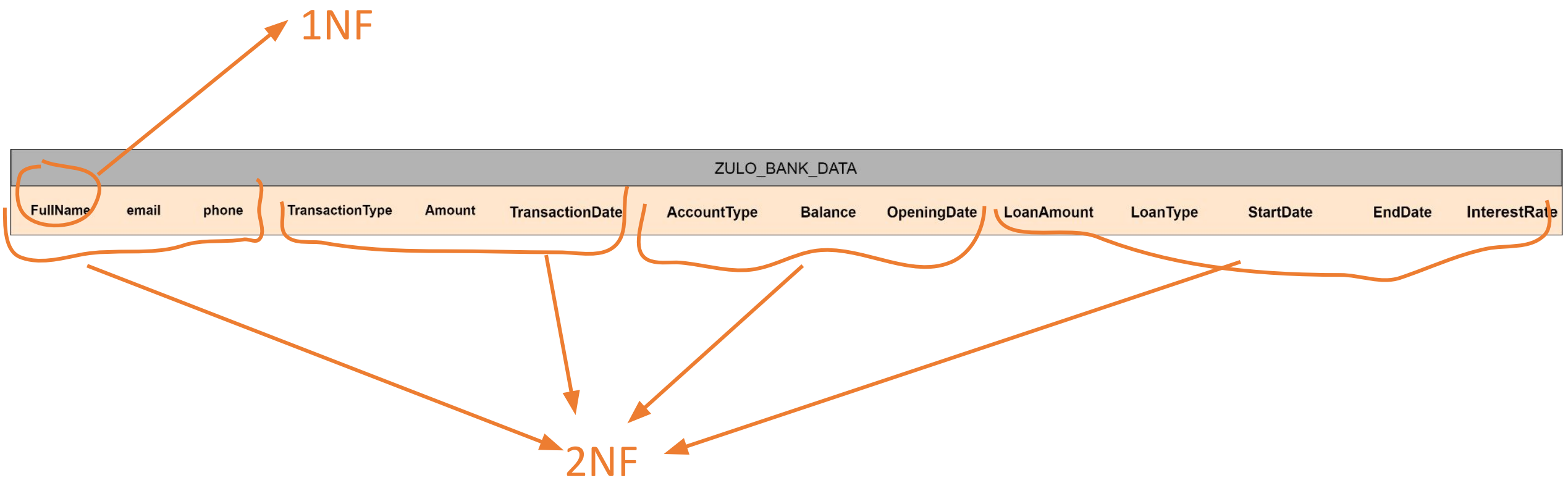
1. **Data Warehouse Modeling:**
   Develop a data warehouse schema optimized for analysis and reporting. This involves transforming the normalized database schema into a star schema, including fact and dimension tables, to support complex queries and business intelligence needs.

1. **Python Programming:**
   Develop python codes to ingest data, transform it and load it into the PostgreSQL database

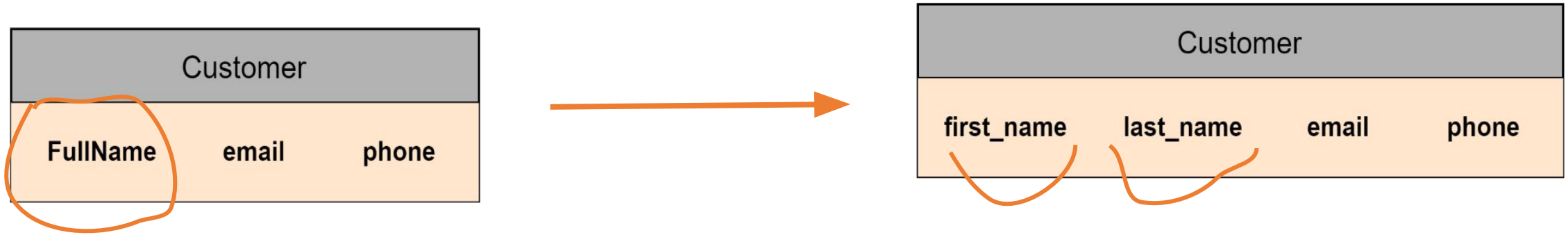
2. **Version Control:**
   Carry out good version control practice but using Git Desktop to Push and pull files from our local repository to git repository

This offers a brief overview of the tools and technologies involved in the Zulo Bank case study, alongside a concise description of the project's scope, from database and data warehouse design to documentation.

DATABASE MODELLING

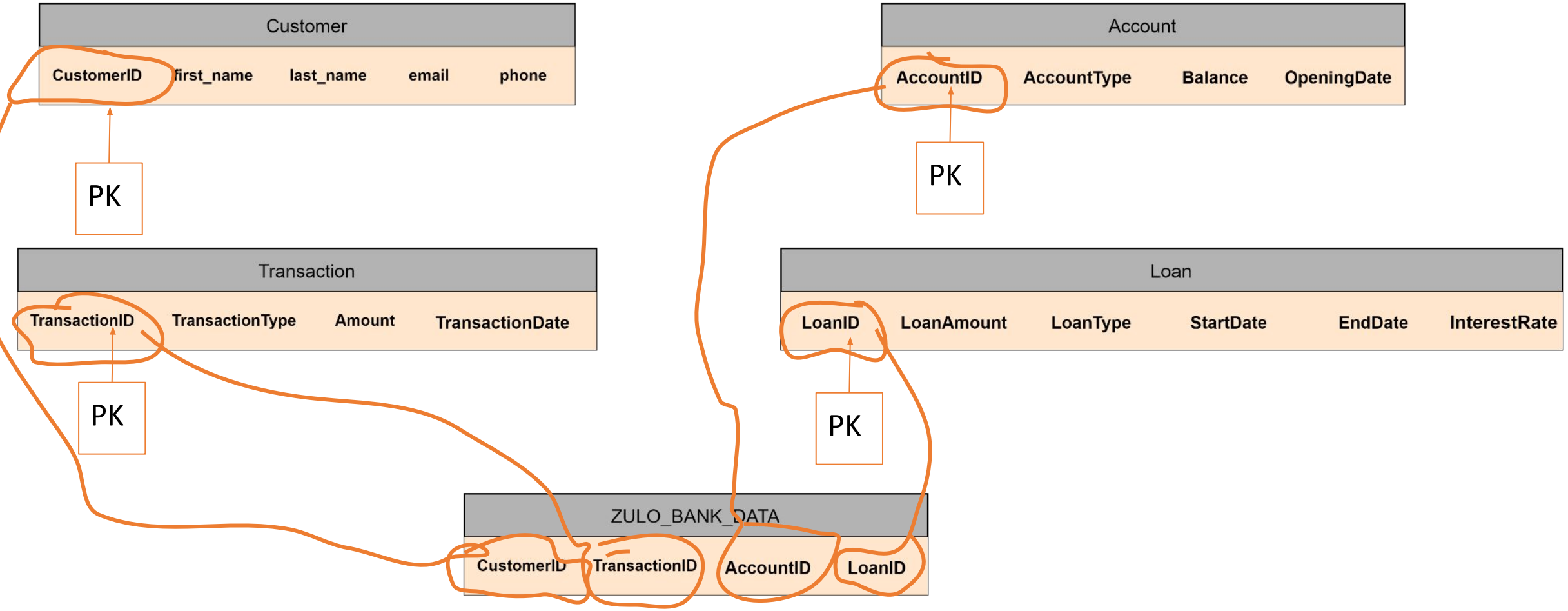


DATABASE MODELLING



Denormalized Table to 1st Normal Form

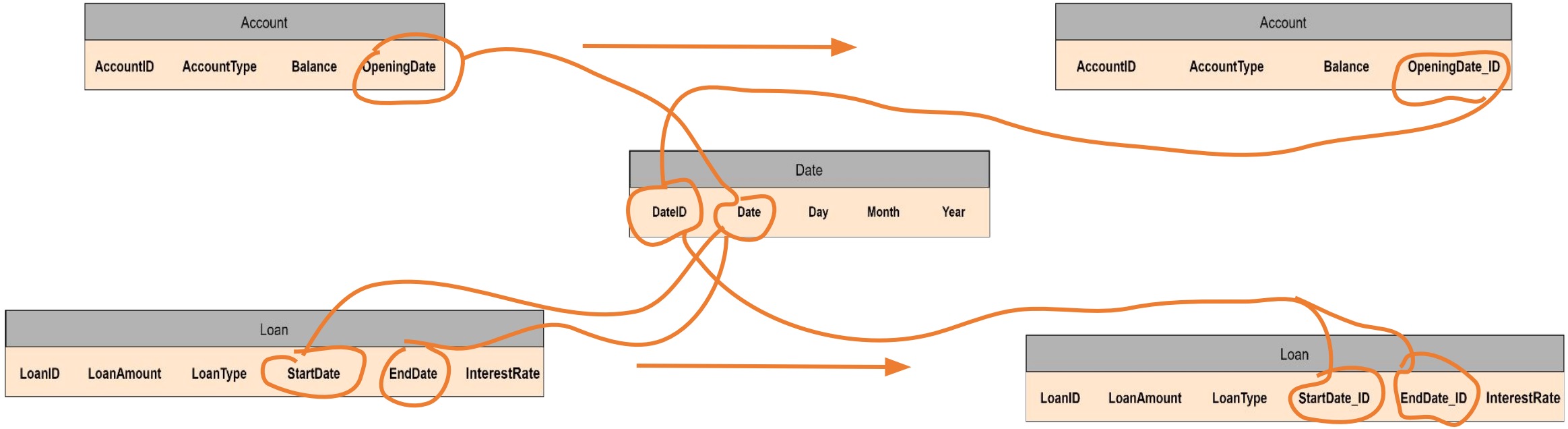
DATABASE MODELLING



1st Normal Form to 2nd Normal Form

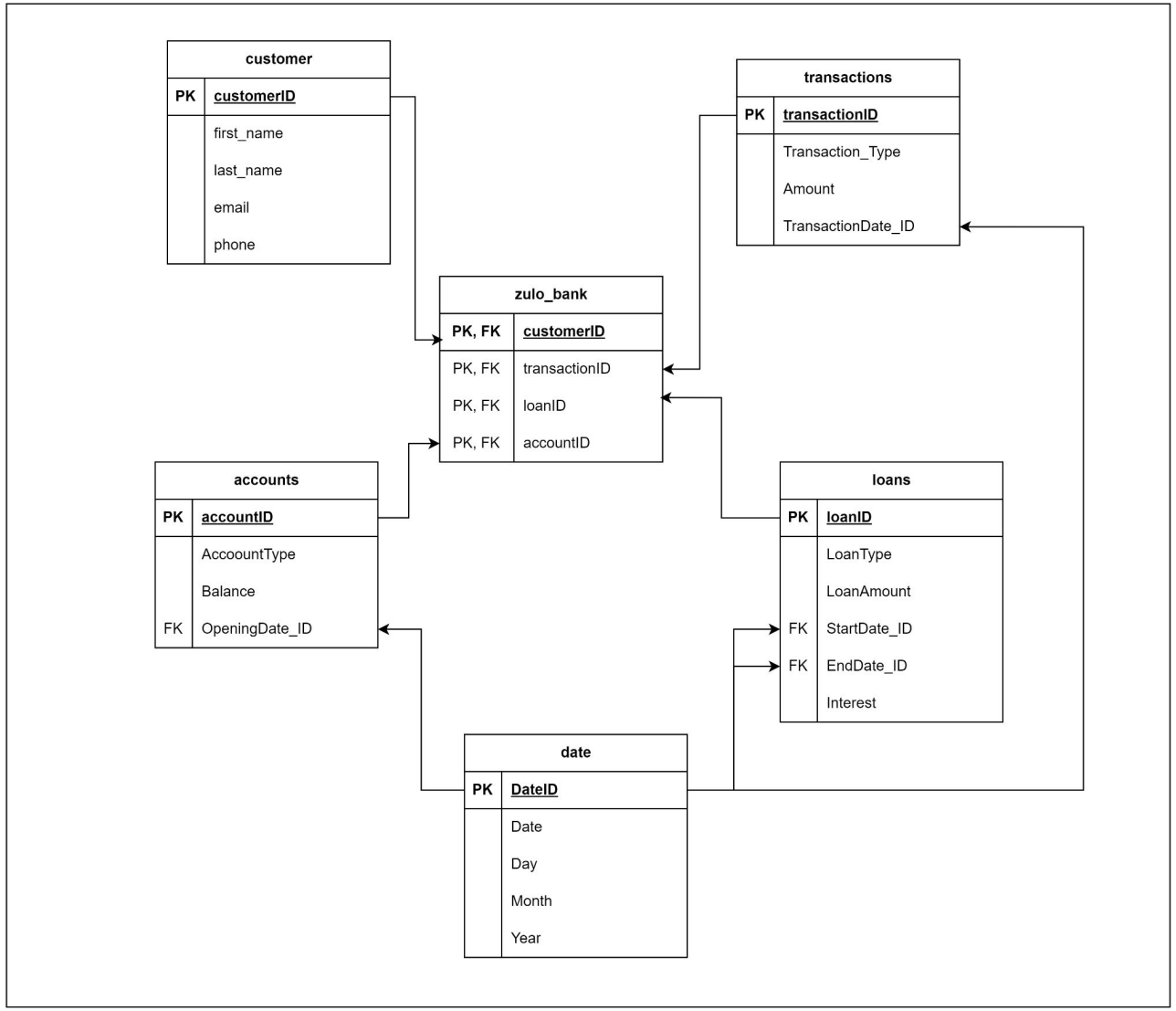
Note: We added primary keys to the Customer tables, Transaction tables, Account tables and Loan tables.

DATABASE MODELLING



2nd Normal Form to 3rd Normal Form

DATABASE MODELLING



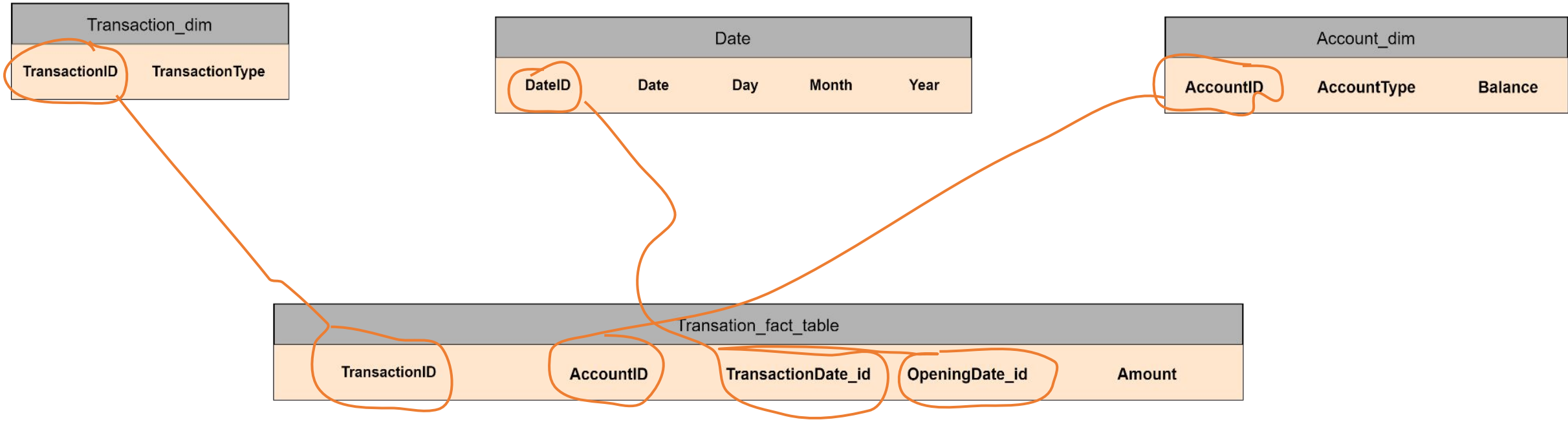
DATA WAREHOUSE MODELLING

Fact Table Consideration:

- Transactions Fact Table: Could include TransactionID, AccountID (as a foreign key to the Account dimension), date_id (as a foreign key to the Date dimension), and measurable metrics like Amount.

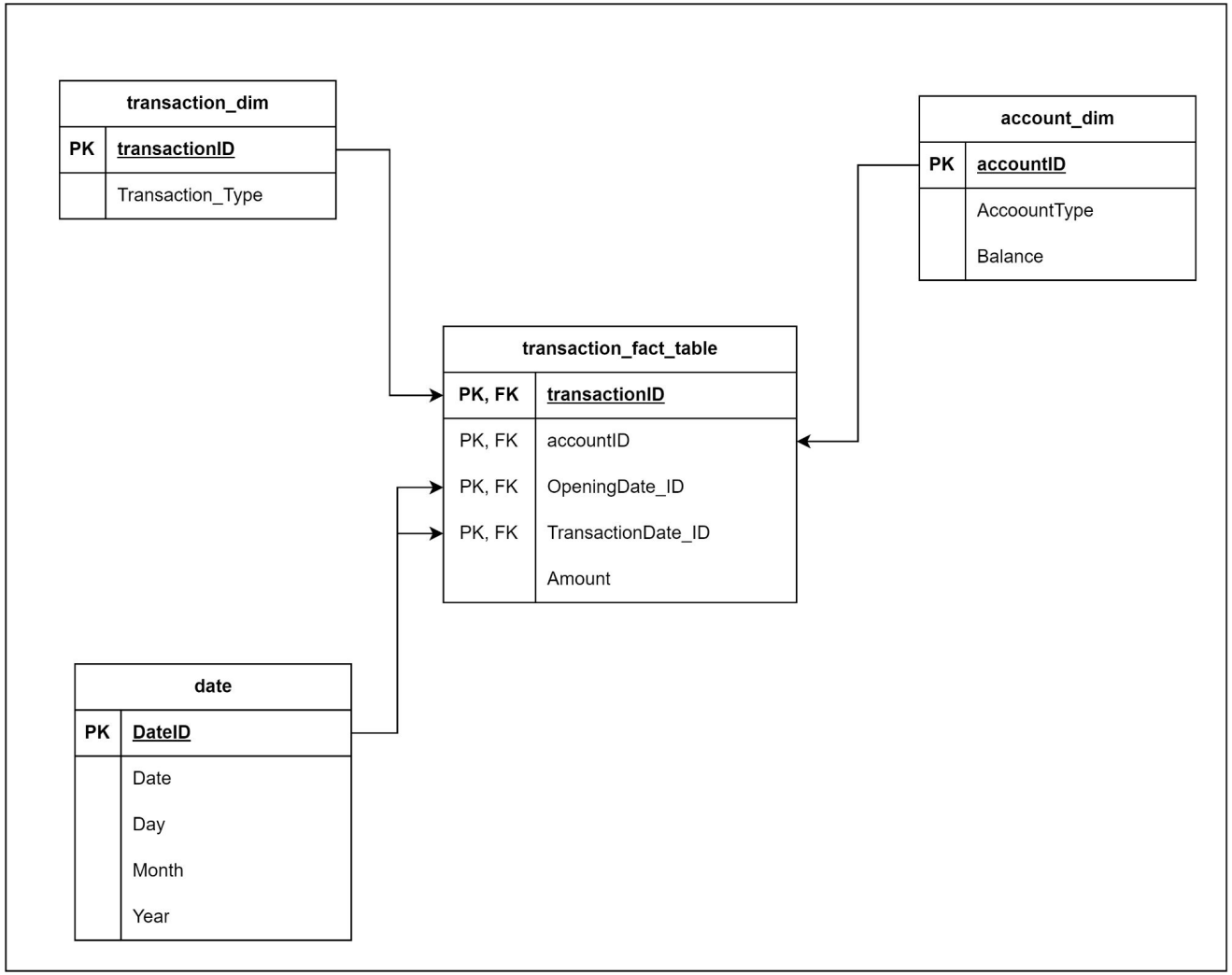
- Loans Fact Table: Could include LoanID, CustomerID (as a foreign key to the Customer dimension), start_date_id, end_date_id (as foreign keys to the Date dimension), and metrics like LoanAmount and InterestRate.

DATA WAREHOUSE MODELLING

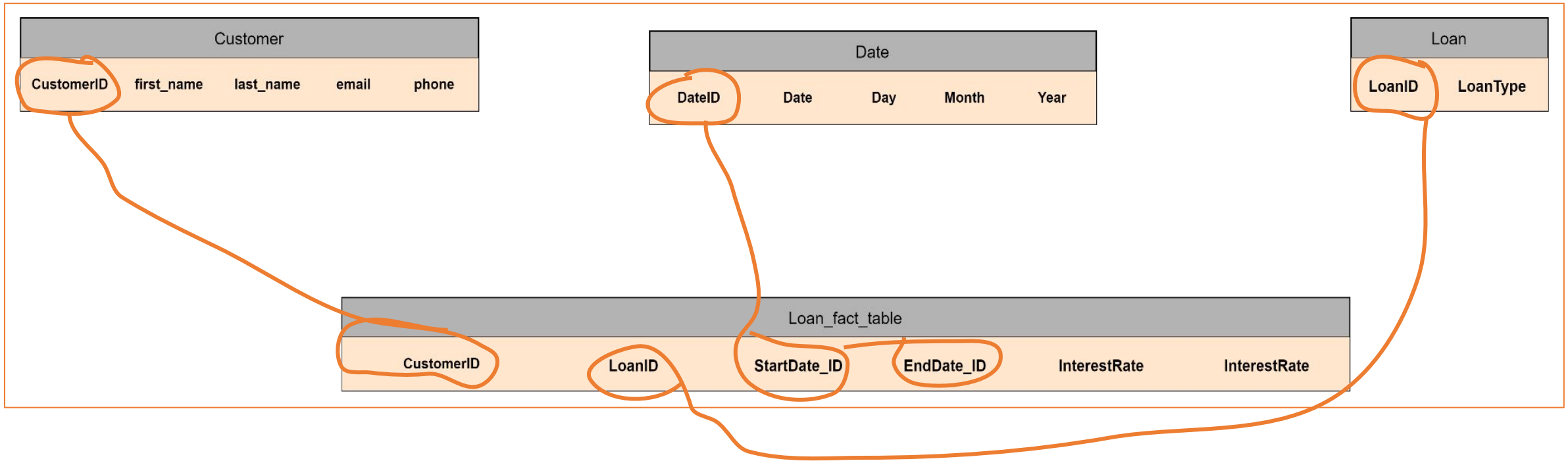


Transactions Fact Table

DATA WAREHOUSE MODELLING



DATA WAREHOUSE MODELLING



Loans Fact Table