



**SAMSUNG**

# Bank Loans **ELT** Data Pipeline

**Team members:**

Ahmed Mohsen - Hanin Baher - Mahmoud Ashraf

**Facilitator:**

Rawan Ehab

# Table of Contents

**01** Introduction

**03** Key Tasks

**02** Problem Statment

**04** Methodology

# Table of Contents

**05** Our pipeline

**07** Conclusion

**06** Results & Insights

**08** Future Work



01

# Introduction

# Introduction

- Big data is increasingly used for financial risk analysis.
  - The goal: Analyze loan default risk using a full ELT pipeline.
  - Pipeline spans from raw data ingestion to visualization.
- 
- In the U.S., the average default rate for personal loans ranges from 2–6%, but this can spike above 10% during economic downturns.
  - According to the World Bank, non-performing loans (NPLs) globally account for over \$1.4 trillion, highlighting the critical need for predictive risk management.



02

# **Problem Statment**

A decorative graphic on the left side of the slide consisting of two overlapping squares. The top square is a lighter blue, and the bottom square is a darker blue, creating a cross-like shape.

# The Problem

Loan providers struggle with fragmented data, making it hard to compute risk metrics like DTI and LTV.

Manual analysis is slow, and without centralized dashboards, real-time monitoring, and data-driven decisions are limited.



03

## Key Tasks



# Key tasks

Cluster setup



Using Docker and Postgres for the source database.

Data Ingestion



Extract raw loan data from Postgres into HDFS using Sqoop.

Data Transformation



By using PySpark (Zeppelin) in:

- Cleaning Data
- Dimensional modeling (fact + dimension tables)

Data Warehouse Loading



Into Hive with Parquet storage.

Visualization & Analytics



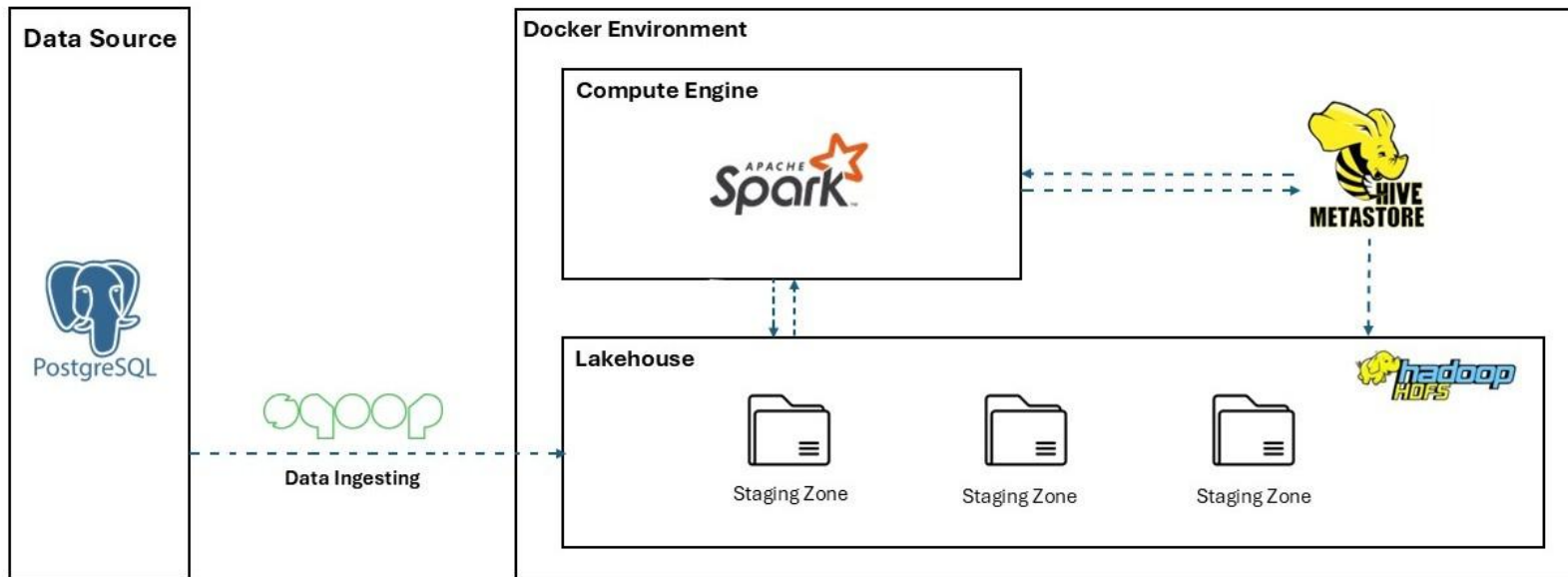
Using Power BI dashboards.



04

# Methodology

# Methodology





05

**Our pipeline**

# Cluster Setup

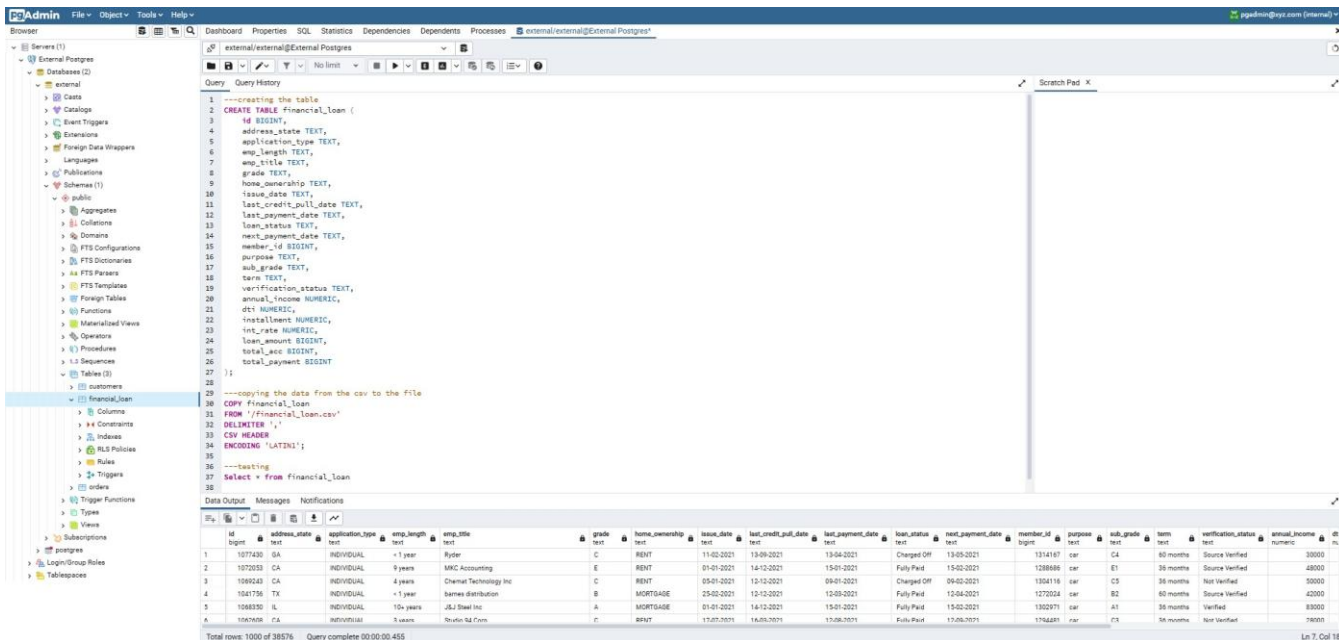
- Docker Compose used for multi-container cluster
- Services included: Postgres, Hive, Sqoop, Zeppelin, HDFS, Hue, Power BI connection

## URL references:

- PgAdmin (<http://localhost:5000>)
- Zeppelin (<http://localhost:8082>)
- Hue (<http://localhost:8888>)
- Verified services via docker ps.

```
[+] Running 14/14
✓Network big-data-cluster_default      Created
✓Container nodemanager                 Started
✓Container resourcemanager             Started
✓Container historyserver               Started
✓Container external_postgres_db        Started
✓Container cassandra                   Started
✓Container huedb                       Started
✓Container hive-metastore-postgresql    Started
✓Container namenode                    Started
✓Container datanode                     Started
✓Container hive-metastore               Started
✓Container hue                          Started
✓Container external_pgadmin            Started
✓Container hive-server                  Started
```

# Data Creation (Postgres)






The screenshot shows the pgAdmin interface with the 'Query' tab selected. The query history displays the following SQL commands:

```
1 ---creating the table
2 CREATE TABLE financial_loan (
3     id BIGINT,
4     address_state TEXT,
5     application_type TEXT,
6     emp_length TEXT,
7     emp_title TEXT,
8     grade TEXT,
9     home_ownership TEXT,
10    issue_date TEXT,
11    last_credit_pull_date TEXT,
12    last_payment_date TEXT,
13    loan_status TEXT,
14    next_payment_date TEXT,
15    member_id BIGINT,
16    purpose TEXT,
17    sub_grade TEXT,
18    term TEXT,
19    verification_status TEXT,
20    annual_income NUMERIC,
21    dti NUMERIC,
22    installment NUMERIC,
23    int_rate NUMERIC,
24    loan_amount BIGINT,
25    total_acc BIGINT,
26    total_payment BIGINT
27 );
28
29 ---copying the data from the csv to the file
30 COPY financial_loan
31 FROM '/financial_loan.csv'
32 DELIMITER ','
33 CSV HEADER
34 ENCODING 'LATIN1';
35
36 ---listing
37 select * from financial_loan
38
```

The 'Data Output' tab shows the first 10 rows of the data:

id	address_state	application_type	emp_length	emp_title	grade	home_ownership	issue_date	last_credit_pull_date	last_payment_date	loan_status	next_payment_date	member_id	purpose	sub_grade	term	verification_status	annual_income	dti
1	1077430 CA	INDIVIDUAL	< 1 year	Ryder	C	RENT	11-03-2021	13-09-2021	13-04-2021	Charged Off	13-03-2021	1314187	car	C4	60 months	Source Verified	30000	
2	1072053 CA	INDIVIDUAL	8 years	MKC Accounting	E	RENT	01-01-2021	14-12-2021	15-01-2021	Fully Paid	15-02-2021	1288686	car	E1	36 months	Source Verified	48000	
3	1089243 CA	INDIVIDUAL	4 years	Chemel Technology Inc	C	RENT	05-01-2021	12-12-2021	09-01-2021	Charged Off	09-02-2021	1304118	car	C5	36 months	Not Verified	50000	
4	1041756 TX	INDIVIDUAL	< 1 year	Barnes distribution	B	MORTGAGE	25-02-2021	12-12-2021	12-03-2021	Fully Paid	12-04-2021	1272024	car	B2	60 months	Source Verified	42000	
5	1068330 IL	INDIVIDUAL	10+ years	J&J Steel Inc	A	MORTGAGE	01-01-2021	14-12-2021	15-01-2021	Fully Paid	15-02-2021	1302971	car	A1	36 months	Verified	83000	
6	1057658 CA	INDIVIDUAL	4 years	NOVA H2 Corp	F	MORTGAGE	17-07-2021	14-01-2021	17-08-2021	Fully Paid	17-08-2021	1762291	car	F1	36 months	Not Verified	14000	

Total rows: 1000 of 38576 Query complete 00:00:00.455

-  CSV dataset (financial\_loan.csv) uploaded into Postgres via pgAdmin.
-  Created table financial\_loan with 20+ attributes (borrower details, loan amount, payment dates, etc.).
-  Used COPY command for efficient data import.

Verified with SQL queries (SELECT \* LIMIT 10).

# Data Ingesting (Sqoop)

**Sqoop imports data from Postgres → HDFS as Parquet files**

**First:**

Opening Sqoop inside the Hive container:

`docker exec -it hive-server bash`

**Second:**

Running Sqoop import command

Path: `/staging_zone/financial_loan`

```
sqoop import \  
--connect jdbc:postgresql://external_postgres_db/postgres \  
--username external \  
--password external \  
--table financial_loan \  
--target-dir /staging_zone/financial_loan \  
--as-parquetfile \  
--m 1
```

# Data Storage (HDFS)

Raw financial loan data is stored in HDFS.

Benefits:

- High availability & fault tolerance.
- Scalability for millions of records.
- Parallel access for Spark transformations.

```
root@90b03649b1d5:/# hdfs dfs -ls /staging_zone/financial_loan
Found 3 items
drwxr-xr-x - root supergroup          0 2025-10-01 00:42 /staging_zone/financial_loan/.metadata
drwxr-xr-x - root supergroup          0 2025-10-01 00:42 /staging_zone/financial_loan/.signals
-rw-r--r-- 3 root supergroup    1698990 2025-10-01 00:42 /staging_zone/financial_loan/983a2c69-23b0-482e-81c8-035f5d8e1366.parquet
```



# Data Transformations & Modeling (Spark)

## Data Cleaning

Null handling in Emp\_title cloumn, emp\_length and term columns formatting, date type conversions.

## Dimensional Modeling

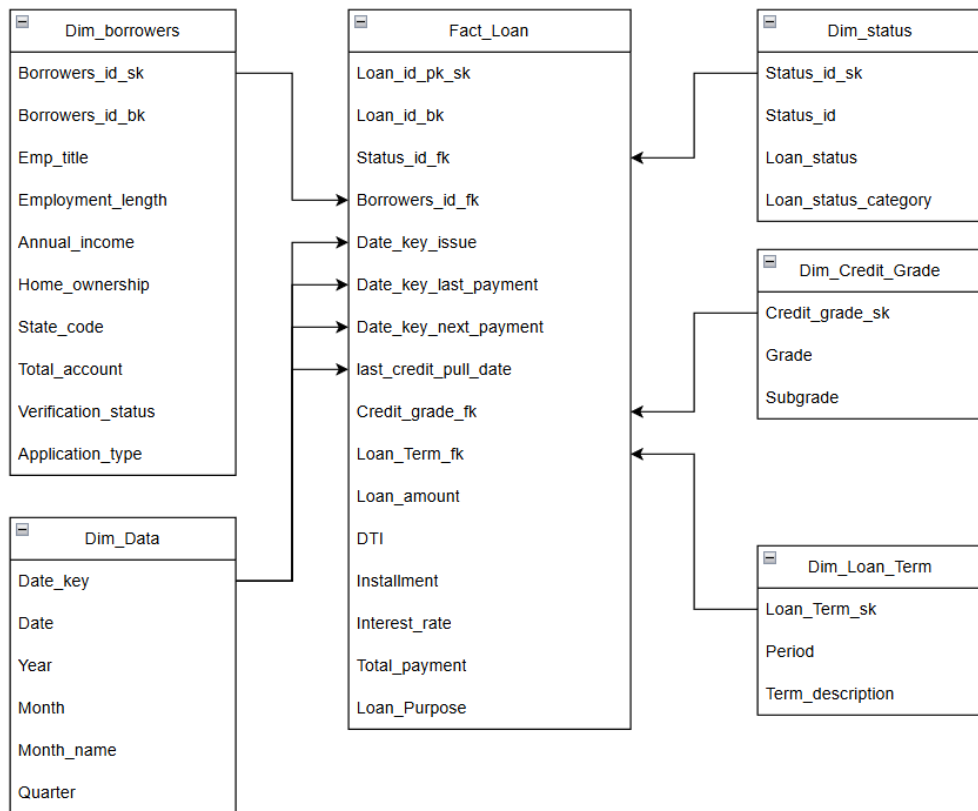
Fact Table:

- fact\_loan (central metrics, risks, loan status).

Dimensions:

- dim\_borrowers (borrower details) - dim\_status (loan status categories)
- dim\_credit\_grade (risk grades) - dim\_loan\_term (term duration)
- dim\_date (date dimension).

# Data Transformations & Modeling (Spark)



**Hive**

Query History   Saved Queries   Query Builder   Results (10)

```
SELECT * FROM fact_loan LIMIT 10
```

	fact_loans.loan_id_pk_sk	fact_loans.loan_id_bk	fact_loans.borrowers_id_fk	fact_loans.date_key_issue	fact_loans.date_key_last_payment	fact_loans.date
1	0	1077430	8589953946	20210211	20210413	20210513
2	1	1077430	8589953946	20210211	20210413	20210513
3	2	1077430	8589953946	20210211	20210413	20210513
4	3	1077430	8589953946	20210211	20210413	20210513
5	4	1077430	8589953946	20210211	20210413	20210513
6	5	1072053	8589953515	20210101	20210115	20210215
7	6	1072053	8589953515	20210101	20210115	20210215
8	7	1072053	8589953515	20210101	20210115	20210215
9	8	1072053	8589953515	20210101	20210115	20210215
10	9	1072053	8589953515	20210101	20210115	20210215

**Tables**

- default.fact\_loans
  - loan\_id\_pk\_sk: bigint
  - loan\_id\_bk: bigint
  - borrowers\_id\_fk: bigint
  - date\_key\_issue: int
  - date\_key\_last\_payment: int
  - date\_key\_next\_payment: int
  - date\_key\_last\_credit\_pull: int
  - credit\_grade\_fk: bigint
  - loan\_term\_fk: bigint
  - loan\_amount: bigint
  - dtt: string
  - installment: string
  - interest\_rate: string
  - total\_payment: string
  - loan\_purpose: string
  - status\_id\_fk: bigint

		fact_loans.loan_id_pk_sk	fact_loans.loan_id_bk	fact_loans.borrowers_id_fk	fact_loans.date_key_issue	fact_loans.date_key_last_payment	fact_loans.date_key_maturity
<div><div><div></div><div></div><div></div><div></div></div><div></div></div>	1	0	1077430	8589953946	20210211	20210413	20210513
	2	1	1077430	8589953946	20210211	20210413	20210513
	3	2	1077430	8589953946	20210211	20210413	20210513
	4	3	1077430	8589953946	20210211	20210413	20210513
	5	4	1077430	8589953946	20210211	20210413	20210513
	6	5	1072053	8589953515	20210101	20210115	20210215
	7	6	1072053	8589953515	20210101	20210115	20210215
	8	7	1072053	8589953515	20210101	20210115	20210215
	9	8	1072053	8589953515	20210101	20210115	20210215
	10	9	1072053	8589953515	20210101	20210115	20210215



06

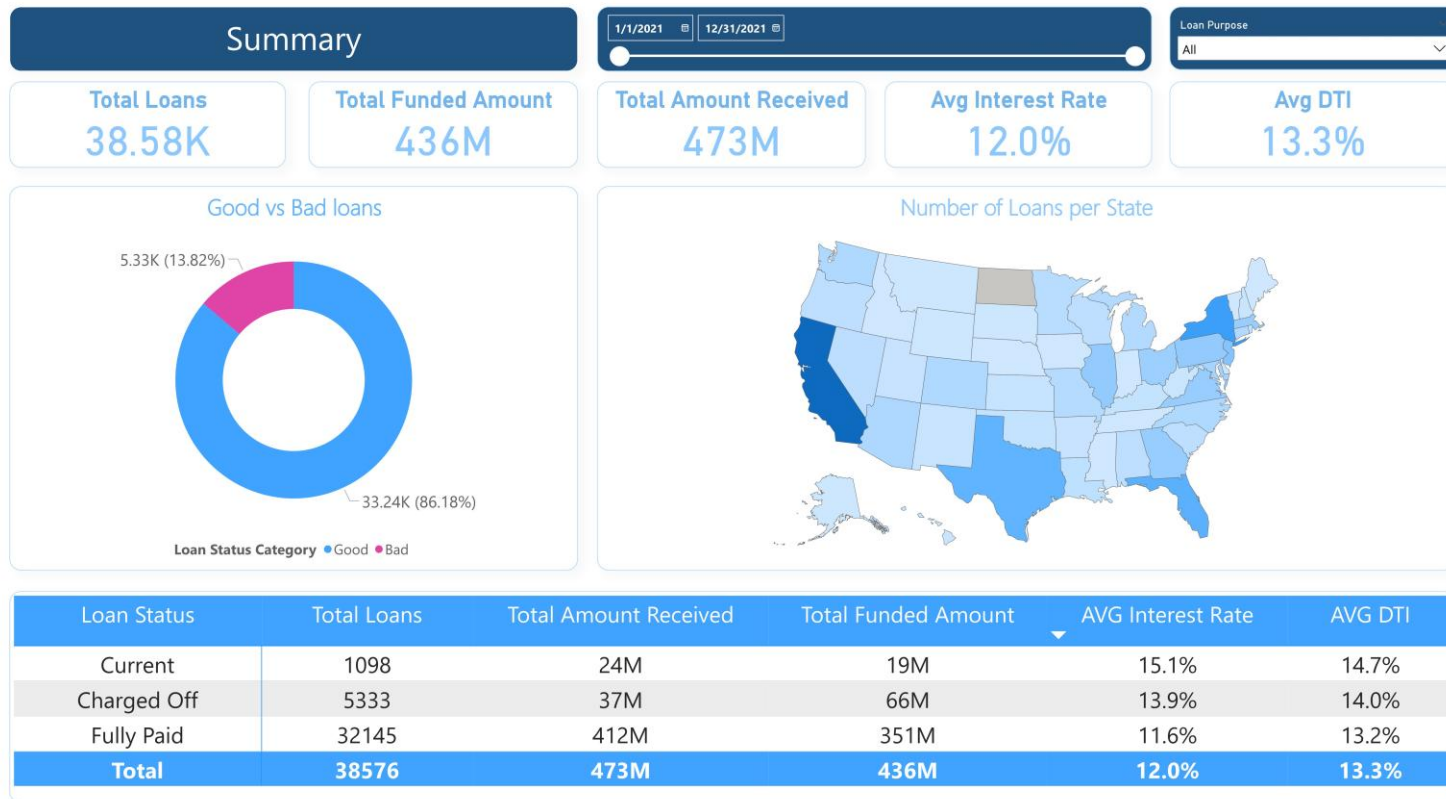
## **Results & Insights**

# Power BI Connection

Connected Power BI to Hive via ODBC.

- Host: localhost
- Port: 10000
- Authentication: Username & Password
- Database: your Hive DB
- Test connection → Save

# Power BI Dashboards



# Power BI Dashboards

## Overview

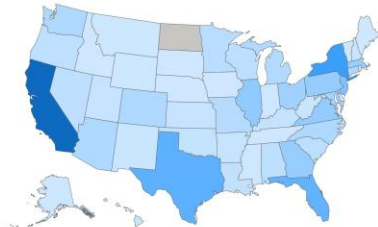
1/1/2021

12/31/2021

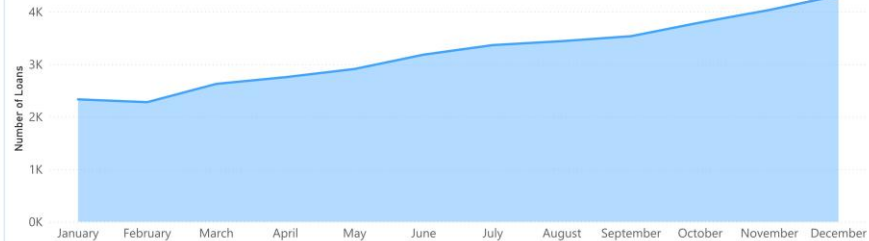
Loan Purpose

All

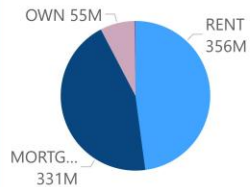
### Loan Amount per State



### Total Loan Applications by Month



### Loan Applications by Home Ownership



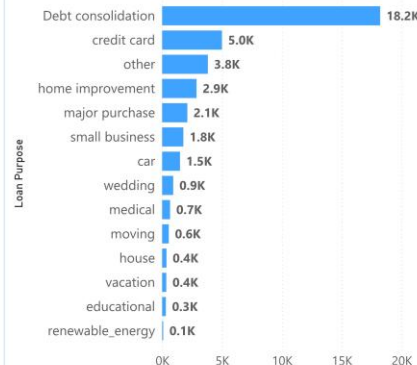
● RENT ● MORTG... ● OWN ● OTHER

### Loan Terms Distribution

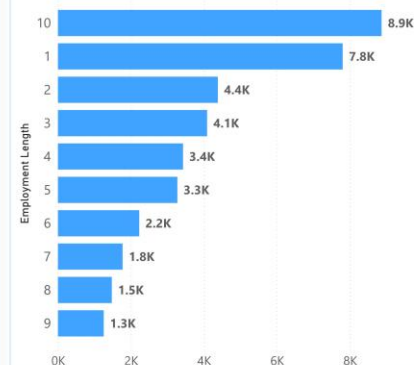


● 36 months ● 60 months

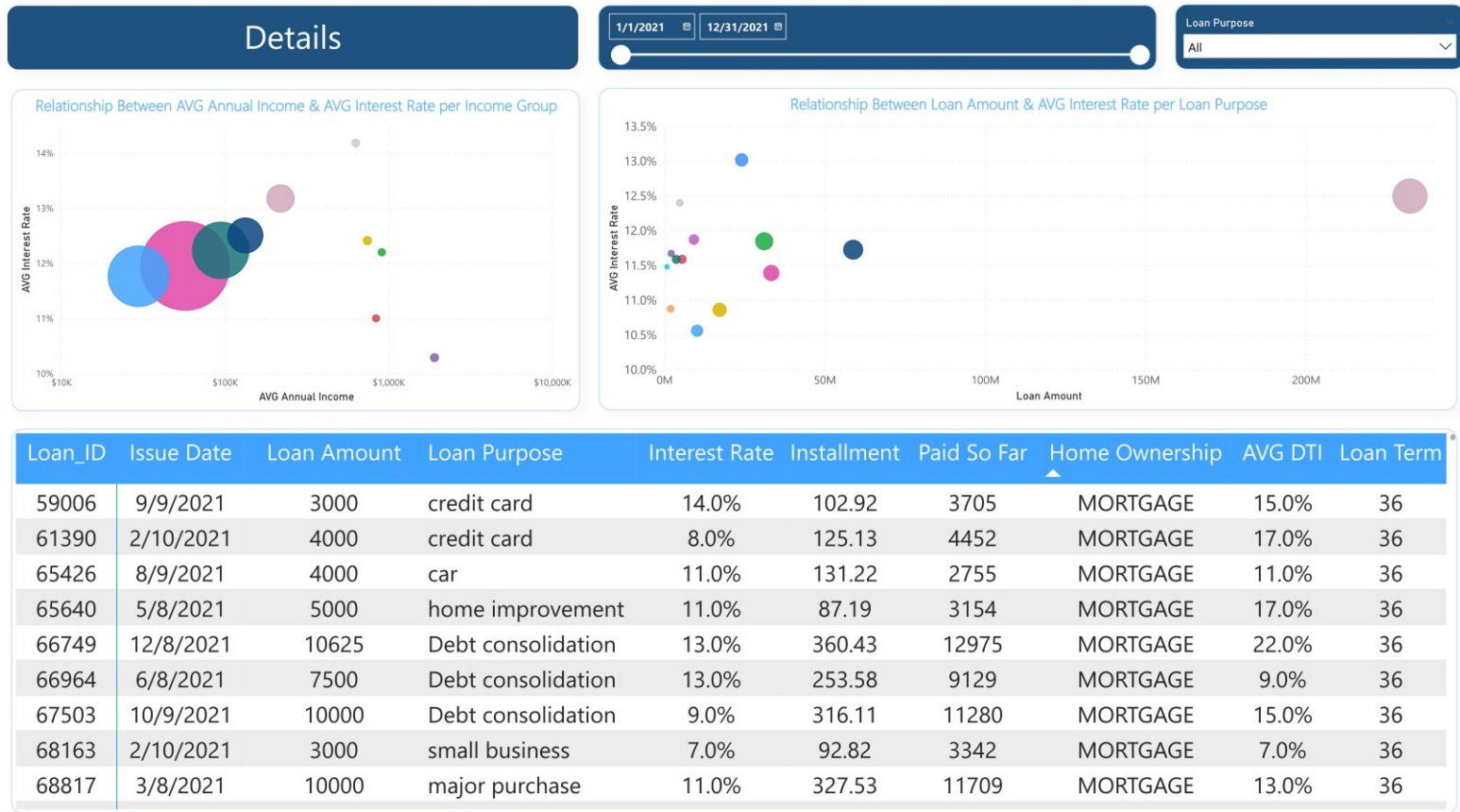
### Number of Loans per Purpose



### Number of Loans per Years of Employment



# Power BI Dashboards







07

**Conclusion**

A decorative graphic on the left side of the slide consisting of two overlapping squares. The top square is a lighter blue, and the bottom square is a darker blue, creating a cross-like shape.

## Conclusion

- End-to-end Big Data ELT pipeline successfully built.
- Automated data ingestion, transformation, and warehousing.
- Power BI dashboards turned raw data into actionable insights.
- Framework can be extended to predictive loan default models.



08

**Future Work**

# Future Work

<b>ML</b>	Integrate ML models for loan default prediction.
<b>Orchestration</b>	Automate pipeline orchestration with Apache Airflow.
<b>Multiple Sources</b>	Add external data sources (credit bureau reports, customer profiles).
<b>Streaming</b>	Enable real-time streaming ingestion (Kafka + Spark Streaming).

# Thanks

Do you have any questions?