Kiva Crowdfunding Dataset Analysis

Poojitha Katta
Department of Applied Data Science
San Jose State University
San Jose, United States
poojitha.katta@sjsu.edu

Purnima Bhukya
Department of Applied Data Science
San Jose State University
San Jose, United States
purnima.bhukya@sjsu.edu

Deepali Zutshi
Department of Applied Data Science
San Jose State University
San Jose, United States
deepali.zutshi@sjsu.edu

Yasaman Emami
Department of Applied Data Science
San Jose State University
San Jose, United States
yasaman.emami@sjsu.edu

Abstract—The main aim of this project is to create a database for the crowdfunding platform-Kiva, to analyze and evaluate factors which influence various aspects of a crowdfunded project and draw conclusions about them.

Keywords—MySQL, Crowdfunding, Database, Kiva

I. Introduction

Crowdfunding is a system that enables individuals or ventures to seek small investments, contributions, or loans from a variety of funders online. Kiva is a crowdfunding platform that offers a new financing channel for small and micro businesses as well as individuals. The aim of the project is to build a database management system for Kiva platform to analyze the factors that influence crowdfunded projects by estimating the welfare level of partners in specific regions, based on shared financial and demographic aspects. Technologies such as MySQL workbench would be used to create the database management system and its schema including relationships based on region, funding, project type, etc. Using Tableau, a powerful visualization tool, we can observe, understand, and draw conclusions on better funded categories, borrowing patterns and regional analysis from the data. The system can be deployed on cloud-based platforms such as AWS using Python for better accessibility and security. This project can help improve access to crowdfunding, assess borrower welfare levels, by analyzing the growth of previously funded projects and benefit Kiva with a better database system to enhance their platform.

II. Dataset

A. Source

The data was obtained from Kaggle, an online community of data scientists and machine learning practitioners. It was made available by Kiva, an online crowdfunding platform, for the "Data Science for Good" Challenge and invited people to help them build a localized model to estimate various metrics in regions where Kiva has active loans.

B. Dataset Description

The dataset contains 4 csv files with 54 attributes total, which includes 30 string, 7 decimal, 4 datetime and 13 other data types. The first table consists of 20 columns, detailing the id, funded amount, loan amount, country code, country, currency, region, etc. Similarly, the second, third and fourth table outlines data snapshot and can be matched to the loan theme regions to get a loan's location and provides details for id, loan theme id, loan theme type, partner id and MPI (Multidimensional Poverty Index). Extracting several insights from the historical micro-loans over a period and correlating the regional averages by gender, sector, or borrowing behavior to estimate the welfare rate is to be followed.

C. Data Cleaning

The data required cleaning and correcting to be processed and stored into the database. Many of the tables had missing data which was replaced by the mode of the data in that column. Attributes with a majority of the data missing were removed from the tables altogether. Many of the tables also contained attributes which were duplicated in the same table as well as across multiple tables. These were removed and the data was normalized to create tables for columns which were repeated across several tables.

```
In [ ]: import pandas as pd
        import matplotlib.pyplot as plt
        import numpy as np
        import seaborn as sns
        import warnings
        warnings.filterwarnings('ignore')

In [ ]: df = pd.read_csv('kiva_loans.csv')

In [ ]: data = pandas.read_csv('kiva_loans.csv', encoding='utf-8', quotechar='"', delimiter=',')

In [ ]: df.info()

In [ ]: df.describe()

In [ ]: null  = df.isnull().sum().sort_values(ascending = False).reset_index()
        null.columns = ['Column', 'Frequency']
        null

In [ ]: ## tags column consists of 10000 null values so remove the column.
        df.drop('tags', axis=1,inplace=True)

In [ ]: ## this column null values are replaced with the mode
        df['funded_time'].mode()

In [ ]: df['borrower_genders'].mode()

In [ ]: df['funded_time'].fillna(df['funded_time'].mode(), inplace = True)

In [ ]: df['borrower_genders'].fillna(df['borrower_genders'].mode(), inplace = True)

In [ ]: df.dropna(inplace = True)

In [ ]: df.isna().sum()

In [ ]: data.to_csv('kiva_mpi_region_locations', encoding='utf-8', index = False)
```

Fig. 1 Jupyter Notebook for Data Cleaning

III. ENTITY RELATIONSHIP DIAGRAM

An entity relationship diagram is a flowchart that explains how the entities (place or object or person) are related to one another within an organization. An ER diagram is essential in the modelling of data of any organization in a graphical manner that is easily understood by the users of the database.

A. Denormalized ER Diagram

- The data as available in its raw form online was not normalized. The tables contained repeated attributes with a few columns missing the data.
- The ER diagram for the raw data was created as follows:

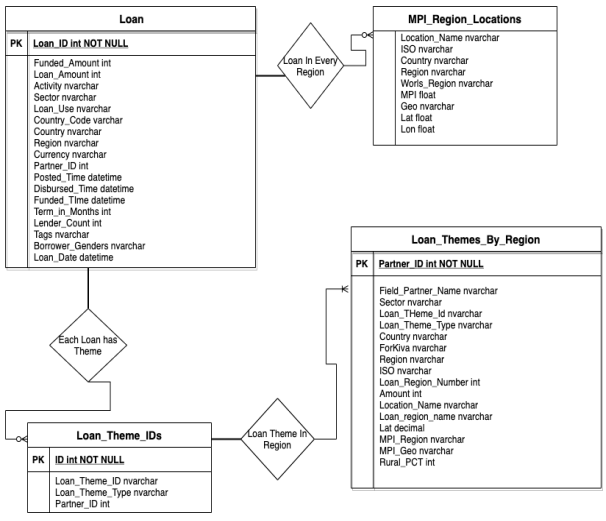

Fig. 2 ER Diagram (1)

B. Normalization

- The process of normalization structures the database to the "normal forms" in order to reduce data redundancy, as well as to improve the integrity of the data.
- The raw dataset contained 4 tables which were further broken down to create 7 tables and improve the overall quality of data.
- The attributes 'ISO ', 'country', 'latitude', 'longitude', 'world_region', 'location' were common in many of the entities. These were extracted to create a separate entity table called 'Region' which was then linked to the table 'Loan' to define the loan details for each region.
- The entity 'Loan' contained the 'borrower genders' attribute which was multi-values and was thus moved to create a separate table called 'Gender Generalization' to specify the genders of borrowers for each of the projects.
- Similarly, an entity 'Category' was created to relate the various project sectors to the loan ID number and how the loan amount was being used.

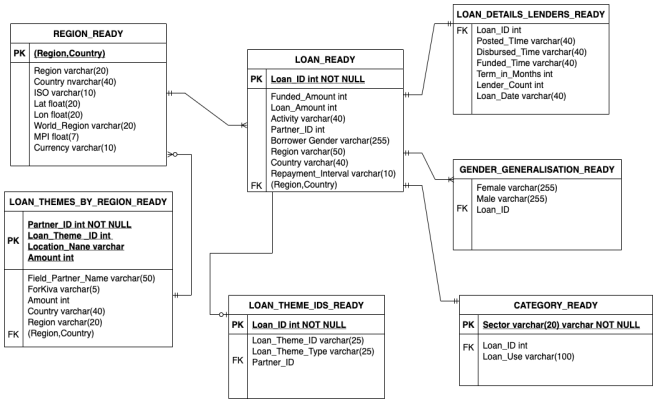

Fig. 3 ER Diagram (2)

C. Primary Keys & Foreign Keys

- Primary keys are attributes in each entity table that are used to uniquely identify each entry in the entity table.
- Foreign keys are attributes in a table that refer to the primary keys of another table. A primary key, foreign key presence defines a relation between the two entity tables.
- In the database that was created, attributes such as 'Loan_ID', 'Loan_Theme_ID', 'Sector' are used to create primary key-foreign key relations among the entities.
- Composite keys are a set of attributes in a table that uniquely define the data for each row of the table. For the entity, 'Loan_Themes_By_Region_Ready' a composite key containing 'Partner_ID', 'Loan_Theme_ID, 'Location_Name', and 'Amount' was created.
- The composite key '(Region,Country)' was defined for the 'Region_Ready' entity table and used as a foregin composite key in the 'Loan_Ready' table.

D. Business Rules

- Kiva is a crowdfunding platform which has lots of borrowers and lenders as members.
- The first business requirement is that a member cannot be resident of Crimea, Cuba, Iran, Syria,

North Korea, or Sudan, which is stored in several attributes like country, country_code and region in the first table in the database and should be checked against this rule.

IV. SQL QUERIES

After the designing and creation of the database and loading the data into the tables, the following SQL queries were performed to update the data and get insights from it.

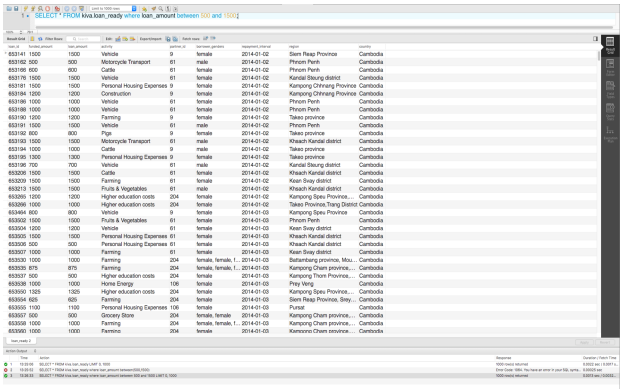

Fig 4. Loan Amount Between (500,1500)

In Fig. 4, all of the attributes like activity, loan amount, borrower gender, region,.. for the records of the table loan_ready that the loan_amount value is between 500 and 1500 are displayed.

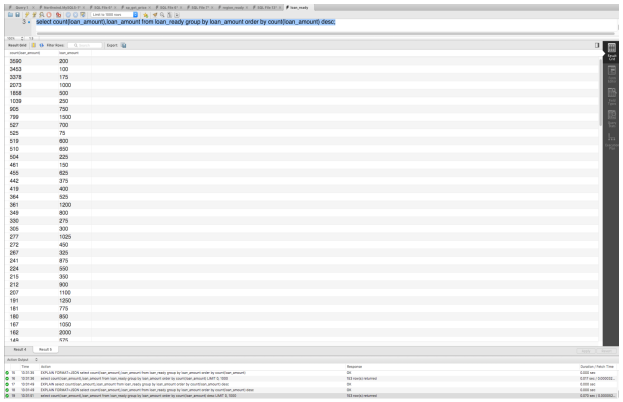

Fig 5. Count of Each Loan

Fig. 5 is displaying how many members applied for the same amount of loan regardless of region and other attributes.

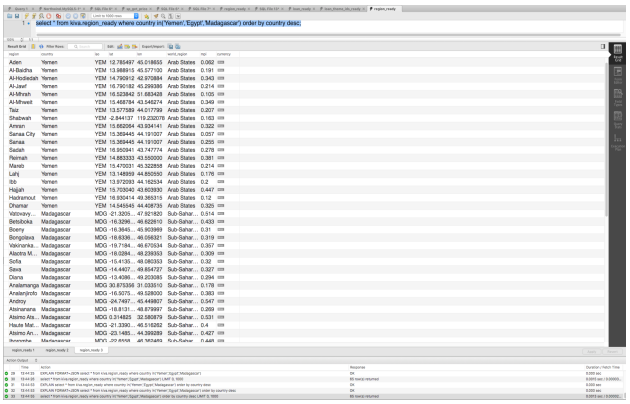

Fig 6. Members in Specific Region

In Fig. 6, all the members who are from specific countries which includes 'Yemen, Egypt or Madagascar' are shown.

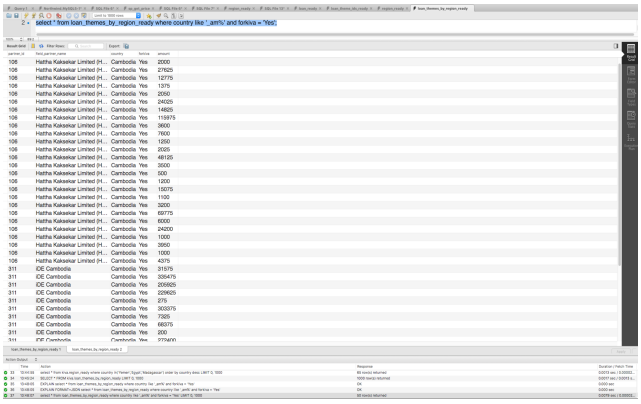

Fig7. Countries Like '_am%'

In Fig. 7 all the members that their country name contains letters 'am' as their second and third letter and they are applying for kiva are displayed.

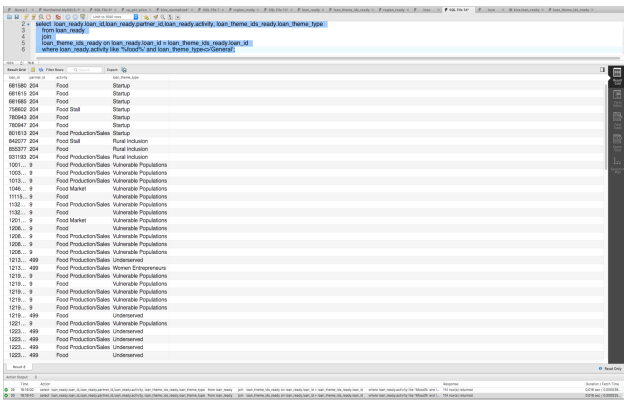

Fig8. Display records from join of two tables to show the loans in food industry specific type (exclude general types)

Fig. 8 is showing the records in which loan_id along with the partner_id their activity and the loan theme which might be different for each partner and the data comes from joining two tables of loan_ready and themes_ids_ready for partners which they are active in food industry and the loan_theme_type is not 'General' they are applying for a specific type.

```
select count(loan_ready.activity) as cnt ,loan_ready.activity from loan_ready
    join loan_lender_details_ready
    on loan_ready.loan_id = loan_lender_details_ready.loan_id
    where loan_lender_details_ready.term_in_months>10
    group by loan_ready.activity
    having cnt>10
    order by cnt;
```

cnt	activity
11	Fuel/Firewood
11	Consumer Goods
11	Health
13	Furniture Making
14	Blacksmith
16	Well digging
16	Sewing
18	Home Products Sales
18	Cereals
20	Beauty Salon
21	Primary/secondary…
21	Crafts
26	Taxi
26	Motorcycle Repair
27	Food Stall
29	Clothing Sales
29	Recycled Materials
32	Transportation
36	Property
36	Wedding Expenses
38	Land Rental
44	Services
47	Tailoring
69	Construction Supplies
71	Beverages
72	Fish Selling
74	Retail
75	Livestock

Fig. 9 Display number of times each activity get funded with terms of bigger 10 months and if they are funded more than 10 times

In Fig. 9, the activities which were funded more than 10 times and they are getting funded with the terms more than 10 months from lenders are shown the result is coming from the source of two tables and filtered for above a threshold.

V. Triggers & Procedures

- Procedures are a set of statements that are stored and called upon to be run multiple times and help reduce network traffic and computations.
- We have created several stored procedures to load the data from original tables into the normalized form tables.

```
DELIMITER $$
CREATE PROCEDURE load_data_into_loan_lender_details_ready()
BEGIN
    INSERT INTO loan_lender_details_ready(posted_time, disbursed_time,
    funded_time, term_in_months, lender_count, loan_date, loan_id)
    select posted_time, disbursed_time, funded_time,
    term_in_months, lender_count, loan_date,loan_id from loan;
END $$
DELIMITER ;

call load_data_into_loan_lender_details_ready();
```

Action Output

	Time	Action	Response	Duration / Fetch Time
71	20:12:59	TRUNCATE `kiva`.`loan_lender_details_ready`	OK	0.000 sec
72	20:13:10	call load_data_into_loan_lender_details_ready()	33437 row(s) affected	0.463 sec

Fig. 10 sp1 load_data_lender_details_ready()

- In Fig. 10, we are inserting data from the loan table into loan_lender_details_ready and calling the stored procedure would pull 33437 tuples of data into our normalized table.
- Fig. 11 is shows the data load procedure for another table which loads 892 records from the original mpi_region_locations table into the region_ready.

```
DELIMITER $$
CREATE PROCEDURE region_ready()
BEGIN
    INSERT INTO region_ready (region,country, iso, lat, lon, world_region,
    mpi) SELECT region,country, iso, lat, lon, world_region,
    mpi FROM mpi_region_locations;
END $$
DELIMITER ;

call region_ready();
```

Action Output

	Time	Action	Response	Duration / Fetch Time
92	20:27:29	CREATE PROCEDUR...	0 row(s) affected	0.067 sec
93	20:27:45	call region_ready()	892 row(s) affected	0.043 sec

Fig. 11 sp2 region_ready()

- In Fig. 12, using stored procedure to insert data into loan_themes_by_region_ready and by calling the loan_themes stored procedure we were able to insert 8876 records for the required attribute from the original loan_them_by_region table.

```
DELIMITER $$
CREATE PROCEDURE loan_themes()
BEGIN
    INSERT INTO loan_themes_by_region_ready (
    partner_id, field_partner_name, country, forkiva,location_name,loan_theme_id,amount)
    SELECT partner_id, field_partner_name, country , forkiva , location_name ,
    loan_theme_id,amount FROM loan_themes_by_region;
END $$
DELIMITER ;
call loan_themes();
```

Action Output

	Time	Action	Response	Duration / Fetch Time
96	20:40:10	CREATE PROCEDUR...	0 row(s) affected	0.103 sec
97	20:40:25	call loan_themes()	8876 row(s) affected	0.181 sec

Fig. 12 sp3 loan_themes()

Similarly, we have stored procedures for all the tables after 3NF so calling those we load data into different tables.

- Triggers are database objects that are activated when specified events occur.

- In our crowdfunding project there is a restriction for some countries that cannot be a member as a lender or borrower in Kiva platform so every time a new

member need to register the location is needed to be check so we make sure that the new member is not in the restricted area, to satisfy this goal we created a trigger to check the region before insertion to the database as described in Fig. 13.

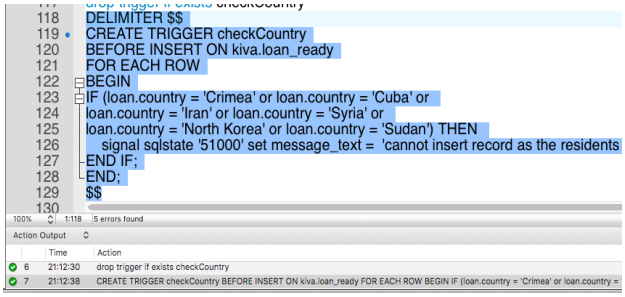

Fig. 13 checkCountry trigger

VI. Data Visualization

The analysis of the Kiva Crowdfunding Dataset gives an insight into the various factors that affect and influence the sustainability of crowdfunded projects. It also reveals the distribution of these projects in the world, gives information about the people involved in the creation of the projects and how the projects are distributed among the different sectors.

- Borrower Gender Distribution

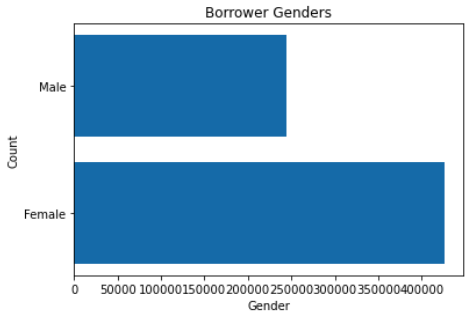

Fig. 14 Borrower gender Distribution

The above bar graph represents the genders of the loan borrowers with the number of women being approximately 40,000 and the men being close to 25,000. This implies that a far greater number of females seek loans from lenders as compared to males. One of the possible reasons for this is that the process of crowdfunding itself attracts more female lenders than the traditional venture capitalists.

- Repayment Method

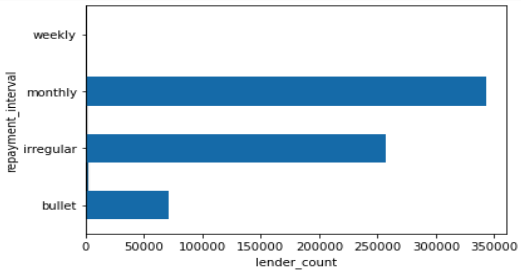

Fig. 15 Repayment Methods

The dataset reveals that there are majorly four types of repayment methods adopted by borrowers for the repayment of the loans, namely, ‘Bullet’, ’Irregular’, ’Monthly’ , and ‘Weekly’. A bullet method involves paying back the entire loan amount in one go at the time of maturity. The method involves gathering large sums and is hence not very popular. On the other hand the methods of monthly and irregular are much more popular whereas the weekly method is not used by anyone at all, perhaps because it involves payments to be made fairly quicker than the other methods.

- Project Sectors

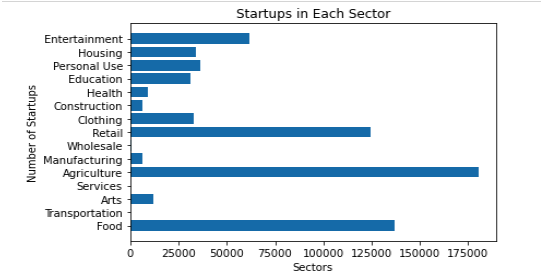

Fig. 16 Project Sectors

On further analysis of the data, it is observed that the projects are categorized into fifteen major sectors such as Food, Arts, Manufacturing, etc with Agriculture having the greatest number of projects. Agriculture involves a large capital investment which can be acquired through crowdfunding rather than by small investment companies. Retail and food sector closely trail agriculture in the number of projects.

- World Region

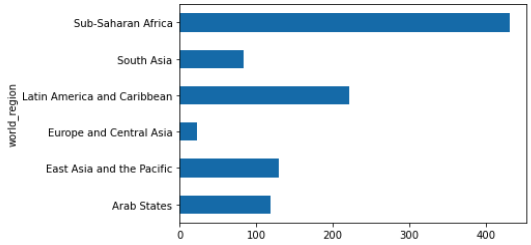

Fig. 17 World Region Distribution

The above bar graph represents the distribution of the projects in different regions of the world. The region of Sub-Saharan Africa sees the maximum number of crowdfunded projects. Because of the lack of alternate sources of financing, crowdfunding has seen a growing popularity here.

- Projects Funded Through Kiva

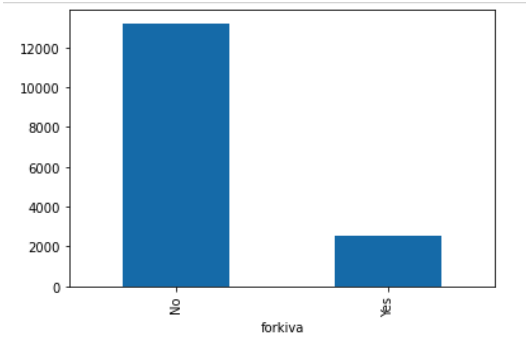

Fig. 18 Kiva Funded Projects

The dataset provided by Kiva contains projects funded not only through their own platform but also by numerous other sources. The graph gives a distribution of the projects that used Kivas' online platform to appeal to people to invest in their idea.

VII. Connectivity to AWS

Amazon Web Services (AWS) is a secure online cloud storage platform that provides various functionalities to businesses such as storage, content delivery, and compute power to help them scale and grow.

Amazon Relational Database Service (RDS) is a service that helps users create, operate on and scale databases on the cloud. RDS not only allows users to create isolated database environments but also provides the necessary security required for the data.

MySQL Workbench when connected to the AWS RDS provides a way for users to deploy their database on the Amazon cloud and manage the creation and manipulation of database entities through Python.

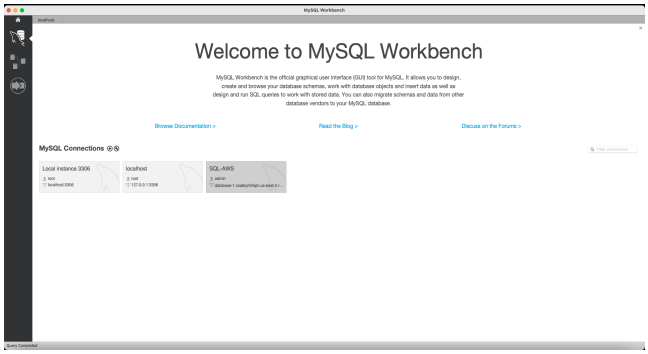

Fig. 19 MySQL-AWS Connection

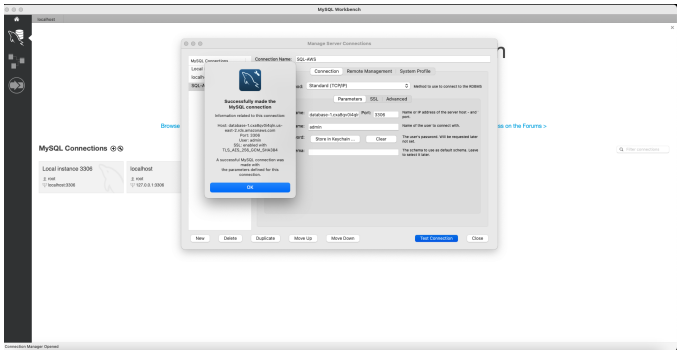

Fig. 20 MySQL-AWS Connection Successful

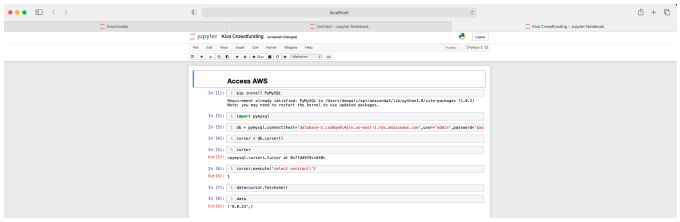

Fig. 21 Python-AWS Connection

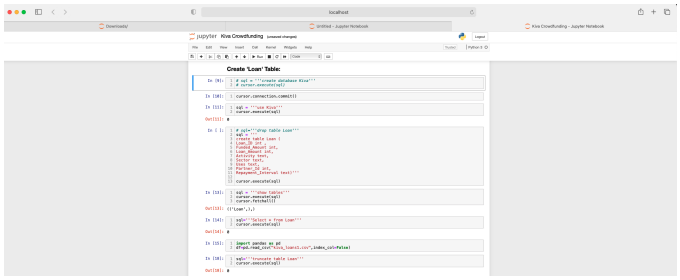

Fig. 22 Database Table Created Using PyMySQL

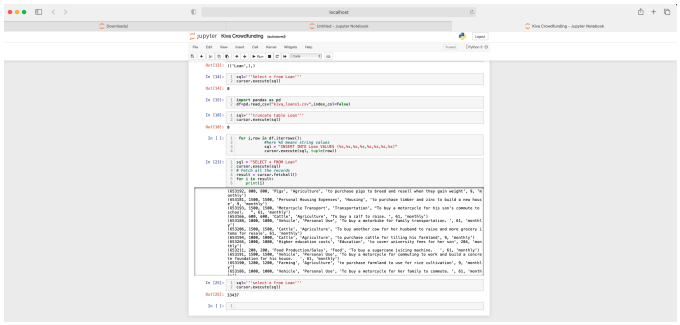

Fig. 23 Querying the data in Python

VIII. SQL Performance Measurement

The SQL performance was measured by analysing the time it took to execute various queries on the database. On average, DML commands require a much longer time to execute as compared to DDL commands. Queries that required data to be selected, updated, altered and joined had a greater execution time. The time taken to run and output the results of triggers and procedures also took more time when compared to the performance of the other data definition commands. The DML and DDL commands are having a very small execution times all the commands are taking less than a second to get executed for example fetching 33437 records into the lender details table was taking only 0.463 second even for pulling data from several tables its very fast for example for the last query performed in the sql query section which returned 50 records taking 0.044 sec and the execution plan is shown in Figure below.

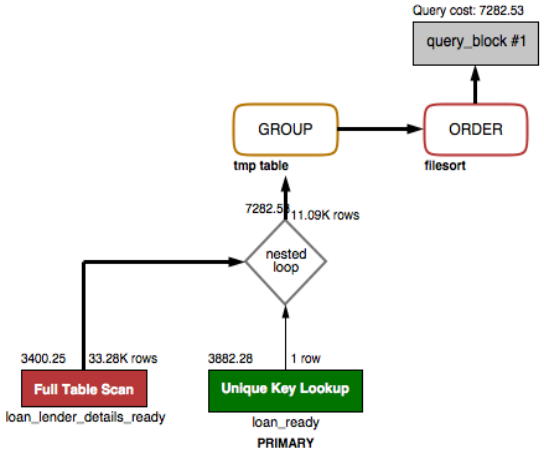

Fig. 24 MySQL Performance Analysis for a sample complex query on Kiva database

Acknowledgment

We would like to express our sincere gratitude to Professor Simon Shim for his guidance that aided in the completion of this project and helped us understand database concepts using practical examples. We would also like to thank Shiva Abhishek Varma Penmetsa and Rushikesh Jagtap for their continued help and support throughout the development of the project.

References

[1] S. Yu, Crowdfunding and regional entrepreneurial investment: an application of the CrowdBerkeley database, Research Policy, Volume 46, Issue 10, 2017, Pages 1723-1737, ISSN 0048-7333

[2] Burtch, G., Ghose, A., Wattal, S., 2014. Cultural Differences and geography as determi- nants of online prosocial lending. MIS Q. 38 (3), 773–794.

[3] Mollick, E., Nanda, R., 2015. Wisdom or Madness? Comparing Crowds with Expert Evaluation in Funding the Arts. Manage. Sci. 62 (6), 1533–1553.Mollick, E.R., 2014. The Dynamics of Crowdfunding: An Exploratory Study. J. Bus.Venturing 29 (January (1)), 1–16.

[4] Smith, Tim. “Crowdfunding.” Investopedia, Investopedia, 13Sept.2021,www.investopedia.com/terms/c/crowdfunding.