

Kiva Loans Analysis Documentation

Impact of Kiva Loans

Background of



Kiva is a non-profit organization based in the United States of America. Kiva allows people to lend money via their website to low-income entrepreneurs and students all over the world, especially in third world countries. Kiva's mission is to expand financial access to help underserved individuals and communities. Since 2005, Kiva has crowd-funded more than 1.6 million loans, totaling over \$1.33 billion, with a repayment rate of 95.8 percent. Over 1.8 million lenders worldwide use the Kiva platform. Lenders do not receive interest on the money they lend. Kiva includes personal stories of each person who applies for a loan so that lenders can connect with borrowers on a human level.

Kiva relies on a network of field partners to administer the loans on the ground. These field partners can be microfinance institutions, social impact businesses, schools or non-profit organizations. Borrowers pay interest on most loans to the field partners, and the field partners are charged small fees by Kiva. Kiva is supported by the above mentioned fees, grants, loans, and donations from its users, corporations, and national institutions.

Multidimensional Poverty Index(MPI)

The Multidimensional poverty index encompasses the various deprivations experienced by poverty stricken persons in their daily lives – such as poor health, lack of education, inadequate living standards, disempowerment, poor quality of work, the threat of violence, and living in areas that are environmentally hazardous, among others. In this Analysis, the deprivations that are taken into consideration are Health which includes child mortality rates and lack of nutrition, Education, which includes numbers of children whom are not able to attend school and Living Conditions which encompasses homes not having electricity, proper sanitation, acceptable clean drinking water, homes that use dirt, sand or dung as their flooring, the availability of cooking fuel and asset ownership. The MPI value summarizes information on multiple deprivations into a single number and it is calculated by multiplying the poverty headcount by the intensity of poverty based on all the aforementioned deprivations. A household is multidimensionally poor if it is deprived in one third or more of the weighted indicators.

Problem Definition

This Analysis attempts to explore the extent of impact Kiva loans have on decreasing the multidimensional poverty index of third world countries and answer the following the questions :

1. Do the loans go to the poorest and the most deprived ?
2. Do most of the loans go to the countries and regions with the highest MPIs ?
3. Are the loans in these regions addressing the appropriate deprivation faced by the region ?

If not, how can the process of matching loans to needs be improved ?

Source and Description of Datasets

1. Kiva Dataset

<https://www.kaggle.com/kiva/data-science-for-good-kiva-crowdfunding>

kiva_loans.csv – records of individual loans

kiva_kiva_mpi_region_locations.csv – records of region of individual loans and the partner for the loan

loan_themes_by_region.csv – MPI of regions and countries

2. MPI Dataset

<https://www.kaggle.com/ophi/mpi>

MPI_national.csv- MPI of countries

3. Contribution of Deprivations Dataset

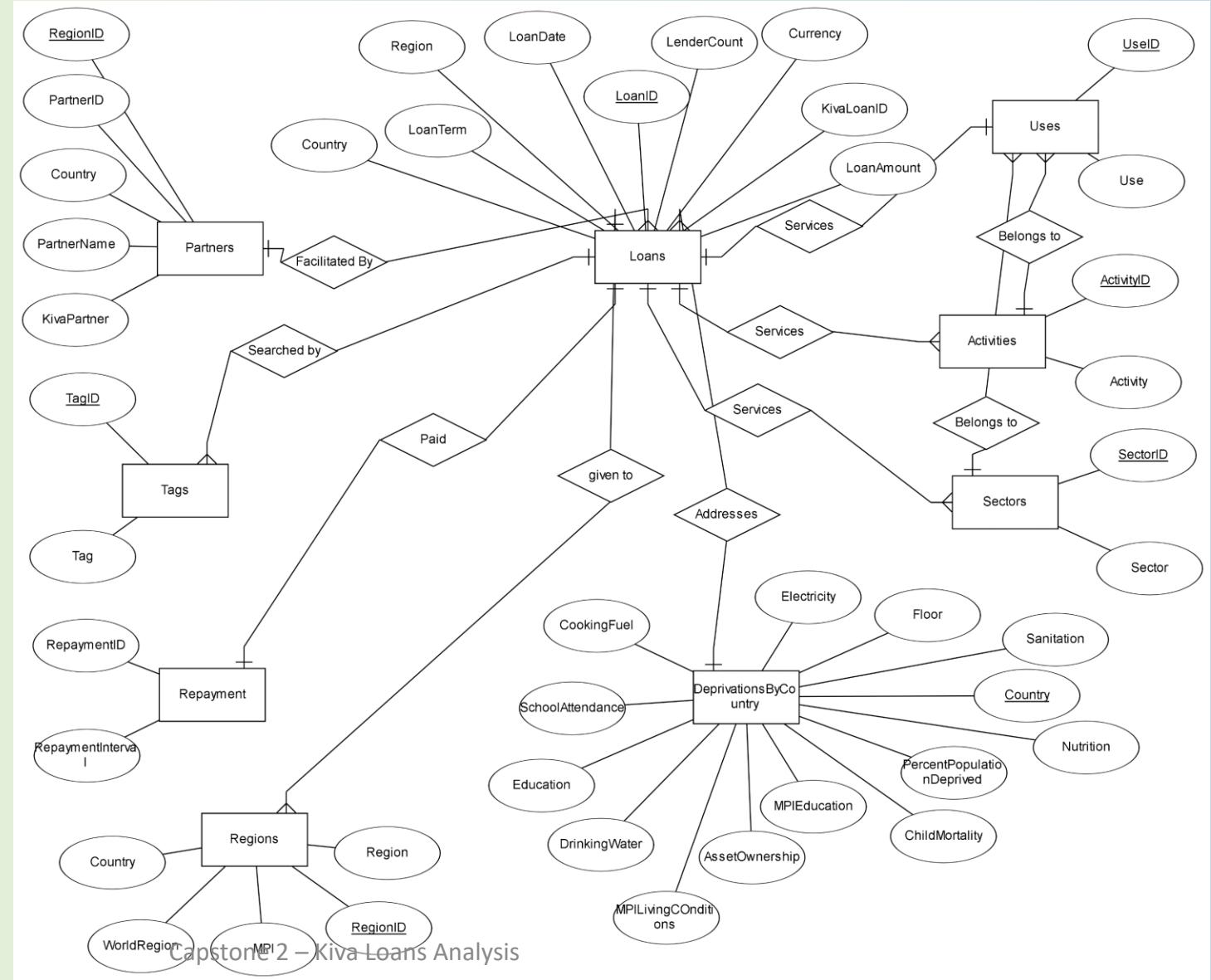
https://www.kaggle.com/lucian18/mpi-on-regions?select=Tables_5.3_Contribution_of_Deprivations.csv

Tables_5.3_Contribution_of_Deprivations.csv - individual records of regions and countries with the percentage of total population deprived, the percentage of collective deprivations(Health, Education and Living Conditions) that contributes to the overall MPI of the region and the percentage of populations deprived in the individual deprivations

Entity Relationship Diagram

Entities :

1. Loans
2. Regions
3. Partners
4. DeprivationsByCountry
5. Sectors
6. Activities
7. Uses
8. Tags
9. Repayment

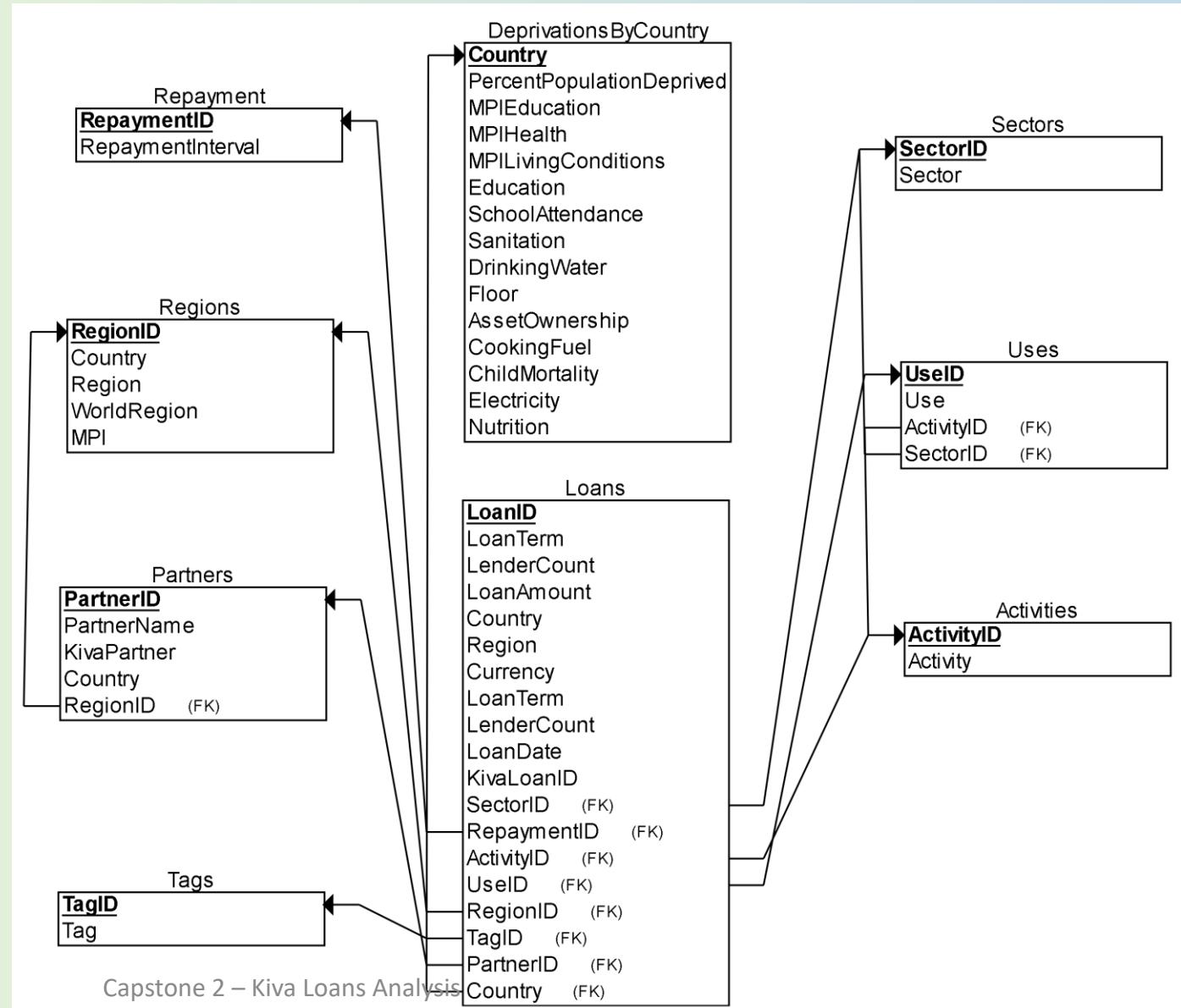


Database Schema

Tables in Database

1. dbo.Loans
2. dbo.DeprivationsByCountry
3. dbo.Partners
4. dbo.Regions
5. dbo.Sectors
6. dbo.Activities
7. dbo.Uses
8. dbo.Repayment
9. dbo.Tags
10. dbo.MPI_National*

*omitted from schema as this table was used
for data cleaning and not for analysis



Normalising the Dataset

Example: Normalising the kiva_loans table to have an ActivityID to reference Activities tables. This ensures that the activity field will not have to be duplicated within the Loans table and only the ActivityID will be used to reference the activity in the Activities table

Step 1 : Load the loans table to Power Query

The screenshot shows the Microsoft Power Query Editor interface. On the left, there's a sidebar labeled 'Queries' with a list of loaded queries. The main area displays a table titled 'kiva_loans'. The table has columns: 'id', 'funded_amount', 'loan_amount', 'activity', 'sector', and 'use'. The 'activity' column contains values like 'Fruits & Vegetables', 'Rickshaw', 'Transportation', etc. The 'sector' column contains values like 'Food', 'Transportation', 'Arts', etc. The 'use' column contains descriptive text about the loan purpose. The top ribbon bar shows the 'File', 'Home', 'Transform', 'Add Column', and 'View' tabs. The 'Transform' tab is selected. The 'Applied Steps' pane on the right shows the steps taken: 'Promoted Headers' and 'Changed Type'. The 'Properties' pane shows the query is named 'kiva_loans'.

id	funded_amount	loan_amount	activity	sector	use
653051	300	300	Fruits & Vegetables	Food	To buy seasonal, fresh fruits to sell.
653053	575	575	Rickshaw	Transportation	to repair and maintain the auto rickshaw u
653068	150	150	Transportation	Transportation	To repair their old cycle-van and buy ano
653063	200	200	Embroidery	Arts	to purchase an embroidery machine and a
653084	400	400	Milk Sales	Food	to purchase one buffalo.
1080148	250	250	Services	Services	purchase leather for my business using ksh
653067	200	200	Dairy	Agriculture	To purchase a dairy cow and start a milk pr
653078	400	400	Beauty Salon	Services	to buy more hair and skin care products.
653082	475	475	Manufacturing	Manufacturing	to purchase leather, plastic soles and heels
653048	625	625	Food Production/Sales	Food	to buy a stall, gram flour, ketchup, and coa
653060	200	200	Rickshaw	Transportation	to cover the cost of repairing rickshaw
653088	400	400	Wholesale	Wholesale	to purchase biscuits, sweets and juices in b
653089	400	400	General Store	Retail	to buy stock of rice, sugar and flour .
653062	400	400	Clothing Sales	Clothing	to purchase variety of winter clothes to sel
653075	225	225	Poultry	Agriculture	to expand her existing poultry farm busine
653054	300	300	Rickshaw	Transportation	to buy a three-wheeled rickshaw.
653091	400	400	General Store	Retail	to buy packs of salts, biscuits and beverage
653052	875	875	Tailoring	Services	To buy a sewing machine, lace, zippers and
653066	250	250	Sewing	Services	to purchase a sewing machine.
653080	475	475	Beauty Salon	Services	to buy more cosmetics products for her be
653065	250	250	Bakery	Food	to buy ingredients to make bakery product
653055	350	350	Restaurant	Food	to purchase vegetables, chicken, and oil to
653050	575	575	Clothing Sales	Clothing	To buy winter clothing to sell
653079	350	350	Embroidery	Arts	to buy reels of threads in different colors a
653061	250	250	Food Stall	Food	to purchase a variety of needed food item
653074	250	250	Farming	Agriculture	to purchase potato seeds and fertilizers fo
653069	250	250	Food	Food	to purchase stones for starting a business !

Normalising the Dataset

Example: Normalising the kiva_loans table to have an ActivityID to reference Activities tables. This ensures that the activity field will not have to be duplicated within the Loans table and only the ActivityID will be used to reference the activity in the Activities table

Step 2 : Create a new table with unique values from the activity field

The screenshot shows the Power Query Editor interface with the following details:

- File Bar:** File, Home, Transform, Add Column, View.
- Toolbars:** Close & Load, Refresh, Advanced Editor, Properties, Manage, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Split Column, Group By, Replace Values, Data Type: Text, Use First Row as Headers, Merge Queries, Append Queries, Combine Files, Manage Parameters, Data Source Settings, New Source, Recent Sources, Enter Data, New Query.
- Queries List:** Shows the current query "kiva_loans".
- Table View:** Displays the "Promoted Headers" table with columns: id, funded_amount, loan_amount, activity, sector, and use. The "activity" column is selected for grouping.
- Group By Dialog:** A modal dialog titled "Group By" is open, showing the selection of "activity" as the column to group by and "Count Rows" as the operation.
- Query Settings Panel:** Shows the "kiva_loans" query settings with the "Name" set to "kiva_loans".
- Applied Steps Panel:** Lists the applied steps: "Source" (Promoted Headers) and "Changed Type".
- Bottom Status Bar:** Capstone 2 – Kiva Loans Analysis.
- Page Footer:** 9/6/2021, 9.

Normalising the Dataset

Example: Normalising the kiva_loans table to have an ActivityID to reference Activities tables. This ensures that the activity field will not have to be duplicated within the Loans table and only the ActivityID will be used to reference the activity in the Activities table

Step 3 : Assign an ActivityID to each activity

The screenshot shows two side-by-side views of the Power Query Editor interface, illustrating the transformation of the 'activity' column.

Left View: The 'activity' column has been grouped into a single row, resulting in a table with 2 columns and 289 rows. The formula bar shows the query: `= Table.Group(#"Changed Type", {"activity"})`. The table contains activity names and their counts.

Activity	Count
Fruits & Vegetables	16610
Rickshaw	2878
Transportation	4121
Embroidery	2380
Milk Sales	1771
Services	9807
Dairy	7914
Beauty Salon	6611
Manufacturing	2643
Food Production/Sales	28106
Wholesale	357
General Store	64729
Clothing Sales	22339
Poultry	9783
Tailoring	9657
Sewing	8414
Bakery	3620
Restaurant	4981
Food Stall	8905
Farming	72955
Construction Supplies	2003
Personal Products Sales	3866
Home Products Sales	3817
Natural Medicines	814
Fish Selling	13060
Education provider	4401
Shoe Sales	2760
Machinery/Rental	131

Right View: The 'activity' column has been removed, and a new 'ActivityID' column has been added. The formula bar shows the query: `= Table.RemoveColumns(#"Reordered Columns")`. The table now has 2 columns and 289 rows, with the 'ActivityID' column containing numerical values corresponding to the activities listed in the previous step.

ActivityID	Activity
1	Fruits & Vegetables
2	Rickshaw
3	Transportation
4	Embroidery
5	Milk Sales
6	Services
7	Dairy
8	Beauty Salon
9	Manufacturing
10	Food Production/Sales
11	Wholesale
12	General Store
13	Clothing Sales
14	Poultry
15	Tailoring
16	Sewing
17	Bakery
18	Restaurant
19	Food Stall
20	Farming
21	Construction Supplies
22	Personal Products Sales
23	Home Products Sales
24	Natural Medicines
25	Fish Selling
26	Education provider
27	Shoe Sales
28	Machinery/Rental

Normalising the Dataset

Example: Normalising the kiva_loans table to have an ActivityID to reference Activities tables. This ensures that the activity field will not have to be duplicated within the Loans table and only the ActivityID will be used to reference the activity in the Activities table

Step 4 : Use INDEX and MATCH functions in Excel to replace the activity fields in the Loans table with its respective ActivityIDs

The image shows two Microsoft Excel spreadsheets side-by-side.

Left Spreadsheet (kiva_loans):

- Columns:** A through S.
- Rows:** 1 to 36.
- Data:** Contains a dataset of 36 rows of loan information. Each row includes details like the loan ID (e.g., 653051), funded amount (e.g., 300), loan amount (e.g., 300), activity (e.g., Fruits & Vegetables, Food), sector (e.g., Transport), use (e.g., To buy sea PK), country (e.g., Pakistan), region (e.g., Lahore), currency (e.g., PKR), partner ID (e.g., 247), posted time (e.g., 2014-01-01), disbursed time (e.g., 2013-12-17), funded time (e.g., 2014-01-02), term (e.g., 12), lender count (e.g., 12), tags (e.g., female, irregular), borrower gender (e.g., female), repayment date (e.g., 1/1), and repayment status (e.g., 1/1).

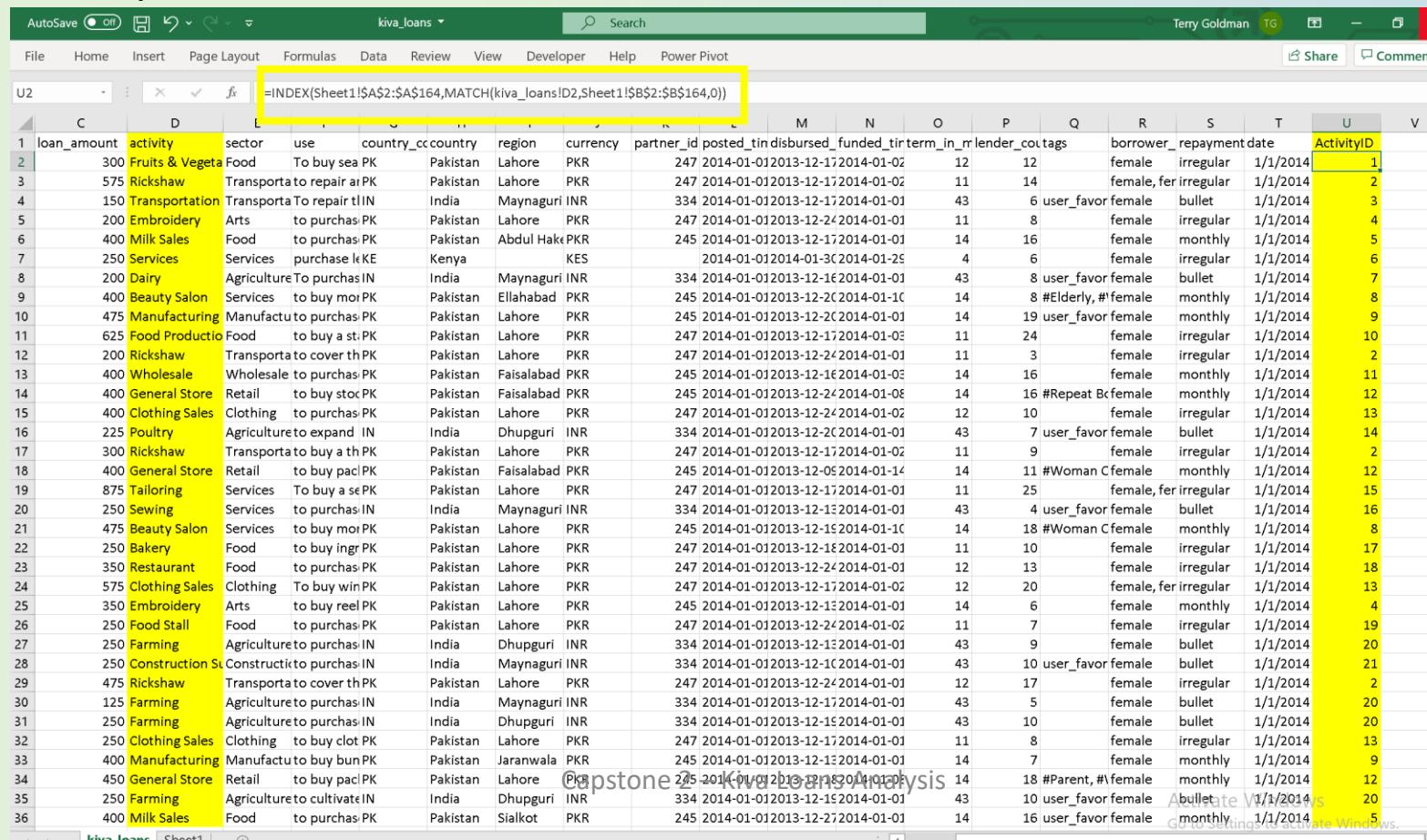
Right Spreadsheet (Activities):

- Columns:** A through D.
- Rows:** 1 to 36.
- Data:** Contains a list of 36 activities, each assigned an ActivityID (e.g., 1 to 36) and a corresponding activity name (e.g., Fruits & Vegetables, Rickshaw, Transportation, Embroidery, Milk Sales, Services, Dairy, Beauty Salon, Manufacturing, Food Production/Sales, Wholesale, General Store, Clothing Sales, Poultry, Rickshaw, Tailoring, Sewing, Bakery, Restaurant, Food Stall, Farming, Construction Supplies, Personal Products Sales, Home Products Sales, Natural Medicines, Fish Selling, Education provider, Shoe Sales, Machinery Rental, Butcher Shop, Pigs, Personal Expenses, Food Market, Cosmetics Sales, Personal Housing Expenses, Retail).

Normalising the Dataset

Example: Normalising the kiva_loans table to have an ActivityID to reference Activities tables. This ensures that the activity field will not have to be duplicated within the Loans table and only the ActivityID will be used to reference the activity in the Activities table

Step 4 : Use INDEX and MATCH functions in Excel to replace the activity fields in the Loans table with its respective ActivityIDs



loan_amount	activity	sector	use	country_ccountry	region	currency	partner_id	posted_tin	disbursed_funded_tir	term_in_mr	lender_col_tags	borrower_repayment date	ActivityID
300	Fruits & Vegeta	Food	To buy sea	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-17 2014-01-02	12	12	female irregular	1/1/2014 1
575	Rickshaw	Transporta	to repair a	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-17 2014-01-02	11	14	female, fer irregular	1/1/2014 2
150	Transportation	Transporta	To repair t	IN	India	Maynaguri	INR	334 2014-01-01	2013-12-17 2014-01-01	43	6	user_favor female bullet	1/1/2014 3
200	Embroidery	Arts	to purchas	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-24 2014-01-01	11	8	female irregular	1/1/2014 4
400	Milk Sales	Food	to purchas	PK	Pakistan	Abdul Hak	PKR	245 2014-01-01	2013-12-17 2014-01-01	14	16	female monthly	1/1/2014 5
250	Services	Services	purchase k	KE	Kenya	KES		2014-01-01	2014-01-32 2014-01-25	4	6	female irregular	1/1/2014 6
200	Dairy	Agriculture	To purchas	IN	India	Maynaguri	INR	334 2014-01-01	2013-12-1f 2014-01-01	43	8	user_favor female bullet	1/1/2014 7
400	Beauty Salon	Services	to buy mor	PK	Pakistan	Ellahabad	PKR	245 2014-01-01	2013-12-22 2014-01-1C	14	8	#Elderly, #female monthly	1/1/2014 8
475	Manufacturing	Manufactu	to purchas	PK	Pakistan	Lahore	PKR	245 2014-01-01	2013-12-2C 2014-01-01	14	19	user_favor female monthly	1/1/2014 9
625	Food Productio	Food	to buy a st	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-17 2014-01-03	11	24	female irregular	1/1/2014 10
200	Rickshaw	Transporta	to cover th	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-24 2014-01-01	11	3	female irregular	1/1/2014 2
400	Wholesale	Wholesale	to purchas	PK	Pakistan	Faisalabad	PKR	245 2014-01-01	2013-12-1f 2014-01-03	14	16	female monthly	1/1/2014 11
400	General Store	Retail	to buy stoc	PK	Pakistan	Faisalabad	PKR	245 2014-01-01	2013-12-24 2014-01-08	14	16	#Repeat Bc female monthly	1/1/2014 12
400	Clothing Sales	Clothing	to purchas	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-24 2014-01-02	12	10	female irregular	1/1/2014 13
225	Poultry	Agriculture	to expand IN	IN	India	Dhupguri	INR	334 2014-01-01	2013-12-2C 2014-01-01	43	7	user_favor female bullet	1/1/2014 14
300	Rickshaw	Transporta	to buy a th	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-17 2014-01-02	11	9	female irregular	1/1/2014 2
400	General Store	Retail	to buy paci	PK	Pakistan	Faisalabad	PKR	245 2014-01-01	2013-12-05 2014-01-14	14	11	#Woman C female monthly	1/1/2014 12
875	Tailoring	Services	To buy a se	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-17 2014-01-01	11	25	female, fer irregular	1/1/2014 15
250	Sewing	Services	to purchas	IN	India	Maynaguri	INR	334 2014-01-01	2013-12-1f 2014-01-01	43	4	user_favor female bullet	1/1/2014 16
475	Beauty Salon	Services	to buy mor	PK	Pakistan	Lahore	PKR	245 2014-01-01	2013-12-15 2014-01-1C	14	18	#Woman Cf female monthly	1/1/2014 8
250	Bakery	Food	to buy ingr	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-1f 2014-01-01	11	10	female irregular	1/1/2014 17
350	Restaurant	Food	to purchas	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-24 2014-01-01	12	13	female irregular	1/1/2014 18
575	Clothing Sales	Clothing	To buy wi	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-17 2014-01-02	12	20	female, fer irregular	1/1/2014 13
350	Embroidery	Arts	to buy reel	PK	Pakistan	Lahore	PKR	245 2014-01-01	2013-12-1f 2014-01-01	14	6	female monthly	1/1/2014 4
250	Food Stall	Food	to purchas	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-24 2014-01-02	11	7	female irregular	1/1/2014 19
250	Farming	Agriculture	to purchas	IN	India	Dhupguri	INR	334 2014-01-01	2013-12-1f 2014-01-01	43	9	female bullet	1/1/2014 20
250	Construction St	Construct	to purchas	IN	India	Maynaguri	INR	334 2014-01-01	2013-12-1f 2014-01-01	43	10	user_favor female bullet	1/1/2014 21
475	Rickshaw	Transporta	to cover th	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-24 2014-01-01	12	17	female irregular	1/1/2014 2
125	Farming	Agriculture	to purchas	IN	India	Maynaguri	INR	334 2014-01-01	2013-12-17 2014-01-01	43	5	female bullet	1/1/2014 20
250	Farming	Agriculture	to purchas	IN	India	Dhupguri	INR	334 2014-01-01	2013-12-1f 2014-01-01	43	10	female bullet	1/1/2014 20
250	Clothing Sales	Clothing	to buy clot	PK	Pakistan	Lahore	PKR	247 2014-01-01	2013-12-17 2014-01-01	11	8	female irregular	1/1/2014 13
400	Manufacturing	Manufactu	to buy bun	PK	Pakistan	Jaranwala	PKR	245 2014-01-01	2013-12-1f 2014-01-01	14	7	female monthly	1/1/2014 9
450	General Store	Retail	to buy paci	PK	Pakistan	Lahore	PKR	245 2014-01-01	2013-12-1f 2014-01-01	14	18	#Parent, #female monthly	1/1/2014 12
250	Farming	Agriculture	to cultivate	IN	India	Dhupguri	INR	334 2014-01-01	2013-12-15 2014-01-01	43	10	user_favor female bullet	1/1/2014 20
400	Milk Sales	Food	to purchas	PK	Pakistan	Sialkot	PKR	245 2014-01-01	2013-12-27 2014-01-01	14	16	user_favor female monthly	1/1/2014 5

Importing Data to Database

Example: Importing the Loans Table to the Database

The diagram illustrates the steps for importing data into a database:

- Import Flat File 'KivaCapstone2' (Preview Data):** This window shows a preview of the data from the 'Loans' table. The columns listed are LoanID, LoanAmo, ActivityID, SectorID, UselD, RegionID, Country, region, Currency, and L. The data shows various loans from different countries like Pakistan, India, and Kenya.
- Modify Columns:** This window displays the schema generated from the input file. It lists 15 columns with their data types: LoanID (int), LoanAmount (int), ActivityID (int), SectorID (int), UselD (int), RegionID (int), Country (varchar(100)), Region (varchar(100)), Currency (varchar(3)), LoanTerm (int), LenderCount (int), TagID (int), RepaymentID (int), and LoanDate (datetime2). Primary keys and nullability are also specified.
- Microsoft SQL Server Management Studio (Object Explorer):** This window shows the 'KivaCapstone2' database structure. Under the 'Tables' node, the 'Loans' table is listed, along with other tables like 'Activities', 'DeprivationsByCountry', 'MPI_National', 'Partners', 'Regions', 'Repayment', 'Sectors', 'Tags', 'Uses', and 'Views'.

Cleaning Dataset

Example 1: Removing NULL Regions values and NULL MPI values from dbo.Regions

SQLQuery8.sql - DE...K4PBESE\Terry (59)* ↗ X

```
DELETE FROM Regions WHERE Region IS NULL and MPI IS NULL
```

33	33	Afghanistan	Wardak	South Asia	0.34	34.3913494	68.2385339
34	34	Afghanistan	Zabul	South Asia	0.175	32.1918782	67.1894488
35	35	Albania	NULL	Europe an...	NU...	NULL	NULL
36	36	Armenia	NULL	Europe an...	NU...	NULL	NULL
37	37	Azerbaijan	NULL	Europe an...	NU...	NULL	NULL
38	38	Burundi	Bujumbur...	Sub-Saharan...	0.164	-3.4146958	29.3598782
39	39	Burundi	Nord	Sub-Saharan...	0.501	-3.373056	29.918886
40	40	Burundi	Centre-Est	Sub-Saharan...	0.473	9.0630743	7.464915
41	41	Burundi	Ouest	Sub-Saharan...	0.499	-3.373056	29.918886
42	42	Burundi	Sud	Sub-Saharan...	0.422	14.6090537	121.0222565

Cleaning Dataset

Example 2: For some countries the Regions field was NULL, so those fields were replaced by the Country value

	LoanID	LoanAmount	ActivityID	SectorID	UserID	RegionID	Country	Region	Currency	LoanTerm	LenderCount	TagID	RepaymentID	LoanDate	
61...	671186	25	74	5	#N/A	#N/A	Kenya	NULL	KES	13	0	0	2	2017-07-24 00:00:00.0000000	
61...	671187	25	74	5	672...	#N/A	Parag...	Concepción	USD	13	1	0	2	2017-07-24 00:00:00.0000000	
61...	671188	25	74	5	672...	#N/A	Kenya	NULL	KES	13	0	0	2	2017-07-24 00:00:00.0000000	
61...	671189	250	74	5	658...	#N/A	Ghana	Dansoman	GHS	13	0	0	2	2017-07-24 00:00:00.0000000	
61...	671190	25	74	5	672...	#N/A	Kenya	NULL	KES	13	0	0	2	2017-07-25 00:00:00.0000000	
61...	671191	25	74	5	672...	#N/A	Kenya	NULL	KES	13	0	0	2	2017-07-25 00:00:00.0000000	
61...	671194	125	74	5	672...	#N/A	Mexico	Iztacalco	MXN	13	0	0	2	2017-07-25 00:00:00.0000000	
61...	671195	25	74	5	672...	#N/A	Kenya	NULL	KES	13	0	0	2	2017-07-25 00:00:00.0000000	
61...	671196	50	74	5	672...	#N/A	Ghana	Dansoman	GHS	13	0	0	2	2017-07-25 00:00:00.0000000	
61...	671197	250	74	5	658...	#N/A	Ghana	Dansoman	GHS	13	0	0	2	2017-07-25 00:00:00.0000000	
61...	671198	25	74	5	672...	#N/A	Kenya	NULL	KES	13	0	0	2	2017-07-25 00:00:00.0000000	
61...	671199	75	74	5	672...	#N/A	Mexico	Iztacalco	MXN	13	0	0	2	2017-07-25 00:00:00.0000000	
61...	671200	25	74	5	672...	#N/A	Parag...	Concepción	USD	13	0	0	2	2017-07-25 00:00:00.0000000	
61...	671201	25	74	5	672...	#N/A	Parag...	Concepción	USD	13	0	0	2	2017-07-25 00:00:00.0000000	
61...	671202	25	74	5	672...	#N/A	Kenya	NULL	KES	13	1	0	2	2017-07-25 00:00:00.0000000	
61...	671203	25	129	15	#N/A	#N/A	Kenya	NULL	KES	13	0	0	2	2017-07-26 00:00:00.0000000	
61...	671204	25	74	5	672...	#N/A	Kenya	NULL	KES	13	0	0	2	2017-07-26 00:00:00.0000000	
61...	671205	25	74	5	672...	#N/A	Kenya	NULL	KES	13	0	0	2	2017-07-26 00:00:00.0000000	

✓ Query executed successfully.

| DESKTOP-K4PBESE (15.0 RTM) | DESKTOP-K4PBESE\Terry ... | KivaCapstone2 00:00:0

Go to Settings to activate W
Activate Windows

Cleaning Dataset

Example 2: For some countries the Regions field was NULL, so those fields were replaced by the Country value

UPDATE Loans SET Region = Country WHERE Region IS NULL

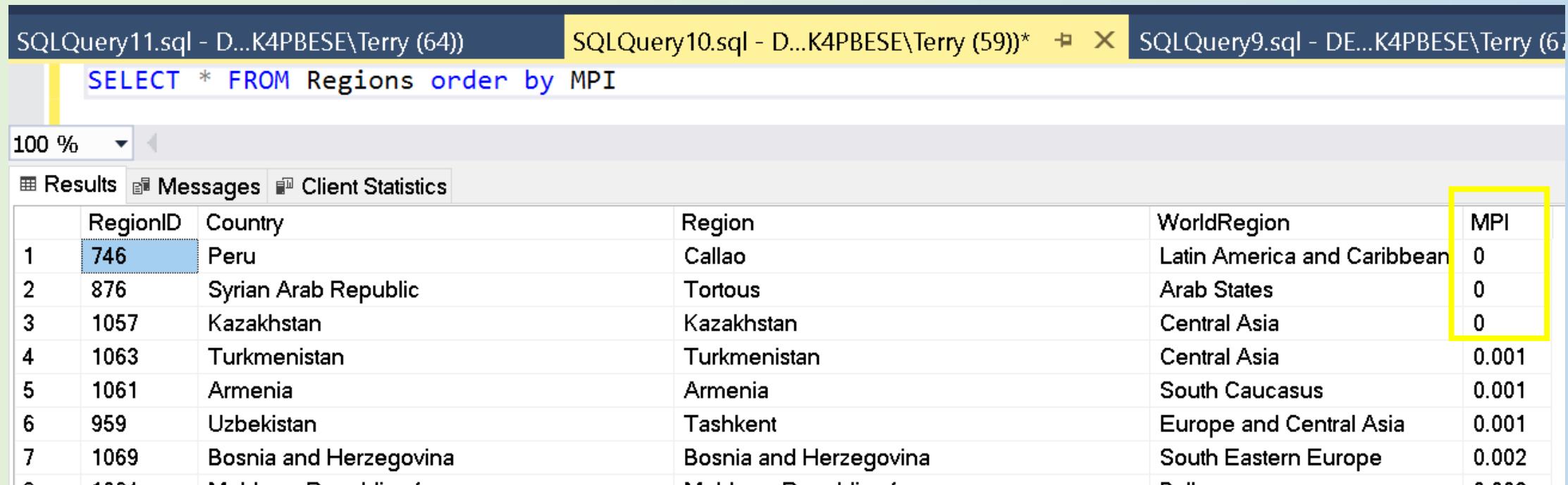


61...	671186	25	74	5	#N/A	#N/A	Kenya	Kenya	KES	13	0	0	2	2017-07-24 00:00:00.0000000
61...	671187	25	74	5	672...	#N/A	Parag...	Concepción	USD	13	1	0	2	2017-07-24 00:00:00.0000000
61...	671188	25	74	5	672...	#N/A	Kenya	Kenya	KES	13	0	0	2	2017-07-24 00:00:00.0000000
61...	671189	250	74	5	658...	#N/A	Ghana	Dansom...	GHS	13	0	0	2	2017-07-24 00:00:00.0000000
61...	671190	25	74	5	672...	#N/A	Kenya	Kenya	KES	13	0	0	2	2017-07-25 00:00:00.0000000
61...	671191	25	74	5	672...	#N/A	Kenya	Kenya	KES	13	0	0	2	2017-07-25 00:00:00.0000000
61...	671194	125	74	5	672...	#N/A	Mexico	Iztacalco	MXN	13	0	0	2	2017-07-25 00:00:00.0000000
61...	671195	25	74	5	672...	#N/A	Kenya	Kenya	KES	13	0	0	2	2017-07-25 00:00:00.0000000
61...	671196	50	74	5	672...	#N/A	Ghana	Dansom...	GHS	13	0	0	2	2017-07-25 00:00:00.0000000
61...	671197	250	74	5	658...	#N/A	Ghana	Dansom...	GHS	13	0	0	2	2017-07-25 00:00:00.0000000
61...	671198	25	74	5	672...	#N/A	Kenya	Kenya	KES	13	0	0	2	2017-07-25 00:00:00.0000000
61...	671199	75	74	5	672...	#N/A	Mexico	Iztacalco	MXN	13	0	0	2	2017-07-25 00:00:00.0000000
61...	671200	25	74	5	672...	#N/A	Parag...	Concepción	USD	13	0	0	2	2017-07-25 00:00:00.0000000
61...	671201	25	74	5	672...	#N/A	Parag...	Concepción	USD	13	0	0	2	2017-07-25 00:00:00.0000000
61...	671202	25	74	5	672...	#N/A	Kenya	Kenya	KES	13	1	0	2	2017-07-25 00:00:00.0000000
61...	671203	25	129	15	#N/A	#N/A	Kenya	Kenya	KES	13	0	0	2	2017-07-26 00:00:00.0000000
61...	671204	25	74	5	672...	#N/A	Kenya	Kenya	KES	13	0	0	2	2017-07-26 00:00:00.0000000
61...	671205	25	74	5	672...	#N/A	Kenya	Kenya	KES	13	0	0	2	2017-07-26 00:00:00.0000000

Activate Windows

Cleaning Dataset

Example 3: Removing rows where the MPI values are 0



SQLQuery11.sql - D...K4PBESE\Terry (64) SQLQuery10.sql - D...K4PBESE\Terry (59)* SQLQuery9.sql - DE...K4PBESE\Terry (6)

```
SELECT * FROM Regions order by MPI
```

100 %

Results Messages Client Statistics

	RegionID	Country	Region	WorldRegion	MPI
1	746	Peru	Callao	Latin America and Caribbean	0
2	876	Syrian Arab Republic	Tortous	Arab States	0
3	1057	Kazakhstan	Kazakhstan	Central Asia	0
4	1063	Turkmenistan	Turkmenistan	Central Asia	0.001
5	1061	Armenia	Armenia	South Caucasus	0.001
6	959	Uzbekistan	Tashkent	Europe and Central Asia	0.001
7	1069	Bosnia and Herzegovina	Bosnia and Herzegovina	South Eastern Europe	0.002

Cleaning Dataset

Example 3: Removing rows where the MPI values are 0

The screenshot shows a SQL Server Management Studio window with two tabs: 'SQLQuery11.sql - D...K4PBESE\Terry (64)' and 'SQLQuery10.sql - D...K4PBE'. The 'SQLQuery11.sql' tab is active and contains the following SQL code:

```
DELETE FROM Regions WHERE MPI = 0
```

Below the code, the status bar shows '100 %' and a progress bar. At the bottom, there are two tabs: 'Messages' (selected) and 'Client Statistics'.

The 'Messages' tab displays the output of the query:

```
(3 rows affected)
```

Below the message, the completion time is shown:

```
Completion time: 2021-05-01T14:50:34.6213517+08:00
```

Cleaning Dataset

Example 4: Removing erroneous data; in this case there were 2 partner entries with the same partnerID but with 2 different locations. Upon searching on the internet, only one location was accurate so the wrong record was manually removed

The screenshot shows a SQL Server Management Studio window with two tabs: 'SQLQuery6.sql - DE...K4PBESE\Terry (70)*' and 'SQLQuery5.sql - DE...K4PBESE\Terry (75)'. The 'Results' tab is selected, displaying a table of partner data. A yellow box highlights the first two rows, which have the same PartnerID (127) but different locations (Costa Rica and Panama). The third row has PartnerID 129 and PartnerName 'Tanaoba Lais Manekat Foundation (TLM)'. The fourth row has PartnerID 130 and PartnerName 'Vision Fund Mexico'. The fifth row has PartnerID 131 and PartnerName 'Accion East'. The 'Country' column lists Costa Rica, Panama, Indonesia, Mexico, and United States respectively. The 'KivaPartner' column shows 'No' for all rows.

	PartnerID	PartnerName	Country	KivaPartner
43	127	EDESA	Costa Rica	No
44	127	EDESA	Panama	No
45	129	Tanaoba Lais Manekat Foundation (TLM)	Indonesia	No
46	130	Vision Fund Mexico	Mexico	No
47	131	Accion East	United States	No

Cleaning Dataset

Example 4: Removing erroneous data ; in this case there were 2 partner entries with the same partnerID but with 2 different locations. Upon searching on the internet, only one location was accurate so the wrong record was manually removed

The screenshot shows a SQL Server Management Studio window with two tabs: 'SQLQuery6.sql' and 'SQLQuery5.sql'. The 'SQLQuery6.sql' tab contains a DELETE statement to remove a partner entry where PartnerID=127 and Country like 'Panama'. The 'SQLQuery5.sql' tab contains a SELECT statement to retrieve all partner records ordered by PartnerID.

	PartnerID	PartnerName	Country	KivaPartner
40	123	Alalay sa Kaunlaran (ASKI)	Philippines	No
41	125	Community Economic Ventures, Inc. (CEVI)	Philippines	No
42	126	Paglaum Multi-Purpose Cooperative (PMPC)	Philippines	No
43	127	EDESA	Costa Rica	No
44	129	Tanaoba Lais Manekat Foundation (TLM)	Indonesia	No
45	130	Vision Fund Mexico	Mexico	No
46	131	Accion East	United States	No

Cleaning Dataset

Example 5: In the Regions table, there were instances where the Country values were not the same as those in the Loans table so when the MPI values were referenced based on Country when a JOIN was done on the 2 tables, NULL values resulted. The Country values in the Regions table had to changed to be consistent with those in the Loans table

Loans Table :

SQLQuery1.sql - DE...K4PBESE\Terry (67)* X

```
SELECT * FROM Loans WHERE Country LIKE 'Viet%' OR Country LIKE 'Bol%' OR Country LIKE 'Con'
```

100 %

Results Messages

	LoanID	LoanAmount	ActivityID	SectorID	UselD	RegionID	Country	Region	Currency	LoanTerm	LenderCount	TagID	RepaymentID	LoanDate	P:
2732	88860	3850	46	5	89144	1158	Bolivia	La Paz / El Alto	BOB	7	75	16	2	2014-07-17 00:00:00.0000000	5:
2733	88928	3625	23	8	89212	1158	Bolivia	La Paz / El Alto	BOB	12	111	12155	2	2014-07-18 00:00:00.0000000	5:
2734	88944	4550	13	1	89220	1158	Bolivia	La Paz / El Alto	BOB	12	122	12156	2	2014-07-18 00:00:00.0000000	5:
2735	88963	4550	131	3	89247	1158	Bolivia	La Paz / El Alto	BOB	9	75	4066	2	2014-07-18 00:00:00.0000000	5:
2736	89198	650	52	12	1068	1158	Bolivia	Bolivia	USD	27	22	59	2	2014-07-18 00:00:00.0000000	1:
2737	89221	5775	13	9	89505	1158	Bolivia	La Paz / El Alto	BOB	9	169	4955	2	2014-07-18 00:00:00.0000000	5:
2738	89562	725	34	14	36394	1158	Vietnam	Thanh Hoá	VND	20	22	1	2	2014-07-18 00:00:00.0000000	1:
2739	102450	5200	50	1	102...	202	Congo	Brazzaville	XAF	6	18	0	4	2014-08-14 00:00:00.0000000	1:
2740	102455	1500	34	14	102	1158	Bolivia	La Paz	USD	20	8	0	2	2014-08-14 00:00:00.0000000	4:
2741	102489	800	3	2	14435	1158	Bolivia	La Paz	USD	20	23	305	2	2014-08-14 00:00:00.0000000	4:
2742	102516	1025	20	5	102...	1158	Bolivia	Cochabamba	BOB	15	26	10	4	2014-08-14 00:00:00.0000000	1:
2743	102544	4750	29	1	102...	202	Congo	Brazzaville	XAF	6	30	0	4	2014-08-14 00:00:00.0000000	1:
2744	102671	1500	24	11	102...	1158	Bolivia	El Alto	USD	26	57	13861	2	2014-08-14 00:00:00.0000000	1:
2745	102678	4800	37	1	103...	202	Congo	Brazzaville	XAF	6	20	0	4	2014-08-14 00:00:00.0000000	1:
2746	102702	525	13	9	103...	1158	Bolivia	La Paz	BOB	20	6	8446	2	2014-08-14 00:00:00.0000000	4:

Cleaning Dataset

Example 5: In the Regions table, there were instances where the Country values were not the same as those in the Loans table so when the MPI values were referenced based on Country when a JOIN was done on the 2 tables, NULL values resulted. The Country values in the Regions table had to changed to be consistent with those in the Loans table

Regions Table :

SQLQuery1.sql - DE...K4PBESE\Terry (67)*

```
SELECT * FROM Regions WHERE Country LIKE 'Viet%' OR Country LIKE 'Bol%' OR Country LIKE 'Con%' |
```

RegionID	Country	Region	WorldRegion	MPI
7	Bolivia, Plurinational State of	Santa Cruz	Latin America and Caribbean	0.043
8	Bolivia, Plurinational State of	Beni	Latin America and Caribbean	0.099
9	Bolivia, Plurinational State of	Pando	Latin America and Caribbean	0.087
10	Congo, Democratic Republic of the	Kinshasa	Sub-Saharan Africa	0.102
11	Congo, Democratic Republic of the	Bandundu	Sub-Saharan Africa	0.435
12	Congo, Democratic Republic of the	Bas-Congo	Sub-Saharan Africa	0.393
13	Congo, Democratic Republic of the	équateur	Sub-Saharan Africa	0.48
14	Congo, Democratic Republic of the	Kasaï-Occidental	Sub-Saharan Africa	0.49
15	Congo, Democratic Republic of the	Kasaï-Oriental	Sub-Saharan Africa	0.431
16	Congo, Democratic Republic of the	Katanga	Sub-Saharan Africa	0.4
17	Congo, Democratic Republic of the	Mai-Ndombe	Sub-Saharan Africa	0.409
18	Congo, Democratic Republic of the	Nord-Kivu	Sub-Saharan Africa	0.349
19	Congo, Democratic Republic of the	Oriental	Sub-Saharan Africa	0.401
20	Congo, Democratic Republic of the	Sud-Kivu	Sub-Saharan Africa	0.462
21	Congo, Republic of	Kouilou	Sub-Saharan Africa	0.352
22	Congo, Republic of	Nia	Sub-Saharan Africa	0.238
23	Congo, Republic of	Lékumou	Sub-Saharan Africa	0.323
24	Congo, Republic of	Bouenza	Sub-Saharan Africa	0.274
25	Congo, Republic of	Pointe-Noire	Sub-Saharan Africa	0.299
26	Congo, Republic of	Réveaux	Sub-Saharan Africa	0.357
27	Congo, Republic of	Cuvette	Sub-Saharan Africa	0.246
28	Congo, Republic of	Cuvette-Ouest	Sub-Saharan Africa	0.312
29	Congo, Republic of	Sangha	Sub-Saharan Africa	0.285
30	Congo, Republic of	Likouala	Sub-Saharan Africa	0.304
31	Congo, Republic of	Brazzaville	Sub-Saharan Africa	0.103
32	Congo, Republic of	Pointe-Noire	Sub-Saharan Africa	0.094
33	Viet Nam	Red River Delta	East Asia and the Pacific	0.013
34	Viet Nam	Northern Midla...	East Asia and the Pacific	0.064
35	Viet Nam	North Central a...	East Asia and the Pacific	0.024
36	Viet Nam	Central Highlands	East Asia and the Pacific	0.057
37	Viet Nam	South East	East Asia and the Pacific	0.017
38	Viet Nam	Mekong River ...	East Asia and the Pacific	0.03
39	Viet Nam	Viet Nam	East Asia and the Pacific	0.034
40	Bolivia, Plurinational State of	Bolivia, Plurinati...	Latin America and Caribbean	0.098
41	Congo, Democratic Republic of the	Congo, Democr...	Sub-Saharan Africa	0.396

Cleaning Dataset

Example 5: In the Regions table, there were instances where the Country values were not the same as those in the Loans table so when the MPI values were referenced based on Country when a JOIN was done on the 2 tables, NULL values resulted. The Country values in the Regions table had to changed to be consistent with those in the Loans table

Regions Table :

The screenshot shows a SQL Server Management Studio window titled "SQLQuery1.sql - DE...K4PBESE\Terry (67)". It contains three statements:

```
UPDATE Regions SET Country = 'Bolivia' WHERE Country LIKE 'Bolivia%'  
UPDATE Regions SET Country = 'Congo' WHERE Country LIKE 'Congo%'  
UPDATE Regions SET Country = 'Vietnam' WHERE Country LIKE 'Viet%'  
  
SELECT * FROM Regions WHERE Country LIKE 'Viet%' OR Country LIKE 'Bol%' OR Country LIKE 'Con%'
```

The "Results" tab displays the output of the SELECT statement, showing 42 rows of data from the Regions table. The columns are RegionID, Country, Region, WorldRegion, and MPI. The data includes various regions like Congo, Vietnam, and Bolivia.

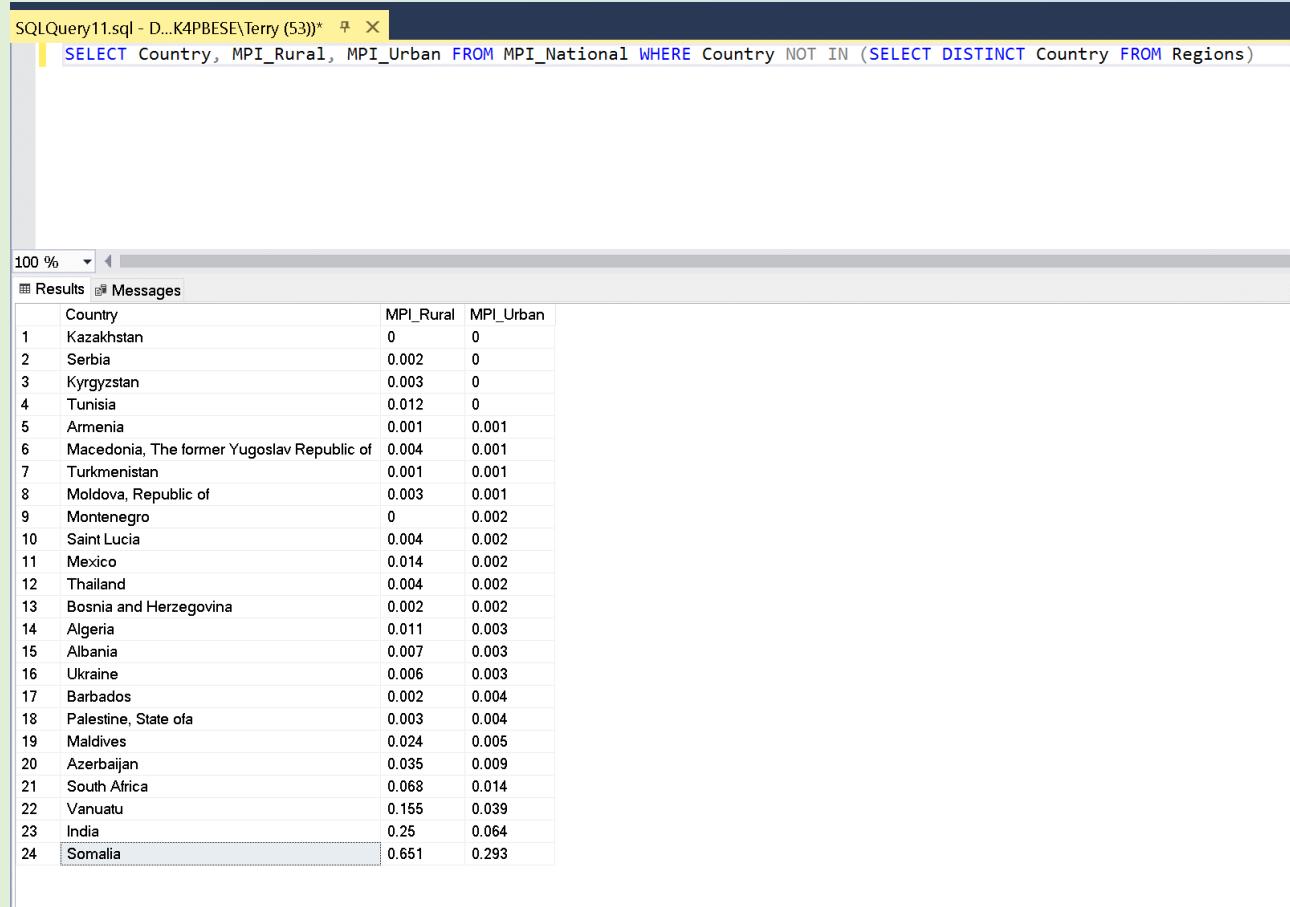
RegionID	Country	Region	WorldRegion	MPI	
13	184	Congo	équateur	Sub-Saharan Africa	0.48
14	185	Congo	Kasaô-Occidental	Sub-Saharan Africa	0.49
15	186	Congo	Kasaô-Oriental	Sub-Saharan Africa	0.431
16	187	Congo	Katanga	Sub-Saharan Africa	0.4
17	188	Congo	Maniema	Sub-Saharan Africa	0.409
18	189	Congo	Nord-Kivu	Sub-Saharan Africa	0.349
19	190	Congo	Orientale	Sub-Saharan Africa	0.401
20	191	Congo	Sud-Kivu	Sub-Saharan Africa	0.462
21	192	Congo	Kouilou	Sub-Saharan Africa	0.352
22	193	Congo	Niari	Sub-Saharan Africa	0.238
23	194	Congo	Lékoumou	Sub-Saharan Africa	0.323
24	195	Congo	Bouenza	Sub-Saharan Africa	0.274
25	196	Congo	Pool	Sub-Saharan Africa	0.299
26	197	Congo	Plateaux	Sub-Saharan Africa	0.357
27	198	Congo	Cuvette	Sub-Saharan Africa	0.246
28	199	Congo	Cuvette-Ouest	Sub-Saharan Africa	0.312
29	200	Congo	Sangha	Sub-Saharan Africa	0.285
30	201	Congo	Likouala	Sub-Saharan Africa	0.304
31	202	Congo	Brazzaville	Sub-Saharan Africa	0.103
32	203	Congo	Pointe-Noire	Sub-Saharan Africa	0.094
33	960	Vietn...	Red River Delta	East Asia and the Pacific	0.013
34	961	Vietn...	Northern Midla...	East Asia and the Pacific	0.064
35	962	Vietn...	North Central a...	East Asia and the Pacific	0.024
36	963	Vietn...	Central Highlands	East Asia and the Pacific	0.057
37	964	Vietn...	South East	East Asia and the Pacific	0.017
38	965	Vietn...	Mekong River ...	East Asia and the Pacific	0.03
39	1096	Vietn...	Viet Nam	East Asia and the Pacific	0.034
40	1100	Bolivia	Bolivia, Plurinati...	Latin America and Caribbean	0.098
41	1126	Congo	Congo, Democr...	Sub-Saharan Africa	0.396
42	1127	Congo	Congo, Republi...	Sub-Saharan Africa	0.266

At the bottom of the results, it says "Query executed successfully." and "42 rows".

Data Transformation

Example: Replacing MPI values for countries in the Regions table that had no MPI values from another dataset MPI_National

In the Data cleaning process, countries that had NULL MPI values were removed. Another dataset, MPI_National, has the MPI values for these missing countries



The screenshot shows a SQL query window and its results pane. The query is:

```
SQLQuery11.sql - D...K4PBSE\Terry (53)*  ×  
SELECT Country, MPI_Rural, MPI_Urban FROM MPI_National WHERE Country NOT IN (SELECT DISTINCT Country FROM Regions)
```

The results pane displays a table with three columns: Country, MPI_Rural, and MPI_Urban. The data consists of 24 rows, each representing a country and its corresponding MPI values. The countries listed are: Kazakhstan, Serbia, Kyrgyzstan, Tunisia, Armenia, Macedonia, The former Yugoslav Republic of, Turkmenistan, Moldova, Republic of, Montenegro, Saint Lucia, Mexico, Thailand, Bosnia and Herzegovina, Algeria, Albania, Ukraine, Barbados, Palestine, State of, Maldives, Azerbaijan, South Africa, Vanuatu, India, and Somalia.

	Country	MPI_Rural	MPI_Urban
1	Kazakhstan	0	0
2	Serbia	0.002	0
3	Kyrgyzstan	0.003	0
4	Tunisia	0.012	0
5	Armenia	0.001	0.001
6	Macedonia, The former Yugoslav Republic of	0.004	0.001
7	Turkmenistan	0.001	0.001
8	Moldova, Republic of	0.003	0.001
9	Montenegro	0	0.002
10	Saint Lucia	0.004	0.002
11	Mexico	0.014	0.002
12	Thailand	0.004	0.002
13	Bosnia and Herzegovina	0.002	0.002
14	Algeria	0.011	0.003
15	Albania	0.007	0.003
16	Ukraine	0.006	0.003
17	Barbados	0.002	0.004
18	Palestine, State of	0.003	0.004
19	Maldives	0.024	0.005
20	Azerbaijan	0.035	0.009
21	South Africa	0.068	0.014
22	Vanuatu	0.155	0.039
23	India	0.25	0.064
24	Somalia	0.651	0.293

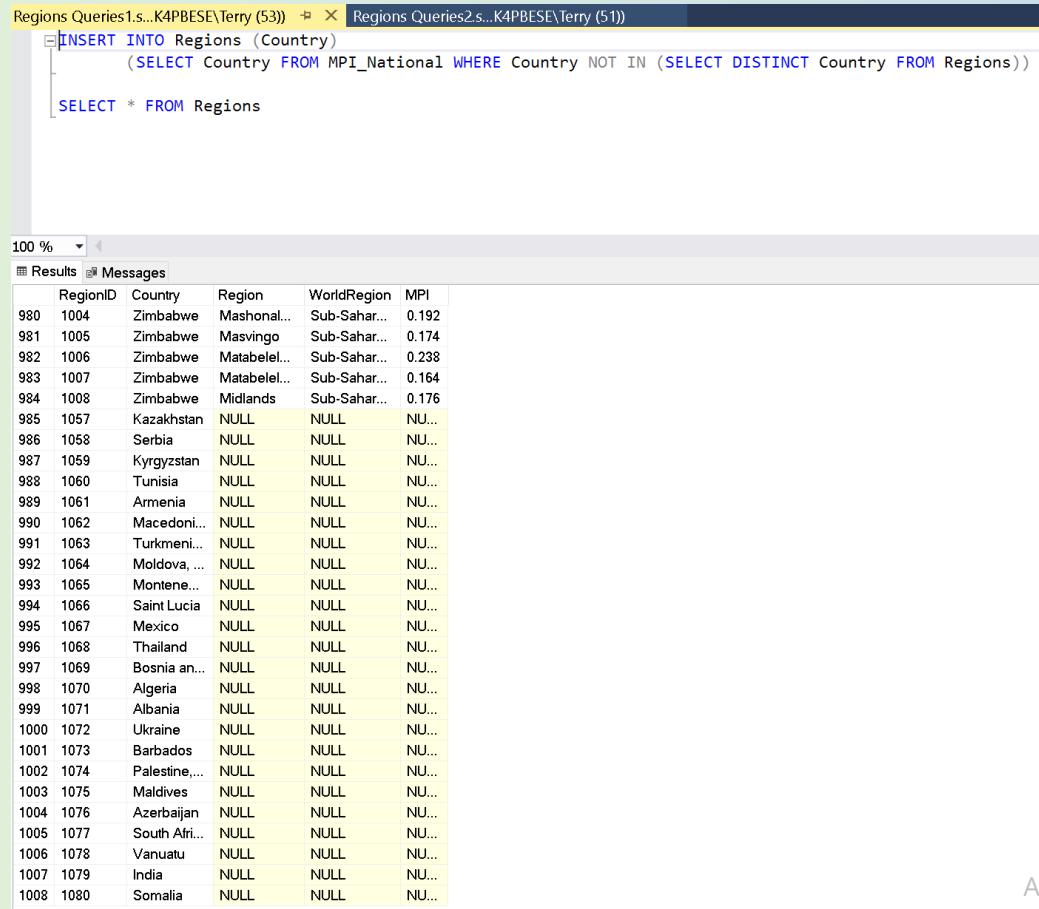
Data Transformation

Example: Replacing MPI values for countries in the Regions table that had no MPI values from another dataset MPI_National

Step 1:

Adding the countries that are in MPI_National table but not in Regions table to the Regions table.

The Region field will have the same value as the corresponding country field as the MPI_National table only has the MPI for a country whereas the Regions table has MPI values for regions of a country. The RegionID which has been set as the Primary Key in the Regions table has been set with IDENTITY(1,1) so the RegionID autoincrements for each country when the INSERT command is executed



The screenshot shows the SQL Server Management Studio interface with two tabs: 'Regions Queries1.s...K4PBSE\Terry (53)' and 'Regions Queries2.s...K4PBSE\Terry (51)'. The 'Regions Queries1' tab contains the following T-SQL code:

```
INSERT INTO Regions (Country)
    (SELECT Country FROM MPI_National WHERE Country NOT IN (SELECT DISTINCT Country FROM Regions))
SELECT * FROM Regions
```

The 'Results' tab displays the output of the query, showing a list of countries being inserted into the Regions table. The table has columns: RegionID, Country, Region, WorldRegion, and MPI. The MPI column for all rows is NULL. The data includes countries like Zimbabwe, Kazakhstan, Serbia, Kyrgyzstan, Tunisia, Armenia, Macedonia, Turkmenistan, Moldova, Montenegro, Saint Lucia, Mexico, Thailand, Bosnia and Herzegovina, Algeria, Albania, Ukraine, Barbados, Palestine, Maldives, Azerbaijan, South Africa, Vanuatu, India, and Somalia.

RegionID	Country	Region	WorldRegion	MPI	
980	1004	Zimbabwe	Mashonaland	Sub-Saharan Africa	0.192
981	1005	Zimbabwe	Masvingo	Sub-Saharan Africa	0.174
982	1006	Zimbabwe	Matabeleland	Sub-Saharan Africa	0.238
983	1007	Zimbabwe	Matabeleland	Sub-Saharan Africa	0.164
984	1008	Zimbabwe	Midlands	Sub-Saharan Africa	0.176
985	1057	Kazakhstan	NULL	NULL	NULL
986	1058	Serbia	NULL	NULL	NULL
987	1059	Kyrgyzstan	NULL	NULL	NULL
988	1060	Tunisia	NULL	NULL	NULL
989	1061	Armenia	NULL	NULL	NULL
990	1062	Macedonia	NULL	NULL	NULL
991	1063	Turkmenistan	NULL	NULL	NULL
992	1064	Moldova, Republic of	NULL	NULL	NULL
993	1065	Montenegro	NULL	NULL	NULL
994	1066	Saint Lucia	NULL	NULL	NULL
995	1067	Mexico	NULL	NULL	NULL
996	1068	Thailand	NULL	NULL	NULL
997	1069	Bosnia and Herzegovina	NULL	NULL	NULL
998	1070	Algeria	NULL	NULL	NULL
999	1071	Albania	NULL	NULL	NULL
1000	1072	Ukraine	NULL	NULL	NULL
1001	1073	Barbados	NULL	NULL	NULL
1002	1074	Palestine, State of	NULL	NULL	NULL
1003	1075	Maldives	NULL	NULL	NULL
1004	1076	Azerbaijan	NULL	NULL	NULL
1005	1077	South Africa	NULL	NULL	NULL
1006	1078	Vanuatu	NULL	NULL	NULL
1007	1079	India	NULL	NULL	NULL
1008	1080	Somalia	NULL	NULL	NULL

Data Transformation

Example: Replacing MPI values for countries in the Regions table that had no MPI values from another dataset MPI_National

Step 2: Taking the average of MPI_Rural and MPI_Urban from the MPI_National table and updating the value to the MPI column in the Regions table for the respective countries

```
UPDATE Regions SET MPI = a.MPI
  FROM
  (
    SELECT Country,
    CASE
      WHEN MPI_Rural = 0 THEN MPI_Urban
      WHEN MPI_Urban = 0 THEN MPI_Rural
      ELSE (MPI_Rural+MPI_Urban) / 2
    END AS MPI
    FROM MPI_National
    WHERE Country IN
      (SELECT Country FROM MPI_National WHERE Country IN (SELECT DISTINCT Country FROM Regions WHERE MPI IS NULL))
  ) a
 WHERE Regions.Country = a.Country

UPDATE Regions SET Region = Country WHERE Region IS NULL

SELECT * FROM Regions |
```

Data Transformation

Example: Replacing MPI values for countries in the Regions table that had no MPI values from another dataset MPI_National

Result from Step 2

	RegionID	Country	Region	WorldRegion	MPI
991	1063	Turkmenistan	Turkmenis...	Central Asia	0.001
992	1064	Moldova, Repub...	Moldova, ...	Balkans	0.002
993	1065	Montenegro	Montenegro	Europe	0.002
994	1066	Saint Lucia	Saint Lucia	Caribbean Sea	0.003
995	1067	Mexico	Mexico	South America	0.008
996	1068	Thailand	Thailand	South East Asia	0.003
997	1069	Bosnia and Herz...	Bosnia an...	South Eastern E...	0.002
998	1070	Algeria	Algeria	Northern Africa	0.007
999	1071	Albania	Albania	South Europe	0.005
1000	1072	Ukraine	Ukraine	Eastern Europe	0.0045
1001	1073	Barbados	Barbados	Caribbean	0.003
1002	1074	Palestine, State ofa	Palestine, ...	Gaza	0.0035
1003	1075	Maldives	Maldives	Indian Ocean	0.0145
1004	1076	Azerbaijan	Azerbaijan	South Caucasus	0.022
1005	1077	South Africa	South Africa	Africa	0.041
1006	1078	Vanuatu	Vanuatu	Oceania	0.097
1007	1079	India	India	Asia	0.157
1008	1080	Somalia	Somalia	Sub-Saharan Afri...	0.472

Data Transformation

Example: Replacing MPI values for countries in the Regions table that had no MPI values from another dataset MPI_National

Due to the fact that there are many loans in the Loans table that has '#N/A' values because there are many loans in the Loans table that are in regions that are not in the Regions table but their countries are in the Regions table. These regions with '#N/A' for the RegionID will be replaced with the newly generated RegionID done in Step 1

fill in NA in region... - K4PBESE\Terry (59)* - X SQLQuery9.sql - DE...K4PBESE\Terry (60)*

SELECT * FROM Loans

SELECT Country, Region, RegionID FROM Loans WHERE RegionID LIKE '#N/A'

100 %

Results Messages

	LoanID	LoanAmount	ActivityID	SectorID	UseID	RegionID	Country	Region	Currency	LoanTerm	LenderCount	TagID	RepaymentID	LoanDate
33...	33965	350	20	5	2731	#N/A	Philip...	Narra, Palawan	PHP	8	11	16	4	2014-03-20 00:00:00.000000
33...	33966	450	20	5	34059	#N/A	Philip...	Clarin, Mis...	PHP	10	17	0	2	2014-03-20 00:00:00.000000
33...	33967	1475	39	5	34060	#N/A	Hond...	taulabe	HNL	14	42	12	2	2014-03-20 00:00:00.000000
33...	33968	575	41	8	8845	#N/A	Philip...	Puerto Princ...	PHP	5	11	0	4	2014-03-20 00:00:00.000000
33...	33969	375	1	1	34062	744	Peru	Ayacucho	PEN	10	13	1495	2	2014-03-20 00:00:00.000000
33...	33970	3000	8	4	34063	#N/A	Palesti...	Palestine	USD	38	60	5513	2	2014-03-20 00:00:00.000000
33...	33971	750	74	5	9784	#N/A	Kyrgy...	Talas	KGS	13	30	2218	2	2014-03-20 00:00:00.000000
33...	33972	525	34	14	24984	475	Camb...	Pursat	KHR	14	18	3	2	2014-03-20 00:00:00.000000
33...	33973	475	57	5	29902	#N/A	Pakistan	Rahim Yar K...	PKR	14	19	3	2	2014-03-20 00:00:00.000000
33...	33974	2950	44	1	34067	#N/A	India	Sonepur, Od...	INR	43	107	3	1	2014-03-20 00:00:00.000000
33...	33975	1325	52	12	34068	#N/A	Jordan	Irbid	JOD	17	51	3753	2	2014-03-20 00:00:00.000000
33...	33976	725	39	5	34069	#N/A	Peru	Pomahuaca ...	PEN	8	26	0	1	2014-03-20 00:00:00.000000
33...	33977	2100	13	9	34070	909	Tajikis...	Dushanbe	TJS	14	69	599	2	2014-03-20 00:00:00.000000
33...	33978	2000	39	5	34071	#N/A	Tajikis...	Shahristan	USD	14	72	5514	2	2014-03-20 00:00:00.000000
33...	33979	575	40	2	8920	#N/A	Philip...	Puerto Princ...	PHP	8	17	0	4	2014-03-20 00:00:00.000000
33...	33980	4975	13	9	34073	#N/A	Senegal	Senegal	XOF	8	161	4643	2	2014-03-20 00:00:00.000000
33...	33981	1475	20	5	34074	#N/A	Kyrgy...	Jylkol village	KGS	11	37	1	2	2014-03-20 00:00:00.000000

	Country	Region	RegionID
1	Pakistan	Lahore	#N/A
2	Pakistan	Lahore	#N/A
3	India	Maynaguri	#N/A
4	Pakistan	Lahore	#N/A
5	Pakistan	Abdul Hakeem	#N/A
6	Kenya	Kenya	#N/A
7	India	Maynaguri	#N/A
8	Pakistan	Ellahabad	#N/A
9	Pakistan	Lahore	#N/A
10	Pakistan	Lahore	#N/A
11	Pakistan	Lahore	#N/A
12	Pakistan	Faisalabad	#N/A
13	Pakistan	Faisalabad	#N/A
14	Pakistan	Lahore	#N/A
15	India	Dhupguri	#N/A
16	Pakistan	Lahore	#N/A
17	Pakistan	Faisalabad	#N/A

Activate Windows

Data Transformation

Example: Replacing MPI values for countries in the Regions table that had no MPI values from another dataset MPI_National

Step 3: Update the RegionID field in the Loans table where the RegionID has '#N/A' value

The screenshot shows a SQL query in the SQL Query window of SSMS. The query uses an UPDATE statement to set the RegionID in the Loans table to the RegionID from the Regions table where the Country name matches the Region name. It includes a WHERE clause to filter rows where the current RegionID is '#N/A'. Below the query, a SELECT statement is shown to verify the data.

```
SQLQuery16.sql - D...K4PBESE\Terry (61)* fill in NA in region...K4PBESE\Terry (59) ×
UPDATE Loans SET Loans.RegionID = a.RegionID
  FROM
  (
    SELECT Country, RegionID from Regions where Country=Region
  ) a
 WHERE Loans.Country = a.Country and Loans.RegionID like '#N/A'

SELECT * FROM Loans WHERE Country IN (SELECT DISTINCT Country FROM Regions)
```

Results

LoanID	LoanAmount	ActivityID	SectorID	UserID	RegionID	Country	Region	Currency	LoanTerm	LenderCount	TagID	RepaymentID	LoanDate
1	300	1	1	1	1119	Pakistan	Lahore	PKR	12	12	0	4	2014-01-01 00:00:00.0000000
2	575	2	2	2	1119	Pakistan	Lahore	PKR	11	14	0	4	2014-01-01 00:00:00.0000000
3	150	3	2	3	1079	India	Maynaguri	INR	43	6	1	1	2014-01-01 00:00:00.0000000
4	200	4	3	4	1119	Pakistan	Lahore	PKR	11	8	0	4	2014-01-01 00:00:00.0000000
5	400	5	1	5	1119	Pakistan	Abdul Hakeem	PKR	14	16	0	2	2014-01-01 00:00:00.0000000
6	250	6	4	6	1135	Kenya	Nanya	KES	4	6	0	4	2014-01-01 00:00:00.0000000
7	200	7	5	7	1079	India	Maynaguri	INR	43	8	1	1	2014-01-01 00:00:00.0000000
8	400	8	4	8	1119	Pakistan	Ellahabad	PKR	14	8	2	2	2014-01-01 00:00:00.0000000
9	475	9	6	9	1119	Pakistan	Lahore	PKR	14	19	3	2	2014-01-01 00:00:00.0000000
10	625	10	1	10	1119	Pakistan	Lahore	PKR	11	24	0	4	2014-01-01 00:00:00.0000000
11	200	2	2	11	1119	Pakistan	Lahore	PKR	11	3	0	4	2014-01-01 00:00:00.0000000
12	400	11	7	12	1119	Pakistan	Faisalabad	PKR	14	16	0	2	2014-01-01 00:00:00.0000000
13	400	12	8	13	1119	Pakistan	Faisalabad	PKR	14	16	4	2	2014-01-01 00:00:00.0000000
14	400	13	9	14	1119	Pakistan	Lahore	PKR	12	10	0	4	2014-01-01 00:00:00.0000000
15	225	14	5	15	1079	India	Dhupguri	INR	43	7	3	1	2014-01-01 00:00:00.0000000
16	300	2	2	16	1119	Pakistan	Lahore	PKR	11	9	0	4	2014-01-01 00:00:00.0000000
17	400	12	8	17	1119	Pakistan	Faisalabad	PKR	14	11	5	2	2014-01-01 00:00:00.0000000
18	875	15	4	18	1119	Pakistan	Lahore	PKR	11	25	0	4	2014-01-01 00:00:00.0000000
19	250	16	4	19	1079	India	Maynaguri	INR	43	4	1	1	2014-01-01 00:00:00.0000000
20	475	8	4	20	1119	Pakistan	Lahore	PKR	14	18	6	2	2014-01-01 00:00:00.0000000
21	250	17	1	21	1119	Pakistan	Lahore	PKR	11	10	0	4	2014-01-01 00:00:00.0000000
22	350	18	1	22	1119	Pakistan	Lahore	PKR	12	13	0	4	2014-01-01 00:00:00.0000000
23	575	13	9	23	1119	Pakistan	Lahore	PKR	12	20	0	4	2014-01-01 00:00:00.0000000
24	350	4	3	24	1119	Pakistan	Lahore	PKR	14	6	0	2	2014-01-01 00:00:00.0000000
25	250	19	1	25	1119	Pakistan	Lahore	PKR	11	7	0	4	2014-01-01 00:00:00.0000000

Query executed successfully.

Data Transformation

Example: Replacing MPI values for countries in the Regions table that had no MPI values from another dataset MPI_National

There are still some records in the Loans table where the RegionID still has the '#N/A' value because these records are for countries that are not in the Regions table and not in the MPI_National table and there are no MPI values for these countries available. Any analysis done based on MPI will not be done on these loan records. However the '#N/A' will be replaced with a new RegionID with a 'Not Specified' value

SQLQuery16.sql - D...K4PBESE\Terry (61)* fill in NA in region...-K4PBESE\Terry (59)* X

```
SELECT DISTINCT Country, RegionID FROM Loans WHERE RegionID LIKE '#N/A' ORDER BY Country
```

100 %

Results Messages

	Country	RegionID
1	Bolivia	#N/A
2	Chile	#N/A
3	Costa Rica	#N/A
4	Georgia	#N/A
5	Guam	#N/A
6	Israel	#N/A
7	Kosovo	#N/A
8	Lebanon	#N/A
9	Moldova	#N/A
10	Myanmar (Burma)	#N/A
11	Palestine	#N/A
12	Panama	#N/A
13	Paraguay	#N/A
14	Puerto Rico	#N/A
15	Saint Vincent and the Grenadines	#N/A
16	Samoa	#N/A
17	Solomon Islands	#N/A
18	Tanzania	#N/A
19	The Democratic Republic of the Congo	#N/A
20	Turkey	#N/A
21	United States	#N/A
22	Vietnam	#N/A
23	Virgin Islands	#N/A

Data Transformation

Example: Replacing MPI values for countries in the Regions table that had no MPI values from another dataset MPI_National

Step 4: Insert a record in the Regions table with a 'Not Specified' entry with a unique RegionID

```
INSERT INTO Regions VALUES ('Not Specified','Not Specified','Not Specified',NULL)
```

Step 5: Update the records in the Loans table with this RegionID where the countries are not found in the Regions table



The screenshot shows a SQL query editor window with a code editor and a messages panel.

```
UPDATE Loans SET RegionID = '1158' WHERE Country IN
(SELECT distinct Country FROM Loans
WHERE Country NOT IN (SELECT DISTINCT Country FROM Regions) AND RegionID LIKE '#N/A')
```

Messages panel:

- (80636 rows affected)
- Completion time: 2021-05-01T20:15:26.1190071+08:00

Data Analysis

Analysis 1: Investigate if the loans given matches the needs of the countries

Process: 5 deprivations were analyzed in this analysis – Drinking water, Education, Sanitation, Electricity and Health.

Queries to determine the number of loans Kiva has given for each of these deprivations per country

```
-- Find Number of Loans Kiva has given in The Living Conditions Deprivation per Country
SELECT Country, Count(*) AS NumLoans FROM Loans
WHERE SectorID=14 OR
Useid IN (SELECT DISTINCT useid FROM uses
WHERE [use] LIKE '%house%' OR [use] LIKE '%renovate%' OR [use] LIKE '%solar%'
OR [use] LIKE '%appliance%' OR [use] LIKE '%furniture%' OR [use] LIKE '%lamp%' OR [use] LIKE '%light%'
OR [use] LIKE '%bed%' OR [use] LIKE '%refigerator%' OR [use] LIKE '%room%' OR [use] LIKE '%paint%'
OR [use] LIKE '%television%' OR [use] LIKE '%roof%' OR [use] LIKE '%door%' OR [use] LIKE '%window%'
OR [use] LIKE '%stove%' OR [use] LIKE '%insulat%' OR [use] LIKE '%motorc%' OR [use] LIKE '%sand%'
OR [use] LIKE '%brick%')
GROUP BY Country ORDER BY Country

-- Find Number of Loans Kiva has given in The Electricity Sector per Country
SELECT Country, Count(*) AS NumLoans FROM Loans
WHERE Useid IN (SELECT DISTINCT useid FROM uses
WHERE [use] LIKE '%electricity%' OR [use] LIKE '%electric%' OR [use] LIKE '%power%')
GROUP BY Country ORDER BY Country

-- Find Number of Loans Kiva has given in The Health Sector per Country
SELECT Country, Count(*) AS NumLoans FROM Loans
WHERE SectorID=11 OR
Useid IN (SELECT DISTINCT useid FROM uses
WHERE [use] LIKE '%health%' OR [use] LIKE '%doctor%' OR [use] LIKE '%medic%' OR [use] LIKE '%sick%')
GROUP BY Country ORDER BY Country
```

Data Analysis

Analysis 1: Investigate if the loans given matches the needs of the countries

Process: 5 deprivations were analyzed in this analysis – Drinking water, Education, Sanitation, Electricity and Health.

Queries to determine the number of loans Kiva has given for each of these deprivations per country

```
-- Find Number of Loans Kiva has given in The Drinking Water Sector per Country
SELECT Country, Count(*) AS NumLoans FROM Loans
WHERE Useid IN (SELECT DISTINCT useid FROM uses
WHERE [use] LIKE '%water%' OR [use] LIKE '%tank%' OR [use] LIKE '%filter%' OR [use] LIKE '%pipe%' OR
[use] LIKE '%purif%')
GROUP BY Country
ORDER BY Country

-- Find Number of Loans Kiva has given in The Sanitation Sector per Country
SELECT Country, Count(*) AS NumLoans FROM Loans
WHERE Useid IN (SELECT DISTINCT useid FROM uses
WHERE [use] LIKE '%toilet%' OR [use] LIKE '%bath%' OR [use] LIKE '%latrine%' OR
[use] LIKE '%septic%')
GROUP BY Country
ORDER BY Country
```

Data Analysis

Analysis 1: Investigate if the loans given matches the needs of the countries

Process: 5 deprivations were analyzed in this analysis – Drinking water, Education, Sanitation, Electricity and Health.

Results of Queries to determine the number of loans Kiva has given for each of these deprivations per country were consolidated in an excel table

Country	Education	Sanitation	DrinkingWater	Living Conditions	Health	Electricity
Ethiopia	0	0	0	0	0	0
Gabon	0	0	0	0	0	0
Gambia	0	0	0	0	0	0
Ghana	135	57	189	321	40	46
Guatemala	58	23	51	1049	76	10
Guinea	0	0	0	0	0	0
Guinea-Bissau	0	0	0	0	0	0
Guyana	0	0	0	0	0	0
Haiti	22	30	20	485	27	320
Honduras	125	72	139	972	59	131
Indonesia	245	271	2207	2186	21	91
Iraq	262	0	2	238	2	24
Jamaica	0	0	0	0	0	0
Jordan	2049	0	13	166	23	21
Kenya	2772	325	1599	10278	1048	191
Lao People's Democ	0	0	1485	1	18	0
Lesotho	0	0	0	422	198	0
Liberia	1	19	179	42	29	3
Madagascar	43	4	31	302	24	23
Malawi	6	2	1	10	1	3
Mali	31	0	8	32	22	3
Mauritania	0	0	0	0	0	0
Mongolia	89	1	2	433	7	53
Morocco	0	0	0	0	0	0
Mozambique	44	52	58	1792	9	40
Myanmar	0	0	0	0	0	0
Namibia	0	0	0	8	0	0
Nepal	2	0	0	19	26	7
Nicaragua	1136	486	247	3019	229	228
Niger	0	0	0	0	0	0
Nigeria	1397	0	4	470	0	1
Pakistan	1306	10	89	2059	228	230
Peru	1317	18	94	2079	283	73
Philippines	2585	7766	1409	13267	1731	174
Rwanda	542	2	84	478	16	273
Sao Tome and Principe	0	0	0	0	0	0

Data Analysis

Analysis 2: Investigate the number of loans given to countries with higher MPIs and lower MPIs

Total Number of Loans for Lower MPI Countries

```
--3 Number of Loans for the Countries with the Lowest MPI
SELECT Country, COUNT(*) AS NumOfLoans FROM Loans WHERE Country IN
(SELECT Country FROM Regions GROUP BY Country HAVING AVG(MPI) < 0.33)
GROUP BY Country ORDER BY NumOfLoans DESC
```

100 %

Results Messages

Country	NumOfLoans
1 Philippines	160441
2 Kenya	75825
3 El Salvador	39875
4 Cambodia	34836
5 Pakistan	26857
6 Peru	22233
7 Colombia	21995
8 Tajikistan	19580
9 Ecuador	13521
10 Nicaragua	11781
11 India	11237
12 Vietnam	10843
13 Nigeria	10136
14 Bolivia	8806
15 Armenia	8631
16 Guatemala	7310
17 Rwanda	6735
18 Honduras	6557
19 Indonesia	6214
20 Kyrgyzstan	5774
21 Togo	5749
22 Mexico	5741
23 Ghana	4374
24 Jordan	4167
...	...
Total	1001

Total Number of Loans for Higher MPI Countries

```
-- 2. Number of Loans for the Countries with the Highest MPI
SELECT Country, COUNT(*) AS NumOfLoans FROM Loans WHERE Country IN
(SELECT Country FROM Regions GROUP BY Country HAVING AVG(MPI) >= 0.33)
GROUP BY Country ORDER BY NumOfLoans
```

100 %

Results Messages

Country	NumOfLoans
1 Cote D'Ivoire	1
2 Somalia	75
3 South Sudan	160
4 Burundi	880
5 Burkina Faso	2460
6 Timor-Leste	2690
7 Senegal	3269
8 Mozambique	3483
9 Liberia	3682
10 Madagascar	3821
11 Sierra Leone	5415
12 Mali	6639
13 Uganda	20601

Data Analysis

Analysis 3: Investigate the total loan amounts given to countries with higher MPIs and lower MPIs

Total Loan Amounts for Lower MPI Countries

```
--5 Amount of Loans that Countries with the Highest MPI are getting
SELECT Country, FORMAT(SUM(LoanAmount), '#,###,###') AS TotalLoansAmt FROM Loans WHERE Country IN
(SELECT Country FROM Regions GROUP BY Country HAVING AVG(MPI) < 0.33)
GROUP BY Country ORDER BY SUM(LoanAmount) DESC

-- 6 Query 5 but with MPI values displayed
SELECT r.Country, FORMAT(SUM(l.LoanAmount), '#,###,###') AS TotalLoansAmt , r.MPI
FROM Regions r JOIN Loans l ON r.RegionID=l.RegionID
GROUP BY r.Country, r.MPI
HAVING AVG(r.MPI) < 0.33
ORDER BY SUM(l.LoanAmount) DESC
```

100 %

Results Messages

Country	TotalLoansAmt
1 Philippines	55,342,225
2 Kenya	34,534,300
3 Peru	30,867,400
4 El Salvador	26,603,650
5 Cambodia	19,872,700

Country	TotalLoansAmt	MPI
1 Philippines	55,342,225	0.062
2 Kenya	33,323,800	0.209
3 El Salvador	24,245,650	0.031
4 Peru	20,269,400	0.055
5 Rwanda	16,646,300	0.243
6 Cambodia	16,448,950	0.164
7 Ecuador	14,761,825	0.018
8 Colombia	14,359,400	0.025
9 Tajikistan	13,813,675	0.047
10 Pakistan	13,281,025	0.233
11 Armenia	12,915,400	0.001
12 Mexico	9,618,425	0.008
13 Peru	8,727,275	0.062
14 Guatemala	7,516,875	0.114
15 Kyrgyzstan	7,152,300	0.003
16 India	6,727,225	0.157

Total Loan Amounts for Higher MPI Countries

```
-- 7 Query 4 but with MPI values displayed
SELECT r.Country, FORMAT(SUM(l.LoanAmount), '#,###,###') AS TotalLoansAmt , r.MPI
FROM Regions r JOIN Loans l ON r.RegionID=l.RegionID
GROUP BY r.Country, r.MPI
HAVING AVG(r.MPI) > 0.33
ORDER BY SUM(l.LoanAmount)
```

100 %

Results Messages

Country	TotalLoansAmt	MPI
1 Malawi	13,225	0.334
2 Sierra Leone	33,550	0.522
3 Cote d'Ivoire	50,000	0.348
4 Timor-Leste	64,500	0.379
5 Timor-Leste	73,400	0.41
6 South Sudan	120,900	0.55
7 Sierra Leone	171,150	0.477
8 Chad	173,550	0.744
9 Timor-Leste	179,400	0.357
10 Timor-Leste	221,200	0.497
11 Somalia	225,875	0.472
12 Mali	296,450	0.484
13 Sierra Leone	343,875	0.449
14 Liberia	1,197,825	0.434
15 Madagascar	1,236,025	0.39
16 Timor-Leste	1,675,400	0.373
17 Mozambique	2,066,450	0.347
18 Burundi	2,667,850	0.412
19 Burkina Faso	2,972,700	0.548
20 Sierra Leone	3,631,950	0.481
21 Senegal	7,033,925	0.351
22 Mali	7,961,975	0.445

Data Analysis

Analysis 4: Investigate if there is a correlation between the number of partners a country has and the number of loans that the country gets

Countries With Higher MPIs that have less than 3 partners

```
SELECT p.Country, r.MPI, COUNT(p.PartnerID) AS NumOfPartners
FROM Regions r JOIN Partners p ON r.RegionID=p.RegionID
GROUP BY p.Country, r.MPI
HAVING COUNT(p.PartnerID) < 3 AND r.MPI > 0.33

--14 Number of loans that these countries in 13 Have
SELECT country, count(*) AS NumLoans FROM loans GROUP BY country having country in
('Timor-Leste', 'Burundi', 'Liberia', 'Somalia', 'Burkina Faso', 'South Sudan')
```

100 %

Results Messages

	Country	MPI	NumOfPartners
1	Cote D'Ivoire	0.348	1
2	Timor-Leste	0.373	1
3	Burundi	0.412	1
4	Liberia	0.434	1
5	Somalia	0.472	1
6	Burkina Faso	0.548	2
7	South Sudan	0.55	1

	country	NumLoans
1	Burkina Faso	2460
2	Burundi	880
3	Liberia	3682
4	Somalia	75
5	South Sudan	160
6	Timor-Leste	2690

Data Analysis

Analysis 4: Investigate if there is a correlation between the number of partners a country has and the number of loans that country gets

Number of Loans of Countries With more than 10 partners

```
Capstone2Queries.s...K4PBESE\Terry (65)  X
-- 17 MPIs of These countries having more than 10 partners
SELECT p.Country, r.MPI, COUNT(p.PartnerID) AS NumOfPartners
FROM Regions r JOIN Partners p ON r.RegionID=p.RegionID
GROUP BY p.Country, r.MPI
HAVING COUNT(p.PartnerID) > 10 AND r.MPI IS NOT NULL

--18 Number of Loans for countries countries that have large number of partners have
SELECT country, count(*) FROM loans GROUP BY country having country in
('Mexico', 'Peru', 'Indonesia', 'Cambodia', 'Ghana', 'Kenya', 'Uganda')

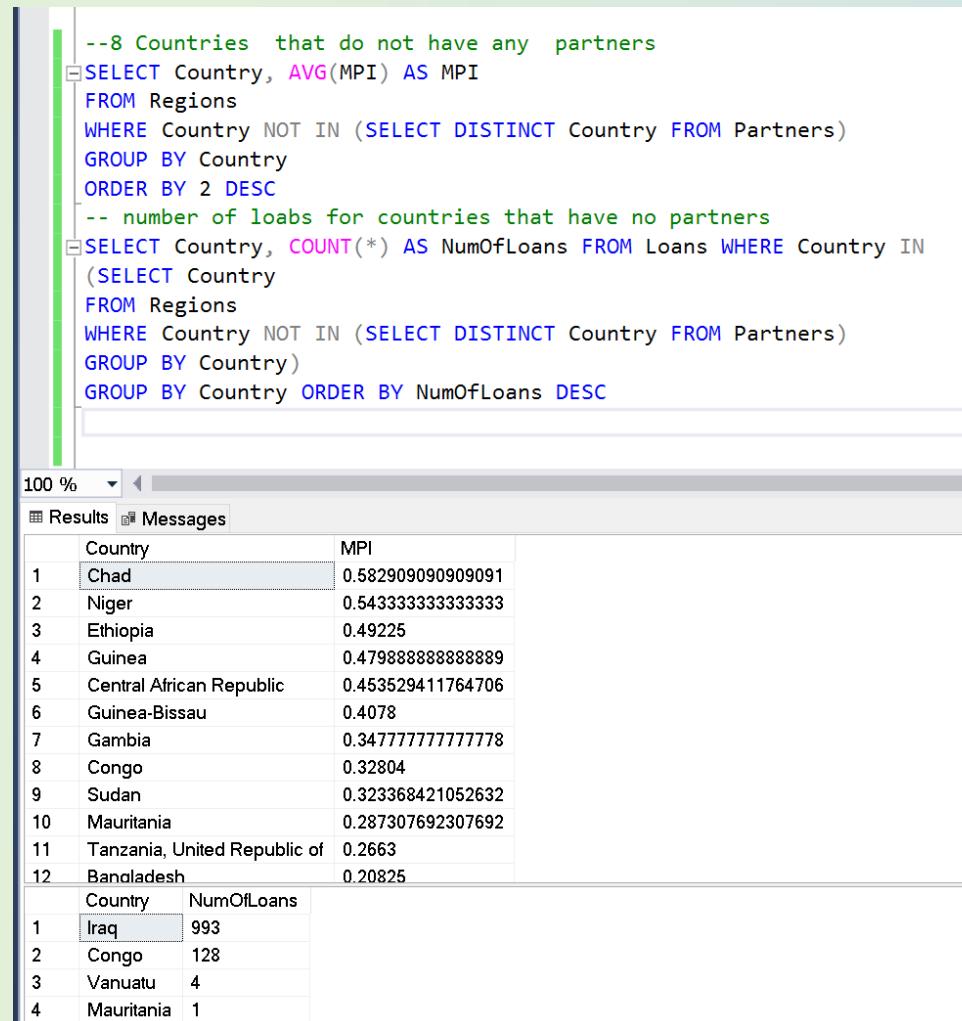
100 %  ◀
Results Messages
```

	Country	MPI	NumOfPartners
1	Mexico	0.008	12
2	Peru	0.055	13
3	Indonesia	0.084	12
4	Cambodia	0.164	11
5	Ghana	0.183	16
6	Kenya	0.209	30
7	Uganda	0.375	15

	country	(No column name)
1	Peru	22233
2	Kenya	75825
3	Uganda	20601
4	Ghana	4374
5	Mexico	5741
6	Indonesia	6214
7	Cambodia	34836

Data Analysis

Analysis 5: Investigate if there are countries that have high MPIs that have no partners and no loans



The screenshot shows a SQL query being run in a database environment. The query consists of two parts:

```
--8 Countries that do not have any partners
SELECT Country, AVG(MPI) AS MPI
FROM Regions
WHERE Country NOT IN (SELECT DISTINCT Country FROM Partners)
GROUP BY Country
ORDER BY 2 DESC

-- number of loans for countries that have no partners
SELECT Country, COUNT(*) AS NumOfLoans FROM Loans WHERE Country IN
(SELECT Country
FROM Regions
WHERE Country NOT IN (SELECT DISTINCT Country FROM Partners)
GROUP BY Country)
GROUP BY Country ORDER BY NumOfLoans DESC
```

The results are displayed in two tables:

Country	MPI
Chad	0.582909090909091
Niger	0.543333333333333
Ethiopia	0.49225
Guinea	0.479888888888889
Central African Republic	0.453529411764706
Guinea-Bissau	0.4078
Gambia	0.347777777777778
Congo	0.32804
Sudan	0.323368421052632
Mauritania	0.287307692307692
Tanzania, United Republic of	0.2663
Bangladesh	0.20825

Country	NumOfLoans
Iraq	993
Congo	128
Vanuatu	4
Mauritania	1

Data Analysis

Analysis 6: Investigate if the loans given to countries are increasing over the years

Number of Loans Over the Years for Lower MPI Countries

```
--Number of loans for countries by year (Low MPI Countries)
SELECT YEAR(LoanDate), Country, COUNT(*) AS NumOfLoans FROM Loans
WHERE Country IN
(SELECT Country FROM Regions GROUP BY Country HAVING AVG(MPI) < 0.33)
GROUP BY Country, LoanDate
--ORDER BY NumOfLoans DESC
ORDER BY YEAR(LoanDate), NumOfLoans DESC
```

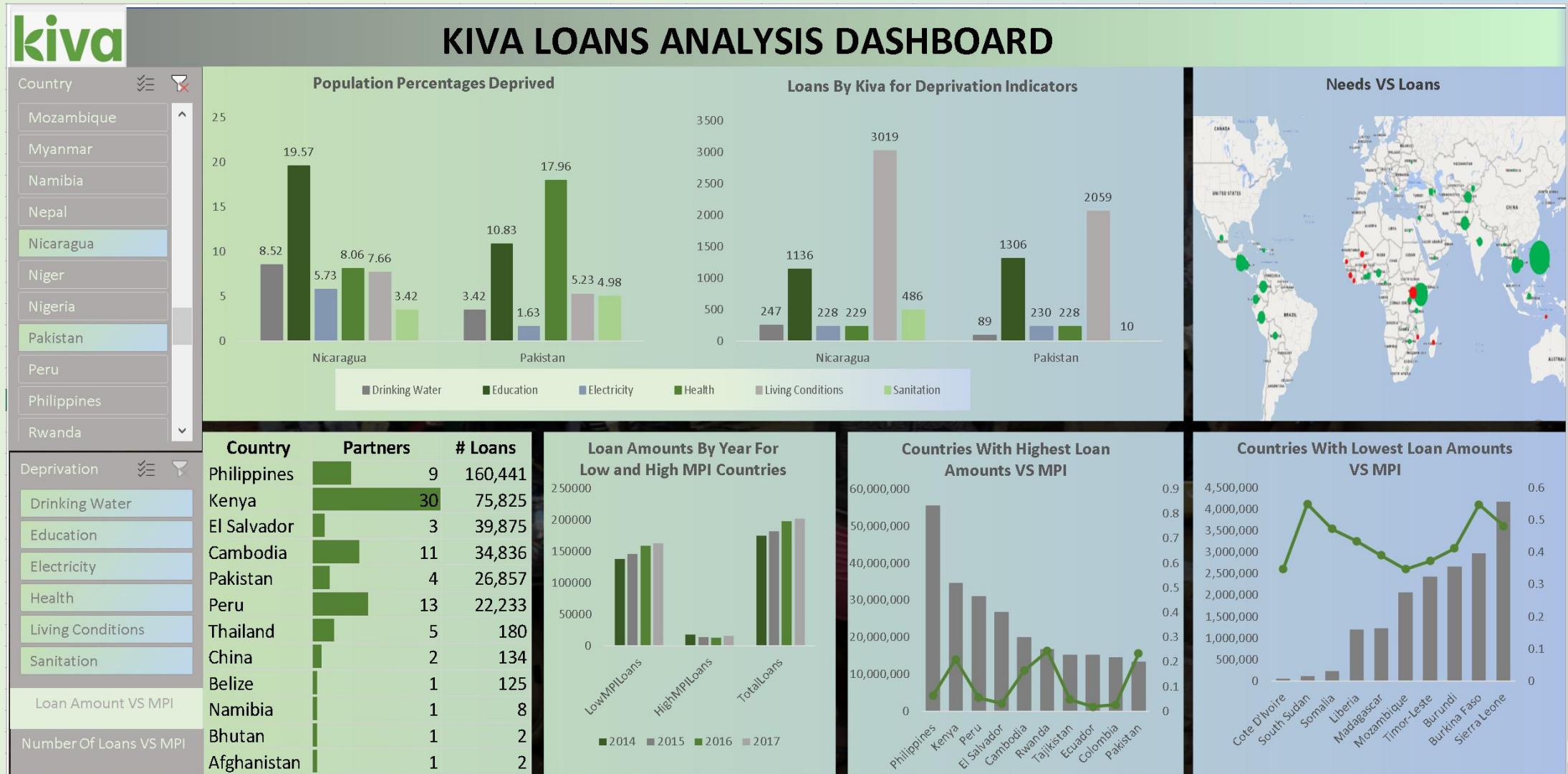
(No column name)	Country	NumOfLoans	
7264	2014	Armenia	1
7265	2014	Vietnam	1
7266	2014	Kenya	1
7267	2014	Ukraine	1
7268	2014	Mongolia	1
7269	2014	Ukraine	1
7270	2014	Tajikistan	1
7271	2014	Colombia	1
7272	2014	El Salvad...	1
7273	2014	Kenya	1
7274	2014	Mongolia	1
7275	2015	Philippines	369
7276	2015	Philippines	356
7277	2015	Philippines	341
7278	2015	Philippines	337
7279	2015	Philippines	334
7280	2015	Philippines	331
7281	2015	Philippines	330
7282	2015	Philippines	286
7283	2015	Philippines	283
7284	2015	Philippines	279
7285	2015	Philippines	277
7286	2015	Philippines	276
7287	2015	Philippines	275
7288	2015	Philippines	273
7289	2015	Philippines	272

Number of Loans Over the Years for Higher MPI Countries

```
--Number of loans for countries by year (High MPI Countries)
SELECT YEAR(LoanDate), Country, COUNT(*) AS NumOfLoans FROM Loans
WHERE Country IN
(SELECT Country FROM Regions GROUP BY Country HAVING AVG(MPI) >= 0.33)
GROUP BY Country, LoanDate
--ORDER BY NumOfLoans DESC
ORDER BY YEAR(LoanDate), NumOfLoans DESC
```

(No column name)	Country	NumOfLoans	
2884	2015	Mali	1
2885	2015	Liberia	1
2886	2015	Burundi	1
2887	2015	Mada...	1
2888	2015	Sene...	1
2889	2015	Mada...	1
2890	2015	Mali	1
2891	2015	Mada...	1
2892	2015	Sene...	1
2893	2015	Mada...	1
2894	2015	Uganda	1
2895	2015	Burundi	1
2896	2015	Timor...	1
2897	2015	Mada...	1
2898	2015	Burundi	1
2899	2016	Uganda	76
2900	2016	Uganda	71
2901	2016	Mali	67
2902	2016	Moza...	66
2903	2016	Mada...	63
2904	2016	Moza...	58
2905	2016	Uganda	58
2906	2016	Uganda	55
2907	2016	Uganda	54
2908	2016	Uganda	52
2909	2016	Mali	52

Dashboard



Insights

1. There were numerous instances when the number of loans that a country receives does not match the most severe deprivation(s) that the country faces. For example, Nicaragua receives many loans for education but the highest deprivation that the people there face is poor Living Conditions
2. This analysis revealed that the countries getting the most loans and the highest collective loan amounts are not the countries with the highest MPIs. In fact there are a small number of countries that get a disproportionately large amount of loans.
3. Over the years (2014-2017), the total number of loans given out by Kiva are increasing steadily. However the loans received by the countries which already have the most loans are also increasing steadily whereas the countries with the highest MPIs showed some decrease in 2015 and 2016 and a minuscule increase in 2017. The trend of the number of loans received over the years for the high MPI countries do not reflect the overall increasing loan trend of Kiva
4. Countries that have more partners tend to have more loans. The data showed that there are many high MPI countries with no partners or just 1 or 2 partners and that contributed to these needy countries not receiving ample loans

Recommendations for Loans to have greater impact

1. Currently how loans are listed on the Kiva website are not based on the needs and deprivations of the countries. Algorithms can be developed to check on the deprivation and MPI of the country when advertising loans on the first few pages of the listed loans when lenders visit the website. When the lender wants to lend to a certain country, then the most severe deprivation can be listed as the first few loans for the lender to choose from. Similarly, if a lender wants to lend to a sector, then the loans in that sector can be matched with the countries with the highest number of people deprived in that corresponding sector, and those loans can be listed first
2. There should be a rating system when loans are given per country and sector, taking into consideration the number of loans the country has already received, the MPI of the country and the percentage of population deprived. This will ensure that the poorer countries get more loans and that a small group of countries do not get the majority of loans
3. Partners working with Kiva in a country does increase the chances of its people getting more loans. Kiva should actively try to find partners and galvanize them for countries with higher MPIs and higher population percentages that are deprived. Kiva should engage partners who can work with Kiva to screen borrowers, post loan requests to Kiva for funding, disburse loans on the ground, collect repayments and advise borrowers on their entrepreneurial ventures to ensure average repayment rates of loans are maintained