Food Prices in Zambia Markets Data Analysis using SQL

Background/Scenario

The essence of this data analysis project was to look at the different food prices which were sold in selected Zambian markets. The food prices data covers foods such as maize, rice, beans, sorghum, etc. The food prices are based on the trading market in which the commodity was sold. The dataset contained data from 15/01/2003 to 15/12/2018.

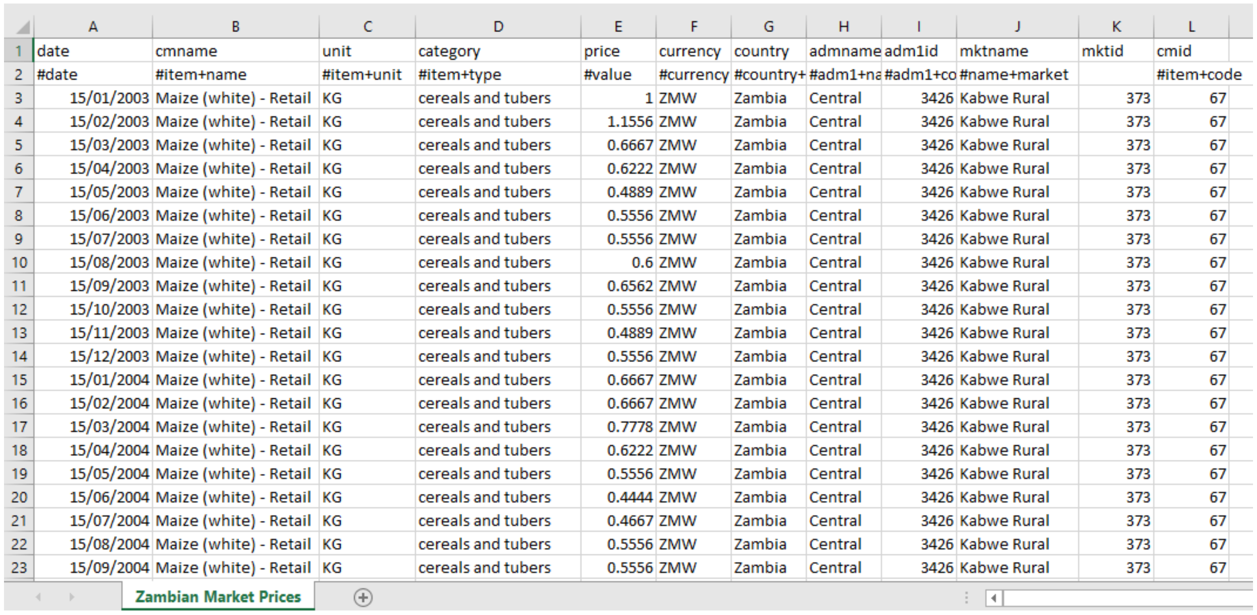
Project Approach

For this project, a dataset from the World Food Programme in collaboration with Humanitarian Data Exchange the was used. The preparation of the data in the csv file was done using **Microsoft Excel.** Thereafter, the dataset was imported into **MySQL Workbench** for exploratory data analysis using **SQL**. Finally, visualizations of the analyzed data were created using **Tableau**.

Part 1: Prepare the data

Before the csv file can be imported into MySQL Workbench it has to be prepared. Good data preparation allows for efficient data analysis, limits errors and inaccuracies that can occur to data during analysis. The data preparation was done using Microsoft Excel.

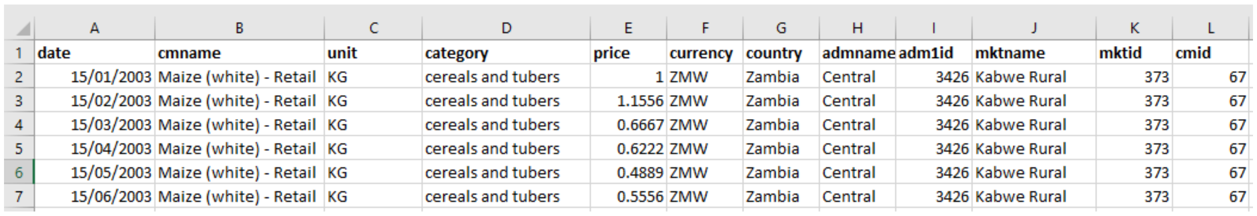
Below is a snapshot of a portion of the dataset.



	A	B	C	D	E	F	G	H	I	J	K	L
1	date	cmname	unit	category	price	currency	country	admname	adm1id	mktname	mktid	cmid
2	#date	#item+name	#item+unit	#item+type	#value	#currency	#country+	#adm1+na	#adm1+co	#name+market		#item+code
3	15/01/2003	Maize (white) - Retail	KG	cereals and tubers	1	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
4	15/02/2003	Maize (white) - Retail	KG	cereals and tubers	1.1556	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
5	15/03/2003	Maize (white) - Retail	KG	cereals and tubers	0.6667	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
6	15/04/2003	Maize (white) - Retail	KG	cereals and tubers	0.6222	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
7	15/05/2003	Maize (white) - Retail	KG	cereals and tubers	0.4889	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
8	15/06/2003	Maize (white) - Retail	KG	cereals and tubers	0.5556	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
9	15/07/2003	Maize (white) - Retail	KG	cereals and tubers	0.5556	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
10	15/08/2003	Maize (white) - Retail	KG	cereals and tubers	0.6	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
11	15/09/2003	Maize (white) - Retail	KG	cereals and tubers	0.6562	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
12	15/10/2003	Maize (white) - Retail	KG	cereals and tubers	0.5556	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
13	15/11/2003	Maize (white) - Retail	KG	cereals and tubers	0.4889	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
14	15/12/2003	Maize (white) - Retail	KG	cereals and tubers	0.5556	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
15	15/01/2004	Maize (white) - Retail	KG	cereals and tubers	0.6667	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
16	15/02/2004	Maize (white) - Retail	KG	cereals and tubers	0.6667	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
17	15/03/2004	Maize (white) - Retail	KG	cereals and tubers	0.7778	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
18	15/04/2004	Maize (white) - Retail	KG	cereals and tubers	0.6222	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
19	15/05/2004	Maize (white) - Retail	KG	cereals and tubers	0.5556	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
20	15/06/2004	Maize (white) - Retail	KG	cereals and tubers	0.4444	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
21	15/07/2004	Maize (white) - Retail	KG	cereals and tubers	0.4667	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
22	15/08/2004	Maize (white) - Retail	KG	cereals and tubers	0.5556	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
23	15/09/2004	Maize (white) - Retail	KG	cereals and tubers	0.5556	ZMW	Zambia	Central	3426	Kabwe Rural	373	67

Step 1: Delete the second row

The second row of the dataset had to be deleted as it contained duplicate data of the column headers.



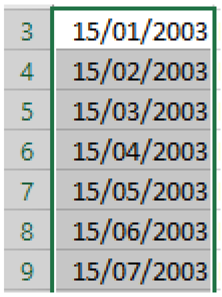
	A	B	C	D	E	F	G	H	I	J	K	L
1	date	cmname	unit	category	price	currency	country	admname	adm1id	mktname	mktid	cmid
2	15/01/2003	Maize (white) - Retail	KG	cereals and tubers	1	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
3	15/02/2003	Maize (white) - Retail	KG	cereals and tubers	1.1556	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
4	15/03/2003	Maize (white) - Retail	KG	cereals and tubers	0.6667	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
5	15/04/2003	Maize (white) - Retail	KG	cereals and tubers	0.6222	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
6	15/05/2003	Maize (white) - Retail	KG	cereals and tubers	0.4889	ZMW	Zambia	Central	3426	Kabwe Rural	373	67
7	15/06/2003	Maize (white) - Retail	KG	cereals and tubers	0.5556	ZMW	Zambia	Central	3426	Kabwe Rural	373	67

Step 2: Change all columns with dates to 'yyyy-mm-dd' format.

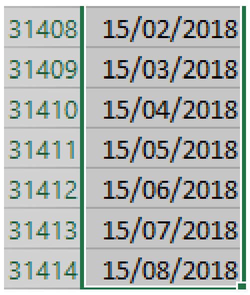
The csv file contained a field named 'date', the data in this field had the 'DD/MM/YYYY' format. This had to be changed to 'YYYY/MM/DD', as this is the accepted MySQL format.

Step 2.1

Having opened the csv file using Microsoft Excel, select the date cells in the cell range A3:A31414 using the keyboard shortcut Shift + Ctrl + down arrow.



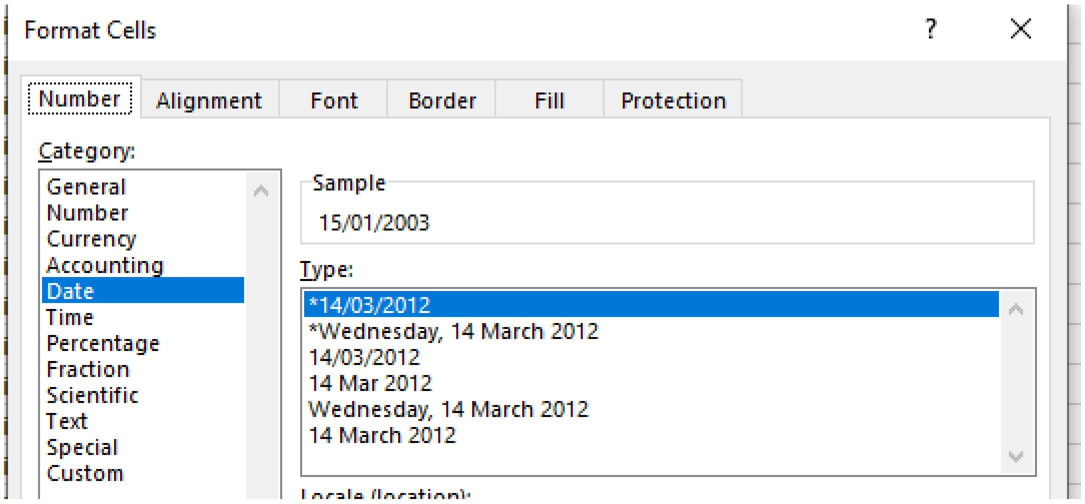
3	15/01/2003
4	15/02/2003
5	15/03/2003
6	15/04/2003
7	15/05/2003
8	15/06/2003
9	15/07/2003



31408	15/02/2018
31409	15/03/2018
31410	15/04/2018
31411	15/05/2018
31412	15/06/2018
31413	15/07/2018
31414	15/08/2018

Step 2.2

The 'Format Cells' dialogue box was activated using the keyboard shortcut Ctrl + 1.



Format Cells

Number | Alignment | Font | Border | Fill | Protection

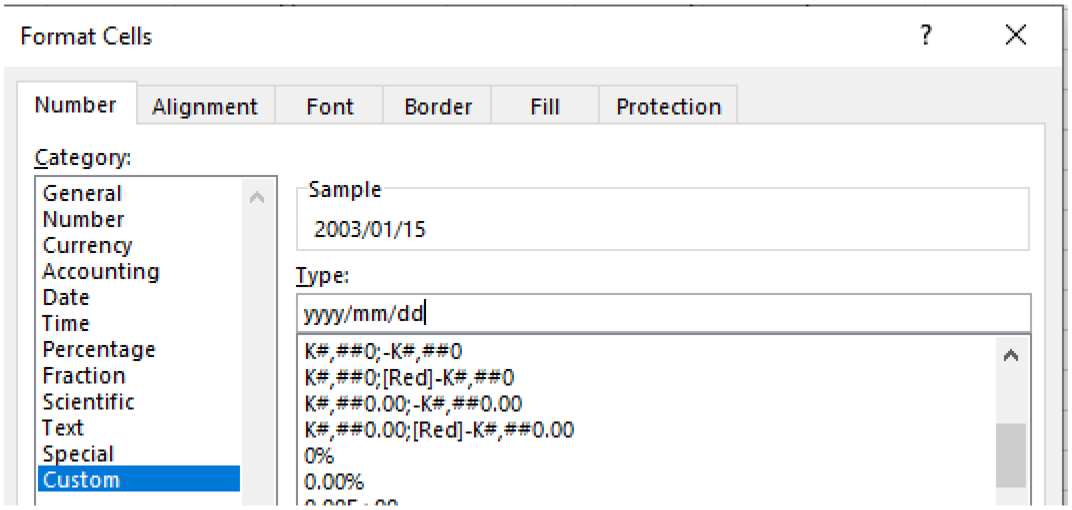
Category:
- General
- Number
- Currency
- Accounting
- Date
- Time
- Percentage
- Fraction
- Scientific
- Text
- Special
- Custom

Sample
15/01/2003

Type:
- *14/03/2012
- *Wednesday, 14 March 2012
- 14/03/2012
- 14 Mar 2012
- Wednesday, 14 March 2012
- 14 March 2012

Locale (location):

Step 2.3
The 'Custom' Category was selected and the 'yyyy/mm/dd' format was entered in the 'Type:' combo box. Thereafter, click the 'OK' button.

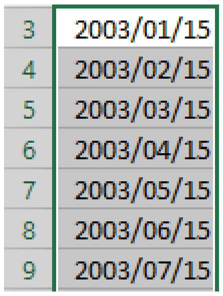


2003/01/15

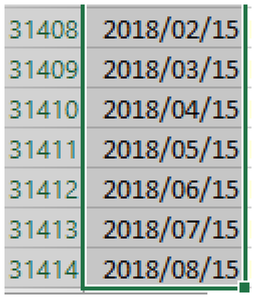
yyyy/mm/dd

K#,##0;-K#,##0
K#,##0;[Red]-K#,##0
K#,##0.00;-K#,##0.00
K#,##0.00;[Red]-K#,##0.00
0%
0.00%

Step 2.4
The date format was successfully converted from 'dd/mm/yyyy' to 'yyyy/mm/dd' as can be seen below.



3	2003/01/15
4	2003/02/15
5	2003/03/15
6	2003/04/15
7	2003/05/15
8	2003/06/15
9	2003/07/15



31408	2018/02/15
31409	2018/03/15
31410	2018/04/15
31411	2018/05/15
31412	2018/06/15
31413	2018/07/15
31414	2018/08/15

Step 3: Check for other format issues

1. Apart from changing the date format, there was a need to check for blank cells. There weren't any blank cells in the dataset.
2. Additionally, the dataset had to be checked if it contained cells containing currency e.g. '$100'. Fortunately, there weren't any cells having currency in the '$100' format.
3. The dataset also had to be inspected for cells containing numeric data with the thousandth separator, for example, 1,000. MySQL treats the comma in 1,000 as a delimiter, as such any cells containing such data has to be change to, for example, 1000 and not 1,000. The dataset used did not contain any numeric data such as '1,000'.

Step 4: Save the csv file in the folder containing the database.

In this case, the Zambian Market Prices csv file is saved in the following directory: ‘C:\Program Data\MySQL\MySQL Server 8.0\Data\Zambian Markets Data Analysis\Zambian Market Prices.csv’

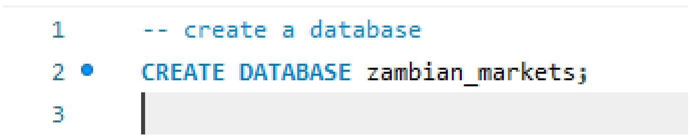
Having prepared the dataset, it was now time to import the data.

Part 2: Import the Data

After the dataset has been downloaded from data.world has been downloaded, it’s now time to import it into MySQL Workbench. For Part 2, the tasks will be carried out using SQL.

Step 1: Create a database

First off, there’s need to have a database where the data will be imported into. The database can be created from scratch or an existing one can be used. For this project, a new database will be created using MySQL.

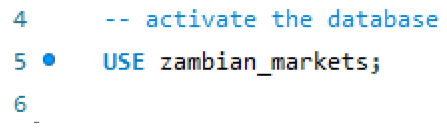


```
1      -- create a database
2 ●    CREATE DATABASE zambian_markets;
3
```

A new database named ‘zambian_markets’ has been created. The two - - hyphens at the beginning of line 1 indicate that it is a comment and it doesn’t affect the code.

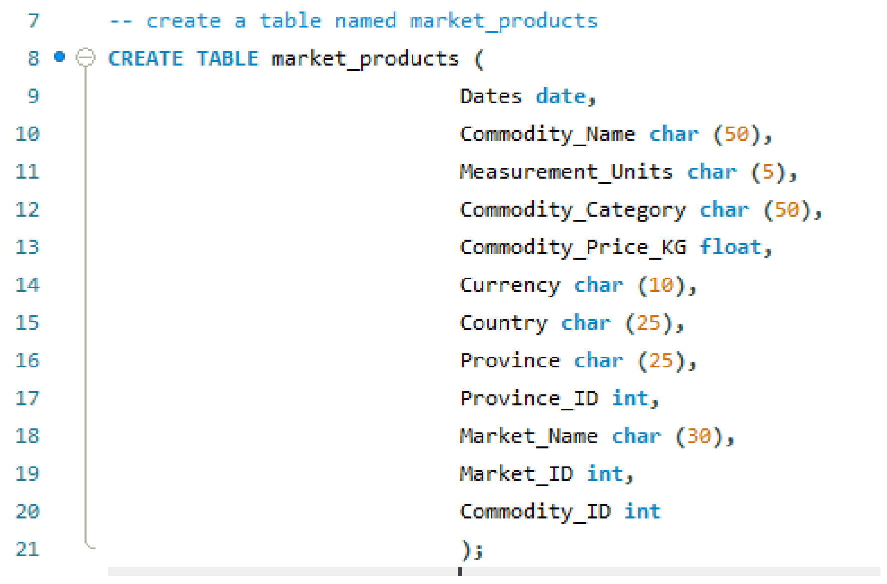
Step 2: Activate the database

Next, to get to work on a specific database there’s need to select it, and to do that the ‘use’ keyword is typed.



```
4      -- activate the database
5 ●    USE zambian_markets;
6
```

After this, we need to create a table in the database that will fit the data and match it. The SQL statement below was used to create a table named ‘market_products’.

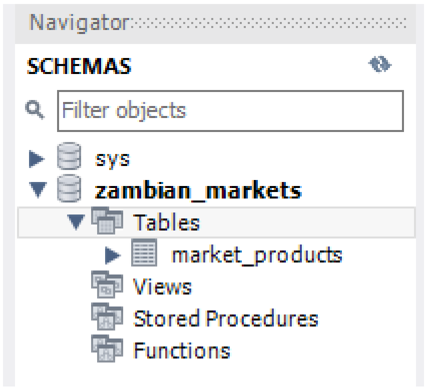


```
-- create a table named market_products
CREATE TABLE market_products (
                              Dates date,
                              Commodity_Name char (50),
                              Measurement_Units char (5),
                              Commodity_Category char (50),
                              Commodity_Price_KG float,
                              Currency char (10),
                              Country char (25),
                              Province char (25),
                              Province_ID int,
                              Market_Name char (30),
                              Market_ID int,
                              Commodity_ID int
                              );
```

Here, a table called ‘market_products’ was created, and designed in a way to fit the csv dataset by matching the columns and data types. It’s done column by column as in the csv file and in the same order when building the ‘market_products’ table.

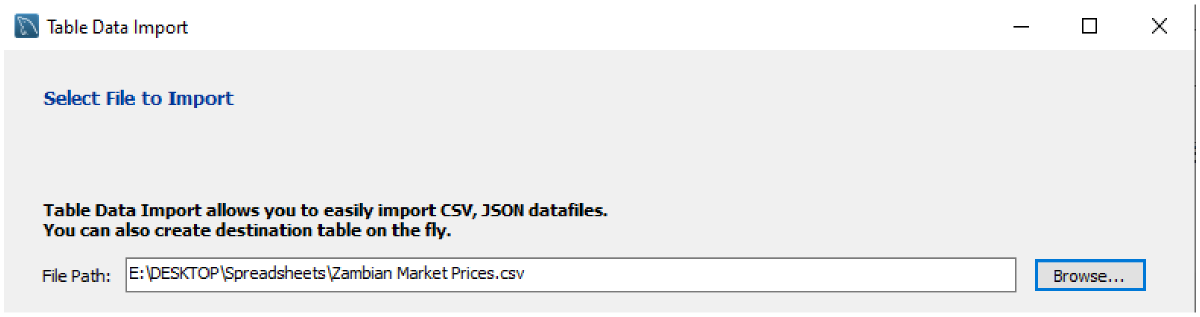
The columns are created in this manner (in example above): first off, specify the column name, then add the data type of that column and in the parenthesis the size of the variable is entered.
It’s imperative to check the data in the csv file before specifying the field sizes in the database table. For example, by specifying that city, or state, or whatever field it is, is of max size of 20 means that rows which contain fields that have more than 20 characters will not be imported.

To confirm if the table has been created, go to the left panel of the MySQL Workbench; double click on the database (zambian_markets) name in the left panel and then right click on the table recently created: If the created table does not appear in the left panel, click the refresh button next to ‘SCHEMAS’.



Step 4: Import the csv file

To import the data from the csv file, place the cursor over the table and right click. From the drop - down menu, select the 'Table Data Import Wizard' option. After selecting the needed option, a 'Table Data Import' window will open pop up. This window allows us to browse for the csv file that has to be imported. Look it up and then when the file is located click on next button.

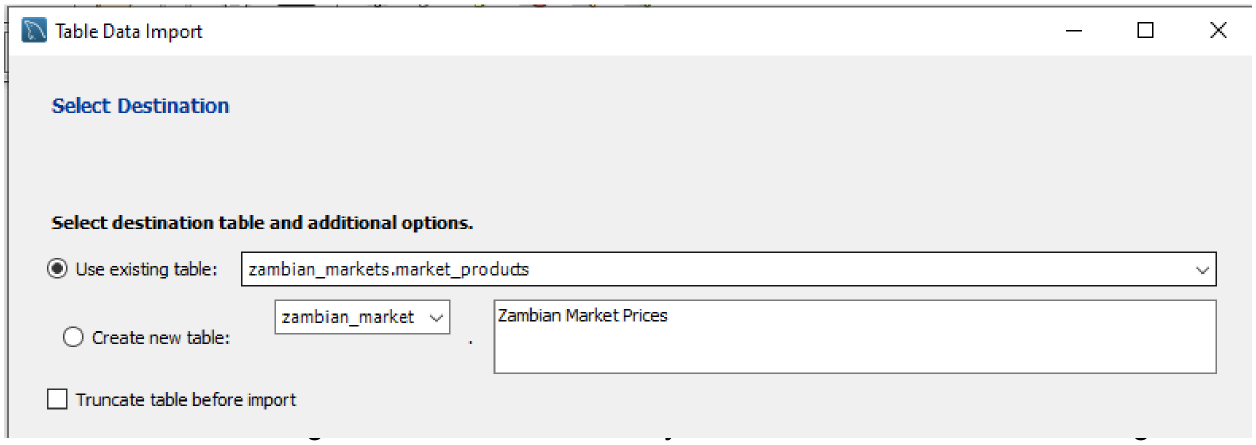


Select File to Import

Table Data Import allows you to easily import CSV, JSON datafiles.
You can also create destination table on the fly.

File Path: E:\DESKTOP\Spreadsheets\Zambian Market Prices.csv

After clicking 'Next >', MySQL will ask for the destination table where the data from the csv file will go to:



Select Destination

Select destination table and additional options.

(●) Use existing table: zambian_markets.market_products

( ) Create new table: zambian_market . Zambian Market Prices

[ ] Truncate table before import

and then press next. Next, ensure the source and destination columns match. Where necessary make changes using the drop down arrow. When the columns match, click on the next button and finally the finish button when the dataset has been loaded into MySQL.

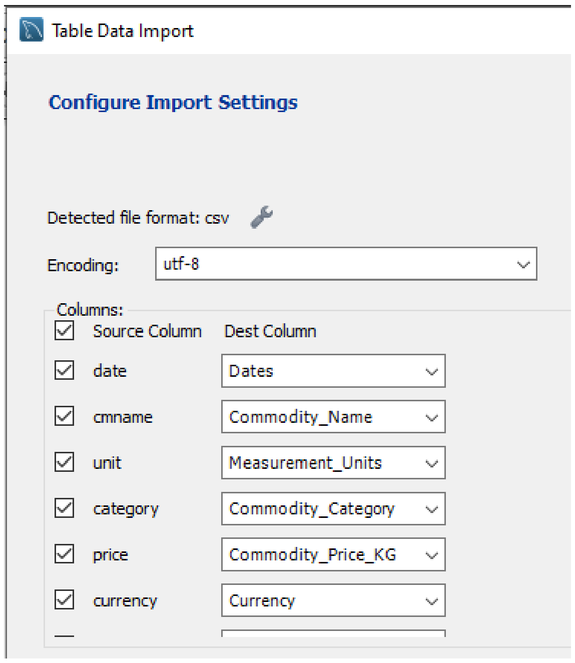


Table Data Import

Configure Import Settings

Detected file format: csv

Encoding: utf-8

Columns:

☑ Source Column	Dest Column
☑ date	Dates
☑ cmname	Commodity_Name
☑ unit	Measurement_Units
☑ category	Commodity_Category
☑ price	Commodity_Price_KG
☑ currency	Currency

After clicking the ‘Next >’ button, the window shown below appears.

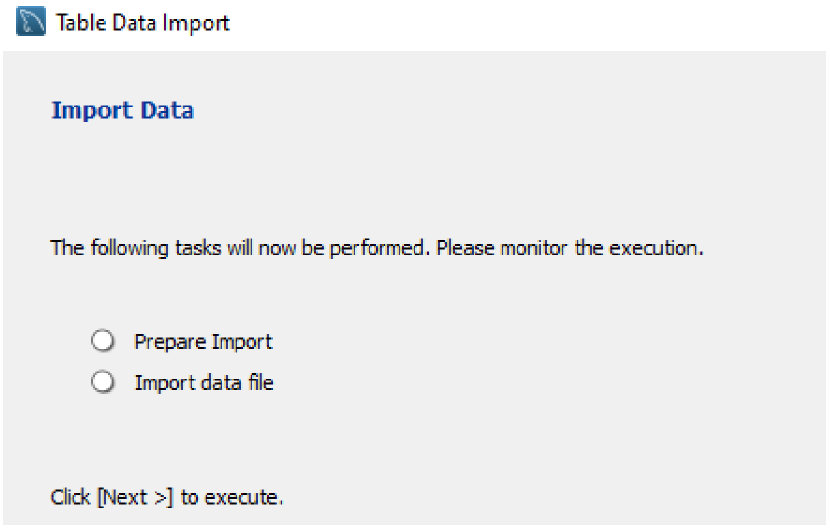


Table Data Import

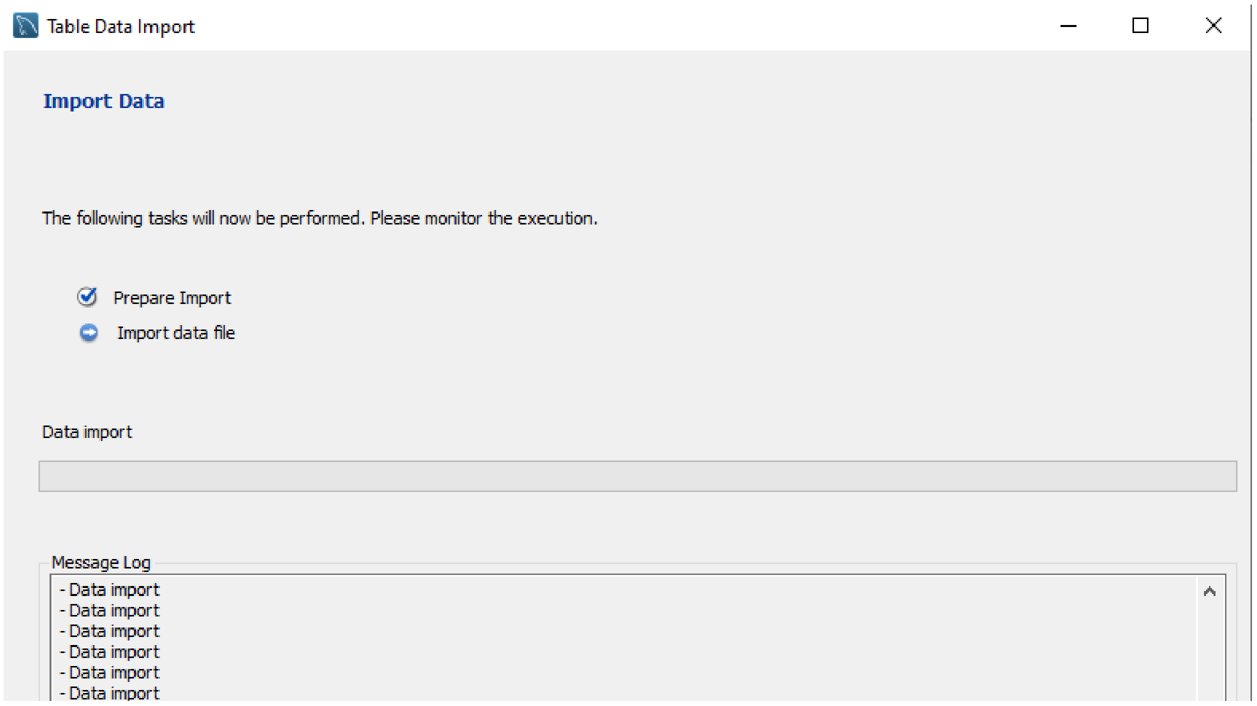
Import Data

The following tasks will now be performed. Please monitor the execution.

○ Prepare Import
○ Import data file

Click [Next >] to execute.

The data importation begins as soon as the ‘Next >’ button is clicked.



The following tasks will now be performed. Please monitor the execution.

- ✓ Prepare Import
- ● Import data file

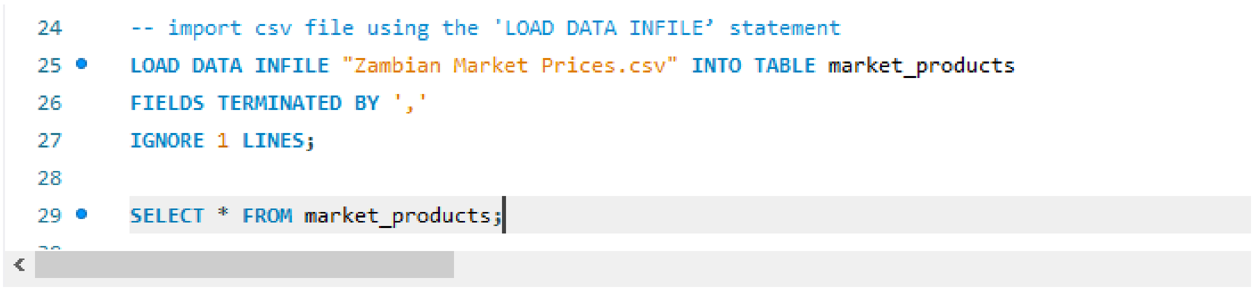
Data import

Message Log

```
- Data import
- Data import
- Data import
- Data import
- Data import
- Data import
```

Unfortunately, the data import process was painstaking slow as the csv file was very large having over 30,000 records and 12 fields. The ‘Table Data Import’ method in MySQL as depicted above is ideal if the csv or excel file is not very large. In view of this predicament, the data import process was cancelled and a different method had to be used.

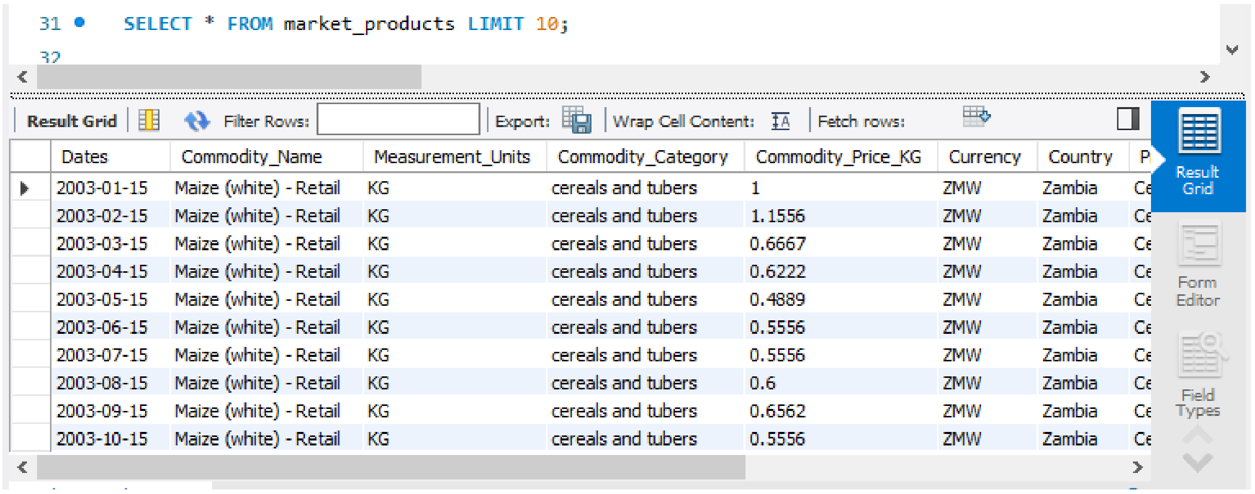
The ‘LOAD DATA INFILE’ SQL statement was used to import the data as using this method imports a large dataset quicker. After executing the ‘LOAD DATA INFILE’ statement, an error appeared.



```
-- import csv file using the 'LOAD DATA INFILE' statement
LOAD DATA INFILE "Zambian Market Prices.csv" INTO TABLE market_products
FIELDS TERMINATED BY ','
IGNORE 1 LINES;

SELECT * FROM market_products;
```

After successful execution of the statement above all the records are imported successfully. As, the table is way too big to see all of the records, so let's see a portion of the first 10 rows using the SQL statement below.



```sql
SELECT * FROM market_products LIMIT 10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Dates	Commodity_Name	Measurement_Units	Commodity_Category	Commodity_Price_KG	Currency	Country	P
2003-01-15	Maize (white) - Retail	KG	cereals and tubers	1	ZMW	Zambia	Ce
2003-02-15	Maize (white) - Retail	KG	cereals and tubers	1.1556	ZMW	Zambia	Ce
2003-03-15	Maize (white) - Retail	KG	cereals and tubers	0.6667	ZMW	Zambia	Ce
2003-04-15	Maize (white) - Retail	KG	cereals and tubers	0.6222	ZMW	Zambia	Ce
2003-05-15	Maize (white) - Retail	KG	cereals and tubers	0.4889	ZMW	Zambia	Ce
2003-06-15	Maize (white) - Retail	KG	cereals and tubers	0.5556	ZMW	Zambia	Ce
2003-07-15	Maize (white) - Retail	KG	cereals and tubers	0.5556	ZMW	Zambia	Ce
2003-08-15	Maize (white) - Retail	KG	cereals and tubers	0.6	ZMW	Zambia	Ce
2003-09-15	Maize (white) - Retail	KG	cereals and tubers	0.6562	ZMW	Zambia	Ce
2003-10-15	Maize (white) - Retail	KG	cereals and tubers	0.5556	ZMW	Zambia	Ce

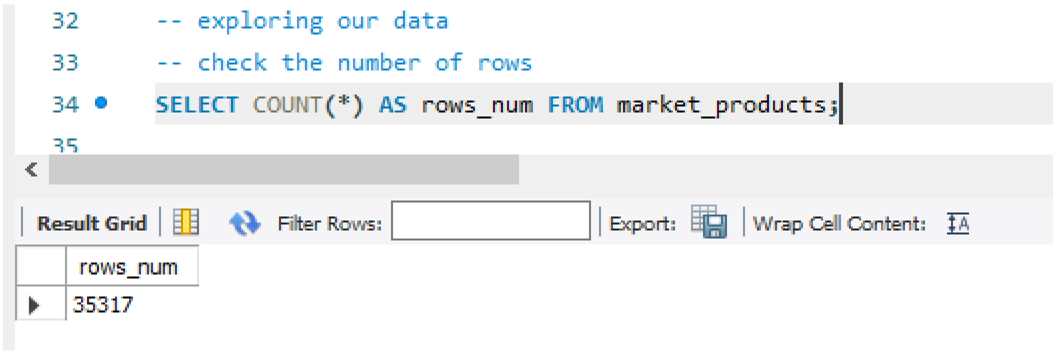
Part 3: Clean the data

The dataset was clean as such no data cleaning was required.

Part 4: Exploratory Data Analysis (EDA)

Step 1: Number of columns and rows in the 'market_products' table

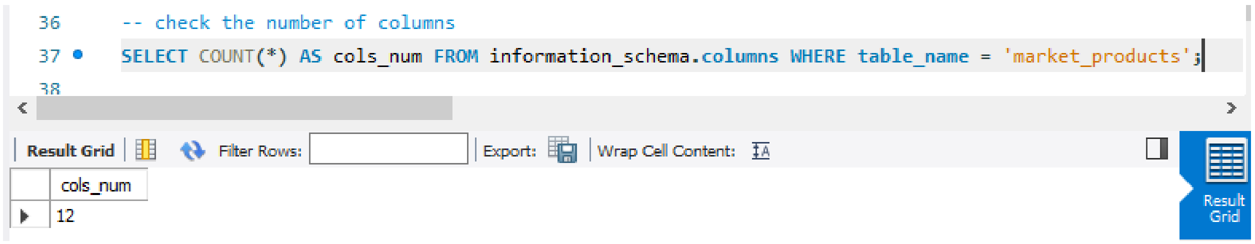
What is the shape of the database table, i.e., the number of columns and rows? To answer this question, the following SQL statements were used:



```sql
-- exploring our data
-- check the number of rows
SELECT COUNT(*) AS rows_num FROM market_products;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

rows_num
35317



```sql
-- check the number of columns
SELECT COUNT(*) AS cols_num FROM information_schema.columns WHERE table_name = 'market_products';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

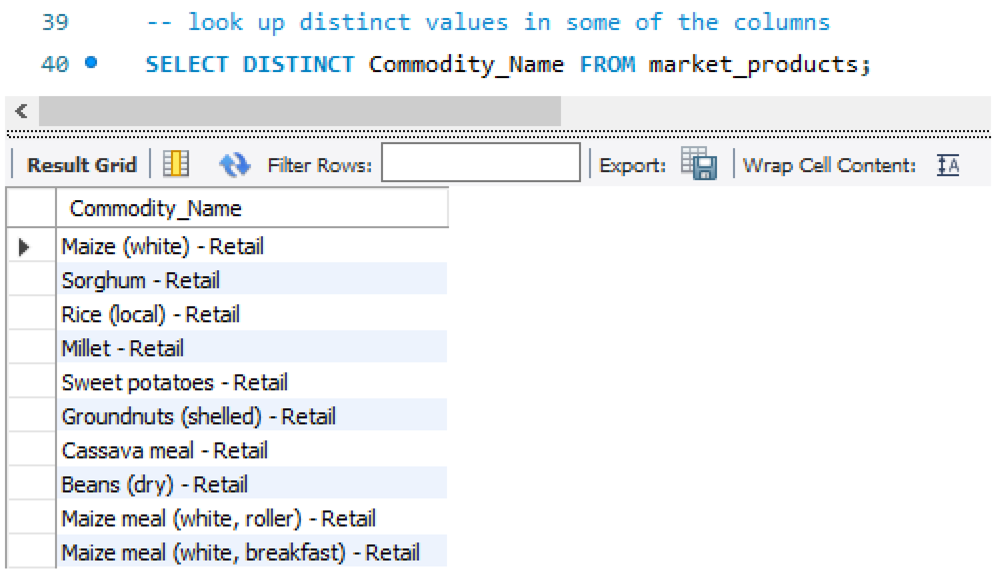
cols_num
12

The Zambian Market Prices table is made up of **35,317** rows and **12** columns.

Step 2: Distinct values

The SQL statements below were used to look up distinct values in some of the fields.

2.1 Commodity Name

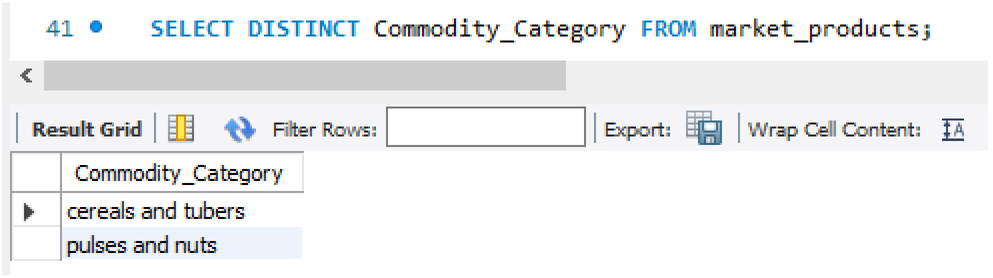


```
-- look up distinct values in some of the columns
SELECT DISTINCT Commodity_Name FROM market_products;
```

Commodity_Name
Maize (white) - Retail
Sorghum - Retail
Rice (local) - Retail
Millet - Retail
Sweet potatoes - Retail
Groundnuts (shelled) - Retail
Cassava meal - Retail
Beans (dry) - Retail
Maize meal (white, roller) - Retail
Maize meal (white, breakfast) - Retail

The Commodity Name field was made up of **ten** (**10**) different commodities.

2.2 Commodity Category

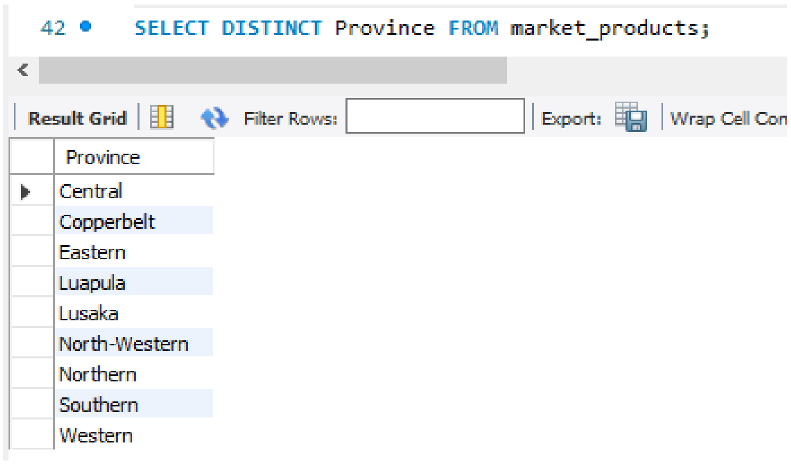


```
SELECT DISTINCT Commodity_Category FROM market_products;
```

Commodity_Category
cereals and tubers
pulses and nuts

There were only **two (2)** categories of the commodities sold in markets in the dataset under review.

2.3 Number of provinces

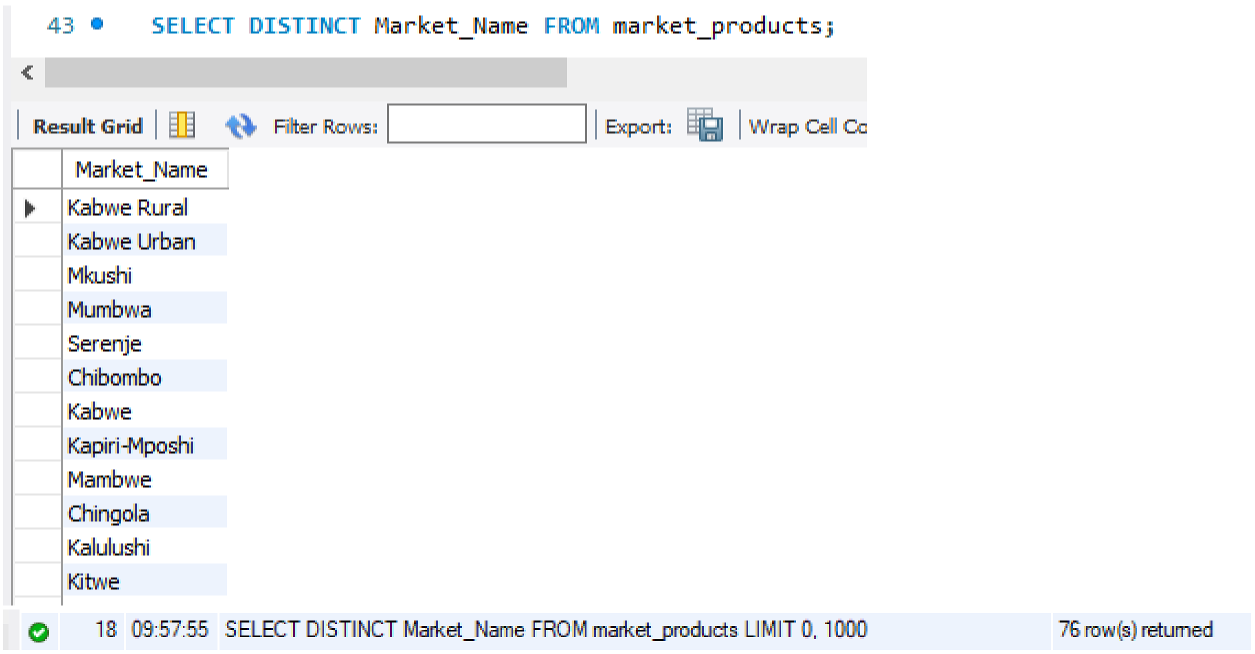


```
SELECT DISTINCT Province FROM market_products;
```

Province
Central
Copperbelt
Eastern
Luapula
Lusaka
North-Western
Northern
Southern
Western

The market products dataset had a total of **nine** (**9**) provinces.

2.4 Number of markets



```
SELECT DISTINCT Market_Name FROM market_products;
```

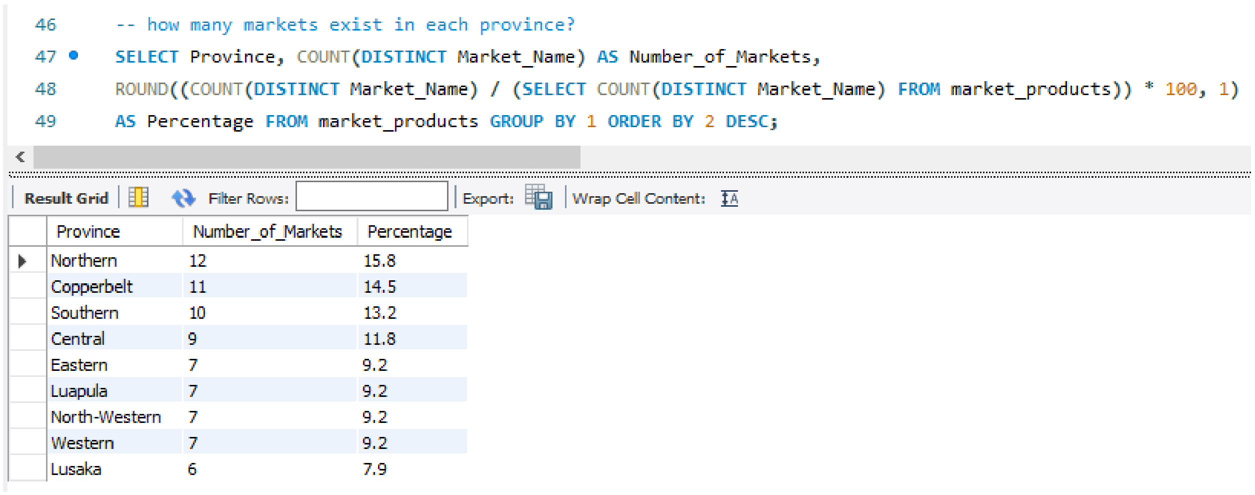
Market_Name
Kabwe Rural
Kabwe Urban
Mkushi
Mumbwa
Serenje
Chibombo
Kabwe
Kapiri-Mposhi
Mambwe
Chingola
Kalulushi
Kitwe

```
18  09:57:55  SELECT DISTINCT Market_Name FROM market_products LIMIT 0, 1000    76 row(s) returned
```

There were a total of **seventy-six** (**76**) markets in the dataset.

Step 3: Answering business questions?

Step 3.1: How many markets exist in each province?

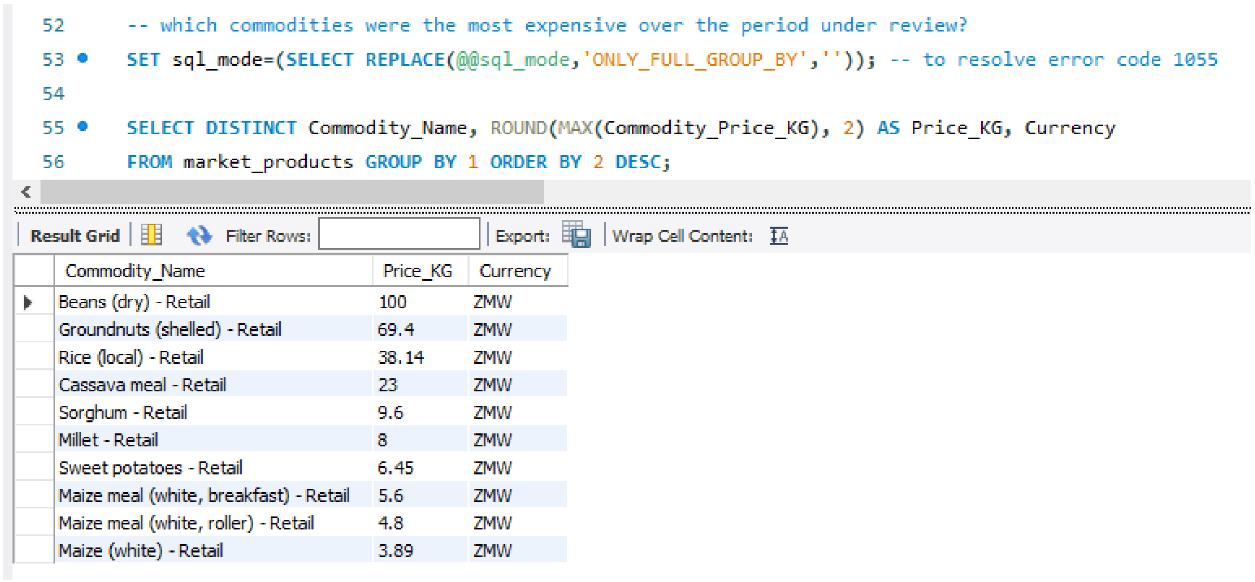


```
-- how many markets exist in each province?
SELECT Province, COUNT(DISTINCT Market_Name) AS Number_of_Markets,
ROUND((COUNT(DISTINCT Market_Name) / (SELECT COUNT(DISTINCT Market_Name) FROM market_products)) * 100, 1)
AS Percentage FROM market_products GROUP BY 1 ORDER BY 2 DESC;
```

Province	Number_of_Markets	Percentage
Northern	12	15.8
Copperbelt	11	14.5
Southern	10	13.2
Central	9	11.8
Eastern	7	9.2
Luapula	7	9.2
North-Western	7	9.2
Western	7	9.2
Lusaka	6	7.9

Based on the screenshot above, it is evident that **Northern** province had the highest number of markets in the country with **12** markets translating to **15.8%. Lusaka** province had the least number of markets at **6** which is **7.9%** of all the markets.

Step 3.2: Which commodity was the most expensive over the review period?



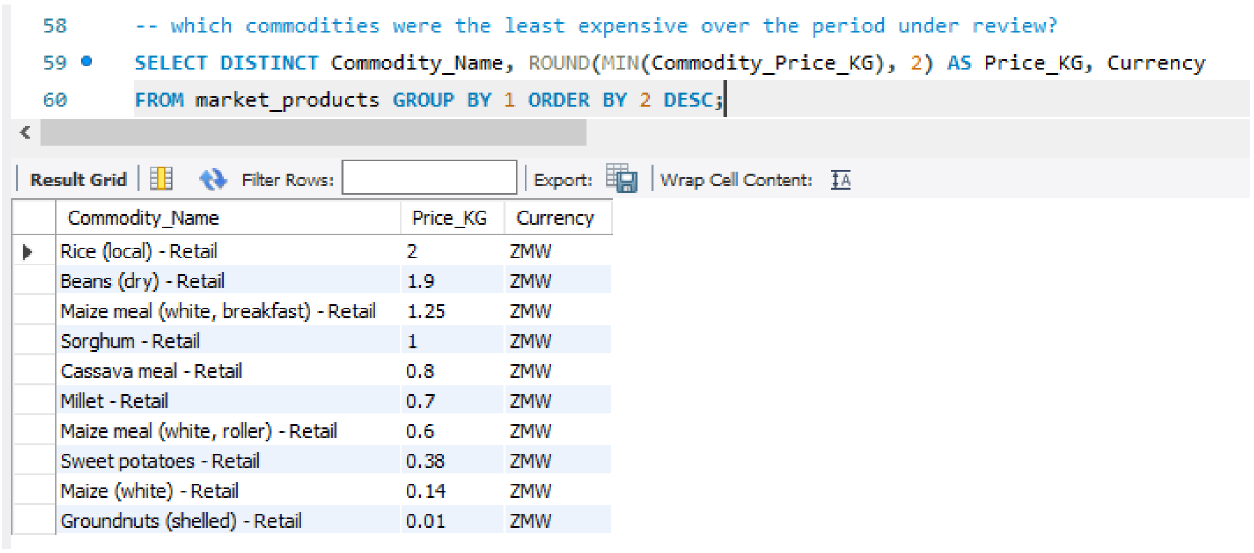
```
-- which commodities were the most expensive over the period under review?
SET sql_mode=(SELECT REPLACE(@@sql_mode,'ONLY_FULL_GROUP_BY','')); -- to resolve error code 1055

SELECT DISTINCT Commodity_Name, ROUND(MAX(Commodity_Price_KG), 2) AS Price_KG, Currency
FROM market_products GROUP BY 1 ORDER BY 2 DESC;
```

Commodity_Name	Price_KG	Currency
Beans (dry) - Retail	100	ZMW
Groundnuts (shelled) - Retail	69.4	ZMW
Rice (local) - Retail	38.14	ZMW
Cassava meal - Retail	23	ZMW
Sorghum - Retail	9.6	ZMW
Millet - Retail	8	ZMW
Sweet potatoes - Retail	6.45	ZMW
Maize meal (white, breakfast) - Retail	5.6	ZMW
Maize meal (white, roller) - Retail	4.8	ZMW
Maize (white) - Retail	3.89	ZMW

Based on the data in the screenshot above, **Beans (dry) – Retail** was the most expensive commodity in the period under review retailing at **K100** per kilogram.

Step 3.3: Which commodity was the least expensive over the review period?

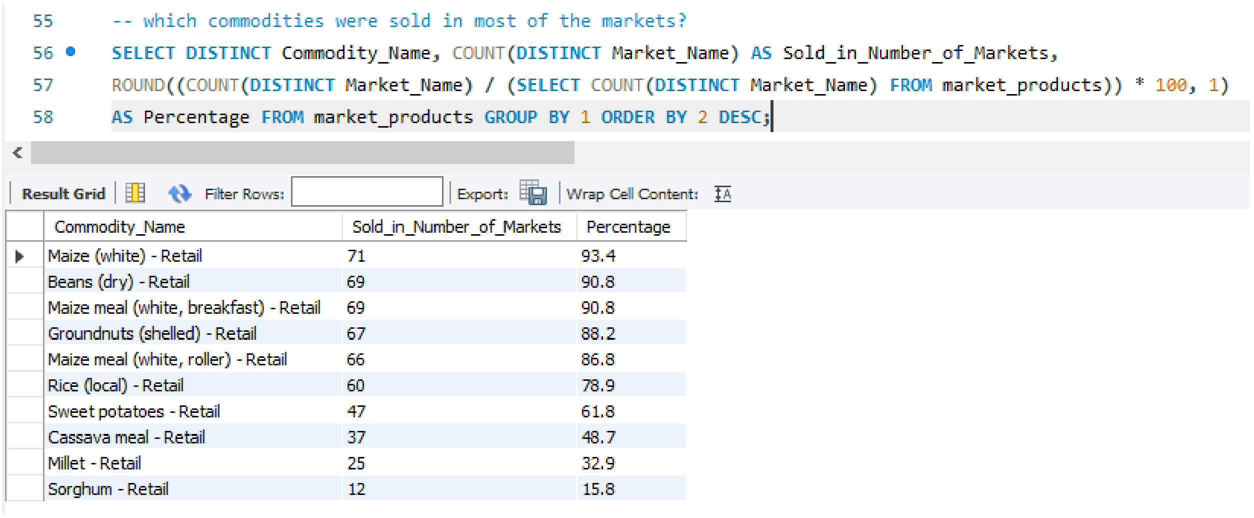


```sql
-- which commodities were the least expensive over the period under review?
SELECT DISTINCT Commodity_Name, ROUND(MIN(Commodity_Price_KG), 2) AS Price_KG, Currency
FROM market_products GROUP BY 1 ORDER BY 2 DESC;
```

Commodity_Name	Price_KG	Currency
Rice (local) - Retail	2	ZMW
Beans (dry) - Retail	1.9	ZMW
Maize meal (white, breakfast) - Retail	1.25	ZMW
Sorghum - Retail	1	ZMW
Cassava meal - Retail	0.8	ZMW
Millet - Retail	0.7	ZMW
Maize meal (white, roller) - Retail	0.6	ZMW
Sweet potatoes - Retail	0.38	ZMW
Maize (white) - Retail	0.14	ZMW
Groundnuts (shelled) - Retail	0.01	ZMW

Based on the data above, **Groundnuts (shelled) - Retail** was the least expensive commodity in the period under review retailing at **K0.01** per kilogram.

Step 3.4: Which commodities were sold in most of the markets?

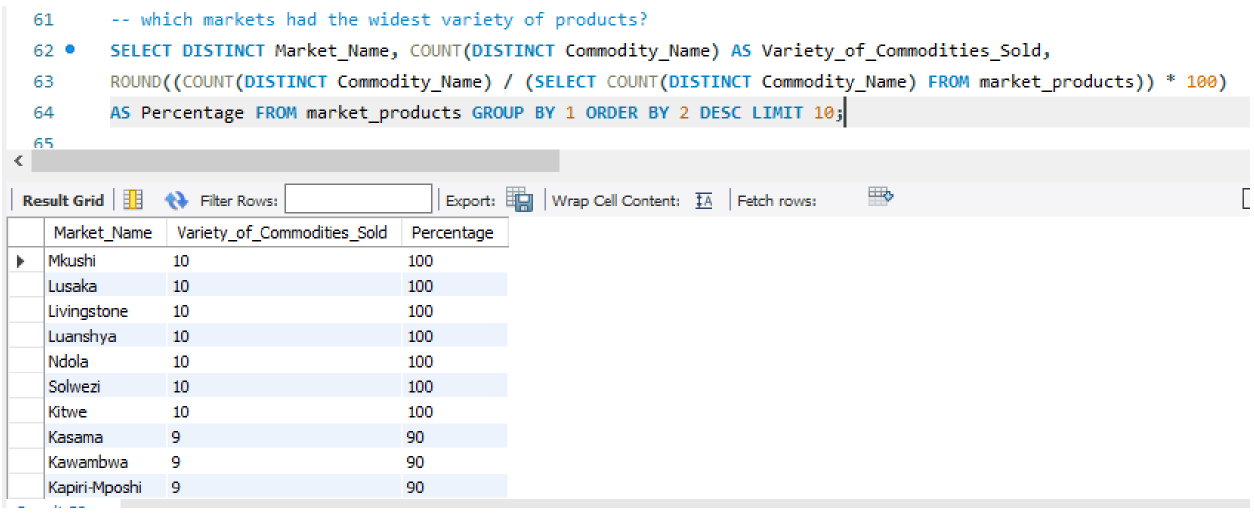


```sql
-- which commodities were sold in most of the markets?
SELECT DISTINCT Commodity_Name, COUNT(DISTINCT Market_Name) AS Sold_in_Number_of_Markets,
ROUND((COUNT(DISTINCT Market_Name) / (SELECT COUNT(DISTINCT Market_Name) FROM market_products)) * 100, 1)
AS Percentage FROM market_products GROUP BY 1 ORDER BY 2 DESC;
```

Commodity_Name	Sold_in_Number_of_Markets	Percentage
Maize (white) - Retail	71	93.4
Beans (dry) - Retail	69	90.8
Maize meal (white, breakfast) - Retail	69	90.8
Groundnuts (shelled) - Retail	67	88.2
Maize meal (white, roller) - Retail	66	86.8
Rice (local) - Retail	60	78.9
Sweet potatoes - Retail	47	61.8
Cassava meal - Retail	37	48.7
Millet - Retail	25	32.9
Sorghum - Retail	12	15.8

Based on the data in the screenshot, **Maize (white) – Retail** was sold in **71** markets, a representation of **93.4%.** The commodity which was sold in the least number of markets was **Sorghum**. It was only sold in **12** markets which was **15.8%** of all the markets across the country.

Step 3.5: Which markets had the widest variety of products?

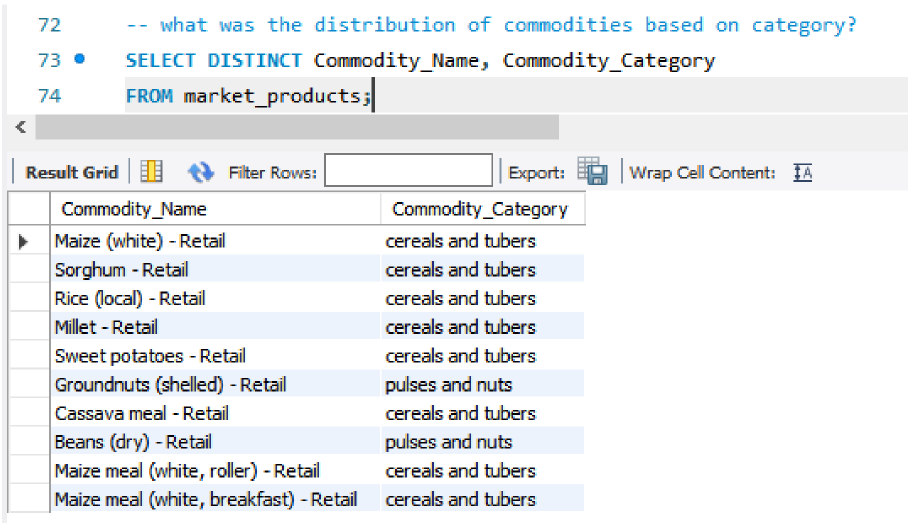


```sql
-- which markets had the widest variety of products?
SELECT DISTINCT Market_Name, COUNT(DISTINCT Commodity_Name) AS Variety_of_Commodities_Sold,
ROUND((COUNT(DISTINCT Commodity_Name) / (SELECT COUNT(DISTINCT Commodity_Name) FROM market_products)) * 100)
AS Percentage FROM market_products GROUP BY 1 ORDER BY 2 DESC LIMIT 10;
```

Market_Name	Variety_of_Commodities_Sold	Percentage
Mkushi	10	100
Lusaka	10	100
Livingstone	10	100
Luanshya	10	100
Ndola	10	100
Solwezi	10	100
Kitwe	10	100
Kasama	9	90
Kawambwa	9	90
Kapiri-Mposhi	9	90

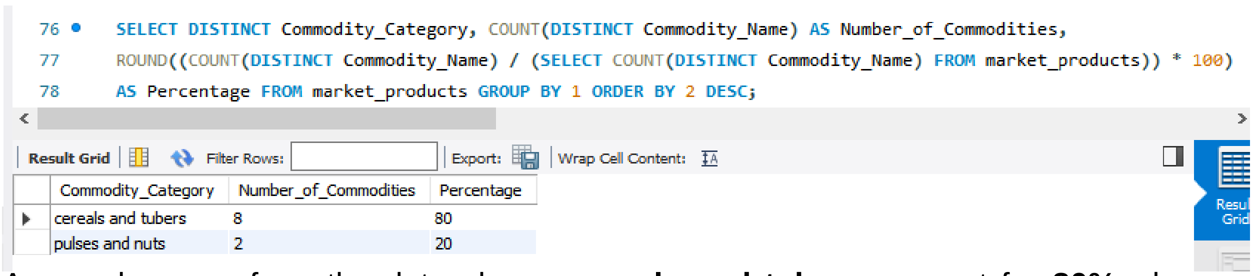
The screenshot only shows the top **ten** (**10)** markets. The first **eight** (**8**) markets stock all the **ten** (**10**) commodities whereas as the bottom **three** (**3**), stock nine (**9**) of the commodities.

Step 3.6: What was the distribution of commodities based on category?



```sql
-- what was the distribution of commodities based on category?
SELECT DISTINCT Commodity_Name, Commodity_Category
FROM market_products;
```

Commodity_Name	Commodity_Category
Maize (white) - Retail	cereals and tubers
Sorghum - Retail	cereals and tubers
Rice (local) - Retail	cereals and tubers
Millet - Retail	cereals and tubers
Sweet potatoes - Retail	cereals and tubers
Groundnuts (shelled) - Retail	pulses and nuts
Cassava meal - Retail	cereals and tubers
Beans (dry) - Retail	pulses and nuts
Maize meal (white, roller) - Retail	cereals and tubers
Maize meal (white, breakfast) - Retail	cereals and tubers

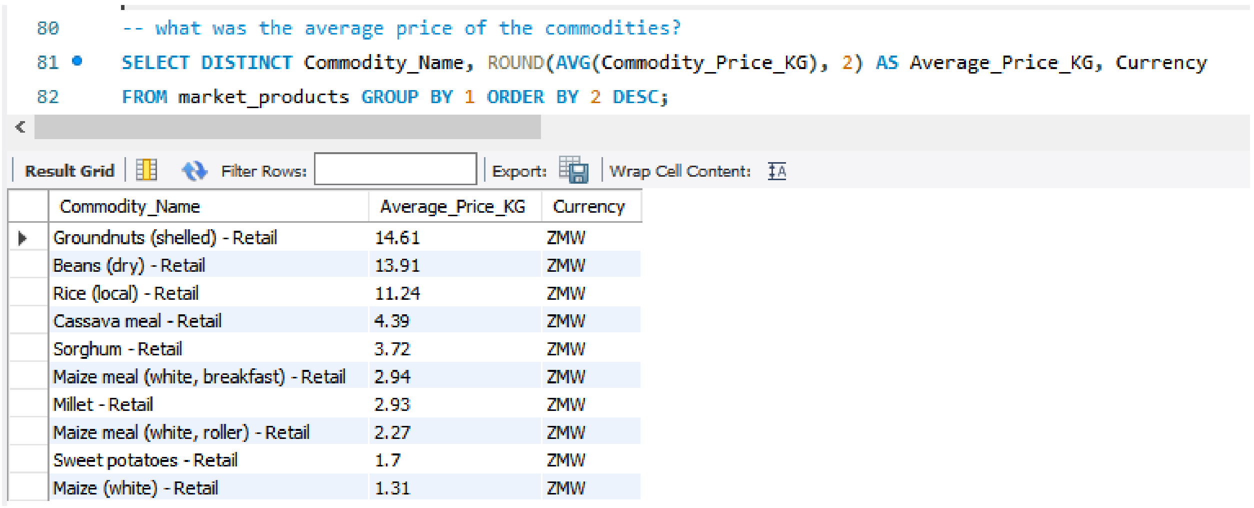


```sql
SELECT DISTINCT Commodity_Category, COUNT(DISTINCT Commodity_Name) AS Number_of_Commodities,
ROUND((COUNT(DISTINCT Commodity_Name) / (SELECT COUNT(DISTINCT Commodity_Name) FROM market_products)) * 100)
AS Percentage FROM market_products GROUP BY 1 ORDER BY 2 DESC;
```

Commodity_Category	Number_of_Commodities	Percentage
cereals and tubers	8	80
pulses and nuts	2	20

As can be seen from the data above, **cereals and tubers** account for **80%** whereas **pulses and nuts** represent **20%** of the commodities.

Step 3.7: What was the average price of the commodities over the review period?



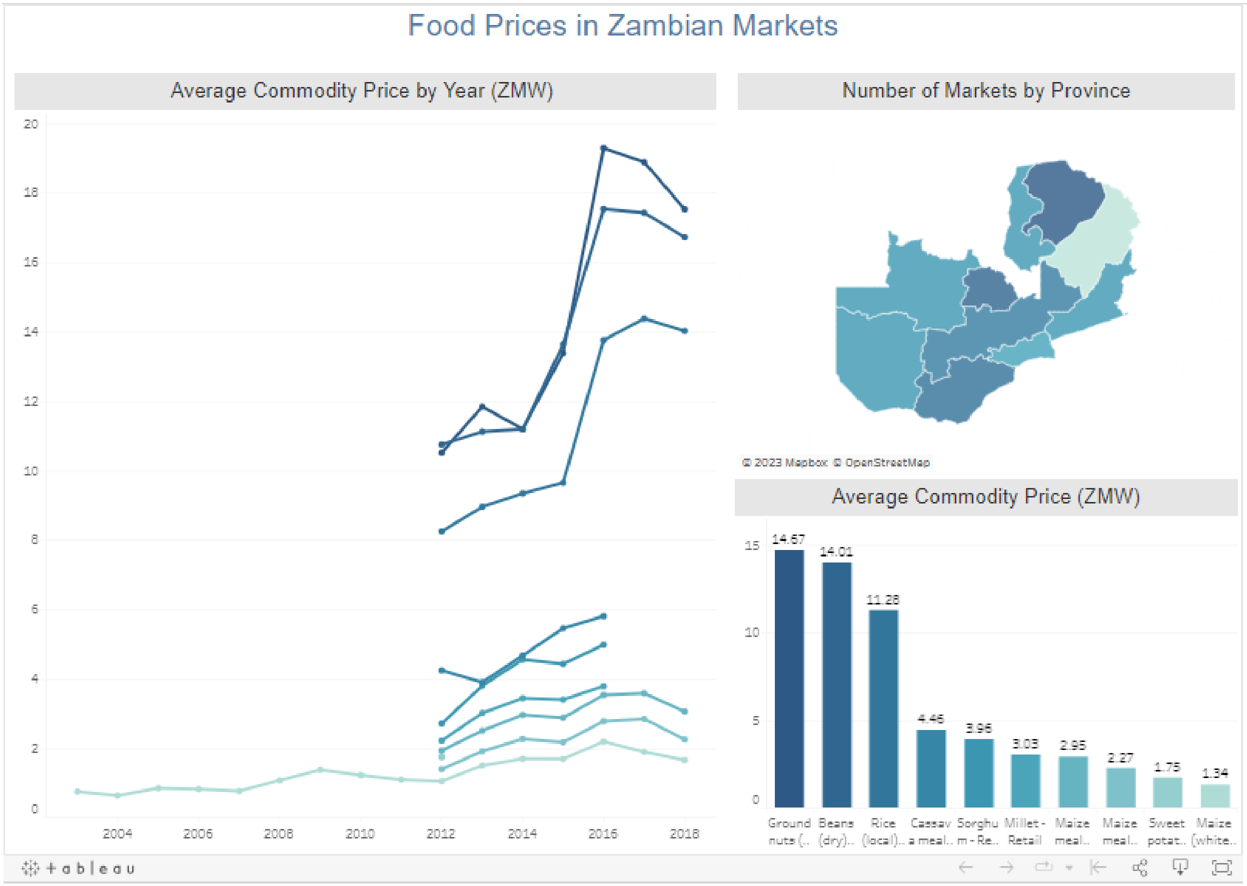
```
-- what was the average price of the commodities?
SELECT DISTINCT Commodity_Name, ROUND(AVG(Commodity_Price_KG), 2) AS Average_Price_KG, Currency
FROM market_products GROUP BY 1 ORDER BY 2 DESC;
```

Commodity_Name	Average_Price_KG	Currency
Groundnuts (shelled) - Retail	14.61	ZMW
Beans (dry) - Retail	13.91	ZMW
Rice (local) - Retail	11.24	ZMW
Cassava meal - Retail	4.39	ZMW
Sorghum - Retail	3.72	ZMW
Maize meal (white, breakfast) - Retail	2.94	ZMW
Millet - Retail	2.93	ZMW
Maize meal (white, roller) - Retail	2.27	ZMW
Sweet potatoes - Retail	1.7	ZMW
Maize (white) - Retail	1.31	ZMW

Groundnuts (shelled) – Retail had the highest average price of **K14.61**. On the other hand, **Maize (white) – Retail** had the least average price pegged at **K1.31** per kilogram.

Part 4: Visualization

The visualization was created using Tableau


Food Prices in Zambian Markets
Average Commodity Price by Year (ZMW)
Number of Markets by Province
© 2023 Mapbox © OpenStreetMap
Average Commodity Price (ZMW)
Ground nuts (..
Beans (dry)..
Rice (local)..
Cassav a meal..
Sorghu m - Re..
Millet - Retail
Maize meal..
Maize meal..
Sweet potat..
Maize (white..
+ a b l e a u