

Sales Insights Data Analysis Project

1. Data Analysis Using SQL

- Show all customer records.

```
4 • SELECT * FROM sales.customers;
```

Result Grid			Filter Rows:	Edit:
customer_code	customer_name	customer_type		
Cus001	Surge Stores	Brick & Mortar		
Cus002	Nomad Stores	Brick & Mortar		
Cus003	Excel Stores	Brick & Mortar		
Cus004	Surface Stores	Brick & Mortar		
Cus005	Premium Stores	Brick & Mortar		
Cus006	Electricalsara Stores	Brick & Mortar		
Cus007	Info Stores	Brick & Mortar		
Cus008	Acclaimed Stores	Brick & Mortar		
Cus009	Electricalsquipo Stores	Brick & Mortar		
Cus010	Atlas Stores	Brick & Mortar		
Cus011	Flawless Stores	Brick & Mortar		
Cus012	Integration Stores	Brick & Mortar		
Cus013	Unity Stores	Brick & Mortar		
Cus014	Forward Stores	Brick & Mortar		
Cus015	Electricalsbea Stores	Brick & Mortar		
Cus016	Logic Stores	Brick & Mortar		
Cus017	Epic Stores	Brick & Mortar		
Cus018	Electricalslance Stores	Brick & Mortar		
Cus019	Electricalsopedia Stores	Brick & Mortar		
Cus020	Nixon	E-Commerce		
Cus021	Modular	E-Commerce		
Cus022	Electricalslytical	E-Commerce		
Cus023	Sound	E-Commerce		
Cus024	Power	E-Commerce		
Cus025	Path	E-Commerce		
Cus026	Insight	E-Commerce		
Cus027	Control	E-Commerce		
Cus028	Sage	E-Commerce		

- Show all date records.

```
4 • SELECT * FROM sales.date;
```

Result Grid					
		Filter Rows:		Edit:	
	date	cy_date	year	month_name	date_yy_mmm
▶	2017-06-01	2017-06-01	2017	June	17-Jun
	2017-06-02	2017-06-01	2017	June	17-Jun
	2017-06-03	2017-06-01	2017	June	17-Jun
	2017-06-04	2017-06-01	2017	June	17-Jun
	2017-06-05	2017-06-01	2017	June	17-Jun
	2017-06-06	2017-06-01	2017	June	17-Jun
	2017-06-07	2017-06-01	2017	June	17-Jun
	2017-06-08	2017-06-01	2017	June	17-Jun
	2017-06-09	2017-06-01	2017	June	17-Jun
	2017-06-10	2017-06-01	2017	June	17-Jun
	2017-06-11	2017-06-01	2017	June	17-Jun
	2017-06-12	2017-06-01	2017	June	17-Jun
	2017-06-13	2017-06-01	2017	June	17-Jun
	2017-06-14	2017-06-01	2017	June	17-Jun
	2017-06-15	2017-06-01	2017	June	17-Jun
	2017-06-16	2017-06-01	2017	June	17-Jun
	2017-06-17	2017-06-01	2017	June	17-Jun
	2017-06-18	2017-06-01	2017	June	17-Jun
	2017-06-19	2017-06-01	2017	June	17-Jun
	2017-06-20	2017-06-01	2017	June	17-Jun
	2017-06-21	2017-06-01	2017	June	17-Jun
	2017-06-22	2017-06-01	2017	June	17-Jun
	2017-06-23	2017-06-01	2017	June	17-Jun
	2017-06-24	2017-06-01	2017	June	17-Jun
	2017-06-25	2017-06-01	2017	June	17-Jun
	2017-06-26	2017-06-01	2017	June	17-Jun
	2017-06-27	2017-06-01	2017	June	17-Jun

- Show all market records.



```
4 • SELECT * FROM sales.markets;
```

Result Grid | Filter Rows:

	markets_code	markets_name	zone
▶	Mark001	Chennai	South
	Mark002	Mumbai	Central
	Mark003	Ahmedabad	North
	Mark004	Delhi NCR	North
	Mark005	Kanpur	North
	Mark006	Bengaluru	South
	Mark007	Bhopal	Central
	Mark008	Lucknow	North
	Mark009	Patna	North
	Mark010	Kochi	South
	Mark011	Nagpur	Central
	Mark012	Surat	North
	Mark013	Bhopal	Central
	Mark014	Hyderabad	South
	Mark015	Bhubaneshwar	South
	Mark097	New York	
	Mark999	Paris	
✱	NULL	NULL	NULL

- Show all products records.

```
4 • SELECT * FROM sales.products;
```

Result Grid   Filter Rows:

product_code	product_type
Prod251	Distribution
Prod252	Distribution
Prod253	Own Brand
Prod254	Own Brand
Prod255	Own Brand
Prod256	Own Brand
Prod257	Own Brand
Prod258	Own Brand
Prod259	Own Brand
Prod260	Own Brand
Prod261	Own Brand
Prod262	Own Brand
Prod263	Distribution
Prod264	Distribution
Prod265	Distribution
Prod266	Distribution
Prod267	Distribution
Prod268	Own Brand
Prod269	Own Brand
Prod270	Own Brand
Prod271	Own Brand
Prod272	Own Brand
Prod273	Own Brand
Prod274	Own Brand
Prod275	Own Brand
Prod276	Own Brand
Prod277	Own Brand
Prod278	Distribution
Prod279	Distribution
NULL	NULL

- Show all transactions records.

21 • `SELECT * FROM sales.transactions;`

Result Grid							
		Filter Rows:		Export:	Wrap Cell Content:		Fetch rows:
product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency	
Prod001	Cus001	Mark001	2017-10-10	100	41241	INR	
Prod001	Cus002	Mark002	2018-05-08	3	-1	INR	
Prod002	Cus003	Mark003	2018-04-06	1	875	INR	
Prod002	Cus003	Mark003	2018-04-11	1	583	INR	
Prod002	Cus004	Mark003	2018-06-18	6	7176	INR	
Prod003	Cus005	Mark004	2017-11-20	59	500	USD	
Prod003	Cus005	Mark004	2017-11-22	36	250	USD	
Prod003	Cus005	Mark004	2017-11-23	39	21412	INR	
Prod003	Cus005	Mark004	2017-11-27	35	19213	INR	
Prod003	Cus005	Mark004	2017-11-28	310	170185	INR	
Prod003	Cus005	Mark004	2017-11-29	184	101194	INR	
Prod003	Cus005	Mark004	2017-11-30	35	19213	INR	

- Show total number of transactions.

```

3      -- total number of transactions
4 •    select count(*) as total_transactions
5      from sales.transactions;

```

total_transactions
150283

- Showing the result of transactions done in Mumbai

```

8      -- Mumbai
9 •    select * from sales.transactions
10     where market_code = 'Mark002';

```

product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
Prod001	Cus002	Mark002	2018-05-08	3	-1	INR
Prod009	Cus009	Mark002	2018-11-14	3	3032	INR
Prod011	Cus016	Mark002	2017-10-06	1	505	INR
Prod011	Cus016	Mark002	2017-10-13	1	1514	INR
Prod011	Cus016	Mark002	2017-10-20	1	2023	INR
Prod011	Cus016	Mark002	2017-10-27	25	37889	INR
Prod011	Cus016	Mark002	2018-02-08	1	593	INR
Prod011	Cus016	Mark002	2018-03-01	1	593	INR
Prod011	Cus016	Mark002	2018-10-12	1	514	INR
Prod011	Cus016	Mark002	2018-10-31	3	4106	INR
Prod011	Cus016	Mark002	2018-12-07	1	514	INR
Prod011	Cus016	Mark002	2018-12-14	1	514	INR
Prod011	Cus016	Mark002	2018-12-21	1	514	INR
Prod011	Cus018	Mark002	2018-12-28	1	0	INR
Prod011	Cus016	Mark002	2019-01-28	1	514	INR
Prod011	Cus016	Mark002	2019-02-01	1	514	INR
Prod011	Cus016	Mark002	2019-02-08	1	1028	INR
Prod011	Cus016	Mark002	2019-02-15	1	2056	INR
Prod011	Cus016	Mark002	2019-02-22	1	1542	INR

■ Total number of transactions in Mumbai city

```
13 • select count(*) as transaction_in_mumbai from sales.transactions
14 where market_code = 'Mark002';
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
transaction_in_mumbai				
11396				

While going through the data I saw some transactions are made outside of the India i.e. New York and Paris let's find out that:

■ Showing result of New York

```
17 • select * from sales.transactions
18 where market_code = 'Mark097';
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
--------------	---------------	-------------	------------	-----------	--------------	----------

■ Showing result of New York

```
17 • select * from sales.transactions
18 where market_code = 'Mark999';
```

Result Grid

Filter Rows:

Export:




Wrap Cell Content:

product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
--------------	---------------	-------------	------------	-----------	--------------	----------

As we can see there is no record for transactions made to these cities.

- Let's check how many transactions are made in USD.

```
21 • select * from sales.transactions
22   where currency = 'USD';
```



Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 							
	product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
▶	Prod003	Cus005	Mark004	2017-11-20	59	500	USD
	Prod003	Cus005	Mark004	2017-11-22	36	250	USD

As we can see there are two transactions which were done in USD. We need to change these transactions to INR while creating the dashboard in PowerBI.

Let's check the transaction table.

- Checking years of transactions we got.




```
21 • select distinct year(order_date) as years
22   from sales.transactions;
```

Result Grid  Filter Rows: <input type="text"/> Export: 	
	years
▶	2017
	2018
	2019
	2020

As we can see we have transactions details from 2017 to 2020. Let's check into year 2020.



■ Transaction details of year 2020

```
25 • select * from sales.transactions
26   where year(order_date) = 2020;
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content:  Fetch rows:							
	product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
▶	Prod005	Cus007	Mark004	2020-01-09	1	630	INR
	Prod005	Cus007	Mark004	2020-01-10	1	472	INR
	Prod005	Cus007	Mark004	2020-01-17	2	2042	INR
	Prod005	Cus007	Mark004	2020-02-07	1	417	INR
	Prod005	Cus007	Mark004	2020-02-14	1	310	INR
	Prod005	Cus007	Mark004	2020-02-28	1	208	INR
	Prod005	Cus007	Mark004	2020-03-06	1	620	INR
	Prod005	Cus007	Mark004	2020-03-13	1	620	INR
	Prod005	Cus007	Mark004	2020-04-03	1	829	INR
	Prod005	Cus007	Mark004	2020-04-14	4	2694	INR
	Prod005	Cus007	Mark004	2020-04-20	1	102	INR
	Prod005	Cus007	Mark004	2020-05-15	2	528	INR
	Prod011	Cus016	Mark002	2020-06-12	1	1028	INR
	Prod011	Cus016	Mark002	2020-06-16	1	514	INR
	Prod005	Cus007	Mark004	2020-01-09	1	630	INR
	Prod005	Cus007	Mark004	2020-01-10	1	472	INR
	Prod005	Cus007	Mark004	2020-01-17	2	2042	INR
	Prod005	Cus007	Mark004	2020-02-07	1	417	INR

■ Let's count how many numbers of transactions are there.

```
25 • select count(*) from sales.transactions
26   where year(order_date) = 2020;
```

Result Grid  Filter Rows: <input type="text"/> Export: 	
	count(*)
▶	21550

One another way we can extract the details by using Inner Join where we join table date and transaction and extract the information from both the tables from year 2020.

■ Using Inner Join

```
28 • select sales.transactions.*, sales.date.*
29 from sales.transactions
30 inner join sales.date on sales.transactions.order_date = sales.date.date
31 where year = 2020;
```

product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency	date	cy_date	year	month_name	date_yy_mmm
Prod005	Cus007	Mark004	2020-01-09	1	630	INR	2020-01-09	2020-01-01	2020	January	20-Jan
Prod005	Cus007	Mark004	2020-01-10	1	472	INR	2020-01-10	2020-01-01	2020	January	20-Jan
Prod005	Cus007	Mark004	2020-01-17	2	2042	INR	2020-01-17	2020-01-01	2020	January	20-Jan
Prod005	Cus007	Mark004	2020-02-07	1	417	INR	2020-02-07	2020-02-01	2020	February	20-Feb
Prod005	Cus007	Mark004	2020-02-14	1	310	INR	2020-02-14	2020-02-01	2020	February	20-Feb
Prod005	Cus007	Mark004	2020-02-28	1	208	INR	2020-02-28	2020-02-01	2020	February	20-Feb
Prod005	Cus007	Mark004	2020-03-06	1	620	INR	2020-03-06	2020-03-01	2020	March	20-Mar
Prod005	Cus007	Mark004	2020-03-13	1	620	INR	2020-03-13	2020-03-01	2020	March	20-Mar
Prod005	Cus007	Mark004	2020-04-03	1	829	INR	2020-04-03	2020-04-01	2020	April	20-Apr
Prod005	Cus007	Mark004	2020-04-14	4	2694	INR	2020-04-14	2020-04-01	2020	April	20-Apr
Prod005	Cus007	Mark004	2020-04-20	1	102	INR	2020-04-20	2020-04-01	2020	April	20-Apr
Prod005	Cus007	Mark004	2020-05-15	2	528	INR	2020-05-15	2020-05-01	2020	May	20-May
Prod011	Cus016	Mark002	2020-06-12	1	1028	INR	2020-06-12	2020-06-01	2020	June	20-Jun
Prod011	Cus016	Mark002	2020-06-16	1	514	INR	2020-06-16	2020-06-01	2020	June	20-Jun
Prod005	Cus007	Mark004	2020-01-09	1	630	INR	2020-01-09	2020-01-01	2020	January	20-Jan
Prod005	Cus007	Mark004	2020-01-10	1	472	INR	2020-01-10	2020-01-01	2020	January	20-Jan

Here we can see we have the result from both the tables now we can precisely see the transactions made in 2020.

■ Total revenue in year 2020

```
28 • select SUM(sales.transactions.sales_amount) as total_revenue_2020
29 from sales.transactions
30 inner join sales.date on sales.transactions.order_date = sales.date.date
31 where year = 2020;
```

total_revenue_2020
142235559

Likewise let's get the revenue of other years too just to find the difference in revenue.

■ Total revenue in year 2019

```
28 • select SUM(sales.transactions.sales_amount) as total_revenue_2019
29   from sales.transactions
30   inner join sales.date on sales.transactions.order_date = sales.date.date
31   where year = 2019;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_revenue_2019			
▶	336452114			

■ Total revenue in year 2018

```
28 • select SUM(sales.transactions.sales_amount) as total_revenue_2018
29   from sales.transactions
30   inner join sales.date on sales.transactions.order_date = sales.date.date
31   where year = 2018;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_revenue_2018			
▶	414308941			

■ Total revenue in year 2017

```
28 • select SUM(sales.transactions.sales_amount) as total_revenue_2017
29   from sales.transactions
30   inner join sales.date on sales.transactions.order_date = sales.date.date
31   where year = 2017;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_revenue_2017			
▶	93569152			

By looking at the output the revenue is decreasing every year. It was highest in 2017 and lowest in 2020.

Let's find out the total revenue in some of the cities in 2017 (highest) and 2020 (lowest).

■ Total revenue in Mumbai in 2017.

```
28 • select SUM(sales.transactions.sales_amount) as total_revenue_mumbai_2017
29 from sales.transactions
30 inner join sales.date on sales.transactions.order_date = sales.date.date
31 where year = 2017 and sales.transactions.market_code = "Mark002";
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_revenue_mumbai_2017			
▶	14916377			

■ Total revenue in Mumbai in 2020

```
28 • select SUM(sales.transactions.sales_amount) as total_revenue_mumbai_2020
29 from sales.transactions
30 inner join sales.date on sales.transactions.order_date = sales.date.date
31 where year = 2020 and sales.transactions.market_code = "Mark002";
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_revenue_mumbai_2020			
▶	20183077			

As we can see there is huge decline in the revenue between 2017 and 2020. Let's do for one more city.

■ Total revenue in Chennai 2017

```
28 • select SUM(sales.transactions.sales_amount) as total_revenue_chennai_2017
29   from sales.transactions
30   inner join sales.date on sales.transactions.order_date = sales.date.date
31   where year = 2017 and sales.transactions.market_code = "Mark001";
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_revenue_chennai_2017			
▶	1670830			

■ Total revenue in Chennai 2020

```
28 • select SUM(sales.transactions.sales_amount) as total_revenue_chennai_2020
29   from sales.transactions
30   inner join sales.date on sales.transactions.order_date = sales.date.date
31   where year = 2020 and sales.transactions.market_code = "Mark001";
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_revenue_chennai_2020			
▶	2463024			

Wow, surprisingly the revenue of Chennai went up comparing between 2017 and 2020 as overall revenue of the year was less than 2017. It means they did a good sale in Chennai city.

While going through the dataset I came across some negative values in transaction table under 'sales_amount' column let's find out that.

■ -ve sales amount

```
18 • Select * from sales.transactions
19 where sales_amount < 0;
```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:							
	product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
▶	Prod001	Cus002	Mark002	2018-05-08	3	-1	INR
	Prod001	Cus002	Mark002	2018-05-08	3	-1	INR

As we can see we have multiple negative number. We will try to remove these while performing ETL in powerBI.

Let's try to find 0 (zero) sales amount too because we need to remove zero value too. It doesn't make any sense if you sell something to someone it can be zero. Let's check it out.

■ Zero sales amount

```
5 • Select * from sales.transactions
6 where sales_amount <= 0;
7
```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content: Fetch rows:							
	product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
	Prod001	Cus002	Mark002	2018-05-08	3	-1	INR
	Prod010	Cus015	Mark006	2018-05-26	1	0	INR
	Prod010	Cus003	Mark003	2019-04-30	1	0	INR
	Prod011	Cus018	Mark002	2018-12-28	1	0	INR
	Prod001	Cus002	Mark002	2018-05-08	3	-1	INR
	Prod010	Cus015	Mark006	2018-05-26	1	0	INR
	Prod010	Cus003	Mark003	2019-04-30	1	0	INR
	Prod011	Cus018	Mark002	2018-12-28	1	0	INR
	Prod016	Cus005	Mark006	2018-01-10	5	0	INR
	Prod021	Cus003	Mark006	2017-10-10	1	0	INR
	Prod021	Cus027	Mark006	2017-10-16	1	0	INR
	Prod021	Cus003	Mark006	2017-10-18	1	0	INR
	Prod021	Cus003	Mark006	2017-10-19	1	0	INR
	Prod039	Cus003	Mark006	2017-10-10	1	0	INR
	Prod039	Cus013	Mark006	2017-10-10	1	0	INR
	Prod039	Cus003	Mark006	2018-01-15	1	0	INR
	Prod039	Cus011	Mark006	2018-03-08	5	0	INR
	Prod040	Cus005	Mark006	2018-01-19	40	0	INR
	Prod040	Cus020	Mark006	2018-01-31	3	0	INR
	Prod040	Cus020	Mark006	2018-02-06	1	0	INR
	Prod040	Cus005	Mark006	2018-05-25	2	0	INR
	Prod040	Cus006	Mark004	2018-07-02	53	0	INR
	Prod040	Cus005	Mark006	2018-07-03	3	0	INR
	Prod040	Cus005	Mark006	2018-07-18	12	0	INR
	Prod040	Cus034	Mark010	2018-10-15	1	0	INR
	Prod040	Cus006	Mark004	2019-01-29	1	0	INR

```

5 • Select count(*) as total_zero_sales
6   from sales.transactions
7   where sales_amount <= 0;

```

Result Grid		Filter Rows:	Export:
	total_zero_sales		
	1611		

As we can see we have 1611 zero sales including two negative values. This is a great insight which we can use while doing out ETL and we will try to eliminate these before creating PowerBI dashboard.

Furthermore, when I checked the currency column from sales.transaction I realized that the occurring of currency is twice. What I mean by it is in out dataset there are two types of currency shown i.e. USD and INR but when I use distinct command in SQL, I am getting 2 USD and 2 INR.

```

9 • select distinct currency
10  from sales.transactions;

```

Result Grid		Filter Rows:
	currency	
▶	INR	
	USD	
	INR	
	USD	

Further analysis got me this:

```
9 • select distinct currency
10   from sales.transactions;
11
12 ✖ 'INR'
13   'USD'
14   'INR\r'
15   'USD\r'
```

Result Grid | Filter Rows:

currency
INR
USD
INR
USD

Where currencies are stored in /r format. Let's calculate the number of such currencies in our dataset.

```
17 • select count(*) from sales.transactions
18   where transactions.currency = 'INR\r';
```

Result Grid | Filter Rows: Export:

count(*)
150000

```
17 • select count(*) from sales.transactions
18   where transactions.currency = 'USD\r';
```

Result Grid | Filter Rows: Export:

count(*)
2

As we can see we have a lot of count value under INR\r currency where \r is nothing but a new line character. But here is one interesting thing as we can see USD and USD\r have value count 2. Let's check if they are duplicate or not.


```

17 • select * from sales.transactions
18   where transactions.currency = 'USD\r' or transactions.currency = 'USD';

```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:							
	product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
▶	Prod003	Cus005	Mark004	2017-11-20	59	500	USD
	Prod003	Cus005	Mark004	2017-11-22	36	250	USD
	Prod003	Cus005	Mark004	2017-11-20	59	500	USD
	Prod003	Cus005	Mark004	2017-11-22	36	250	USD

As we can see above data are duplicated so this makes sense that INR and INR\r might have same duplication. Let's go for further analysis.

```

23 • SELECT
24     product_code,
25     customer_code,
26     market_code,
27     order_date,
28     sales_qty,
29     sales_amount,
30     COUNT(*) as row_value_count
31 FROM sales.transactions
32 GROUP BY
33     product_code,
34     customer_code,
35     market_code,
36     order_date,
37     sales_qty,
38     sales_amount
39 HAVING COUNT(*) > 1;

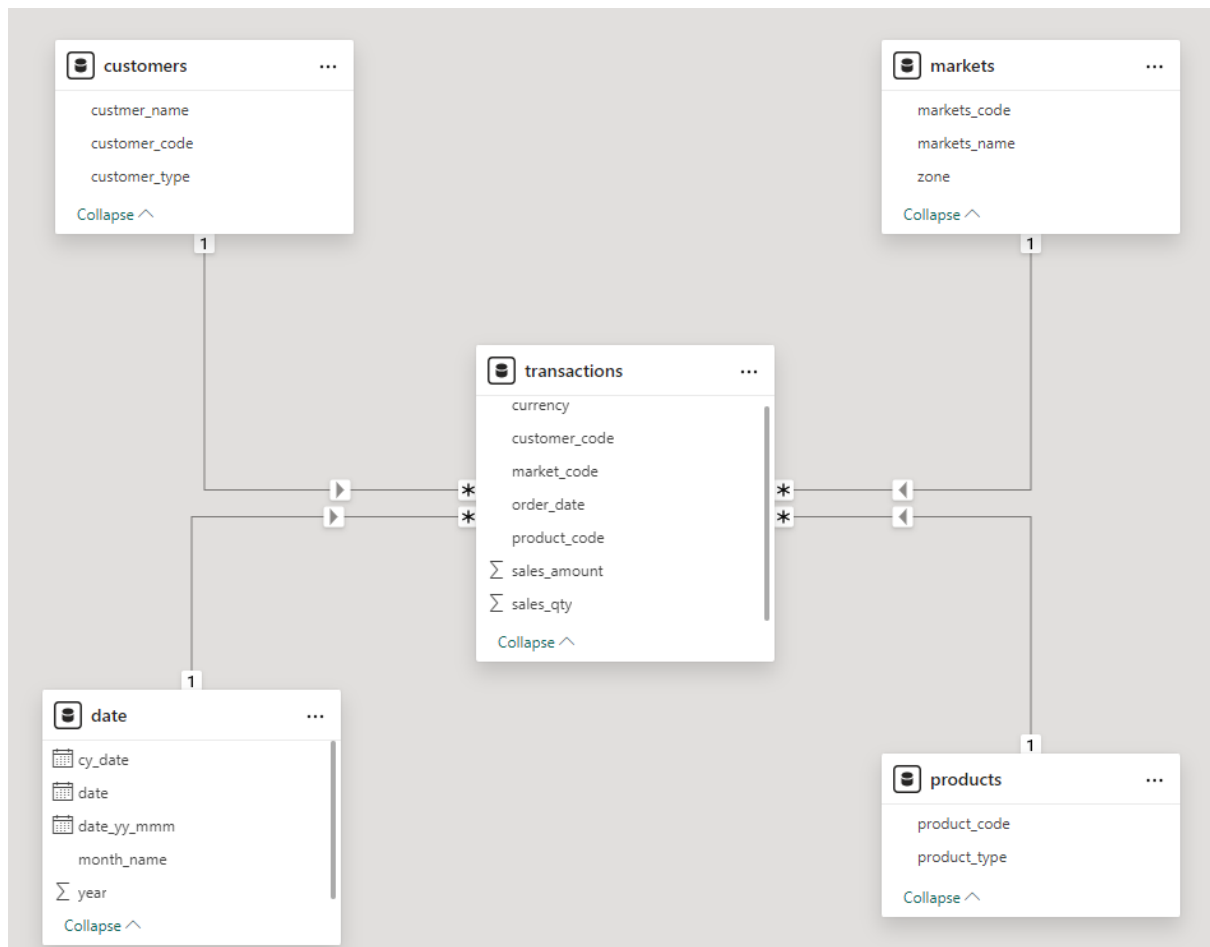
```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:							
	product_code	customer_code	market_code	order_date	sales_qty	sales_amount	row_value_count
	Prod001	Cus001	Mark001	2017-10-10	100	41241	2
	Prod001	Cus002	Mark002	2018-05-08	3	-1	2
	Prod002	Cus003	Mark003	2018-04-06	1	875	2
	Prod002	Cus003	Mark003	2018-04-11	1	583	2
	Prod002	Cus004	Mark003	2018-06-18	6	7176	2
	Prod003	Cus005	Mark004	2017-11-20	59	500	2
	Prod003	Cus005	Mark004	2017-11-22	36	250	2
	Prod003	Cus005	Mark004	2017-11-23	39	21412	2
	Prod003	Cus005	Mark004	2017-11-27	35	19213	2
	Prod003	Cus005	Mark004	2017-11-28	310	170185	2
	Prod003	Cus005	Mark004	2017-11-29	184	101194	2
	Prod003	Cus005	Mark004	2017-11-30	35	19213	2
	Prod004	Cus005	Mark004	2017-11-29	17	9426	2

This query works by grouping the rows based on the specified columns and then counting the number of rows within each group. The **HAVING COUNT (*) > 1** condition filters out groups where the count is greater than 1, meaning those are the groups with duplicate values.

2. ETL using PowerBI

Model view in star schema:



2.1. Transforming data

■ Removing New York and Paris

	markets_code	markets_name	zone
1	Mark001	Chennai	South
2	Mark002	Mumbai	Central
3	Mark003	Ahmedabad	North
4	Mark004	Delhi NCR	North
5	Mark005	Kanpur	North
6	Mark006	Bengaluru	South
7	Mark007	Bhopal	Central
8	Mark008	Lucknow	North
9	Mark009	Patna	North
10	Mark010	Kochi	South
11	Mark011	Nagpur	Central
12	Mark012	Surat	North
13	Mark013	Bhopal	Central
14	Mark014	Hyderabad	South
15	Mark015	Bhubaneshwar	South
16	Mark097	New York	
17	Mark999	Paris	

From above picture we can see there is no zone mentioned for New York and Paris and while analysing using SQL we saw no transaction was made in those cities it means there is no data for these cities. Its better if we remove these two cities.

	markets_code	markets_name	zone
1	Mark001	Chennai	South
2	Mark002	Mumbai	Central
3	Mark003	Ahmedabad	North
4	Mark004	Delhi NCR	North
5	Mark005	Kanpur	North
6	Mark006	Bengaluru	South
7	Mark007	Bhopal	Central
8	Mark008	Lucknow	North
9	Mark009	Patna	North
10	Mark010	Kochi	South
11	Mark011	Nagpur	Central
12	Mark012	Surat	North
13	Mark013	Bhopal	Central
14	Mark014	Hyderabad	South
15	Mark015	Bhubaneshwar	South

SO we used a formula where zone is null remove it.

■ Filtering out negative and zero sales value

✕ ✓ *fx* = Table.SelectRows(#"Changed Type", each ([sales_amount] = -1 or [sales_amount] = 0))

	A ^B _C product_code	A ^B _C customer_code	A ^B _C market_code	order_date	1 ² ₃ sales_qty	1 ² ₃ sales_amount	A ^B _C currency
1	Prod001	Cus002	Mark002	5/8/2018	3	-1	INR
2	Prod010	Cus015	Mark006	5/26/2018	1	0	INR
3	Prod010	Cus003	Mark003	4/30/2019	1	0	INR
4	Prod011	Cus018	Mark002	12/28/2018	1	0	INR
5	Prod001	Cus002	Mark002	5/8/2018	3	-1	INR
6	Prod010	Cus015	Mark006	5/26/2018	1	0	INR
7	Prod010	Cus003	Mark003	4/30/2019	1	0	INR
8	Prod011	Cus018	Mark002	12/28/2018	1	0	INR
9	Prod016	Cus005	Mark006	1/10/2018	5	0	INR
10	Prod021	Cus003	Mark006	10/10/2017	1	0	INR
11	Prod021	Cus027	Mark006	10/16/2017	1	0	INR
12	Prod021	Cus003	Mark006	10/18/2017	1	0	INR
13	Prod021	Cus003	Mark006	10/19/2017	1	0	INR
14	Prod039	Cus003	Mark006	10/10/2017	1	0	INR

We have 1609 zeros and two negative sales value which we got while analysing using SQL.
Let's remove these data because it doesn't make any sense.

✕ ✓ *fx* = Table.SelectRows(#"Changed Type", each ([sales_amount] <> -1 and [sales_amount] <> 0))

	A ^B _C product_code	A ^B _C customer_code	A ^B _C market_code	order_date	1 ² ₃ sales_qty	1 ² ₃ sales_amount	A ^B _C currency
1	Prod001	Cus001	Mark001	10/10/2017	100	41241	INR
2	Prod002	Cus003	Mark003	4/6/2018	1	875	INR
3	Prod002	Cus003	Mark003	4/11/2018	1	583	INR
4	Prod002	Cus004	Mark003	6/18/2018	6	7176	INR
5	Prod003	Cus005	Mark004	11/20/2017	59	500	USD
6	Prod003	Cus005	Mark004	11/22/2017	36	250	USD
7	Prod003	Cus005	Mark004	11/23/2017	39	21412	INR
8	Prod003	Cus005	Mark004	11/27/2017	35	19213	INR
9	Prod003	Cus005	Mark004	11/28/2017	310	170185	INR
10	Prod003	Cus005	Mark004	11/29/2017	184	101194	INR
11	Prod003	Cus005	Mark004	11/30/2017	35	19213	INR
12	Prod004	Cus005	Mark004	11/29/2017	17	9426	INR
13	Prod004	Cus005	Mark004	12/19/2017	1	218	INR
14	Prod005	Cus005	Mark004	8/7/2018	5	3093	INR
15	Prod003	Cus006	Mark004	12/4/2017	58	30306	INR
16	Prod005	Cus006	Mark004	6/29/2018	38	52319	INR
17	Prod005	Cus006	Mark004	7/2/2018	93	126296	INR
18	Prod005	Cus006	Mark004	7/3/2018	79	107500	INR
19	Prod005	Cus006	Mark004	7/4/2018	1	273	INR
20	Prod005	Cus006	Mark004	7/6/2018	3	3574	INR
21	Prod005	Cus006	Mark004	7/13/2018	1	273	INR
22	Prod005	Cus006	Mark004	6/7/2019	20	23403	INR
23	Prod005	Cus006	Mark004	7/29/2019	81	76329	INR
24	Prod005	Cus006	Mark004	8/1/2019	5	4542	INR
25	Prod005	Cus006	Mark004	9/19/2019	18	16579	INR

So, we filtered out the sales amount column and removed the negative and zero value.

■ Types of currencies

As we mentioned in SQL too, we are viewing 4 currency type as we only got two i.e. USD and INR.

123 sales_qty	123 sales_amount	A ^B currency
79	107500	INR
1	273	INR
3	3574	INR
1	273	INR
20	23403	INR
81	76329	INR
5	4542	INR
18	16579	INR
90	105301	INR

Let's analyse for it.

= Table.SelectRows("#filtered out -1 and 0 sales", each ([currency] = "INR#(cr)" or [currency] = "USD#(cr)"))

A ^B product_code	A ^B customer_code	A ^B market_code	order_date	123 sales_qty	123 sales_amount	A ^B currency
1 Prod001	Cus001	Mark001	10/10/2017			
2 Prod002	Cus003	Mark003	4/6/2018			
3 Prod002	Cus003	Mark003	4/11/2018			
4 Prod002	Cus004	Mark003	6/18/2018			
5 Prod003	Cus005	Mark004	11/20/2017			
6 Prod003	Cus005	Mark004	11/22/2017			
7 Prod003	Cus005	Mark004	11/23/2017			
8 Prod003	Cus005	Mark004	11/27/2017			
9 Prod003	Cus005	Mark004	11/28/2017			
10 Prod003	Cus005	Mark004	11/29/2017		184	101194 INR

As underlined with get we can see that they are attached with some garbage words but that's okay we can use it anyway I just need to mentioned whole garbage word in a condition while changing the currency INR to USD.

- Conversion of currency

= Table.AddColumn("#Filtered Rows", "normalize_sales_amount", each if [currency] = "USD" or [currency] = "USDK(₹)" then [sales_amount] else [sales_amount]/80)

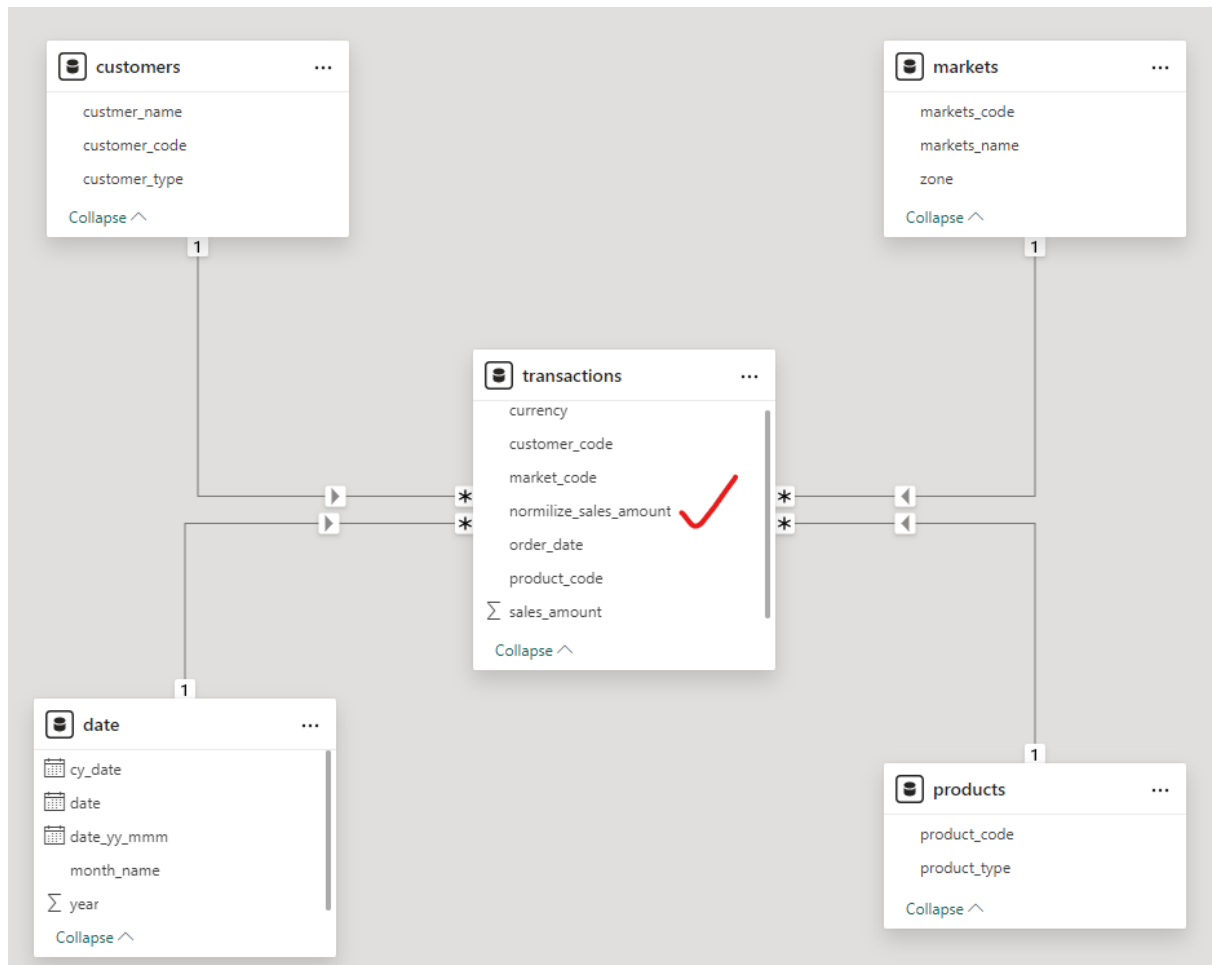
	A ₀ product_code	A ₀ customer_code	A ₀ market_code	order_date	1 ₂ sales_qty	1 ₂ sales_amount	A ₀ currency	A ₀ normalize_sales_amount
1	Prod001	Cus001	Mark001	10/10/2017		100	41241 INR	515.5125
2	Prod002	Cus003	Mark003	4/6/2018	1	875	INR	10.9375
3	Prod002	Cus003	Mark003	4/11/2018	1	583	INR	7.2875
4	Prod002	Cus004	Mark003	6/18/2018	6	7176	INR	89.7
5	Prod003	Cus005	Mark004	11/20/2017	59	500	USD	500
6	Prod003	Cus005	Mark004	11/22/2017	36	250	USD	250
7	Prod003	Cus005	Mark004	11/23/2017	39	21412	INR	267.65
8	Prod003	Cus005	Mark004	11/27/2017	35	19213	INR	240.1625
9	Prod003	Cus005	Mark004	11/28/2017	310	170185	INR	2127.3125
10	Prod003	Cus005	Mark004	11/29/2017	184	101194	INR	1264.925
11	Prod003	Cus005	Mark004	11/30/2017	35	19213	INR	240.1625
12	Prod004	Cus005	Mark004	11/29/2017	17	9426	INR	117.825
13	Prod004	Cus005	Mark004	12/19/2017	1	218	INR	2.725
14	Prod005	Cus005	Mark004	8/7/2018	5	3093	INR	38.6625
15	Prod003	Cus006	Mark004	12/4/2017	58	30306	INR	378.825
16	Prod005	Cus006	Mark004	6/29/2018	38	52319	INR	653.9875
17	Prod005	Cus006	Mark004	7/2/2018	93	126296	INR	1578.7
18	Prod005	Cus006	Mark004	7/3/2018	79	107500	INR	1343.75
19	Prod005	Cus006	Mark004	7/4/2018	1	273	INR	3.4125

I have created a new column (i.e. `normilize_sales_amount`) that replicates the `sales_amount` column but as mentioned before I have converted INR into USD.

2.2. Removing duplicate values

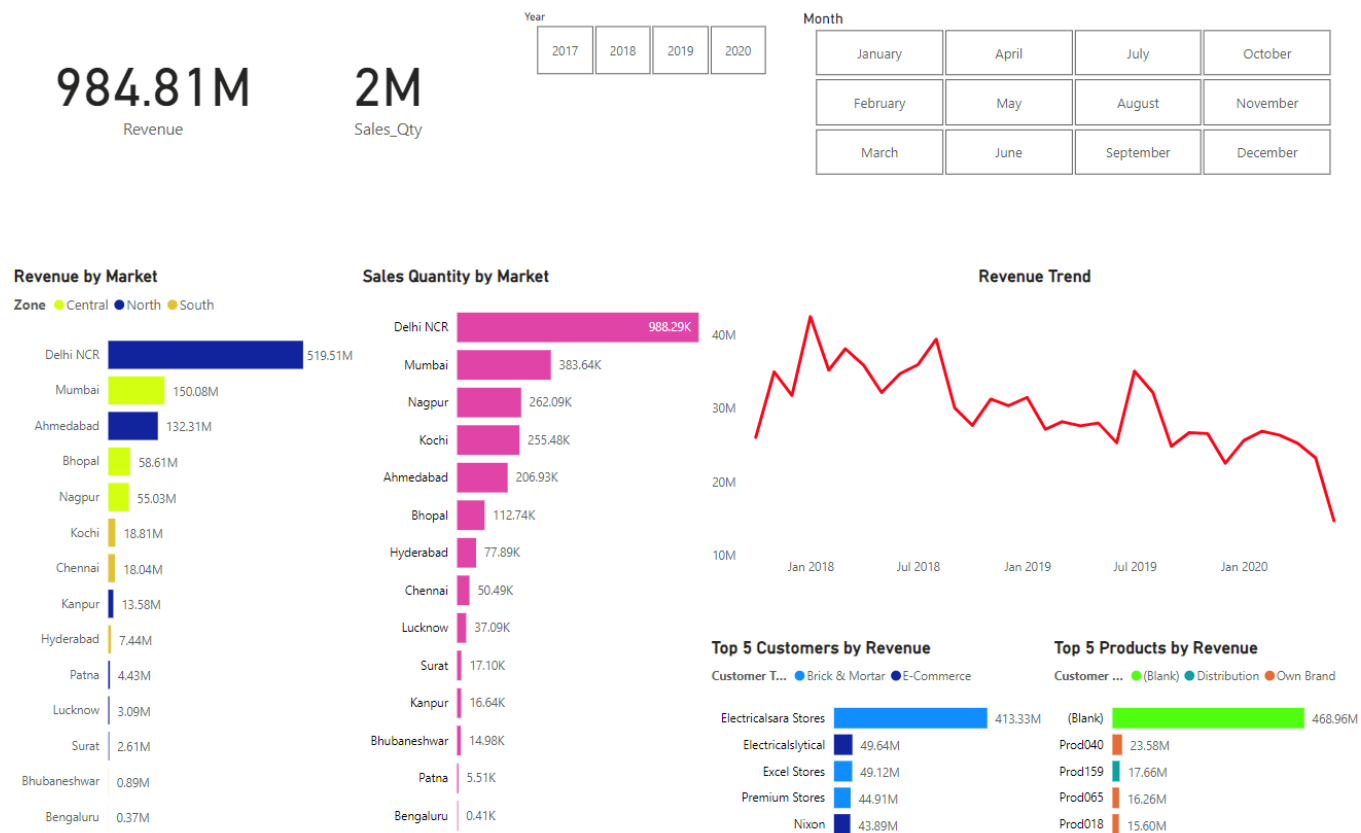
= Table.SelectRows(#"filtered out -1 and 0 sales", each ([currency] = "INR#" or [currency] = "USD#"))							
	product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
1	Prod001	Cus001	Mark001	10/10/2017	100	41241	INR
2	Prod002	Cus003	Mark003	4/6/2018	1	875	INR
3	Prod002	Cus003	Mark003	4/11/2018	1	583	INR
4	Prod002	Cus004	Mark003	6/18/2018	6	7176	INR
5	Prod003	Cus005	Mark004	11/20/2017	59	500	USD
6	Prod003	Cus005	Mark004	11/22/2017	36	250	USD
7	Prod003	Cus005	Mark004	11/23/2017	39	21412	INR
8	Prod003	Cus005	Mark004	11/27/2017	35	19213	INR
9	Prod003	Cus005	Mark004	11/28/2017	310	170185	INR
10	Prod003	Cus005	Mark004	11/29/2017	184	101194	INR
11	Prod003	Cus005	Mark004	11/30/2017	35	19213	INR
12	Prod004	Cus005	Mark004	11/29/2017	17	9426	INR
13	Prod004	Cus005	Mark004	12/19/2017	1	218	INR

As we figure out that we have duplicate values i.e. INR and USD as a currency we managed to remove them in PowerBI, which is one of the crucial parts in data analysis.

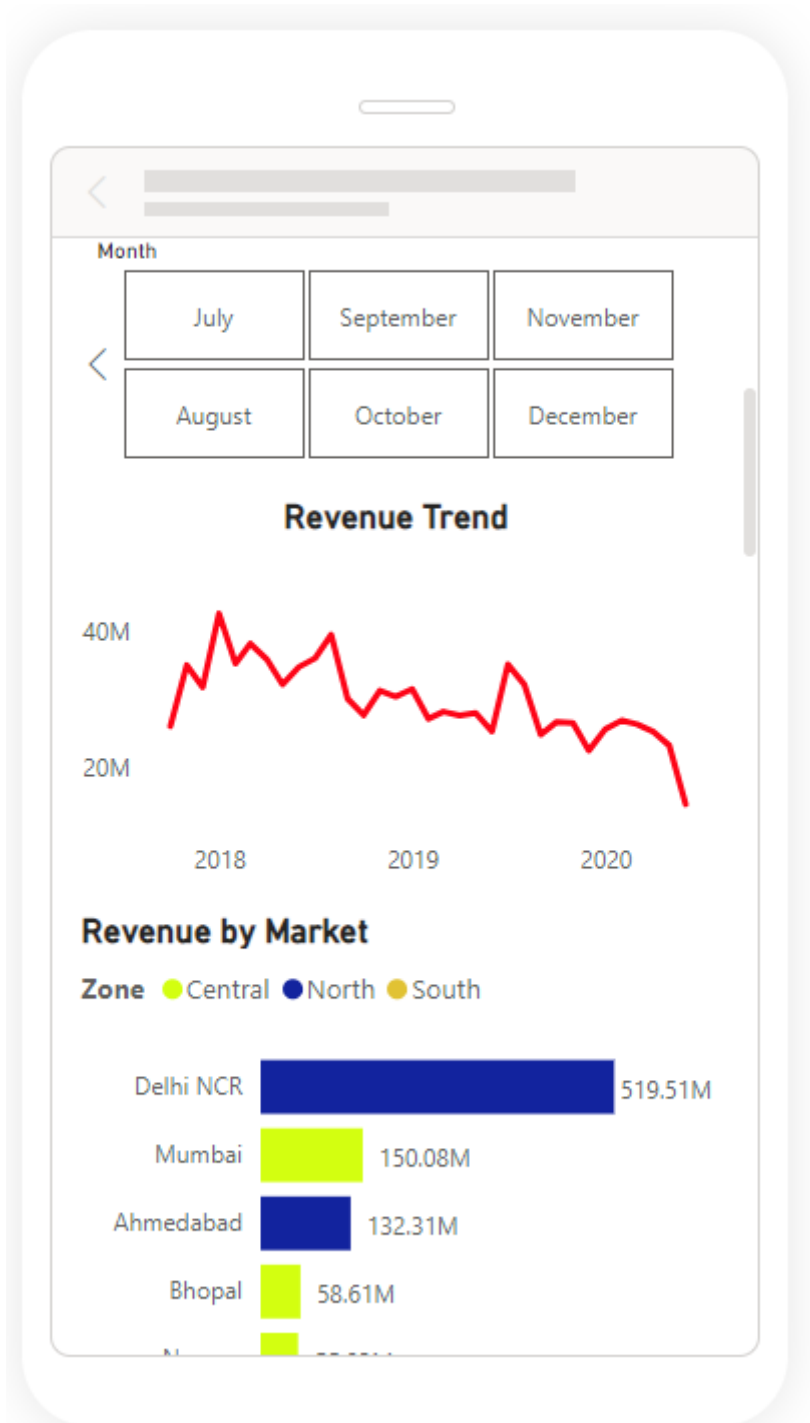


Now we can see we have a transformed data as we can see extra column we have created.
Now we can go and create a dashboard.

3. Dashboard



This interactive sales dashboard provides valuable insights into customer behaviour across different markets. We see the Central zone leading in revenue, with Delhi NCR being the standout performer. Interestingly, the South zone boasts higher sales volume, suggesting potential for market expansion or product diversification there. Top contributors include Electricalsara Stores, Electricalslytical, and Excel Stores, while ‘Brick & Mortar E-Commerce’ emerges as the leading product, followed by ‘Electricals’ and ‘Premium Stores.’ These findings empower us to make informed decisions and target resources effectively.



Same dashboard I have created for mobile view.