SQL & POWER BI PROJECT SHOWCASE

- - Chaitnya Paliwal

PROBLEM STATEMENT

- Sales data is stored in raw transactional tables, making it difficult to extract meaningful insights.
- Business stakeholders lack clear visibility into sales performance, profitability, and customer behavior.
- No centralized system exists to analyze product, market, and customer-level performance.
- Manual reporting is time-consuming and prone to errors, limiting timely decision-making.
- The business needs a scalable analytics solution using MySQL and Power BI to enable data-driven decisions.

This project follows a progressive SQL approach, starting with foundational queries and advancing to complex analytical queries that deliver strategic insights.

List all customers along with their customer code, name, and customer type.

```sql
SELECT
    customer_code, custmer_name, customer_type
FROM
    customers;
```

	customer_code	custmer_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

Find the total number of transactions recorded in the database.

```sql
SELECT
    COUNT(*) AS 'Total Transactions'
FROM
    transactions;
```

	Total Transactions
▶	148395

Display all products along with their product code and product type.

```sql
SELECT
    product_code, Product_type
FROM
    products;
```

product_code	Product_type
Prod001	Own Brand
Prod002	Own Brand
Prod003	Own Brand
Prod004	Own Brand
Prod005	Own Brand
Prod006	Own Brand
Prod007	Own Brand
Prod008	Own Brand
Prod009	Own Brand
Prod010	Own Brand
Prod011	Own Brand
Prod012	Own Brand
Prod013	Own Brand
Prod014	Own Brand
Prod015	Own Brand
Prod016	Own Brand

Show all markets that belong to each zone.

```
SELECT DISTINCT
    markets_name, zone
FROM
    markets
ORDER BY zone;
```

	markets_name	zone
	Bhopal	Central
	Mumbai	Central
	Nagpur	Central
	Ahmedabad	North
	Delhi NCR	North
	Kanpur	North
	Lucknow	North
	Patna	North
	Surat	North
	Bengaluru	South
	Bhubaneshwar	South

Calculate the total sales amount across all transactions .

```sql
SELECT
    SUM(sales_amount) AS 'Total Sales Amount'
FROM
    transactions;
```

Total Sales Amount
984813463

Find total sales amount and total sales quantity by product type.

```sql
SELECT
    product_type AS 'Product Type',
    SUM(sales_amount) AS 'Total Sales Amount',
    SUM(sales_qty) AS 'Total Sales Quantity'
FROM
    transactions t
        JOIN
    products p ON t.product_code = p.product_code
GROUP BY product_type;
```

	Product Type	Total Sales Amount	Total Sales Quantity
▶	Distribution	146039576	496576
	Own Brand	369812648	1513287

Identify the top 5 customers based on total sales amount.

```sql
SELECT
    custmer_name AS 'Customer Name',
    SUM(sales_amount) AS 'Total Sales Amount'
FROM
    transactions t
        JOIN
    customers c ON t.customer_code = c.customer_code
GROUP BY custmer_name
ORDER BY 'Total Sales Amount' DESC
LIMIT 5;
```

Customer Name	Total Sales Amount
Nixon	43893083
Electricalsopedia Stores	10281203
Power	5727123
Epic Stores	18750440
Electricalslytical	49644189

Calculate total sales amount for each market zone.

```sql
SELECT
    Zone, SUM(sales_amount) AS 'Total Sales Amount'
FROM
    transactions t
        JOIN
    markets m ON t.market_code = m.markets_code
GROUP BY zone;
```

Zone	Total Sales Amount
Central	263720983
North	675532517
South	45559963

Show monthly sales trends (total sales amount per month and year).

```sql
SELECT
    cy_date AS `Date`,
    CONCAT(`year`, ' ', month_name) AS 'Time of Year',
    SUM(sales_amount) AS 'Total Sales Amount'
FROM
    transactions t
        JOIN
    date d ON t.order_date = d.`date`
GROUP BY CONCAT(`year`, ' ', month_name) , cy_date
ORDER BY cy_date ASC;
```

Date	Time of Year	Total Sales Amount
2017-10-01	2017 October	26087017
2017-11-01	2017 November	34998960
2017-12-01	2017 December	31796676
2018-01-01	2018 January	42520492
2018-02-01	2018 February	35243966
2018-03-01	2018 March	38153155
2018-04-01	2018 April	35891903
2018-05-01	2018 May	32188350
2018-06-01	2018 June	34744297
2018-07-01	2018 July	35975169
2018-08-01	2018 August	39454467
2018-09-01	2018 September	30101586
2018-10-01	2018 October	27713450
2018-11-01	2018 November	31296202
2018-12-01	2018 December	30404126
2019-01-01	2019 January	31530566

Find the average order value (sales amount per transaction) for each customer type.

```sql
SELECT
    customer_type AS 'Customer Type',
    AVG(sales_amount) AS 'Average Sales Amount'
FROM
    transactions t
        JOIN
    customers c ON t.customer_code = c.customer_code
GROUP BY customer_type
ORDER BY AVG(sales_amount) DESC;
```

Customer Type	Average Sales Amount
Brick & Mortar	7739.57623453581 4
E-Commerce	4603.843022698975

Identify the top 3 products in each market based on total sales amount

```sql
WITH product_sales AS (
    SELECT
        t.market_code,
        t.product_code,
        SUM(t.sales_amount) AS total_sales
    FROM transactions t
    GROUP BY
        t.market_code,
        t.product_code
),
ranked_products AS (
    SELECT
        market_code,
        product_code,
        total_sales,
        DENSE_RANK() OVER (
            PARTITION BY market_code
            ORDER BY total_sales DESC
        ) AS sales_rank
    FROM product_sales
)
```

```sql
SELECT
    market_code,
    product_code,
    total_sales
FROM ranked_products
WHERE sales_rank <= 3
ORDER BY market_code, total_sales DESC;
```

Identify the top 3 products in each market based on total sales amount

market_code	product_code	total_sales
Mark001	Prod319	2965403
Mark001	Prod040	2280636
Mark001	Prod328	1939812
Mark002	Prod318	30443024
Mark002	Prod018	14594472
Mark002	Prod306	12439440
Mark003	Prod318	23986404
Mark003	Prod060	12888932
Mark003	Prod327	11470504
Mark004	Prod316	58733953
Mark004	Prod324	36828751
Mark004	Prod329	34319592
Mark005	Prod324	1856503
Mark005	Prod040	1561194
Mark005	Prod298	1555410
Mark006	Prod024	290750
Mark006	Prod334	26255
Mark006	Prod106	17194
Mark007	Prod334	9502847
Mark007	Prod337	2949851
Mark007	Prod106	2425380
Mark008	Prod318	1317495
Mark008	Prod334	409797
Mark008	Prod306	373846
Mark009	Prod324	1873574
Mark009	Prod339	482678

Analyze year-over-year sales growth by comparing total sales amount for each year.

```sql
WITH yearly_sales AS (
    SELECT
        d.year,
        SUM(t.sales_amount) AS total_sales
    FROM transactions t
    JOIN date d
        ON t.order_date = d.date
    GROUP BY d.year
),
yoy_growth AS (
    SELECT
        year,
        total_sales,
        LAG(total_sales) OVER (ORDER BY year) AS previous_year_sales
    FROM yearly_sales
)
```

year	total_sales	previous_year_sales	yoy_growth_percentage
2017	92882653	NULL	NULL
2018	413687163	92882653	345.39
2019	336019102	413687163	-18.77
2020	142224545	336019102	-57.67

```sql
SELECT
    year,
    total_sales,
    previous_year_sales,
    ROUND(
        (total_sales - previous_year_sales) / previous_year_sales * 100,
        2
    ) AS yoy_growth_percentage
FROM yoy_growth
ORDER BY year;
```

Find customers who have placed orders in more than one market zone.

```sql
SELECT
    c.customer_code,
    c.custmer_name,
    COUNT(DISTINCT m.zone) AS number_of_zones
FROM
    transactions t
        JOIN
    customers c ON t.customer_code = c.customer_code
        JOIN
    markets m ON t.market_code = m.markets_code
GROUP BY c.customer_code , c.custmer_name
HAVING COUNT(DISTINCT m.zone) > 1
ORDER BY number_of_zones DESC;
```

customer_code	custmer_name	number_of_zones
Cus002	Nomad Stores	3
Cus003	Excel Stores	3
Cus004	Surface Stores	3
Cus006	Electricalsara Stores	3
Cus007	Info Stores	3
Cus011	Flawless Stores	3
Cus023	Sound	3
Cus026	Insight	3
Cus005	Premium Stores	2
Cus008	Acclaimed Stores	2
Cus010	Atlas Stores	2
Cus012	Integration Stores	2
Cus013	Unity Stores	2
Cus014	Forward Stores	2
Cus015	Electricalsbea Stores	2
Cus017	Epic Stores	2
Cus018	Electricalslance Stores	2
Cus020	Nixon	2
Cus021	Modular	2
Cus028	Sage	2
Cus031	Zone	2
Cus032	Elite	2
Cus033	All-Out	2

Detect seasonality by identifying the month with the highest sales amount for each year.

```sql
WITH monthly_sales AS (
    SELECT
        d.year,
        d.month_name,
        SUM(t.sales_amount) AS total_sales
    FROM transactions t
    JOIN date d
        ON t.order_date = d.date
    GROUP BY
        d.year,
        d.month_name
),
ranked_months AS (
    SELECT
        year,
        month_name,
        total_sales,
        DENSE_RANK() OVER (
            PARTITION BY year
            ORDER BY total_sales DESC
        ) AS sales_rank
    FROM monthly_sales
)
SELECT
    year,
    month_name,
    total_sales
FROM ranked_months
WHERE sales_rank = 1
ORDER BY year;
```

year	month_name	total_sales
2017	November	34998960
2018	January	42520492
2019	July	35111879
2020	February	26924799

Create a customer segmentation report that classifies customers as:

- High Value (top 20% by total sales),
- Medium Value (middle 60%),
- Low Value (bottom 20%).

```sql
WITH customer_sales AS (
    SELECT
        c.customer_code,
        c.custmer_name,
        SUM(t.sales_amount) AS total_sales
    FROM transactions t
    JOIN customers c
        ON t.customer_code = c.customer_code
    GROUP BY
        c.customer_code,
        c.custmer_name
),
ranked_customers AS (
    SELECT
        customer_code,
        custmer_name,
        total_sales,
        NTILE(5) OVER (ORDER BY total_sales DESC) AS sales_bucket
    FROM customer_sales
)
```

```sql
SELECT
    customer_code,
    custmer_name,
    total_sales,
    CASE
        WHEN sales_bucket = 1 THEN 'High Value'
        WHEN sales_bucket IN (2, 3, 4) THEN 'Medium Value'
        WHEN sales_bucket = 5 THEN 'Low Value'
    END AS customer_segment
FROM ranked_customers
ORDER BY total_sales DESC;
```

Create a customer segmentation report that classifies customers as:

- High Value (top 20% by total sales),
- Medium Value (middle 60%),
- Low Value (bottom 20%).

customer_code	custmer_name	total_sales	customer_segment
Cus029	Electricalsocity	17489935	Medium Value
Cus021	Modular	17379851	Medium Value
Cus010	Atlas Stores	16666713	Medium Value
Cus038	Leader	16529970	Medium Value
Cus004	Surface Stores	15242562	Medium Value
Cus012	Integration Stores	13979716	Medium Value
Cus016	Logic Stores	13201366	Medium Value
Cus025	Path	12995938	Medium Value
Cus013	Unity Stores	12589257	Medium Value
Cus019	Electricalsopedia St...	10281203	Medium Value
Cus011	Flawless Stores	9156412	Medium Value
Cus030	Synthetic	6173068	Medium Value
Cus033	All-Out	6068432	Medium Value
Cus024	Power	5727123	Medium Value
Cus035	Relief	5230158	Medium Value
Cus031	Zone	5067349	Medium Value
Cus023	Sound	4966707	Medium Value
Cus032	Elite	4837239	Medium Value
Cus037	Propel	4183862	Medium Value
Cus026	Insight	3342051	Low Value
Cus036	Novus	2359799	Low Value
Cus028	Sage	2252506	Low Value
Cus018	Electricalslance Stores	1868461	Low Value
Cus009	Electricalsquipo Sto...	1330361	Low Value
Cus034	Expression	430368	Low Value
Cus015	Electricalsbea Stores	336367	Low Value

POWER BI

KEY INSIGHTS

₹142M
Revenue

350K
Sales Qty

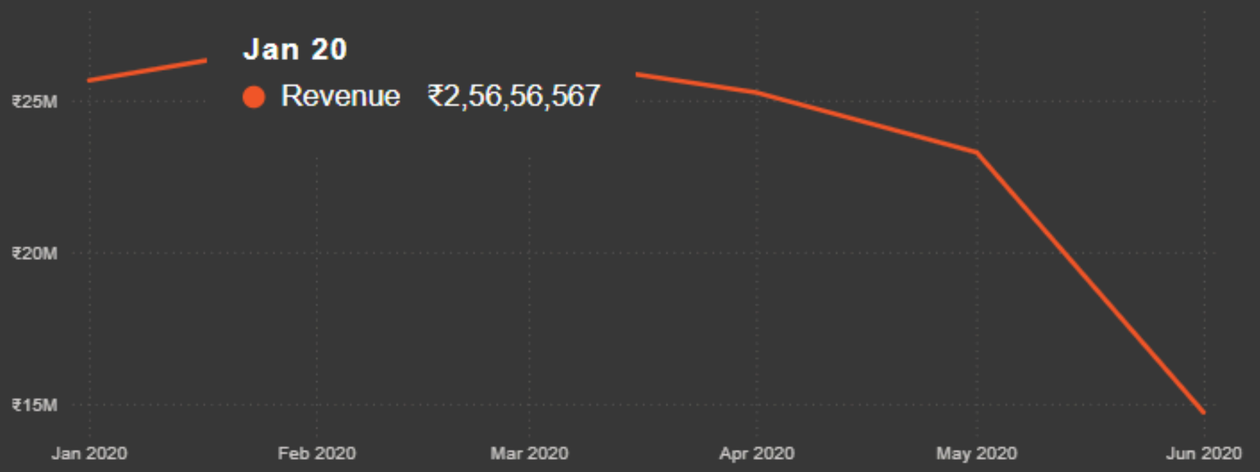
2017 | 2018 | 2019 | 2020

Jan 20 | Feb 20 | Mar 20 | Apr 20 | May 20 | Jun 20

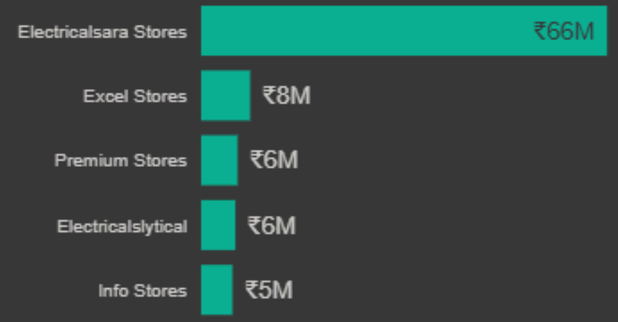
Revenue by Markets

Sales Qty by Markets

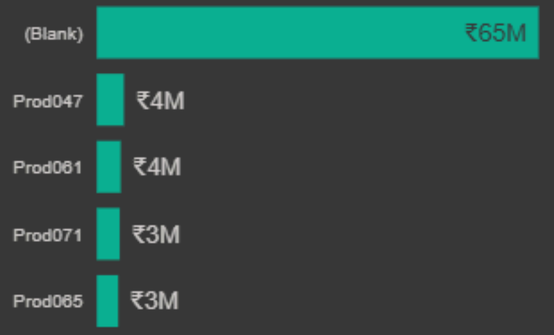
Revenue Trend



Top 5 Customers



Top 5 Products



PROFIT ANALYSIS

2017 | 2018 | 2019 | 2020

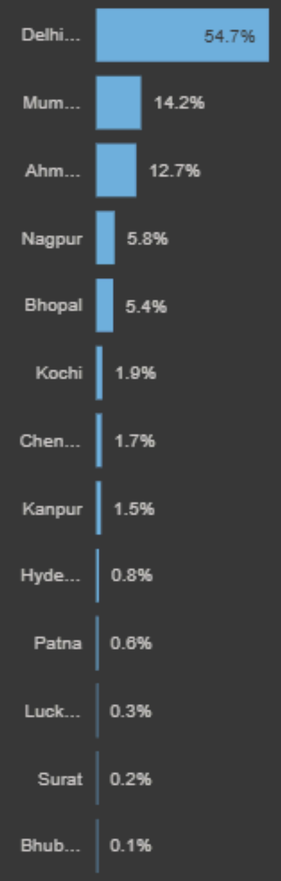
Jan 20 | Feb 20 | Mar 20 | Apr 20 | May 20 | Jun 20

₹142M Revenue

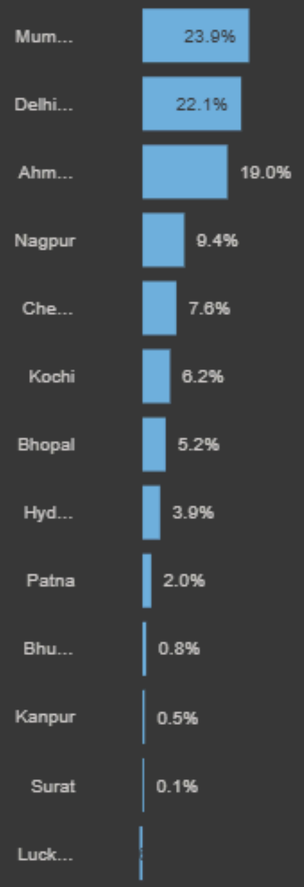
350K Sales Qty

₹2.1M Total Profit Margin

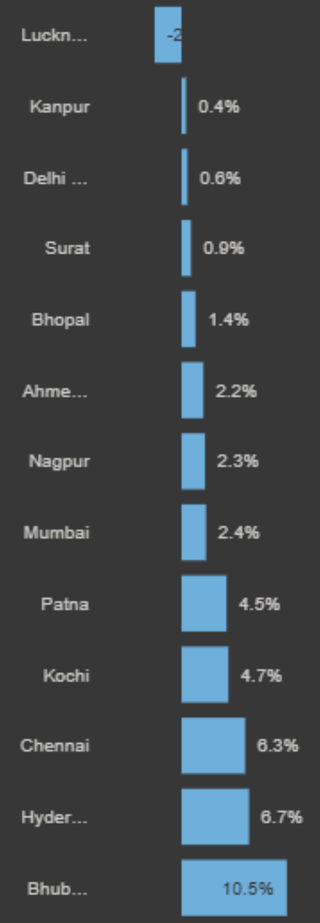
Revenue Contribution % by Market



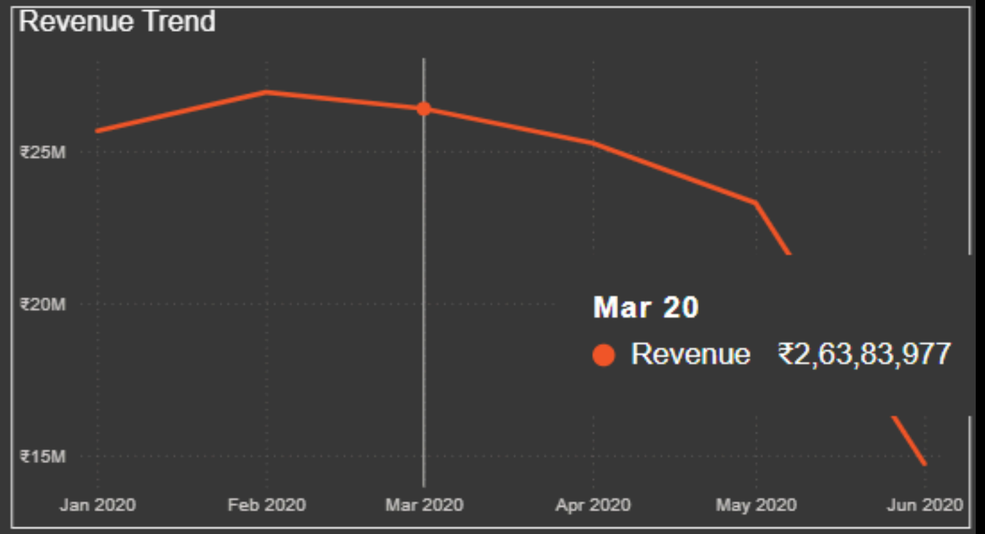
Profit Contribution % by Market



Profit % by market



Revenue Trend



custmer_name	Revenue	Revenue Contribution %	Profit Margin Contribution %	Profit Margin %
Electricalsquipo Stores	₹86,908	0.1%	-0.5%	-11.5%
Epic Stores	₹30,02,834	2.1%	-6.8%	-4.7%
Expression	₹82,845	0.1%	-0.2%	-4.2%
Insight	₹5,71,430	0.4%	-0.5%	-1.9%
Synthetic	₹9,16,267	0.6%	-0.4%	-1.0%
Integration Stores	₹19,09,611	1.3%	-0.2%	-0.2%
Relief	₹7,73,301	0.5%	0.1%	0.3%
Electricalsara Stores	₹6,56,41,977	46.2%	11.9%	0.4%
Logic Stores	₹33,19,243	2.3%	0.6%	0.4%
Electricalslytical	₹55,37,904	3.9%	1.3%	0.5%
Premium Stores	₹58,99,748	4.1%	1.4%	0.5%
Zone	₹9,27,154	0.7%	0.2%	0.5%
Path	₹17,23,307	1.2%	1.0%	1.1%
Control	₹41,82,662	2.9%	2.5%	1.2%
Sound	₹7,40,856	0.5%	0.5%	1.3%
Novus	₹2,58,543	0.2%	0.2%	1.6%
Atlas Stores	₹21,89,613	1.5%	2.1%	2.0%
Electricalsocity	₹25,20,938	1.8%	3.3%	2.7%
Surface Stores	₹24,20,004	1.7%	2.2%	2.7%
Total	**₹14,22,24,545**	**100.0%**	**100.0%**	**1.4%**

PERFORMANCE INSIGHTS

₹142M
Revenue

350K
Sales Qty

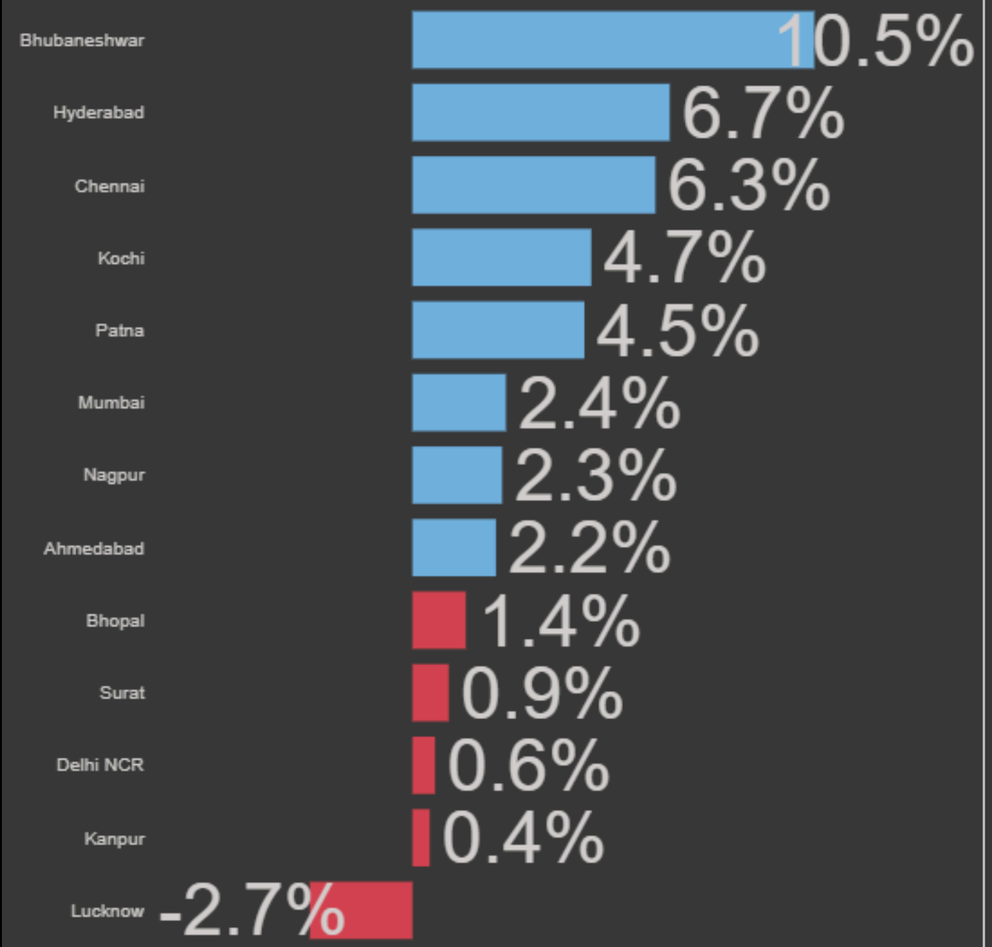
₹2.1M
Total Profit Margin

Profit Target
2%

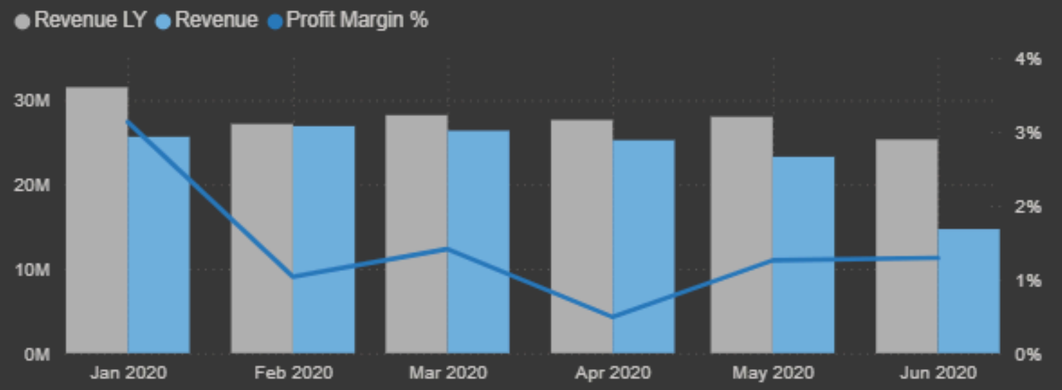
2017 | 2018 | 2019 | 2020

Jan 20 | Feb 20 | Mar 20 | Apr 20 | May 20 | Jun 20

Revenue Contribution % by Market



Revenue Trend



custmer_name	Revenue	Revenue Contribution %	Profit Margin Contribution %	Profit Margin %
Electricalsara Stores	₹6,56,41,977	46.2%	11.9%	0.4%
Excel Stores	₹79,28,385	5.6%	12.5%	3.3%
Premium Stores	₹58,99,748	4.1%	1.4%	0.5%
Electricalslytical	₹55,37,904	3.9%	1.3%	0.5%
Info Stores	₹50,64,374	3.6%	7.8%	3.2%
Control	₹41,82,662	2.9%	2.5%	1.2%
Surge Stores	₹39,53,600	2.8%	11.9%	6.2%
Logic Stores	₹33,19,243	2.3%	0.6%	0.4%
Acclaimed Stores	₹31,16,384	2.2%	6.1%	4.0%
Nixon	₹30,28,345	2.1%	4.7%	3.2%
Epic Stores	₹30,02,834	2.1%	-6.8%	-4.7%
Forward Stores	₹25,60,961	1.8%	5.4%	4.3%
Electricalsocity	₹25,20,938	1.8%	3.3%	2.7%
Modular	₹24,53,519	1.7%	5.0%	4.2%
Surface Stores	₹24,20,004	1.7%	3.2%	2.7%
Atlas Stores	₹21,89,613	1.5%	2.1%	2.0%
Nomad Stores	₹19,78,550	1.4%	4.0%	4.1%
Integration Stores	₹19,09,611	1.3%	-0.2%	-0.2%
Total	₹14,22,24,545	100.0%	100.0%	1.4%

BUSINESS IMPACT

- Improved decision-making speed and accuracy by converting raw sales data into actionable insights using MySQL and Power BI.
- Identified high-performing products, customers, and markets, enabling targeted growth and optimization strategies.
- Revealed profitability and performance gaps, helping the business reduce losses and improve margins.
- Enhanced sales forecasting and planning through seasonality and year-over-year performance analysis.
- Streamlined reporting with automated dashboards, reducing manual effort and enabling real-time performance tracking.

THANKS