

# Maven Mexico Toy Sales Analysis using SQL and Power BI

By

P Ajay Kumar



[Github Link](#)

# Project Overview:

This project involves analyzing sales data from toy stores across Mexico. The dataset includes product, store, sales, inventory, and calendar information. The goal is to gain insights into product performance, store profitability, sales trends, and inventory management.

The analysis is performed using SQL for data extraction, transformation, and aggregation, followed by visualization and reporting in Power BI.

Tools Used : PostgreSQL , Power BI

# Dataset Overview: -

The dataset consists of the following tables:

- **Products:** Contains product information, including cost, price, and category.
- **Stores:** Information about the stores, including location and opening date.
- **Sales:** Records of daily sales, including the date, store, product, and units sold.
- **Inventory:** Tracks stock availability of products across stores.
- **Calendar:** A date dimension table for time-based analysis.

Data Set Link : - [Maven Toys Datatset](#)

# Database Schema : -

## Table Structures:

### 1.Products Table

product\_id, product\_name,  
product\_category, product\_cost,  
product\_price

### 2.Stores Table

store\_id, store\_name, store\_city,  
store\_location, store\_open\_date

### 3.Sales Table

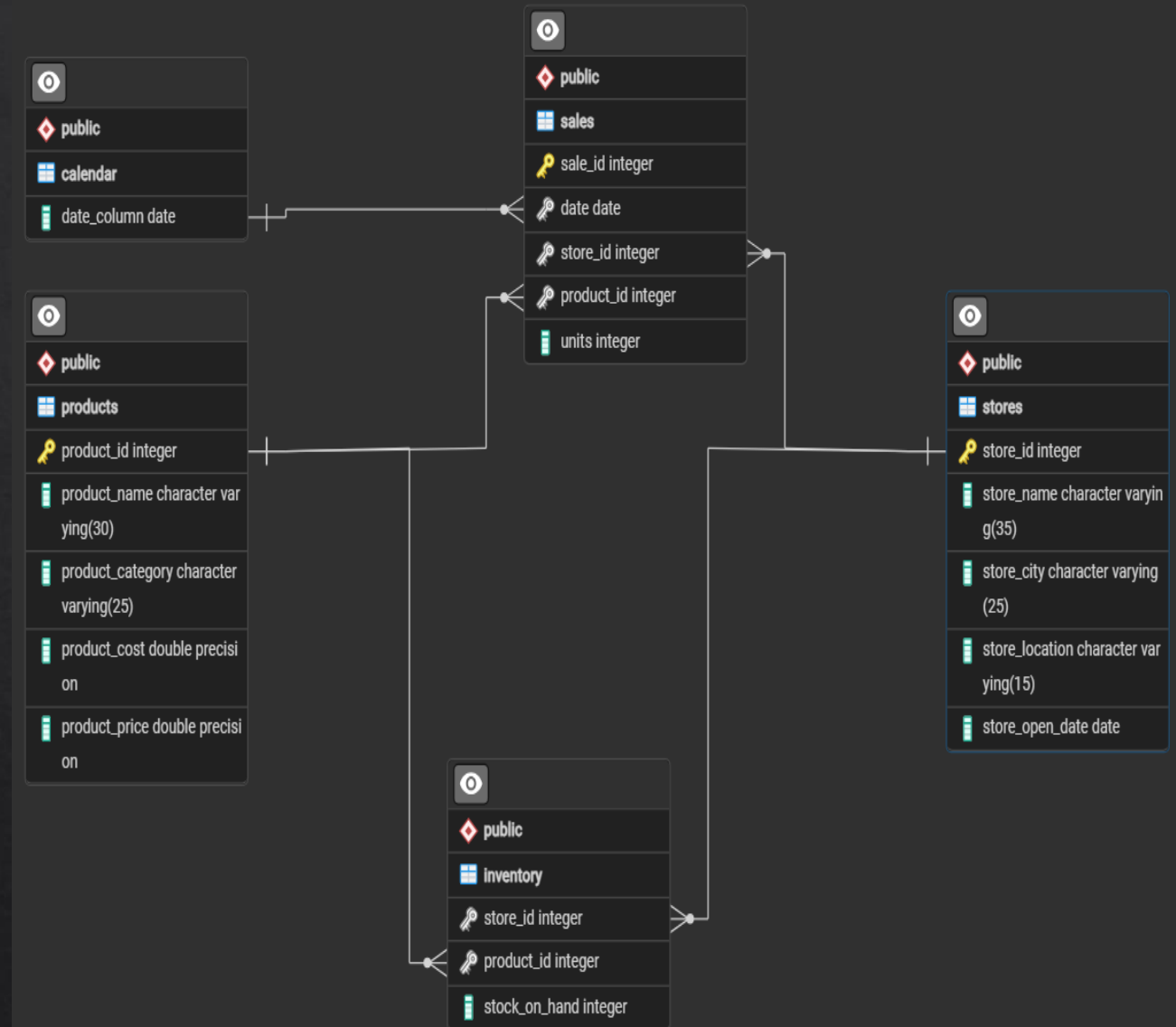
sale\_id, date, store\_id,  
product\_id, units

### 4.Inventory Table

store\_id, product\_id,  
stock\_on\_hand

### 5.Calendar Table

date\_column



# Data Cleaning and Preparation: -

## Key Steps:

**1.Handling Null Values:** Checked for nulls in all tables using SQL queries. No null values were found.

**2.Checking for Duplicates:** Verified duplicates in the sales, stores, and products tables. No duplicates were detected.

**3.Standardization:** Standardized store\_name and other text fields by converting them to lowercase and removing any leading/trailing spaces.




# Analysis and Insights

Key Metrics & KPI'S: -

```
-- Calculate the total revenue generated
SELECT
    ROUND(CAST(SUM(units * product_price) AS numeric), 2) AS total_revenue
FROM sales AS s
JOIN products AS p
ON s.product_id = p.product_id;
```

	total_revenue numeric 
1	14444572.35

```
-- Calculate the total cost of goods
SELECT
    ROUND(CAST(SUM(units * product_cost) AS numeric), 2) AS total_cogs
FROM sales AS s
JOIN products AS p
ON s.product_id = p.product_id;
```

	total_cogs numeric 
1	10430543.35

-- Calculate the total profit


SELECT

ROUND(CAST(SUM(units \* (product\_price - product\_cost)) AS numeric), 2) AS total\_profit

FROM sales AS s

JOIN products AS p

ON s.product\_id = p.product\_id;


	total_profit numeric 
1	4014029.00

-- Calculate the total units sold

SELECT

SUM(units) AS total\_units\_sold

FROM sales;


	total_units_sold bigint 
1	1090565

-- Calculate the avg units sold per day

SELECT

ROUND(CAST(AVG(units) AS numeric), 2) AS avg\_units

FROM sales;

	avg_units numeric 
1	1.32

```
-- Find the average price for each product category.
```

```
SELECT
    product_category,
    ROUND(CAST(AVG(product_price) AS numeric), 2) AS avg_price
FROM products
GROUP BY product_category;
```

	product_category character varying (25) 🔒	avg_price numeric 🔒
1	Sports & Outdoors	15.28
2	Electronics	20.66
3	Art & Crafts	13.12
4	Toys	15.99
5	Games	12.37

```
-- Get the total number of sales transactions.
```

```
SELECT
    COUNT(sale_id) AS total_sales
FROM sales;
```

	total_sales bigint 🔒
1	829262



1. Identify the top 5 products with the highest units sold.

Output

```
SELECT
    s.product_id,
    pr.product_name,
    SUM(s.units) AS total_units_sold
FROM sales AS s
JOIN products AS pr
ON s.product_id = pr.product_id
GROUP BY s.product_id , pr.product_name
ORDER BY SUM(s.units) DESC
LIMIT 5;
```

	product_id integer	product_name character varying (30)	total_units_sold bigint
1	6	Colorbuds	104368
2	25	PlayDoh Can	103128
3	3	Barrel O' Slime	91663
4	8	Deck Of Cards	84034
5	19	Magic Sand	60598

Insight: - Color Buds & PlayDoh Can are best selling products with over 1 lakh units sold each

2. Calculate the total revenue (units sold \* product price) per store.

```
SELECT
    s.store_id,
    ROUND(CAST(SUM(s.units*pr.product_price) AS numeric), 0) AS total_revenue
FROM sales AS s
JOIN products AS pr
ON s.product_id = pr.product_id
GROUP BY s.store_id
ORDER BY ROUND(CAST(SUM(s.units*pr.product_price) AS numeric), 2) DESC;
```

## Output

	store_id integer	total_revenue numeric
1	31	554553
2	30	449355
3	9	433556
4	17	411157
5	7	372999
6	46	348467
7	42	344847
8	39	344307
9	37	337425
10	4	330409
Total rows: 50 of 50		
Query com		

3. Show the stock availability for the top 5 best-selling products.

```
SELECT
    inv.product_id,
    SUM(stock_on_hand) AS total_stock_available
FROM inventory AS inv
GROUP BY product_id
HAVING product_id IN (
    SELECT
        s.product_id
    FROM sales AS s
    JOIN products AS pr
    ON s.product_id = pr.product_id
    GROUP BY s.product_id
    ORDER BY SUM(s.units) DESC
    LIMIT 5
);
```

Output

product_id integer	total_stock_available bigint
3	1282
6	1159
8	2738
19	1922
25	2129

Insight: - The stocks for top selling products are not low which indicates overall sales are not effected by stock availability

## 4. What are the top 5 Stores for each Product Category sales

```
SELECT
    product_category,
    store_name,
    total_units_sold
FROM(
    SELECT
        product_category,
        store_name,
        SUM(sa.units) AS total_units_sold,
        ROW_NUMBER() OVER(PARTITION BY product_category ORDER BY SUM(sa.units) DESC) AS tn_rank
    FROM sales AS sa
    JOIN products AS pr
    ON sa.product_id = pr.product_id
    JOIN stores AS st
    ON sa.store_id = st.store_id
    GROUP BY product_category, store_name
)
WHERE tn_rank <= 5;
```

## Output

	product_category character varying (25) 🔒	store_name character varying (35) 🔒	total_units_sold bigint 🔒
1	Art & Crafts	maven toys toluca 1	12998
2	Art & Crafts	maven toys ciudad de mexico 2	10871
3	Art & Crafts	maven toys ciudad de mexico 1	10613
4	Art & Crafts	maven toys guadalajara 3	9550
5	Art & Crafts	maven toys xalapa 2	9288
6	Electronics	maven toys ciudad de mexico 2	6928
7	Electronics	maven toys mexicali 1	6325
8	Electronics	maven toys hermosillo 3	5353
9	Electronics	maven toys guadalajara 3	4788
10	Electronics	maven toys guadalajara 4	4696
11	Games	maven toys ciudad de mexico 2	10173
12	Games	maven toys morelia 1	8534
13	Games	maven toys monterrey 3	6779
14	Games	maven toys ciudad de mexico 1	6003
15	Games	maven toys campeche 1	5196
16	Sports & Outdoors	maven toys campeche 1	6827
17	Sports & Outdoors	maven toys monterrey 4	6608
18	Sports & Outdoors	maven toys ciudad de mexico 2	6486
Total rows: 25 of 25			Query complete 00:00:00.723

5. Sales Trend Over Time: Show the total units sold per month.

## Output

```
SELECT
    EXTRACT(MONTH FROM date) AS sales_month,
    TO_CHAR(date, 'Month') AS month_name,
    SUM(units) AS total_units_sold
FROM sales
GROUP BY sales_month, month_name
ORDER BY sales_month;
```

	sales_month numeric	month_name text	total_units_sold bigint
1	1	January	94487
2	2	February	91516
3	3	March	109317
4	4	April	112695
5	5	May	110761
6	6	June	111453
7	7	July	109791
8	8	August	91120
9	9	September	94348
10	10	October	47861
11	11	November	51185
12	12	December	66031

Insight: - Sales are highest in the summer season (March - July)



## 6. Calculate Year-Over-Year Sales Growth percentage

```
SELECT
  EXTRACT(YEAR FROM date) AS sales_year,
  SUM(units) AS total_units_sold,
  LAG(SUM(units)) OVER (ORDER BY EXTRACT(YEAR FROM date)) AS previous_year_sales,
  ROUND((SUM(units) - LAG(SUM(units)) OVER (ORDER BY EXTRACT(YEAR FROM date))) * 100.0 / LAG(SUM(units)) OVER (ORDER BY EXTRACT(YEAR FROM date)))
FROM sales
GROUP BY EXTRACT(YEAR FROM date)
ORDER BY sales_year;
```

### Output

	sales_year numeric 🔒	total_units_sold bigint 🔒	previous_year_sales bigint 🔒	percentage_growth numeric 🔒
1	2022	549492	[null]	[null]
2	2023	541073	549492	-1.53

Insight: - There is a decrease of 1.53 % in 2023 year sales compared to previous year

## 7. Are sales being lost with out-of-stock products at certain locations?

```
SELECT
  city_top_sales.store_id,
  city_top_sales.product_id,
  city_top_sales.total_units_sold,
  stock_status
FROM (
  SELECT
    store_id,
    product_id,
    total_units_sold
  FROM(
    SELECT
      sa.store_id,
      product_id,
      SUM(units) AS total_units_sold,
      ROW_NUMBER() OVER(PARTITION BY sa.store_id ORDER BY SUM(units) DESC) AS city_rank
    FROM sales AS sa
    GROUP BY sa.store_id,sa.product_id
  )
  WHERE city_rank <= 10
) AS city_top_sales
JOIN (
  SELECT
    *,
    CASE NTILE(3) OVER(PARTITION BY store_id ORDER BY stock_on_hand DESC)
      WHEN 1 THEN 'HIGH'
      WHEN 2 THEN 'MEDIUM'
      WHEN 3 THEN 'LOW'
    END AS stock_status
  FROM inventory
) AS inv_rank
ON city_top_sales.store_id = inv_rank.store_id
AND city_top_sales.product_id = inv_rank.product_id
WHERE stock_status = 'LOW';
```

### Output

store_id	product_id	total_units_sold	stock_status
integer	integer	bigint	text
2	6	2053	LOW
2	9	827	LOW
2	1	746	LOW
3	9	901	LOW
4	19	1212	LOW
4	1	1027	LOW
5	3	1428	LOW
5	21	1132	LOW
5	1	876	LOW
6	13	1257	LOW
6	1	908	LOW
7	2	1080	LOW

rows: 78 of 78    Query complete 00:00:00.573

Insight: Out of 500 top 10 selling products in various location 78 have low stock available. About 16% of top product sales are being lost because of less stock available.

8. Which product categories drive the biggest profits? Is this the same across store locations?

```
SELECT
    product_category,
    SUM((s.units*product_price) - (s.units*product_cost)) AS total_profit
FROM sales AS s
JOIN products AS p
ON s.product_id = p.product_id
GROUP BY product_category
ORDER BY SUM((s.units*product_price) - (s.units*product_cost)) DESC
LIMIT 1;
```

Output

	product_category character varying (25) 🔒	total_profit double precision 🔒
1	Toys	1079527

```
SELECT
    store_location,
    product_category AS top_selling_category,
    profit
FROM (
    SELECT
        *,
        ROW_NUMBER() OVER(PARTITION BY store_location ORDER BY profit DESC) AS profit_rank
    FROM (
        SELECT
            st.store_location,
            p.product_category,
            SUM((s.units*product_price) - (s.units*product_cost)) AS profit
        FROM sales AS s
        JOIN products AS p
        ON s.product_id = p.product_id
        JOIN stores AS st
        ON s.store_id = st.store_id
        GROUP BY st.store_location , p.product_category
    )
) WHERE profit_rank = 1 ;
```

	store_location character varying (15) 🔒	top_selling_category character varying (25) 🔒	profit double precision 🔒
1	Airport	Electronics	108197
2	Commercial	Electronics	287574
3	Downtown	Toys	630029
4	Residential	Toys	136214

Insight: - Toys is the biggest profit driving category. Whereas, In two locations toys remains biggest profit driving category and in other two locations Electronics is the biggest profit driving category

# PowerBI DashBoard Overview

The Power BI dashboard for this project visualizes key metrics from the toy sales dataset and provides an intuitive, interactive interface to explore sales performance, profitability, and inventory status.

Concepts Used: -

- 1) Data Cleaning (Validating Datatypes, Splitting the columns for more data etc)
- 2) Data Modelling to connect all tables for better insights
- 3) Calculated Columns and Measures using DAX functions
- 4) Interactive Charts like Bar chart, Donut Chart, Scatter Chart and KPI's etc
- 5) Filters to filter data
- 6) Bookmarks, Page Navigation,



# Maven Toy Sales Dash Board



Sales

Quantity

Report

Revenue

14.44M

COGS

10.43M

Profit

4.01M

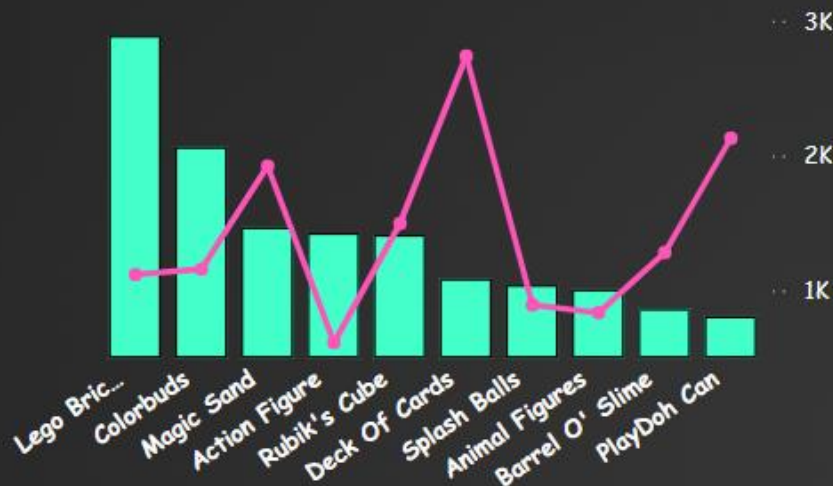
Avg Units

1.32

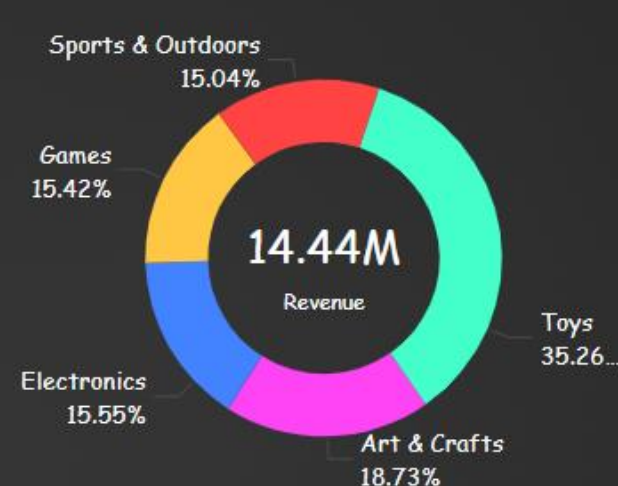


Month  
2022  
All  
2023  
Store Location  
All

Revenue vs Stock for Products [Top 10]



Revenue by product category



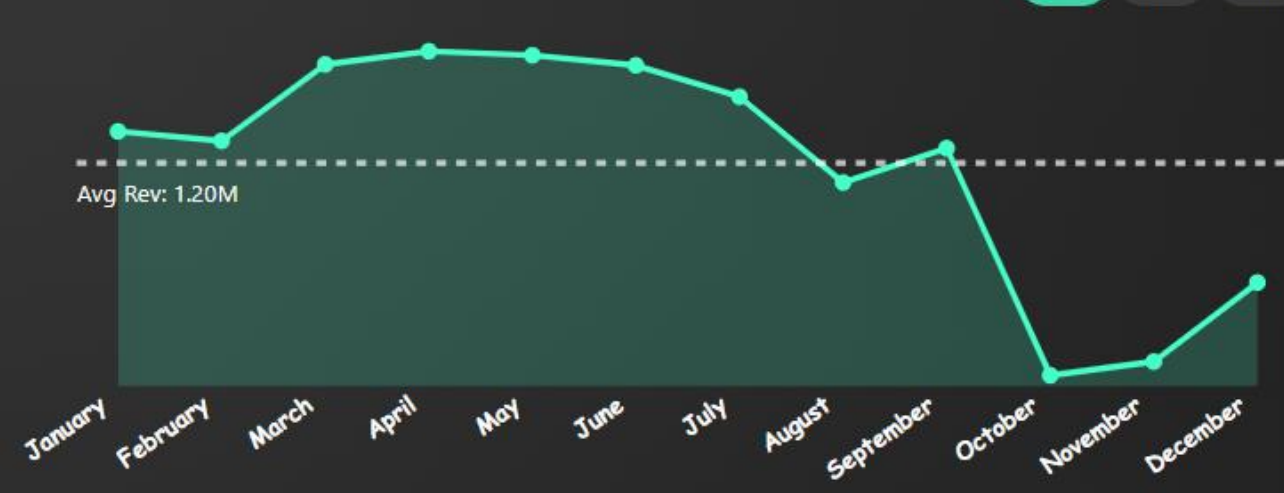
Total Revenue by store city



Revenue vs Units



Revenue trends over period





# Maven Toy Sales Dash Board



Sales

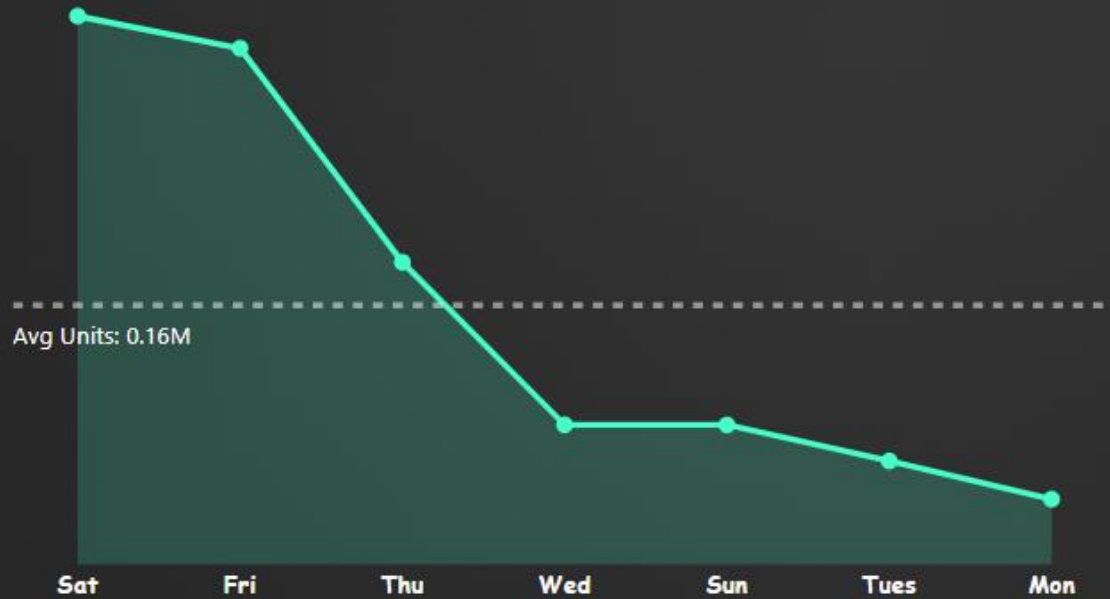
Quantity

Report

### Units sold by products [Top 10]



### Units trends over period



Total Units

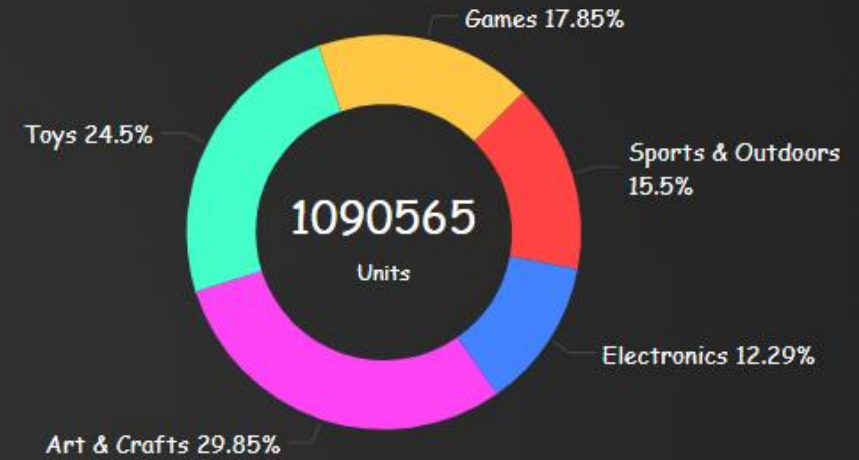
1M

1  
2  
?

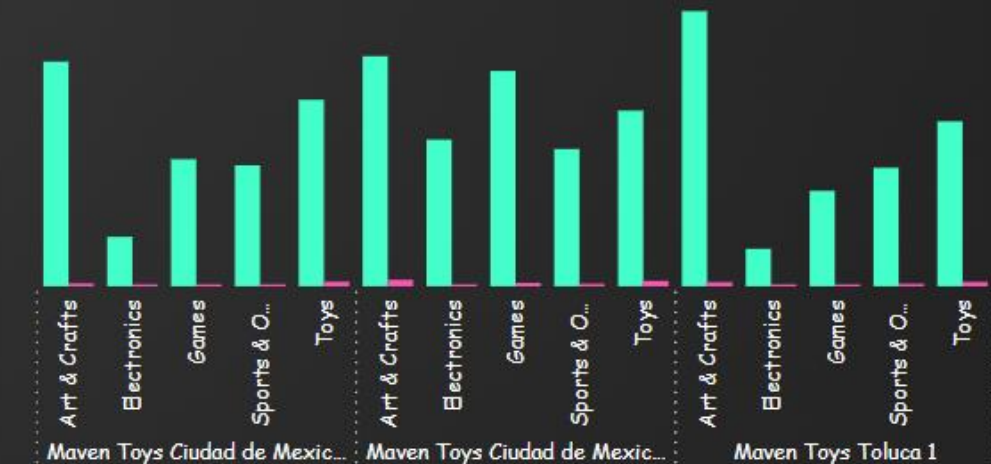


Month  
2022  
All  
2023  
Store City  
All

### Units sold by product category



### Stock Availability in top selling stores



# Conclusion: -

## Key Findings:

- Both revenue and units sold have shown strong growth in the most recent periods, driven primarily by toys and high-demand products like **Lego Bricks** and **ColorBuds**.
- Products with higher stock but lower revenue, such as **Deck of Cards** and **PlayDoh Can**, represent opportunities for promotions or adjustments in marketing strategies to improve turnover.
- The revenue trend analysis highlights clear seasonal peaks, with March-July and November-December being critical periods. Preparing inventory and promotional campaigns around these times can maximize revenue potential.
- While **Toys** continue to lead in revenue, increasing focus on **Electronics, Art & Crafts**, and **Games** can diversify the revenue streams and mitigate risks associated with heavy reliance on a single category.

# Recommendations:

- Targeted Marketing:** Focus marketing campaigns on the top-performing products and categories, particularly during the peak sales periods identified in the trend analysis. Consider promotional discounts or bundle deals for underperforming products with high inventory levels.
- Stock Management:** Optimize stock levels by closely monitoring products with low sales and high stock availability, reallocating resources to products that are consistently in demand but face stock shortages.
- Location-Based Insights:** Populate the **Revenue by Store City** section with data to identify which locations contribute the most to overall revenue, enabling region-specific strategies to boost sales where needed.



