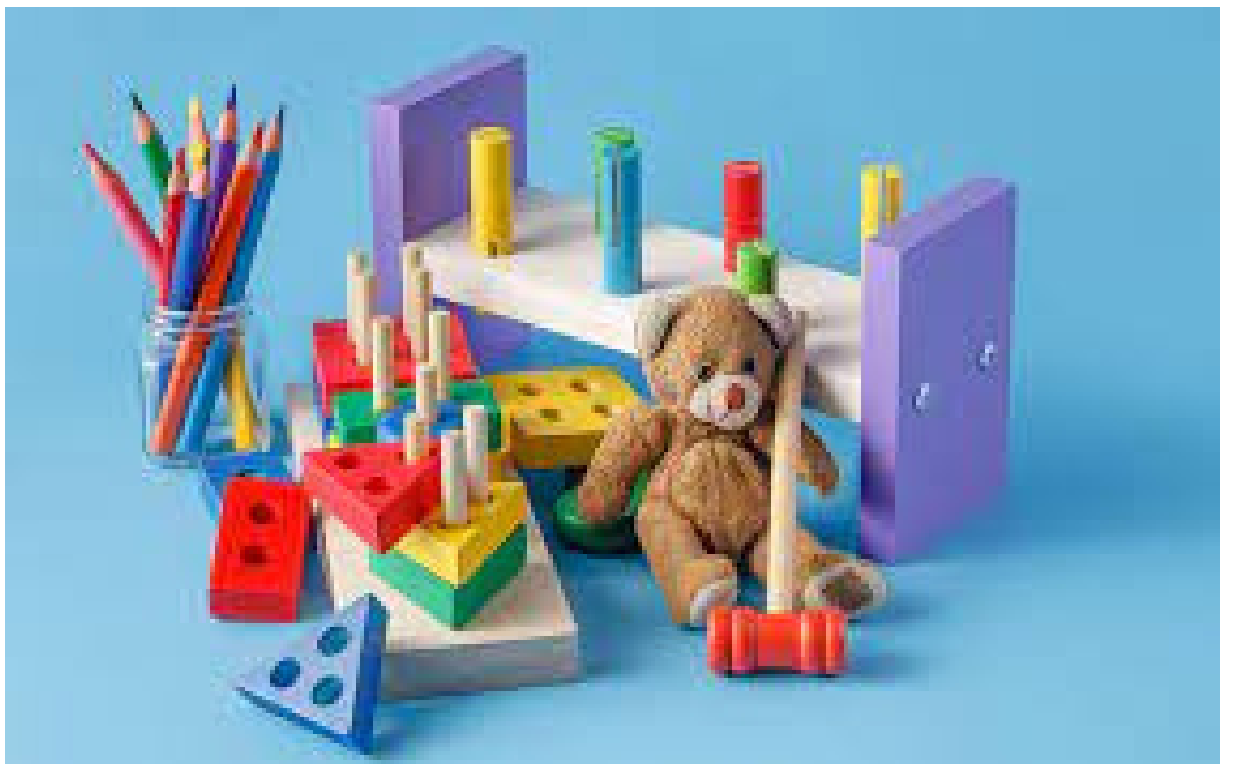
Mexico Toys' Stores Case Study

Based on sales & inventory data for a fictitious chain of toy stores in Mexico called Maven Toys, including information about products, stores, daily sales transactions, and current inventory levels at each location.

Index

- Tables Used
- Questions that are to be answered
- SQL queries and results
- Benefits of this case study



Tables

Produt Table

	product_id [PK] integer	product_name character varying (255)	product_category character varying (255)	product_cost numeric (10,2)	product_price numeric (10,2)
1	1	Action Figure	Toys	9.99	15.99
2	2	Animal Figures	Toys	9.99	12.99
3	3	Barrel O' Slime	Art & Crafts	1.99	3.99
4	4	Chutes & Ladders	Games	9.99	12.99
5	5	Classic Dominoes	Games	7.99	9.99

Sales Table

	sale_id [PK] integer	date date	store_id integer	product_id integer	units integer
1	1	2017-01-01	24	4	1
2	2	2017-01-01	28	1	1
3	3	2017-01-01	6	8	1
4	4	2017-01-01	48	7	1
5	5	2017-01-01	44	18	1

Stores Table

	store_id [PK] integer	store_name character varying (255)	store_city character varying (255)	store_location character varying (255)	store_open_date date
1	1	Maven Toys Guadalajara 1	Guadalajara	Residential	1992-09-18
2	2	Maven Toys Monterrey 1	Monterrey	Residential	1995-04-27
3	3	Maven Toys Guadalajara 2	Guadalajara	Commercial	1999-12-27
4	4	Maven Toys Saltillo 1	Saltillo	Downtown	2000-01-01
5	5	Maven Toys La Paz 1	La Paz	Downtown	2001-05-31

Inventory Table

	store_id [PK] integer	product_id [PK] integer	stock_on_hand integer
1	1	1	27
2	1	2	0
3	1	3	32
4	1	4	6
5	1	5	0

Questions

1. What are the top 5 selling products?
2. What are the sales trends for each product?
3. Which stores are generating the most sales?
4. What is the inventory turnover rate for each product?
5. What are the most popular products in each store?
6. For each city, find the top-selling product.
7. How much money is tied up in inventory at the toy stores?
8. Which products have the highest profit margin (sales revenue minus product cost) in each category?
9. How does the sales performance of each store compare to the average sales of all stores?
10. Which products have been sold in all stores?
11. What is the monthly sales growth rate for each store over the past year?
12. What is the most popular product category in each store?
13. For each product category, which store has the highest total sales in the last quarter?
14. Which products have had a consistent increase in sales for the last three consecutive months?
15. For each store, calculate the cumulative revenue generated from the top-selling product each month.
16. Find the store with the highest total revenue, considering only the sales of products with a price above the average price across all stores.
17. For each city, find the top-selling product.
18. For each product, calculate the percent revenue change for each each month.
19. Find the products which are contributing approximately 50% to the total revenue from all the products.

1. What are the top 5 selling products?

Query Query History

```sql
select product_name, count(units) as total_sales from sales s
join products p
on p.product_id = s.product_id
group by product_name
order by total_sales desc
limit 5
```

Data Output Messages Notifications

	product_name character varying (255)	total_sales bigint
1	Colorbuds	72988
2	Deck Of Cards	68083
3	PlayDoh Can	64834
4	Barrel O' Slime	54078
5	Action Figure	48497

2. What are the sales trends for each product?

Query Query History

```sql
select product_name, extract(month from date) as month_name,
sum(units * product_price) as total_sales
from sales s
join products p
on p.product_id = s.product_id
group by product_name, month_name
limit 40
```

	product_name character varying (255)	month_name numeric	total_sales numeric
1	Action Figure	1	91382.85
2	Action Figure	2	82892.16
3	Action Figure	3	110091.15
4	Action Figure	4	102224.07
5	Action Figure	5	100225.32
6	Action Figure	6	112313.76

Total rows: 40 of 40 Query complete 00:00:01.281

3. Which stores are generating the most sales?

```sql
select store_name,sum(units) as total_sales from sales
join stores
on stores.store_id = sales.store_id
group by store_name
order by total_sales desc
limit 5
```

	store_name character varying (255)	total_sales bigint
1	Maven Toys Ciudad de Mexico 2	42757
2	Maven Toys Ciudad de Mexico 1	33479
3	Maven Toys Toluca 1	32668
4	Maven Toys Guadalajara 3	31609
5	Maven Toys Monterrey 2	28318

4. What is the inventory turnover rate for each product?

```sql
select product_name, (sum(product_cost) / sum(units)) as inventory_turnover_rate
from sales
join products
on products.product_id = sales.product_id
group by product_name
```

	product_name character varying (255)	inventory_turnover_rate numeric
1	Action Figure	8.359243417647261 8
2	Animal Figures	8.242152523727903 0
3	Barrel O' Slime	1.174031179428995 3
4	Chutes & Ladders	9.653434317054061 1
5	Classic Dominoes	7.629011406844106 5
6	Colorbuds	4.888338571209566 2

Total rows: 35 of 35 Query complete 00:00:00.437

5. What are the most popular products in each store?

Query Query History

```sql
with temp
as
(select store_name, product_name, count(product_name) as total_sales,
       rank() over (partition by store_name order by count(product_name) desc) as product_rank
from sales
join stores on stores.store_id = sales.store_id
join products on products.product_id = sales.product_id
group by store_name, product_name
)
select * from temp
where product_rank = 1
```

	store_name character varying (255)	product_name character varying (255)	total_sales bigint	product_rank bigint
1	Maven Toys Aguascalientes 1	Deck Of Cards	1465	1
2	Maven Toys Campeche 1	Mini Ping Pong Set	2198	1
3	Maven Toys Campeche 2	Barrel O' Slime	1152	1
4	Maven Toys Chetumal 1	Colorbuds	1408	1
5	Maven Toys Chihuahua 1	Colorbuds	1465	1
6	Maven Toys Chihuahua 2	Mini Ping Pong Set	2196	1

Total rows: 50 of 50 Query complete 00:00:00.557

6. For each city, find the top-selling product.

```sql
with cte as
(select store_city as city,product_name, sum(units*product_price) as product_revenue
from sales s
join stores st on st.store_id = s.store_id
join products p on p.product_id = s.product_id
group by store_city,product_name
order by store_city,product_revenue desc
)

select distinct city,first_value(product_name)
over(partition by city order by product_revenue desc)
as famous_product from cte
```

	city character varying (255)	famous_product character varying
1	Pachuca	Magic Sand
2	Campeche	Lego Bricks
3	Villahermosa	Colorbuds
4	Durango	Lego Bricks
5	Merida	Lego Bricks
6	Zacatecas	Lego Bricks

Total rows: 29 of 29 Query complete 00:00:00.707

7. Find the product that has most sales in the last 6 months.

```sql
with sales_counts as (
  select product_name, count(sales.product_id) as total_sales
  from sales
    join products
    on products.product_id = sales.product_id
  where date between '2017-01-01' and '2017-06-30'
  group by product_name
)
select product_name, total_sales
from sales_counts
where total_sales = (select max(total_sales) from sales_counts);
```

	product_name character varying (255)	total_sales bigint
1	Colorbuds	28855

8. How much money is tied up in inventory at the toy stores?

Query Query History

```sql
SELECT SUM(product_cost * stock_on_hand) AS total_inventory_value
FROM inventory
JOIN products ON inventory.product_id = products.product_id
JOIN stores ON inventory.store_id = stores.store_id
WHERE stores.store_name LIKE '%Toys%';
```

	total_inventory_value numeric
1	300209.58

9. Which products have the highest profit margin?

Query Query History

```sql
with product_profit as (
  select p.product_name, product_category,
         sum((s.units * p.product_price) - (s.units * p.product_cost)) as profit_margin
  from sales s
  join products p on p.product_id = s.product_id
  group by p.product_name, product_category
),
ranked_products as (
  select *,
         row_number() over (partition by product_category order by profit_margin desc) as ran
  from product_profit
)
select product_category, product_name, profit_margin, rank
from ranked_products
where rank = 1;
```

	product_category character varying (255)	product_name character varying (255)	profit_margin numeric	rank bigint
1	Art & Crafts	Barrel O' Slime	183326.00	1
2	Electronics	Colorbuds	834944.00	1
3	Games	Deck Of Cards	252102.00	1
4	Sports & Outdoors	Nerf Gun	132715.00	1
5	Toys	Action Figure	347748.00	1

10. How does the sales performance of each store compare to the average sales of all stores?

```sql
with temp as (
  select store_name, sum(units * product_price) as store_sale
  from sales s
  join products p on p.product_id = s.product_id
  join stores st on st.store_id = s.store_id
  group by store_name
)
select store_name, store_sale, cast(avg(store_sale) over() as decimal(10,2))
as avg_sale,
cast(((store_sale - avg(store_sale) over()) / store_sale) * 100 as decimal(10,2))
as percent_difference
from temp;
```

Data Output | Messages | Notifications

	store_name character varying (255)	store_sale numeric	avg_sale numeric (10,2)	percent_difference numeric (10,2)
1	Maven Toys Aguascalientes 1	239997.35	288891.45	-20.37
2	Maven Toys Campeche 1	311786.44	288891.45	7.34
3	Maven Toys Campeche 2	206055.23	288891.45	-40.20
4	Maven Toys Chetumal 1	258919.35	288891.45	-11.58
5	Maven Toys Chihuahua 1	248008.30	288891.45	-16.48
6	Maven Toys Chihuahua 2	268704.74	288891.45	-7.51
7	Maven Toys Chilpancingo 1	242539.73	288891.45	-19.11
8	Maven Toys Ciudad de Mexic...	433556.21	288891.45	33.37
9	Maven Toys Ciudad de Mexic...	554553.43	288891.45	47.91
10	Maven Toys Ciudad de Mexic...	337424.66	288891.45	14.38

11. Which products have been sold in all stores?

Query Query History

```sql
select product_name, count(distinct s.store_id) as no_of_stores
from sales s
join products p on p.product_id = s.product_id
join stores st on st.store_id = s.store_id
group by product_name
having count(distinct s.store_id) = (
  select max(store_count)
  from (
    select count(distinct s.store_id) as store_count
    from sales s
    join products p on p.product_id = s.product_id
    join stores st on st.store_id = s.store_id
    group by p.product_id
  ) as store_counts
)
order by no_of_stores desc;
```

	product_name character varying (255)	no_of_stores bigint
1	Action Figure	50
2	Animal Figures	50
3	Barrel O' Slime	50
4	Colorbuds	50
5	Dart Gun	50
6	Deck Of Cards	50
7	Dinosaur Figures	50
8	Etch A Sketch	50
9	Glass Marbles	50
10	Hot Wheels 5-Pack	50

12. What is the monthly sales growth rate for each store over the past year?

Query Query History

```sql
with monthly_sales as (
  select store_name, extract(month from date) as month,
         sum(units * product_price) as cur_sales
  from sales s
  join products p on p.product_id = s.product_id
  join stores st on st.store_id = s.store_id
  group by store_name, month
),
monthly_growth as (
  select store_name, month,
         cur_sales,
         lag(cur_sales) over (partition by store_name order by month) as previous_month_sales
  from monthly_sales
)
select m.store_name, m.month,
       m.cur_sales, m.previous_month_sales,
       cast((m.cur_sales - m.previous_month_sales) / m.previous_month_sales * 100 as decimal(10,2)) as growt
from monthly_growth m
limit 14
```

	store_name character varying (255)	month numeric	cur_sales numeric	previous_month_sales numeric	growth_rate numeric (10,2)
1	Maven Toys Aguascalientes 1	1	22898.01	[null]	[null]
2	Maven Toys Aguascalientes 1	2	24335.55	22898.01	6.28
3	Maven Toys Aguascalientes 1	3	20952.50	24335.55	-13.90
4	Maven Toys Aguascalientes 1	4	22568.04	20952.50	7.71
5	Maven Toys Aguascalientes 1	5	30355.42	22568.04	34.51
6	Maven Toys Aguascalientes 1	6	18204.65	30355.42	-40.03
7	Maven Toys Aguascalientes 1	7	18578.16	18204.65	2.05
8	Maven Toys Aguascalientes 1	8	16162.31	18578.16	-13.00
9	Maven Toys Aguascalientes 1	9	26791.33	16162.31	65.76
10	Maven Toys Aguascalientes 1	10	12216.93	26791.33	-54.40
11	Maven Toys Aguascalientes 1	11	12106.86	12216.93	-0.90
12	Maven Toys Aguascalientes 1	12	14827.59	12106.86	22.47
13	Maven Toys Campeche 1	1	25321.96	[null]	[null]
14	Maven Toys Campeche 1	2	29049.69	25321.96	14.72

13.What is the most popular product_category in each store?

Query Query History

```sql
with temp as (
  select store_name, product_category, sum(units * product_price) as sales
  from sales s
  join stores st on st.store_id = s.store_id
  join products p on p.product_id = s.product_id
  group by store_name, product_category
)
select store_name, product_category, rank
from (
  select store_name, product_category, rank() over (partition by store_name order by sales)
  as rank
  from temp
) as subquery
where rank = 1;
```

	store_name character varying (255)	product_category character varying (255)	rank bigint
1	Maven Toys Aguascalientes 1	Art & Crafts	1
2	Maven Toys Campeche 1	Electronics	1
3	Maven Toys Campeche 2	Electronics	1
4	Maven Toys Chetumal 1	Electronics	1
5	Maven Toys Chihuahua 1	Games	1

Total rows: 50 of 50 Query complete 00:00:00.460

14. For each product category, which store has the highest total sales in the last quarter?

```sql
with quarterly_sales as (
  select store_name, product_category, sum(units * product_price) as total_sales
  from sales s
  join stores st on st.store_id = s.store_id
  join products p on p.product_id = s.product_id
  where date >= date '2018-09-30' - interval '3 months' and date < date '2018-09-30'
  group by store_name, product_category
)
select product_category, store_name, total_sales, rank
from (
  select product_category, store_name, total_sales,
         rank() over (partition by product_category order by total_sales desc) as rank
  from quarterly_sales
) as subquery
where rank = 1;
```

	product_category character varying (255)	store_name character varying (255)	total_sales numeric	rank bigint
1	Art & Crafts	Maven Toys Toluca 1	19572.39	1
2	Electronics	Maven Toys Puebla 1	9841.17	1
3	Games	Maven Toys Ciudad de Mexico 2	9845.95	1
4	Sports & Outdoors	Maven Toys Ciudad de Mexico 2	16825.22	1
5	Toys	Maven Toys Ciudad de Mexico 2	31102.59	1

15. Find the store with the highest total revenue, considering only the sales of products with a price above the average price across all stores.

Query Query History

```sql
with cte as (
  select store_name, product_name, product_price, cast(avg(product_price)
  over (order by product_name rows between unbounded preceding and unbounded following)
  as decimal(10,2)) as avg_price, sum(units * product_price) as revenue
  from sales s
  join products p on p.product_id = s.product_id
  join stores st on st.store_id = s.store_id
  group by store_name, product_name, product_price
  order by store_name, product_name
)
select store_name, sum(revenue) as total_revenue
from cte
where product_price > avg_price
group by store_name
order by total_revenue desc
limit 1;
```

	store_name character varying (255)	total_revenue numeric
1	Maven Toys Ciudad de Mexico 2	368621.48

16.For each product, calculate the percent revenue change for each each month.

```sql
select *,
  cast(((revenue::float - prev_month_rev::float) / revenue::float) * 100 as decimal(10,2))
  as perc_change
from (
  with cte as (
    select
      product_name,
      extract(month from date) as month,
      count(sale_id) as revenue
    from sales s
    join products p on p.product_id = s.product_id
    group by product_name, extract(month from date)
  )
  select *,
    lag(revenue) over(partition by product_name order by month) as prev_month_rev
  from cte
) x
limit 100;
```

	product_name character varying (255)	month numeric	revenue bigint	prev_month_rev bigint	perc_change numeric (10,2)
1	Action Figure	1	4769	[null]	[null]
2	Action Figure	2	4401	4769	-8.36
3	Action Figure	3	5629	4401	21.82
4	Action Figure	4	5332	5629	-5.57
5	Action Figure	5	5422	5332	1.66
6	Action Figure	6	5841	5422	7.17
7	Action Figure	7	3978	5841	-46.83
8	Action Figure	8	3538	3978	-12.44
9	Action Figure	9	3311	3538	-6.86
10	Action Figure	10	2119	3311	-56.25
11	Action Figure	11	2070	2119	-2.37
12	Action Figure	12	2087	2070	0.81
13	Animal Figures	1	2517	[null]	[null]
14	Animal Figures	2	2727	2517	7.70

17. Which products have had a consistent increase in sales for the last three consecutive months?

Query Query History

```sql
with monthly_sales as (
  select p.product_id, extract(month from date) as month,
         sum(units * product_price) as total_sales
  from sales s
  join products p on p.product_id = s.product_id
  where date < date '2018-09-30' and date >= date '2018-09-30' - interval '3 months'
  group by p.product_id, month
),
monthly_growth as (
  select product_id, month,
         total_sales,
         lag(total_sales) over (partition by product_id order by month)
         as previous_month_sales
  from monthly_sales
),
temp as (
  select p.product_name, m.month,
         m.total_sales, m.previous_month_sales,
         cast((m.total_sales - m.previous_month_sales) / m.previous_month_sales * 100
         as decimal(10,2)) as growth_rate
  from monthly_growth m
  join products p on p.product_id = m.product_id
)
select distinct product_name
from temp
where month in (7, 8, 9) and growth_rate > 0
group by product_name
having count(distinct month) = 3;
```

	product_name character varying (255)
1	Action Figure

18. For each store, calculate the cumulative revenue generated from the top-selling product each month.

Query Query History

```sql
WITH top_product AS (
  SELECT store_id, product_name,
         SUM(units * product_price) AS revenue,
         RANK() OVER (PARTITION BY store_id ORDER BY SUM(units * product_price) DESC) AS rnk
  FROM sales s
  JOIN products p ON p.product_id = s.product_id
  GROUP BY store_id, product_name
), monthly_sales AS (
  SELECT store_id, extract(month FROM date) AS year_month,
         product_name,
         SUM(units * product_price) AS revenue
  FROM sales s
  JOIN products p ON p.product_id = s.product_id
  GROUP BY store_id, product_name, extract(month FROM date)
)
SELECT m.store_id, m.product_name, m.year_month,
       SUM(m.revenue) OVER (PARTITION BY m.store_id
ORDER BY m.year_month ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS cumulative_revenue
FROM monthly_sales m
JOIN top_product t ON t.store_id = m.store_id AND t.product_name = m.product_name
WHERE t.rnk = 1
ORDER BY m.store_id, m.year_month;
```

	store_id integer	product_name character varying (255)	year_month numeric	cumulative_revenue numeric
1	1	Lego Bricks	1	3839.04
2	1	Lego Bricks	2	8117.97
3	1	Lego Bricks	3	10357.41
4	1	Lego Bricks	4	14556.36
5	1	Lego Bricks	5	19835.04
6	1	Lego Bricks	6	24753.81
7	1	Lego Bricks	7	26193.45
8	1	Lego Bricks	8	28392.90
9	1	Lego Bricks	9	31312.17
10	1	Lego Bricks	10	32631.84

19. Find the products which are contributing approximately 50% to the total revenue from all the products.

```sql
select product_name, prod_revenue, cum_revenue, cast(perc_total_revenue as decimal(10,2))
from (
  with cte as (
    select
      product_name,
      sum(product_price * units) as prod_revenue
    from sales s
    join products p on p.product_id = s.product_id
    group by product_name
  ),
  cte2 as (
    select
      *,
      sum(prod_revenue) over (order by prod_revenue desc
      rows between unbounded preceding and current row) as cum_revenue
    from cte
  )
  select
    product_name,
    prod_revenue,
    cum_revenue,
    (cum_revenue / max(cum_revenue) over ()) * 100 as perc_total_revenue
  from cte2
) x
where perc_total_revenue < 51;
```

	product_name character varying (255)	prod_revenue numeric	cum_revenue numeric	perc_total_revenue numeric (10,2)
1	Lego Bricks	2388882.63	2388882.63	16.54
2	Colorbuds	1564476.32	3953358.95	27.37
3	Magic Sand	968962.02	4922320.97	34.08
4	Action Figure	926748.42	5849069.39	40.49
5	Rubik's Cube	912983.28	6762052.67	46.81
6	Deck Of Cards	587397.66	7349450.33	50.88

Total rows: 6 of 6 Query complete 00:00:00.345

Benefits from this case study:

By finding the answers to the above questions, one can gain valuable insights into the toys market. Analyzing sales data and trends can provide a clear understanding of the market's dynamics.
Some of the benefits are:

- Gain valuable insights into the toys market.
- Understand market dynamics, product popularity, sales patterns, and revenue generation.
- Identify top-selling products for focused stocking and promotion.
- Analyze sales performance of each store to identify high-performing locations and areas for improvement.
- Optimize stock levels and reduce tied-up capital in inventory through inventory turnover rate evaluation.
- Identify the most profitable product categories through profit margin analysis.
- Gain insights into market competitiveness and expansion opportunities through sales growth rate analysis.
- Assess the contribution of individual products to overall sales by tracking revenue from top-selling products.
- Make informed decisions and allocate resources based on products with consistent sales growth and availability in all stores.
- Evaluate the performance of products over time by examining revenue change percentage.
- Target specific markets and develop strategies based on top-selling products in each city.