DATA ANALYSIS SQL PROJECT

Insights for Maven Toys Sales

Yen Phi Do

1. INTRODUCTION

This portfolio project is geared towards showcasing my proficiency in SQL through an in-depth analysis of the Mexico Toy Sales dataset sourced from Kaggle. This dataset offers valuable insights into toy sales trends in Mexico, enabling a thorough exploration of sales patterns, customer behavior, and product performance. By harnessing the power of SQL, I will extract, transform, and analyze the dataset to unveil significant findings and trends that can guide strategic business decisions.

Throughout this project, I will demonstrate my prowess in crafting effective SQL queries, manipulating data to extract meaningful insights, and presenting findings in a concise and coherent manner. Through the analysis of the Mexico Toy Sales dataset, my objective is to gain insights into the drivers of toy sales, identify top-performing products, discern customer preferences, and uncover avenues for enhancing sales performance.

To execute this project, I employed Oracle SQL Developer, a robust tool tailored for database development and management using Oracle Database. With its intuitive interface and comprehensive features, Oracle SQL Developer facilitated the efficient writing, execution, and optimization of SQL queries.

The dataset encompasses multiple tables containing information on Products, Inventory, Stores, Sales, and Calendar dates, providing a rich foundation for thorough analysis and interpretation.

2. SUMMARY

Observations on Sales Data:

- Missing data for October to December 2023 was observed.
- Excluding October to December 2023, sales for January to September exceeded those of 2022.

2022 (Jan-Sep)	297,055 units
2023 (Jan-Sep):	408,417 units

Percentage increase in sales from 2022 to 2023: ((408,417 - 297,055) / 297,055) * 100 ≈ 37.47%

- Forecasted Sales for Oct-Dec 2023:

Upon forecasting sales for October to December 2023, the projected figures are as follows:

2022 (Jan-Dec):	420,845 units
2023 (Jan-Dec):	546,457 units

Percentage growth in sales from2022 to 2023: ((546,457 - 420,845) / 420,845) * 100 ≈ 29.80%

- Monthly Sales Growth in 2023:

Based on the projected data, monthly sales in 2023 outpaced those of 2022, with growth rates varying between 27% and 50%.

Product categories:

According to the project data, the product categories comprise toys, arts and crafts, electronics, games, and sports and outdoors.

- In 2022, the top-selling product category, yielding the highest profit, was toys, with 141,345 units sold.
- In 2023, the top-selling product category, generating the most profit, was arts and crafts, with 203,062 units sold.

Sales by store location:

- Downtown topped sales in 2022, driven by the robust performance of toys as their primary product category.
- Retaining its lead in 2023, the Downtown location witnessed a shift in sales dominance to arts and crafts as their top-selling category.

Store location analysis:

Downtown:

- Boasting the highest count of stores, sales, and profit among all locations.
- Nonetheless, the profit percentage in Downtown is relatively lower.

Airport:

- Despite having the fewest stores, both sales and profit are superior.
- The Airport location exhibits the highest profit percentage.

CITY	STORES
Ciudad de Mexico	FOUR
Hermosillo	THREE
Toluca	TWO
Villahermosa	ONE

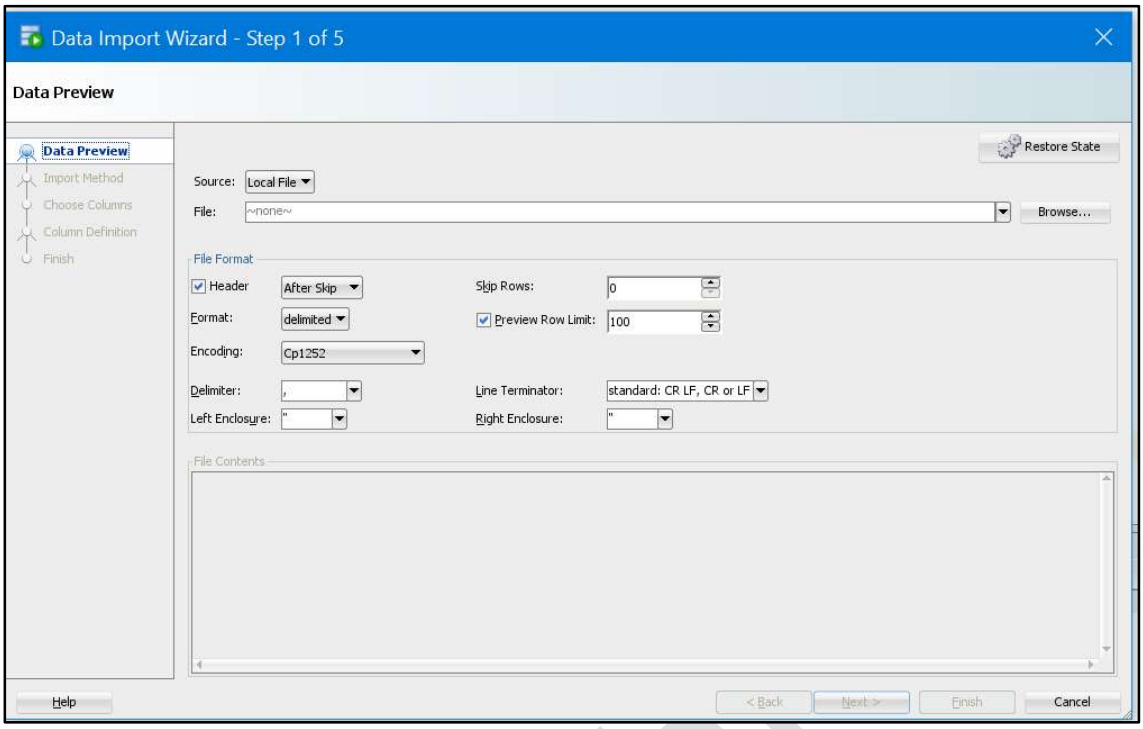
Forecast value

Upon forecasting sales for October to December 2023, the total projected sales for the entire year are as follows:

- Total sales in 2022 amounted to 420,845 units.
- Total sales in 2023 are forecasted to reach 546,457 units.

3. ANALYSIS WITH SQL

Initially, all dataset files were imported into Oracle SQL Developer for analysis purpose.



Data cleaning:

During the process, I conducted a thorough check for null values across all tables, including the Sales table. No null values were detected in any of the tables.

```
SELECT * FROM A_CALENDAR
WHERE DATE_ IS NULL;
```

```
SELECT * FROM A_INVENTORY
WHERE STORE_ID IS NULL OR PRODUCT_ID IS NULL OR STOCK_ON_HAND IS NULL;
```

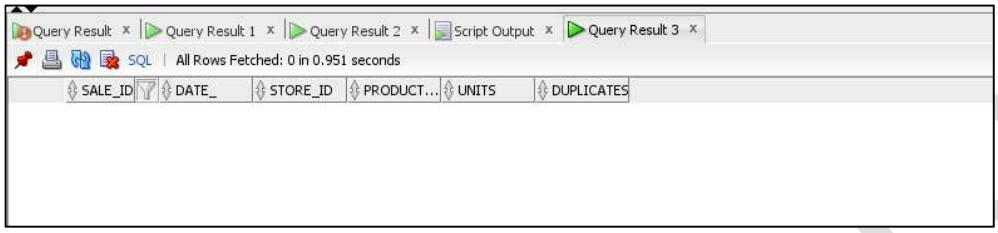
```
SELECT * FROM A_PRODUCTS
WHERE PRODUCT_ID IS NULL OR product_name IS NULL OR product_cost IS NULL OR
product_category IS NULL OR product_price IS NULL;
```

```
SELECT * FROM A_SALES
WHERE sale_id IS NULL OR date_ IS NULL OR product_id IS NULL OR store_id IS NULL OR
units IS NULL;
```

☆ I performed a comprehensive examination for duplicate values across all tables to uphold data integrity and validate that each transaction is uniquely recorded without any duplications.

Example:

```
SELECT  sale_id, date_, store_id, product_id, units, COUNT(*) AS DUPLICATES FROM a_sales
GROUP BY sale_id, date_, store_id, product_id, units
HAVING COUNT(*)>1;
```



☆ To maintain referential integrity between the Sales and Stores tables, I executed a query to identify any records in the Sales table where the Store_ID does not correspond to an existing entry in the Stores table. This process aids in detecting any discrepancies or inconsistencies in the data that may contravene the established relationships between these tables.

```
-- Validating referential integrity

SELECT * FROM A_SALES
WHERE STORE_ID NOT IN (SELECT  STORE_ID FROM A_STORE);
```

☆ Post-Import Data Verification:

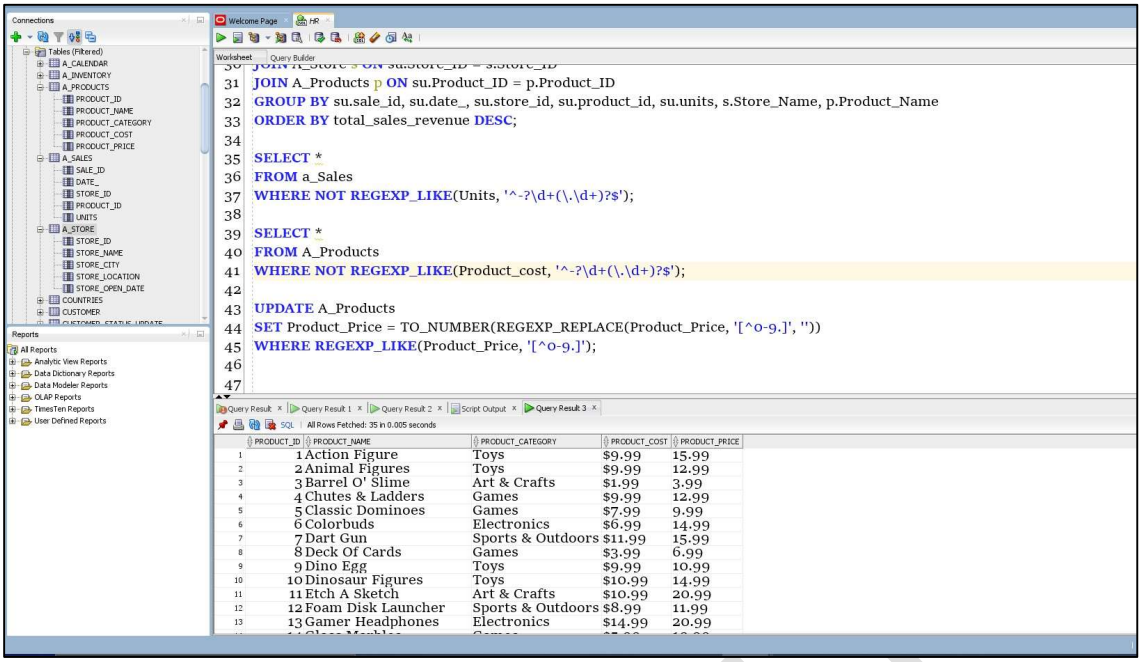
Upon importing the data, I encountered an error linked to the presence of dollar signs ('$'). This indicates potential issues with either the data import process or the data format itself. To ensure the correctness of the imported data, I will conduct a thorough review and address any problems associated with the '$' signs.

Check using :

```
SELECT *
FROM a_Sales
WHERE NOT REGEXP_LIKE(Units, '^-?\d+(\.\d+)?$');
```

Update the value :

```
UPDATE A_Products
SET Product_cost = TO_NUMBER (REGEXP_REPLACE (Product_cost, '[^0-9.]', ''))
WHERE REGEXP_LIKE(Product_cost, '[^0-9.]');
```



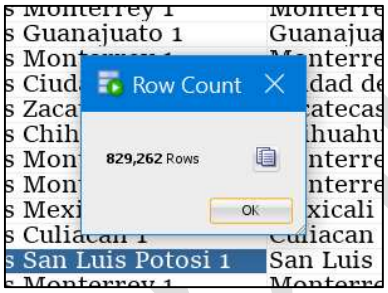
Combining tables for enhanced analysis:

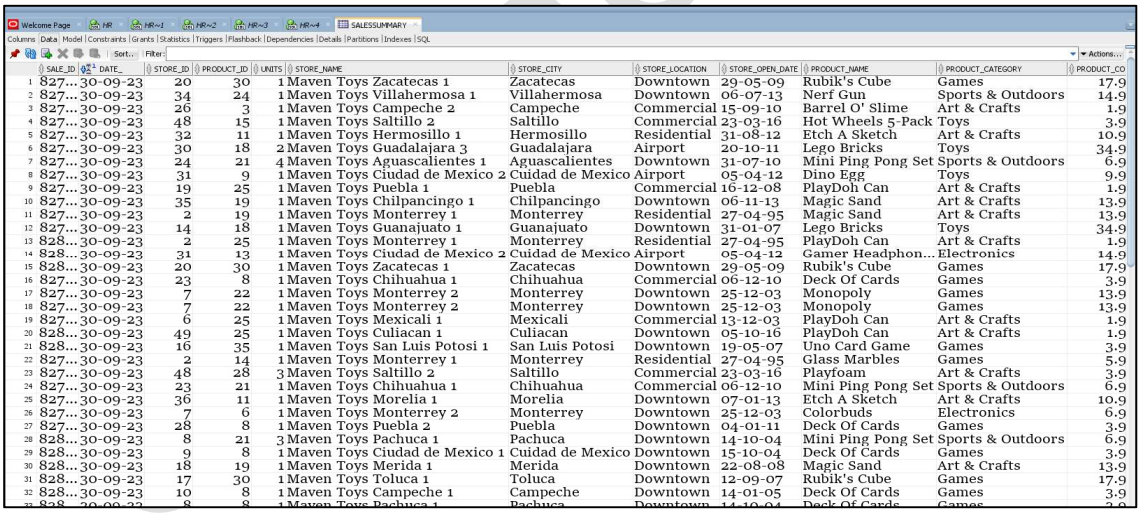
☆ I integrated three tables - Products, Sales, and Stores - by joining them based on a common column present in each table. This consolidation enables the establishment of relationships between related data, thereby ensuring data integrity and consistency. By merging these tables into one, it facilitates simplified queries, enhances efficiency, enables comprehensive analysis, and promotes better data organization.

```
CREATE TABLE SalesSummary ( sale_id INT,
date_ DATE, Store_ID INT, Product_ID INT, Units INT,
Store_Name VARCHAR (255),
store_city VARCHAR (255),
store_location VARCHAR (255),
store_open_date DATE,
Product_Name VARCHAR (255),
PRODUCT_CATEGORY VARCHAR (255),
product_cost DECIMAL (10, 2),
product_price DECIMAL (10, 2),
total_sales_revenue DECIMAL (10, 2));

INSERT INTO SalesSummary (sale_id, date_, Store_ID, Product_ID, Units, Store_Name, store_city,
store_location, store_open_date,
Product_Name, PRODUCT_CATEGORY, product_cost, product_price, total_sales_revenue)
SELECT su.sale_id, su.date_, su.Store_ID, su.Product_ID, su.Units, s.Store_Name, s.store_city,
s.store_location, s.store_open_date,
p.Product_Name, p.PRODUCT_CATEGORY, p.product_cost, p.product_price, SUM(su.Units *
```

p.Product_Price) **AS** total_sales_revenue
FROM a_Sales su
JOIN A_Store s ON su.Store_ID = s.Store_ID
JOIN A_Products p ON su.Product_ID = p.Product_ID
GROUP BY su.sale_id, su.date_, su.Store_ID, su.Product_ID, su.Units, s.Store_Name, s.store_city, s.store_location, s.store_open_date,
p.Product_Name, p.PRODUCT_CATEGORY, p.product_cost, p.product_price;

After joining the tables, it's essential to verify the rows of the resulting table to ensure they match the original data. In Oracle SQL Developer, you can conveniently count rows by right-clicking on any value and selecting the "Count Rows" option. This provides a quick means to obtain the total number of rows in a table or the number of rows that meet specific criteria. This verification step ensures the accuracy and completeness of the combined dataset.



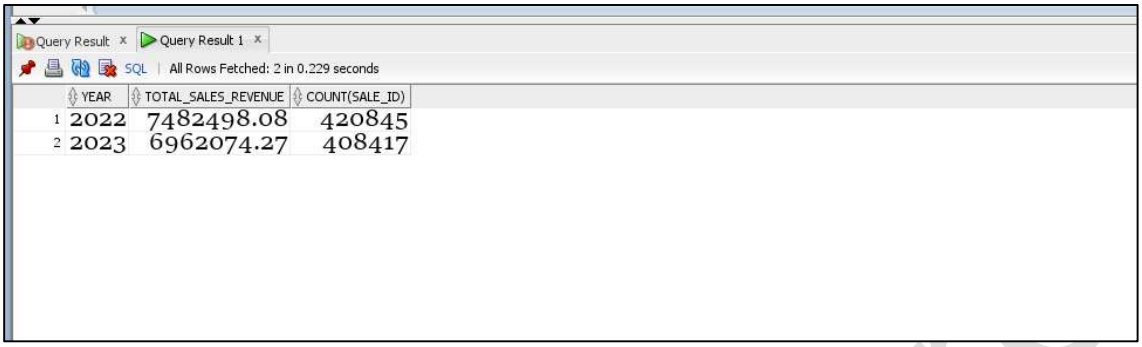


	SALE_ID	DATE_	STORE_ID	PRODUCT_ID	UNITS	STORE_NAME	STORE_CITY	STORE_LOCATION	STORE_OPEN_DATE	PRODUCT_NAME	PRODUCT_CATEGORY	PRODUCT_CO
1	827...	30-09-23	20	30	1	Maven Toys Zacatecas 1	Zacatecas	Downtown	29-05-09	Rubik's Cube	Games	17.9
2	827...	30-09-23	34	24	1	Maven Toys Villahermosa 1	Villahermosa	Downtown	06-07-13	Nerf Gun	Sports & Outdoors	14.9
3	827...	30-09-23	26	3	1	Maven Toys Campeche 2	Campeche	Commercial	15-09-10	Barrel O' Slime	Art & Crafts	1.9
4	827...	30-09-23	48	15	1	Maven Toys Saltillo 2	Saltillo	Commercial	23-03-16	Hot Wheels 5-Pack	Toys	3.9
5	827...	30-09-23	32	11	1	Maven Toys Hermosillo 1	Hermosillo	Residential	31-08-12	Etch A Sketch	Art & Crafts	10.9
6	827...	30-09-23	30	18	2	Maven Toys Guadalajara 3	Guadalajara	Airport	20-10-11	Lego Bricks	Toys	34.9
7	827...	30-09-23	24	21	4	Maven Toys Aguascalientes 1	Aguascalientes	Downtown	31-07-10	Mini Ping Pong Set	Sports & Outdoors	6.9
8	827...	30-09-23	31	9	1	Maven Toys Ciudad de Mexico 2	Cuidad de Mexico	Airport	05-04-12	Dino Egg	Toys	9.9
9	827...	30-09-23	19	25	1	Maven Toys Puebla 1	Puebla	Commercial	16-12-08	PlayDoh Can	Art & Crafts	1.9
10	827...	30-09-23	35	19	1	Maven Toys Chilpancingo 1	Chilpancingo	Downtown	06-11-13	Magic Sand	Art & Crafts	13.9
11	827...	30-09-23	2	19	1	Maven Toys Monterrey 1	Monterrey	Residential	27-04-95	Magic Sand	Art & Crafts	13.9
12	827...	30-09-23	14	18	1	Maven Toys Guanajuato 1	Guanajuato	Downtown	31-01-07	Lego Bricks	Toys	34.9
13	828...	30-09-23	2	25	1	Maven Toys Monterrey 1	Monterrey	Residential	27-04-95	PlayDoh Can	Art & Crafts	1.9
14	828...	30-09-23	31	13	1	Maven Toys Ciudad de Mexico 2	Cuidad de Mexico	Airport	05-04-12	Gamer Headphon...	Electronics	14.9
15	828...	30-09-23	20	30	1	Maven Toys Zacatecas 1	Zacatecas	Downtown	29-05-09	Rubik's Cube	Games	17.9
16	827...	30-09-23	23	8	1	Maven Toys Chihuahua 1	Chihuahua	Commercial	06-12-10	Deck Of Cards	Games	3.9
17	827...	30-09-23	7	22	1	Maven Toys Monterrey 2	Monterrey	Downtown	25-12-03	Monopoly	Games	13.9
18	827...	30-09-23	7	22	1	Maven Toys Monterrey 2	Monterrey	Downtown	25-12-03	Monopoly	Games	13.9
19	827...	30-09-23	6	25	1	Maven Toys Mexicali 1	Mexicali	Commercial	13-12-03	PlayDoh Can	Art & Crafts	1.9
20	828...	30-09-23	49	25	1	Maven Toys Culiacan 1	Culiacan	Downtown	05-10-16	PlayDoh Can	Art & Crafts	1.9
21	828...	30-09-23	16	35	1	Maven Toys San Luis Potosi 1	San Luis Potosi	Downtown	19-05-07	Uno Card Game	Games	3.9
22	827...	30-09-23	2	14	1	Maven Toys Monterrey 1	Monterrey	Residential	27-04-95	Glass Marbles	Games	5.9
23	827...	30-09-23	48	28	3	Maven Toys Saltillo 2	Saltillo	Commercial	23-03-16	Playfoam	Art & Crafts	3.9
24	827...	30-09-23	23	21	1	Maven Toys Chihuahua 1	Chihuahua	Commercial	06-12-10	Mini Ping Pong Set	Sports & Outdoors	6.9
25	827...	30-09-23	36	11	1	Maven Toys Morelia 1	Morelia	Downtown	07-01-13	Etch A Sketch	Art & Crafts	10.9
26	827...	30-09-23	7	6	1	Maven Toys Monterrey 2	Monterrey	Downtown	25-12-03	Colorbuds	Electronics	6.9
27	827...	30-09-23	28	8	1	Maven Toys Puebla 2	Puebla	Downtown	04-01-11	Deck Of Cards	Games	3.9
28	828...	30-09-23	8	21	3	Maven Toys Pachuca 1	Pachuca	Downtown	14-10-04	Mini Ping Pong Set	Sports & Outdoors	6.9
29	828...	30-09-23	9	8	1	Maven Toys Ciudad de Mexico 1	Cuidad de Mexico	Downtown	15-10-04	Deck Of Cards	Games	3.9
30	828...	30-09-23	18	19	1	Maven Toys Merida 1	Merida	Downtown	22-08-08	Magic Sand	Art & Crafts	13.9
31	828...	30-09-23	17	30	1	Maven Toys Toluca 1	Toluca	Downtown	12-09-07	Rubik's Cube	Games	17.9
32	828...	30-09-23	10	8	1	Maven Toys Campeche 1	Campeche	Downtown	14-01-05	Deck Of Cards	Games	3.9

In many cases, the initial step involves calculating the total sales for specified time periods or categories. This foundational analysis provides essential insights into overall performance and helps guide subsequent analyses and decision-making processes.

SELECT EXTRACT(**YEAR FROM** date_) **AS** YEAR,
SUM(TOTAL_SALES_REVENUE) **AS** TOTAL_SALES_REVENUE,
COUNT(SALE_ID)

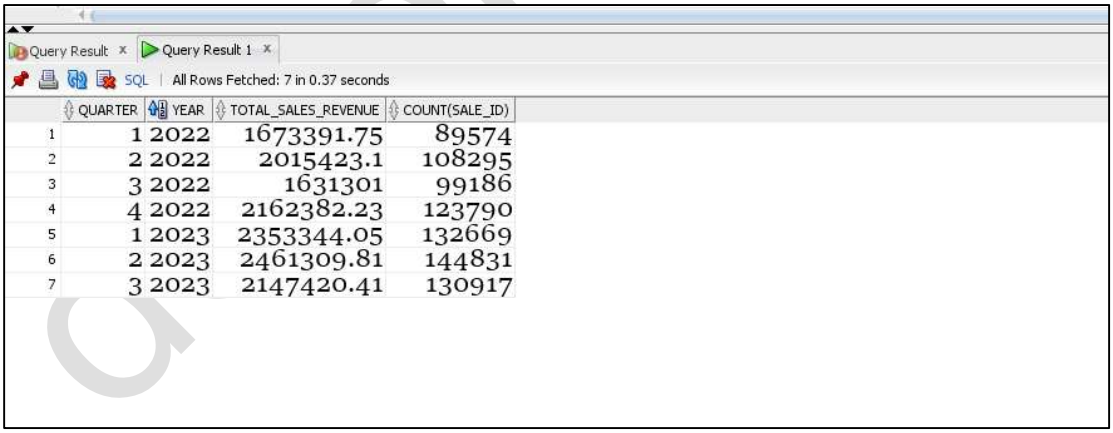
```
FROM SalesSummary
GROUP BY EXTRACT(YEAR FROM date_)
ORDER BY EXTRACT(YEAR FROM date_);
```



	YEAR	TOTAL_SALES_REVENUE	COUNT(SALE_ID)
1	2022	7482498.08	420845
2	2023	6962074.27	408417

Based on the total sales generated per year, I aimed to investigate the reasons behind 2022 outperforming 2023. Subsequently, I delved into quarterly sales data to gain a deeper understanding of the trends and factors influencing performance over specific time periods.

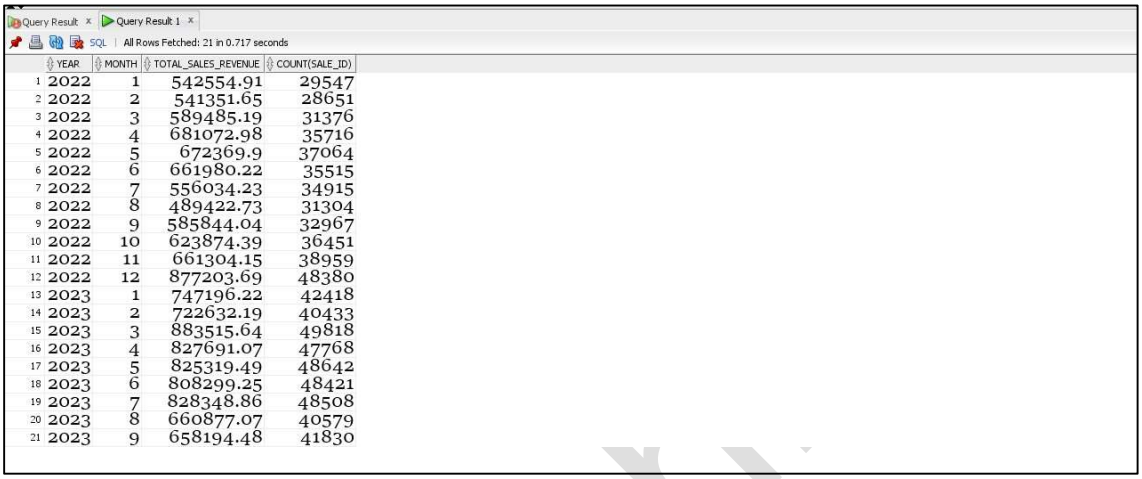
```
SELECT CEIL(EXTRACT(MONTH FROM date_) / 3) AS QUARTER,
EXTRACT(YEAR FROM date_) AS YEAR,SUM(TOTAL_SALES_REVENUE) AS
TOTAL_SALES_REVENUE,
COUNT(SALE_ID)
FROM SalesSummary
GROUP BY EXTRACT(YEAR FROM date_), CEIL(EXTRACT(MONTH FROM date_) / 3)
ORDER BY YEAR;
```



	QUARTER	YEAR	TOTAL_SALES_REVENUE	COUNT(SALE_ID)
1	1	2022	1673391.75	89574
2	2	2022	2015423.1	108295
3	3	2022	1631301	99186
4	4	2022	2162382.23	123790
5	1	2023	2353344.05	132669
6	2	2023	2461309.81	144831
7	3	2023	2147420.41	130917

In the quarterly sales analysis, only three quarters of 2023 were included. Upon reviewing the dataset, it became apparent that some months were missing data for 2022, resulting in an incomplete comparison with the sales data of 2023. Furthermore, based on the analysis of total sales generated per month, it was noted that records were available for only 9 months in 2023. These observations highlight the need for further data validation and potentially rectifying missing or incomplete data to ensure accurate and comprehensive analysis.

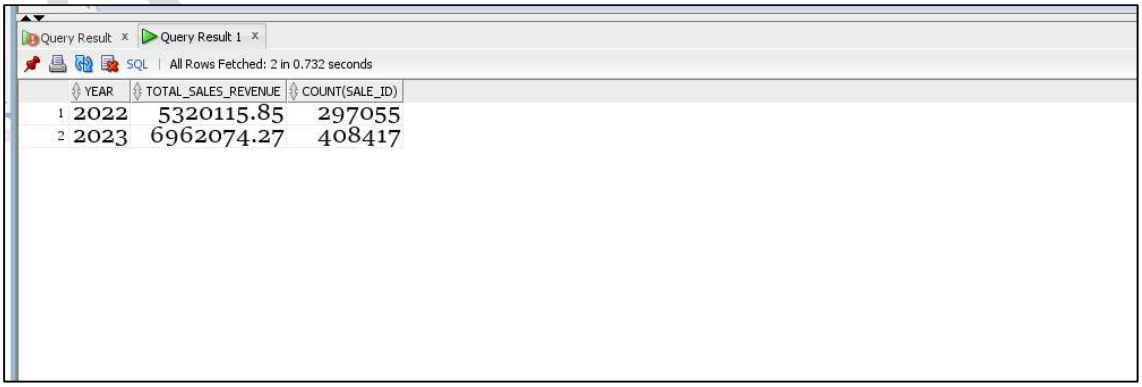
```
SELECT EXTRACT(YEAR FROM date_) AS YEAR,
EXTRACT(MONTH FROM date_) AS MONTH,
SUM(TOTAL_SALES_REVENUE) AS TOTAL_SALES_REVENUE,
COUNT(SALE_ID)
FROM SalesSummary
GROUP BY EXTRACT(YEAR FROM date_), EXTRACT(MONTH FROM date_)
ORDER BY EXTRACT(YEAR FROM date_), EXTRACT(MONTH FROM date_);
```



	YEAR	MONTH	TOTAL_SALES_REVENUE	COUNT(SALE_ID)
1	2022	1	542554.91	29547
2	2022	2	541351.65	28651
3	2022	3	589485.19	31376
4	2022	4	681072.98	35716
5	2022	5	672369.9	37064
6	2022	6	661980.22	35515
7	2022	7	556034.23	34915
8	2022	8	489422.73	31304
9	2022	9	585844.04	32967
10	2022	10	623874.39	36451
11	2022	11	661304.15	38959
12	2022	12	877203.69	48380
13	2023	1	747196.22	42418
14	2023	2	722632.19	40433
15	2023	3	883515.64	49818
16	2023	4	827691.07	47768
17	2023	5	825319.49	48642
18	2023	6	808299.25	48421
19	2023	7	828348.86	48508
20	2023	8	660877.07	40579
21	2023	9	658194.48	41830

Given this information, I re-evaluated the total sales generated per year to ensure accuracy and completeness in light of the observed data discrepancies and missing records

```
SELECT EXTRACT(YEAR FROM date_) AS YEAR,
SUM(TOTAL_SALES_REVENUE) AS TOTAL_SALES_REVENUE,
COUNT(SALE_ID)
FROM SalesSummary
WHERE EXTRACT(MONTH FROM date_) NOT IN (10,11, 12) -- Exclude January, February, and March
GROUP BY EXTRACT(YEAR FROM date_)
ORDER BY EXTRACT(YEAR FROM date_);
```



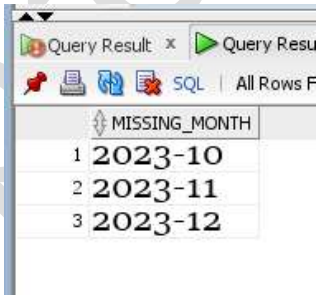
	YEAR	TOTAL_SALES_REVENUE	COUNT(SALE_ID)
1	2022	5320115.85	297055
2	2023	6962074.27	408417

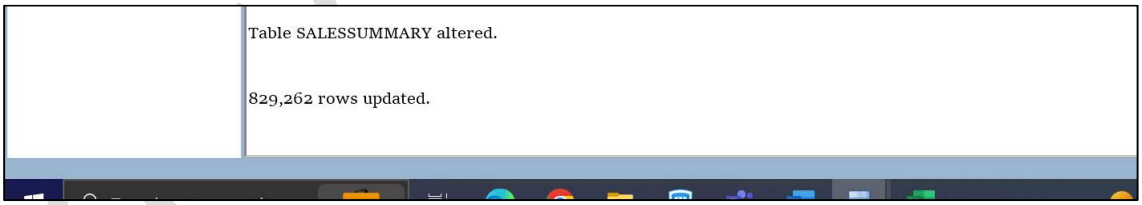
Based on this information, even after excluding the missing data for October to December, the

total sales for 2023 still surpass those of 2022. In 2023, sales reached a count of 408,417, compared to 2022's 297,055. This represents a sales growth of approximately 37.49% in 2023 compared to 2022.

I utilized Common Table Expressions (CTE) to re-examine the sales data and identify the months that are missing from the sales data in 2023.

```
WITH Months AS (
SELECT  ADD_MONTHS (TO_DATE('01-01-2023', 'DD-MM-YYYY'), LEVEL - 1) AS Month
FROM dual
CONNECT BY ADD_MONTHS(TO_DATE('01-01-2023', 'DD-MM-YYYY'), LEVEL - 1) <=
TO_DATE('31-12-2023', 'DD-MM-YYYY')
)
SELECT  TO_CHAR(Month, 'YYYY-MM') AS Missing_Month
FROM Months
WHERE NOT EXISTS (
SELECT  1
FROM SalesSummary
WHERE EXTRACT(YEAR FROM date_) = 2023
AND EXTRACT(MONTH FROM date_) = EXTRACT (MONTH FROM Month)
)
ORDER BY Month;
ALTER TABLE SalesSummary
ADD PROFIT FLOAT;
UPDATE SalesSummary
SET PROFIT = (PRODUCT_PRICE - PRODUCT_COST);
```





Analyze the profit per product category by year

```
SELECT  Product_Category, SUM (Units) AS Total_Units, COUNT (Sale_ID) AS
Sale_per_category,
SUM (Profit) AS Profit, EXTRACT(YEAR FROM date_) AS YEAR
FROM SalesSummary
GROUP BY Product_Category, EXTRACT(YEAR FROM date_)
ORDER BY EXTRACT (YEAR FROM date_);
```

Query Result | SQL | All Rows Fetched: 10 in 0.406 seconds

	PRODUCT_CATEGORY	TOTAL_UNITS	SALE_PER_CATEGORY	PROFIT	YEAR
1	Art & Crafts	122512	86954	211190	2022
2	Electronics	88481	62052	467300	2022
3	Games	108441	87400	301484	2022
4	Sports & Outdoors	88713	66573	206273	2022
5	Toys	141345	117866	509581	2022
6	Art & Crafts	203062	133719	347012	2023
7	Electronics	45594	36973	262332	2023
8	Games	86232	69606	235753	2023
9	Sports & Outdoors	80330	64758	211789	2023
10	Toys	125855	103361	395054	2023

In 2022:

- Toys had the highest sales with 141,345 units sold.
- Electronics had the lowest sales with 88,481 units sold.

In 2023:

- Toys maintained the highest sales with 125,855 units sold.
- Arts and Crafts had the lowest sales with 122,512 units sold.

```
SELECT  EXTRACT (YEAR FROM date_) AS YEAR,
store_location, COUNT (Sale_id) AS total_sales, Product_category
FROM salessummary
GROUP BY EXTRACT (YEAR FROM date_),store_location, Product_category
ORDER BY total_sales DESC, Product_category DESC;
```

Script Output | Query Result | SQL | All Rows Fetched: 40 in 0.251 seconds

	YEAR	STORE_LOCATION	TOTAL_SALES	PRODUCT_CATEGORY
1	2023	Downtown	80945	Art & Crafts
2	2022	Downtown	68934	Toys
3	2023	Downtown	61280	Toys
4	2022	Downtown	51529	Art & Crafts
5	2022	Downtown	49500	Games
6	2023	Downtown	39408	Games
7	2022	Downtown	38674	Sports & Outdoors
8	2023	Downtown	37613	Sports & Outdoors
9	2022	Downtown	32914	Electronics
10	2023	Commercial	27170	Art & Crafts
11	2022	Commercial	24761	Toys
12	2023	Commercial	21692	Toys
13	2022	Commercial	20026	Games
14	2023	Downtown	19427	Electronics
15	2022	Commercial	18544	Art & Crafts
16	2022	Commercial	17101	Electronics
17	2023	Commercial	15851	Games
18	2023	Residential	15808	Art & Crafts
19	2022	Residential	15627	Toys
20	2022	Commercial	15077	Sports & Outdoors
21	2023	Commercial	14371	Sports & Outdoors
22	2023	Residential	11706	Toys
23	2022	Residential	10140	Art & Crafts
24	2023	Commercial	10112	Electronics
25	2022	Residential	9804	Games

After analyzing the sales data, I proceeded to investigate the distribution of store IDs across various regions.

```
SELECT  STORE_LOCATION, COUNT(DISTINCT STORE_ID) AS
NUMBER_OF_STORE_ASSIGNED FROM A_STOR
GROUP BY STORE_LOCATION
```

Query Result

SQL | All Rows Fetched: 4 in 0.008 seconds

	STORE_LOCATION	NUMBER_OF_STORE_ASSIGNED
1	Residential	6
2	Commercial	12
3	Downtown	29
4	Airport	3

I conducted an examination of the distribution of stores across different regions, including Downtown, Commercial, Residential, and Airport.

```
SELECT
Store_City, Downtown, Commercial, Residential, Airport, (Downtown + Commercial + Residential + Airport) AS Total
FROM (
SELECT  Store_City, Store_Location
FROM SalesSummary)
PIVOT (
COUNT(Store_Location)
FOR Store_Location IN ('Downtown' AS Downtown, 'Commercial' AS Commercial, 'Residential' AS Residential, 'Airport' AS Airport))
ORDER BY Store_City;
```

Query Result

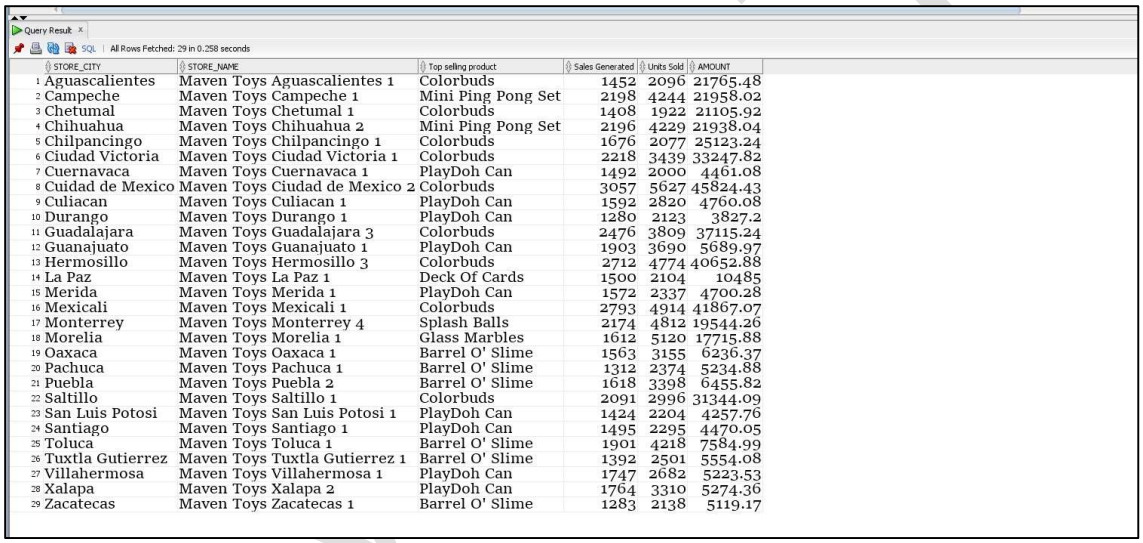
SQL | All Rows Fetched: 29 in 0.257 seconds

	STORE_CITY	DOWNTOWN	COMMERCIAL	RESIDENTIAL	AIRPORT	TOTAL
1	La Paz	13217	0	0	0	13217
2	Zacatecas	13501	0	0	0	13501
3	Cuernavaca	13643	0	0	0	13643
4	Oaxaca	13957	0	0	0	13957
5	Durango	14110	0	0	0	14110
6	Aguascalientes	14588	0	0	0	14588
7	Chilpancingo	14592	0	0	0	14592
8	Culiacan	14594	0	0	0	14594
9	Tuxtla Gutierrez	14618	0	0	0	14618
10	Chetumal	14644	0	0	0	14644
11	Merida	14875	0	0	0	14875
12	Morelia	14956	0	0	0	14956
13	Pachuca	14969	0	0	0	14969
14	San Luis Potosi	15499	0	0	0	15499
15	Ciudad Victoria	16034	0	0	0	16034
16	Santiago	16111	0	0	0	16111
17	Villahermosa	16324	0	0	0	16324
18	Campeche	17695	12805	0	0	30500
19	Chihuahua	16580	13998	0	0	30578
20	Saltillo	18924	14166	0	0	33090
21	Xalapa	18809	14998	0	0	33807
22	Mexicali	16991	16864	0	0	33855
23	Toluca	23533	12776	0	0	36309
24	Puebla	16764	15776	14868	0	47408
25	Guanajuato	18157	16494	14569	0	49220
26	Hermosillo	18018	16553	15264	0	49835
27	Monterrey	21300	16276	15698	16049	69323
28	Guadalajara	18739	16331	15926	23384	74380
29	Cuidad de Mexico	24482	17668	19551	29024	90725

The determination of the top-selling products in each store city is based on the number of units sold for each product.

```
WITH Top_selling_products AS (
SELECT
Store_City, Store_Name, Product_Name,
```

```
COUNT(Sale_ID) AS sales, SUM(Units) AS units, SUM(PRODUCT_PRICE) P,
ROW_NUMBER() OVER (PARTITION BY Store_City ORDER BY SUM(Units) DESC) AS
Products_ranked
FROM
SalesSummary
GROUP BY
Store_City, Product_Name, Store_Name
)
SELECT
Store_City, Store_Name,
Product_Name AS "Top selling product", sales AS "Sales Generated",
units AS "Units Sold", P AS Amount
FROM
Top_selling_products
WHERE
Products_ranked = 1;
```

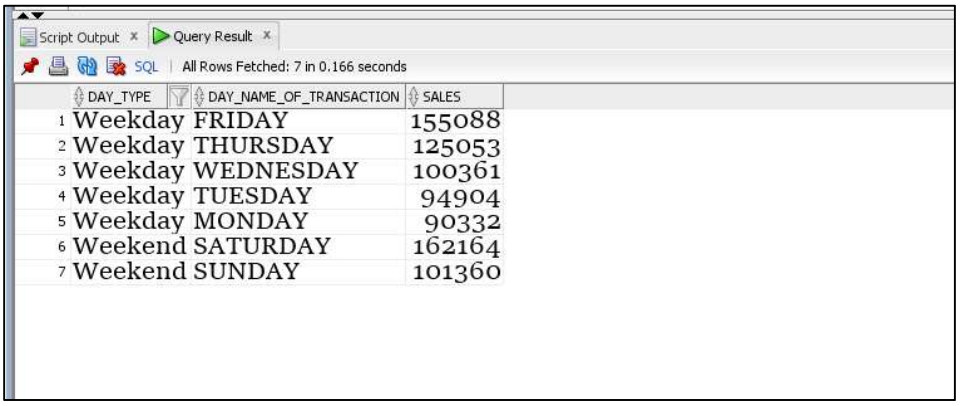


Query Result | All Rows Fetched: 29 in 0.258 seconds

	STORE_CITY	STORE_NAME	Top selling product	Sales Generated	Units Sold	AMOUNT
1	Aguascalientes	Maven Toys Aguascalientes 1	Colorbuds	1452	2096	21765.48
2	Campeche	Maven Toys Campeche 1	Mini Ping Pong Set	2198	4244	21958.02
3	Chetumal	Maven Toys Chetumal 1	Colorbuds	1408	1922	21105.92
4	Chihuahua	Maven Toys Chihuahua 2	Mini Ping Pong Set	2196	4229	21938.04
5	Chilpancingo	Maven Toys Chilpancingo 1	Colorbuds	1676	2077	25123.24
6	Ciudad Victoria	Maven Toys Ciudad Victoria 1	Colorbuds	2218	3439	33247.82
7	Cuernavaca	Maven Toys Cuernavaca 1	PlayDoh Can	1492	2000	4461.08
8	Cuidad de Mexico	Maven Toys Ciudad de Mexico 2	Colorbuds	3057	5627	45824.43
9	Culiacan	Maven Toys Culiacan 1	PlayDoh Can	1592	2820	4760.08
10	Durango	Maven Toys Durango 1	PlayDoh Can	1280	2123	3827.2
11	Guadalajara	Maven Toys Guadalajara 3	Colorbuds	2476	3809	37115.24
12	Guanajuato	Maven Toys Guanajuato 1	PlayDoh Can	1903	3690	5689.97
13	Hermosillo	Maven Toys Hermosillo 3	Colorbuds	2712	4774	40652.88
14	La Paz	Maven Toys La Paz 1	Deck Of Cards	1500	2104	10485
15	Merida	Maven Toys Merida 1	PlayDoh Can	1572	2337	4700.28
16	Mexicali	Maven Toys Mexicali 1	Colorbuds	2793	4914	41867.07
17	Monterrey	Maven Toys Monterrey 4	Splash Balls	2174	4812	19544.26
18	Morelia	Maven Toys Morelia 1	Glass Marbles	1612	5120	17715.88
19	Oaxaca	Maven Toys Oaxaca 1	Barrel O' Slime	1563	3155	6236.37
20	Pachuca	Maven Toys Pachuca 1	Barrel O' Slime	1312	2374	5234.88
21	Puebla	Maven Toys Puebla 2	Barrel O' Slime	1618	3398	6455.82
22	Saltillo	Maven Toys Saltillo 1	Colorbuds	2091	2996	31344.09
23	San Luis Potosi	Maven Toys San Luis Potosi 1	PlayDoh Can	1424	2204	4257.76
24	Santiago	Maven Toys Santiago 1	PlayDoh Can	1495	2295	4470.05
25	Toluca	Maven Toys Toluca 1	Barrel O' Slime	1901	4218	7584.99
26	Tuxtla Gutierrez	Maven Toys Tuxtla Gutierrez 1	Barrel O' Slime	1392	2501	5554.08
27	Villahermosa	Maven Toys Villahermosa 1	PlayDoh Can	1747	2682	5223.53
28	Xalapa	Maven Toys Xalapa 2	PlayDoh Can	1764	3310	5274.36
29	Zacatecas	Maven Toys Zacatecas 1	Barrel O' Slime	1283	2138	5119.17

Following that, I proceeded to analyze the sales data to distinguish between weekday and weekend sales.

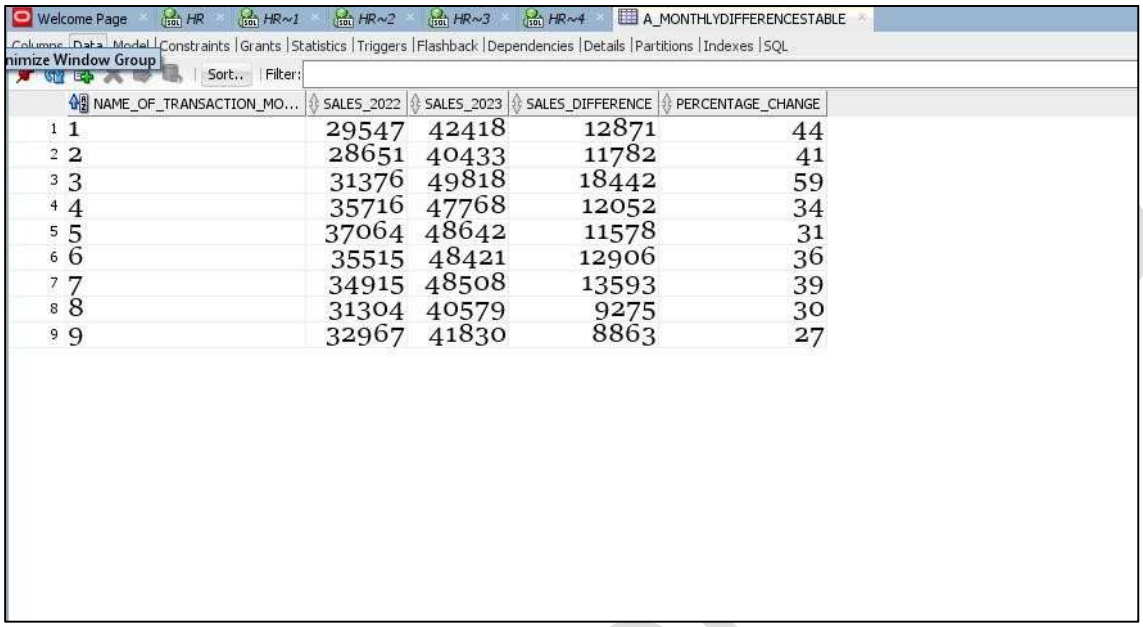
```
WITH new_name AS (
SELECT
CASE WHEN Day_Name_Of_Transaction IN ('Saturday', 'Sunday') THEN 'Weekend'
ELSE 'Weekday'
END AS Day_Type, Day_Name_Of_Transaction,  COUNT(Sale_ID) AS Sales
FROM SalesSummary
GROUP BY Day_Name_Of_Transaction)
SELECT  * FROM new_name
ORDER BY Day_Type, Sales DESC;
```



Predicted sales and percentage change for October, November, and December 2023 are as follows:

```
CREATE TABLE A_MonthlyDifferencesTable ( Name_Of_Transaction_Month VARCHAR(20),
Sales_2022 INT,
Sales_2023 INT,
Sales_Difference INT, Percentage_Change FLOAT);

INSERT INTO A_MonthlyDifferencesTable (Name_Of_Transaction_Month, Sales_2022, Sales_2023,
Sales_Difference, Percentage_Change)
WITH MonthlySales AS (
SELECT
EXTRACT(MONTH FROM Date_) AS Name_Of_Transaction_Month,
EXTRACT(YEAR FROM Date_) AS Year_Of_Transaction,
COUNT(Sale_ID) AS Sales
FROM SalesSummary
WHERE (EXTRACT(YEAR FROM Date_) = 2022 AND EXTRACT(MONTH FROM Date_) <= 12) OR
(EXTRACT(YEAR FROM Date_) = 2023 AND EXTRACT(MONTH FROM Date_) <= 12)
GROUP BY EXTRACT(MONTH FROM Date_), EXTRACT(YEAR FROM Date_)
),
MonthlyDifferences AS (
SELECT
M1.Name_Of_Transaction_Month, M1.Sales AS Sales_2022,
M2.Sales AS Sales_2023,
M2.Sales - M1.Sales AS Sales_Difference,
ROUND((M2.Sales - M1.Sales) / CAST(M1.Sales AS FLOAT) * 100, 0) AS Percentage_Change
FROM MonthlySales M1
JOIN MonthlySales M2 ON M1.Name_Of_Transaction_Month = M2.Name_Of_Transaction_Month
AND M1.Year_Of_Transaction = 2022
AND M2.Year_Of_Transaction = 2023
)
SELECT * FROM MonthlyDifferences;
```

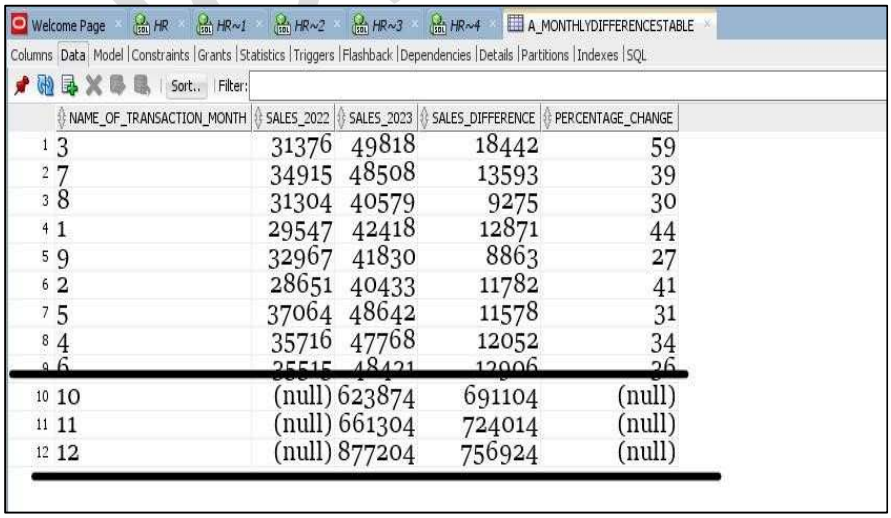
```
Table A_MONTHLYDIFFERENCESTABLE created.

9 rows inserted.
```



	NAME_OF_TRANSACTION_MO...	SALES_2022	SALES_2023	SALES_DIFFERENCE	PERCENTAGE_CHANGE
1	1	29547	42418	12871	44
2	2	28651	40433	11782	41
3	3	31376	49818	18442	59
4	4	35716	47768	12052	34
5	5	37064	48642	11578	31
6	6	35515	48421	12906	36
7	7	34915	48508	13593	39
8	8	31304	40579	9275	30
9	9	32967	41830	8863	27

```
-- Insert forecasted sales for October, November, and December 2023 into the MonthlyDifferences table
INSERT INTO A_MonthlyDifferencesTable (Name_Of_Transaction_Month, Sales_2022, Sales_2023)
WITH MonthlySales AS (
-- Calculate total sales revenue for each month in 2022 and 2023
SELECT
EXTRACT(MONTH FROM Date_) AS Month,
EXTRACT(YEAR FROM Date_) AS Year,
COUNT(CASE WHEN EXTRACT(YEAR FROM Date_) = 2022 THEN Sale_ID END) AS
Sales_2022,
COUNT(CASE WHEN EXTRACT(YEAR FROM Date_) = 2023 THEN Sale_ID END) AS
Sales_2023
FROM SalesSummary
WHERE EXTRACT(YEAR FROM Date_) IN (2022, 2023)
GROUP BY EXTRACT(YEAR FROM Date_), EXTRACT(MONTH FROM Date_)
),
Forecast AS (
-- Generate sales forecasts for October, November, and December 2023
SELECT
10          AS Name_Of_Transaction_Month, (SELECT  COUNT(Sale_ID)
FROM SalesSummary
WHERE EXTRACT(YEAR FROM Date_) = 2022 AND EXTRACT(MONTH FROM Date_) = 10 )
```

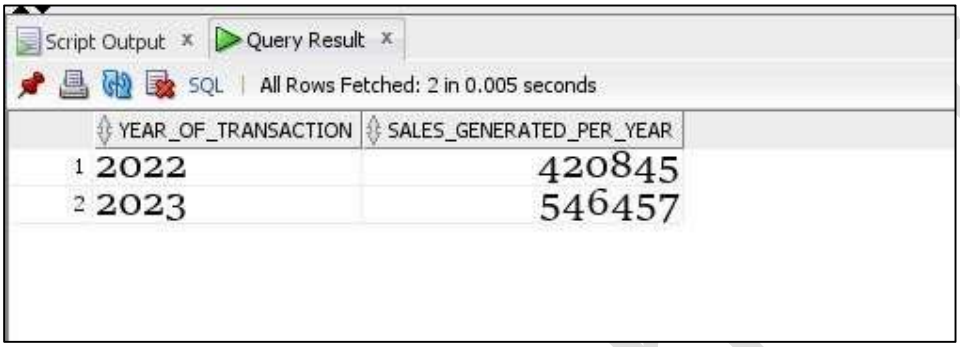
```
AS Sales_2022,
(Sales_2023 * 1.05) AS Sales_2023,-- Example: increase sales by 5% for simplicity
FROM MonthlySales
WHERE Year = 2023 AND Month = 9
UNION ALL SELECT
11 AS Name_Of_Transaction_Month, (SELECT COUNT(Sale_ID)
FROM SalesSummary
WHERE EXTRACT(YEAR FROM Date_) = 2022 AND EXTRACT(MONTH FROM Date_) = 11 ) AS
Sales_2022,
(Sales_2023 * 1.1) AS Sales_2023 -- Example: Increase sales by 10% for simplicity
FROM MonthlySales
WHERE Year = 2023 AND Month = 9
UNION ALL SELECT
12 AS Name_Of_Transaction_Month, (SELECT COUNT(Sale_ID)
FROM SalesSummary
WHERE EXTRACT(YEAR FROM Date_) = 2022 AND EXTRACT(MONTH FROM Date_) = 12 )
AS Sales_2022,
(Sales_2023 * 1.15) AS Sales_2023 -- Example: Increase sales by 15% for simplicity
FROM MonthlySales
WHERE Year = 2023 AND Month = 9
)
SELECT * FROM Forecast;
```



	NAME_OF_TRANSACTION_MONTH	SALES_2022	SALES_2023	SALES_DIFFERENCE	PERCENTAGE_CHANGE
1	3	31376	49818	18442	59
2	7	34915	48508	13593	39
3	8	31304	40579	9275	30
4	1	29547	42418	12871	44
5	9	32967	41830	8863	27
6	2	28651	40433	11782	41
7	5	37064	48642	11578	31
8	4	35716	47768	12052	34
9	6	35515	48421	12906	36
10	10	(null)	623874	691104	(null)
11	11	(null)	661304	724014	(null)
12	12	(null)	877204	756924	(null)

Total annual sales, incorporating forecasted figures for October through December 2023.

```
SELECT  '2022' AS Year_Of_Transaction, SUM (Sales_2022) AS Sales_Generated_per_year
FROM A_MonthlyDifferencesTable
UNION ALL
SELECT  '2023' AS Year_Of_Transaction, SUM (Sales_2023) AS Sales_Generated_per_year
FROM A_MonthlyDifferencesTable;
```



4. CONCLUCSION:

This SQL project on Maven Toys sales data demonstrated proficiency in data analysis using Oracle SQL Developer. Beginning with data cleaning and validation, the analysis merged multiple tables to unveil insights into sales trends, product performance, and store dynamics. Despite missing data for October to December 2023, total sales for 2023 exceeded those of 2022, indicating growth. Arts and crafts emerged as the top-selling category in 2023, contrasting with toys in 2022. Store location analysis revealed Downtown's sales leadership despite lower profit margins. Common Table Expressions identified missing months in 2023 sales data, while analysis differentiated weekday and weekend sales patterns. Predicted sales for October to December 2023 were also calculated. Overall, the project underscored SQL's efficacy in deriving actionable insights for informed decision-making.