

MILESTONE 2

Next Gen Retail Analytics Dashboard

Group: Team_04_02

Aarushi Kotwani

Vatsal Dalal

Tileshwar Narayan (Group Leader)

Mission Statement

At New Gen Retail Analytics, our mission is to revolutionize the retail sector through the strategic use of comprehensive data analytics. Our commitment is to deliver a state-of-the-art dashboard seamlessly integrating key variables – Product, Customer, Supplier, Sales, Employee, Rating, Stores and Profitability. We envision a dynamic transformation of raw data into actionable insights, empowering our clients to make informed decisions, optimize operations, and elevate customer experiences.

Our mission extends beyond data integration, it is a pledge to redefine the retail analytics landscape. We aspire to be the driving force behind adaptability, ensuring our clients not only keep pace with industry shifts but lead them. By streamlining operations through data-driven precision, our dashboard becomes a strategic guide, fostering operational excellence.

Customer experiences are paramount in our mission. We facilitate a deep understanding of customer preferences, behavior, and satisfaction, enabling personalized interactions. Sustainability is ingrained in our approach, contributing to the creation of enduring growth models. In summary, our mission is to empower businesses with confidence, agility, and forward-thinking insights, propelling them toward sustainable success in the data-driven era.

Summary of Challenges

In the dynamic realm of the retail sector, the imperative of data-driven decision-making is integral to daily operations. Some prominent challenge facing businesses centers on the continual and instantaneous synchronization of data across diverse variables. The revolutionary New Gen retail analytics dashboard addresses this complex issue by transforming how businesses engage with and harness their data.

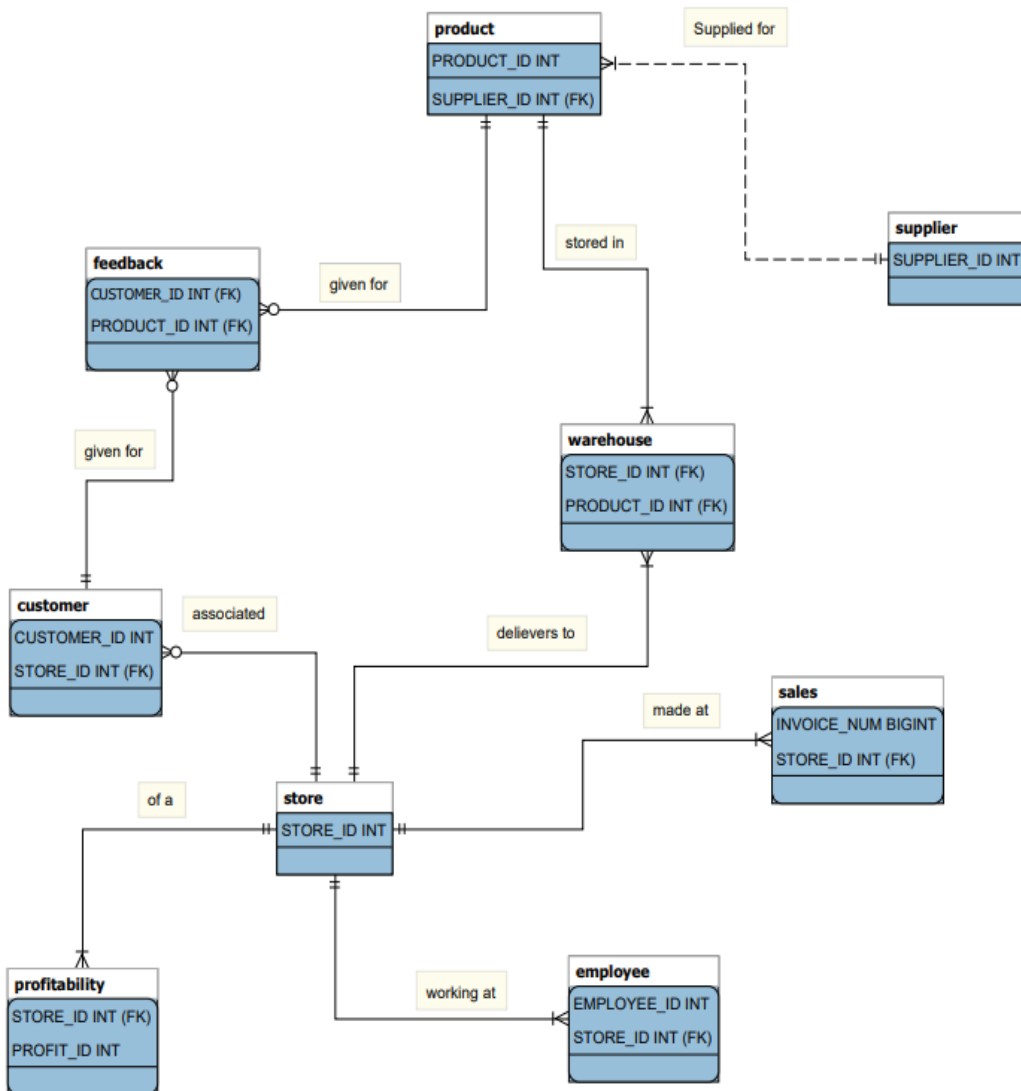
At the heart of this challenge lies the need for regular updates and real-time connectivity with critical data fields. In the conventional retail analytics landscape, data uploads and updates often occur at intervals that may not align with the rapid pace of change in the retail environment. This time lag in data refreshments presents a significant obstacle for businesses striving to maintain a competitive edge.

Moreover, the dashboard addresses the challenge of comprehensively connecting data with each field. In the absence of an integrated solution, businesses often encounter disjointed data, resulting in a fragmented view of operations. For instance, disconnected data might hinder the ability to correlate customer behavior with product preferences, impeding targeted marketing efforts. The New Gen retail analytics dashboard alleviates these issues by establishing a robust framework that interconnects data fields. This interconnectedness empowers businesses to draw correlations between variables, unlocking insights that were previously obscured.

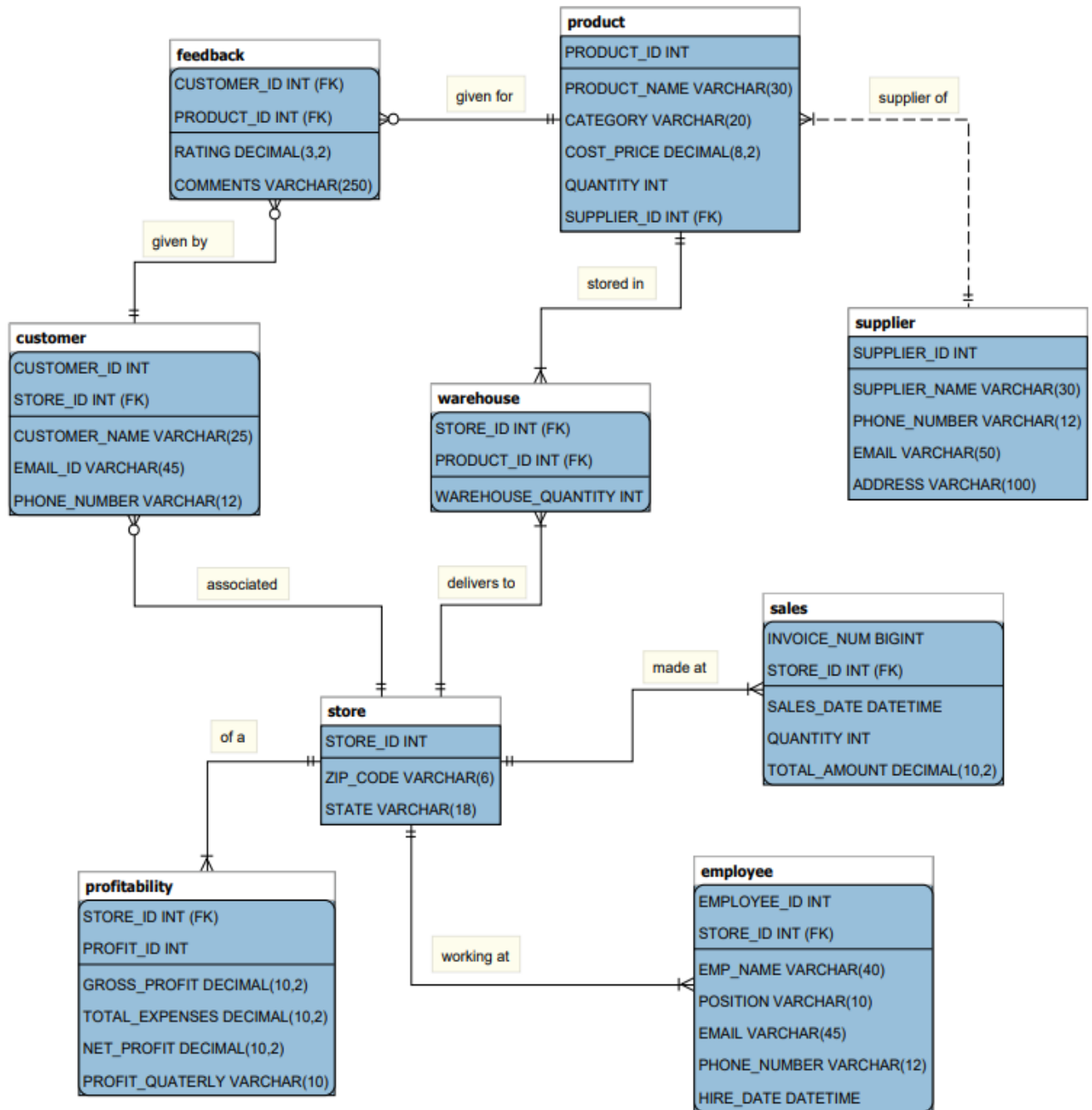
The importance of real-time data feeds becomes even more apparent in the context of customer satisfaction and engagement. In an era where customer expectations are rapidly evolving, businesses must be attuned to real-time feedback and sentiments. Delayed or outdated information in customer satisfaction metrics can lead to misguided strategies and missed opportunities to enhance the overall customer experience. The New Gen retail analytics dashboard addresses this critical aspect by ensuring that customer satisfaction data is continuously updated, enabling businesses to respond promptly to customer feedback and preferences.

In essence, the New Gen retail analytics dashboard not only recognizes the challenges posed by traditional data analytics methods but also presents a transformative solution. By providing a platform that facilitates real-time data updates, seamless connectivity across variables, and a holistic view of operations, the dashboard empowers businesses to navigate the complexities of the retail landscape with agility, precision, and an enhanced ability to meet evolving customer expectations. In doing so, it propels retail enterprises into a new era of data-driven decision-making, where timely insights pave the way for sustained growth and competitiveness.

ER Diagram



Logical Diagram



Queries

1.

Relevance:

The query provides visibility into workforce distribution across stores by presenting employee counts alongside store identifiers and location details. This information is instrumental in understanding staffing patterns and identifying stores with the highest employee presence, offering valuable insights into store management and resource allocation decisions in retail operations.

Query:

```
SELECT
    s.STORE_ID,
    s.ZIP_CODE,
    s.STATE,
    COUNT(*) AS EMPLOYEE_COUNT
FROM
    employee e
JOIN
    store s ON e.STORE_ID = s.STORE_ID
GROUP BY
    s.STORE_ID, s.ZIP_CODE, s.STATE
ORDER BY
    EMPLOYEE_COUNT DESC;
```

Output:

Result Grid					Filter Rows:	Export:	Wrap Cell Content:
STORE_ID	ZIP_CODE	STATE	EMPLOYEE_COUNT				
1228	98018	WA	53				
1225	98015	WA	53				
1216	98006	WA	53				
1221	98011	WA	53				
1223	98013	WA	51				
1219	98009	WA	51				
1230	98020	WA	50				
1211	98001	WA	50				
1226	98016	WA	50				
1224	98014	WA	50				
1222	98012	WA	50				
1217	98007	WA	50				

Result 24 x

Output

Action Output

#	Time	Action	Message	Duration / Fets
35	13:05:55	SELECT	st.STORE_ID, st.ZIP_CODE, st.STATE, DATE_FORMAT(s..., 200 row(s) returned	0.000 sec / 0.

2.

Relevance:

The query fetches supplier information alongside the products they provide, establishing connections between supplier IDs in the 'supplier' table and corresponding product entries. By incorporating supplier addresses, it enriches procurement analysis, offering valuable insights into supplier distribution and optimizing product sourcing strategies for retail analytics.

Query:

```
SELECT s.SUPPLIER_ID, s.SUPPLIER_NAME, s.ADDRESS AS SUPPLIER_ADDRESS,
p.PRODUCT_NAME, p.PRODUCT_ID
FROM supplier s
JOIN product p ON s.SUPPLIER_ID = p.SUPPLIER_ID;
```

Output:

Result Grid					
Filter Rows:		Export:	Wrap Cell Content:	Fetch rows:	
SUPPLIER_ID	SUPPLIER_NAME	SUPPLIER_ADDRESS	PRODUCT_NAME	PRODUCT_ID	
1111	Quantum Innovations Inc.	356 Burning Wood Road	Fitness Watch	10029	
1111	Quantum Innovations Inc.	356 Burning Wood Road	Body Scrub	10092	
1111	Quantum Innovations Inc.	356 Burning Wood Road	Body Mist	10117	
1111	Quantum Innovations Inc.	356 Burning Wood Road	Art Supplies Kit	10188	
1112	StellarTech Solutions	089 Jenna Crossing	Drone	10003	
1112	StellarTech Solutions	089 Jenna Crossing	Sunscreen	10086	
1112	StellarTech Solutions	089 Jenna Crossing	Biography	10120	
1113	Apex Dynamics Group	9 Autumn Leaf Road	E-reader	10026	
1113	Apex Dynamics Group	9 Autumn Leaf Road	Green Tea Lemonade	10072	
1113	Apex Dynamics Group	9 Autumn Leaf Road	Toilet Paper	10107	
1113	Apex Dynamics Group	9 Autumn Leaf Road	Play Set	10175	
1114	Fusion Forge Enterprises	99 Luster Point	VR Headset	10008	

Result 26 x

Output

Action Output

#	Time	Action	Message	Duration / Fetc
37	13:26:32	SELECT s.SUPPLIER_ID, s.SUPPLIER_NAME, s.ADDRESS AS SUPPLIER_A...	200 row(s) returned	0.015 sec / 0.

3.

Relevance:

The SQL query extracts vital product details such as ID, name, and category, alongside store IDs and ZIP codes. By incorporating store location information, it enriches the retail analytics dashboard, providing valuable insights into product distribution across various store locations. This data aids in optimizing inventory management and refining product placement strategies, ultimately enhancing sales performance analysis.

Query:

SELECT

p.PRODUCT_ID,

p.PRODUCT_NAME,

p.CATEGORY,

w.STORE_ID,

s.ZIP_CODE

FROM

product p

JOIN

warehouse w ON p.PRODUCT_ID = w.PRODUCT_ID

JOIN

store s ON w.STORE_ID = s.STORE_ID;

Output:

PRODUCT_ID	PRODUCT_NAME	CATEGORY	STORE_ID	ZIP_CODE
10004	Power Bank	ELECTRONICS	1211	98001
10005	Speaker	ELECTRONICS	1211	98001
10010	Desktop Computer	ELECTRONICS	1211	98001
10014	Smart TV	ELECTRONICS	1211	98001
10017	Wireless Charger	ELECTRONICS	1211	98001
10025	Wireless Mouse	ELECTRONICS	1211	98001
10026	E-reader	ELECTRONICS	1211	98001
10027	MP3 Player	ELECTRONICS	1211	98001
10028	Gaming Keyboard	ELECTRONICS	1211	98001
10031	Portable Hard Drive	ELECTRONICS	1211	98001
10038	Streaming Device	ELECTRONICS	1211	98001
10042	Soda	BEVERAGES	1211	98001

Result 28 ×

Output

Action Output

#	Time	Action	Message
39	13:29:41	SELECT p.PRODUCT_ID, p.PRODUCT_NAME, p.CATEGORY, w....	200 row(s) returned

4.

Relevance:

The query aggregates key financial indicators like total sales, gross profit, expenses, and net profit per store, providing a overall view of financial performance. By incorporating store identifiers, ZIP codes, and states, it offers geographical insights that can inform strategic decisions and enable performance benchmarking within the realm of retail analytics.

Query:

SELECT

st.STORE_ID,

st.ZIP_CODE,

st.STATE,

SUM(s.TOTAL_AMOUNT) AS TOTAL_SALES_AMOUNT,

SUM(p.GROSS_PROFIT) AS TOTAL_GROSS_PROFIT,

SUM(p.TOTAL_EXPENSES) AS TOTAL_EXPENSES,

SUM(p.NET_PROFIT) AS TOTAL_NET_PROFIT

FROM

sales s

JOIN

store st ON s.STORE_ID = st.STORE_ID

JOIN

profitability p ON p.STORE_ID = st.STORE_ID

GROUP BY

st.STORE_ID, st.ZIP_CODE, st.STATE;

Output:

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	STORE_ID	ZIP_CODE	STATE	TOTAL_SALES_AMOUNT	TOTAL_GROSS_PROFIT	TOTAL_EXPENSES	TOTAL_NET_PROFIT
▶	1211	98001	WA	297045.00	72500.00	232000.00	58000.00
	1212	98002	WA	283185.00	91250.00	292000.00	73000.00
	1213	98003	WA	257670.00	76250.00	244000.00	61000.00
	1214	98004	WA	632507.04	658788.00	328000.00	527030.00
	1215	98005	WA	4204383.68	1104516.00	56000.00	883612.00
	1216	98006	WA	1288792.32	2235816.00	396000.00	1788652.00
	1217	98007	WA	2726766.00	3467100.00	312000.00	2773680.00
	1218	98008	WA	253890.00	116250.00	372000.00	93000.00
	1219	98009	WA	3982227.76	4496958.00	228000.00	3597566.00
	1220	98010	WA	3958994.52	5754872.00	304000.00	4603898.00
	1221	98011	WA	1017145.80	1144202.00	292000.00	915362.00
	1222	98012	WA	3265946.64	2166912.00	128000.00	1733530.00

Result 30

Output

Action Output

#	Time	Action	Message
41	13:32:16	SELECT st.STORE_ID, st.ZIP_CODE, st.STATE, SUM(s.TOTAL_A...	20 row(s) returned

5.

Relevance:

The query retrieves the store IDs, ZIP codes, and states along with the highest net profit and average gross profit for each store, facilitating analysis of store performance based on profitability metrics. It offers insights into the financial health of stores across different geographic locations in the retail analytics dashboard.

Query:

```
SELECT
    s.STORE_ID,
    s.ZIP_CODE,
    s.STATE,
    MAX(p.NET_PROFIT) AS HIGHEST_NET_PROFIT,
    AVG(p.GROSS_PROFIT) AS AVERAGE_GROSS_PROFIT
FROM
    profitability p
JOIN
    store s ON p.STORE_ID = s.STORE_ID
GROUP BY
    s.STORE_ID, s.ZIP_CODE, s.STATE
ORDER BY
    HIGHEST_NET_PROFIT DESC;
```

Output:

STORE_ID	ZIP_CODE	STATE	HIGHEST_NET_PROFIT	AVERAGE_GROSS_PROFIT
1226	98016	WA	36702.96	45878.700000
1220	98010	WA	23019.49	28774.360000
1219	98009	WA	17987.83	22484.790000
1223	98013	WA	17672.26	22090.320000
1224	98014	WA	14797.80	18497.250000
1217	98007	WA	13868.40	17335.500000
1225	98015	WA	10563.70	13204.620000
1216	98006	WA	8943.26	11179.080000
1222	98012	WA	8667.65	10834.560000
1230	98020	WA	5417.28	6771.600000
1228	98018	WA	5149.15	6436.440000
1221	98011	WA	4576.81	5721.010000

#	Time	Action	Message	Duration / Fetch
43	13:35:07	SELECT s.STORE_ID, s.ZIP_CODE, s.STATE, MAX(p.NET_PROFI...	20 row(s) returned	0.000 sec / 0.000 sec

6.

Relevance:

The query provides insights into the distribution of products across different categories in the warehouse. By calculating the total quantity and total cost of products for each category, it facilitates inventory management and analysis for retail operations. This information aids in optimizing stock levels, identifying high-demand categories, and evaluating the cost-effectiveness of storing various product types.

Query:

SELECT

p.CATEGORY,

SUM(w.WAREHOUSE_QUANTITY) AS TOTAL_QUANTITY,

SUM(w.WAREHOUSE_QUANTITY * p.COST_PRICE) AS
TOTAL_COST_OF_MATERIAL_IN_WAREHOUSE

FROM

product p

JOIN

warehouse w ON p.PRODUCT_ID = w.PRODUCT_ID

GROUP BY p.CATEGORY

ORDER BY TOTAL_QUANTITY DESC;

Output:

CATEGORY	TOTAL_QUANTITY	TOTAL_COST_OF_MATERIAL_IN_WAREHOUSE
BOOKS & MEDIA	110192	32963685.68
ELECTRONICS	106510	27370606.37
KIDS	106096	28136339.06
BEVERAGES	101764	24315718.92
HYGIENE	91326	21207872.34

result 34	x		
Output			
Action Output			
#	Time	Action	Message
45	13:38:09	SELECT p.CATEGORY, SUM(w.WAREHOUSE_QUANTITY) AS TOTAL_...	5 row(s) returned

7.

Relevance:

The SQL query retrieves information about the most senior employees in each store based on their hire dates. By identifying employees with the earliest hire dates within each store, the query assists in recognizing long-standing staff members and understanding the workforce's tenure distribution across different store locations. This information is valuable for assessing employee retention, recognizing experienced personnel, and identifying potential candidates for leadership roles within the organization.

Query:

```
SELECT
  e.STORE_ID,
  e.EMPLOYEE_ID,
  e.EMP_NAME,
  e.POSITION,
  e.EMAIL,
  e.PHONE_NUMBER,
  e.HIRE_DATE
FROM
  employee e
JOIN
  (
    SELECT
      STORE_ID,
      MIN(HIRE_DATE) AS SENIOR_DATE
    FROM
      employee
    GROUP BY
      STORE_ID
  ) AS t ON e.STORE_ID = t.STORE_ID AND e.HIRE_DATE = t.SENIOR_DATE;
```

Output:

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	STORE_ID	EMPLOYEE_ID	EMP_NAME	POSITION	EMAIL	PHONE_NUMBER	HIRE_DATE
▶	1211	501296	Clotilda Skupinski	Assistant	cskupinski14@digg.com	815-191-3149	2000-07-03 00:00:00
	1227	501320	Colleen Adamini	Assistant	cadamini1s@acquirethisname.com	500-478-7280	2000-05-04 00:00:00
	1216	501352	Violette Skilling	Assistant	vskillinggm@networkadvertising.org	590-876-8430	2000-05-24 00:00:00
	1222	501395	Harmonia Bloxsome	Assistant	hbloxsomeht@scribd.com	920-490-9960	2000-04-07 00:00:00
	1218	501403	Alvira Collar	Manager	acollar1@java.com	527-192-2318	2000-07-24 00:00:00
	1226	501406	Mellisa Gorner	Janitor	mgorner46@nyu.edu	512-671-6659	2000-04-01 00:00:00
	1212	501428	Bertrando Sherrington	Janitor	bsherrington4s@bluehost.com	647-777-3568	2000-02-20 00:00:00
	1217	501502	Dode Pockett	Janitor	dpocketts@yahoo.co.jp	694-394-3953	2000-03-24 00:00:00
	1223	501506	Nari Hatchette	Assistant	nhatchettekw@woothemes.com	770-197-9511	2001-04-20 00:00:00
	1228	501506	Noreen Yanshin	Janitor	nyanshin6y@storify.com	658-640-3697	2000-03-22 00:00:00
	1229	501510	Ottile Petrushkevich	Assistant	opetrushkevichl0@netscape.com	832-597-7699	2000-03-07 00:00:00
	1224	501516	Hermine Winkell	Assistant	hwinkell6@360.cn	853-733-6691	2000-12-13 00:00:00

Result 36

×

Output

⌵

Action Output

⌵

#	Time	Action	Message
✓ 47	13:40:47	SELECT e.STORE_ID, e.EMPLOYEE_ID, e.EMP_NAME, e.POSITI...	20 row(s) returned

8.

Relevance

The query retrieves product details along with their average ratings and the number of feedback entries, facilitating the identification of popular products based on customer feedback. By sorting the results by average rating and the number of feedback entries, retail analytics can gain insights into product performance and customer satisfaction, aiding in product optimization and marketing strategies.

Query:

```
SELECT
    p.PRODUCT_ID,
    p.PRODUCT_NAME,
    p.CATEGORY,
    AVG(f.RATING) AS AVERAGE_RATING,
    COUNT(f.RATING) AS NUM_FEEDBACK
FROM
    product p
LEFT JOIN
    feedback f ON p.PRODUCT_ID = f.PRODUCT_ID
GROUP BY
    p.PRODUCT_ID, p.PRODUCT_NAME, p.CATEGORY
ORDER BY
    AVERAGE_RATING DESC, NUM_FEEDBACK DESC;
```

Output:

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

	PRODUCT_ID	PRODUCT_NAME	CATEGORY	AVERAGE_RATING	NUM_FEEDBACK
▶	10034	Portable Projector	ELECTRONICS	4.415000	2
	10187	Construction Set	KIDS	4.352500	4
	10190	Outdoor Playset	KIDS	4.320000	5
	10157	Toy Car	KIDS	4.256667	3
	10099	Hand Cream	HYGIENE	4.203333	6
	10062	Orange Juice	BEVERAGES	4.103333	3
	10159	Puzzle Set	KIDS	4.044286	7
	10081	Shampoo	HYGIENE	3.987500	4
	10030	Portable Speaker	ELECTRONICS	3.984000	5
	10140	Thriller Novel	BOOKS & MEDIA	3.930000	3
	10094	Antiperspirant	HYGIENE	3.830000	3
	10043	Cold Brew Coffee	BEVERAGES	3.798333	6

Result 39

Output

Action Output

#	Time	Action	Message
50	13:42:51	SELECT p.PRODUCT_ID, p.PRODUCT_NAME, p.CATEGORY, AV...	200 row(s) returned

9.

Relevance:

The query offers a comprehensive analysis of product performance within the retail analytics dashboard. By combining feedback count, average rating, and total quantity available for each product, retailers gain valuable insights into customer satisfaction, product popularity, and inventory management. This information enables data-driven decision-making to optimize product offerings, enhance customer experience, and maximize sales potential.

Query:

```
SELECT
    p.PRODUCT_ID,
    p.PRODUCT_NAME,
    p.CATEGORY,
    COUNT(f.PRODUCT_ID) AS FEEDBACK_COUNT,
    AVG(f.RATING) AS AVERAGE_RATING,
    SUM(p.QUANTITY) AS TOTAL_QUANTITY_AVAILABLE
FROM
    product p
LEFT JOIN
    feedback f ON p.PRODUCT_ID = f.PRODUCT_ID
GROUP BY
    p.PRODUCT_ID, p.PRODUCT_NAME, p.CATEGORY
ORDER BY
    FEEDBACK_COUNT DESC;
```


Output:

Result Grid						
Filter Rows:		Export:		Wrap Cell Content:		Fetch rows:
PRODUCT_ID	PRODUCT_NAME	CATEGORY	FEEDBACK_COUNT	AVERAGE_RATING	TOTAL_QUANTITY_AVAILABLE	
10039	Electric Toothbrush	ELECTRONICS	11	2.270909	44	
10006	Console	ELECTRONICS	10	2.903000	110	
10084	Deodorant	HYGIENE	10	3.500000	710	
10085	Facial Cleanser	HYGIENE	10	3.356000	950	
10135	Music CD	BOOKS & MEDIA	10	3.332000	300	
10163	Play Dough	KIDS	10	3.091000	630	
10166	Building Blocks	KIDS	10	3.173000	120	
10104	Razors	HYGIENE	9	2.600000	1485	
10129	Fiction Novel	BOOKS & MEDIA	9	3.198889	1062	
10049	Coconut Water	BEVERAGES	8	2.392500	1104	
10053	Lemonade	BEVERAGES	8	3.356250	184	
10055	Coffee	BEVERAGES	8	3.421250	136	

Result 42 x

Output

Action Output

#	Time	Action	Message
53	13:45:08	SELECT p.PRODUCT_ID, p.PRODUCT_NAME, p.CATEGORY, CO...	200 row(s) returned

10.

Relevance:

The query provides a comprehensive overview of sales performance for each store in each month within the specified time period. By aggregating sales data and calculating average total sales amounts, it enables stakeholders to analyze store-level sales trends and identify potential areas for improvement or optimization in the retail analytics dashboard.

Query:

```
SELECT
    st.STORE_ID,
    st.ZIP_CODE,
    st.STATE,
    DATE_FORMAT(s.SALES_DATE, '%m-%Y') AS Month,
    COUNT(*) AS TotalSalesCount,
    AVG(s.TOTAL_AMOUNT) AS AvgTotalSalesAmount
FROM
```

sales s

JOIN

store st ON s.STORE_ID = st.STORE_ID

GROUP BY

st.STORE_ID,

st.ZIP_CODE,

st.STATE,

Month

ORDER BY

st.STORE_ID,

Month;

Output:

Result Grid			Filter Rows:	Export:		Wrap Cell Content:	Fetch rows:	
	STORE_ID	ZIP_CODE	STATE	Month	TotalSalesCount	AvgTotalSalesAmount		
▶	1211	98001	WA	01-2024	4	1968.750000		
	1211	98001	WA	02-2023	2	656.250000		
	1211	98001	WA	03-2023	7	1650.000000		
	1211	98001	WA	04-2023	3	1461.250000		
	1211	98001	WA	05-2023	5	1496.250000		
	1211	98001	WA	06-2023	7	1631.250000		
	1211	98001	WA	07-2023	5	1821.750000		
	1211	98001	WA	08-2023	6	1050.000000		
	1211	98001	WA	09-2023	3	1225.000000		
	1211	98001	WA	11-2023	4	1141.875000		
	1211	98001	WA	12-2023	4	1647.187500		
	1212	98002	WA	01-2024	6	1365.000000		

Result 44 ×

Output

Action Output

#	Time	Action	Message
✓ 55	13:49:22	SELECT st.STORE_ID, st.ZIP_CODE, st.STATE, DATE_FORMAT(s....	200 row(s) returned

Stored Procedures

We have created one Stored procedures for our database. The following are the details of the same.

Procedure:

GetFeedbackByCategory

Relevance:

In a retail analytics dashboard, this query provides valuable insights by presenting the average rating and comment count for products in a specific category, enabling businesses to gauge customer satisfaction and engagement, aiding in data-driven decision-making for product improvements or marketing strategies.

Query:

```
CREATE DEFINER=`mm_team04_02`@`%` PROCEDURE `GetFeedbackByCategory`(IN
category_name VARCHAR(255))

BEGIN

    SELECT p.PRODUCT_NAME, AVG(f.RATING) AS AVERAGE_RATING, COUNT(f.COMMENTS)
AS COMMENT_COUNT

    FROM product p

    LEFT JOIN feedback f ON p.PRODUCT_ID = f.PRODUCT_ID

    WHERE p.CATEGORY = category_name

    GROUP BY p.PRODUCT_NAME;

END
```

Output:

Navigator: CHEMAS

Filter objects

- mm_team04_02
 - Tables
 - Views
 - Stored Procedures
 - GetFeedbackByCategory
 - Functions

Administration Schemas Information

No object selected

SQLStatements* SQL File 12* SQL File 13* x GetFeedbackByCategory

1 • CALL GetFeedbackByCategory('ELECTRONICS');
2 • CALL GetFeedbackByCategory('BEVERAGES');

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	PRODUCT_NAME	AVERAGE_RATING	COMMENT_COUNT
▶	Camera	3.665000	2
	Earbuds	2.920000	3
	Drone	3.330000	4
	Power Bank	3.275000	5
	Speaker	2.276667	4
	Console	2.903000	8
	Headset	2.718000	4
	VR Headset	2.442857	3
	Laptop	3.426667	6
	Desktop Computer	3.105000	3
	Gaming Mouse	2.441429	6
	Smartwatch	2.480000	3

Result 3 x

For CATEGORY - BEVARAGE

1 • CALL GetFeedbackByCategory('ELECTRONICS');
2 • CALL GetFeedbackByCategory('BEVERAGES');

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	PRODUCT_NAME	AVERAGE_RATING	COMMENT_COUNT
▶	Energy Drink	3.210000	3
	Soda	2.692500	3
	Cold Brew Coffee	3.798333	6
	Sparkling Water	2.866000	4
	Fruit Juice	3.156667	3
	Protein Shake	2.323333	1
	Herbal Tea	1.670000	1
	Sports Drink	3.726667	4
	Coconut Water	2.392500	5

Result 2 x

Output

Action Output

#	Time	Action	Message
58	14:09:14	CALL GetStoreAverageProfitForMonth(2)	20 row(s) returned