

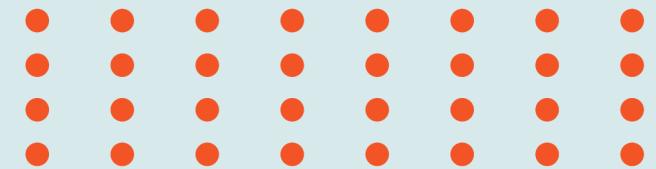
# RETAIL SALES

## REPORT

### SQL

03 April, 2025





# Introduction

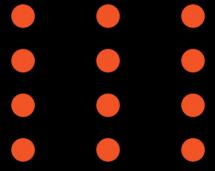


Ingoude  
Company

Welcome to our Sales Report

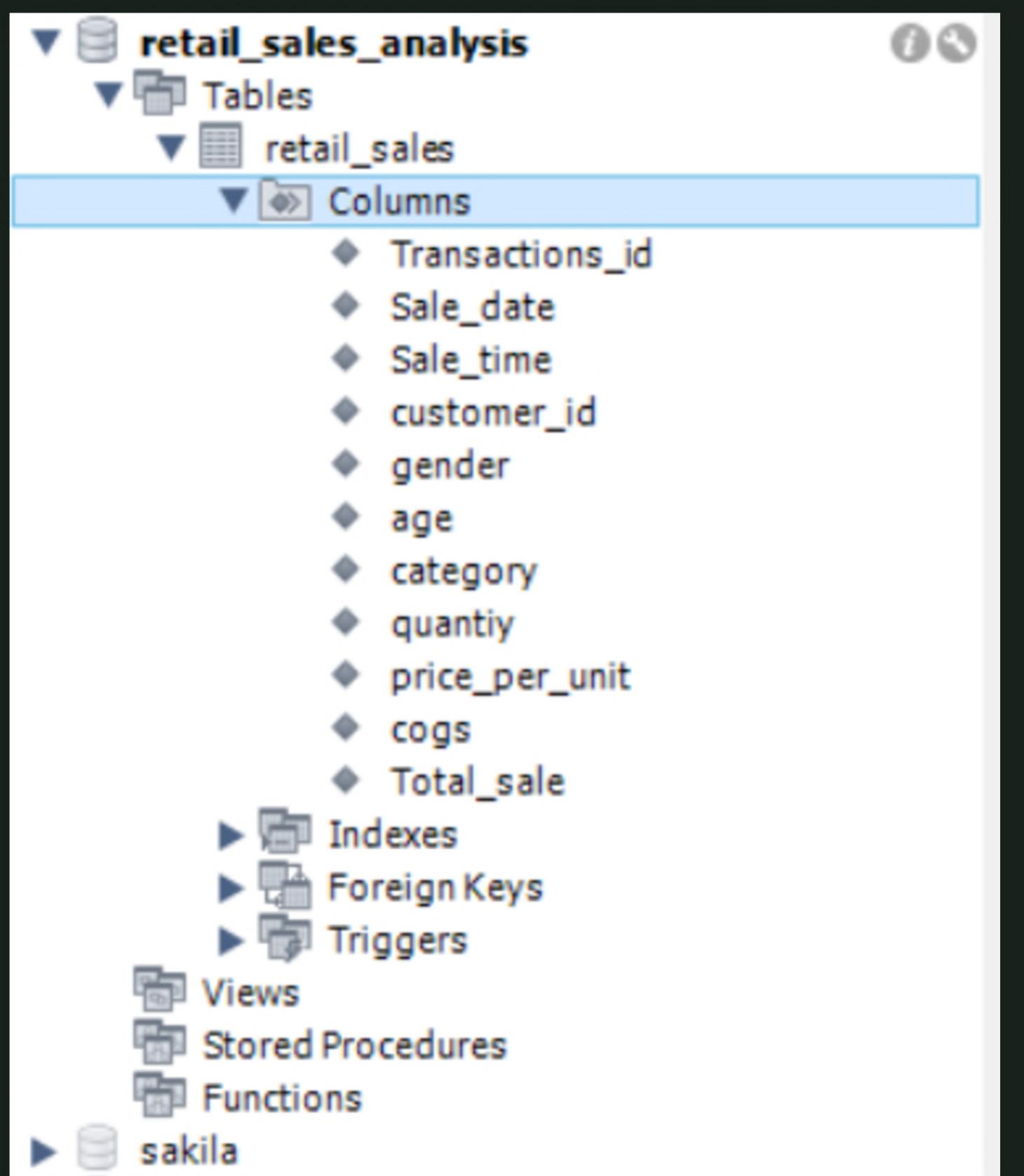
Presentation. Today, we delve into a comprehensive overview of our sales performance, exploring the highs, challenges, and strategic insights that have shaped our journey. This presentation is more than just numbers, it's a narrative of our collective efforts, showcasing the impact of our sales strategies and the pathways to future success.

# DATABASE CREATION



```
1 ● CREATE DATABASE Retail_Sales_Analysis;
2 ● USE Retail_Sales_Analysis;
3
4 ● DROP TABLE IF EXISTS Retail_Sales;
5 ● ⏺ CREATE TABLE Retail_Sales (
6     Transactions_id INT,
7     Sale_date DATE,
8     Sale_time TIME,
9     customer_id INT,
10    gender VARCHAR(15),
11    age INT,
12    category VARCHAR(15),
13    quantiy INT,
14    price_per_unit FLOAT,
15    cogs FLOAT,
16    Total_sale FLOAT,
17    PRIMARY KEY(Transactions_id)
18 );
19
20    -- DATA CLEANING
21
22 ● SELECT *
23     FROM Retail_sales
24     WHERE Transactions_id IS NULL
25     OR
26         Sale_date IS NULL
27     OR
28         Sale_time IS NULL
29     OR
30         customer_id IS NULL
31     OR
32        gender IS NULL
33        OR
34        age IS NULL
35        OR
36        category IS NULL
37        OR
38        quantiy IS NULL
39        OR
40        price_per_unit IS NULL
41        OR
42        cogs IS NULL
43        OR
44        Total_sale IS NULL;
45
46    -- DATA EXPLORATION
47    -- how many sales we have
48
49 ● SELECT COUNT(*) AS Total_sale
50     FROM retail_sales;
51
52    -- how many unique customer we have
53 ● SELECT COUNT(DISTINCT customer_id) AS Total_customer
54     FROM retail_sales;
55
56    -- how many category we have
57 ● SELECT DISTINCT category AS Total_customer
58     FROM retail_sales;
59
60
```

# SHEMAS



Write a SQL query to retrieve all columns for sales made on '2022-11-05'

```
3 •   SELECT *
4     FROM retail_sales
5    WHERE Sale_date = "2022-11-05";
```

Write a SQL query to retrieve all transactions where the category is 'Clothing'  
and the quantity sold is more than 4 in the month of Nov-2022

```
4 • SELECT *
5   FROM retail_sales
6 WHERE category = "Clothing"
7 AND
8 quantiy >= 4
9 AND
10 Sale_date BETWEEN "2022-11-01" AND "2022-11-30";
11
```

Result Grid											
	Transactions_id	Sale_date	Sale_time	customer_id	gender	age	category	quantiy	price_per_unit	cogs	Total_s
▶	64	2022-11-15	06:34:00	7	Male	49	Clothing	4	25	8.5	100
	146	2022-11-10	22:01:00	74	Male	38	Clothing	4	50	49	200
	159	2022-11-10	21:30:00	42	Male	26	Clothing	4	50	23.5	200
	284	2022-11-12	09:17:00	129	Male	43	Clothing	4	50	20.5	200
	547	2022-11-14	07:36:00	3	Male	63	Clothing	4	500	250	2000
	699	2022-11-21	22:21:00	129	Female	37	Clothing	4	30	16.2	120
	735	2022-11-26	21:38:00	153	Female	64	Clothing	4	500	515	2000
	943	2022-11-05	19:29:00	90	Female	57	Clothing	4	300	318	1200
	965	2022-11-27	21:45:00	84	Male	22	Clothing	4	50	13	200
	1259	2022-11-03	17:31:00	105	Female	45	Clothing	4	50	21	200
	1296	2022-11-26	20:42:00	45	Female	22	Clothing	4	300	342	1200
	1476	2022-11-11	22:27:00	130	Female	27	Clothing	4	500	555	2000
	1484	2022-11-23	09:29:00	22	Female	19	Clothing	4	300	147	1200
	1497	2022-11-19	21:44:00	109	Male	41	Clothing	4	30	32.4	120
	1615	2022-11-17	13:43:00	82	Female	61	Clothing	4	25	13.5	100
	1696	2022-11-21	17:59:00	24	Female	50	Clothing	4	50	55	200

Write a SQL query to calculate the total sales (total sale) for each category.

```
3 • SELECT category , SUM(Total_sale) AS Total_Sales  
4   FROM retail_sales  
5 GROUP BY category;
```

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

	category	Total_Sales
▶	Beauty	287272
	Clothing	311330
	Electronics	313810

Write a SQL query to find the average age of customers who purchased items from the 'Beauty' category.

```
3 • SELECT ROUND(AVG(age),0) AS Average  
4   FROM retail_sales  
5 WHERE category = 'Beauty';  
6
```

The screenshot shows a database interface with a query editor and a results grid. The query editor contains the SQL code provided above. The results grid shows one row with the label 'Average' and the value '40'. The interface includes standard buttons for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'.

	Average
▶	40

Write a SQL query to find all transactions where the total\_sale is greater than 1000

```
3 • SELECT *
4   FROM retail_sales
5 WHERE Total_sale > 1000
6 ORDER BY Total_sale ASC;
```

	Transactions_id	Sale_date	Sale_time	customer_id	gender	age	category	quantiy	price_per_unit	cogs	Total_sale
	31	2023-12-31	17:47:00	3	Male	44	Electronics	4	300	129	1200
	46	2022-11-08	17:50:00	54	Female	20	Electronics	4	300	84	1200
	58	2023-09-16	19:18:00	53	Male	18	Clothing	4	300	75	1200
	67	2023-08-19	20:19:00	119	Female	48	Beauty	4	300	129	1200
	99	2023-11-19	15:12:00	71	Female	50	Electronics	4	300	132	1200
	107	2022-10-06	09:18:00	75	Female	21	Clothing	4	300	78	1200
	142	2022-04-08	10:05:00	61	Male	35	Electronics	4	300	138	1200
	165	2022-09-24	19:01:00	75	Female	60	Clothing	4	300	318	1200
	181	2023-11-06	19:59:00	79	Male	19	Electronics	4	300	327	1200
	202	2023-01-25	06:24:00	61	Female	34	Clothing	4	300	120	1200
	292	2022-10-10	06:33:00	111	Male	20	Beauty	4	300	141	1200
	296	2022-12-31	22:44:00	73	Female	22	Clothing	4	300	369	1200
	298	2023-10-05	19:05:00	3	Male	27	Beauty	4	300	84	1200
	308	2022-03-31	19:12:00	130	Female	34	Beauty	4	300	126	1200
	320	2022-04-20	08:35:00	57	Female	28	Electronics	4	300	159	1200

Write a SQL query to find the total number of transactions (transaction\_id) made by each gender in each category.

```
4 • SELECT      category , gender , COUNT(Transactions_id) AS Total_transaction  
5      FROM retail_sales  
6      GROUP BY gender , category  
7      ORDER BY 1;
```

Result Grid			
	category	gender	Total_transaction
▶	Beauty	Female	332
	Clothing	Female	348
	Electronics	Female	340
	Beauty	Male	282
	Clothing	Male	354
	Electronics	Male	344

Write a SQL query to calculate the average sale for each month. Find out best selling month in each year

```
4 •     SELECT *
5      FROM
6      (
7          SELECT
8              YEAR(sale_date) AS years,
9              MONTH(sale_date) AS months,
10             ROUND(AVG(Total_sale),2) AS Avg_sale,
11             RANK() OVER(PARTITION BY YEAR(sale_date) ORDER BY AVG(Total_sale) DESC) AS RANKS
12         FROM retail_sales
13         GROUP BY 1 , 2) AS T1
14         WHERE RANKS = 1;
15         -- ORDER BY 1, 3 DESC;
```

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

	years	months	Avg_sale	RANKS
▶	2022	7	534.64	1
	2023	2	535.53	1

Write a SQL query to find the top 5 customers based on the highest total sales

```
3 • SELECT customer_id , SUM(Total_sale) AS Total_sales  
4   FROM retail_sales  
5   GROUP BY customer_id  
6   ORDER BY Total_sales DESC LIMIT 5;  
7  
8
```

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

	customer_id	Total_sales
▶	3	38440
	1	30750
	5	30405
	2	25295
	4	23580

Write a SQL query to find the number of unique customers who purchased items from each category.

```
3 •   SELECT category , COUNT(DISTINCT customer_id) AS Customer  
4     FROM retail_sales  
5   GROUP BY category;
```

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

	category	Customer
▶	Beauty	141
	Clothing	149
	Electronics	144

Write a SQL query to create each shift and number of orders  
(Example Morning <12, Afternoon Between 12 & 17, Evening >17)

```
4 • SELECT
5     CASE
6         WHEN HOUR(sale_time) < 12 THEN 'MORNING'
7         WHEN HOUR(sale_time) BETWEEN 12 AND 17 THEN 'AFTERNOON'
8         ELSE 'EVENING'
9     END AS Shift,
10    COUNT(Transactions_id) AS Total_orders
11   FROM retail_sales
12   GROUP BY Shift;
```

---

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

	Shift	Total_orders
▶	EVENING	1062
	MORNING	561
	AFTERNOON	377



# Thank You

