Sales Store project for MSSQL

Store Background

A retail store chain tracks daily sales transactions, including order details, customer info, product categories, order times, and order status. The business wants to optimize operations, improve customer experience, and increase profitability using data-driven decisions.

Problem Statement

The store doesn't have a clear idea about :-

which products sell the most,

customers preference,

which items bring in the most profit,and

where things are going wrong indelivery or operations. Because of this, they aremissing chances to earnmore, losing customers, andmaking poor businessdecisions.

Solution

They need proper reports and simpleinsights to understand their sales,customers, and product performancebetter.

Check Dataset

```
select * from [sales store sql]
```

```
select * INTO sales_store from [sales store sql]
```

```
select * from sales_store
```

--- Data cleaning

Step 1 - To check for duplicates

```
select transaction_id , count(*)
from sales_store
group by transaction_id
having count(transaction_id) >1
```

	transaction_id
1	TXN240646
2	TXN342128
3	TXN855235
4	TXN981773

```
with CTE AS(
select *,
   ROW_NUMBER() over (partition by transaction_id order by transaction_id) AS Row_num
from sales_store)
select * from CTE
where transaction_id IN ('TXN240646','TXN342128','TXN855235','TXN981773')
```

	transaction_id	customer_id	customer_name	customer_age	gender	product_id	product_name	product_category	Quantity	Price	payment_mode	purchase_date	time_of_purchase	status	Row_num
1	TXN240646	CUST5356	Renee Kant	37	F	P4193	Textbook	Books	2	9770	UPI	2023-09-23	11:43:16.0000000	cancelled	1
2	TXN342128	CUST1214	Nehmat Batta	39	M	P1286	Sweater	Clothing	3	1284	Cash	2023-08-24	20:05:11.0000000	returned	1
3	TXN855235	CUST5938	Yashvi Sachar	44	M	P6308	Smartwatch	Electronics	3	11214	EMI	2023-08-26	16:59:09.0000000	delivered	1
4	TXN981773	CUST2365	Tushar Chakrabarti	50	M	P5760	Milk	Groceries	4	4120	Cash	2023-11-19	11:01:31.0000000	cancelled	1

---delete this 4 duplicate records

```
with CTE AS(
select *, ROW_NUMBER() over (partition by transaction_id order by transaction_id) AS Row_num
from sales_store)
Delete from CTE
where Row_Num = 2
```

Step 2 :- Correction of headers

```
EXEC sp_rename 'sales_store.quantiy', 'Quantity','COLUMN'
EXEC sp_rename 'sales_store.prce', 'Price','COLUMN'
```

```
Commands completed successfully.

Completion time: 2025-07-31T11:50:23.6600762+05:30
```

Step 3 :- check datatype

```
select COLUMN_NAME, DATA_TYPE
FROM INFORMATION_SCHEMA.COLUMNS
```

WHERE TABLE_NAME = 'sales_store'

	COLUMN_NAME	DATA_TYPE
1	transaction_id	varchar
2	customer_id	varchar
3	customer_name	varchar
4	customer_age	int
5	gender	varchar
6		
7	product_name	varchar
8	product_category	varchar
9	Quantity	int
10	Price	float
11	payment_mode	varchar
12	purchase_date	date
13	time_of_purchase	time
14	status	varchar

Click to select the whole row

--change price to float

Alter Table sales_store

Alter Column Price float

```
Commands completed successfully.

Completion time: 2025-07-31T11:50:23.6600762+05:30
```

Step 4 :- To check null values

--Exceute the dynamic sql

EXEC sp_executesql @sql;

Declare @SQL NVARCHAR(MAX) = '';

SELECT @SQL = STRING_AGG(

'SELECT ''' + COLUMN_NAME + ''' AS ColumnName,

COUNT(*) AS NullCount

FROM ' + QUOTENAME(TABLE_SCHEMA) + '.sales_store

WHERE ' + QUOTENAME(COLUMN_NAME) + 'IS NULL',

'UNION ALL'

)

WITHIN GROUP(ORDER BY COLUMN_NAME)

FROM INFORMATION_SCHEMA.COLUMNS

WHERE TABLE_NAME = 'sales_stores';

	ColumnName	NullCount
1	customer_age	2
2	customer_id	3
3	customer_name	2
4	gender	2
5	payment_mode	1
6	price	0
7	product_categ...	0
8	product_id	0
9	product_name	0
10	purchase_date	1
11	quantity	0
12	status	1
13	time_of_purch...	1
14	transaction_id	1

--- exceute the dynamic sql

EXEC sp_excecutesql @SQL;

-----treting null values

select * from sales_store

where transaction_id is null

or transaction_id is null

or customer_id is null

or customer_name is null

or customer_age is null

or gender is null

or product_id is null

or product_name is null

or product_category is null

or Quantity is null

or Quantity is null

or Price is null

or payment_mode is null

or time_of_purchase is null

or purchase_date is null

	transaction_id	customer_id	customer_name	customer_age	gender	product_id	product_name	product_category	quantity	price	payment_mode	purchase_date	time_of_purchase	status
1	NULL	NULL	NULL	NULL	NULL	P4524	T-Shirt	Clothing	4	14788	NULL	NULL	NULL	NULL
2	TXN432798	CUST1003	NULL	NULL	NULL	P5717	Dining Table	Furniture	3	2346	EMI	2023-04-30	05:46:48.0000000	cancelled
3	TXN977900	NULL	Ehsaan Ram	25	M	P8212	Milk	Groceries	3	12342	Cash	2023-08-13	07:46:18.0000000	returned
4	TXN985663	NULL	Damini Raju	49	Female	P3367	Notebook	Books	5	8130	EMI	2023-01-24	01:40:38.0000000	cancelled

--delete record for transaction id is null bcz not having proper records

Delete from sales_store

where transaction_id is null

```
Commands completed successfully.

Completion time: 2025-07-31T11:50:23.6600762+05:30
```

--check cust id of ehsaan ram from previous sales and update his customer id

select * from sales_store

where Customer_name = 'Ehsaan Ram'

	transaction_id	customer_id	customer_name	customer_age	gender	product_id	product_name	product_category	Quantity	Price	payment_mode	purchase_date	time_of_purchase	status
1	TXN553735	CUST9494	Ehsaan Ram	25	M	P3115	Handbag	Accessories	2	4220	Debit Card	2023-11-22	11:21:36.0000000	cancelled
2	TXN977900	CUST9494	Ehsaan Ram	25	M	P8212	Milk	Groceries	3	12342	Cash	2023-08-13	07:46:18.0000000	returned
3	TXN495746	CUST9494	Ehsaan Ram	25	M	P7385	Wardrobe	Furniture	3	7164	Credit Card	2023-04-28	23:40:54.0000000	cancelled
4	TXN586594	CUST9494	Ehsaan Ram	25	M	P4479	Shirt	Clothing	4	11832	Cash	2023-11-26	22:05:45.0000000	pending

From above records we can see custID is CUST9494

update sales_store

SET customer_id = 'CUST9494'

where transaction_id = 'TXN977900'

```
(1 row affected)

Completion time: 2025-07-31T12:10:54.9591041+05:30
```

--check cust id of 'Damini Raju' from previous sales and update his customer id

select * from sales_store

where Customer_name = 'Damini Raju'

	transaction_id	customer_id	customer_name	customer_age	gender	product_id	product_name	product_category	Quantity	Price	payment_mode	purchase_date	time_of_purchase	status
1	TXN749812	CUST1401	Damini Raju	49	F	P2348	Dining Table	Furniture	5	2455	Credit Card	2023-04-29	02:25:23.0000000	returned
2	TXI	Click to select the whole column		49	F	P3367	Notebook	Books	5	8130	EMI	2023-01-24	01:40:38.0000000	cancelled
3	TXN227567	CUST1401	Damini Raju	49	F	P9862	Handbag	Accessories	1	4496	Debit Card	2023-07-25	19:30:41.0000000	pending
4	TXN287174	CUST1401	Damini Raju	49	F	P4575	Jacket	Clothing	3	11532	Cash	2023-08-19	00:36:11.0000000	pending

From above records we can see custID is CUST1401 so update this records

update sales_store

SET customer_id = 'CUST1401'

where transaction_id = 'TXN985663'

```
(1 row affected)

Completion time: 2025-07-31T12:12:28.0279962+05:30
```

----- check cust name, gender, cust age for CUST1003 and update details with previous details

select * from sales_store

where customer_id = 'CUST1003'

	transaction_id	customer_id	customer_name	customer_age	gender	product_id	product_name	product_category	Quantity	Price	payment_mode	purchase_date	time_of_purchase	status
1	TXN737152	CUST1003	Mahika Saini	35	M	P3228	Bed	Furniture	4	13332	Credit Card	2023-10-14	20:59:58.0000000	pending
2	TXN432798	CUST1003	Mahika Saini	35	M	P5717	Dining Table	Furniture	3	2346	EMI	2023-04-30	05:46:48.0000000	cancelled
3	TXN553240	CUST1003	Mahika Saini	35	M	P5769	Wallet	Accessories	4	1412	Credit Card	2023-03-12	08:45:44.0000000	cancelled
4	TXN834755	CUST1003	Mahika Saini	35	M	P1071	Jacket	Clothing	1	655	Credit Card	2023-06-18	00:06:38.0000000	pending

Check above details and update details

update sales_store

SET customer_name = 'Mahika Saini', customer_age = '35' , gender = 'Male'

where transaction_id = 'TXN432798'

--Step 5 :- Data Cleaning

--update gender f to female and m to male

select distinct gender

from sales_store

	gender
1	F
2	Male
3	M
4	Female

update sales_store

SET gender = 'M'

where gender = 'Male'

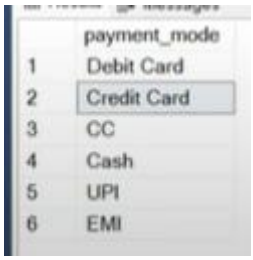
```
(1 row affected)

Completion time: 2025-07-31T12:13:51.3910399+05:30
```

--update payment mode cc to credit card

select distinct payment_mode

from sales_store



	payment_mode
1	Debit Card
2	Credit Card
3	CC
4	Cash
5	UPI
6	EMI

update sales_store

SET Payment_mode = 'Credit Card'

where Payment_mode = 'CC'

```
(1 row affected)

Completion time: 2025-07-31T12:13:51.3910399+05:30
```

--DATA ANALYSIS

--# 1. What are the top 5 most selling products by quantity

select Top 5 product_name, sum(Quantity) AS total_quantity_sold

from sales_store

where status = 'delivered'   -- we don't want returned, cancelled or pending products

Group by product_name

order by total_quantity_sold DESC

	product_name	total_quantity_sold
1	Wardrobe	70
2	Vegetables	69
3	Sofa	66
4	Dining Table	65
5	Fruits	60

--Business problem : we don't know which products are most in demand

--Business Impact : Helps prioritize stock nd boost sales through targeted promotions.

---# 2. which products are mostly frequently cancelled

select Top 5 product_name, Count(*) AS total_cancelled

```
from sales_store
where status = 'cancelled'   -- we don't want returned, cancelled or pending products
Group by product_name
order by total_cancelled DESC
```

	product_name	total_cancelled
1	Comics	24
2	Sweater	23
3	Chair	21
4	Vegetables	21
5	Smartphone	20

--Business problem : Frequent cancellation affect revenue and customer trust

--Business Impact : Identify poor-performing products to improve quality or remove from catlog

---

---# 3. What time of the day has highest number of purchase

```
select     case   when DATEPART(Hour, time_of_purchase) BETWEEN 0 AND 5 THEN 'NIGHT'
                  when DATEPART(Hour, time_of_purchase) BETWEEN 5 AND 11 THEN 'Morning'
                  when DATEPART(Hour, time_of_purchase) BETWEEN 12 AND 17 THEN 'AFTERNOON'
                  when DATEPART(Hour, time_of_purchase) BETWEEN 18 AND 23 THEN 'EVENING'
          End AS Time_of_day,
          count(*) AS total_order
           from sales_store
             Group by
             case  when DATEPART(Hour, time_of_purchase) BETWEEN 0 AND 5 THEN 'NIGHT'
                   when DATEPART(Hour, time_of_purchase) BETWEEN 5 AND 11 THEN 'Morning'
                   when DATEPART(Hour, time_of_purchase) BETWEEN 12 AND 17 THEN 'AFTERNOON'
                   when DATEPART(Hour, time_of_purchase) BETWEEN 18 AND 23 THEN 'EVENING'
          End
```

	Time_of_day	total_order
1	Morning	514
2	NIGHT	496
3	EVENING	515
4	AFTERNOON	475

--Business problem : Find peak sales times.

--Business Impact : optimize staffing, promotions, and server loads

---

---# 4. Who are top 5 highest spending customers?

```
select Top 5 customer_name,
    Format(sum(Price * Quantity), 'C0', 'en-IN') AS highest_spending     -----C0 - c stands for curent, o-seprated by (,) ,'en-IN' - In stands for indian curreny
from sales_store
Group by Customer_name
order by sum(Price * Quantity) DESC
```

	customer_name	highest_spending
1	Darshit Mann	₹ 5,07,530
2	Anahita Shenoy	₹ 4,55,637
3	Saira Ahluwalia	₹ 4,47,933
4	Gatik Khare	₹ 3,86,156
5	Samaira Subramaniam	₹ 3,57,388

--Business problem : Identify VIP customers

--Business Impact : Personalized offers, loyalty rewards, and retention

---

---# 5. which product category generate highest revenue

```
select product_category,
    Format(sum(Price * Quantity), 'C0', 'en-IN') AS highest_revenue     -----C0 - c stands for curent, o-seprated by (,) ,'en-IN' - In stands for indian curreny
from sales_store
Group by product_category
order by sum(Price * Quantity) DESC
```

	product_category	highest_revenue
1	Accessories	₹ 1,03,65,306
2	Clothing	₹ 1,01,95,727
3	Books	₹ 99,12,929
4	Furniture	₹ 96,59,478
5	Electronics	₹ 95,04,028
6	Groceries	₹ 94,64,153

--Business problem : Identifytop-performing product categories

--Business Impact : Allowing business to invest more in high- margin or high demand categories

---

---# 6. what is return/cancellation rate per product category

---for cancellation

```
select product_category,

format(count( case when status = 'cancelled' Then 1 END) * 100.0 /  Count(*), 'N3') + ' %' AS cancelled_percent   ---- N3- N stand for no fomat 3 for 3 decimal digit and % to add percentage symbol at end

from sales_store

Group by product_category

order by cancelled_percent DESC
```

	product_category	cancelled_percent
1	Books	26.205 %
2	Clothing	25.634 %
3	Electronics	24.675 %
4	Accessories	23.547 %
5	Furniture	22.832 %
6	Groceries	22.289 %

---for returned

```
select product_category,

   format(count( case when status = 'Returned' Then 1 END) * 100.0 /  Count(*), 'N3') + ' %' AS Returned_percent   ---- N3- N stand for no fomat 3 for 3 decimal digit and % to add percentage symbol at end

from sales_store

Group by product_category

order by Returned_percent DESC
```

	product_category	Returned_percent
1	Accessories	31.498 %
2	Books	25.602 %
3	Clothing	24.789 %
4	Groceries	23.494 %
5	Furniture	23.410 %
6	Electronics	20.779 %

--Business problem : Reduce returns, improve product desciptions/expectations

--Business Impact :help idntify and fix producr or logistics issues

---

----7. what is most prefeered payment mode ?

```
select payment_mode, count(payment_mode) as Prefered_mode
```

```
from sales_store
group by payment_mode
order by Prefered_mode desc
```

	payment_mode	Prefered_mode
1	Credit Card	648
2	EMI	350
3	Debit Card	344
4	Cash	332
5	UPI	326

--Business problem : Know which pyment options customers prefer

--Business Impact : stremline payment procesing, prioritize popular modes

---

----#8. How does age group affect purchasing behavior?

```
--select min(customer_age), max(customer_age)
--from sales_store

select
    case
            when customer_age BETWEEN  18 AND 25 THEN '18-25'
            when customer_age BETWEEN  26 AND 35 THEN '26-35'
            when customer_age BETWEEN  36 AND 50 THEN '36-50'
            else '50+'
    End as Customer_age,
        format(sum(price*quantity), 'C0', 'en-IN' )AS total_purchase
from sales_store
group by case
            when customer_age BETWEEN  18 AND 25 THEN '18-25'
            when customer_age BETWEEN  26 AND 35 THEN '26-35'
            when customer_age BETWEEN  36 AND 50 THEN '36-50'
            else '50+'
           End
order by total_purchase DESC
```

	Customer_age	total_purchase
1	36-50	₹ 1,94,60,276
2	50+	₹ 1,43,86,538
3	26-35	₹ 1,36,96,027
4	18-25	₹ 1,15,58,780

--Business problem : Understand customer demographics

--Business Impact : Trageted maketing and roduct recommondations by age group

---

----#9. Whats the monthly sales trend?

--Method 1

```
select
   format(purchase_date, 'yyyy-MM') AS Month_year,
          format(sum(Price * Quantity),'C0', 'en-IN' ) AS total_sales,
          sum(Quantity) AS total_quantity
from sales_store
group by format(purchase_date, 'yyyy-MM')
```

	Month_year	total_sales	total_quantity
1	2023-01	₹ 46,28,608	478
2	2023-02	₹ 46,98,929	529
3	2023-03	₹ 52,41,364	471
4	2023-04	₹ 49,89,315	505
5	2023-05	₹ 39,02,263	418
6	2023-06	₹ 41,00,112	478
7	2023-07	₹ 51,29,904	577
8	2023-08	₹ 47,88,207	497
9	2023-09	₹ 50,37,847	512
10	2023-10	₹ 58,86,414	547
11	2023-11	₹ 51,09,229	523
12	2023-12	₹ 52,49,987	521
13	2024-01	₹ 3,39,442	31

--method 2

```
   select
          year(purchase_date) AS Years,
                Month(purchase_date) AS Months,
                format(sum(Price * Quantity),'C0', 'en-IN' ) AS total_sales,
          sum(Quantity) AS total_quantity
from sales_store
group by YEAR(purchase_date), Month(purchase_date)
order by Months
```

	Years	Months	total_sales	total_quantity
1	2023	1	₹ 46,28,608	478
2	2024	1	₹ 3,39,442	31
3	2023	2	₹ 46,98,929	529
4	2023	3	₹ 52,41,364	471
5	2023	4	₹ 49,89,315	505
6	2023	5	₹ 39,02,263	418
7	2023	6	₹ 41,00,112	478
8	2023	7	₹ 51,29,904	577
9	2023	8	₹ 47,88,207	497
10	2023	9	₹ 50,37,847	512
11	2023	10	₹ 58,86,414	547
12	2023	11	₹ 51,09,229	523
13	2023	12	₹ 52,49,987	521

--Business problem : Sales fluctutions go unnoticed

--Business Impact : plan inventory and marketing according to seasonal trends

---

--#10. are certain genders being more specific product categories?

--Method 1

```
select gender, product_category, count(product_category) AS total_purchase
from sales_store
group by gender, product_category
order by gender DESC
```

	gender	product_category	total_purchase
1	Male	Furniture	1
2	M	Furniture	162
3	M	Accessories	171
4	M	Groceries	167
5	M	Clothing	175
6	M	Books	180
7	M	Electronics	161
8	F	Furniture	183
9	F	Books	152
10	F	Clothing	180
11	F	Accessories	156
12	F	Groceries	165
13	F	Electronics	147

--Method 2

```
 select *
 from(
    select gender, product_category
            from sales_store
```

```
        ) AS source_table
pivot (
  count(gender)
  for gender IN ([F], [M])
  ) AS pivot_table
 order by product_category
```

	product_category	F	M
1	Accessories	156	171
2	Books	152	180
3	Clothing	180	175
4	Electronics	147	161
5	Furniture	183	162
6	Groceries	165	167

--**Business problem :** gender based product preferences

--**Business Impact :** personalized ads, gender focused campaigns

---