NAME-SONALI KUMARI

RESUME LINK- [Sonali\_Kumari\_2024PGCSIS02 (1).pdf](https://drive.google.com/file/d/1LPGLj2mmw7-b_DfXRB5cag5iQ8dhvgMC/view?usp=sharing)

OUTPUT FOLDER LINK-

[Assignment output](https://drive.google.com/drive/folders/1PaaTL_1wZpUn0b73XSaPFezrnYM-cpaj?usp=drive_link)

I’ve used SQL Server for the assignment task in which I first did the data cleaning part by checking for null, duplicate, and negative values and datatypes before proceeding with the further assignment tasks.

Code Link:

[SQLQuery1.sql](https://drive.google.com/file/d/1Xt7GyR6hC9o6nDabjghn0a7ZFcIfe93L/view?usp=drive_link)

select \* from assignmentData;

--To check for duplicates

SELECT ENTRY\_ID, SALES\_REP\_ID, CUSTOMER\_ID, CUSTOMER\_CODE, SKU\_NAME,

COUNT(\*) AS duplicate\_count

FROM assignmentData

GROUP BY ENTRY\_ID, SALES\_REP\_ID, CUSTOMER\_ID, CUSTOMER\_CODE, SKU\_NAME

HAVING COUNT(\*) > 1;

WITH ranked AS (

SELECT \*,

ROW\_NUMBER() OVER (

PARTITION BY ENTRY\_ID, SALES\_REP\_ID, CUSTOMER\_ID, CUSTOMER\_CODE, SKU\_NAME

ORDER BY ENTRY\_ID

) AS rn

FROM assignmentData

)

DELETE FROM ranked

WHERE rn > 1;

--Check for Null values and handle it

SELECT

SUM(CASE WHEN ENTRY\_ID IS NULL THEN 1 ELSE 0 END) AS null\_entry\_id,

SUM(CASE WHEN SALES\_REP\_ID IS NULL THEN 1 ELSE 0 END) AS null\_sales\_rep\_id,

SUM(CASE WHEN CUSTOMER\_CODE IS NULL THEN 1 ELSE 0 END) AS null\_customer\_code,

SUM(CASE WHEN CUSTOMER\_NAME IS NULL THEN 1 ELSE 0 END) AS null\_customer\_name,

SUM(CASE WHEN SKU\_NAME IS NULL THEN 1 ELSE 0 END) AS null\_sku\_name,

SUM(CASE WHEN UNIT\_SOLD IS NULL THEN 1 ELSE 0 END) AS null\_unit\_sold,

SUM(CASE WHEN TOTAL\_VALUE\_SOLD IS NULL THEN 1 ELSE 0 END) AS null\_total\_value\_sold,

SUM(CASE WHEN CHECKOUT\_TIME IS NULL THEN 1 ELSE 0 END) AS null\_checkout\_time

FROM assignmentData;

-- Replace missing CUSTOMER\_NAME with 'Unknown'

UPDATE assignmentData

SET CUSTOMER\_NAME = 'Unknown'

WHERE CUSTOMER\_NAME IS NULL;

-- Replace missing UNIT\_SOLD with 0

UPDATE assignmentData

SET UNIT\_SOLD = 0

WHERE UNIT\_SOLD IS NULL;

--Change the datatype of req field

UPDATE assignmentData

SET CUSTOMER\_CODE = TRY\_CAST(REPLACE(CUSTOMER\_CODE, '.0', '') AS INT)

WHERE CUSTOMER\_CODE IS NOT NULL;

UPDATE assignmentData

SET ENTRY\_ID = TRY\_CAST(REPLACE(ENTRY\_ID, '.0', '') AS INT)

WHERE ENTRY\_ID IS NOT NULL;

UPDATE assignmentData

SET SALES\_REP\_ID = TRY\_CAST(REPLACE(SALES\_REP\_ID, '.0', '') AS INT)

WHERE SALES\_REP\_ID IS NOT NULL;

--Handle negative values

EXEC sp\_help 'assignmentData';

SELECT \*

FROM assignmentData

WHERE TRY\_CAST(UNIT\_SOLD AS FLOAT) < 0

OR TRY\_CAST(TOTAL\_VALUE\_SOLD AS FLOAT) < 0;

-- First, update values (remove unwanted characters, just in case)

UPDATE assignmentData

SET UNIT\_SOLD = REPLACE(UNIT\_SOLD, '.0', '');

UPDATE assignmentData

SET TOTAL\_VALUE\_SOLD = REPLACE(TOTAL\_VALUE\_SOLD, '.0', '');

-- Then alter column types

ALTER TABLE assignmentData

ALTER COLUMN UNIT\_SOLD FLOAT;

ALTER TABLE assignmentData

ALTER COLUMN TOTAL\_VALUE\_SOLD FLOAT;

SELECT \*

FROM assignmentData

WHERE UNIT\_SOLD < 0 OR TOTAL\_VALUE\_SOLD < 0;

-- Fixing inconsistent data formats and InvalidData

-- Standardize CUSTOMER\_NAME to uppercase

UPDATE assignmentData

SET CUSTOMER\_NAME = UPPER(CUSTOMER\_NAME);

-- Convert CHECKIN\_TIME to proper DATETIME

ALTER TABLE assignmentData

ALTER COLUMN CHECKIN\_TIME DATETIME;

--Checking the datatype

-- Preview converted values

SELECT

CHECKIN\_TIME,

TRY\_CONVERT(DATETIME, CHECKIN\_TIME, 101) AS CHECKIN\_CONVERTED,

CHECKOUT\_TIME,

TRY\_CONVERT(DATETIME, CHECKOUT\_TIME, 101) AS CHECKOUT\_CONVERTED

FROM assignmentData;

ALTER TABLE assignmentData

ALTER COLUMN CHECKIN\_TIME DATETIME;

ALTER TABLE assignmentData

ALTER COLUMN CHECKOUT\_TIME DATETIME;

ALTER TABLE assignmentData

ALTER COLUMN ENTRY\_ID INT;

ALTER TABLE assignmentData

ALTER COLUMN SALES\_REP\_ID INT;

ALTER TABLE assignmentData

ALTER COLUMN CUSTOMER\_ID INT;

ALTER TABLE assignmentData

ALTER COLUMN CUSTOMER\_CODE INT;

SELECT COLUMN\_NAME, DATA\_TYPE

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'assignmentData'; -- if table name is all lowercase in metadata

select \* from assignmentData;

**## 1. Top 10 SKUs Analysis**

**### 1.1. Identify the Top 10 most-selling SKUs:**

- By **\*\*Quantity Sold\*\***

- By **\*\*Value Sold\*\***

**### 1.2. Identify the Top 10 least-selling SKUs:**

- By **\*\*Quantity Sold\*\***

- By **\*\*Value Sold\*\***

**/\* Q1 - TOP 10 SKUs Analysis**

**1.1 - Identify the top 10 most selling SKUs:**

**a) By Quantity Sold**

**\*/**

SELECT SKU\_NAME, SUM(UNIT\_SOLD) AS Total\_Quantity\_Sold

FROM assignmentData

GROUP BY SKU\_NAME

ORDER BY Total\_Quantity\_Sold DESC

OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

**--b) BY Value sold**

SELECT SKU\_NAME,SUM(TOTAL\_VALUE\_SOLD) AS Total\_Value\_Sold

FROM assignmentData

GROUP BY SKU\_NAME

ORDER BY Total\_Value\_Sold DESC

OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

**/\*1.2-Identify the Top 10 least-selling SKUs:**

**a)By Quantity Sold\*/**

SELECT SKU\_NAME, SUM(UNIT\_SOLD) AS Total\_Quantity\_Sold

FROM assignmentData

GROUP BY SKU\_NAME

ORDER BY Total\_Quantity\_Sold asc

OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

**--b)By Value Count**

SELECT SKU\_NAME,SUM(TOTAL\_VALUE\_SOLD) AS Total\_Value\_Sold

FROM assignmentData

GROUP BY SKU\_NAME

ORDER BY Total\_Value\_Sold asc

OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

**## 2. Customer Analysis**

**- Find the \*\*Top 10 Customers\*\* by \*\*Total Value Purchased\*\***

**/\*2-Customer Analysis**

**Find the top 10 customers by total value purchased\*/**

SELECT

CUSTOMER\_ID,

CUSTOMER\_NAME,

SUM(TOTAL\_VALUE\_SOLD) AS total\_value\_purchased

FROM

assignmentData

WHERE

TOTAL\_VALUE\_SOLD IS NOT NULL

GROUP BY

CUSTOMER\_ID,

CUSTOMER\_NAME

ORDER BY

total\_value\_purchased DESC

OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

**## 3. Sales Representative Performance**

**### 3.1. Identify the Top 10 Sales Representatives:**

**- By \*\*Value Sold\*\***

**- By \*\*Time Spent\*\***

**### 3.2. For the Top 10 Sales Representatives (by Value), calculate:**

**- \*\*Day-wise Average Value Sold\*\***

**- \*\*Day-wise Average Time Spent\*\***

**/\*3-Sales Representative Performance**

**3.1-Identify the Top 10 Sales Performance:**

**a)By Value sold\*/**

SELECT

SALES\_REP\_ID,

SALES\_REP,

SUM(TOTAL\_VALUE\_SOLD) AS total\_value\_sold

FROM

assignmentData

WHERE

TOTAL\_VALUE\_SOLD IS NOT NULL

GROUP BY

SALES\_REP\_ID,

SALES\_REP

ORDER BY

total\_value\_sold DESC

OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

**--b)By Time spent**

SELECT

SALES\_REP\_ID,

SALES\_REP,

SUM(DATEDIFF(

MINUTE,

TRY\_CONVERT(DATETIME, CHECKIN\_TIME, 101), -- 101 = mm/dd/yyyy

TRY\_CONVERT(DATETIME, CHECKOUT\_TIME, 101)

)) AS total\_minutes\_spent

FROM assignmentData

WHERE

CHECKIN\_TIME IS NOT NULL

AND CHECKOUT\_TIME IS NOT NULL

GROUP BY

SALES\_REP\_ID,

SALES\_REP

ORDER BY

total\_minutes\_spent DESC

OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

**/\*3.2. For the Top 10 Sales Representatives (by Value), calculate:**

**a) Day-wise Average Value sold\*/**

**/\* --- 1. Identify Top 10 Sales Reps by Total Value --- \*/**

WITH TopValueReps AS (

SELECT

SALES\_REP\_ID,

SALES\_REP,

SUM(TOTAL\_VALUE\_SOLD) AS total\_value\_sold

FROM assignmentData

WHERE TOTAL\_VALUE\_SOLD IS NOT NULL

GROUP BY SALES\_REP\_ID, SALES\_REP

ORDER BY total\_value\_sold DESC

OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY

),

**/\* --- 2. Compute each rep’s total value per day --- \*/**

DailyTotals AS (

SELECT

SALES\_REP\_ID,

SALES\_REP,

CAST(TRY\_CONVERT(DATE, CHECKIN\_TIME, 101) AS DATE) AS visit\_date,

SUM(TOTAL\_VALUE\_SOLD) AS daily\_value

FROM assignmentData

WHERE TOTAL\_VALUE\_SOLD IS NOT NULL

GROUP BY

SALES\_REP\_ID,

SALES\_REP,

CAST(TRY\_CONVERT(DATE, CHECKIN\_TIME, 101) AS DATE)

)

**/\* --- 3. Average the daily totals --- \*/**

SELECT

d.SALES\_REP\_ID,

d.SALES\_REP,

AVG(d.daily\_value) AS avg\_daily\_value\_sold

FROM DailyTotals d

JOIN TopValueReps t

ON d.SALES\_REP\_ID = t.SALES\_REP\_ID

GROUP BY

d.SALES\_REP\_ID,

d.SALES\_REP

ORDER BY

avg\_daily\_value\_sold DESC;

**--b)Day wise Average Time Spent**

**/\* --- 1. Identify Top 10 Sales Reps by Total Value --- \*/**

WITH TopValueReps AS (

SELECT

SALES\_REP\_ID,

SALES\_REP,

SUM(TOTAL\_VALUE\_SOLD) AS total\_value\_sold

FROM assignmentData

WHERE TOTAL\_VALUE\_SOLD IS NOT NULL

GROUP BY SALES\_REP\_ID, SALES\_REP

ORDER BY total\_value\_sold DESC

OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY

),

**/\* --- 2. Compute each rep’s total minutes spent per day --- \*/**

DailyTime AS (

SELECT

SALES\_REP\_ID,

SALES\_REP,

CAST(TRY\_CONVERT(DATE, CHECKIN\_TIME, 101) AS DATE) AS visit\_date,

SUM(

DATEDIFF(

MINUTE,

TRY\_CONVERT(DATETIME, CHECKIN\_TIME, 101),

TRY\_CONVERT(DATETIME, CHECKOUT\_TIME, 101)

)

) AS daily\_minutes

FROM assignmentData

WHERE CHECKIN\_TIME IS NOT NULL

AND CHECKOUT\_TIME IS NOT NULL

GROUP BY

SALES\_REP\_ID,

SALES\_REP,

CAST(TRY\_CONVERT(DATE, CHECKIN\_TIME, 101) AS DATE)

)

**/\* --- 3. Average the daily minutes --- \*/**

SELECT

d.SALES\_REP\_ID,

d.SALES\_REP,

AVG(d.daily\_minutes) AS avg\_daily\_minutes\_spent

FROM DailyTime d

JOIN TopValueReps t

ON d.SALES\_REP\_ID = t.SALES\_REP\_ID

GROUP BY

d.SALES\_REP\_ID,

d.SALES\_REP

ORDER BY

avg\_daily\_minutes\_spent DESC;

**## 4. Detailed Reports for Top 3 Sales Representatives (by Value)**

For each of the **\*\*Top 3 Sales Representatives\*\***, create an Excel file for each sales rep with the following sheets:

**### Day-wise Transactions Sheets with following columns**

- SKU Sold (Index)

- Price of each SKU

- Quantity Sold

- Value Sold

**–To get top 3 SALES\_REP for clarity**

WITH SalesTotals AS (

SELECT

SALES\_REP\_ID,

SALES\_REP,

SUM(TOTAL\_VALUE\_SOLD) AS total\_value\_sold

FROM assignmentData

WHERE TOTAL\_VALUE\_SOLD IS NOT NULL

GROUP BY SALES\_REP\_ID, SALES\_REP

)

SELECT TOP 3 \*

FROM SalesTotals

ORDER BY total\_value\_sold DESC;

**/\* —-DAY WISE Transaction Sheet— \*/**

SELECT

SALES\_REP\_ID,

SKU\_NAME AS [SKU Sold],

CAST(TOTAL\_VALUE\_SOLD \* 1.0 / UNIT\_SOLD AS DECIMAL(18,2)) AS [Price of each SKU],

UNIT\_SOLD AS [Quantity Sold],

TOTAL\_VALUE\_SOLD AS [Value Sold],

CAST(CHECKIN\_TIME AS DATE) AS [Sale Date]

FROM assignmentData

WHERE SALES\_REP\_ID IN (663, 433, 493)

ORDER BY SALES\_REP\_ID, [Sale Date], SKU\_NAME;

SELECT

CAST(CHECKIN\_TIME AS DATE) AS [Date],

SUM(UNIT\_SOLD) AS [Total Quantity Sold],

SUM(TOTAL\_VALUE\_SOLD) AS [Total Value Sold],

COUNT(DISTINCT SKU\_NAME) AS [Number of Unique SKUs Sold],

COUNT(DISTINCT CUSTOMER\_ID) AS [Count of Unique Customers Served],

COUNT(ENTRY\_ID) AS [Number of Visits Made],

-- Conversion Percentage: assuming "visit converted" if TOTAL\_VALUE\_SOLD > 0

CAST(100.0 \* SUM(CASE WHEN TOTAL\_VALUE\_SOLD > 0 THEN 1 ELSE 0 END) / COUNT(ENTRY\_ID) AS DECIMAL(5,2)) AS [Conversion Percentage],

-- Total time spent: difference between CHECKOUT\_TIME and CHECKIN\_TIME in minutes

SUM(DATEDIFF(MINUTE, CHECKIN\_TIME, CHECKOUT\_TIME)) AS [Total Time Spent (Minutes)]

FROM Sales

WHERE SALES\_REP\_ID IN (663, 433, 493)

GROUP BY CAST(CHECKIN\_TIME AS DATE)

ORDER BY [Date];

**/\* —- FINAL SUMMARY REPORT —- \*/**

SELECT

CAST(CHECKIN\_TIME AS DATE) AS [Date],

SUM(UNIT\_SOLD) AS [Total Quantity Sold],

SUM(TOTAL\_VALUE\_SOLD) AS [Total Value Sold],

COUNT(DISTINCT SKU\_NAME) AS [Number of Unique SKUs Sold],

COUNT(DISTINCT CUSTOMER\_ID) AS [Count of Unique Customers Served],

COUNT(ENTRY\_ID) AS [Number of Visits Made],

-- Conversion Percentage: assuming "visit converted" if TOTAL\_VALUE\_SOLD > 0

CAST(100.0 \* SUM(CASE WHEN TOTAL\_VALUE\_SOLD > 0 THEN 1 ELSE 0 END) / COUNT(ENTRY\_ID) AS DECIMAL(5,2)) AS [Conversion Percentage],

-- Total time spent: difference between CHECKOUT\_TIME and CHECKIN\_TIME in minutes

SUM(DATEDIFF(MINUTE, CHECKIN\_TIME, CHECKOUT\_TIME)) AS [Total Time Spent (Minutes)]

FROM Sales

WHERE SALES\_REP\_ID IN (663, 433, 493)

GROUP BY CAST(CHECKIN\_TIME AS DATE)

ORDER BY [Date];