

Salesorderdetailid has the details of the productid, salesid and the quantity.

Hence first look at the details to understand which is the most sold product

Step1: EDA

Take a high level look of all the columns

```
SELECT *  
FROM [SalesLT].[SalesOrderDetail];
```

Check what is the max order quantity of any product

```
SELECT MAX(OrderQty)  
FROM [SalesLT].[SalesOrderDetail];
```

Lets check the number of customers we have:

```
SELECT COUNT(*)  
FROM [SalesLT].[Customer];
```

847

```
SELECT COUNT(*)  
FROM [SalesLT].[CustomerAddress];
```

417

```
SELECT COUNT(*)  
FROM [SalesLT].[Address];
```

450

Answer: 25

STEP 2: To check the most sold product

=====

```

SELECT TOP 5
ProductID, SUM(OrderQty) AS TotalQuantity
FROM [SalesLT].[SalesOrderDetail]
GROUP BY ProductID
ORDER BY SUM(OrderQty) DESC;

```

Results

Query 1 ✕

 Run
  Cancel query
  Save query
  Export data as
  Show only Editor

```

1 SELECT TOP 5
2 ProductID, SUM(OrderQty) AS TotalQuantity
3 FROM [SalesLT].[SalesOrderDetail]
4 GROUP BY ProductID
5 ORDER BY SUM(OrderQty) DESC;
6 |

```

Results Messages

ProductID	TotalQuantity
864	87
884	57
877	55
870	54
712	52

Shows that the ProductID is the most sold with total quantity of 87

STEP 3: DOWNLOAD THE CSV FILE after joining appropriate columns

=====

To Fetch the results by combining Customer address, like city and pincode, unit price, Company Name of the customer, Unitprize of the product ID, and the Product Name itself form the Tables, CustomerADDRESS, CUSTOMER, ADDRESS, SALESORDERHEADER, SALESORDERDETAIL.

```

WITH previous_results AS (
SELECT
[SalesLT].[CustomerAddress].CustomerID,

```




```

[SalesLT].[Customer].CompanyName,
[SalesLT].[Address].City,
[SalesLT].[Address].PostalCode,
[SalesLT].[SalesOrderHeader].SalesOrderID,
[SalesLT].[SalesOrderHeader].OrderDate,
[SalesLT].[SalesOrderDetail].ProductID,
[SalesLT].[SalesOrderDetail].UnitPrice,
[SalesLT].[SalesOrderDetail].OrderQty,
[SalesLT].[Product].Name as ProductName

FROM [SalesLT].[CustomerAddress]
INNER JOIN [SalesLT].[Address]
ON [SalesLT].[CustomerAddress].AddressID = [SalesLT].[Address].AddressID
INNER JOIN [SalesLT].[Customer]
ON [SalesLT].[CustomerAddress].CustomerID = [SalesLT].[Customer].CustomerID
JOIN [SalesLT].[SalesOrderHeader]
ON [SalesLT].[CustomerAddress].CustomerID = [SalesLT].[SalesOrderHeader].CustomerID
JOIN [SalesLT].[SalesOrderDetail]
ON [SalesLT].[SalesOrderHeader].SalesOrderID = [SalesLT].[SalesOrderDetail].SalesOrderID
JOIN [SalesLT].[Product]
ON [SalesLT].[SalesOrderDetail].ProductID = [SalesLT].[Product].ProductID
)
SELECT *,
CASE
WHEN ProductID = 864 THEN 'YES'
ELSE 'NO'
END as 'Product_864_Bought'
FROM previous_results;

```

Query 1 X

Run ☐ Cancel query  Save query  Export data as  Show only Result

```
1
2 WITH previous_results AS (
3 SELECT
4 [SalesLT].[CustomerAddress].CustomerID,
5 [SalesLT].[Customer].CompanyName,
6 [SalesLT].[Address].City,
7 [SalesLT].[Address].PostalCode,
8 [SalesLT].[SalesOrderHeader].OrderDate,
9 [SalesLT].[SalesOrderDetail].ProductID
10
11 FROM [SalesLT].[CustomerAddress]
12 INNER JOIN [SalesLT].[Address]
13 ON [SalesLT].[CustomerAddress].AddressID = [SalesLT].[Address].AddressID
14 INNER JOIN [SalesLT].[Customer]
15 ON [SalesLT].[CustomerAddress].CustomerID = [SalesLT].[Customer].CustomerID
16 JOIN [SalesLT].[SalesOrderHeader]
17 ON [SalesLT].[CustomerAddress].CustomerID = [SalesLT].[SalesOrderHeader].CustomerID
18 JOIN [SalesLT].[SalesOrderDetail]
19 ON [SalesLT].[SalesOrderHeader].SalesOrderID = [SalesLT].[SalesOrderDetail].SalesOrderID
20 JOIN [SalesLT].[Product]
21 ON [SalesLT].[SalesOrderDetail].ProductID = [SalesLT].[Product].ProductID
22 )
23 SELECT *,
24 CASE
25 WHEN ProductID = 864 THEN 'YES'
26 ELSE 'NO'
27 END as 'Product_864_Bought'
28 FROM previous_results;
```

The query result is exported as CSV is saved to a csv file : **Sales_File1_to_Pandas**

Which is then processed using pandas to remove the duplicate entries of CustomerID and then removing the unwanted columns.

After processing the final CSV file is saved as : **Final_csv_for_Task1**

C:\CompletePython\DataScience\BigData

Final CSV file looks as below in the jupyter notebook

```
1newdf.head(40)
```

executed in 30ms, finished 23:52:35 2022-01-24

	CustomerID	CompanyName	City	PostalCode	SalesOrderID	OrderDate	UnitPrice	OrderQty	Product_864_Bought
39	29736	Action Bicycle Specialists	Woolston	WA1 4SY	71784	2008-06-01T00:00:00.0000000	34.9250	23	YES
5	29584	Futuristic Bikes	Culver City	90232	71895	2008-06-01T00:00:00.0000000	38.1000	3	YES
29	29612	Channel Outlet	High Wycombe	HP10 9QY	71885	2008-06-01T00:00:00.0000000	38.1000	3	YES
8	29485	Professional Sales and Service	Van Nuys	91411	71782	2008-06-01T00:00:00.0000000	38.1000	6	YES
35	29929	Many Bikes Store	Fullerton	92831	71902	2008-06-01T00:00:00.0000000	38.1000	6	YES
18	29967	Eastside Department Store	Union City	94587	71783	2008-06-01T00:00:00.0000000	34.9250	15	YES
11	29982	Discount Tours	Daly City	94015	71920	2008-06-01T00:00:00.0000000	38.1000	3	YES
24	29796	Riding Cycles	Liverpool	L4 4HB	71797	2008-06-01T00:00:00.0000000	34.9250	23	YES
13	29546	Bulk Discount Store	London	SE1 8HL	71938	2008-06-01T00:00:00.0000000	38.1000	4	YES
15	29877	Paints and Solvents Company	Englewood	80110	71897	2008-06-01T00:00:00.0000000	38.1000	1	YES
0	30027	Engineered Bike Systems	Oxnard	93030	71816	2008-06-01T00:00:00.0000000	63.9000	1	NO
31	30089	Thrifty Parts and Sales	Oxnard	93030	71815	2008-06-01T00:00:00.0000000	5.3940	2	NO
26	29922	Closest Bicycle Store	Milton Keynes	MK8 8ZD	71832	2008-06-01T00:00:00.0000000	158.4300	1	NO
27	30102	Sports Store	Cambridge	CB4 4BZ	71846	2008-06-01T00:00:00.0000000	461.6940	2	NO
30	30025	Essential Bike Works	Abingdon	OX14 4SE	71917	2008-06-01T00:00:00.0000000	5.3940	7	NO
33	29932	Instruments and Parts Company	Gloucestershire	GL7 1RY	71898	2008-06-01T00:00:00.0000000	445.4100	2	NO
32	30050	Metropolitan Bicycle Supply	London	W1N 9FA	71936	2008-06-01T00:00:00.0000000	23.4840	6	NO
34	29975	Sports Products Store	Santa Ana	92701	71863	2008-06-01T00:00:00.0000000	24.2940	2	NO
37	29660	Extreme Riding Supplies	Sherman Oaks	91403	71796	2008-06-01T00:00:00.0000000	602.3480	2	NO
38	29781	The Bicycle Accessories Company	Alhambra	91801	71923	2008-06-01T00:00:00.0000000	5.2142	14	NO
21	30113	Nearby Cycle Shop	West Sussex	RH15 9UD	71780	2008-06-01T00:00:00.0000000	32.9940	1	NO
23	29531	Remarkable Bike Store	Las Vegas	89106	71935	2008-06-01T00:00:00.0000000	5.2142	13	NO
10	29741	Central Bicycle Specialists	Maidenhead	SL67RJ	71946	2008-06-01T00:00:00.0000000	31.5840	1	NO
2	30033	Transport Bikes	Sandy	84070	71856	2008-06-01T00:00:00.0000000	54.8940	1	NO
3	29938	Trailblazing Sports	Cerritos	90703	71845	2008-06-01T00:00:00.0000000	12.1440	3	NO
6	29638	Aerobic Exercise Company	Camarillo	93010	71915	2008-06-01T00:00:00.0000000	48.5940	1	NO
7	29653	Thrilling Bike Tours	Santa Fe	87501	71858	2008-06-01T00:00:00.0000000	5.3940	3	NO
22	29568	Coalition Bike Company	El Segundo	90245	71899	2008-06-01T00:00:00.0000000	48.5940	1	NO
17	29847	Good Toys	Auburn	95603	71774	2008-06-01T00:00:00.0000000	356.8980	1	NO
20	29644	Vigorous Sports Store	Oxon	OX16 8RS	71867	2008-06-01T00:00:00.0000000	858.9000	1	NO
1	30072	West Side Mart	Wokingham	RG41 1QW	71776	2008-06-01T00:00:00.0000000	63.9000	1	NO
41	30019	Tachometers and Accessories	Milton Keynes	MK8 8DF	71831	2008-06-01T00:00:00.0000000	41.9940	6	NO