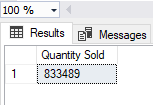
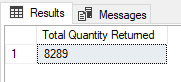
use Commercial\_Project

-- Pull the sum of quantity from each data table. Total Quantity Sold = 833,489 and Total Quantity Returned = 8,289

select SUM(quantity) AS 'Quantity Sold' from Transaction\_Data

select SUM(quantity) AS 'Total Quantity Returned' from Return\_Data





-- Pull the 'Total Transactions' and 'Total Returns' to calculate the count of rows from each data table. 269,720 transactions and 7,087 returns

select COUNT(\*) AS 'Total Transactions' from Transaction\_Data

select COUNT(return\_date) AS 'Total Returns' from Return\_Data

-- Pull the 'Return Rate' to calculate the ratio of quantity returned to quantity sold (format as %). Overall return rate of 0.99%

select (SUM(return\_data.quantity) / (select SUM(transaction\_data.quantity) from Transaction\_Data)) \* 100 as result from Return\_Data

left join Calendar on Return\_Data.return\_date = Calendar.Date

inner join Transaction\_Data on Transaction\_Data.transaction\_date = Calendar.Date

select cast(833489 / 7223 as decimal(10,2)) as result \* CAST(100 as decimal(10,2))) as result

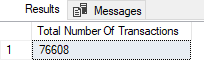
select 8289/8290 as result

-- Pull the total number of transactions on weekends. 76,608 total weekend transactions

select count(t.quantity) as 'Total Number Of Transactions' from Transaction\_Data t

left join Calendar c on c.Date = t.transaction\_date

where c.Weekend = 'Y'



-- Create a new measure named "% Weekend Transactions" to calculate weekend transactions as a percentage of total transactions (format as %).

-- I should see 28.4% weekend transactions

select CAST(COUNT(transaction\_date) as decimal(10,2)) / (select CAST(COUNT(transaction\_date) AS decimal(10,2)) from Transaction\_Data) \* 100 AS '% Number Of Transactions at Weekend' from Transaction\_Data

inner join calendar on calendar.date = Transaction\_Data.transaction\_date

where Calendar.weekend = 'Y'

A screenshot of a computer

Description automatically generated

-- Calculate % Total Count of Returns and % Of Total Count of Transactions

select cast(COUNT(return\_date) as decimal(10,2)) / (select cast(COUNT(return\_date) as decimal(10,2)) from Return\_Data) \* 100 from Return\_Data

select cast(COUNT(transaction\_date) as decimal(10,2)) / (select CAST(COUNT(transaction\_date) AS decimal(10,2)) from Transaction\_Data) \* 100 from Transaction\_Data

-- Create a new measure to calculate the AVG Retail Price

select CAST(AVG(product\_retail\_price) AS decimal(10,2)) AS AVG\_retail\_price from Products

-- OR I could have used the OVER() clause Window function to calculate the AVG Retail\_Price for all company

select \*, AVG(product\_retail\_price) OVER() as AVG\_retail\_price from Products

A screenshot of a computer

Description automatically generated

-- Calculate the Total Count of Transactions, but only for products > than the average price.

-- So, I want to see the # of transactions for product prices higher than Overall AVG priced items.

-- SCALAR UNCORRELATED SUBQUERY IN A WHERE CLAUSE

select \* from Transaction\_Data

select \* from products

select COUNT(transaction\_date) AS number\_of\_transactions from Transaction\_Data

right join Products on products.product\_id = Transaction\_Data.product\_id

WHERE products.product\_retail\_price >=

(select CAST(AVG(products.product\_retail\_price) AS decimal(10,2)) AS avg\_retail\_price from Products)

-- Calculate 'Total Revenue' based on transaction quantity and product retail price. I should see a total revenue of $1,764,546

select ROUND(SUM(t.quantity \* p.product\_retail\_price),0) from Transaction\_Data t

inner join Products p on p.product\_id = t.product\_id

-- I can also add a new column called Revenue to the Transaction\_data table and update it multiplying quantity by retail price

ALTER TABLE Transaction\_data

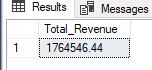
ADD Revenue DECIMAL(10,2)

UPDATE Transaction\_Data

SET Revenue = t.quantity \* p.product\_retail\_price from Transaction\_Data t

inner join Products p on p.product\_id = t.product\_id

select SUM(revenue) AS Total\_Revenue from Transaction\_Data



-- Calculate 'Total COGS' based on transaction quantity and product cost, and format. Spot check: I should see a total cost of $711,728

select ROUND(SUM(t.quantity \* p.product\_cost),0) from Transaction\_Data t

inner join Products p on p.product\_id = t.product\_id

-- I can also add a new column called COGS and update it multiplying quantity by product cost

ALTER TABLE Transaction\_data

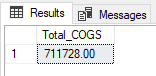
ADD COGS DECIMAL(10,2)

UPDATE Transaction\_Data

SET COGS = t.quantity \* p.product\_cost from Transaction\_Data t

inner join Products p on p.product\_id = t.product\_id

select ROUND(SUM(COGS),0) AS Total\_COGS from Transaction\_Data





-- Calculate 'Total Profit' to calculate total revenue minus total cost, and format as $. Spot check: I should see a total profit of $1,052,819

select ROUND(SUM(t.quantity \* p.product\_retail\_price) - SUM(t.quantity \* p.product\_cost),0) from Transaction\_Data t

inner join Products p on p.product\_id = t.product\_id

-- OR

select ROUND(SUM(Revenue) - SUM(COGS),0) AS 'Total\_Profit' from Transaction\_Data

-- I can also add a new column called Profit and update it subtracting COGS from Revenue

select \* from Transaction\_Data

ALTER TABLE transaction\_data

ADD Profit decimal(10,2)

UPDATE Transaction\_Data

SET Profit = (Revenue - COGS)

select ROUND(SUM(Profit),0) AS 'Total Profit' from Transaction\_Data

A screenshot of a computer

Description automatically generated

-- Calculate "Profit Margin" by dividing total profit by total revenue (format as %). Spot check: You should see an overall profit margin of 59.67%

select ROUND(SUM(Profit) / SUM(Revenue) \* 100,2) AS 'Profit Margin %' from Transaction\_Data

A screenshot of a computer screen

Description automatically generated

-- Report the ‘AVG Revenue Per Sales Customer'

select ROUND(SUM(Revenue) / COUNT(DISTINCT customer\_id),2) AS 'AVG Revenue Per Customer' from Transaction\_Data

A screenshot of a computer

Description automatically generated

-- Calculate the number of unique product names in the Products table. Spot check: I should see 1,560 unique products

select COUNT(DISTINCT product\_name) AS 'Unique Products' from Products

A screenshot of a computer

Description automatically generated