# Sql 2

DATA UNDERSTANDING

* This database contains details on cell phone sales or transactions. The data is organized into the following tables:
* Dim\_manufacturer: Contains details about the cell phone manufacturers.
* Dim\_model: Contains details about the different models of cell phones.
* Dim\_customer: Contains details about the customers who are buying the cell phones.
* Dim\_Location: Contains details about the locations where the transactions are taking place.
* Fact\_Transactions: This is the fact table that stores all the information about sales of specific cellphones.
* Each of these tables stores entries for their respective elements. The Fact\_Transactions table links these dimensions together and provides a comprehensive view of each transaction.

**The business objective of this case study is to analyze the sales data of a cellphone company to gain insights into customer behavior, product performance, and sales trends.**

**Business understanding based on question**

* **Identifying customer base:** By listing all the states where customers have bought cellphones since 2005, the company can understand its geographical reach and identify areas for potential expansion.
* **Understanding brand preference:** Identifying the state that buys the most ‘Samsung’ cell phones helps the company understand brand preferences in different regions. This can guide marketing and sales strategies.
* **Detailed transaction analysis:** Showing the number of transactions for each model per zip code per state can help identify popular models in specific areas. This can inform inventory management and targeted advertising.
* **Price competitiveness:** Identifying the cheapest cellphone can help the company understand its price competitiveness in the market.
* **Pricing strategy:** Finding the average price for each model from the top 5 manufacturers can help the company position its products effectively in the market.
* **Identifying high**-value customers: Listing customers who spent more than an average of 500 in 2009 can help the company identify high-value customers for targeted marketing campaigns.
* **Product consistency:** Identifying models that were in the top 5 in terms of quantity for three consecutive years can help the company understand product consistency and longevity in the market.
* **Competitor analysis:** Showing the manufacturer with the 2nd top sales in 2009 and 2010 can help the company understand its competition and market dynamics.
* **Market entry and exi**t: Identifying manufacturers that sold cellphones in 2010 but not in 2009 can help the company track market entries and exits, which is useful for understanding market trends and competition.
* C**ustomer loyalty and spending habits:** Finding the top 100 customers and their average spend and quantity each year, along with the percentage change in their spend, can help the company understand customer loyalty and spending habits. This can inform customer retention strategies.

## QUSIONS AND KEY ANSWER

1. **List all the states in which we have customers who have bought cellphones from 2005 till today.**

select f.IDCustomer ,l.State,d.YEAR from [dbo].[FACT\_TRANSACTIONS] as f

join [dbo].[DIM\_LOCATION] as l on l.IDLocation =f.IDLocation

join [dbo].[DIM\_DATE] as d on d.DATE = f.Date

where d.YEAR >= '2005'

order by d.YEAR

* The query selects the IDCustomer, State, and YEAR fields from the FACT\_TRANSACTIONS table (f), DIM\_LOCATION table (l), and DIM\_DATE table (d).
* It joins these tables on the IDLocation and Date fields.
* It then filters the results to include only transactions where the YEAR is 2005 or later.
* Finally, it orders the results by YEAR.
* So, the query returns a list of all the states in which you have customers who have bought cellphones from 2005 till today.

1. **What state in the US is buying the most 'Samsung' cell phones?**

select top 1 l.State, m.Manufacturer\_Name, sum(Quantity) qty from [dbo].[FACT\_TRANSACTIONS] f

join [dbo].[DIM\_LOCATION] l on l.IDLocation=f.IDLocation

join [dbo].[DIM\_MODEL] ml on ml.IDModel=f.IDModel

join [dbo].[DIM\_MANUFACTURER] m on m.IDManufacturer = ml.IDManufacturer

where m.Manufacturer\_Name = 'samsung' and l.Country = 'us'

group by l.State,m.Manufacturer\_Name

order by qty desc

* The query selects the State and Manufacturer\_Name fields from the FACT\_TRANSACTIONS table (f), DIM\_LOCATION table (l), DIM\_MODEL table (ml), and DIM\_MANUFACTURER table (m).
* It joins these tables on the IDLocation, IDModel, and IDManufacturer fields.
* It then filters the results to include only transactions where the Manufacturer\_Name is ‘Samsung’ and the Country is ‘US’.
* The results are grouped by the State and Manufacturer\_Name, and the sum of the Quantity for each state and manufacturer is calculated.
* The TOP 1 clause combined with the ORDER BY clause ensures that only the state with the highest sum of quantity is returned.

1. **Show the number of transactions for each model per zip code per state.**

select m.Model\_Name,l.ZipCode ,l.State,count(TotalPrice) [number of transaction] from [dbo].[FACT\_TRANSACTIONS] f

join [dbo].[DIM\_MODEL] m on m.IDModel = f.IDModel

join [dbo].[DIM\_LOCATION] l on l.IDLocation = f.IDLocation

group by m.Model\_Name,l.ZipCode ,l.State

order by [number of transaction]

* The query selects the Model\_Name, ZipCode, State, and the count of TotalPrice (which represents the number of transactions) from the FACT\_TRANSACTIONS table (f), DIM\_MODEL table (m), and DIM\_LOCATION table (l).
* It joins these tables on the IDModel and IDLocation fields.
* The results are grouped by the Model\_Name, ZipCode, and State.
* Finally, it orders the results by the number of transactions.
* So, the query returns the number of transactions for each model per zip code per state.

***4. Show the cheapest cellphone (Output should contain the price also)***

select top 1 mf.Manufacturer\_Name,m.Unit\_price from [dbo].[DIM\_MODEL] m

join [dbo].[DIM\_MANUFACTURER] mf on mf.IDManufacturer = m.IDManufacturer

order by Unit\_price asc

* select top 1 mf.Manufacturer\_Name, m.Unit\_price: This line is selecting the top 1 record (which means only one record) of the manufacturer’s name and the unit price of the model.
* from [dbo].[DIM\_MODEL] m: This line is specifying the table [dbo].[DIM\_MODEL] from which to select the records. The m after the table name is an alias for the [dbo].[DIM\_MODEL] table.
* join [dbo].[DIM\_MANUFACTURER] mf on mf.IDManufacturer = m.IDManufacturer: This line is joining the [dbo].[DIM\_MANUFACTURER] table with the [dbo].[DIM\_MODEL] table. The join is based on the condition that the IDManufacturer field in both tables must be equal. The mf is an alias for the [dbo].[DIM\_MANUFACTURER] table.
* order by Unit\_price asc: This line is ordering the records in ascending order based on the Unit\_price. Since we’re using top 1, it will return the record with the lowest unit price, which is the cheapest cellphone.
* So, in summary, this query returns the manufacturer’s name and the unit price of the cheapest cellphone in the database. The cheapest cellphone is determined by the Unit\_price field. If there are multiple cellphones with the same lowest price

**5.Find out the average price for each model in the top5 manufacturers in terms of sales quantity and order by average price.**

with mnf as (select top 5 mf.Manufacturer\_Name,sum(f.Quantity) s\_QTY from [dbo].[FACT\_TRANSACTIONS] f

join [dbo].[DIM\_MODEL] m on m.IDModel = f.IDModel

join [dbo].[DIM\_MANUFACTURER] mf on mf.IDManufacturer= m.IDManufacturer

group by mf.Manufacturer\_Name

)

select mn.Manufacturer\_Name,m.Model\_Name, avg(TotalPrice) [avg price] from [dbo].[FACT\_TRANSACTIONS] f

join [dbo].[DIM\_MODEL] m on m.IDModel = f.IDModel

join [dbo].[DIM\_MANUFACTURER] mf on mf.IDManufacturer = m.IDManufacturer

cross join mnf as mn

group by mn.Manufacturer\_Name,m.Model\_Name

order by [avg price]

* with mnf as (...): This is a common table expression (CTE) named mnf. It’s a temporary result set that is defined within the execution scope of a single SQL statement. This CTE is calculating the total quantity of sales (s\_QTY) for each manufacturer.
* select mn.Manufacturer\_Name, m.Model\_Name, avg(TotalPrice) [avg price]: This line is selecting the manufacturer’s name, the model’s name, and the average total price of the transactions.
* from [dbo].[FACT\_TRANSACTIONS] f join [dbo].[DIM\_MODEL] m on m.IDModel = f.IDModel join [dbo].[DIM\_MANUFACTURER] mf on mf.IDManufacturer = m.IDManufacturer: These lines are joining the FACT\_TRANSACTIONS, DIM\_MODEL, and DIM\_MANUFACTURER tables based on the IDModel and IDManufacturer fields.
* cross join mnf as mn: This line is performing a cross join with the mnf CTE. A cross join returns the Cartesian product of rows from both tables.
* group by mn.Manufacturer\_Name, m.Model\_Name: This line is grouping the results by the manufacturer’s name and the model’s name.
* order by [avg price]: This line is ordering the results by the average price in ascending order.

1. **List the names of the customers and the average amount spent in 2009, where the average is higher than 500**

select c.Customer\_Name,d.YEAR,avg(TotalPrice) [avg amt] from [dbo].[FACT\_TRANSACTIONS] f

join [dbo].[DIM\_CUSTOMER] c on c.IDCustomer = f.IDCustomer

join [dbo].[DIM\_DATE] d on d.DATE = f.Date

where d.YEAR = 2009

group by c.Customer\_Name,d.YEAR

having avg(TotalPrice)> 500

* select c.Customer\_Name, d.YEAR, avg(TotalPrice) [avg amt]: This line is selecting the customer’s name, the year, and the average total price of the transactions.
* from [dbo].[FACT\_TRANSACTIONS] f join [dbo].[DIM\_CUSTOMER] c on c.IDCustomer = f.IDCustomer join [dbo].[DIM\_DATE] d on d.DATE = f.Date: These lines are joining the FACT\_TRANSACTIONS, DIM\_CUSTOMER, and DIM\_DATE tables based on the IDCustomer and Date fields.
* where d.YEAR = 2009: This line is filtering the results to only include transactions from the year 2009.
* group by c.Customer\_Name, d.YEAR: This line is grouping the results by the customer’s name and the year.
* having avg(TotalPrice) > 500: This line is filtering the grouped results to only include groups where the average total price is greater than 500.
* So, in summary, this query returns the names of the customers and the average amount they spent in 2009, where the average amount spent is higher than 500.

1. **List if there is any model that was in the top 5 in terms of quantity, simultaneously in 2008, 2009 and 2010**

with a as

(select top 5 m.Model\_Name, sum([Quantity]) as 'total' from [dbo].[FACT\_TRANSACTIONS] t

join [dbo].[DIM\_MODEL] m on m.IDModel = t.IDModel

where year ([Date]) = 2008

group by m.Model\_Name

order by sum([Quantity]) desc),

b as

(select top 5 m.Model\_Name, sum([Quantity]) as 'total' from [dbo].[FACT\_TRANSACTIONS] t

join [dbo].[DIM\_MODEL] m on m.IDModel = t.IDModel

where year ([Date]) = 2009

group by m.Model\_Name

order by sum([Quantity]) desc),

c as

(select top 5 m.Model\_Name, sum([Quantity]) as 'total' from [dbo].[FACT\_TRANSACTIONS] t

join [dbo].[DIM\_MODEL] m on m.IDModel = t.IDModel

where year ([Date]) = 2010

group by m.Model\_Name

order by sum([Quantity]) desc)

select a.Model\_Name from a

inner join b on a.Model\_Name=b.Model\_Name

inner join c on a.Model\_Name=c.Model\_Name ;

* The query uses three common table expressions (CTEs) named a, b, and c. Each CTE calculates the top 5 models in terms of quantity sold for a specific year (2008, 2009, and 2010 respectively).
* In each CTE, select top 5 m.Model\_Name, sum([Quantity]) as 'total' selects the model name and the total quantity sold for that model.
* from [dbo].[FACT\_TRANSACTIONS] t join [dbo].[DIM\_MODEL] m on m.IDModel = t.IDModel joins the FACT\_TRANSACTIONS and DIM\_MODEL tables based on the IDModel field.
* where year ([Date]) = 2008 (or 2009 or 2010) filters the results to only include transactions from the specified year.
* group by m.Model\_Name groups the results by the model name.
* order by sum([Quantity]) desc orders the results by the total quantity sold in descending order, so the models with the highest quantities sold are at the top.
* The main query select a.Model\_Name from a inner join b on a.Model\_Name = b.Model\_Name inner join c on a.Model\_Name = c.Model\_Name then selects the model names that appear in all three CTEs, i.e., the models that were in the top 5 in terms of quantity sold in all three years.
* So, in summary, this query returns the models that were in the top 5 in terms of quantity sold, simultaneously in 2008, 2009, and 2010.

1. **Show the manufacturer with the 2nd top sales in the year of 2009 and the manufacturer with the 2nd top sales in the year of 2010.**

(select top 1 d1.IDManufacturer,m.Manufacturer\_Name,d1.sales from ( select d.[IDManufacturer],m.Manufacturer\_Name,da.YEAR,sum(TotalPrice) as sales from DIM\_MODEL d

inner join FACT\_TRANSACTIONS f on d.IDModel = f.IDModel

join DIM\_DATE da on da.DATE = f.Date

join [dbo].[DIM\_MANUFACTURER] m on m.IDManufacturer = d.IDManufacturer

where da.YEAR = 2009

group by d.IDManufacturer,m.Manufacturer\_Name,da.YEAR

order by sum(TotalPrice) desc offset 1 row )D1 join [dbo].[DIM\_MANUFACTURER] m on m.IDManufacturer = d1.IDManufacturer )

union

(select top 1 d1.IDManufacturer,m.Manufacturer\_Name,d1.sales from ( select d.[IDManufacturer],m.Manufacturer\_Name,da.YEAR,sum(TotalPrice) as sales from DIM\_MODEL d

inner join FACT\_TRANSACTIONS f on d.IDModel = f.IDModel

join DIM\_DATE da on da.DATE = f.Date

join [dbo].[DIM\_MANUFACTURER] m on m.IDManufacturer = d.IDManufacturer

where da.YEAR = 2010

group by d.IDManufacturer,m.Manufacturer\_Name,da.YEAR

order by sum(TotalPrice) desc offset 1 row )D1 join [dbo].[DIM\_MANUFACTURER] m on m.IDManufacturer = d1.IDManufacturer )

* The query uses two subqueries for the years 2009 and 2010. Each subquery calculates the 2nd top sales for that year.
* In each subquery, select d.[IDManufacturer], m.Manufacturer\_Name, da.YEAR, sum(TotalPrice) as sales selects the manufacturer ID, the manufacturer’s name, the year, and the total sales for that manufacturer.
* from DIM\_MODEL d inner join FACT\_TRANSACTIONS f on d.IDModel = f.IDModel join DIM\_DATE da on da.DATE = f.Date join [dbo].[DIM\_MANUFACTURER] m on m.IDManufacturer = d.IDManufacturer joins the DIM\_MODEL, FACT\_TRANSACTIONS, DIM\_DATE, and DIM\_MANUFACTURER tables based on the IDModel, Date, and IDManufacturer fields.
* where da.YEAR = 2009 (or 2010) filters the results to only include transactions from the specified year.
* group by d.IDManufacturer, m.Manufacturer\_Name, da.YEAR groups the results by the manufacturer ID, the manufacturer’s name, and the year.
* order by sum(TotalPrice) desc offset 1 row orders the results by the total sales in descending order, and then skips the first row (the top sales) to get the 2nd top sales.
* The main query then joins the results of the two subqueries on the manufacturer ID to get the manufacturers with the 2nd top sales in both 2009 and 2010.
* So, in summary, this query returns the manufacturers with the 2nd top sales in the years 2009 and 2010.

***9. Show the manufacturers that sold cellphones in 2010 but did not in 2009.***

select mf.Manufacturer\_Name,d.YEAR from [dbo].[FACT\_TRANSACTIONS] f

join [dbo].[DIM\_DATE] d on d.DATE = f.date

join [dbo].[DIM\_MODEL] m on m.IDModel= f.IDModel

join [dbo].[DIM\_MANUFACTURER] mf on mf.IDManufacturer = m.IDManufacturer

where not d.YEAR = 2009 and d.YEAR = 2010

group by mf.Manufacturer\_Name,d.YEAR

* select mf.Manufacturer\_Name, d.YEAR: This line is selecting the manufacturer’s name and the year.
* from [dbo].[FACT\_TRANSACTIONS] f join [dbo].[DIM\_DATE] d on d.DATE = f.date join [dbo].[DIM\_MODEL] m on m.IDModel= f.IDModel join [dbo].[DIM\_MANUFACTURER] mf on mf.IDManufacturer = m.IDManufacturer: These lines are joining the FACT\_TRANSACTIONS, DIM\_DATE, DIM\_MODEL, and DIM\_MANUFACTURER tables based on the date, IDModel, and IDManufacturer fields.
* where not d.YEAR = 2009 and d.YEAR = 2010: This line is filtering the results to only include transactions from the year 2010 and exclude transactions from the year 2009.
* group by mf.Manufacturer\_Name, d.YEAR: This line is grouping the results by the manufacturer’s name and the year.
* So, in summary, this query returns the manufacturers that sold cellphones in 2010 but did not in 2009.

***10..Find top 100 customers and their average spend, average quantity by each year. Also find the percentage of change in their spend.***

select [IDCustomer],

avg ([Quantity]) as avg\_qty,

YEAR(date) as year, avg(totalprice)as total\_price,

lag(sum(totalprice))over(partition by idcustomer order by year(date)) as lag1,

concat ( ((sum(totalprice) - lag(sum(totalprice))over (partition by idcustomer order by year(date))/

lag(sum(totalprice))over (partition by idcustomer order by year(date)))\*100),'%') as yoy from [dbo].[FACT\_TRANSACTIONS]

group by idcustomer,date

* select [IDCustomer], avg ([Quantity]) as avg\_qty, YEAR(date) as year, avg(totalprice) as total\_price: This line is selecting the customer ID, the average quantity, the year, and the average total price of the transactions.
* lag(sum(totalprice)) over(partition by idcustomer order by year(date)) as lag1: This line is calculating the total price for the previous year for each customer. The lag function is a window function that provides access to a row at a specified physical offset that comes before the current row.
* concat ( ((sum(totalprice) - lag(sum(totalprice)) over (partition by idcustomer order by year(date)) / lag(sum(totalprice)) over (partition by idcustomer order by year(date)))\*100),'%') as yoy: This line is calculating the year-over-year (YoY) percentage change in the total price for each customer.
* from [dbo].[FACT\_TRANSACTIONS]: This line is specifying the table [dbo].[FACT\_TRANSACTIONS] from which to select the records.
* group by idcustomer, date: This line is grouping the results by the customer ID and the date.
* So, in summary, this query returns the top 100 customers and their average spend, average quantity by each year, and the percentage of change in their spend. Please note that you might need to adjust the query based on your actual database schema and data. If you encounter any issues, feel free to ask for further assistance.