**OLTP database Diagram**

|  |
| --- |
|  |

**Data Warehouse Schema** (Star Schema was used)

|  |
| --- |
|  |

**Queries explanation**

|  |  |  |
| --- | --- | --- |
| Query | Explanation | SQL script |
| 1 | Business Question: Analyse Sales/Staff  This query shows information about all employees who are sales representatives in the company. It includes their first name, last name, hire date, the number of orders they have sold, the quantity of items they have sold, the revenue generated from their sales, and the profit generated from their sales. The results are sorted based on the profit generated, showing the most profitable sales representatives first.  Insights From Query:   * Freya Gomez is the most valuable employee. Even though she was hired later in November 2016, she has sold the most orders and items and gotten the most profit at 1.5 million. * Most Employees have sold 20-40 orders and 1000-4000 items with profits of 200k-700k * Sophia Reynolds is the least productive employee who, despite joining around the same time as others has gotten significantly lower orders and profits than other employees | SELECT  E.FIRST\_NAME,  E.LAST\_NAME,  E.HIRE\_DATE AS DATE\_HIRED,  COUNT(DISTINCT SF.ORDER\_KEY) AS NUMBER\_OF\_ORDERS,  SUM(SF.QUANTITY) AS QUANTITY\_SOLD,  FORMAT(SUM(SF.QUANTITY \* SF.UNIT\_PRICE), 'N2') AS SALES\_REVENUE,  FORMAT(ROUND(SUM((SF.UNIT\_PRICE - PD.STANDARD\_COST) \* SF.QUANTITY), 2), 'N2') AS PROFIT  FROM  SALES\_FACT SF  JOIN EMPLOYEE\_DIM E ON SF.EMPLOYEE\_KEY = E.EMPLOYEE\_KEY  JOIN PRODUCT\_DIM PD ON SF.PRODUCT\_KEY = PD.PRODUCT\_KEY  WHERE  E.JOB\_TITLE = 'Sales Representative'  GROUP BY  E.FIRST\_NAME,  E.LAST\_NAME,  E.HIRE\_DATE  ORDER BY CAST(REPLACE(FORMAT(ROUND(SUM((SF.UNIT\_PRICE - PD.STANDARD\_COST) \* SF.QUANTITY), 2), 'N2'), ',', '') AS DECIMAL(18, 2)) DESC; |
| 2 | Business Question: Find Trends in Time Series of Sales  This query presents the quarterly growth of total profit and profit for each product category. It calculates the raw number of orders and profit for each product category in each quarter, allowing the analysis of quarterly sales growth. The results are shown in a tabular format with columns for year, quarter, total profit growth, growth for each product category (CPU, Motherboard, Storage, Video Card), total number of orders, and profit for each product category.  Insights:   * Total profit of the company saw big increases in 2015 Q2 and 2016 Q2 * Most of this growth was fueled by growth in CPU and Storage sales profits. They are the most profitable ones overall * All product categories saw a dip in profits in 2016 Q1 but quicly rebounded in the next quarter * 2016 Q3, 2016 Q4 and 2017 Q1 are the most profitable quarters overall | WITH base\_query AS (  SELECT  T.TIME\_YEAR AS Year,  T.TIME\_QUARTER AS Quarter,  COUNT(S.ORDER\_KEY) AS Total\_No\_Of\_Orders,  SUM((S.UNIT\_PRICE - P.STANDARD\_COST) \* S.QUANTITY) AS Total\_Profit,  COUNT(DISTINCT CASE WHEN P.CATEGORY\_NAME = 'CPU' THEN S.ORDER\_KEY END) AS No\_Of\_CPU\_Orders,  SUM(CASE WHEN P.CATEGORY\_NAME = 'CPU' THEN (S.UNIT\_PRICE - P.STANDARD\_COST) \* S.QUANTITY ELSE 0 END) AS CPU\_Profit,  COUNT(DISTINCT CASE WHEN P.CATEGORY\_NAME = 'Mother Board' THEN S.ORDER\_KEY END) AS No\_Of\_Motherboard\_Orders,  SUM(CASE WHEN P.CATEGORY\_NAME = 'Mother Board' THEN (S.UNIT\_PRICE - P.STANDARD\_COST) \* S.QUANTITY ELSE 0 END) AS Motherboard\_Profit,  COUNT(DISTINCT CASE WHEN P.CATEGORY\_NAME = 'Storage' THEN S.ORDER\_KEY END) AS No\_Of\_Storage\_Orders,  SUM(CASE WHEN P.CATEGORY\_NAME = 'Storage' THEN (S.UNIT\_PRICE - P.STANDARD\_COST) \* S.QUANTITY ELSE 0 END) AS Storage\_Profit,  COUNT(DISTINCT CASE WHEN P.CATEGORY\_NAME = 'Video Card' THEN S.ORDER\_KEY END) AS No\_Of\_Video\_Card\_Orders,  SUM(CASE WHEN P.CATEGORY\_NAME = 'Video Card' THEN (S.UNIT\_PRICE - P.STANDARD\_COST) \* S.QUANTITY ELSE 0 END) AS Video\_Card\_Profit  FROM  SALES\_FACT S  JOIN  TIME\_DIM T ON S.TIME\_KEY = T.TIME\_KEY  JOIN  PRODUCT\_DIM P ON S.PRODUCT\_KEY = P.PRODUCT\_ID  GROUP BY  T.TIME\_YEAR,  T.TIME\_QUARTER  )  SELECT  Year,  Quarter,  CONCAT(FORMAT(ROUND(((Total\_Profit - LAG(Total\_Profit, 1) OVER (ORDER BY Year, Quarter)) / NULLIF(LAG(Total\_Profit, 1) OVER (ORDER BY Year, Quarter), 0)) \* 100, 2), 'N2'), '%') AS Total\_Profit\_Growth,  CONCAT(FORMAT(ROUND(((CPU\_Profit - LAG(CPU\_Profit, 1) OVER (ORDER BY Year, Quarter)) / NULLIF(LAG(CPU\_Profit, 1) OVER (ORDER BY Year, Quarter), 0)) \* 100, 2), 'N2'), '%') AS CPU\_Profit\_Growth,  CONCAT(FORMAT(ROUND(((Motherboard\_Profit - LAG(Motherboard\_Profit, 1) OVER (ORDER BY Year, Quarter)) / NULLIF(LAG(Motherboard\_Profit, 1) OVER (ORDER BY Year, Quarter), 0)) \* 100, 2), 'N2'), '%') AS Motherboard\_Profit\_Growth,  CONCAT(FORMAT(ROUND(((Storage\_Profit - LAG(Storage\_Profit, 1) OVER (ORDER BY Year, Quarter)) / NULLIF(LAG(Storage\_Profit, 1) OVER (ORDER BY Year, Quarter), 0)) \* 100, 2), 'N2'), '%') AS Storage\_Profit\_Growth,  CONCAT(FORMAT(ROUND(((Video\_Card\_Profit - LAG(Video\_Card\_Profit, 1) OVER (ORDER BY Year, Quarter)) / NULLIF(LAG(Video\_Card\_Profit, 1) OVER (ORDER BY Year, Quarter), 0)) \* 100, 2), 'N2'), '%') AS Video\_Card\_Profit\_Growth,  Total\_No\_Of\_Orders,  CONCAT('$', FORMAT(ROUND(Total\_Profit, 2), 'N2')) AS Total\_Profit,  No\_Of\_CPU\_Orders,  CONCAT('$', FORMAT(ROUND(CPU\_Profit, 2), 'N2')) AS CPU\_Profit,  No\_Of\_Motherboard\_Orders,  CONCAT('$', FORMAT(ROUND(Motherboard\_Profit, 2), 'N2')) AS Motherboard\_Profit,  No\_Of\_Storage\_Orders,  CONCAT('$', FORMAT(ROUND(Storage\_Profit, 2), 'N2')) AS Storage\_Profit,  No\_Of\_Video\_Card\_Orders,  CONCAT('$', FORMAT(ROUND(Video\_Card\_Profit, 2), 'N2')) AS Video\_Card\_Profit  FROM  base\_query  ORDER BY  Year,  Quarter; |
| 3 | Business Question: Analyse Sales/Orders/Products  This query provides insights into quarterly order trends for the company. It shows the number of orders in each quarter, the average number of products in each order, and the percentage of pending and shipped orders for each quarter. Additionally, it identifies the most popular product (highest quantity sold) for each quarter. The results are sorted by year and quarter.  Insights:   * Average number of products in a order has been increasing over the years * 2016 Q4 and 2017 Q1 saw the most orders at 22 each * All the orders from 2013 has been shipped * In 2016 Q2 almost half of the orders were cancelled * Most of the orders from 2015 Q4 are still pending which is concerning as it is quite long ago | USE CompAccInc\_DWRam;  WITH QuarterOrders AS (  SELECT  T.TIME\_YEAR,  T.TIME\_QUARTER,  COUNT(DISTINCT O.ORDER\_ID) AS Orders\_In\_Quarter  FROM ORDERS\_DIM O  JOIN TIME\_DIM T ON O.ORDER\_DATE = CAST(T.TIME\_KEY AS DATE)  GROUP BY T.TIME\_YEAR, T.TIME\_QUARTER  ),  QuarterProducts AS (  SELECT  T.TIME\_YEAR,  T.TIME\_QUARTER,  SUM(S.QUANTITY) / COUNT(DISTINCT O.ORDER\_ID) AS Avg\_Products\_Per\_Order  FROM ORDERS\_DIM O  JOIN SALES\_FACT S ON O.ORDER\_ID = S.ORDER\_KEY  JOIN PRODUCT\_DIM P ON S.PRODUCT\_KEY = P.PRODUCT\_KEY  JOIN TIME\_DIM T ON O.ORDER\_DATE = CAST(T.TIME\_KEY AS DATE)  GROUP BY T.TIME\_YEAR, T.TIME\_QUARTER  ),  QuarterStatus AS (  SELECT  T.TIME\_YEAR,  T.TIME\_QUARTER,  ROUND(CAST(SUM(CASE WHEN O.ORDER\_STATUS = 'Pending' THEN 1 ELSE 0 END) AS FLOAT) / COUNT(\*) \* 100,0) AS Pending\_Percentage,  ROUND(CAST(SUM(CASE WHEN O.ORDER\_STATUS = 'Canceled' THEN 1 ELSE 0 END) AS FLOAT) / COUNT(\*) \* 100,0) AS Cancel\_Percentage,  ROUND(CAST(SUM(CASE WHEN O.ORDER\_STATUS = 'Shipped' THEN 1 ELSE 0 END) AS FLOAT) / COUNT(\*) \* 100, 0) AS Shipped\_Percentage  FROM ORDERS\_DIM O  JOIN TIME\_DIM T ON O.ORDER\_DATE = CAST(T.TIME\_KEY AS DATE)  GROUP BY T.TIME\_YEAR, T.TIME\_QUARTER  ),  QuarterProductSales AS (  SELECT  T.TIME\_YEAR,  T.TIME\_QUARTER,  P.PRODUCT\_NAME,  SUM(S.QUANTITY) AS Total\_Sold,  ROW\_NUMBER() OVER (PARTITION BY T.TIME\_YEAR, T.TIME\_QUARTER ORDER BY SUM(S.QUANTITY) DESC) AS PopularityRank  FROM SALES\_FACT S  JOIN PRODUCT\_DIM P ON S.PRODUCT\_KEY = P.PRODUCT\_KEY  JOIN TIME\_DIM T ON S.TIME\_KEY = T.TIME\_Key  GROUP BY T.TIME\_YEAR, T.TIME\_QUARTER, P.PRODUCT\_NAME  )  SELECT  QO.TIME\_YEAR,  'Q' + CAST(QO.TIME\_QUARTER AS VARCHAR) AS Quarter,  QO.Orders\_In\_Quarter,  QP.Avg\_Products\_Per\_Order,  QS.Pending\_Percentage,  QS.Shipped\_Percentage,  QS.Cancel\_Percentage,  QPS.PRODUCT\_NAME AS Most\_Popular\_Product,  QPS.Total\_Sold AS Most\_Popular\_Product\_Sales  FROM QuarterOrders QO  JOIN QuarterProducts QP ON QO.TIME\_YEAR = QP.TIME\_YEAR AND QO.TIME\_QUARTER = QP.TIME\_QUARTER  JOIN QuarterStatus QS ON QO.TIME\_YEAR = QS.TIME\_YEAR AND QO.TIME\_QUARTER = QS.TIME\_QUARTER  JOIN QuarterProductSales QPS ON QO.TIME\_YEAR = QPS.TIME\_YEAR AND QO.TIME\_QUARTER = QPS.TIME\_QUARTER AND QPS.PopularityRank = 1  ORDER BY QO.TIME\_YEAR, QO.TIME\_QUARTER; |
| 4 | Business Question: Analyse Sales/Customers/Products  This query lists all customers of the company, along with their most valuable product category, most valuable product, and the total profit they have generated for the company. It also calculates the percentage of margin contributed by the most valuable product and product category to the overall profit for each customer. The results are sorted in descending order based on the total profit margin.  Insights:   * General Mills is our most valuable customer who has generated over 700k in profit for us. He buys mostly video cards (41%) but his most bought item is Intel Xeon CPU (11%) * For many customers theur most valuable product is Intel Xeon which means it is well loved by the market * For our high value customers they are usually buying CPU, Storage or Video Cards the most, never RAM or Motherboard * As customers buy more items from us, they tend to buy more diverse items | with customer\_order as (  select  c.CUSTOMER\_NAME as cust\_name,  s.QUANTITY \* (s.UNIT\_PRICE - p.STANDARD\_COST) as total\_margin,  p.CATEGORY\_NAME as cat\_name,  p.PRODUCT\_NAME as prod\_name  from CUSTOMER\_DIM c, SALES\_FACT s, PRODUCT\_DIM p, orderS\_DIM o  where c.CUSTOMER\_KEY = s.CUSTOMER\_KEY and p.PRODUCT\_KEY = s.PRODUCT\_KEY and o.order\_key = s.order\_key  ),  customer\_cat\_ranked as (  select  cust\_name,  cat\_name,  sum(total\_margin) as highest\_cat\_margin,  rank() over (partition by cust\_name order by sum(total\_margin) desc) as cat\_ranking  from customer\_order  group by cust\_name, cat\_name  ),  customer\_prod\_ranked as (  select  cust\_name,  prod\_name,  sum(total\_margin) as highest\_prod\_margin,  rank() over (partition by cust\_name order by sum(total\_margin) desc) as prod\_ranking  from customer\_order  group by cust\_name, prod\_name  ),  customer\_total\_margin as (  select cust\_name, sum(total\_margin) as overall\_margins  from customer\_order  group by cust\_name  )  select  ccr.cust\_name,  ccr.cat\_name as 'Most valuable category',  ccr.highest\_cat\_margin / ctm.overall\_margins \* 100 as '% of margin of most valuable category',  cpr.prod\_name as 'Most valuable product',  cpr.highest\_prod\_margin / ctm.overall\_margins \* 100 as '% of margin of most valuable product',  ctm.overall\_margins as 'Total profit margin'  from customer\_cat\_ranked ccr, customer\_total\_margin ctm, customer\_prod\_ranked cpr  where ccr.cat\_ranking = 1 and cpr.prod\_ranking = 1 and ccr.cust\_name = ctm.cust\_name and cpr.cust\_name = ctm.cust\_name  order by 'Total profit margin' DESC; |
| 5 | Business Question: Analyse stock of components in relation to sales  This query presents information about the stock of each product category. It classifies the stock as high, low, or medium and calculates the average profit and sales for each product category, also classifying them as high, low, or medium profit/sales. The results are grouped by the product category and display the average stock status, average profit, and average sales status.  Insights:   * Video Cards are the most items in stock followed by storage. Despite having medium sales it has high profits. We should stock this a lot to deal with demand. * CPU, despite being medium stock, it has high sales and high profit which is good. We should stock this up so we can make more profit. * Motherboard and Storage, have low to medium stock but less profit than sales. So for these items we are selling a lot but not making high profits. | USE CompAccInc\_DWRam;  with wrangling as (  select  p.CATEGORY\_NAME,  avg(coalesce(p.STOCK, 0)) AS stock,  sum(coalesce(s.QUANTITY, 0) \* (coalesce(s.UNIT\_PRICE, 0) - p.STANDARD\_COST)) as profit\_margin,  sum(coalesce(s.QUANTITY, 0)) as sales  FROM  PRODUCT\_DIM p  LEFT JOIN  SALES\_FACT s ON p.PRODUCT\_KEY = s.PRODUCT\_KEY  group by p.PRODUCT\_NAME, p.PRODUCT\_ID, p.CATEGORY\_NAME  )  select  CATEGORY\_NAME,  avg(stock) as 'Average stock',  case when avg(stock) < 200 then 'Low stock' when avg(stock) < 500 then 'Medium stock' else 'High stock' end as 'Average stock status',  avg(profit\_margin) as 'Average profit',  case when avg(profit\_margin) < 20000 then 'Low profit' when avg(profit\_margin) < 50000 then 'Medium profit' else 'High profit' end as 'Average profit status',  avg(sales) as 'Average sales',  case when avg(sales) < 100 then 'Low sales' when avg(sales) < 200 then 'Medium sales' else 'High sales' end as 'Average sales status'  from wrangling  group by CATEGORY\_NAME; |