GROUP REPORT

GROUP MEMBERS:

ADITYA DATTATRAY KINGARE : 20025807

AKSHAY THOZHUKKATIL CHANDRASEKARAN : 20028929

GURU PRAKASH UPPILIAPPAN : 20023844

DUBLIN BUSINESS SCHOOL

1. Introduction

The primary objective of this report is to build a detailed data warehousing solution for the analysis of sales and shipping data. The project involves preparing a dimensional model, production of the Extract, Transform, and Load processes, creation of detailed reports, and comparing the performative differences between the relational and graph databases. The data warehouse is a subject-oriented, integrated, time-variant collection of data in support of management's decision-making process

2. Vision and Business Requirements

Business Vision:

Project Vision: Build an integrated data warehouse to unify sales, shipping, customer, and product data. The warehouse should deliver a single platform to run in-depth reporting and analytics. The information will be used to boost customer satisfaction, inventory management, and shipping operations. This corporation can take strategic decisions from this centralized repository of data that improves work efficiencies and ways to increase its overall performance.

Business Requirements:

Some of the business requirements are noted down below:

Data Integration: Combining various data data sources under a unitary repository in single form and in a consistent manner.

Accurate Reporting: Reports on sales performance should be made correctly and timely.

Customer Analysis: sales behavior of the customer is studied, and customers are grouped into different segments for making targeted marketing strategy.

Shipping Monitoring: Various shipping efficiencies can be monitored, and risk can be estimated in the case of late deliveries.

Database Performance Comparison: One can assess the performance of two types of databases it is feasible to compare the performance of two types of databases between complex query execution time.

3. Data Warehouse Design

Schema Design:

The data warehouse follows a star schema design. This design has a single central fact table called "SalesFact_source" and several dimension tables around that. It is quite comfortable to process many significantly affect performance queries

Fact Table:

SalesFact_source

Columns: Product_Id, Shipping_Id, Department_Id, Order_Id, Customer_Id, Sales, Order_Item_Discount, Order_Item_Quantity, Order_Item_Total, Order_Profit_Per_Order, Benefit_per_order, Sales_per_customer, Late_delivery_risk, Delivery_Status, Sales_Id

Purpose: This is where we store all the important

Columns: Customer_Id, Customer_Fname, Customer_Lname, Customer_Email, Customer_Password, Customer_Segment, Customer_Street, Customer_City, Customer_State, Customer_Zipcode, Customer_Country
Purpose: to maintain the customer in-depth file for segmentation and analysis.
Shipping Dimension (Shipping_source)
Columns: Shipping_Id, Shipping_Mode, Days_for_shipping_real, Days_for_shipment_scheduled
Purpose: PROCUREMENT shipping detail and calculated performance METRIC
Product Dimension (Product_source)
Columns: Product_Id
Columns : Department_Id, \ Department_Name,\ Latitude,\ Longitude.
Objective : Departmental business description to have an understanding and analyzing performance of department.
Design Justification
Star schema is selected among the others because it is simple in understanding. \Also, it is easy to work with the queries. One-to-one joins between fact table and \dimension table are very much needed when the analytical report and graphs are to be drawn.

4. Implementation of Data Warehouse

Data Source:

The datasets were collected from Kaggle. Raw data consists of various CSV files with information about customers, products, orders, shipping, and department details.

Data Cleaning:

Data Cleaning was executed with the help of Python. The processes followed included standardizing the data format, missing data treatment, inconsistency treatment, and ensuring improved data quality. This is a critical stage of data preparation for loading into a data warehouse for zero defect and great value.

Exploratory Data Analysis (EDA):

EDA was conducted to present data distributions, identify patterns, and detect possible anomalies. Analysis that provided light about the structure of the data was used to guide the schema that defines the data warehouse.

Splitting Tables:

Data has been organized as dimension tables and a fact table based upon the columns and primary key. It contains descriptive attributes under each dimension, whereas the fact table is the measurable business processes.

Creation of the Diagram:

Created ER diagrams and other visual schema representations using Draw.io to reflect the relationship between the tables.
This was important to grasp the right understanding of the data flows inside the warehouse.

SSMS Implementation:

Each of the implementation was done using the SQL Server Management Studio (SSMS).

Server Connection
Connected with the server.
Table Creation:

Defined and created source tables with the imported CSV files.
Defined and created destination tables for the dimensions.
Defined and created destination tables for the fact tables.
Data Import: Flat files imported for each table in such a way as to maintain schema consistency with all the destination tables in the process.
General Database Diagram: A general database diagram was created to help visualize the data model as a whole and the relationships within .
SSIS for ETL Process
Using Visual Studio and SQL Server Integration Services (SSIS):
Design Control Flow: Designed a control flow in SSIS to outline the sequence of data loading and transformation tasks.
Data Flow Tasks: Developed separate data flows for the source and destination tables of all dimension and fact tables that consisted of the following components:
OLE DB Source and Destination. Opened connections to the source and destination tables,
Transformations: Some of the data transformations enforced in the flow loads are described as lookups, conditional splits, data conversion, and custom SQL scripts
Data Population: Loaded transformed and cleansed data to the destination tables so that consistent and correct data is maintained.
General Steps Applied to All Tables
The above process was followed systematically in all dimension tables as well as the fact table to have a complete and integrated data warehouse architecture.

5. Reporting and Visualization
SSRS Reports
Four unique reports were developed with the use of SQL Server Reporting Services (SSRS) catering to the following needs:
Sales Performance Report: Gave insights about the total as well as the average sales of the firm and the trends in sales over time.
Customer Segmentation Report: Segregated customers on a three-dimensional basis, i.e., geographical location, type of segment, and purchasing behavior.
Product Performance Report: Best-selling products, product categories, and sales distribution.
Efficiency in Shipping Report: Vessels' mode compared with the stipulated delivery time of cargo and/or petroleum products, while at the same time identifying the risks related to late delivery.
Tableau Visualizations
Four visualizations in Tableau have been done for interactive analysis:

Sales Dashboard: This will give an overall glance of the most important sales KPIs, which include values for total sales, sales by region, and trends.
Customer Insights: Charting the trend of customers in different regions.
Product Analysis: Gave performance of the displayed product, sales visualization by category, and pinning point main products.
Shipping Analysis: Displayed shipping performance, delivery status finally after delivering the analysis of shipping modes.
These visualizations were added to the Tableau dashboard to enable an all-in-one view of the business and, thus, data-driven decisions.

6. Compare and Contrast Relational and Graph Databases

Implementation

This project was based upon comparing the performance of a traditional relational and graph database: SQL Server and Neo4J. A handful of queries were run both in SQL and Cypher Query Language to showcase the data handling in both technologies side by side.

Sample Queries and ComparisonA

Querying Sales Data

SQL: SELECT * FROM SalesFact WHERE Sales > 1000

CQL: MATCH (s:Sales) WHERE s.Sales > 1000 RETURN s

Customer Segmentation

SQL: SELECT Customer_Segment, COUNT(*) FROM Customer GROUP BY Customer_Segment

CQL: MATCH (c:Customer) RETURN c.Customer_Segment, COUNT(c)

Product Sales:

SQL: SELECT Product_Name, SUM(Sales) FROM
Product INNER JOIN SalesFact ON Product.Product_Id = SalesFact.Product_Id GROUP BY Product_Name

CQL: MATCH (p:Product)
-[:SOLD]->(s:Sales)
RETURN p.Product_Name, SUM(s.Sales)

Performance and Flexibility Comparison

Performance: Neo4J was better at complex queries for relationship and connections, while SQL Server was better at aggregations and simple queries.
Flexibility: Neo4J was better for flexibility in schema design, so making.
Complexity: SQL Server offered a familiar SQL syntax, making life a little easier for those who already knew SQL, while Neo4J required learning CQL but at the same time provided the most powerful querying capabilities out there for connected data.