14/11/24

Hemanth Kumar
192224238
CSA0593

Assignment - 1

1) Data Warehouse for a Retail chain

Design a data warehouse for a Retail chain that Consolidates data from Multiple store databases.

1) Creating Dimension & fact Tables:-

We'll design the schema for our data warehouse with appropriate dimension and fact tables.

- Product Dimensions:

```
CREATE TABLE DimproductC
    ProductID INT PRIMARY KEY,
    productName VARCHAR(100),
    category VARCHAR (50),
    Brand VARCHAR(50),
    Price DECIMAL(10,2)
);
```

- location Dimensions:

```
CREATE TABLE DimTimeC
    TimeID INT PRIMARY KEY,
    Date DATE,
    Dayofweek VARCHAR(10),
    Month INT,
    Quarter INT,
    year INT,
```

- Customer Dimensions:-

```
CREATE TABLE DimCustomerC
    customerID INT PRIMARY KEY,
    customerName VARCHAR(100),
```

```
    Gender VARCHAR(10),
    Age INT,
    Email VARCHAR(100)
);
```

2) Implementing ETL Processes:-

The ETL process Involves extracting data from different store databases, Transforming it into the desired format, and Loading it into the data warehouse.

→ Extract:-

```
SELECT * FROM StoreDB.product;
```

→ Transform:-

This can be handled using an ETL tool

For example, ensuring consistency in data formats, deduplication, and data cleaning.

→ Load:-

```
INSERT INTO Dimproduct (ProductID, ProductName, category, Brand, Price)
VALUES (?, ?, ?, ?, ?);
```

3) Designing SQL Queries for Analytical Reports.

Here are some example Queries to generate the analytical Reports:-

- Sales Trends by Location:-

```
SELECT
    l.city,
    t.year,
    t.Month,
    SUM(s.TotalRevenue) AS MonthlyRevenue
FROM
    Factsales s
    JOIN DimLocation l ON s.LocationID = l.LocationID
```

```
JOIN DimTime t ON S.TimeID = t.TimeID
GROUP BY
  l.City, t.Year, t.Month
ORDER BY
  t.Year, t.Month, l.City;
```

- Top-selling Products by Season:

```
SELECT
  p.ProductName,
  t.Quarter,
  t.Year,
  SUM(S.QuantitySold) AS TotalSold
FROM
  FactSales S
  JOIN DimProduct P ON S.ProductID = P.ProductID
  JOIN DimTime t ON S.TimeID = t.TimeID.
GROUP BY
  p.ProductName, t.Quarter, t.Year.
ORDER BY
  t.Year, t.Quarter, TotalSold DESC;
```

4) Using Indexing & Partitioning:-

To optimize query performance on large datasets, You can use Indexing and partitioning.

- Indexing Example:

```
CREATE INDEX idx-product-name ON DimProduct(ProductName);
CREATE INDEX idx-time-date ON DimTime(Date);
```

- Partitioning Example:-

Partition the sales data by Year for better query performance.

```
CREATE TABLE Factsalespartitioned (
    SalesID INT,
    productID INT,
    TimeID INT,
    LocationID INT,
    customer ID INT,
    QuantitySold INT,
    TotalRevenue DECIMAL(10,2)
)
PARTITION BY RANGE (Year) (
        PARTITION p2023 VALUES LESS THAN (2024),
        PARTITION p2024 VALUES LESS THAN (2025),
);
```