KATHMANDU UNIVERSITY

Department of Artificial Intelligence

Dhulikhel, Kavre

Lab Report On: Data Warehousing and OLAP

Submitted By:

Name: Aaryan Shakya

Roll No: 20

Subject Code: [AICC 301]

Submitted To:

Mr. Sunil Regmi

Lecturer, Department of Artificial Intelligence

6th July 2025

Objective

1. Understand and design a basic data warehouse for efficient data storage and integration.
2. Perform OLAP operations like roll-up, drill-down, slice, and dice on data cubes for multidimensional analysis.
3. Learn the ELT process to extract, load, and transform data for effective data processing in warehousing.

Problem Statement

A retail company wants to analyze its sales data based on three dimensions: Product, Time, and Location. The main measure to analyze is the Total Sales Amount. The tasks include designing a star schema data warehouse, populating it with sample data, building and querying a data cube using SQL, and performing OLAP operations for detailed sales analysis.

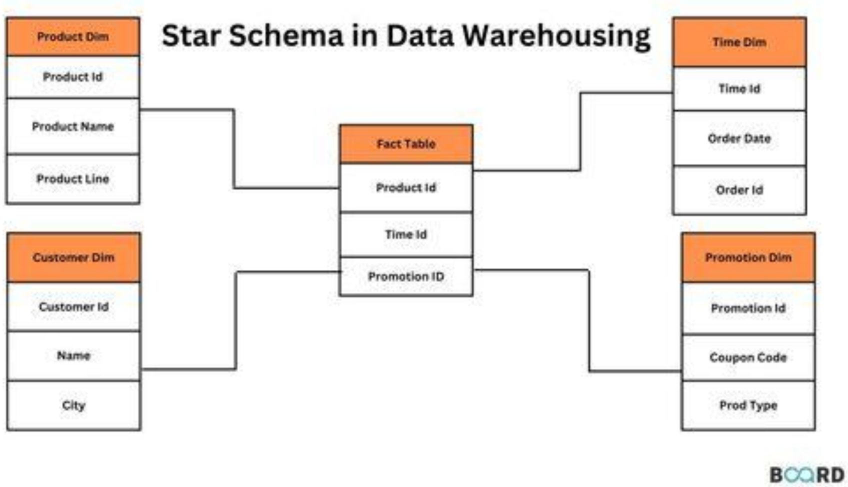
Introduction

Data Warehouse:

A data warehouse is a centralized repository designed to store large volumes of integrated data collected from multiple heterogeneous sources. It supports querying and analysis by organizing data in a way optimized for reporting and decision-making rather than transaction processing. Data warehouses enable businesses to consolidate historical data, ensuring consistency and improving data quality for comprehensive analysis.

Star Schema:

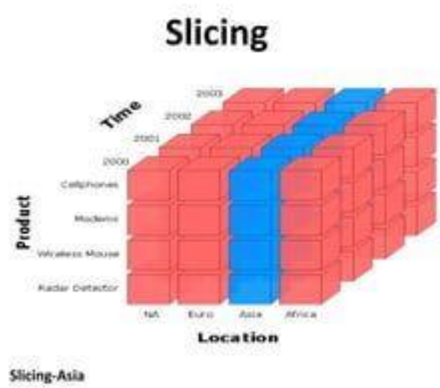
The star schema is a popular data modeling technique used in data warehouses. It consists of one central fact table that contains quantitative data (measures) such as sales amounts, linked to multiple dimension tables that describe the attributes related to these facts, such as Product, Time, and Location. The simple, denormalized structure of star schemas enables efficient querying and fast retrieval of aggregated data for analysis.



Data Cube:
A data cube is a multi-dimensional array of values that allows data to be modeled and viewed in multiple dimensions simultaneously. It organizes data so that users can quickly summarize and analyze measures across various dimensions, such as viewing total sales by product, location, and time periods. Data cubes facilitate rapid access to aggregated data, making them foundational for OLAP systems.

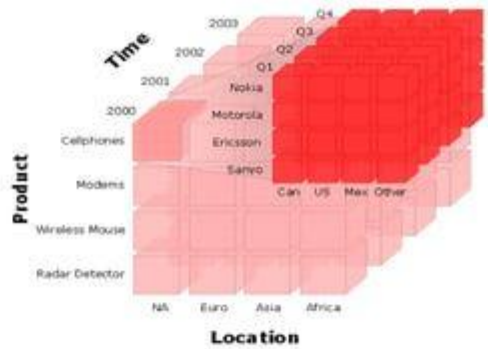
OLAP Operations:
OLAP (Online Analytical Processing) provides interactive analysis of multidimensional data using various operations:

- **Roll-up:** This operation aggregates data by climbing up a hierarchy within a dimension. For example, it can summarize daily sales data into monthly or yearly totals, reducing detail to see broader trends.
- **Drill-down:** The reverse of roll-up, drill-down navigates from summarized data to more detailed data, such as breaking down yearly sales into quarterly or monthly figures to uncover specific insights.
- **Slice:** This filters the data cube by selecting a single value from one dimension, effectively creating a sub-cube. For example, slicing by a specific year to analyze sales only during that year.



- **Dice:** This selects specific values from two or more dimensions to create a smaller cube, enabling focused analysis, such as examining sales of specific products in certain locations during particular time periods.

Dicing


Time
2003
2002
2001
2000
Q4
Q3
Q2
Q1
Nokia
Motorola
Ericsson
Sanyo
Product
Cellphones
Modems
Wireless Mouse
Radar Detector
Can
US
Mex
Other
NA
Euro
Asia
Africa
Location

Experiment

To create a star schema in PostgreSQL, populate the tables with sample data, and perform SQL queries to analyze sales data using OLAP operations.

Step 1: Create the schema in PostgreSQL

```
CREATE TABLE dim_product (
   product_id INT PRIMARY KEY,
   product_name VARCHAR(50)
);

CREATE TABLE dim_location (
   location_id INT PRIMARY KEY,
   location_name VARCHAR(50)
);

CREATE TABLE dim_time (
   time_id INT PRIMARY KEY,
   month VARCHAR(20),
   year INT
);

CREATE TABLE fact_sales (
   product_id INT,
   location_id INT,
   time_id INT,
   amount DECIMAL(10,2),
   FOREIGN KEY (product_id) REFERENCES dim_product(product_id),
   FOREIGN KEY (location_id) REFERENCES dim_location(location_id),
   FOREIGN KEY (time_id) REFERENCES dim_time(time_id)
);
```

Step 2: Populate tables with sample data

```
-- dim_product
INSERT INTO dim_product VALUES
(1, 'Laptop'),
(2, 'Smartphone'),
(3, 'Tablet'),
(4, 'Headphones'),
(5, 'Camera');

-- dim_location
INSERT INTO dim_location VALUES
```

```
(1, 'Kathmandu'),
(2, 'Pokhara'),
(3, 'Biratnagar'),
(4, 'Lalitpur'),
(5, 'Butwal');

-- dim_time
INSERT INTO dim_time VALUES
(1, 'January', 2025),
(2, 'February', 2025),
(3, 'March', 2025),
(4, 'April', 2025),
(5, 'May', 2025);

-- fact_sales
INSERT INTO fact_sales VALUES
(1, 1, 1, 1500.00),
(2, 2, 2, 2000.00),
(3, 3, 3, 1200.00),
(4, 4, 4, 800.00),
(5, 5, 5, 950.00);
```

Step 3: Write SQL queries for analysis

- **Computing total sales by product**

```
SELECT p.product_name, SUM(f.amount) AS total_sales
FROM fact_sales f
JOIN dim_product p ON f.product_id = p.product_id
GROUP BY p.product_name
ORDER BY total_sales DESC;
```

- **Perform a roll-up by year**

```
SELECT t.year, SUM(f.amount) AS total_sales
FROM fact_sales f
JOIN dim_time t ON f.time_id = t.time_id
GROUP BY t.year
ORDER BY t.year;
```

- **Slice by a specific location (e.g., Kathmandu)**

```
SELECT p.product_name, t.month, t.year, f.amount
FROM fact_sales f
JOIN dim_product p ON f.product_id = p.product_id
JOIN dim_time t ON f.time_id = t.time_id
```

```
JOIN dim_location l ON f.location_id = l.location_id
WHERE l.location_name = 'Kathmandu';
```

- **Dice for a specific product and year (e.g., Laptop and 2025)**

```
SELECT l.location_name, t.month, SUM(f.amount) AS total_sales
FROM fact_sales f
JOIN dim_product p ON f.product_id = p.product_id
JOIN dim_time t ON f.time_id = t.time_id
JOIN dim_location l ON f.location_id = l.location_id
WHERE p.product_name = 'Laptop' AND t.year = 2025
GROUP BY l.location_name, t.month
ORDER BY l.location_name, t.month;
```

Conclusion

In this experiment, we successfully designed and implemented a basic data warehouse using the star schema model, representing sales data across three key dimensions: Product, Time, and Location. By creating and populating dimension and fact tables in PostgreSQL, we established a foundational structure for multidimensional analysis.

We then constructed and queried a data cube to compute total sales and applied various OLAP operations such as **roll-up**, **slice**, and **dice**. These operations demonstrated how aggregated and filtered data can be efficiently analyzed to extract meaningful business insights.

This lab reinforced the core concepts of data warehousing, schema design, and OLAP functionality. It highlighted how such systems support strategic decision-making in real-world scenarios by enabling flexible, high-performance analytical queries on large datasets.