Name:	Swarup Satish Kakade
Roll No:	17
Class/Sem:	TE/V
Experiment No.:	2
Title:	Implementation of Dimension and Fact tables and perform OLAP operations.
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim: Implementation of Dimension and Fact tables and perform OLAP operations.

Objective: OLAP stands for Online Analytical Processing. The objective of OLAP is to analyze information from multiple database systems at the same time. It is based on multidimensional data model and allows the user to query on multi-dimensional data.

Theory:
- Online Analytical Processing Server (OLAP) is based on the multidimensional data model.
- The main aim of OLAP is to provide multidimensional analysis to the underlying data.
  Following is the list of OLAP operations:
  1. Roll-up
  2. Drill-down
  3. Slice
  4. Dice
  5. Pivot (rotate)

Roll-up:

- The roll-up operation (also called the drill-up operation) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction.
- Figure 2.1 shows the result of a roll-up operation performed on the central cube by climbing up the concept hierarchy for location.
- This hierarchy was defined as the total order "street < city < province or state < country."
- The roll-up operation aggregates the data by ascending the location hierarchy from the level of city to the level of country.
- In other words, rather than grouping the data by city, the resulting cube groups the data by country.

Drill-down:
- Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data.
- Drill-down can be realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions.
- Figure 2.1 shows the result of a drill-down operation performed on the central cube by stepping down a concept hierarchy for time defined as "day < month < quarter < year."
- Drill-down occurs by descending the time hierarchy from the level of quarter to the more detailed level of month.
- The resulting data cube details the total sales per month rather than summarizing them by quarter.

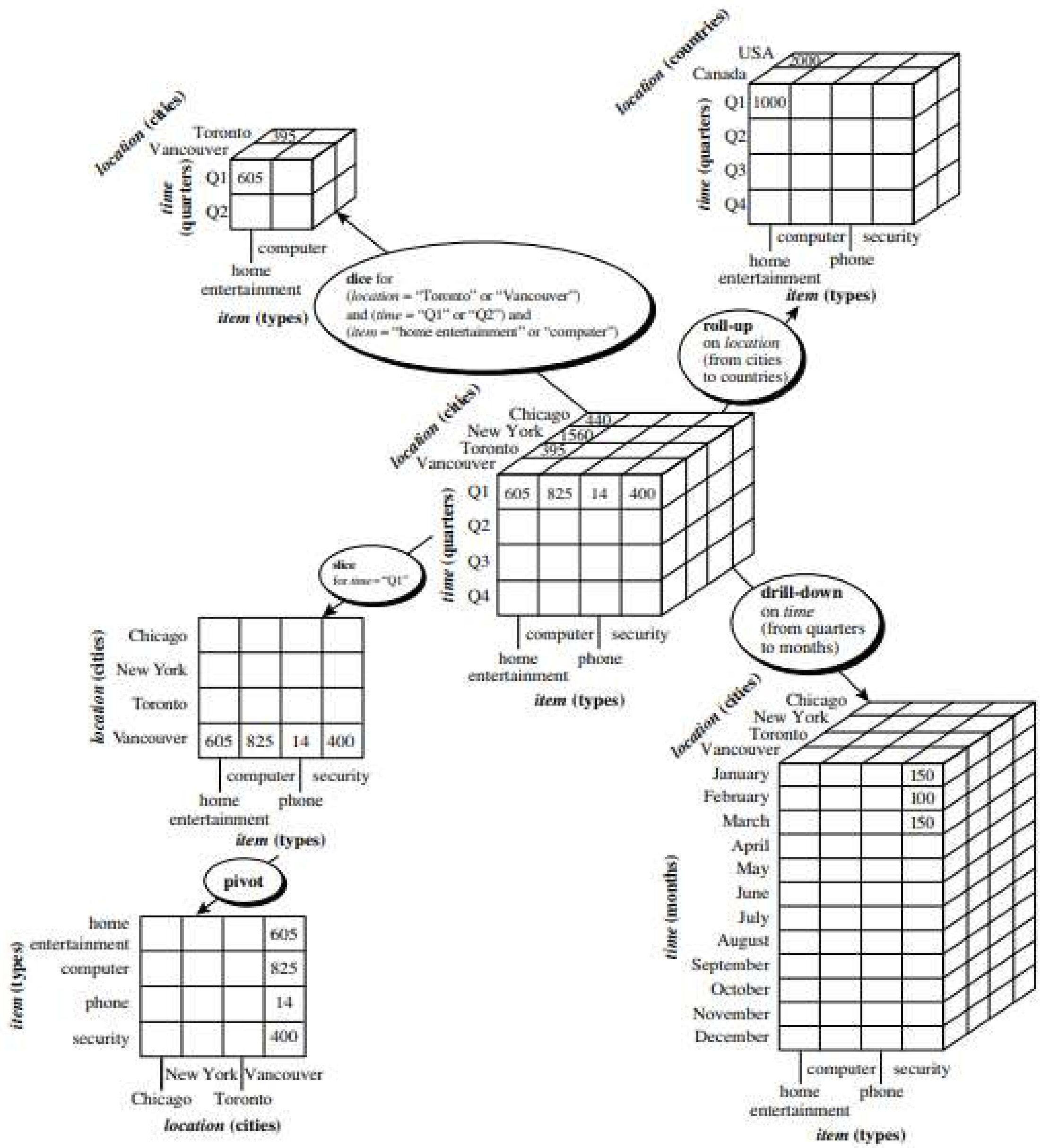

Figure 2.1: Examples of typical OLAP operations on multidimensional data.

Slice:

- The slice operation performs a selection on one dimension of the given cube, resulting in a sub cube.
- Figure 2.1 below shows a slice operation where the sales data are selected from the central cube for the dimension time using the criterion time = "Q1."

Dice:

- The dice operation defines a sub cube by performing a selection on two or more dimensions.
- Figure 2.1 shows a dice operation on the central cube based on the following selection criteria that involve three dimensions: (location = "Toronto" or "Vancouver") and (time = "Q1" or "Q2") and (item = "home entertainment" or "computer").

Pivot:

- Pivot (also called rotate) is a visualization operation that rotates the data axes in view to provide an alternative data presentation.
- Figure 2.1 shows a pivot operation where the item and location axes in a 2-D slice are rotated.

Problem Statement:

You are tasked with analyzing banking transaction data to determine the total transaction amount for each branch within a specific period. The database includes dimension tables for customers, accounts, branches, and time, along with a fact table for transactions. Each transaction is linked to a specific account, branch, and time period. Your goal is to write a SQL query that retrieves the branch names and the corresponding total transaction amounts, summing both deposits and withdrawals, for the specified period.

Output:

Creating the Dimension Tables:

```
CREATE TABLE customers (
 customer_id INT PRIMARY KEY,
 first_name VARCHAR(50),
 last_name VARCHAR(50),
 email VARCHAR(100),
 phone VARCHAR(15),
 address VARCHAR(255)
);

CREATE TABLE accounts (
 account_id INT PRIMARY KEY,
 account_type VARCHAR(50),
 balance DECIMAL(15, 2),
 customer_id INT,
 FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);
```

```
CREATE TABLE branches (
 branch_id INT PRIMARY KEY,
 branch_name VARCHAR(100),
 branch_location VARCHAR(255)
);

CREATE TABLE time (
 time_id INT PRIMARY KEY,
 transaction_date DATE,
 transaction_month INT,
 transaction_year INT,
 transaction_quarter INT
);
```

2. Creating the Fact Table

```
CREATE TABLE transactions (
 transaction_id INT PRIMARY KEY,
 transaction_amount DECIMAL(15, 2),
 transaction_type VARCHAR(50), -- e.g., 'deposit', 'withdrawal'
 account_id INT,
 branch_id INT,
 time_id INT,
 FOREIGN KEY (account_id) REFERENCES accounts(account_id),
 FOREIGN KEY (branch_id) REFERENCES branches(branch_id),
 FOREIGN KEY (time_id) REFERENCES time(time_id)
);
```

3. Inserting values in both dimension and fact tables

```
INSERT INTO accounts (account_id, account_type, balance, customer_id) VALUES
(1, 'Checking', 1500.00, 1),
(2, 'Savings', 2500.50, 2),
(3, 'Checking', 800.75, 3),
(4, 'Savings', 1200.00, 4),
(5, 'Checking', 300.00, 5),
(6, 'Savings', 4000.00, 6);

INSERT INTO branches (branch_id, branch_name, branch_location) VALUES
(1, 'Main Branch', '123 Main St, Springfield'),
```

(2, 'West Branch', '456 West St, Springfield'),
(3, 'East Branch', '789 East St, Springfield'),
(4, 'North Branch', '321 North St, Springfield'),
(5, 'South Branch', '654 South St, Springfield'),
(6, 'Downtown Branch', '987 Downtown St, Springfield');

INSERT INTO time (time_id, transaction_date, transaction_month, transaction_year, transaction_quarter) VALUES
(1, '2023-01-15', 1, 2023, 1),
(2, '2023-02-20', 2, 2023, 1),
(3, '2023-03-10', 3, 2023, 1),
(4, '2023-04-05', 4, 2023, 2),
(5, '2023-05-25', 5, 2023, 2),
(6, '2023-06-30', 6, 2023, 2);

INSERT INTO transactions (transaction_id, transaction_amount, transaction_type, account_id, branch_id, time_id) VALUES
(1, 200.00, 'deposit', 1, 1, 1),
(2, 150.50, 'withdrawal', 2, 2, 2),
(3, 75.00, 'deposit', 3, 3, 3),
(4, 300.00, 'withdrawal', 4, 4, 4),
(5, 50.00, 'deposit', 5, 5, 5),
(6, 400.00, 'withdrawal', 6, 6, 6);

4. Displaying the tables

SELECT * FROM customers;

customer_id	first_name	last_name	email	phone	address
1	John	Doe	john.doe@example.com	555-1234	123 Elm St, Springfield
2	Jane	Smith	jane.smith@example.com	555-5678	456 Oak St, Springfield
3	Emily	Johnson	emily.johnson@example.com	555-8765	789 Pine St, Springfield
4	Michael	Brown	michael.brown@example.com	555-4321	321 Maple St, Springfield
5	Sarah	Davis	sarah.davis@example.com	555-3456	654 Cedar St, Springfield
6	David	Wilson	david.wilson@example.com	555-7890	987 Birch St, Springfield
NULL	NULL	NULL	NULL	NULL	NULL

SELECT * FROM accounts;

account_id	account_type	balance	customer_id
1	Checking	1500.00	1
2	Savings	2500.50	2
3	Checking	800.75	3
4	Savings	1200.00	4
5	Checking	300.00	5
6	Savings	4000.00	6
NULL	NULL	NULL	NULL

SELECT * FROM branches;

branch_id	branch_name	branch_location
1	Main Branch	123 Main St, Springfield
2	West Branch	456 West St, Springfield
3	East Branch	789 East St, Springfield
4	North Branch	321 North St, Springfield
5	South Branch	654 South St, Springfield
6	Downtown Branch	987 Downtown St, Springfield
NULL	NULL	NULL

SELECT * FROM time;

time_id	transaction_date	transaction_month	transaction_year	transaction_quarter
1	2023-01-15	1	2023	1
2	2023-02-20	2	2023	1
3	2023-03-10	3	2023	1
4	2023-04-05	4	2023	2
5	2023-05-25	5	2023	2
6	2023-06-30	6	2023	2
NULL	NULL	NULL	NULL	NULL

SELECT * FROM transactions;

transaction_id	transaction_amount	transaction_type	account_id	branch_id	time_id
1	200.00	deposit	1	1	1
2	150.50	withdrawal	2	2	2
3	75.00	deposit	3	3	3
4	300.00	withdrawal	4	4	4
5	50.00	deposit	5	5	5
6	400.00	withdrawal	6	6	6
NULL	NULL	NULL	NULL	NULL	NULL

5. Write SQL Queries for all the above OLAP operations.

Roll up:

SELECT b.branch_name, SUM(t.transaction_amount) AS total_transaction_amount
FROM transactions t

JOIN branches b ON t.branch_id = b.branch_id
GROUP BY b.branch_name;

branch_name	total_transaction_amount
Main Branch	200.00
West Branch	150.50
East Branch	75.00
North Branch	300.00
South Branch	50.00
Downtown Branch	400.00

Drill Down:
SELECT tm.transaction_month, SUM(t.transaction_amount) AS total_transaction_amount
FROM transactions t
JOIN time tm ON t.time_id = tm.time_id
GROUP BY tm.transaction_month
ORDER BY tm.transaction_month;

transaction_month	total_transaction_amount
1	200.00
2	150.50
3	75.00
4	300.00
5	50.00
6	400.00

Slice:
SELECT t.transaction_id, t.transaction_amount, tm.transaction_date
FROM transactions t
JOIN time tm ON t.time_id = tm.time_id
WHERE tm.transaction_quarter = 1; -- Selecting for Q1

transaction_id	transaction_amount	transaction_date
1	200.00	2023-01-15
2	150.50	2023-02-20
3	75.00	2023-03-10

Dice:
SELECT t.transaction_id, t.transaction_amount, b.branch_name, tm.transaction_month
FROM transactions t
JOIN branches b ON t.branch_id = b.branch_id
JOIN time tm ON t.time_id = tm.time_id

WHERE b.branch_name IN ('Main Branch', 'West Branch')
AND tm.transaction_month IN (1, 2); -- Selecting for January and February

	transaction_id	transaction_amount	branch_name	transaction_month
▶	1	200.00	Main Branch	1
	2	150.50	West Branch	2

Pivot:
SELECT
 b.branch_name,
 SUM(CASE WHEN tm.transaction_month = 1 THEN t.transaction_amount ELSE 0 END) AS January,
 SUM(CASE WHEN tm.transaction_month = 2 THEN t.transaction_amount ELSE 0 END) AS February,
 SUM(CASE WHEN tm.transaction_month = 3 THEN t.transaction_amount ELSE 0 END) AS March
FROM transactions t
JOIN branches b ON t.branch_id = b.branch_id
JOIN time tm ON t.time_id = tm.time_id
GROUP BY b.branch_name;

	branch_name	January	February	March
▶	Main Branch	200.00	0.00	0.00
	West Branch	0.00	150.50	0.00
	East Branch	0.00	0.00	75.00
	North Branch	0.00	0.00	0.00
	South Branch	0.00	0.00	0.00
	Downtown Branch	0.00	0.00	0.00

Conclusion:

Q1. What is the importance of OLAP operations?

1. Data Analysis: OLAP (Online Analytical Processing) operations help in multidimensional analysis of data, allowing users to extract meaningful insights.
2. Data Summarization: They enable quick aggregation and summarization of large datasets for decision-making.
3. Fast Query Performance: OLAP cubes are optimized for high-speed querying, reducing response times for complex queries.
4. Multidimensional View: OLAP provides a multidimensional view of data, facilitating better understanding and analysis across various dimensions (e.g., time, location, product).
5. Data Exploration: Users can easily drill down, roll up, slice, and dice data to explore different granularities of information.

6. Support for Strategic Decision Making: OLAP is crucial for business intelligence, enabling strategic and long-term decision-making through data insights.
7. Complex Calculation Support: OLAP operations support advanced calculations like trend analysis, forecasting, and complex ratios.
8. Data Integrity: They ensure consistent, accurate, and timely data presentation for decision-makers.

Q2. What are the key features of OLAP?

1) Multidimensional View: OLAP provides a multidimensional perspective of data, allowing users to analyze it across various dimensions like time, location, and products.
2) Support for Complex Queries: It handles complex queries and calculations efficiently, including aggregations, trends, and forecasting.
3) Data Aggregation: OLAP supports the automatic aggregation and summarization of data, enabling users to view different levels of detail.
4) Fast Query Performance: OLAP systems are optimized for quick responses to analytical queries, even with large datasets.
5) Interactive Data Exploration: Users can perform operations like drill-down, roll-up, slicing, and dicing to explore data interactively.
6) Time Series Analysis: It supports the analysis of historical data over time for trend identification and forecasting.
7) Data Integration: OLAP integrates data from multiple sources, providing a unified view for analysis.
8) Scalability: OLAP systems can handle large volumes of data efficiently, making them suitable for enterprise-level analytics.
9) User-Friendly Interface: OLAP tools often come with intuitive graphical interfaces for non-technical users to explore data easily.
10) Data Consistency: Ensures consistent and accurate reporting across different users and departments.