University of Dhaka

Department of Computer Science and Engineering

CSE-4255: Introduction to Data mining and warehousing Lab
4th Year 2nd Semester

Session: 2018 -19

Report Topic:
Data Warehousing: OLAP operations

Submitted by:
Amit Roy, Roll: JH- 40
Md. Tanvir Alam, Roll: SH-61

Submitted to:
Dr. Chowdhury Farhan Ahmed, Professor,
Department of Computer Science and Engineering, University of Dhaka

Abu Ahmed Ferdaus, Associate Professor,
Department of Computer Science and Engineering, University of Dhaka

Md. Ashraful Islam, Lecturer
Department of Computer Science and Engineering, University of Dhaka

Date of Submission:
November 11, 2019

Introduction: Data warehousing provides architectures and tools for business executives to systematically organize, understand, and use their data to make strategic decisions. A data warehouse refers to a data repository that is maintained separately from an organization's operational databases. According to William H. Inmon, a leading architect in the construction of data warehouse systems, "A data warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process." This short but comprehensive definition presents the major features of a data warehouse. The four keywords—subject-oriented, integrated, time-variant, and nonvolatile—distinguish data warehouses from other data repository systems, such as relational database systems, transaction processing systems, and file systems.

Data warehouses and OLAP tools are based on a multidimensional data model. The most popular data model for a data warehouse is a multidimensional model, which can exist in the form of a star schema, a snowflake schema, or a fact constellation schema.

Star schema: The most common modeling paradigm is the star schema, in which the data warehouse contains a large central table (fact table) containing the bulk of the data, with no redundancy, and a set of smaller attendant tables (dimension tables), one for each dimension.

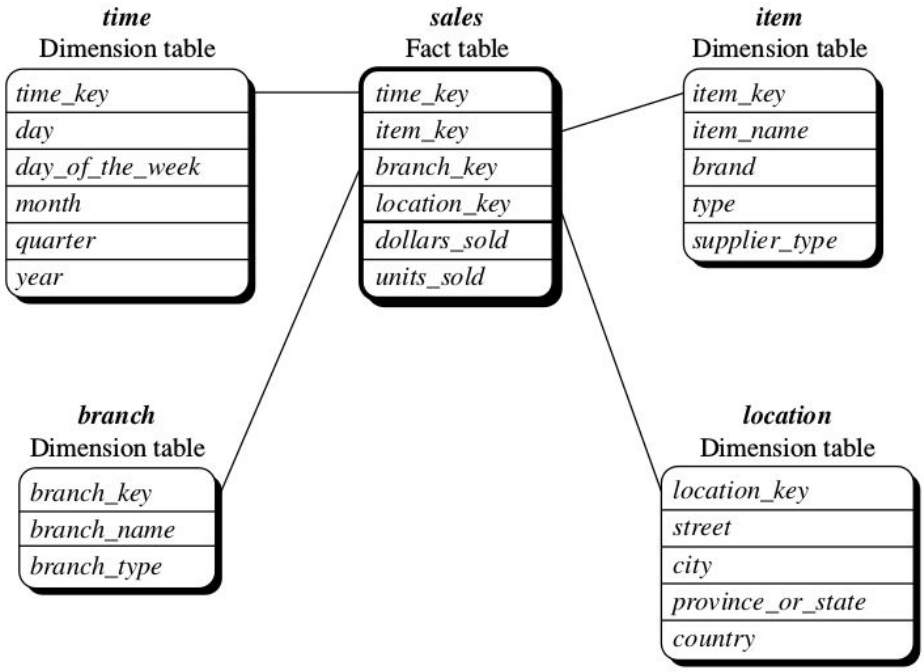

Figure: Star schema of a sales data warehouse

Snowflake schema: The snowflake schema is a variant of the star schema model, where some dimension tables are normalized, thereby further splitting the data into additional tables. The resulting schema graph forms a shape similar to a snowflake.

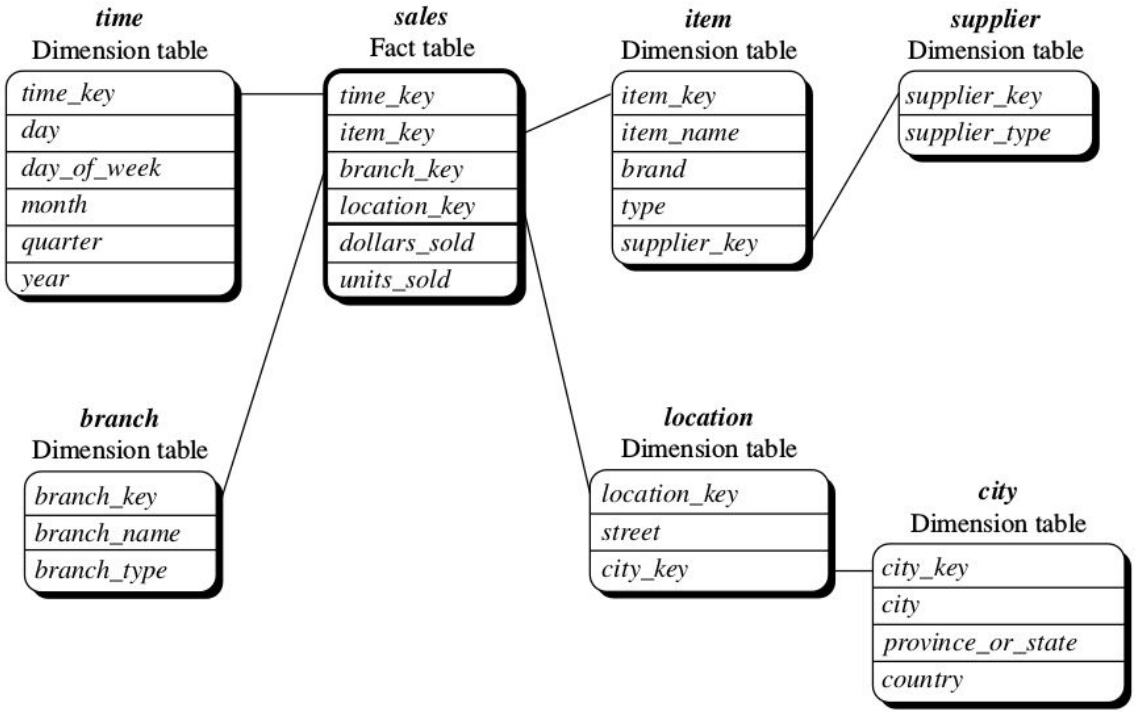

Figure: Snowflake schema of a sales data warehouse

Fact constellation: Sophisticated applications may require multiple fact tables to share dimension tables. This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or a fact constellation.

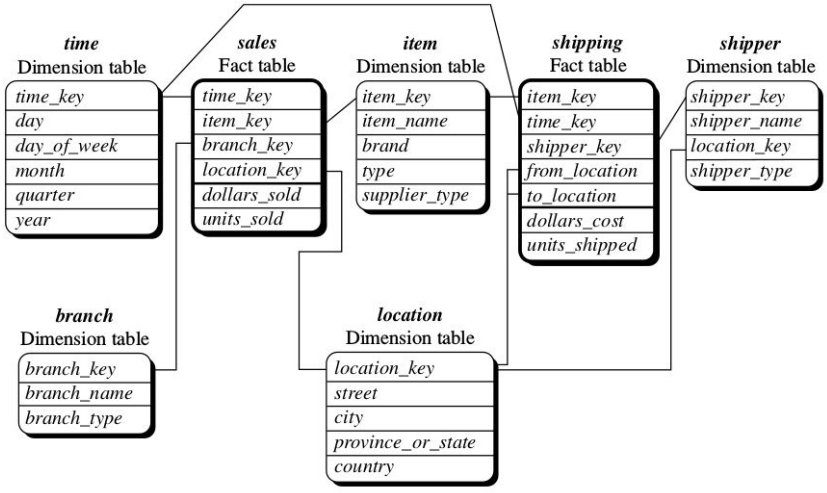

Figure: Fact constellation schema of a sales data warehouse

OLAP operations: Let's look at some typical OLAP operations for multidimensional data.

Roll-up: The roll-up operation (also called the drill-up operation by some vendors) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction. Suppose, a hierarchy is 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 the city to the level of the 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. A drill-down operation can be performed on the central cube by stepping down a concept hierarchy for time defined as "day < month < quarter < year."

Slice and dice: The slice operation performs a selection on one dimension of the given cube, resulting in a subcube.

Pivot (rotate): Pivot also called rotate is a visualization operation that rotates the data axes in view to provide an alternative data presentation.

We have designed the following snowflake schema on which we run the OLAP operations.

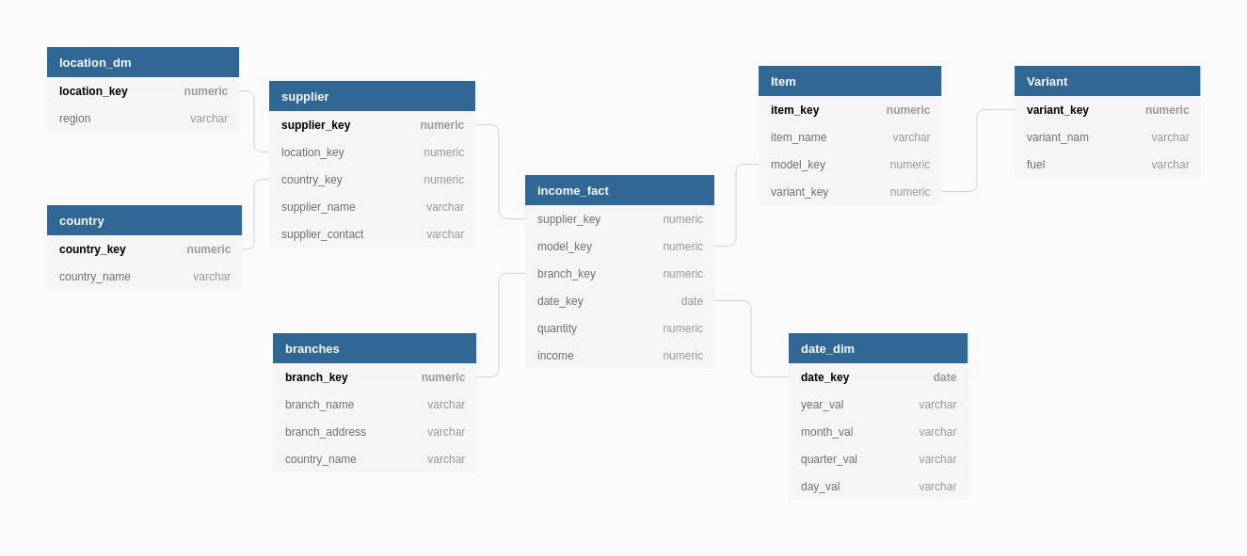

Figure1: A snowflake schema for OLAP operations

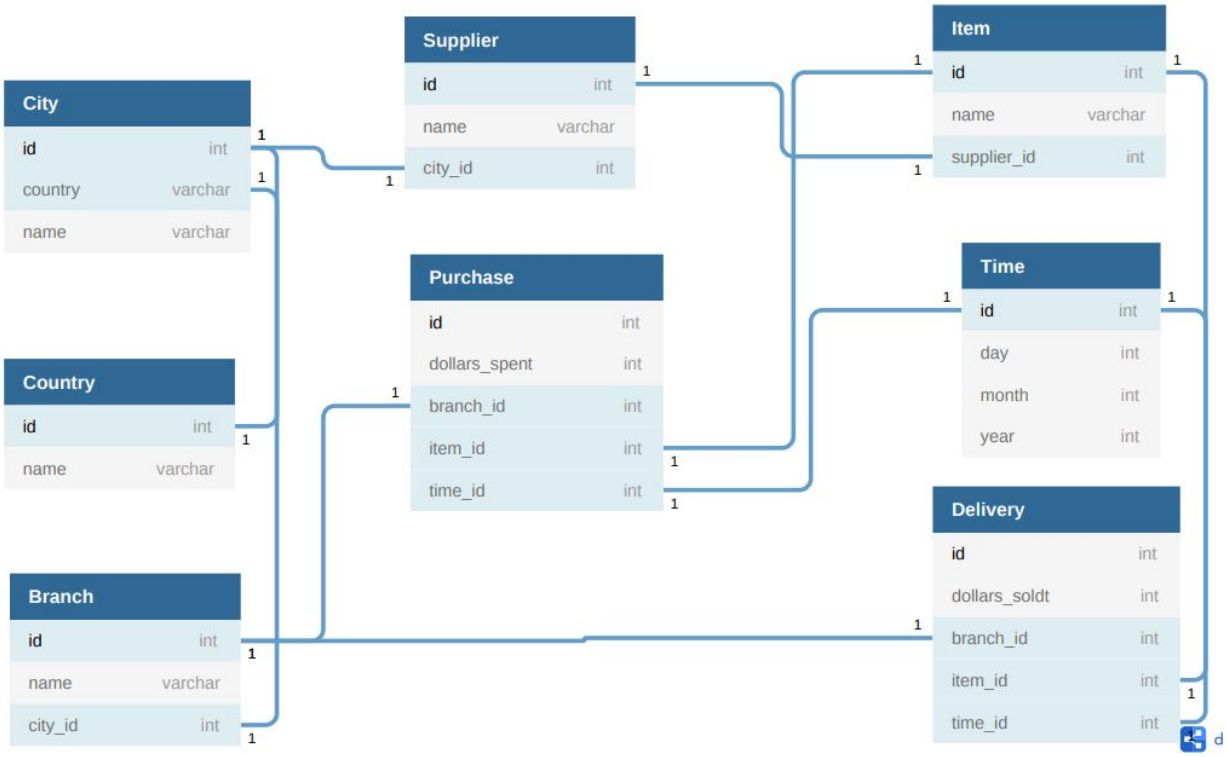

Figure 2: A galaxy schema for OLAP operations

OLAP Operations: To implement the OLAP operations, we have developed our schema in python language using SQLite database. We have also implemented the operations using ORACLE. We have shown various OLAP operations with the desired output for various queries in different domains. Here are some snapshots of different olap operations on our designed schema.

Roll-up:

Oracle SQL command:

```
SELECT supplier_key, model_key, branch_key, date_key,SUM(quantity) as Total_income
FROM income_fact
GROUP BY rollup(supplier_key, model_key, branch_key, date_key)
ORDER BY supplier_key, model_key, branch_key, date_key;
```

Query Result:

	SUPPLIER_KEY	MODEL_KEY	TOTAL_INCOME
1	1	1	1
2	10	10	10
3	11	11	11
4	12	12	12
5	13	13	13
6	14	14	14
7	15	15	15
8	16	16	16
9	17	17	17
10	18	18	18
11	19	19	19
12	2	2	2
13	20	20	20
14	21	21	21
15	22	22	22

Figure: Roll-up operation snapshot output

CUBE:

Oracle SQL command:

```
SELECT supplier_key, model_key, branch_key, date_key,SUM(quantity) as Total_income
FROM income_fact
GROUP BY cube(supplier_key, model_key, branch_key, date_key)
ORDER BY supplier_key, model_key, branch_key, date_key;
```

Query Result:

	SUPPLIER_KEY	MODEL_KEY	BRANCH_KEY	DATE_KEY	TOTAL_INCOME
1	1	1	1	1	1
2	1	1	1	(null)	1
3	1	1	(null)	1	1
4	1	1	(null)	(null)	1
5	1	(null)	1	1	1
6	1	(null)	1	(null)	1
7	1	(null)	(null)	1	1
8	1	(null)	(null)	(null)	1
9	10	10	10	10	10
10	10	10	10	(null)	10
11	10	10	(null)	10	10
12	10	10	(null)	(null)	10
13	10	(null)	10	10	10
14	10	(null)	10	(null)	10
15	10	(null)	(null)	10	10

Figure: Cube operation snapshot output

Pivot:

Oracle SQL command:
SELECT supplier_key, model_key,SUM(quantity) as Total_income
FROM income_fact
GROUP BY (supplier_key, model_key)
ORDER BY supplier_key, model_key;

Query Result:

	SUPPLIER_KEY	MODEL_KEY	BRANCH_KEY	DATE_KEY	TOTAL_INCOME
1	1	1	1	1	1
2	1	1	1	(null)	1
3	1	1	(null)	(null)	1
4	1	(null)	(null)	(null)	1
5	10	10	10	10	10
6	10	10	10	(null)	10
7	10	10	(null)	(null)	10
8	10	(null)	(null)	(null)	10
9	11	11	11	11	11
10	11	11	11	(null)	11
11	11	11	(null)	(null)	11
12	11	(null)	(null)	(null)	11
13	12	12	12	12	12
14	12	12	12	(null)	12
15	12	12	(null)	(null)	12

Figure: Pivot operation snapshot output

Grouping Operation:
Oracle SQL command:
SELECT supplier_key, model_key, branch_key,SUM(quantity) as Total_Sale,
  Grouping(supplier_key) as supplier_key_flag,
  Grouping(branch_key) as branch_key_flag,
  Grouping(model_key) as model_key_flag
FROM income_fact
GROUP BY rollup(supplier_key, model_key, branch_key, date_key)
ORDER BY supplier_key, model_key, branch_key, date_key;

Query Result:

	SUPPLIER_KEY	MODEL_KEY	BRANCH_KEY	TOTAL_SALE	SUPPLIER_KEY_FLAG	BRANCH_KEY_FLAG	MODEL_KEY_FLAG
1	1	1	1	1	0	0	0
2	1	1	1	1	0	0	0
3	1	1	(null)	1	0	1	0
4	1	(null)	(null)	1	0	1	1
5	10	10	10	10	0	0	0
6	10	10	10	10	0	0	0
7	10	10	(null)	10	0	1	0
8	10	(null)	(null)	10	0	1	1
9	11	11	11	11	0	0	0
10	11	11	11	11	0	0	0
11	11	11	(null)	11	0	1	0
12	11	(null)	(null)	11	0	1	1
13	12	12	12	12	0	0	0
14	12	12	12	12	0	0	0
15	12	12	(null)	12	0	1	0

Figure: Grouping Operation snapshot output

Decoding Operation:

Oracle SQL command:

```
SELECT decode(grouping(supplier_key),1, '0', supplier_key) as supplier_key,
  decode(grouping(model_key),1, '0', model_key) as model_key,
  decode(grouping(branch_key),1, '0', branch_key) as branch_key,
  SUM(quantity) as Total_Sale
FROM income_fact
GROUP BY rollup(supplier_key, model_key, branch_key, date_key)
ORDER BY supplier_key, model_key, branch_key, date_key;
```

Query Result:

	SUPPLIER_KEY	MODEL_KEY	BRANCH_KEY	TOTAL_SALE
1	0	0	0	1275
2	1	0	0	1
3	1	1	0	1
4	1	1	1	1
5	1	1	1	1
6	10	0	0	10
7	10	10	0	10
8	10	10	10	10
9	10	10	10	10
10	11	0	0	11
11	11	11	0	11
12	11	11	11	11
13	11	11	11	11
14	12	0	0	12
15	12	12	0	12

Figure: Decoding Operation snapshot output

Conclusion: Data warehousing is a useful tool for the summarization of an organization of various years of data. Various knowledge can be mined from data warehousing and this knowledge can be applied in making business policy to sustain with other competitors.