Chapter #7

Workload & Data Volume

Data volume: - static property of data mart

→ it refers to the size and scale of the data stored in a data mart. it involves the number of different values for each attribute and the total number of records in the data mart.

→ Understanding data volume is crucial for:

Sizing the data mart:

Determining the necessary hardware and storage space to accomodate the data.

Logical and Physical design:

Making informed decisions about how the data is organized and stored to optimize performance.

Query cost estimation:

Predicting the resources required to process different types of queries.

Workload: - Dynamic property change over time.

→ it refers to the types and frequency of queries that users execute against the data mart. These queries can range from simple reports to complex statistical analysis.

Workload analysis help:

Assess conceptual schema:

Ensure that the data model is designed to efficiently handle the expected queries.

Tune implementation:

Optimize the physical design and configuration of the data mart to improve query performance.

monitor & Adjust:

Continuously track the actual workload to make necessary adjustments to the logical & Physical design as user needs evolve.

Definition:

Given a set of fact schemata. the workload is a set of (q_0, n_0)

q_0 — query on one or more fact schemata

n_0 - query frequency

7.1: Dimensional expressions & queries on fact Schemata

Dimensional queries:

→ Dimensional expressions are a way to specify which data need to retrieve from a data mart.

→ They consist of:
- fact name
- Aggregation clause
- Selection predictions — conditions to filter data.

<expression> ::= <fact name><aggregation clause>

Dimensional Queries:

→ Dimensional queries are used to execute the dimensional expression that have created.

→ They specify which means to calculate and which attributes include in the results.

→ The are useful for analyzing large amounts of data.

EX:

SALE (date, product, store
date.month = 12/2008 AND
store.state = 'Virginia' AND
Product = 'Shiny')

→ Dimensional expression, selects all sales of the "Shiny" product in Virginia during December 2008.

→ Dimensional query for stores with high sales:

SALE (date, product, store:
quantity > 100 AND
date.month = 12/2008 AND
store.state = 'Virginia' AND
product = 'Shiny').store

7.2 Drill - Across Queries

→ Drill - across queries allow to compare data across different fact tables in a data warehouse.

→ They need to create relationships between two or more fact schemata.

1. Comparison Drill-Across:

- This type of query compares measures from comparable fact schema.
- it requires an overlapping schema where two or more fact schemata share common dimensions.

Ex:

Query calculates the difference between the shipped and stored quantity of items for each month and product category in 2007.

⇒ SHIPMENT * INVENTORY [month, category : Year = '2007']. (shippedQuantity - incomingQuantity).

2. Flow-Across Drill Across:

→ This type of query sequentially queries across various fact schemata to determine one or more attribute values.

→ it does not require an overlapping schema.

→ The query uses the result of one query to filter the data in the next query.

Ex:

INVENTORY [month, Product : month = 1/2008 AND product IN SHIPMENT [month, product : month = '1/2008' AND ShippedQuantity > 1000]. product]. level

Nested GPSJ Queries:

- it is more advance type of query called Generalized Projection/ Selection/ Join queries. These queries are designed to handle complex data analysis tasks that involve multiple levels of grouping & aggregation.
- They are useful bcz of:
  - flexibility
  - Customizable Aggregations
- They are based on relational algebra, a mathematical framework for manipulating relational databases.
- GPSJ are powerful but complex.

13 Validating a workload in a Conceptual schema:

Validation ensures that the data warehouse can effectively handle the queries that users want to run.

Why validation is important?

1. **Ensuring Completeness:**
The validation process checks if the conceptual schema includes all the necessary attributes to support the desired queries.

2. **Identifying issues:**
It helps identify potential problems that might arise during query execution, such as:

- **Missing attributes:**
if a required attribute is missing, it would not be possible to group or filter data as needed.
- **Incorrect r/s:**
if r/s between tables are not defined correctly, not able to get expected results.
- **Unsupported aggregations:**
if the data warehouse does not support the type of aggregation that need, would not be able to calculate the desired metrics.

14 Workload and Users:

→ Designing a data mart also means defining the options to access data and providing specific users with the rights to access specific data in specific modes.

→ For doing this, classify end users and the types of queries that end users want to submit to a data mart.

Classifying Users:

- **User profiles:** Create user profiles that define the specific data and actions that each user is allowed to perform.
it is created for each user.

Access Restrictions:

These restrictions specify which parts of the fact schemata a user can view, navigate and use.

1. **Measures and descriptive attributes:**

→ These restrictions control which specific data elements a user can see.

2. **Hierarchies & Dimensional Attributes:**

→ These restrictions control how a user can drill down into the data. for example, a user might be able to see data at a regional level but not at a city level.

3. **Data instances:**

→ These restrictions limit the specific data records that a user can access.

ex:

Classification of a commercial data mart:

store managers:

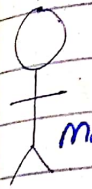
- can access data for their store.

sales district managers:

- can access sales data for all stores in their district.

marketing managers:

- can access marketing data for their department



manager:

- can access all data.

Data Volumes

- it help to determine the size and complexity of data warehouse. Also helps to determine the hardware and software resources needed to build and maintain the data warehouse.

Key Concepts:

1. Cardinality:
This refers to the number of distinct values for a particular attribute or group of attributes.

2. Sparsity:
This refers to the proportion of non-zero values in a fact table.

Estimating Data volumes:

To estimate the size of data-warehouse, consider the following:

1. Domain cardinality:
This refers to the number of distinct values for each dimension attribute.

2. Primary Group-by-set cardinality:
This refers to the number of unique combinations of primary dimension values.

3. Secondary Group-by set cardinality:
This refers to the number of unique combinations of secondary dimension values.

Estimating Sparsity:

- Sparsity is important because it affects the storage requirements and query performance of a data warehouse.
- A sparse fact table has many null values, which can increase storage costs and slow down query performance.
- To estimate sparsity, various statistical techniques can be used such as sampling or histogram analysis.

Cardenas Formula:

→ Statistical method used to estimate the size of a fact table based on the cardinality of its dimensions.

→ It has some limitations:

- overestimation
- Sparsity

Relationship between workload & Data volume:

Workload impacts data volume: A high workload can lead to increased data volume as more data is generated and stored over time.

Data volume impacts workload:

Large data volume affects query performance and response time impacting the overall workload.

Balancing Workload & Data volume:

→ To effectively manage workload and data volume, following strategies need to consider:

- Data partitioning
- Indexing
- Caching
- Data Compression
- ETL Optimization
- Monitoring & Tuning