Explore Dataware Houses

Gowreesh Gunupati

Question 1

Data warehouses are often constructed using relational databases. Explain the use of fact tables and star schemas to construct a data warehouse in a relational database. Also comment on whether a transactional database can and should be used to OLAP.

Answer

Facttable

In a data warehouse, a fact table is a central table that contains the measures or metrics that provide the numerical values of interest for the analysis. It typically contains the facts, or data points, that correspond to business events or transactions, such as sales transactions or customer interactions.

Starschema

A star schema is a type of relational database schema that organizes the data into a central fact table surrounded by a set of related dimension tables. In a star schema, each dimension table represents a specific attribute or category that can be used to filter or group the data in the fact table. This allows for efficient querying and analysis of large amounts of data.

The fact table and dimension tables are linked using foreign keys, which enables data to be efficiently queried and analyzed. The fact table usually contains the primary keys of the dimension tables as foreign keys, while the dimension tables contain descriptive attributes that can be used for filtering and analysis.

Advantage of Star Schema

- Simplified queries In comparison to other join logic required to retrieve data from a transactional schema that is well normalized, star schema join logic is extremely simple.
- Simplified Business Reporting Logic - The star schema simplifies basic business reporting logic, such as reporting and period-over-period, in compared to a transactional schema that is heavily standardized.
- All OLAP systems employ the Feeding Cubes - Star schema to effectively construct OLAP cubes. A ROLAP mode of operation, which uses a star schema as a source without creating a cube structure, is actually provided by most major OLAP systems.

whether a transactional database can and should be used to OLAP

A transactional database is optimized for transaction processing, where the primary concern is to ensure the integrity and consistency of the data during insertion, updating, and deletion. On the other hand, OLAP (Online Analytical Processing) is optimized for complex analytical queries, where the primary concern is to efficiently retrieve and aggregate data for analysis.

While it is possible to use a transactional database for OLAP, it is not recommended due to performance issues. OLAP requires complex aggregations and joins, which can slow down a transactional database. In addition, the schema design of a transactional database is typically optimized for transaction processing, and may not be suitable for analytical querying.

In summary, fact tables and star schemas are key components in constructing a data warehouse using a relational database. While a transactional database can be used for OLAP, it is not recommended due to performance and design considerations.

Question 2

Explain the difference between a data warehouse, a data mart, and a data lake. Provide at least one example of their use from your experience or how you believe they might be used in practice. Find at least one video, article, or tutorial online that explains the differences and embed that into your notebook.

Answer

Data warehouse, data mart, and data lake are all methods for storing and managing large amounts of data, but they differ in terms of the types of data they store and how they are organized.

A data warehouse, a data mart, and a data lake are all types of data storage solutions that are used to manage and store large volumes of data. While they share some similarities, there are also key differences between them in terms of their purpose, design, and usage.

Data warehouse

A data warehouse is a centralized repository of integrated data that is designed to support business intelligence and decision-making activities. It typically involves extracting data from multiple sources, transforming it into a common format, and loading it into a structured database schema. The data is organized around a set of predefined dimensions and subject areas, and is optimized for complex querying and analysis. For example, a retail company might use a data warehouse to analyze sales trends across different products, regions, and time periods.

Data mart

A data mart is a subset of a data warehouse that is designed to serve a specific business unit or department. It contains a subset of the data from the data warehouse, which is tailored to meet the needs of a particular group of users. Data marts are typically easier to build and maintain than a full-scale data warehouse, as they require less complex data integration and can be optimized for a specific set of use cases. For example, a marketing department might use a data mart to analyze customer behavior and campaign effectiveness.

Data lake

A data lake is a storage repository that allows organizations to store and process large volumes of structured and unstructured data at scale. Unlike a data warehouse or a data mart, a data lake does not enforce a predefined schema or data structure, and can store data in its raw format. This makes it more flexible and scalable than traditional data storage solutions, but also more complex to manage and query. Data lakes are often used for exploratory analytics and machine learning applications, where the focus is on exploring large amounts of data to uncover new insights and patterns. For example, a healthcare organization might use a data lake to store and analyze patient records, medical images, and other types of health data to improve patient outcomes and develop new treatments.

Here is a video that explains the differences between data warehouse, data mart, and data lake in more detail:

Click here to watch the video

Click here to read the article

In my experience, I have worked on several projects that involved designing and building data warehouses and data marts. One example is a project for a financial services company, where we built a data warehouse to integrate data from multiple sources and provide a single view of customer transactions and behavior. This allowed the company to analyze customer behavior across different products and channels, and make more informed decisions about marketing and product development.

Question 3

After the general explanation of fact tables and star schemas, design an appropriate fact table for Practicum I's bird strike database. Of course, there are many fact tables one could build, so pick some analytics problem and design a fact table for that. Be sure to explain your approach and design reasons. Just design it (perhaps draw an ERD for it); you do not need to actually implement it or populate it with data (of course, you may do so if you wish in preparation for the next practicum).

Answer

To design an appropriate fact table for Practicum I's bird strike database, let's first consider an analytics problem we want to solve. One possible problem could be: "What are the most common bird species involved in bird strikes at different airports?"

To solve this problem, we need to track bird strikes at different airports and identify the bird species involved in each strike. We can use a fact table to store this data and create a star schema with dimensions such as airport, bird species, time, etc.

Here's an ERD for the fact table:

```
+---------------+
| bird_strike   |
+---------------+
| id            |
| airport_id    |
| bird_species_id|
| flight_date   |
| bird_count    |
+---------------+
```

Explanation:

bird_strike is the fact table we are designing. id is a unique identifier for each bird strike event. airport_id is a foreign key referencing the airport dimension table. bird_species_id is a foreign key referencing the bird_species dimension table. flight_date is the date and time of the flight during which the bird strike occurred. bird_count is the number of birds involved in the strike. With this fact table, we can easily answer questions such as:

What are the top 5 airports with the highest number of bird strikes? What are the top 10 bird species involved in bird strikes? What are the most common bird species involved in bird strikes at a specific airport? What is the trend of bird strikes over time? Overall, the design of the fact table follows the best practices for fact tables, such as having a primary key, foreign keys to dimension tables, and measures (such as bird_count). By using this fact table, we can efficiently analyze the data and gain insights into bird strikes at different airports.

To complete the star schema, we will also need to design the dimension tables required to support the bird_strike fact table.

Here are the dimension tables required:

airport dimension table: Contains information about airports, such as airport name, city, state, and country. The primary key in this table would be airport_id, which is referenced as a foreign key in the bird_strike fact table.

bird_species dimension table: Contains information about different bird species, such as common name, scientific name, and size. The primary key in this table would be bird_species_id, which is referenced as a foreign key in the bird_strike fact table.

time dimension table: Contains information about the date and time of each flight, such as year, month, day, hour, and minute. This dimension table can be used to analyze bird strikes over time. The primary key in this table would be time_id, which is not directly referenced in the bird_strike fact table, but can be used to join with the fact table to perform time-based analysis.

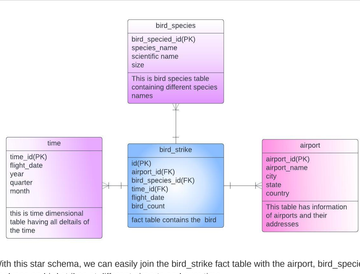
Here's an ERD for the star schema, including the bird_strike fact table and the three dimension tables:

```
          +-----------------------+
          |       airport         |
          +-----------------------+
          | airport_id (PK)       |
          | airport_name          |
          | city                  |
          | state                 |
          | country               |
          +-----------------------+

          +-----------------------+
          |     bird_species      |
          +-----------------------+
          | bird_species_id (PK)  |
          | common_name           |
          | scientific_name       |
          | size                  |
          +-----------------------+

          +-----------------------+
          |         time          |
          +-----------------------+
          | time_id (PK)          |
          | flight_date           |
          | year                  |
          | month                 |
          | day                   |
          | hour                  |
          | minute                |
          +-----------------------+
```

```
knitr::include_graphics("C:/Users/welcome/Downloads/ExploreDataBase.jpeg")
```



With this star schema, we can easily join the bird_strike fact table with the airport, bird_species, and time dimension tables to perform various analyses on bird strikes at different airports and over time.