**Assignment 1**

**AWS**

**Q1. What is OLAP AND OLTP?**

**Ans:** OLAP (Online Analytical Processing) and OLTP (Online Transaction Processing) are two different types of database systems, each designed to serve specific purposes within an organization.

**OLTP (Online Transaction Processing)**

* OLTP systems are designed for transaction-oriented applications. These are systems where many users execute a large volume of transactions, such as adding, updating, or deleting records in a database.
* The primary goal of OLTP systems is to efficiently process a high volume of online transactions in real-time. These transactions are typically short, atomic operations that involve adding or modifying small amounts of data.
* OLTP databases are optimized for fast query processing, data integrity, and concurrent access by multiple users.
* Examples of OLTP applications include online banking systems, airline reservation systems, and e-commerce websites.

**OLAP (Online Analytical Processing):**

* OLAP systems are designed for complex analytical and ad-hoc queries. They are used for data analysis, reporting, and decision-making purposes.
* OLAP databases are optimized for querying and aggregating large volumes of historical data. They typically store summarized, multidimensional data, which allows users to slice and dice data along various dimensions.
* OLAP systems often involve complex queries that require aggregations, calculations, and comparisons across different dimensions and levels of granularity.
* Examples of OLAP applications include business intelligence (BI) systems, data warehouses, and executive dashboards.

**Q2. Difference between OLAP and OLTP?**

**Ans:** The primary purpose of online analytical processing (OLAP) is to analyze aggregated data, while the primary purpose of online transaction processing (OLTP) is to process database transactions.

You use OLAP systems to generate reports, perform complex data analysis, and identify trends. In contrast, you use OLTP systems to process orders, update inventory, and manage customer accounts.

|  |  |  |
| --- | --- | --- |
| **Criteria** | **OLAP** | **OLTP** |

|  |  |  |
| --- | --- | --- |
| Purpose | OLAP helps you analyze large volumes of data to support decision-making. | OLTP helps you manage and process real-time transactions. |
| Data source | OLAP uses historical and aggregated data from multiple sources. | OLTP uses real-time and transactional data from a single source. |
| Data structure | OLAP uses multidimensional (cubes) or relational databases. | OLTP uses relational databases. |
| Data model | OLAP uses star schema, snowflake schema, or other analytical models. | OLTP uses normalized or denormalized models. |
| Volume of data | OLAP has large storage requirements. Think terabytes (TB) and petabytes (PB). | OLTP has comparatively smaller storage requirements. Think gigabytes (GB). |
| Response time | OLAP has longer response times, typically in seconds or minutes. | OLTP has shorter response times, typically in milliseconds |
| Example applications | OLAP is good for analyzing trends, predicting customer behavior, and identifying profitability. | OLTP is good for processing payments, customer data management, and order processing. |

**3Q. Database NormalForms?**

## **Ans: Normalization of DBMS:**

In database management systems (DBMS), normal forms are a series of guidelines that help to ensure that the design of a database is efficient, organized, and free from data anomalies. There are several levels of normalization, each with its own set of guidelines, known as normal forms.

* **First Normal Form (1NF):** This is the most basic level of normalization. In 1NF, each table cell should contain only a single value, and each column should have a unique name. The first normal form helps to eliminate duplicate data and simplify queries.
* **Second Normal Form (2NF):** 2NF eliminates redundant data by requiring that each non-key attribute be dependent on the primary key. This means that each column should be directly related to the primary key, and not to other columns.
* **Third Normal Form (3NF):** 3NF builds on 2NF by requiring that all non-key attributes are independent of each other. This means that each column should be directly related to the primary key, and not to any other columns in the same table.
* **Boyce-Codd Normal Form (BCNF):** BCNF is a stricter form of 3NF that ensures that each determinant in a table is a candidate key. In other words, BCNF ensures that each non-key attribute is dependent only on the candidate key.
* **Fourth Normal Form (4NF):** 4NF is a further refinement of BCNF that ensures that a table does not contain any multi-valued dependencies.
* **Fifth Normal Form (5NF):** 5NF is the highest level of normalization and involves decomposing a table into smaller tables to remove data redundancy and improve data integrity.

**4Q: Dimension versus Fact table and types of Dimensions ?**

**Ans:** The key differences between Fact and Dimension Tables are as follows:

* The Dimension table is a partner to the fact table and contains descriptive qualities that can be used as query constraints. The fact table includes measurements, metrics, or facts about business operations.
* A star and snowflake schema's core is where the table containing it is located, while its edges are where the dimension table is situated.
* A fact table should be lengthy, descriptive, full, and of guaranteed quality, while a fact table is characterized by its grain or even most atomic level.
* While the Dimension database includes extensive information, the Fact table is used to hold report labels.
* In contrast to the Dimension table, which incorporates hierarchies, the Table does not.

To understand these key differences even better, let us go through an example that connects the relationship between Fact and Dimension tables.

**Example of Fact Table vs. Dimension Table**

Let's say a business sells goods to clients. Every transaction is a fact that actually occurs, and the fact table is where these facts are recorded.

For instance:

|  |  |  |  |
| --- | --- | --- | --- |
| Order ID | Good ID | Client ID | Units Sold |
| 142 | 897 | 45 | 9 |
| 143 | 452 | 76 | 6 |
| 144 | 16 | 28 | 30 |

We can now include a dimension table for customers:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Client ID | Name | Gender | Income | Education | Region |
| 28 | Mariana Jones | F | 300000 | 3 | 3 |
| 45 | Christopher Nolan | M | 600000 | 2 | 2 |
| 76 | Mia Riana | F | 1000000 | 3 | 6 |

The client ID column in the fact table is used to connect to the dimension table. For instance, if you look at row 2 of the fact table, you can see that client 3, Mia Riana, made a purchase of six items with order id 143. To find out more information about what Mia purchased and when the company would also have a product table and a timetable.

When designing fact tables, there are considerations related to both physical constraints and data limitations. It's important to think about the size of the table and the ways in which it will be accessed. Indexes can be helpful for optimizing both of these factors. However, from a logical design standpoint, there should be no limitations on the structure of the table. It should be designed based on current and future needs, with as much flexibility as possible to allow for future updates without needing to completely rebuild the data.

**There are 9 types of dimensions:**

* Shrunken Rollup Dimensions
* Outrigger Dimensions
* Role-Playing Dimensions
* Step Dimensions
* Degenerate Dimensions
* Conformed Dimensions
* Junk Dimensions
* Dimension-to-Dimension Table Joins
* Swappable Dimensions

**5Q: Snowflake versus Starschema ?**

**Ans:** It's important to keep in mind the fundamental differences between star and snowflake schemas when comparing them: star schemas provide an effective method for organizing data in a data warehouse, whereas snowflake schemas are a version of star schemas that facilitate faster data processing.

## **What Is a Snowflake Schema?**

This particular kind of data warehouse schema is shaped like a snowflake. The snowflake schema aims to normalize the star schema's denormalized data. When the star schema's dimensions are intricate, highly structured, and have numerous degrees of connection, and the kid tables have several parent tables, the snowflake structure emerges. Some of the star schema's common issues are resolved by the snowflake schema.

The snowflake schema can be thought of as a "multi-dimensional" structure. A snowflake schema's central component comprises Fact Tables that link the data inside the Dimension Tables, which then radiate outward like the Star Schema. The snowflake schema, on the other hand, divides the Dimension Tables into several tables, resulting in a snowflake pattern. Up until they are fully normalized, the Dimension Tables are split across multiple tables.

### **Characteristics of Snowflake Schema**

The snowflake schema is characterized by a normalized data structure, with data divided into smaller, more specialized tables that are related to each other through foreign keys.

These are its main characteristics:

* Small disc space is required by the snowflake schema.
* The new dimension to the schema is simple to implement.
* Performance is impacted because there are numerous tables.
* Two or even more sets of attributes that describe data at various grains make up the dimension table.
* A single dimension table's sets of characteristics are filled in by various source systems.

Now that we have a basic understanding of the snowflake schema, let's dive into the specifics of the star schema and explore what sets it apart from other data organization techniques.

## **What Is a Star Schema?**

The star schema is the most straightforward method for arranging data in the data warehouse. Any or even more Fact Tables that index a number of Dimension Tables may be present in the star schema's central area. Dimensions Keys, Values, and Attributes are found in Dimension Tables, which are used to define Dimensions.

The star schema's objective is to distinguish between the descriptive or "DIMENSIONAL" data and the numerical "FACT" data that pertains to a business.

The information displayed in a numerical format, such as cost, speed, weight, and quantity, might be considered fact data. Along with numbers, dimensional data can also contain non-numerical elements like colors, places, names of salespeople and employees, etc.

While the Dimension Data is contained inside the Dimension Tables, the Fact Data is arranged within the Fact Tables. In a star schema, the Fact Tables are the integrating points at the core of a star.

### **Characteristics of Star Schema**

The star schema is characterized by a denormalized data structure, with all data related to a particular subject stored in a single large table and connected to smaller, dimensional tables through a single join.

These are some of the main characteristics of the star schema:

* A single one-dimension table can represent each aspect in a star schema.
* The collection of attributes should be in the dimension table.
* Using a foreign key, the dimensions table is connected to the fact table.
* No connections are made between the dimension tables.
* Key and measure would be in the fact table.
* The Star schema offers the best possible disc use and is simple to grasp.
* Tables for the dimensions are not standardized. As an OLTP architecture would have it, the Country ID in the image above does not have a Country lookup table.
* BI Tools provide extensive support for the schema.

## **Star Schema vs. Snowflake Schema**

|  |  |  |
| --- | --- | --- |
| **Basis of Distinction** | **Star Schema** | **Snowflake Schema** |
| Definition and Meaning | Both fact tables and dimension tables are present in a star schema. | Dimension tables, sub-dimension tables, and fact tables are all included in a snowflake schema. |
| Type of Model | The star schema is a top-down type of model. | The snowflake schema is a bottom-up type of model. |
| Space | Star schema uses more space compared to Snowflake Schema. | Snowflake schema uses less space comparatively. |
| Joint Relations | In a star schema, relationships between tables are represented by a single join, resulting in a simple data structure for fast query performance and easy data analysis. | The snowflake schema has a complex data structure with multiple levels of relationships between tables, represented by multiple joins. This can make the data structure more difficult to understand and result in slower query performance. |
| Response Time for Queries | Star schemas have faster query execution times due to a single join of a fact table and its attributes in dimensional tables. | Snowflake schemas require complex joins between tables, which can slow down query processing and impact other OLAP products. |
| Normalization | In a star schema, dimension tables are not organized in a normalized form. They are typically denormalized and contain multiple levels of information about a particular subject in a single table. | Dimension tables in snowflake schema are normalized. |
| Design Complexity | Has a simpler design compared to snowflake schema. | More complex design compared to star schema. |
| Query Complexity | Star schemas have simpler query design due to the fact the table is joined to only one level of dimensional tables. | Snowflake schemas, on another hand, have a more complex query design due to the need for multiple joins between the fact table and its dimensional tables. This leads to additional overhead in query writing. |
| Understanding Complexity | It is simpler to understand compared to snowflake schema. | More complex to understand compared to star schema. |
| Foreign Keys | Have a lesser number of foreign keys. | Comparatively has more foreign keys. |
| Data Redundancy | The star schema stores redundant data in the dimension tables. | The snowflake design fully normalizes the dimension tables and prevents data redundancy, |
| Advantages | * Simple and easy-to-understand data structure. * Fast query performance due to the single join between the fact table and its dimensional tables. * Suitable for large volumes of data * Good for ad-hoc querying and data analysis. | * Normalized data structure reduces redundancy and increases data integrity. * Allows for more complex relationships between data. * Allows for easier data maintenance and management. * Good for more structured predictable querying. |
| Disadvantages | * The star schema has a limited ability to depict complex relationships between data. * Can suffer from data redundancy and decreased data integrity. * May not be suitable for smaller volumes of data. | * The more complex data structure can be harder to understand and work with. * Multiple joins between tables can result in slower query performance. * Requires more storage and processing resources due to the larger number of tables. |