Online Analytical Processing (OLAP):

Online Analytical Processing consists of a type of software tools that are used for data analysis for business decisions. OLAP provides an environment to get insights from the database retrieved from multiple database systems at one time. **Examples** – Any type of Data warehouse system is an OLAP system. The uses of OLAP are as follows:

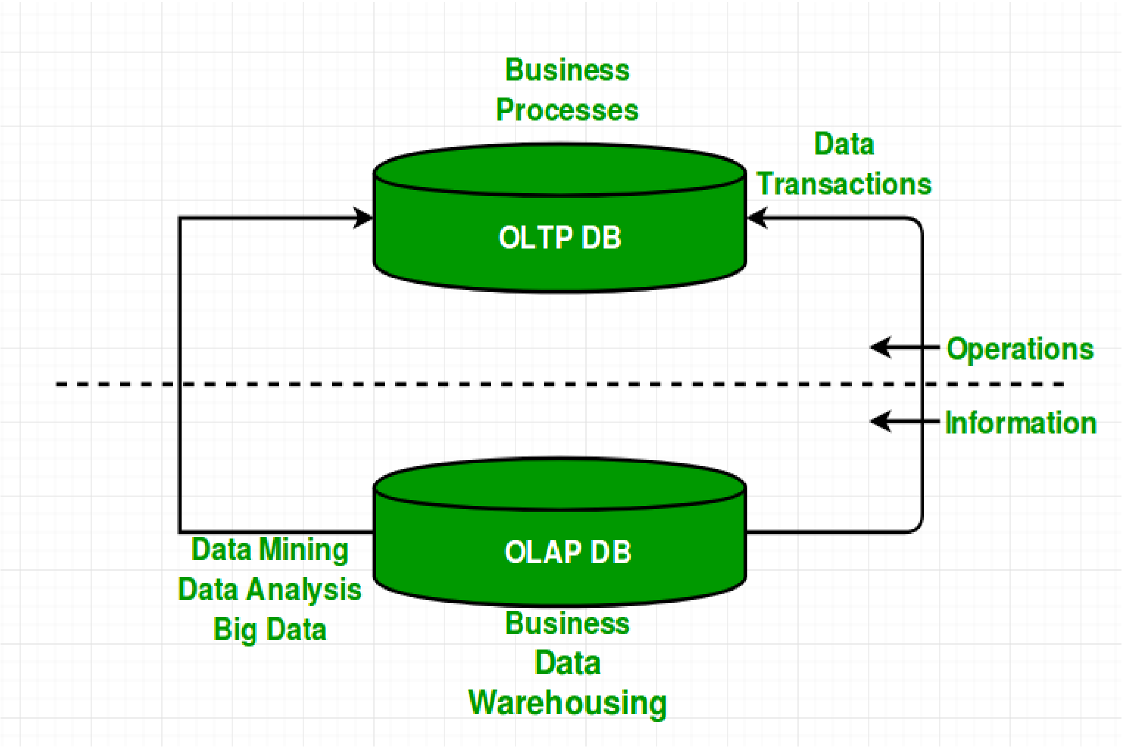
- Spotify analysed songs by users to come up with a personalized homepage of their songs and playlist.
- Netflix movie recommendation system.

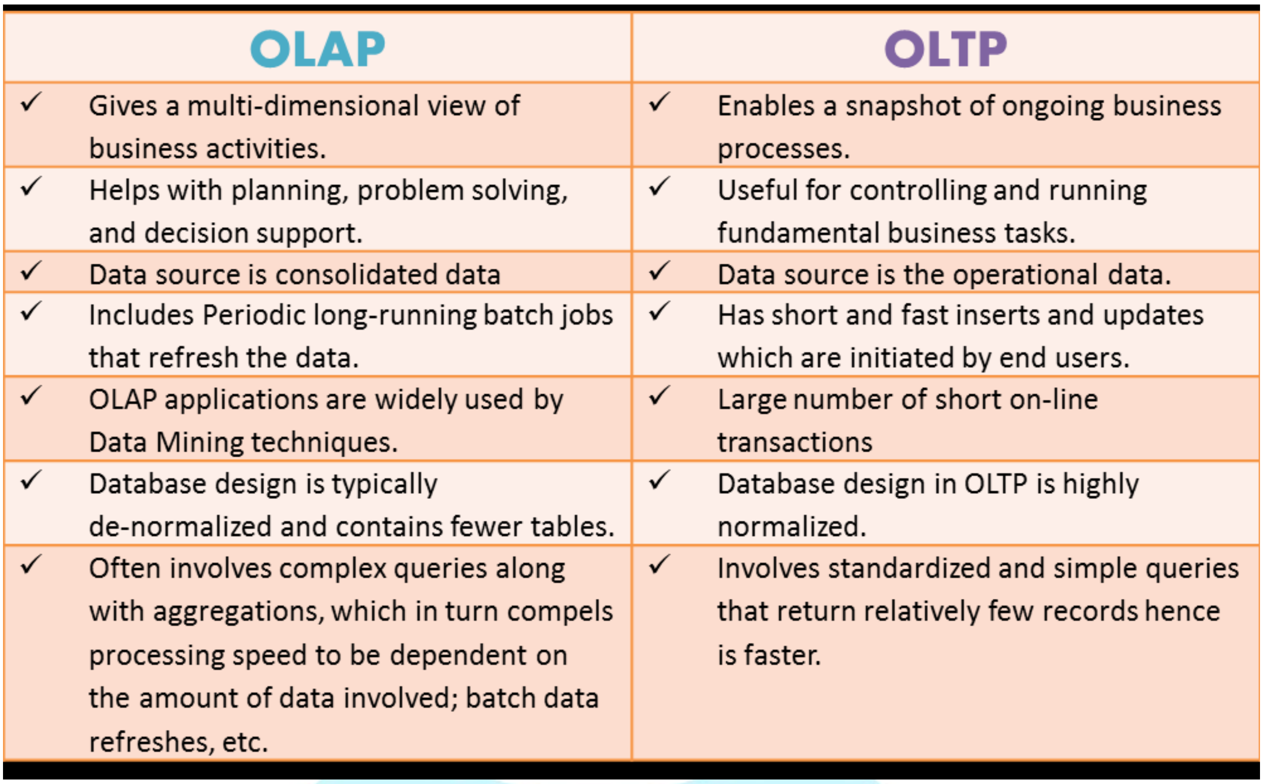
Online transaction processing (OLTP):

OLTP or Online Transaction Processing is a type of data processing that consists of executing a number of transactions occurring concurrently

Examples: Uses of OLTP are as follows:

- ATM center is an OLTP application.
- OLTP handles the ACID properties during data transactions via the application.
- It's also used for Online banking, Online airline ticket booking, sending a text message, add a book to the shopping cart.





OLAP	OLTP
✓ Gives a multi-dimensional view of business activities.	✓ Enables a snapshot of ongoing business processes.
✓ Helps with planning, problem solving, and decision support.	✓ Useful for controlling and running fundamental business tasks.
✓ Data source is consolidated data	✓ Data source is the operational data.
✓ Includes Periodic long-running batch jobs that refresh the data.	✓ Has short and fast inserts and updates which are initiated by end users.
✓ OLAP applications are widely used by Data Mining techniques.	✓ Large number of short on-line transactions
✓ Database design is typically de-normalized and contains fewer tables.	✓ Database design in OLTP is highly normalized.
✓ Often involves complex queries along with aggregations, which in turn compels processing speed to be dependent on the amount of data involved; batch data refreshes, etc.	✓ Involves standardized and simple queries that return relatively few records hence is faster.

Database (DB)

Data

Data is a collection of a distinct small unit of information. It can be used in a variety of forms like text, numbers, media, bytes, etc. it can be stored in pieces of paper or electronic memory, etc.

In computing, Data is information that can be translated into a form for efficient movement and processing. Data is interchangeable.

Database

A database is an organized collection of data, so that it can be easily accessed and managed. Between JDK, JRE, and JVM

You can organize data into tables, rows, columns, and index it to make it easier to find relevant information.

Database handlers create a database in such a way that only one set of software program provides access of data to all the users.

The main purpose of the database is to operate a large amount of information by storing, retrieving, and managing data.

There are many dynamic websites on the World Wide Web nowadays which are handled through databases. For example, a model that checks the availability of rooms in a hotel. It is an example of a dynamic website that uses a database.

There are many databases available like MySQL, Sybase, Oracle, Mongo DB, Informix, PostgreSQL, SQL Server, etc.

Modern databases are managed by the database management system (DBMS).

SQL or Structured Query Language is used to operate on the data stored in a database. SQL depends on relational algebra and tuple relational calculus.

Types of Databases

Here are some popular types of databases.

Distributed databases:

A distributed database is a type of database that has contributions from the common database and information captured by local computers. In this type of database system, the data is not in one place and is distributed at various organizations.

Relational databases:

This type of database defines database relationships in the form of tables. It is also called Relational DBMS, which is the most popular DBMS type in the market. Database example of the RDBMS system includes MySQL, Oracle, and Microsoft SQL Server database.

Object-oriented databases:

This type of computers database supports the storage of all data types. The data is stored in the form of objects. The objects to be held in the database have attributes and methods that define what to do with the data. PostgreSQL is an example of an object-oriented relational DBMS.

Centralized database:

It is a centralized location, and users from different backgrounds can access this data. This type of computers databases store application procedures that help users access the data even from a remote location.

Open-source databases:

This kind of database stored information related to operations. It is mainly used in the field of marketing, employee relations, customer service, of databases.

Cloud databases:

A cloud database is a database which is optimized or built for such a virtualized environment. There are so many advantages of a cloud database, some of which can pay for storage capacity and bandwidth. It also offers scalability on-demand, along with high availability.

Database Access Language:

Database Access language is used to access the data to and from the database, enter new data, update already existing data, or retrieve required data from DBMS. The user writes some specific commands in a database access language and submits these to the database.

Database Management System:

Database Management System (DBMS) is a collection of programs that enable its users to access databases, manipulate data, report, and represent data. It also helps to control access to the database

Advantages of DBMS

- DBMS offers a variety of techniques to store & retrieve data.
- DBMS serves as an efficient handler to balance the needs of multiple applications using the same data.
- Uniform administration procedures for data.
- Application programmers never exposed to details of data representation and storage.
- A DBMS uses various powerful functions to store and retrieve data efficiently.
- Offers Data Integrity and Security.
- The DBMS implies integrity constraints to get a high level of protection against prohibited access to data.
- A DBMS schedules concurrent access to the data in such a manner that only one user can access the same data at a time.
- Reduced Application Development Time.

Disadvantage of DBMS

DBMS may offer plenty of advantages but, it has certain flaws-

- Cost of Hardware and Software of a DBMS is quite high which increases the budget of your organization.
- Most database management systems are often complex systems, so the training for users to use the DBMS is required.
- In some organizations, all data is integrated into a single database which can be damaged because of electric failure or database is corrupted on the storage media.
- Use of the same program at a time by many users sometimes leads to the loss of some data.
- DBMS can’t perform sophisticated calculations.

Data Warehouse

Data Warehouse is a relational database management system (RDBMS) construct to meet the requirement of transaction processing systems. It can be loosely described as any centralized data repository which can be queried for business benefits. It is a database that stores information oriented to satisfy decision-making requests. It is a group of decision support technologies, targets to enabling the knowledge worker (executive, manager, and analyst) to make superior and higher decisions. So, Data Warehousing support architectures and tool for business executives to systematically organize understand and use their information to make strategic decisions.

- Data Warehouse environment contains an extraction, transportation, and loading (ETL) solution, an online analytical processing (OLAP) engine, customer analysis tools, and other applications that handle the process of gathering information and delivering it to business users.

A Data Warehouse can be viewed as a data system with the following attributes:

- It is a database designed for investigative tasks, using data from various applications.
- It supports a relatively small number of clients with relatively long interactions.
- It includes current and historical data to provide a historical perspective of information.
- Its usage is read-intensive.
- It contains a few large tables.

"Data Warehouse is a subject-oriented, integrated, and time-variant store of information in support of management's decisions."

Need for Data Warehouse

1. Business User: Business users require a data warehouse to view summarized data from the past. Since these people are non-technical, the data may be presented to them in an elementary form.
2. Store historical data: Data Warehouse is required to store the time variable data from the past. This input is made to be used for various purposes.
3. Make strategic decisions: Some strategies may be depending upon the data in the data warehouse. So, data warehouse contributes to making strategic decisions.
4. For data consistency and quality: Bringing the data from different sources at a commonplace, the user can effectively undertake to bring the uniformity and consistency in data.
5. High response time: Data warehouse has to be ready for somewhat unexpected loads and types of queries, which demands a significant degree of flexibility and quick response time

Benefits of Data Warehouse

1. Understand business trends and make better forecasting decisions.
2. Data Warehouses are designed to perform well enormous amounts of data.
3. The structure of data warehouses is more accessible for end-users to navigate, understand, and query.

Data Lakehouse:

A data lakehouse is a new, open data management architecture that combines the flexibility, cost-efficiency, and scale of data lakes with the data management and ACID transactions of data warehouses, enabling business intelligence (BI) and machine learning (ML) on all data.

Data Lakehouse: Simplicity, Flexibility, and Low Cost

Data lakehouses are enabled by a new, open system design: implementing similar data structures and data management features to those in a data warehouse, directly on the kind of low-cost storage used for data lakes. Merging them together into a single system means that data teams can move faster as they are able to use data without needing to access multiple systems. Data lakehouses also ensure that teams have the most complete and up-to-date data available for data science, machine learning, and business analytics projects.

There are a few key technology advancements that have enabled the data lakehouse:

- Metadata layers for data lakes
- New query engine designs providing high-performance SQL execution on data lakes
- Optimized access for data science and machine learning tools.

Difference between Schema and Table:

Schema

A schema is a collection of database objects including tables, views, triggers, stored procedures, indexes, etc. A schema is associated with a username which is known as the schema owner, who is the owner of the logically related database objects.

A schema always belongs to one database. On the other hand, a database may have one or multiple schemas. For example, in our `BikeStores` sample database, we have two schemas: `sales` and `production`. An object within a schema is qualified using the `schema_name.object_name` format like `sales.orders`. Two tables in two schemas can share the same name so you may have `hr.employees` and `sales.employees`.

Built-in schemas in SQL Server

SQL Server provides us with some pre-defined schemas which have the same names as the built-in database users and roles, for example: `dbo`, `guest`, `sys`, and `INFORMATION_SCHEMA`.

The default schema for a newly created database is `dbo`, which is owned by the `dbo` user account. By default, when you create a new user with the `CREATE USER` command, the user will take `dbo` as its default schema.

SQL Server CREATE SCHEMA statement overview

The `CREATE SCHEMA` statement allows you to create a new schema in the current database.

The following illustrates the simplified version of the `CREATE SCHEMA` statement:

```
CREATE SCHEMA schema_name
[AUTHORIZATION owner_name]
```

Code language: SQL (Structured Query Language) (sql)

In this syntax,

- First, specify the name of the schema that you want to create in the CREATE SCHEMA clause.
- Second, specify the owner of the schema after the AUTHORIZATION keyword.

SQL Server CREATE SCHEMA statement example

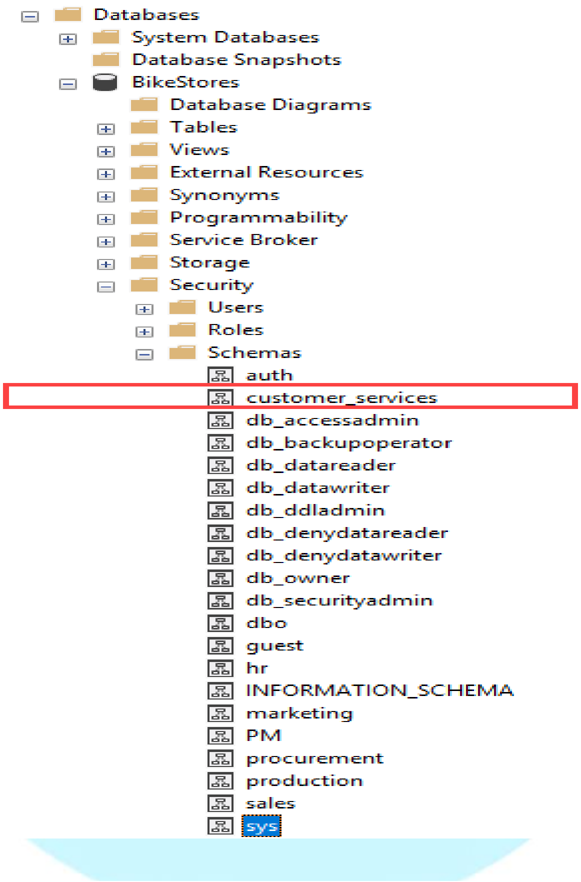
The following example shows how to use the `CREATE SCHEMA` statement to create the `customer_services` schema:

```
CREATE SCHEMA customer_services;
GO
```

Code language: SQL (Structured Query Language) (sql)

Note that `GO` command instructs the SQL Server Management Studio to send the SQL statements up to the `GO` statement to the server to be executed.

Once you execute the statement, you can find the newly created schema under the Security > Schemas of the database name.



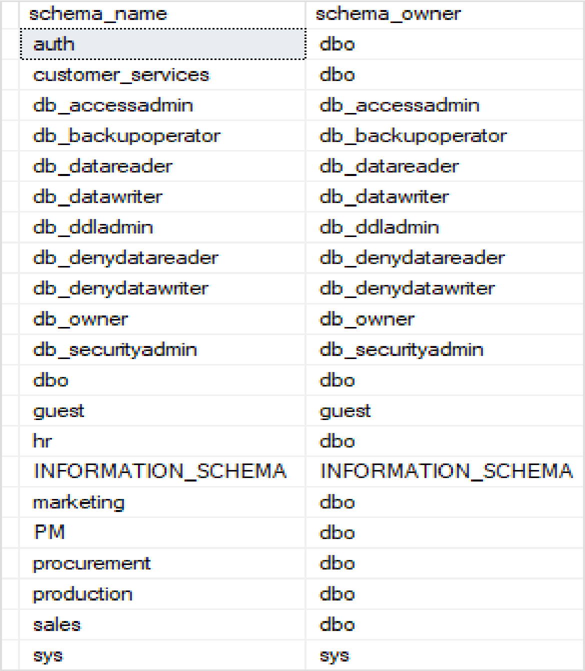
Databases
System Databases
Database Snapshots
BikeStores
Database Diagrams
Tables
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Users
Roles
Schemas
auth
customer_services
db_accessadmin
db_backupoperator
db_datareader
db_datawriter
db_ddladmin
db_denydatareader
db_denydatawriter
db_owner
db_securityadmin
dbo
guest
hr
INFORMATION_SCHEMA
marketing
PM
procurement
production
sales
sys

If you want to list all schemas in the current database, you can query schemas from the `sys.schemas` as shown in the following query:

```
SELECT
s.name AS schema_name,
u.name AS schema_owner
FROM
sys.schemas s
INNER JOIN sys.sysusers u ON u.uid = s.principal_id
ORDER BY
s.name;
```

Code language: SQL (Structured Query Language) (sql)

Here is the output:



schema_name	schema_owner
auth	dbo
customer_services	dbo
db_accessadmin	db_accessadmin
db_backupoperator	db_backupoperator
db_datareader	db_datareader
db_datawriter	db_datawriter
db_ddladmin	db_ddladmin
db_denydatareader	db_denydatareader
db_denydatawriter	db_denydatawriter
db_owner	db_owner
db_securityadmin	db_securityadmin
dbo	dbo
guest	guest
hr	dbo
INFORMATION_SCHEMA	INFORMATION_SCHEMA
marketing	dbo
PM	dbo
procurement	dbo
production	dbo
sales	dbo
sys	sys

After having the `customer_services` schema, you can create objects for the schema. For example, the following statement creates a new table named `jobs` in the `customer_services` schema:

```
CREATE TABLE customer_services.jobs(
job_id INT PRIMARY KEY IDENTITY,
customer_id INT NOT NULL,
description VARCHAR(200),
created_at DATETIME2 NOT NULL
);
```

Code language: SQL (Structured Query Language) (sql)

In this tutorial, you have learned how to use the SQL Server `CREATE SCHEMA` statement to create a new schema in the current database.

Table

Table is a collection of data, organized in terms of rows and columns. In DBMS term, table is known as relation and row as tuple.

Let's see an example of an employee table:

In the above table, "Employee" is the table name, "EMP_NAME", "ADDRESS" and "SALARY" are the column names. The combination of data of multiple columns forms a row e.g. "Ankit", "Lucknow" and 15000 are the data of one row.

Employee

EMP_NAME	ADDRESS	SALARY
Ankit	Lucknow	15000
Raman	Allahabad	18000
Mike	New York	20000

The SQL Table variable is used to create, modify, rename, copy and delete tables. Table variable was introduced by Microsoft.

It was introduced with SQL server 2000 to be an alternative of temporary tables.

It is a variable where we temporary store records and results. This is same like temp table but in the case of temp table we need to explicitly drop it.

Table variables are used to store a set of records. So declaration syntax generally looks like CREATE TABLE syntax.

```
create table "tablename"
("column1" "data type",
"column2" "data type",
...
"columnN" "data type");
```

When a transaction rolled back the data associated with table variable is not rolled back.

A table variable generally uses lesser resources than a temporary variable.

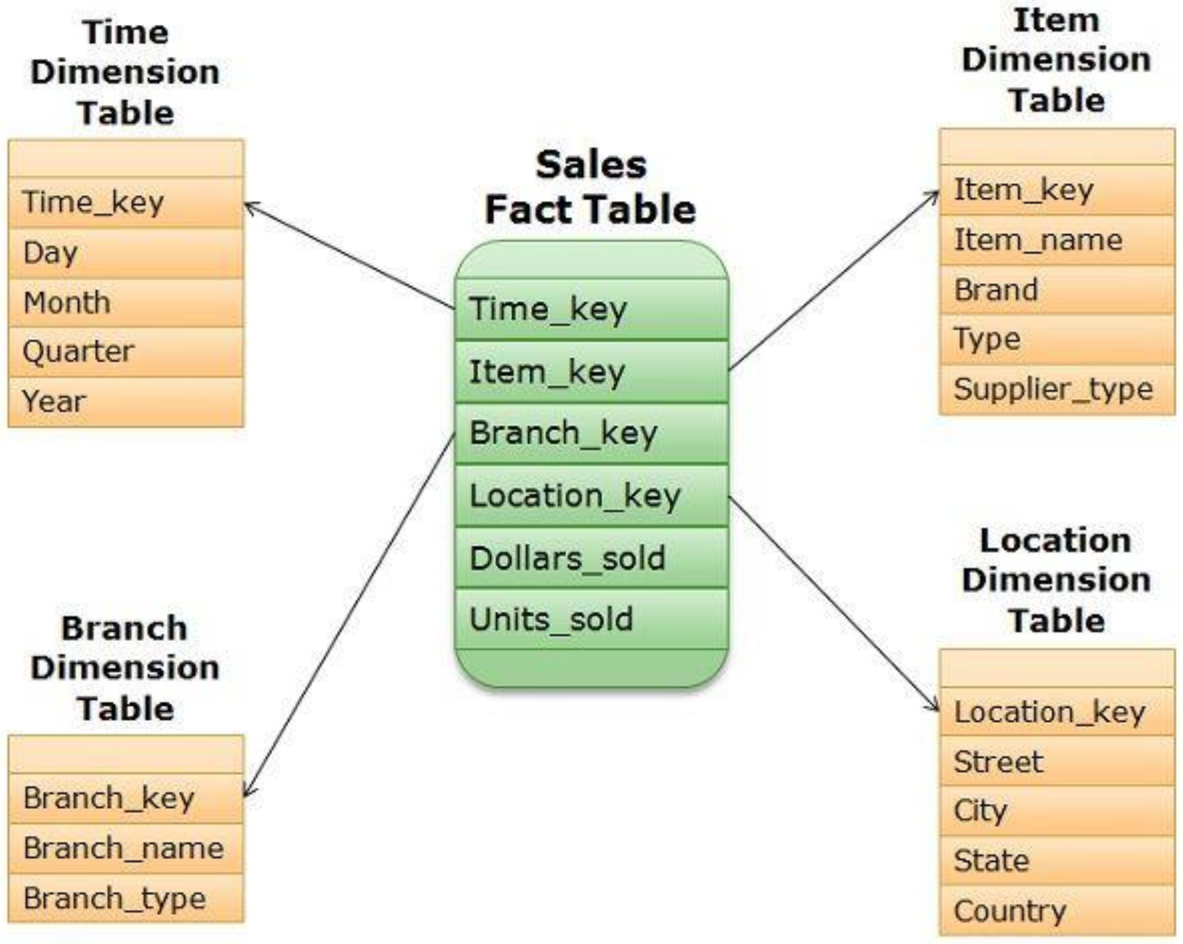
Table variable cannot be used as an input or an output parameter.

Definition of Star Schema

Star schema is the simple and common modeling paradigm where the data warehouse comprises of a fact table with a single table for each dimension. The schema imitates a star, with dimension table presented in an outspread pattern encircling the central fact table. The dimensions in fact table are connected to dimension table through primary key and foreign key.

EXAMPLE:

We are creating a schema which includes the sales of an electronic appliance manufacturing company. Sales are intended along following dimensions: time, item, branch, and location. The schema contains a central fact table for sales that includes keys to each of the four dimensions, along with two measures: dollar-sold and units-sold. The capacity of the fact table is reduced by the generation of dimension identifiers such as time_key and item_key via the system.



Star Schema Example Diagram

Only a single table imitates each dimension, and each table contains a group of attributes as it is shown in the star schema. The location dimension table encompasses the attribute set {location_key, street, city, state and country}. This restriction may introduce some redundancy. For example, two cities can be of same state and country, so entries for such cities in the location dimension table will create redundancy among the state and country attributes.

DEFINITION OF SNOWFLAKE SCHEMA

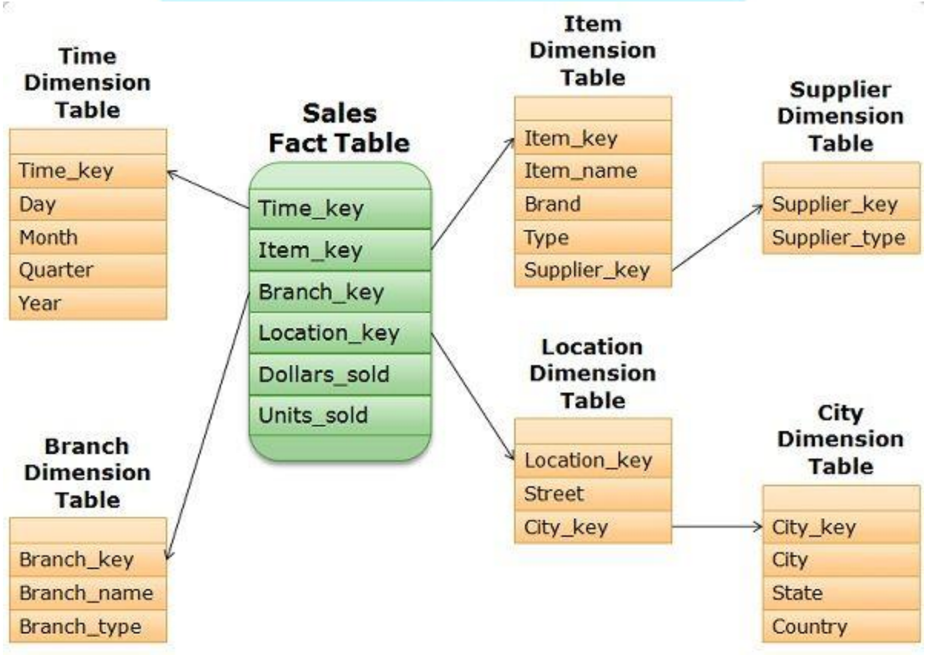
Snowflake schema is the kind of the star schema which includes the hierarchical form of dimensional tables. In this schema, there is a fact table comprise of various dimension and sub-dimension table connected across through primary and foreign key to the fact table. It is named as the snowflake because its structure is similar to a snowflake.

It uses normalization which splits up the data into additional tables. The splitting results in the reduction of redundancy and prevention from memory wastage. A snowflake schema is more easily managed but complex to design and understand. It can also reduce the efficiency of browsing since more joins will be required to execute a query.

Example:

In the snowflake schema, we are taking the same example as we have taken in the star schema. Here the sales fact table is identical to that of the star schema, but the main difference lies in the definition of dimension tables.

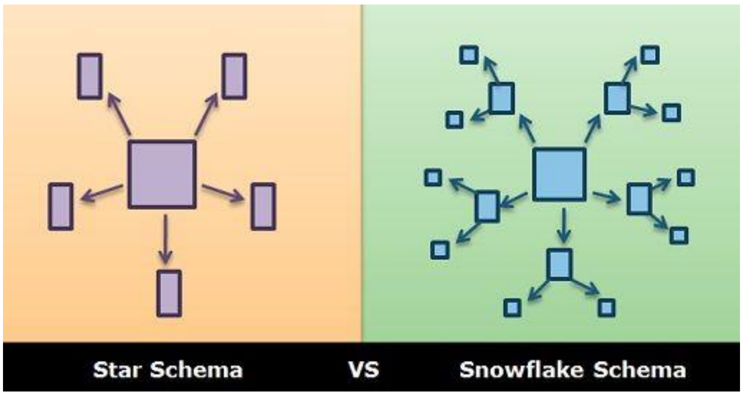
The single dimension table for the item in the star schema is normalized in the snowflake schema, results in creation of new item and supplier tables. For instance, the item dimension table comprised of the attributes item_key, brand, item_name, type, and supplier_key, where supplier_key is connected to the supplier dimension table, which holds supplier_key and supplier_type information.



Snowflake Schema Example Diagram

Similarly, the location dimension table involves the attributes location_key, street, and city_key, and city_key is linked to city dimension table containing the city, state and country attribute. Here state attribute can also further normalized.

Difference between Star and Snowflake Schema



Star and snowflake schemas are the most popular multidimensional data models used for a data warehouse. The crucial difference between Star schema and snowflake schema is that star schema does not use normalization whereas snowflake schema uses normalization to eliminate redundancy of data. Fact and dimension tables are essential requisites for creating schema.

Key Differences between Star and Snowflake Schema

1. Star schema contains just one dimension table for one dimension entry while there may exist dimension and sub-dimension table for one entry.
2. Normalization is used in snowflake schema which eliminates the data redundancy. As against, normalization is not performed in star schema which results in data redundancy.
3. Star schema is simple, easy to understand and involves less intricate queries. On the contrary, snowflake schema is hard to understand and involves complex queries.
4. The data model approach used in a star schema is top-down whereas snowflake schema uses bottom-up.
5. Star schema uses a fewer number of joins. On the other hand, snowflake schema uses a large number of joins.
6. The space consumed by star schema is more as compared to snowflake schema.
7. The time consumed for executing a query in a star schema is less. Conversely, snowflake schema consumes more time due to the excessive use of joins.

Star and Snowflake schema is used for designing the data warehouse. Both have certain merits and demerits where snowflake schema is easy to maintain, lessen the redundancy hence consumes less space but complex to design. Whereas star schema is simple to understand and design, uses less number of joins and simple queries but have some issues such as data redundancy and integrity.

However, use of snowflake schema minimizes redundancy, but it is not popular as star schema which is used most of the time in the design of data warehouse.