WORKSHEET 1 SQL

Q1 and Q2 have one or more correct answer. Choose all the correct option to answer your question.

1. Which of the following is/are DDL commands in SQL?
   A) Create B) Update
   C) Delete D) ALTER
2. Which of the following is/are DML commands in SQL?
   A) Update B) Delete
   C) Select D) Drop

Q3 to Q10 have only one correct answer. Choose the correct option to answer your question.

3. Full form of SQL is:
   A) Strut querying language B) Structured Query Language
   C) Simple Query Language D) None of them
4. Full form of DDL is:
   A) Descriptive Designed Language B) Data Definition Language
   C) Data Descriptive Language D) None of the above.
5. DML is:
   A) Data Manipulation Language B) Data Management Language
   C) Data Modeling Language D) None of these
6. Which of the following statements can be used to create a table with column B int type and C float type?
   A) Table A (B int, C float) B) Create A (b int, C float)
   C) Create Table A (B int, C float) D) All of them
7. Which of the following statements can be used to add a column D (float type) to the table A created above?
   A) Table A (D float) B) Alter Table A ADD COLUMN D float
   C) Table A (B int, C float, D float) D) None of them
8. Which of the following statements can be used to drop the column added in the above question?
   A) Table A Drop D B) Alter Table A Drop Column D
   C) Delete D from A D) none of them
9. Which of the following statements can be used to change the data type (from float to int) of the column D of table A created in above questions?
   A) Table A (D float int) B) Alter Table A Alter Column D int
   C) Alter Table A D float int D) Alter table A Column D float to int
10. Suppose we want to make Column B of Table A as primary key of the table. By which of the following statements we can do it?
    A) Alter Table A Add Constraint Primary Key B B) Alter table (B primary key)
    C) Alter Table A Add Primary key B D) None of them

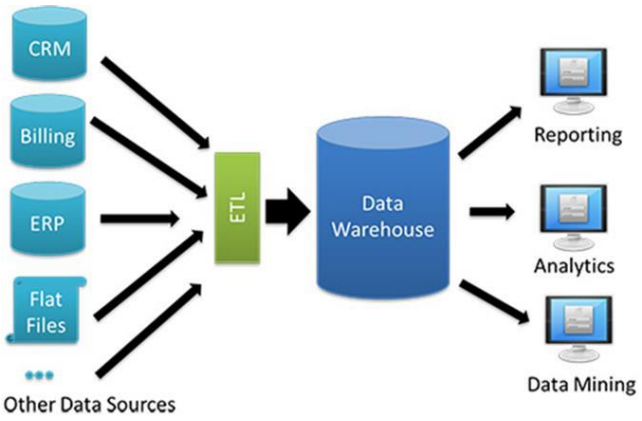
Q11 to Q15 are subjective answer type questions, Answer them briefly.

11. What is data-warehouse?

Definition: A data warehouse is a system that pulls together data from many different sources within an organization for reporting and analysis. The reports created from complex queries within a data warehouse are used to make business decisions.

A data warehouse stores historical data about your business so that you can analyze and extract

insights from it. It does not store current information, nor is it updated in real-time.



Thus a Data Warehouse is defined as a central repository where information is coming from one or more data sources. It allows business users to quickly access critical data from some sources all in one place.

How Data-warehouse works?

A Data Warehouse works as a central repository where information arrives from one or more data sources. Data flows into a data warehouse from the transactional system and other relational databases.

Data may be:

1. Structured
2. Semi-structured
3. Unstructured data

The data is processed, transformed, and ingested so that users can access the processed data in the Data Warehouse (DW) through Business Intelligence tools, SQL clients, and spreadsheets. A data warehouse merges information coming from different sources into one comprehensive database.
By merging all of this information in one place, an organization can analyze its customers more holistically. This helps to ensure that it has considered all the information available. Data warehousing makes data mining possible. Data mining is looking for patterns in the data that may lead to higher sales and profits.

Types of Data Warehouse

Three main types of Data Warehouses (DWH) are:

1. Enterprise Data Warehouse (EDW):
Enterprise Data Warehouse (EDW) is a centralized warehouse. It provides decision support service across the enterprise. It offers a unified approach for organizing and representing data. It also provides the ability to classify data according to the subject and give access according to those divisions.

2. Operational Data Store:
Operational Data Store, which is also called ODS, are nothing but data store required when neither Data warehouse nor OLTP systems support organizations reporting needs. In ODS, Data warehouse is refreshed in real time. Hence, it is widely preferred for routine activities like storing records of the Employees.

3. Data Mart:
A data mart is a subset of the data warehouse. It specially designed for a particular line of business, such as sales and finance. In an independent data mart, data can be collected directly from sources.

Components of Data warehouse

Four components of Data Warehouses are:

1. **Load manager**: Load manager is also called the front component. It performs with all the operations associated with the extraction and load of data into the warehouse. These operations include transformations to prepare the data for entering into the Data warehouse.

2. **Warehouse Manager**: Warehouse manager performs operations associated with the management of data in the warehouse. It performs operations like analysis of data to ensure consistency, creation of indexes and views, generation of demoralization and aggregations, transformation and merging of source data and archiving and baking-up data.

3. **Query Manager**: Query manager is also known as backend component. It performs all the operation operations related to the management of user queries. The operations of these Data warehouse components are direct queries to the appropriate tables for scheduling the execution of queries.

4. **End-user access tools:** This is categorized into five different groups like 1. Data Reporting 2. Query Tools 3. Application development tools 4. EIS tools, 5. OLAP tools and data mining tools.

Steps to Implement Data Warehouse

Implementing Data-warehouse is a 3 prong strategy viz. Enterprise strategy, Phased delivery and Iterative Prototyping.

1. **Enterprise strategy**: Here we identify technical things including current architecture and tools. We also identify facts, dimensions, and attributes. Data mapping and transformation is also passed.

2. **Phased delivery**: Data-warehouse implementation should be phased based on subject areas. Related business entities like booking and billing should be first implemented and then integrated with each other.

3. **Iterative Prototyping**: Rather than a big bang approach to implementation, the Data-warehouse should be developed and tested iteratively.

Advantages of Data Warehouse (DWH):

1. Data warehouse allows users to quick access to critical data from multiple sources all in one place.
2. Provides consistent information on various cross-functional activities.
3. Data Warehouse helps to integrate many sources of data to reduce stress on the production system.
4. Reduces total turn-around time for analysis and reporting.
5. Restructuring and Integration make it easier for the user to use for reporting and analysis.
6. Data warehouse stores a large amount of historical data. This helps users to analyze different time periods and trends to make future predictions.

Disadvantages of Data Warehouse:

1. Not an ideal option for unstructured data.
2. Creation and Implementation of Data Warehouse are surely time confusing.
3. Data Warehouse can be outdated relatively quickly
4. Difficult to make changes in data types and ranges, data source schema, indexes, and queries.
5. Sometime warehouse users will develop different business rules.
6. Organizations need to spend lots of resources for training and Implementation purpose.
7. The data warehouse may seem easy, but actually, it is too complex for the average users.

Most common sectors where Data warehouse is used:

- Airline
- Banking
- Healthcare
- Public sector

- Telecommunication

12. What is the difference between OLTP VS OLAP?

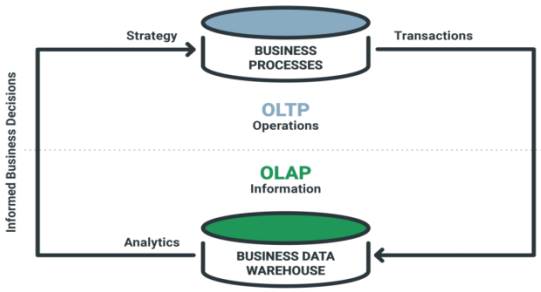
OLAP

OLAP is an Online Analytical Processing system. It is used for analysis of database information from multiple database systems at one time
OLAP database stores historical data that has been inputted by OLTP. It allows a user to view different summaries of multi-dimensional data. Using OLAP, you can extract information from a large database and analyze it for decision making.
OLAP also allow a user to execute complex queries to extract multidimensional data. In OLTP even if the transaction fails in middle it will not harm data integrity as the user use OLAP system to retrieve data from a large database to analyze. Simply the user can fire the query again and extract the data for analysis.
The transaction in OLAP is long and hence takes comparatively more time for processing and requires large space. The transactions in OLAP are less frequent as compared to OLTP. Even the tables in OLAP database may not be normalized.

Examples – Any type of Data warehouse system is an OLAP system. Other examples for OLAP include viewing a financial report, or budgeting, market research, sales report, etc.

Uses of OLAP are as follows:

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



OLTP:

OLTP is an **Online Transaction Processing** system. The main focus of OLTP system is to record the current Update, Insertion and Deletion while transaction.
The OLTP queries are simpler and short and hence require less time in processing, and also require less space.
OLTP database gets updated frequently. It may happen that a transaction in OLTP fails in middle, which may affect data integrity. So, it has to take special care of data integrity. OLTP database has normalized tables (3NF).
The best **example** for OLTP system is an ATM, in which using short transactions we modify the status of our account. OLTP system becomes the source of data for OLAP.

Uses of OLTP are as follows:

- ATM center is an OLTP application.
- It's also used for Online banking,
- Online airline ticket booking,
- sending a text message,
- Add a book to the shopping cart.

Comparison table:

	Parameters	OLAP	OLTP
1.	Definition	OLAP is an **analytical processing** system.	OLTP is a **transactional processing**
2.	Data type	Consists of **historical data** from various databases.	Consists of only **operational data**
3.	focus	It is **subject oriented**, used for data analysis. For example: Database design changes with subjects like sales, marketing, purchasing, etc.	**Application oriented**, for example: Database design changes with industry like Retail, Airline, Banking, etc.
4.	Method	OLAP uses **data-warehouse**	OLTP uses traditional **DBMS**
5.	Primary objective	**Data analysis**. The data is used in planning, problem solving and decision making.	**Data processing**. The data is used to perform day to day fundamental operations.
6.	characteristic	OLAP is characterized by a **large volume of data**.	OLTP is characterized by large numbers of **short online transactions.**
7.	Transactions	OLAP has **long transactions**.	OLTP has **short transactions.**
8.	Processing Time	The **processing time of a transaction is comparatively more** in OLAP as the amount of data involved is large.	The processing time of a transaction is comparatively **less** in OLTP.
9.	Queries	**Complex** queries.	**Simple** queries.
10.	Backup and recovery	It only needs backup **from time to time** as compared to OLTP.	Backup and recovery process is **maintained religiously.**
11.	Performance metric	**Query throughput** is the performance metric.	**Transaction throughput** is the performance metric.
12.	Read write operations	**Only read and rarely write** operation.	**Both read and write** operations.
13.	Table Normalization	Tables in OLAP database are **not normalized.**	Tables in OLTP database are **normalized (3NF).**
14.	Data integrity	OLAP database does not get frequently modified. Hence, **data integrity is not affected.**	OLTP database **must maintain data integrity constraint.**

13. What are the various characteristics of data-warehouse?

A data warehouse is a relational or multidimensional database that is designed for query and analysis. Data warehouses are not optimized for transaction processing, which is the domain of OLTP systems. Data warehouses usually consolidate historical and analytic data derived from multiple sources. Data warehouses separate analysis workload from transaction workload and enable an organization to consolidate data from several sources.

The Key Characteristics of a Data Warehouse

The key characteristics of a data warehouse are as follows:

1. **Subject-oriented** –A data warehouse is always a subject oriented as it delivers information about a theme instead of organization's current operations. It can be achieved on specific theme. That means the data warehousing process is proposed to handle with a specific theme which is more defined. These themes can be sales, distributions, marketing etc.
   Subject orientation can be really useful for decision making. Gathering the required objects is called subject-oriented.

2. **Integrated** –Integration means finding a shared entity to scale all the similar data from different databases. A data warehouse is built by integrating data from various sources of data .Integration of data warehouse aids in effective analysis of data. Since data comes from several operational systems, all inconsistencies must be removed. Consistencies include naming conventions,

measurement of variables, encoding structures, physical attributes of data, and so forth.

3. **Time-Variant –**In this data is maintained via different intervals of time such as weekly, monthly, or annually etc. It finds various time limits which are structured between the large datasets and are held in online transaction process (OLTP). The data resided in data warehouse is predictable with a specific interval of time and delivers information from the historical perspective. Another feature of time-variance is that once data is stored in the data warehouse then it cannot be modified, alter, or updated.

4. **Non-Volatile** –As the name defines the data resided in data warehouse is permanent. It also means that data is not erased or deleted when new data is inserted. In this, data is read-only and refreshed at particular intervals. This is beneficial in analyzing historical data and in comprehension the functionality. It does not need transaction process, recapture and concurrency control mechanism. Functionalities such as delete, update, and insert that are done in an operational application are lost in data warehouse environment.

14. What is Star-Schema?

Data warehouse models are especially modeled using Multidimensional schema. The schemas are designed to address the unique needs of very large databases designed for the analytical purpose (OLAP).

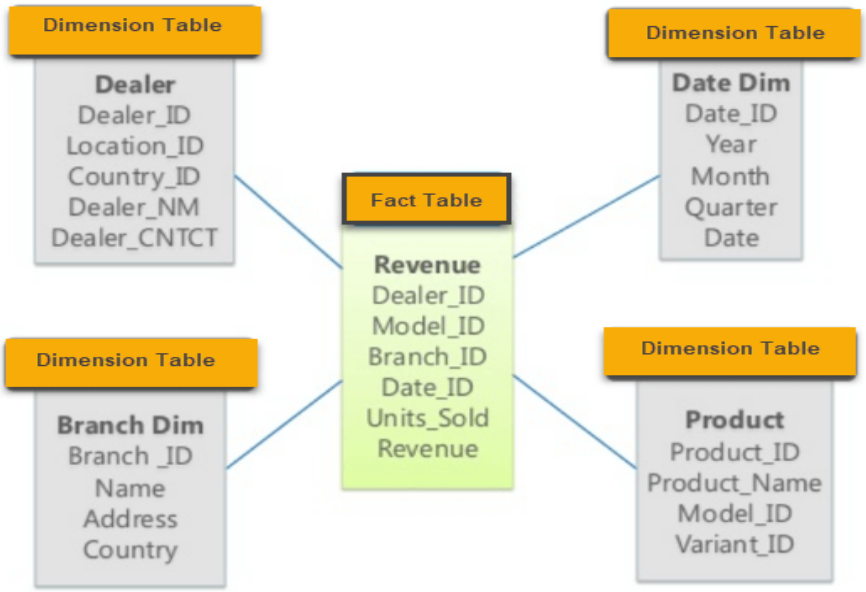
Types of Data Warehouse Schema:

Following are 3 chief types of multidimensional schemas each having its unique advantages.

1. Star Schema
2. Snowflake Schema
3. Galaxy Schema

Star Schema in data warehouse, in which the center of the star can have one fact table and a number of associated dimension tables. It is known as star schema as its structure resembles a star shape having a fact table at its center and the dimension tables at its peripheral representing the star's points. The Star Schema data model is the simplest type of Data Warehouse schema. It is also known as Star Join Schema and is optimized for querying large data sets.

In the following Star Schema example, the fact table is at the center which contains keys to every dimension table like Dealer_ID, Model ID, Date_ID, Product_ID, Branch_ID & other attributes like Units sold and revenue.



Example of Star Schema Diagram

Fact tables record measurements or metrics for a specific event. Fact tables generally consist of numeric values, and foreign keys to dimensional data where descriptive information is kept.

Dimension tables usually have a relatively small number of records compared to fact tables, but each record may have a very large number of attributes to describe the fact data. Dimensions can define a wide variety of characteristics,

Characteristics of Star Schema:

- Every dimension in a star schema is represented with the only one-dimension table.
- The dimension table should contain the set of attributes.
- The dimension table is joined to the fact table using a foreign key
- The dimension tables are not joined to each other
- Fact table would contain key and measure
- The Star schema is easy to understand and provides optimal disk usage.
- The dimension tables are not normalized. For instance, in the above figure, Country_ID does not have Country lookup table as an OLTP design would have.
- The schema is widely supported by BI Tools.
- Simple DB Design.
- It provides a flexible design that can be changed easily or added to throughout the development cycle, and as the database grows.

Advantages of Star Schema –

1. **Simpler Queries**:
   Join logic of star schema is quite cinch in compare to other join logic which are needed to fetch data from a transactional schema that is highly normalized.

2. **Simplified Business Reporting Logic:**
   As compared to a transactional schema that is highly normalized, the star schema makes simpler common business reporting logic, such as as-of reporting and period-over-period.

3. **Feeding Cubes:**
   Star schema is widely used by all OLAP systems to design OLAP cubes efficiently. In fact, major OLAP systems deliver a ROLAP mode of operation which can use a star schema as a source without designing a cube structure.

4. **Query performance gains** – A star schema database has a limited number of tables and clear join paths, the query run faster than they do against OLTP systems. Star schemas can provide performance enhancements for read-only reporting applications when compared to highly normalized schemas.

5. **Fast aggregations** – the simpler queries against a star schema can result in improved performance for aggregation operations

Disadvantages of Star Schema –

1. Data integrity is not enforced well due to its denormalized state.

2. Not flexible in terms of analytical needs as a normalized data model.

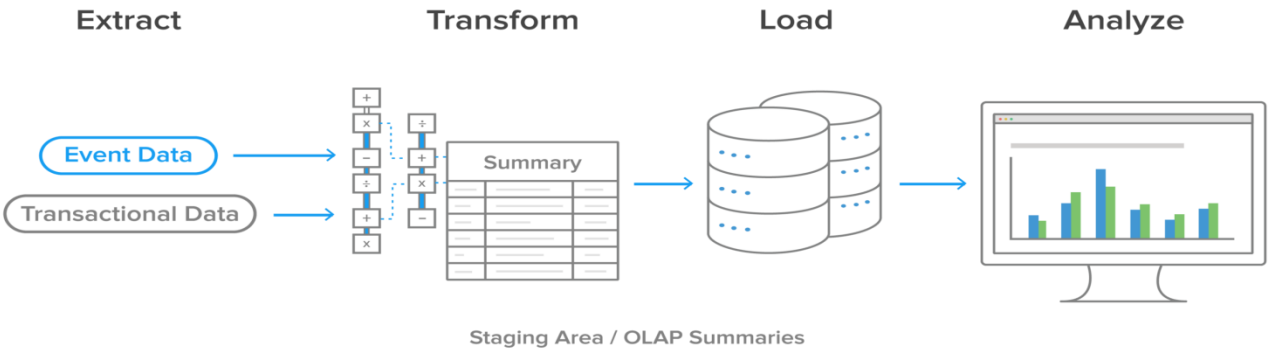
3. Star schemas don't reinforce many-to-many relationships within business entities – at least not frequently.

15. What do you mean by SETL?

Before we study SETL let us understand the ETL (Extract-Transform-Load)

ETL is the process by which data is extracted from data sources (that are not optimized for analytics), and moved to a central host .The exact steps in that process might differ from one ETL tool to the next, but the end result is the same.

At its most basic, the ETL process encompasses data extraction, transformation, and loading. While the abbreviation implies a neat, three-step process – extract, transform, load.



The ETL process: extract, transform and load. Then analyze.

1) **Extract**: Data is extracted from online transaction processing (OLTP) databases, today more commonly known just as 'transactional databases', and other data sources.
2) **Transform**: Data is then transformed in a staging area. These transformations cover both data cleansing and optimizing the data for analysis.
3) **Load**: The transformed data is then loaded into an online analytical processing (OLAP) database, today more commonly known as just an analytics database.

Thus the integration process for extracting data from different sources, translating it according to the underlying semantics of the DW, and loading it into the DW is known as Extract-Transform-Load (ETL).

Business intelligence (BI) teams then run queries on that data, which are eventually presented to end users, or to individuals responsible for making business decisions, or used as input for machine learning algorithms or other data science projects.

SETL:

1. SETL is known as **semantic ETL (Extract-Transform-Load).** Semantic ETL (SETL) is a unified framework for processing and integrating data semantically by bridging SW and DW technologies.

2. In order to create better decisions for business analytics, organizations increasingly use external structured, semi-structured, and unstructured data in addition to the (mostly structured) internal data.

3. Current Extract-Transform-Load (ETL) tools are not suitable for this “open world scenario” because they do not consider semantic issues in the integration processing.

4. Current ETL tools neither support processing semantic data nor create a semantic Data Warehouse (DW), a repository of semantically integrated data.

5. Thus SETL i.e. programmable Semantic ETL (SETL) framework comes in picture. SETL uses and extends SW tools and standards to overcome the limitations of the traditional ETL tools.

6. The SETL system is designed and implemented in order to extract, transform and load large scale data from heterogeneous data sources into data warehouse efficiently.

7. It allows to process both semantic and Non Semantic Data Sources. In order to process a Non Semantic Data Source, it builds a semantic layer on top of the Non Semantic Data Source

8. SETL builds on Semantic Web (SW) standards and tools and supports developers by offering a number of powerful modules, classes, and methods for data- warehouse constructs and tasks.

9. Thus it supports semantic data sources in addition to traditional data sources, semantic integration, and creating or publishing a semantic (multidimensional) DW in terms of a knowledge base.

SETL FRAMEWORK

SETL is a unified framework for semantic ETL. The framework is divided into three layers:

1. The Definition Layer,
2. ETL Layer, and
3. Data Warehouse Layer.

1) **Definition Layer**: This layer integrates different components required to define the schema of a SDW, to define the different sources that feed data in the SDW, and to define the mappings between the sources and the target.

2) **ETL Layer**: This layer integrates different components to design an ETL process to populate the SDW from different data sources

3) **Data Warehouse Layer: the** Data Warehouse Layer manages the storage of transformed semantic data.

Understanding the concept of semantic ETL:

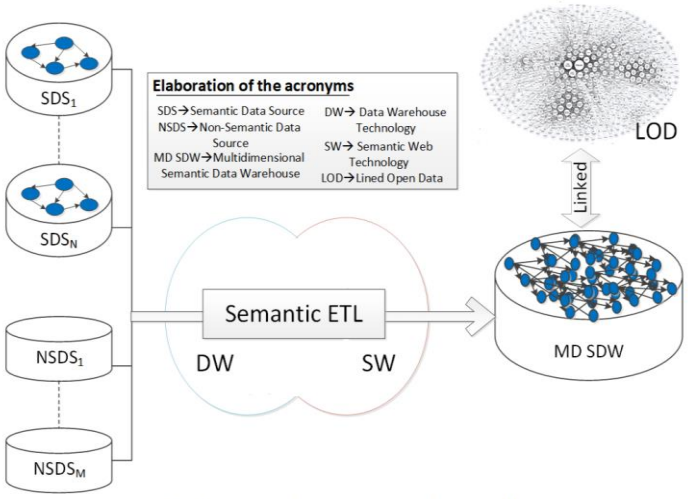


Fig. 1: Understanding the concept of Semantic ETL

The semantic ETL covers the following aspects:

1. Definition of an SDW TBox with MD semantics.

2. Data extraction from semantic and non-semantic sources.

3. Definition of source-to-target mappings to automatically parameterize the ETL operations.
4. A set of high-level conceptual ETL tasks/operations to integrate semantic sources into an SDW.
5. Data transformation according to the (MD) semantics encoded into the target TBox and the source-to-target mapping file.
6. Update of the SDW to reflect the data changed in the source over time.
7. Linking the internal data with external KBs.
8. Loading the transformed data into a triple store.
9. Enabling OLAP queries on the SDW.

Advantages of SETL

1. SETL can implement ETL job easily and efficiently.
2. SETL facilitates developers to build a semantic DW using a single platform instead of using different tools and a manual process.
3. SETL provides better programmer productivity, knowledge base quality, and performance
4. SETL provides functionality to produce semantic data from the source data.
5. Using SETL the users can write their own code to create an ETL process in Python. Therefore, it allows extensibility.

THANK-YOU.