Data Warehouse Sample Questions:

Data Warehousing

A single, complete and consistent store of data obtained from a variety of different sources made available to end users in what they can understand and use in a business context.

A data warehouse is a collection of integrated databases designed to support a DSS.

A data warehouse is

subject-oriented,

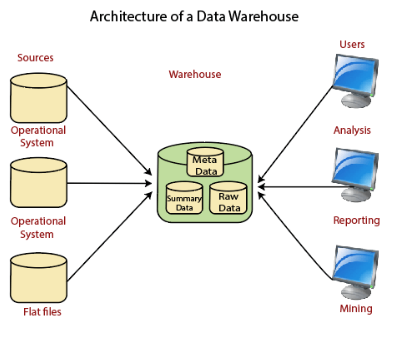
integrated,

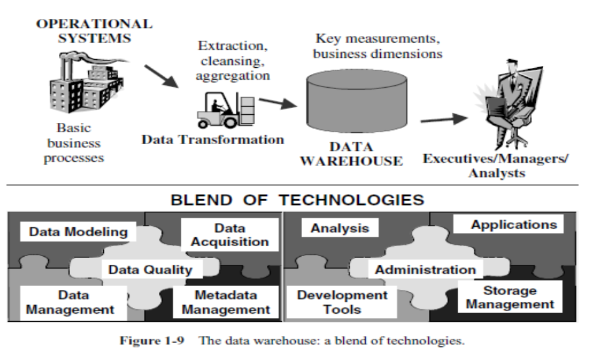
time-variant,

nonvolatile

collection of data in support of management’s decision making process.

Technique for assembling and managing data from various sources for the purpose of answering business questions. Thus making decisions that were not previously possible. A decision support database maintained separately from the organization’s operational database.





**Data Warehouse Users:**

**Casual or novice user:** Uses the data warehouse occasionally and needs a very intuitive information interface.

**Regular user:** Uses the data warehouse almost daily .

These users are comfortable with computing options but can not create reports and queries on their own and thus make use of query templates and predefined reports.

**Power users:**

proficient with technology.

Capable of creating reports and executing query.

**Executives and Managers:**

They need information for making high level strategic decisions.

They prefer customized and personalized reports.

**Technical analyst:**

They perform complex analysis and statistical analysis,perform drill down and rollup,slice and dice operations on data.

**Business analyst:**

Comfortable with technology.

They rely on predefined queries and reports to satisfy their requirement.

What is a data warehouse? Explain its characteristics/features.

A data warehouse, or enterprise data warehouse (EDW), is a system that aggregates data from different sources into a single, central, consistent data store to support data analysis, data mining, artificial intelligence (AI), and machine learning. A data warehouse system enables an organization to run powerful analytics on huge volumes (petabytes and petabytes) of historical data in ways that a standard database cannot.(IBM website)

Features

* subject-oriented
* integrated
* time-variant
* non-volatile

**Subject-oriented:**

* Data warehouses are designed to help you analyze your data.

Data warehouse is organized around subjects such as sales etc.

Subject-oriented means data is organized by business topic not by business processes.

* For example, you might want to learn more about your company's sales data. To do this, you could build a warehouse concentrating on sales. In this warehouse, you could answer questions like "Who was our best customer for this item last year?" This kind of focus on a topic, sales in this case, is what is meant by subject oriented.
* A data warehouse is subject oriented because it provides information around a subject rather than the organization's ongoing operations.

These subjects can be product, customers, suppliers, sales, revenue, etc.

* A data warehouse does not focus on the ongoing operations, rather it focuses on modeling and analysis of data for decision making.
* Data warehousing systems are generally subject-oriented, organized around business areas that the organization needs information about.
* In DW the subject is the organization method.

Subjects vary with enterprise.

These are critical factors that affect performance.

Such subject areas are usually populated with data from one or more operational systems.

As an example Claims is a critical business subject for an insurance company.

Sales for retail companies etc.

Manufacturing Company

Sales

Shipment

Inventory etc.

**Integration**

* A data warehouse is constructed by integrating data from

heterogeneous sources such as relational databases, flat files, etc.

* Data is stored as a single unit , not as a collection of files that may have different structures or organizations.

This integration enhances the effective analysis of data.

Data Preprocessing is applied to ensure consistency.

* Data comes from several applications
* Problems of integration comes into play

File layout, encoding, field names, systems, schema, data heterogeneity are the

issues

Bank example, variance: naming convention, attributes for data item, account no, account type, size, currency

* In addition to internal, external data sources

External companies data sharing

Websites

Others

* Removal of inconsistency

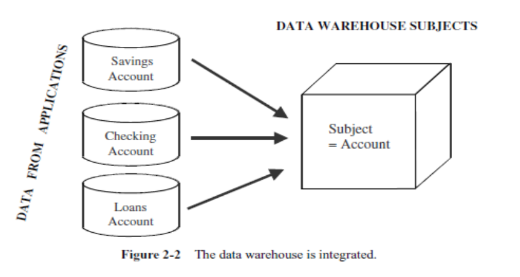
Naming conventions

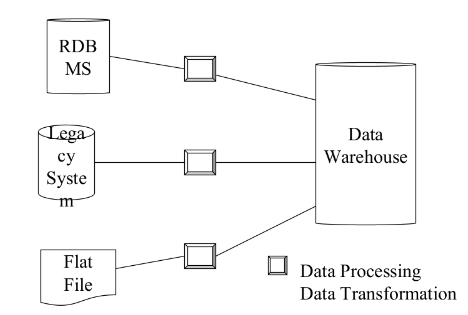
Codes

Data attributes

Measurements

So the process of extraction, transformation & loading.





**Time-variant**

Time variant means that the time dimension is explicitly

included in the data so that trends and changes over time

can be studied.

Time variant simply means that the data available were

identified on a particular period.

Every key structure contains either implicitly or explicitly

an element of time.

Provides information from historical perspective e.g. past

5-10 years

Operational data has current values

Comparative analysis is one of the best techniques for business performance evaluation.

Time is a critical factor for comparative analysis.

Every data structure in DW contains a time element.

In order to promote a product, analysts have to know about current and historical values.

The advantages are:-

* Allows for analysis of the past
* Relates information to the present
* Enables forecasts for the future

**Nonvolatile**

Data doesn't keep changing. New data may be added on a

schedule basis , but old data isn't discarded.

Data once recorded cannot be updated.

Data warehouse requires two operations in data accessing

* Initial loading of data
* Access of data

Data from operational systems are moved into DW after specific

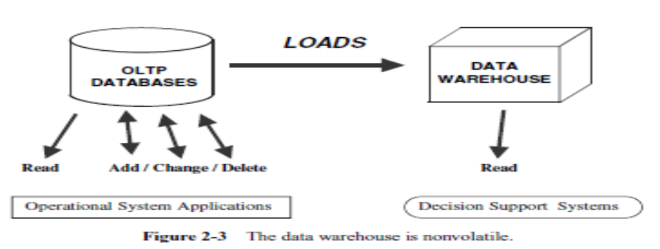
intervals

Data is persistent/ not removed i.e. non volatile

Every business transaction don’t update in DW

Data from DW is not deleted

Data is neither changed by individual transactions



Explain the advantages of using a Data warehouse.

Benefits of Data Warehouse

DW enables end users to access a wide variety of data.

Business analysts and decision makers can analyze the current trends in the market to predict future trends.

For example, the analyst can analyze the product's sales in a particular area for the last two years. This may be helpful for future investment in a particular item.

DW provides consistent data.

It helps to increase productivity and decrease the computing cost.

DW contains data that has been integrated from a number of sources.

The results obtained can be presented in a variety of formats in the form of reports,graphs etc.

DW users can obtain trend reports , for eg. The product that had maximum sales in the Northern region within the last two years and an exception report that shows actual performance versus goal.

DW enhances the value of the operational business applications.eg.CRM

DW are special types of databases that are specifically built for the

propose getting information out rather than putting data in.

DW provides answers to strategic questions and assist the managers of

organizations in planning for the future .

The data warehouse is an informational environment that provides an

integrated and total view of the enterprise.

Makes the enterprise’s current and historical information easily available for decision making.

Makes decision-support transactions possible without hindering

operational systems.

Renders the organization’s information consistent.

Presents a flexible and interactive source of strategic information.

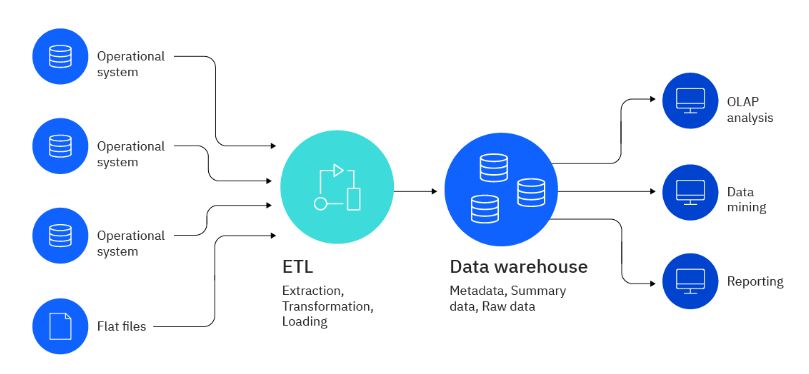
Although a simple concept, it involves different functions: data extraction, the function of loading the data, transforming the data, storing the data,and providing user interfaces.

Explain the following terms:

1)Data warehouse

A single, complete and consistent store of data obtained from a variety of different sources made available to end users in what they can understand and use in a business context.

A data warehouse is a collection of integrated databases designed to support a DSS.



2) Data Mart

* Subset of the enterprise wide data warehouse.
* A data mart is the access layer of the [data warehouse](http://en.wikipedia.org/wiki/Data_warehouse) environment that is used to get data out to the users.
* Departmental data from Data Warehouse is called Data Mart.
* Eg. Finance Department Data mart

Sales Department Data Mart.

Marketing Department Data Mart.

* A data mart is a subset of a data warehouse focused on a particular line of business, department, or subject area. Data marts make specific data available to a defined group of users, which allows those users to quickly access critical insights without wasting time searching through an entire data warehouse. For example, many companies may have a data mart that aligns with a specific department in the business, such as finance, sales, or marketing.(IBM)

Advantages:

* The cost is low.
* Required time to implement is short.
* Controlled locally rather than centrally.
* They contain less information.
* Reduce network traffic

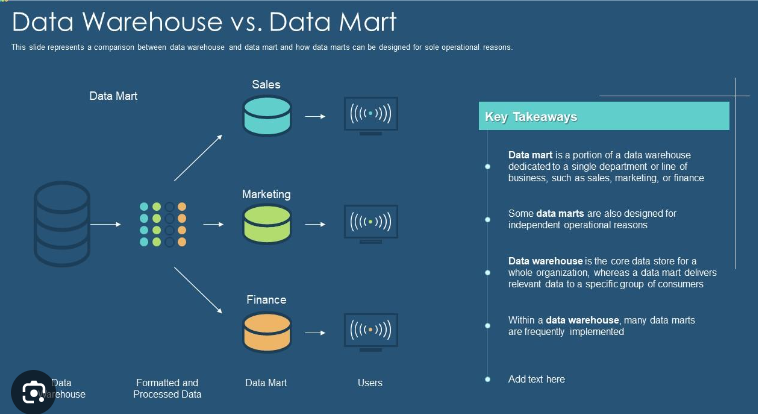
Limitations:

* Performance Degradation as size of the data mart increases.
* Administration becomes difficult.
* Issues in accessing remote Data Mart.

Types of data marts

There are three types of data marts that differ based on their relationship to the data warehouse and the respective data sources of each system.

* **Dependent data marts** are partitioned segments within an enterprise data warehouse. This top-down approach begins with the storage of all business data in one central location. The newly created data marts extract a defined subset of the primary data whenever required for analysis.
* **Independent data marts** act as a standalone system that doesn't rely on a data warehouse. Analysts can extract data on a particular subject or business process from internal or external data sources, process it, and then store it in a data mart repository until the team needs it.
* **Hybrid data marts** combine data from existing data warehouses and other operational sources. This unified approach leverages the speed and user-friendly interface of a top-down approach and also offers the enterprise-level integration of the independent method.



3)Dimensional model

* Dimensional Modeling:
* It is a design technique to structure the business dimensions and metrics.
* Dimensional modeling gets its name from the business dimensions that we incorporate into logical data models.
* The schema for a dimensional model contains a central fact table and multiple dimension tables.
* DM is a logical design technique that seeks to present the data in a standard, intuitive framework that allows for high-performance access.
* Can be implemented using a relational or a multidimensional DBMS
* Every dimensional model is composed of one table with a multipart key, called the fact table, and a set of smaller tables called dimension tables.
* Each dimension table has a single-part primary key that corresponds exactly to one of the components of the multipart key in the fact table.
* This characteristic "star-like" structure is often called a star join. The term star join dates back to the earliest days of relational databases.

**Creating the Dimensional Model**

Identify fact tables

- Translate business measures into fact tables

- Analyze source system information for

additional measures

- Identify base and derived measures

- Document additivity of measures

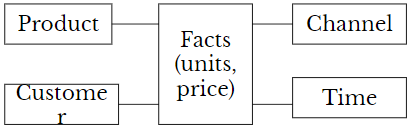
**Identify dimension tables**

Link fact tables to the dimension tables

Create views for users

Dimension tables have the following characteristics:

* Contain textual information that represents the attributes of the business
* Contain relatively static data
* Are joined to a fact table through a foreign key reference

****

* Information that is used for analyzing the elemental data, for example, product hierarchy, time periods, customers, stores
* It is the reference data used for analysis of Facts
* Organizing the information in separate reference tables offers better query performance

5) Dimension

For example, a data warehouse might include a set of facts about a company's product sales. Each sale is a fact that reflects what product was sold, when it was sold, who bought the product, how much it cost and other relevant information.

Dimensions support these facts by providing the background information needed to understand each sale. The data warehouse might include dimensions about products, customers, sales territories, and the dates that various events took place, such as when the product was ordered or shipped.

Dimensions categorize and describe facts and their measures in ways that support meaningful answers to business [queries](https://www.techtarget.com/searchdatamanagement/definition/query). They serve as the fundamental building blocks for developing a [data model](https://www.techtarget.com/searchdatamanagement/definition/data-modeling) that facilitates the efficient analysis of historical data. To this end, dimensions provide the structural underpinnings necessary to make sense of a collection of facts.

6) Measure

In a [data warehouse](https://en.wikipedia.org/wiki/Data_warehouse), a **measure** is a property on which calculations (e.g., sum, count, average, minimum, maximum) can be made. A measure can either be categorical, algebraic or holistic.(wiki)

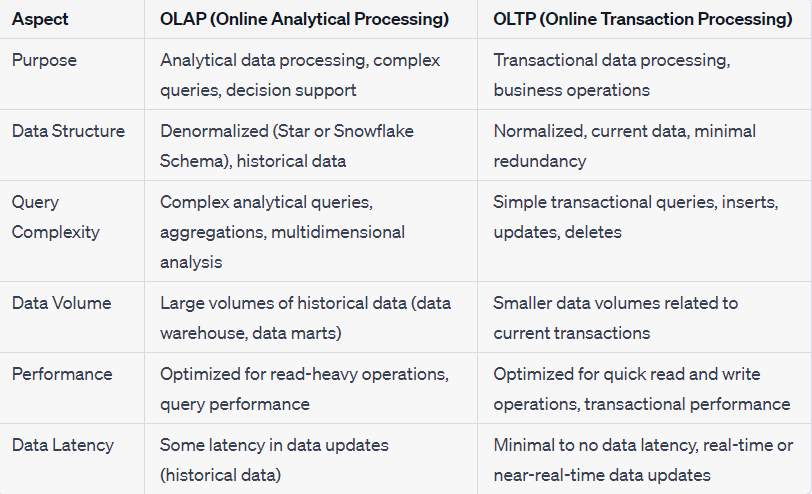
7)Surrogate key

Surrogate keys are primary and foreign keys generated on the data warehouse level instead of being taken out of the source system (business key). Primary (business) key on the source can be e.g. customer number, number of order or product etc.

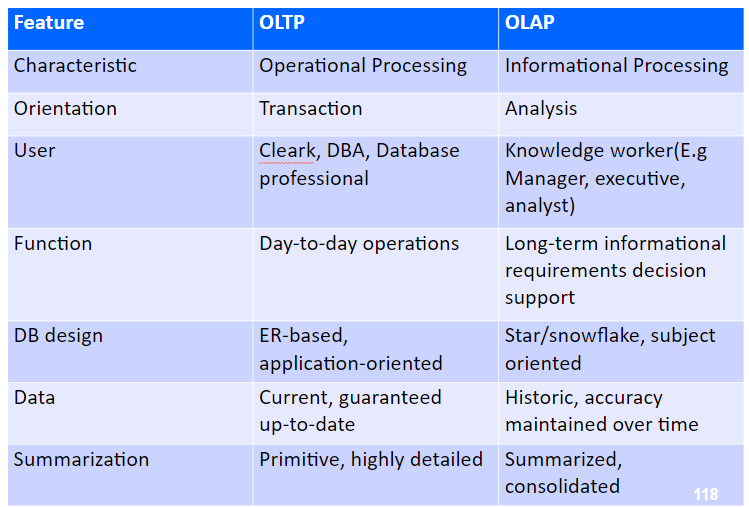
A surrogate key is a unique identifier that is generated and assigned to each row in a data warehouse table, regardless of the source data. Surrogate keys are often used to link dimension tables to fact tables, which store the measures of interest.

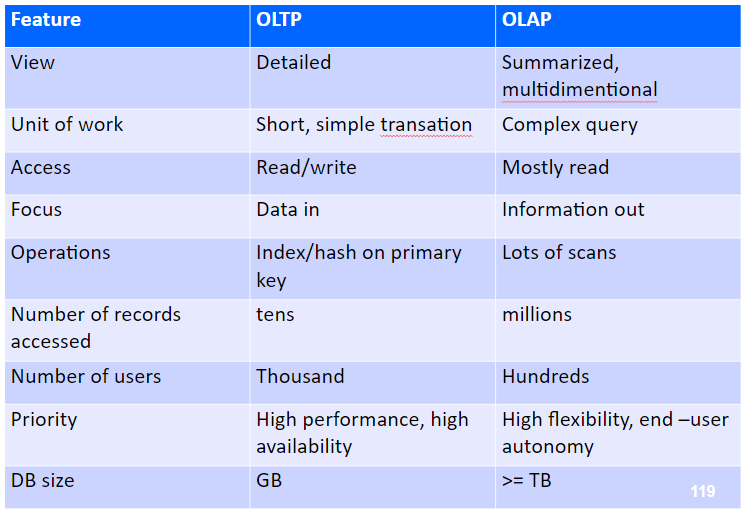
Distinguish following

1) OLAP and OLTP

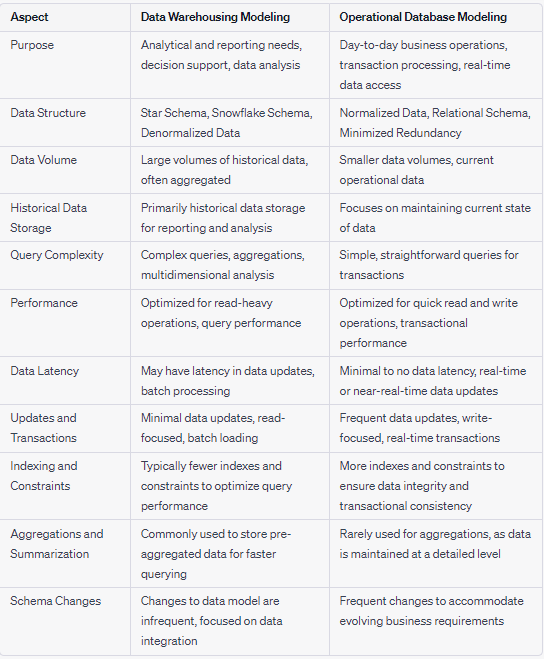


(GPT)



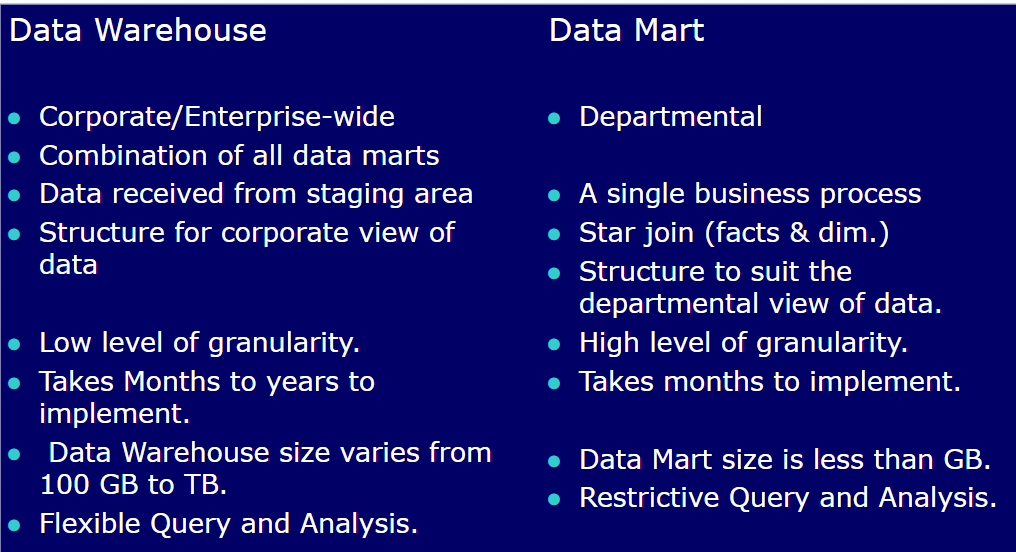


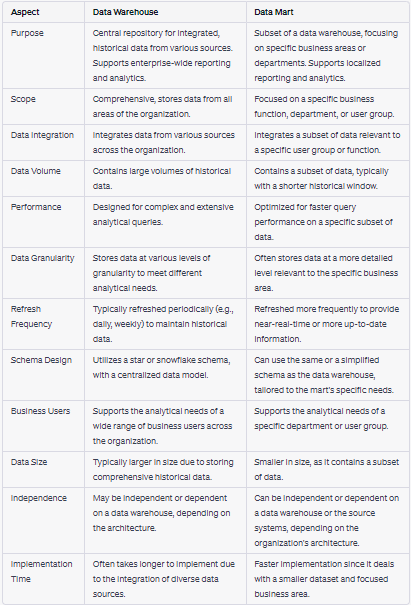
2) Data warehousing modeling and Operational database modeling.



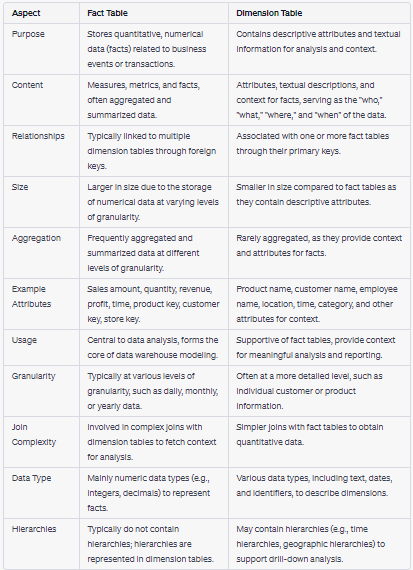
(gpt)

3) Data warehouse and Data Mart.



(GPT)

4) Dimension table and fact table.

(GPT)

Explain Data Mart with examples.(Explained previously)

Explain role of Metadata in Data Warehouse.

* Metadata in a data warehouse is similar to the data dictionary or the data catalog in a database management system.
* In the data dictionary, you keep the information about the logical data structures, the information about the files and addresses, the information about the indexes, and so on.
* The data dictionary contains data about the data in the database.
* Metadata in the data warehouse defines the warehouse objects.
* Metadata acts as a directory. This directory helps the decision support system to locate the contents of a data warehouse.

**Types of Metadata**

* Metadata in a data warehouse fall into three major categories:

**Operational Metadata**

**Extraction and Transformation Metadata**

**End-User Metadata**

* **Operational Metadata.**
* Data for the data warehouse comes from several operational systems of the enterprise.
* These source systems contain different data structures.
* The data elements selected for the data warehouse have various field lengths and data types.
* In selecting data from the source systems for the data warehouse, you split records , combine parts of records from different source files, and deal with multiple coding schemes and field lengths.
* When you deliver information to the end-users, you must be able to tie that back to the original source data sets.
* Operational metadata contains all of this information about the operational data sources.

**Extraction and Transformation Metadata:**

Extraction and transformation metadata contain data about the extraction of data from the source systems, namely, the extraction frequencies, extraction methods, and business rules for the data extraction.

Also, this category of metadata contains information about all the data transformations that take place in the data staging area.

**End-User Metadata.**

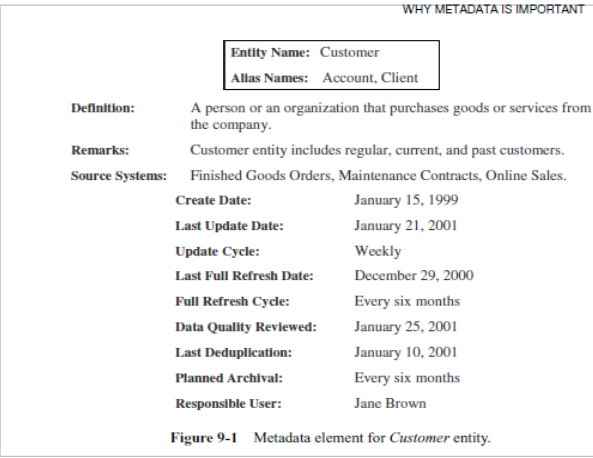
The end-user metadata is the navigational map of the data warehouse.

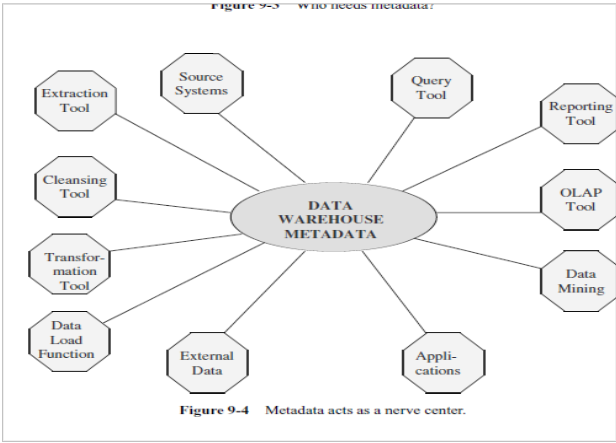
It enables the end-users to find information from the data warehouse.

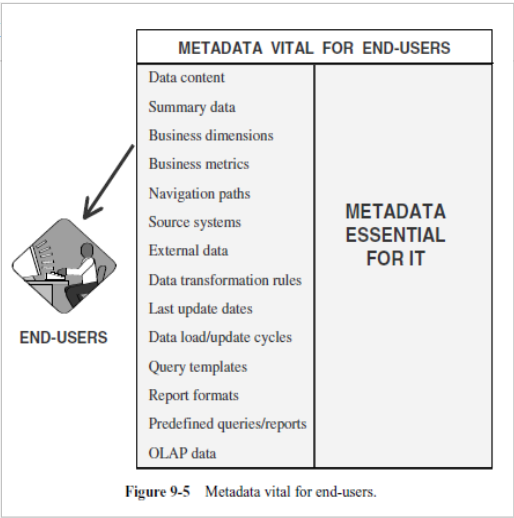
The end-user metadata allows the end-users to use their own business terminology and look for information in those ways in which they normally think of the business.

**Why is metadata especially important in a data warehouse?**

* First, it acts as the glue that connects all parts of the data warehouse.
* Next, it provides information about the contents and structures to the developers.
* Finally, it opens the door to the end-users and makes the contents recognizable in their own terms.



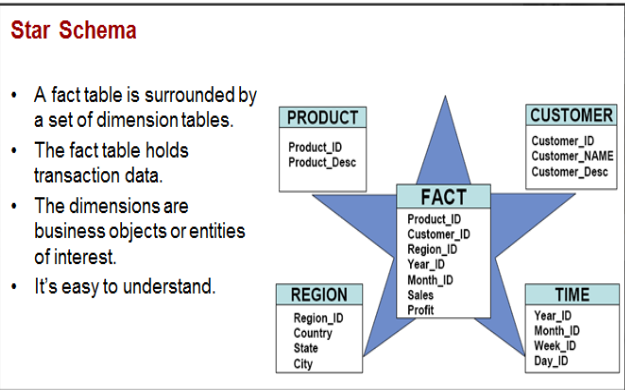




Explain Different types of Metadata in DW.(Explained above)

Explain Different types of dimensional schemas(Star Schema, Snowflake and Galaxy Schema) in DW with example.

* Modeling data warehouses: dimensions & measures
  + **Star schema:** A fact table in the middle connected to a set of dimension tables
  + **Snowflake schema:** A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
  + **Fact constellations:** Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation.



* + **Star Schema Representation**
* Facts and dimensions are normally represented by physical tables in the data warehouse database.
* The fact table is related to each dimension table in a many-to-one (M:1) relationship.
* Fact and dimension tables are related by foreign keys and are subject to the primary/foreign key constraints.
* It’s a RDBMS schema for representing Multidimensional data.
* It’s simplest form of DW schema
* It contains one or more dimension tables & fact tables.
* Here one fact table is connected to multiple dimensions.

**Star Schema characteristics**

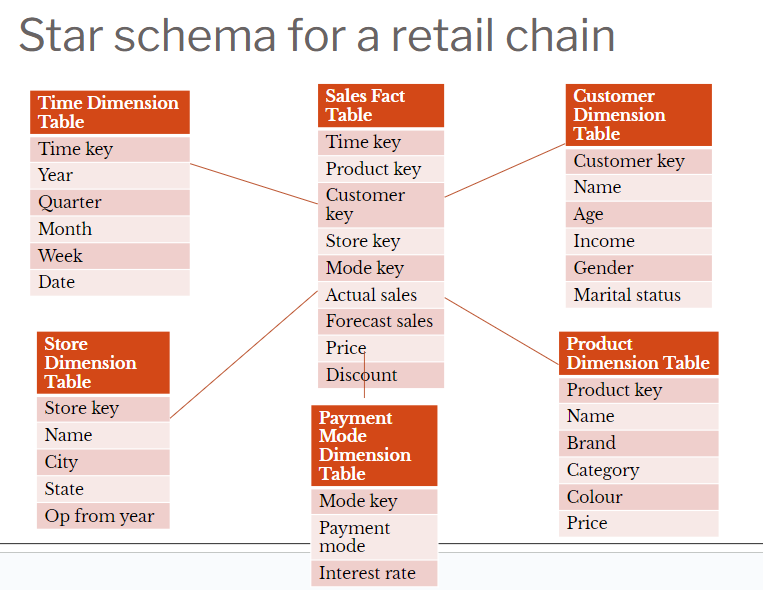
* Star schema is a relational model with one-to-many relationship between the fact table and the dimension tables.
* Denormalized relational model
* Easy to understand. Reflects how users think. This makes it easy for them to query and analyze the data.
* Optimizes navigation.
* Enhances query extraction.
* Ability to drill down or roll up.

**Steps to design star schema**

* Identify a business process for analysis (like sales)
* Identify measures or facts(Sales dollar)
* Identify dimensions for facts(product, location , time , organization dimension)
* List the columns that describe each dimension (region name, branch name etc)
* Determine the lowest level of summary in a fact table (sales dollar).

**IMP aspects of Star schema**

* Every dimension has a primary key.
* Dimension table will not have any parent table
* Hierarchies for the dimensions are stored in the dimensional table itself in the Star schema.



**Advantages of the star schema**

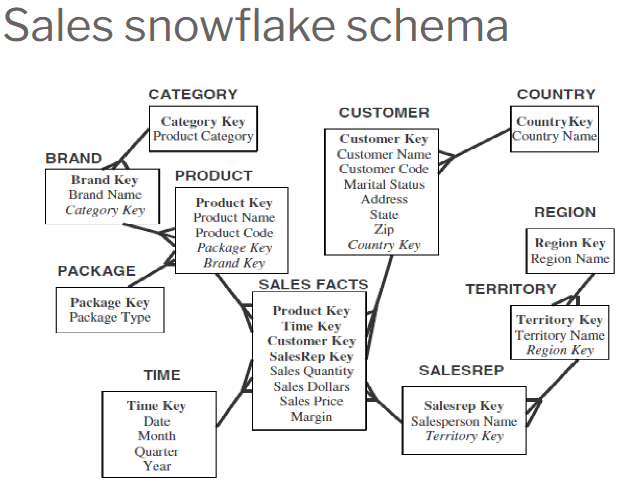
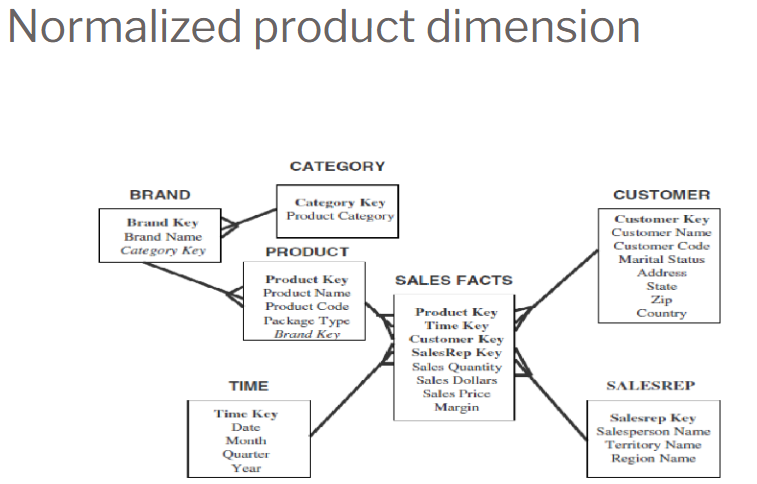
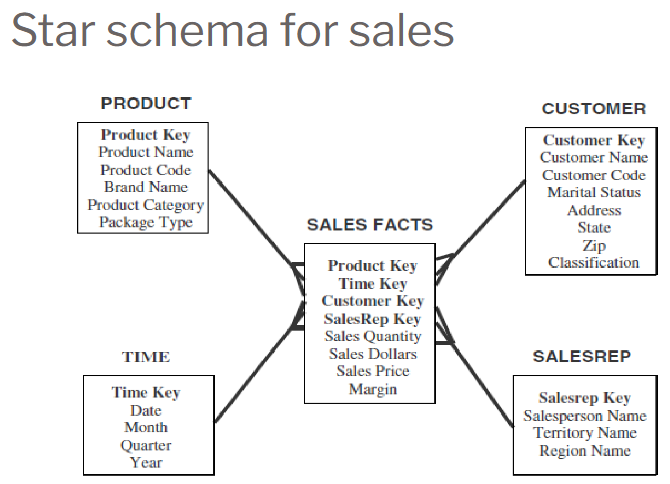
* Simplicity
* Analytical flexibility
* Easy to reconfigure
* Enable summarization
* Easy for users to understand
* Optimizes navigation
* Most suitable for query processing

**Disadvantages:**

* Offers moderate performance
* Not suitable for storing detailed data
* Has a narrow scope in terms of dimensions and facts.

**Snow-Flake Schema**

* The snowflake schema is a more complex data warehouse model than a star schema, and is a type of star schema.
* Snowflake schemas normalize dimensions to eliminate redundancy. For example, a location dimension table in a star schema might be normalized into a location table and city table in a snowflake schema.
* It increases the number of dimension tables and requires more foreign key joins.
* The result is more complex queries and reduced query performance. Figure above presents a graphical representation of a snowflake schema.
* A variation of the star schema, in which all or some of the dimension tables may be normalized.
* Eliminates redundancy
* Generally used when a dimension table is wide.
* Saves space
* Complex querying is required.
* **Advantages**
  + Small savings in storage space
  + Normalized structures are easier to update and maintain
* **Disadvantages**
  + Browsing becomes difficult
  + Complex structure.
  + Degraded query performance because of additional joins

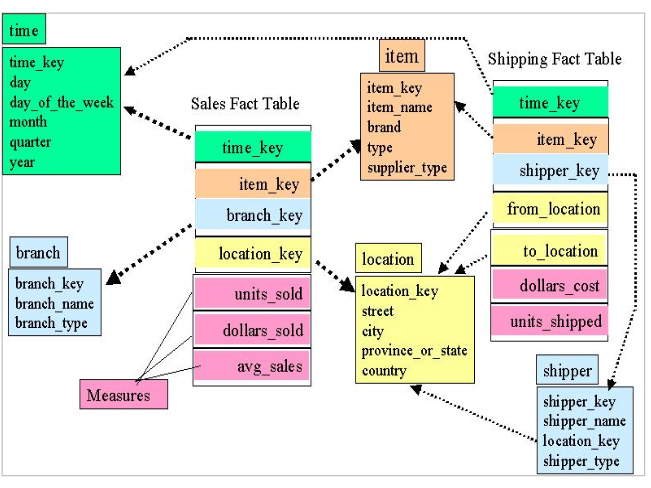


**Fact Constellation**

* Multiple fact tables share dimension tables.
* This schema is viewed as a collection of stars hence called galaxy schema or fact constellation.
* Sophisticated application requires such schema.
* This Schema is used mainly for the aggregate fact tables,

OR

* where we want to split a fact table for better comprehension.
* The split of the fact table is done only when we want to focus on aggregation over a few facts & dimensions.
* It allows dimension tables to be shared between fact tables
* **Fact Constellation**
  + Fact constellation is a set of tables that share some dimension tables. However, fact constellations limit the possible queries for the warehouse.



Explain Data warehouse Architecture with a neat diagram.

<https://www.javatpoint.com/data-warehouse-architecture>

Design Star /Snowflake/Galaxy Schema for following applications

1) Bank

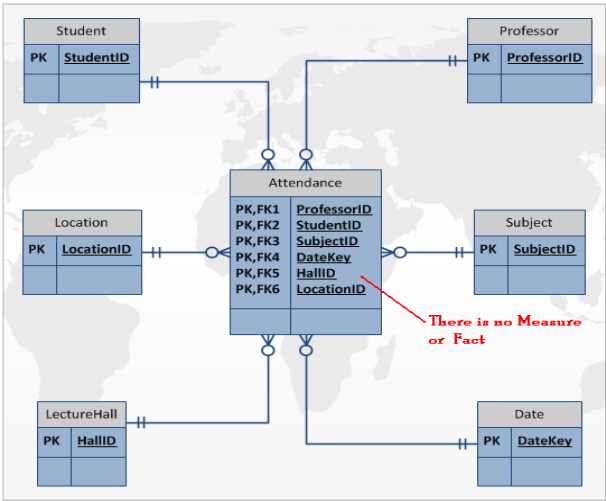
2) Library

3) Railway Reservation

Explain Fact less fact table.

**Fact Less Fact Table:-**

* Fact less Fact table does not contain any facts.
* Generally Fact less fact tables are used to record the events such as students attendance, attendance of participants for an Event like a Meeting.
* There are applications in which tables do not have non key data but that do have foreign keys for the associated dimensions.



Explain Aggregate table

**Aggregate fact tables**

* Contain pre-calculated summaries derived from the most granular (detailed) fact table.
* Created as a specific summarization across any number of dimensions.
* Reduces runtime processing.

**Why need aggregate fact tables?**

* Large size of the fact table
* To speed up query extraction
* **Limitations**
  + Must be re-aggregated each time there is a change in the source data
  + Do not support exploratory analysis
  + Limited interactive use.

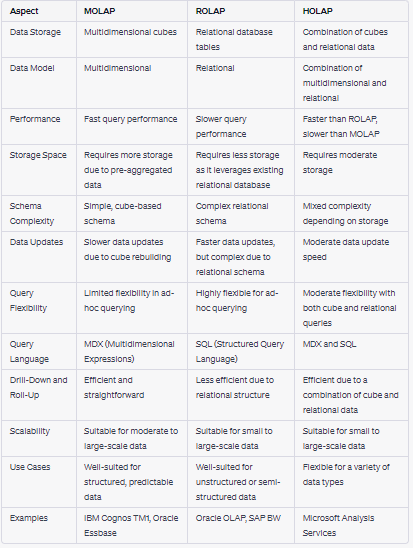
**Aggregate Fact Table:-**

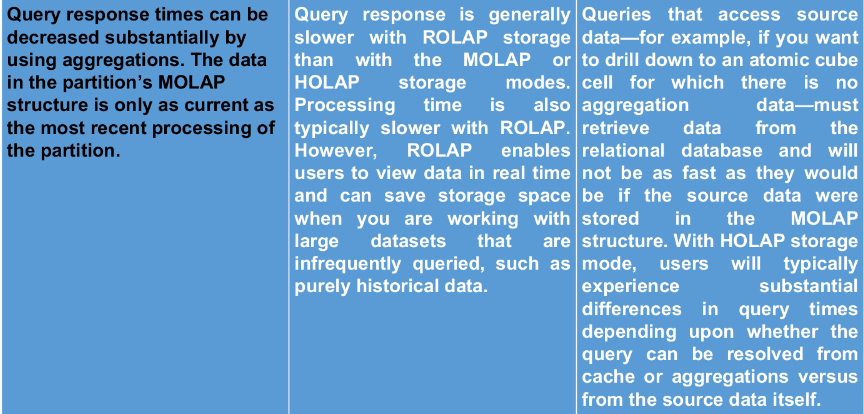
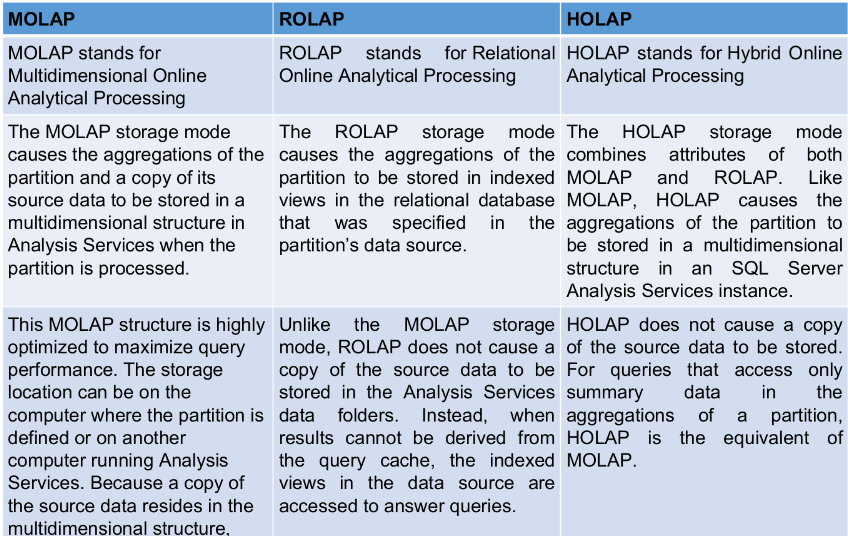
* Choose a simple STAR schema with the fact table at the lowest possible level of granularity.
* Assume a four dimension table with the most granular fact table.
* Operational system-single order, invoice, product and so on.
* Data warehouse- produce large result set(these retrieve hundreds and thousands

Slowly Changing Dimensions (ppt “Modeling the Data Warehouse\_D15” from pg 109)

Explain Different Types of OLAP(Rollup, DrillDown,Dice,Slice,Pivot) operations with example. eg. OLAP Operations on Bank Database.

(“OLAP\_D15\_20\_21” pg 37-52)





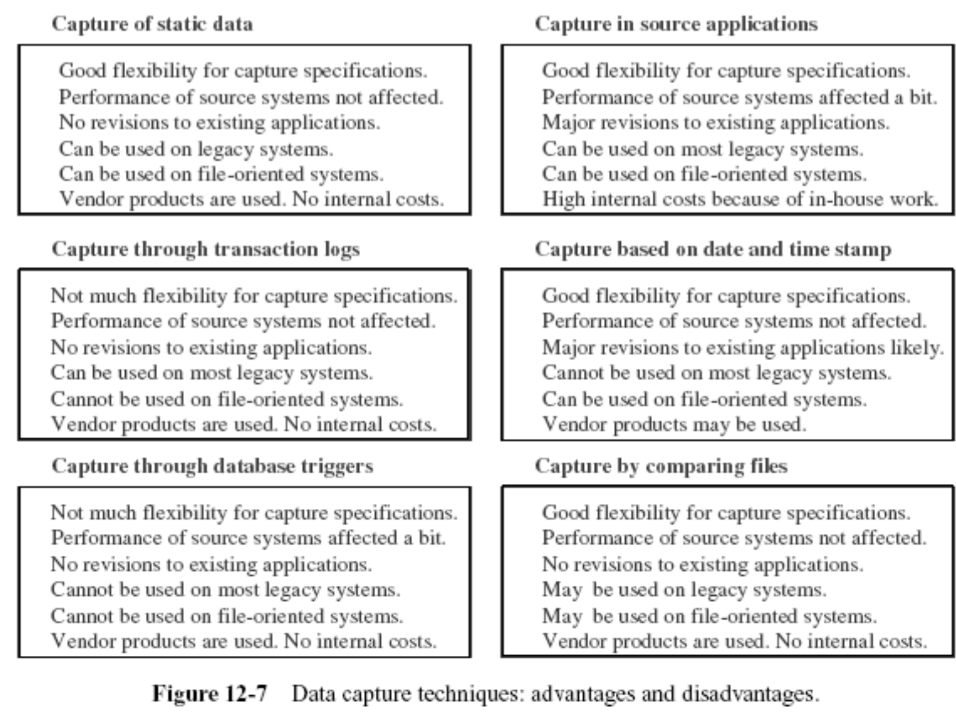
ETL:(For ETL, All answers are from GPT refer PPT for more accurate answers)

Explain ETL Process in Detail.

The ETL process, which stands for Extract, Transform, Load, is a crucial component of data integration and data warehousing. It involves the movement of data from source systems to a target data repository, where the data is transformed and prepared for analysis and reporting. Here's a detailed explanation of the ETL process:

* **Extract (E):**
  + Source Data: The process begins with extracting data from various source systems. Source systems can include databases, files, APIs, web services, and more. These sources could be operational databases, legacy systems, external data providers, or any system containing relevant data.
  + Data Extraction: Data is retrieved from source systems, which may involve querying databases, reading files, or fetching data through APIs. Extraction methods depend on the source and can be either full (extracting all data) or incremental (extracting only new or modified data since the last extraction).
* **Transform (T):**
  + Data Cleaning: Raw data often contains inconsistencies, errors, and missing values. In the transformation phase, data is cleaned, which includes handling missing data, correcting errors, and ensuring data quality.
  + Data Integration: Data from different sources may have varying structures and formats. Data integration involves merging and consolidating data from multiple sources into a common format.
  + Data Transformation: Data may be transformed to fit the requirements of the target data warehouse or data mart. This includes changing data types, aggregating data, splitting columns, and performing other necessary transformations.
  + Data Enrichment: Additional data, such as calculated fields or reference data from external sources, can be added to enhance the value of the data.
  + Data Validation: Ensuring that the data conforms to predefined business rules and quality standards.
* **Load (L):**
  + Data Loading: The transformed data is loaded into the target data repository, which is often a data warehouse or data mart. The loading process can be further divided into three types:
    - Full Load: All data is loaded from scratch, replacing existing data.
    - Incremental Load: Only new or changed data is added to the existing data.
    - Historical Load: Updates and inserts data, but also maintains historical data for trend analysis.
  + Indexing and Aggregation: Indexes and aggregations may be created to improve query performance in the target system.
  + Data Storage: The data is stored in the target repository, following the structure defined in the data warehouse or data mart model.
  + Data Validation: After loading, data is validated to ensure that it was loaded successfully without errors.

Explain Different Types of Extraction.



**Logical Extraction Methods:**

* **Full Extraction:**
  + Purpose: In full extraction, the entire dataset from the source system is extracted without any filtering or conditions. This means all records from the source are included.
  + Use Cases: Full extraction is used when you want to capture the entire dataset for initial loading of the target system or when the source dataset is relatively small and changes are infrequent or easily manageable.
  + Pros: It ensures completeness of data and is straightforward to implement.
  + Cons: It may be inefficient for large datasets or when the majority of the data remains unchanged.
* **Incremental Extraction:**
  + Purpose: Incremental extraction involves extracting only the new or modified data since the last extraction. It is based on identifying changes through timestamps, unique identifiers, or other change-tracking mechanisms.
  + Use Cases: Incremental extraction is suitable for scenarios with large datasets and frequent changes, as it reduces the amount of data transferred and processed.
  + Pros: Efficient for data with frequent updates, reduces the load on source and target systems.
  + Cons: Requires mechanisms to identify changes and may lead to complexity in the ETL process.

**Physical Extraction Methods:**

* **Online Extraction:**
  + Purpose: Online extraction involves retrieving data directly from source systems in real-time or near-real-time. It typically relies on APIs, direct database connections, or other online methods.
  + Use Cases: Online extraction is appropriate when you need immediate or near-instantaneous data updates for real-time analysis, integration, or synchronization.
  + Pros: Provides up-to-the-minute data, suitable for real-time applications.
  + Cons: Can put a significant load on source systems and may require careful monitoring and resource management.
* **Offline Extraction:**
  + Purpose: Offline extraction involves retrieving data from source systems in a batch process or during scheduled intervals, such as daily, weekly, or monthly.
  + Use Cases: Offline extraction is suitable for scenarios where real-time data is not a requirement, and it simplifies scheduling and resource management.
  + Pros: Reduces the load on both source and target systems, simplifies scheduling, and is less resource-intensive.
  + Cons: Data updates are not immediate, and there may be a delay in data availability.

Explain Different Types of Loading.

(unable to obtain accurate from gpt)

Explain Different types of activities done during the Transformation Process.

* **Data Cleansing:**
  + Purpose: Data cleansing involves identifying and correcting errors, inconsistencies, and inaccuracies in the data. It ensures data quality and reliability.
  + Activities: Removing duplicate records, handling missing values, correcting data format issues, standardizing data values, and resolving data quality issues.
* **Data Mapping:**
  + Purpose: Data mapping involves associating source data fields with their corresponding target data fields. It defines how data elements in the source system correspond to elements in the target system.
  + Activities: Creating data mapping documents, specifying field transformations, and documenting relationships between source and target data.
* **Data Transformation:**
  + Purpose: Data transformation involves altering the structure, format, and values of data to fit the requirements of the target system. It may include data enrichment, aggregation, or denormalization.
  + Activities: Applying business rules, calculations, and transformations to data fields, converting data types, aggregating data, and creating calculated fields.
* **Data Enrichment:**
  + Purpose: Data enrichment enhances the data by adding additional information from external sources. This can provide more context and value to the data.
  + Activities: Retrieving and integrating data from external sources (e.g., reference data, third-party data), appending attributes to existing records, and creating new fields with enriched information.
* **Data Validation:**
  + Purpose: Data validation ensures that the data conforms to predefined business rules, quality standards, and consistency requirements.
  + Activities: Applying data validation rules, performing integrity checks, and flagging or rejecting records that fail validation.
* **Data Aggregation:**
  + Purpose: Data aggregation involves summarizing and condensing data to create higher-level insights. This is often done for reporting and analytical purposes.
  + Activities: Summarizing data into categories or time periods, performing statistical calculations, and creating aggregated metrics for reporting.
* **Data Filtering:**
  + Purpose: Data filtering removes unnecessary or irrelevant data to reduce the volume and improve data quality.
  + Activities: Applying filters to exclude data that does not meet specific criteria or is not needed for the analysis.
* **Data Standardization:**
  + Purpose: Data standardization involves converting data values to a common format or structure, ensuring consistency.
  + Activities: Converting date formats, normalizing text fields (e.g., uppercasing), and standardizing units of measurement.
* **Data Joining:**
  + Purpose: Data joining combines data from multiple sources or tables to create a unified dataset.
  + Activities: Merging data from different sources based on common keys or attributes, creating relationships between datasets.
* **Data Splitting:**
  + Purpose: Data splitting divides a field or record into multiple parts, allowing for more granular analysis.
  + Activities: Splitting text fields into separate components (e.g., splitting a full name into first name and last name), or splitting data based on certain criteria.
* **Data Deduplication:**
  + Purpose: Data deduplication identifies and removes duplicate records, ensuring data integrity.
  + Activities: Identifying duplicate records based on specific criteria (e.g., matching keys or fields) and retaining only one instance of each unique record.

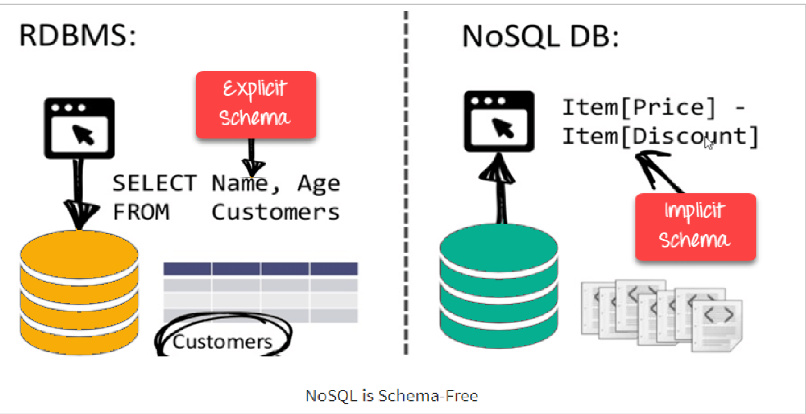
**NOSQL**

What is NoSQL? features , advantages and disadvantages

**What is NoSQL?**

* NoSQL Database is a non-relational Data Management System that does not require a fixed schema.
* It avoids joins, and is easy to scale.
* The major purpose of using a NoSQL database is for distributed data stores with humongous data storage needs.
* NoSQL is used for Big data and real-time web apps.
* For example, companies like Twitter, Facebook and Google collect terabytes of user data every single day.
* NoSQL database stands for “Not Only SQL” or “Not SQL.”
* Though a better term would be “NoREL”, NoSQL caught on. Carl Strozz introduced the NoSQL concept in 1998.
* Traditional RDBMS uses SQL syntax to store and retrieve data for further insights.
* Instead, a NoSQL database system encompasses a wide range of database technologies that can store structured, semi-structured, unstructured and polymorphic data.

* Let’s understand about NoSQL with a diagram in this NoSQL database tutorial:



**Features of NoSQL**

* Non-relational
* NoSQL databases never follow the relational model
* Never provide tables with flat fixed-column records
* Work with self-contained aggregates or BLOBs
* Doesn’t require object-relational mapping and data normalization
* No complex features like query languages, query planners,referential integrity joins, ACID

**Schema-free**

* NoSQL databases are either schema-free or have relaxed schemas
* Do not require any sort of definition of the schema of the data
* Offers heterogeneous structures of data in the same domain

**Simple API**

* Offers easy to use interfaces for storage and querying data provided
* APIs allow low-level data manipulation & selection methods
* Text-based protocols mostly used with HTTP REST with JSON
* Mostly used no standard based NoSQL query language
* Web-enabled databases running as internet-facing services

**Distributed**

* Multiple NoSQL databases can be executed in a distributed fashion
* Offers auto-scaling and fail-over capabilities
* Often ACID concept can be sacrificed for scalability and throughput
* Mostly no synchronous replication between distributed nodes Asynchronous Multi-Master Replication, peer-to-peer, HDFS Replication
* Only providing eventual consistency
* Shared Nothing Architecture.
* This enables less coordination and higher distribution.

**Advantages of NoSQL**

* Can be used as Primary or Analytic Data Source
* Big Data Capability
* No Single Point of Failure
* Easy Replication
* No Need for Separate Caching Layer
* It provides fast performance and horizontal scalability.
* Can handle structured, semi-structured, and unstructured data with equal effect
* Object-oriented programming which is easy to use and flexible
* NoSQL databases don’t need a dedicated high-performance server.
* Support Key Developer Languages and Platforms
* Simple to implement than using RDBMS
* It can serve as the primary data source for online applications.
* Handles big data which manages data velocity, variety, volume, and complexity
* Excels at distributed database and multi-data center operations
* Eliminates the need for a specific caching layer to store data
* Offers a flexible schema design which can easily be altered without downtime or service disruption

Difference between SQL and NOSQL(gpt)



Different NOSQL databases (Models).

**Types of NoSQL Databases**

* NoSQL Databases are mainly categorized into four types: Key-value pair, Column-oriented, Graph-based and Document-oriented.
* Every category has its unique attributes and limitations.
* None of the above-specified database is better to solve all the problems.
* Users should select the database based on their product needs.

**Types of NoSQL Databases:**

* **Key-value Pair Based**
* **Column-oriented Graph**
* **Graphs based**
* **Document-oriented**

**NoSQL Data Architecture Patterns**

* Architecture Pattern is a logical way of categorizing data that will be stored on the Database.
* NoSQL is a type of database which helps to perform operations on big data and store it in a valid format.
* It is widely used because of its flexibility and a wide variety of services.

**Architecture Patterns of NoSQL:**

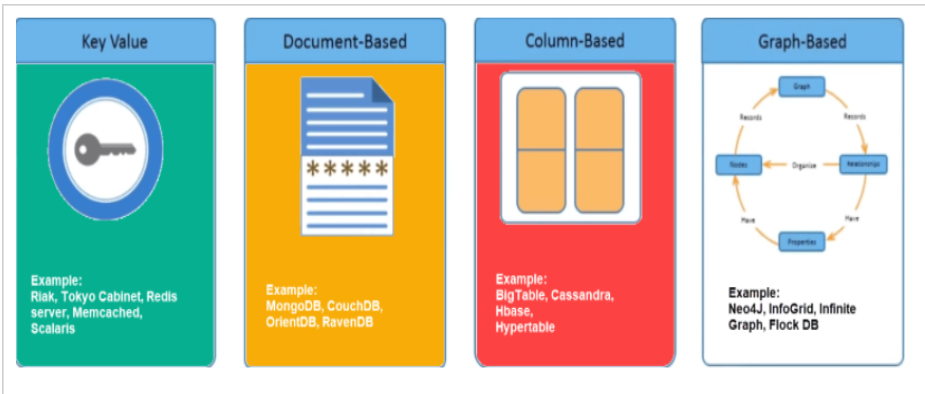
* The data is stored in NoSQL in any of the following four data architecture patterns.

1. Key-Value Store Database

2. Column Store Database

3. Document Database

4. Graph Database



**Key Value Pair Based**

* Data is stored in key/value pairs.
* It is designed in such a way to handle lots of data and heavy load.
* Key-value pair storage databases store data as a hash table where each key is unique, and the value can be a JSON, BLOB(Binary Large Objects), string, etc.

1. **Key-Value Store Database:**

* This model is one of the most basic models of NoSQL databases.
* As the name suggests, the data is stored in the form of Key-Value Pairs.
* The key is usually a sequence of strings, integers or characters but can also be a more advanced data type.
* The value is typically linked or correlated to the key.
* The key-value pair storage databases generally store data as a hash table where each key is unique.
* The value can be of any type (JSON, BLOB(Binary Large Object), strings, etc).
* This type of pattern is usually used in shopping websites or e-commerce applications.

**Advantages:**

* Can handle large amounts of data and heavy load,
* Easy retrieval of data by keys.

**Limitations:**

* Complex queries may attempt to involve multiple key-value pairs which may delay performance.
* Data can involve many-to-many relationships which may collide.

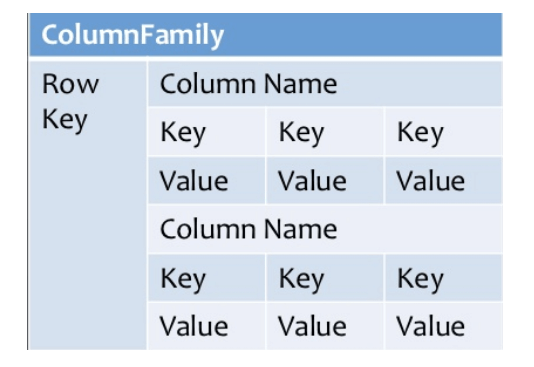
**Examples:**

* DynamoDB
* Berkeley DB



**Column-based**

* Column-oriented databases work on columns and are based on BigTable paper by Google.
* Every column is treated separately.
* Values of single column databases are stored contiguously.
* They deliver high performance on aggregation queries like SUM, COUNT, AVG, MIN etc. as the data is readily available in a column.
* Column-based NoSQL databases are widely used to manage data warehouses, [business intelligence](https://www.guru99.com/business-intelligence-definition-example.html), CRM, Library card catalogs,
* HBase, Cassandra, HBase, Hypertable are NoSQL query examples of column based databases.

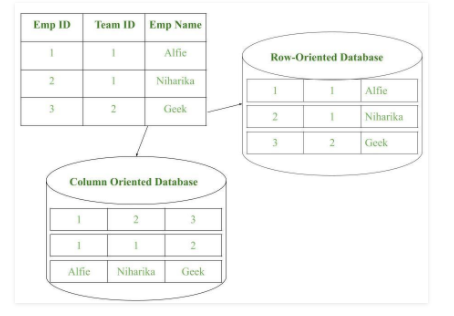


**2. Column Store Database:**

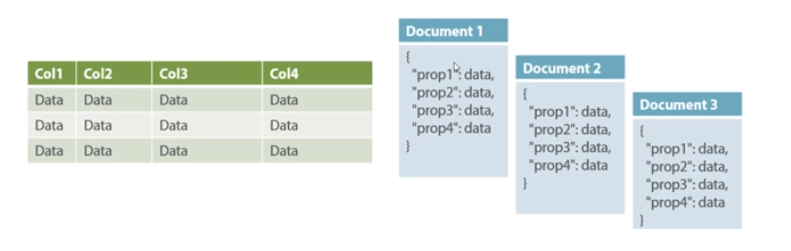
* Rather than storing data in relational tuples, the data is stored in individual cells which are further grouped into columns.
* Column-oriented databases work only on columns.
* They store large amounts of data into columns together.
* Format and titles of the columns can diverge from one row to other. Every column is treated separately.
* But still, each individual column may contain multiple other columns like traditional databases.
* Basically, columns are mode of storage in this type.

**Advantages:**

* Data is readily available
* Queries like SUM, AVERAGE, COUNT can be easily performed on columns.
* Examples:
* HBase
* Bigtable by Google
* Cassandra



* **Document-Oriented:**
* Document-Oriented NoSQL DB stores and retrieves data as a key value pair but the value part is stored as a document.
* The document is stored in JSON or XML formats.
* The value is understood by the DB and can be queried.



**3. Document Database:**

* The document database fetches and accumulates data in form of key-value pairs but here, the values are called as Documents.
* Document can be stated as a complex data structure.
* Document here can be a form of text, arrays, strings, JSON, XML or any such format.
* The use of nested documents is also very common.
* It is very effective as most of the data created is usually in form of JSONs and is unstructured.

**Advantages:**

* This type of format is very useful and apt for semi-structured data.
* Storage retrieval and managing of documents is easy.
* Limitations:
* Handling multiple documents is challenging
* Aggregation operations may not work accurately.
* Examples:
* MongoDB
* CouchDB



**Graph-Based**

* A graph type database stores entities as well the relations amongst those entities.
* The entity is stored as a node with the relationship as edges.
* An edge gives a relationship between nodes.
* Every node and edge has a unique identifier.
* Compared to a relational database where tables are loosely connected, a Graph database is multi-relational in nature.
* Traversing relationships is fast as they are already captured into the DB, and there is no need to calculate them.
* Graph based database mostly used for social networks, logistics, spatial data.
* Neo4J, Infinite Graph, OrientDB, FlockDB are some popular graph-based databases.

**4. Graph Databases:**

* Clearly, this architecture pattern deals with the storage and management of data in graphs.
* Graphs are basically structures that depict connections between two or more objects in some data.
* The objects or entities are called nodes and are joined together by relationships called Edges.
* Each edge has a unique identifier.
* Each node serves as a point of contact for the graph.
* This pattern is very commonly used in social networks where there are a large number of entities and each entity has one or many characteristics which are connected by edges.
* The relational database pattern has tables that are loosely connected, whereas graphs are often very strong and rigid in nature.

**Advantages:**

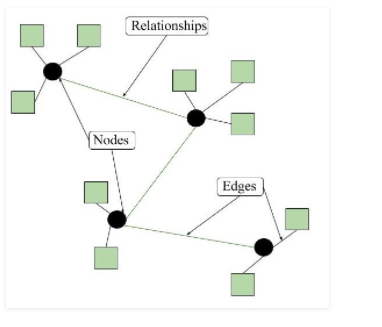
* Fastest traversal because of connections.
* Spatial data can be easily handled.

**Limitations:**

* Wrong connections may lead to infinite loops.

**Examples:**

* Neo4J
* FlockDB( Used by Twitter)



CAP Theorem.

**What is the CAP Theorem?**

* The CAP theorem is also called the brewer's theorem.
* It states that is impossible for a distributed data store to offer more than two out of three guarantees
* Consistency
* Availability
* Partition Tolerance

**Consistency:**

* The data should remain consistent even after the execution of an operation.
* This means once data is written, any future read request should contain that data.
* For example, after updating the order status, all the clients should be able to see the same data.

**Availability:**

* The database should always be available and responsive.
* It should not have any downtime.
* **Partition Tolerance:**
* Partition Tolerance means that the system should continue to function even if the communication among the servers is not stable.
* For example, the servers can be partitioned into multiple groups which may not communicate with each other.
* Here, if part of the database is unavailable, other parts are always unaffected.

Big Data and Features, advantages and Disadvantages

Big Data refers to extremely large and complex datasets that are beyond the capacity of traditional data processing and management tools. It encompasses a wide variety of data sources and types, including structured, semi-structured, and unstructured data. Big Data is characterized by the "3Vs": Volume, Velocity, and Variety. Here are some features, advantages, and disadvantages of Big Data:

**Features of Big Data:**

* **Volume:**
  + Refers to the vast amount of data generated, collected, and stored.
  + Big Data includes petabytes and exabytes of data from various sources, such as social media, sensors, and business transactions.
* Velocity:
  + Denotes the speed at which data is generated and needs to be processed.
  + Big Data often involves real-time or near-real-time data streaming, requiring rapid data ingestion and analysis.
* **Variety:**
  + Encompasses diverse data types, including structured data (e.g., databases), semi-structured data (e.g., XML, JSON), and unstructured data (e.g., text, images, videos).
  + Big Data solutions must handle this variety efficiently.
* **Veracity:**
  + Refers to the trustworthiness and reliability of data.
  + Big Data can contain noisy, inconsistent, or inaccurate data, and managing its quality is crucial.
* **Value:**
  + The ultimate goal of Big Data is to extract valuable insights and knowledge from the data to drive decision-making and innovation.

**Advantages of Big Data:**

* **Informed Decision-Making:**
  + Big Data analytics provide organizations with valuable insights to make data-driven decisions and gain a competitive edge.
* **Improved Customer Experience:**
  + Analysis of Big Data can help understand customer behavior and preferences, leading to better-targeted marketing and enhanced customer satisfaction.
* **Efficiency and Productivity:**
  + Optimizing processes through Big Data analytics can improve operational efficiency and productivity.
* **Real-Time Insights:**
  + Big Data solutions enable real-time or near-real-time analysis, facilitating quicker responses to changing conditions.
* **Innovation:**
  + Big Data fuels innovation by enabling businesses to identify new trends and opportunities.
* **Cost Reduction:**
  + Effective analysis of Big Data can lead to cost savings by optimizing processes and resource allocation.

**Disadvantages of Big Data:**

* **Complexity:**
  + Managing, storing, and processing Big Data can be highly complex and costly.
* **Data Privacy and Security:**
  + Big Data can raise concerns about data privacy and security, as handling large volumes of data increases the risk of breaches.
* **Scalability:**
  + Scaling Big Data solutions to handle ever-growing data volumes can be a significant challenge.
* **Data Quality:**
  + Ensuring data quality in Big Data can be difficult due to the variety and volume of data sources.
* **Talent and Skill Gap:**
  + Finding and retaining talent with expertise in Big Data technologies can be challenging.
* **Infrastructure Costs:**
  + Building and maintaining the necessary infrastructure for Big Data can be expensive.