DWH Concepts

1. Characteristics of DWH:
   1. *Subject -Oriented:* A data warehouse is a subject-oriented approach. Because, it provides information on a specific topic rather than information about an organization’s ongoing operations.

1. *Integrated:* A data warehouse combines data from various sources. These may include a cloud, relational databases, flat files, structured and semi-structured data, metadata, and master data. The sources are combined in a manner that’s consistent, relatable, and ideally certifiable, providing a business with confidence in the data’s quality.

1. *Time-variant*: Time variant keys (e.g., for the date, month, time) are typically present.

1. *Persistent and non-volatile:* Prior data isn’t deleted when new data is added. Historical data is preserved for comparisons, trends, and analytics.

1. Difference between OLAP(ODS vs DWH)

|  |  |
| --- | --- |
| OLTP(ODS) | OLAP(DWH) |
| Application Oriented | Subject Oriented |
| Used to run business | Used to analyze the business |
| Detailed data | Summarized and refined |
| Current up to date | Snapshot data |
| Isolated | Integrated |
| Repetitive access | Ad-hoc access |
| Few Records accessed at a time( tens) | Large volumes  accessed at a time( millions) |
| Database size 100MB - 100GB | Database size 100GB - Few terabytes |

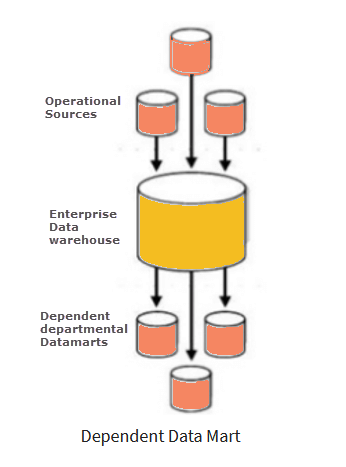
1. Data Warehouse Architecture
   1. *Single-tier Architecture:* The objective of a single layer is to minimize the amount of data stored. This goal is to remove data redundancy.
   2. *Two-tier Architecture:* Two-layer architecture is one of the Data Warehouse layers which separates physically available sources and data warehouses. This architecture is not expandable and also not supporting a large number of end-users. It also has connectivity problems because of network limitations.

1. *Three-tier Architecture:* This is the most widely used Architecture of Data Warehouse.It consists of the Top, Middle and Bottom Tier.
   1. *Bottom Tier:* The database of the Datawarehouse servers as the bottom tier. It is usually a relational database system. Data is cleansed, transformed, and loaded into this layer using back-end tools.
   2. *Middle Tier:* In the middle tier, we have the OLAP Server that can be implemented in either of the following ways.

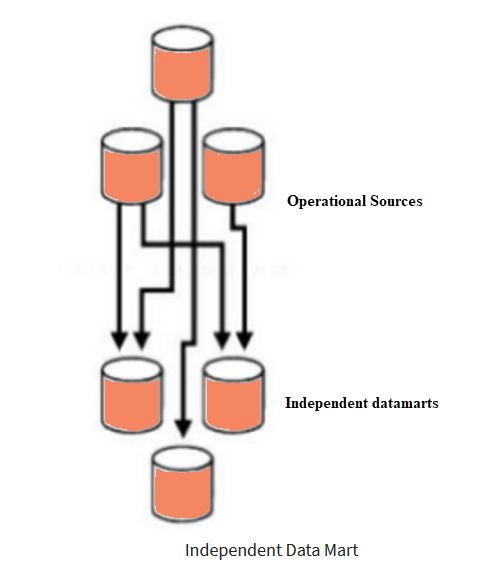
* By Relational OLAP (ROLAP), which is an extended relational database management system. The ROLAP maps the operations on multidimensional data to standard relational operations.
* By Multidimensional OLAP (MOLAP) model, which directly implements the multidimensional data and operations.

1. *Top-Tier:* This tier is the front-end client layer. This layer holds the query tools and reporting tools, analysis tools and data mining tools.

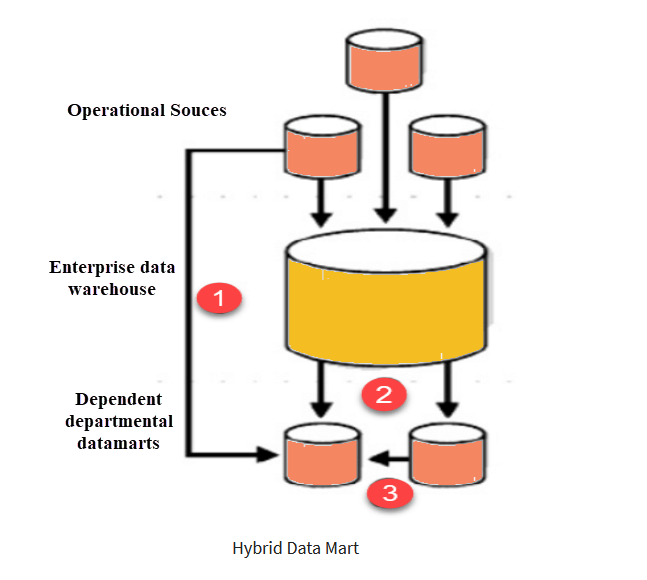
1. Data Mart: A data mart is a subject oriented data warehouse
   1. *Dependent or Top down (Inmon approach):* Dependent data marts are created by drawing data directly from operational, external or both sources.



1. *Independent or Bottom down (Kimball approach):* Independent data mart is created without the use of a central data warehouse.



1. *Hybrid Data Marts:* This type of data marts can take data from data warehouses or operational systems.



1. *Difference between Data warehouse vs Data Marts*

|  |  |
| --- | --- |
| Data Warehouse | Data Marts |
| Data warehouse is a centralized system. | While it is a decentralized system. |
| In data warehouse, lightly denormalization takes place. | While in Data mart, highly denormalization takes place. |
| Data Warehouses have a long life. | While data-mart has a shorter life than warehouse. |
| In the Data Warehouse, Data is contained in detail form. | While in this, data are contained in summarized form. |
| Complicated design process of creating schemas and views. | Easy design process of creating schemas and views. |
| Long time for processing the data because of large data. | Less time for processing the data because of handling only a small amount of data. |

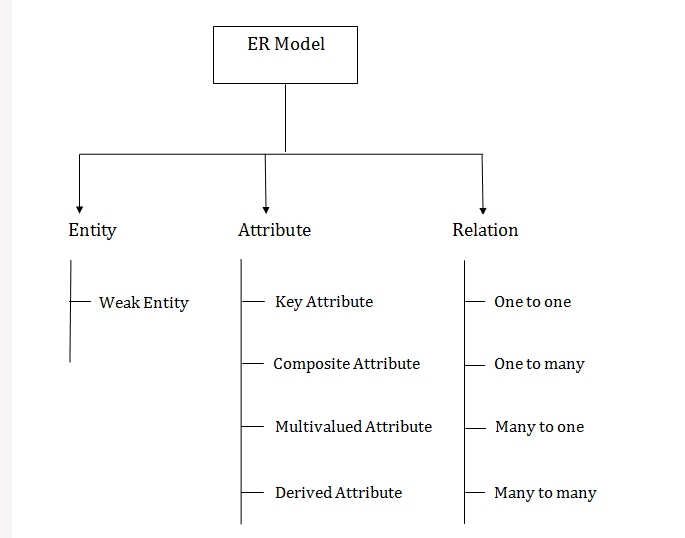
1. *Data Model:* Data modeling is a technique aimed at optimizing the way that information is stored and used within an organization.
2. Types of Data Models
3. *Conceptual Data Model:* This model is typically created by Business stakeholders and Data Architects. The purpose is to organize, scope and define business concepts and rules.

1. *Logical Data Model:* This model is typically created by Data Architects and Business Analysts. The purpose is to develop a technical map of rules and data structures.

1. *Physical Data Model:* This model is typically created by DBA and developers. The purpose is actual implementation of the database.

1. ER Data Model:

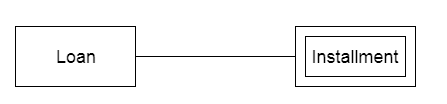
ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.



1. *Entity:* An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.

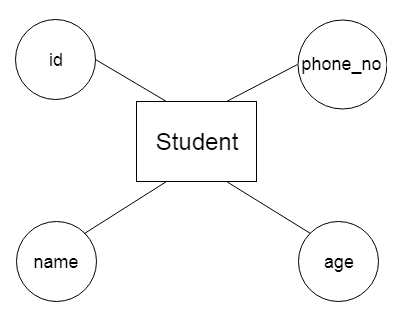
Consider an organization as an example- manager, product, employee, department etc. can be taken as an entity.

i. *Weak Entity:* An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.

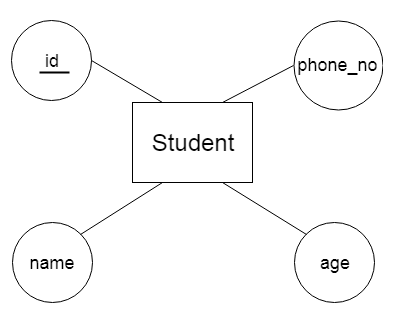


1. *Attribute:* The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.

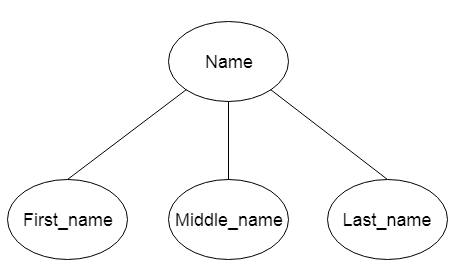
For example:-  id, age, contact number, name, etc. can be attributes of a student



1. *Key Attribute:* The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.

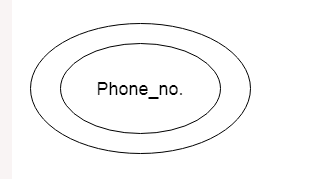


1. *Composite Attribute:* An attribute that is composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.



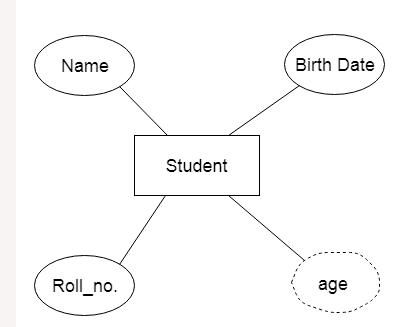
1. *Multivalued Attribute:* An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent a multivalued attribute.

For example: a student can have more than one phone number.

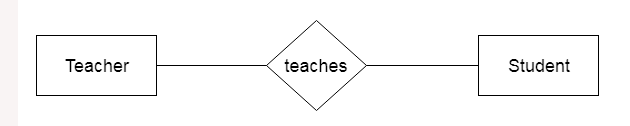


1. *Derived Attribute:* An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.

For example: A person's age changes over time and can be derived from another attribute like Date of birth.

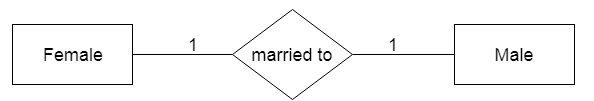


1. *Relationship:* A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.



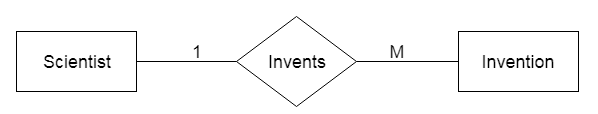
1. *One-to-One Relationship:* When only one instance of an entity is associated with the relationship, then it is known as one to one relationship.

For example: A female can marry to one male, and a male can  marry to one female.



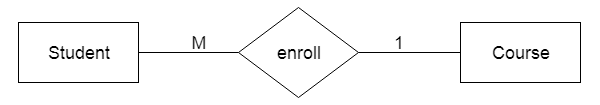
1. *One-to-many relationship*: When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a one-to-many relationship.

For example: scientists can invent many inventions, but the invention is done by the only specific scientist.



1. *Many-to-one relationship:* When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.

For **example:** Student enrolls for only one course, but a course can have many students.



1. *Many-to-many relationship:* When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship.

**For example:-** Employee can assign by many projects and project can have many employees.



1. Dimensional Data Model
2. Gathering Business Requirement - First Phase
3. Conceptual Data Modeling (CDM) - Second Phase
4. Logical Data Modeling (LDM) - Third Phase
5. Physical Data Modeling (PDM) - Fourth Phase
6. Generate Database - Fifth Phase

1. Difference between ER Data Modeling and Dimensional Data Modeling

1. Dimensional table

1. Fact table

1. Dimensions

1. Measures (Facts)

1. Types Schema
   1. Star Schema
   2. SnowFlake Schema
   3. Fact Constellation Schema or Galaxy schema

1. Types of Dimensions
   1. Conformed Dimensions
   2. Degenerate Dimensions
   3. Junk Dimensions
   4. Slowly changing Dimensions
      1. SCD-0: Fixed Dimension.No changes allowed, dimension never changes
      2. SCD-1: Overwrite the changes
      3. SCD-2: History will be added as a new row
      4. SCD-3: History will be added as a new column
      5. SCD-4: A new dimension will be added
      6. SCD-6: Combination of Type 2 and Type 3

1. Types of Facts
   1. Additive
   2. Semi-Additive
   3. Non-Additive

1. Fact less Fact table

1. Types of Fact table
   1. Cumulative
   2. Snapshot

1. Data loading processes
2. Full Load
3. Incremental Load

1. What is Normalization?
   1. Normalization is the process of organizing the data in the database.
   2. Normalization is used to minimize the redundancy from a relation or set of relations.
   3. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.

Types of normalization:

1. 1NF: A relation is in 1NF if it contains an atomic value.
2. 2NF: A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.
3. 3NF: A relation will be in 3NF if it is in 2NF and no transition dependency exists.
4. BCNF: A stronger definition of 3NF is known as Boyce Codd's normal form
5. 4NF: A relation will be in 4NF if it is in Boyce Codd's normal form and has no multivalued dependency.
6. 5NF: A relation is in 5NF. If it is in 4NF and does not contain any join dependency, joining should be lossless.