**Blue Print of DBMS**

* E-R Diagram
* Conversion of E-R diagram into relational model
* Basics of relational model and functional dependency
* Idea about keys/types
* Normalization (1NF – BCNF)
* Lossless decomposition and Dependency preserving
* Indexing and Physical structure (B, B+ tree)
* SQL, Relational Algebra, Relational Calculus
* Transaction
* Concurrency control

**Disadvantages of file processing system**

1. Data Redundancy *( same data stored in multiple files/places)*
2. Data Inconsistency
3. Difficulty in accessing data
4. Data isolation *(file system is pro. Lang. dependent and they keep on changing day by day. )*
5. Security problem *( It is difficult to apply authentication and specify various provide different access to different users )*
6. Atomicity problem *(Atomicity describes that either a transaction must be completely executed or not at all. It is not possible in file system )*
7. Concurrent – access anomalies *(difficult to perform concurrent operations accurately )*
8. Integrity problem *(difficult to define integrity constraints)*

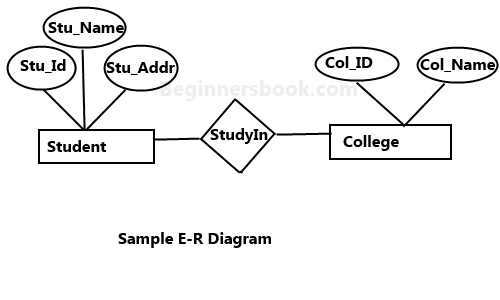
**OLAP V/s OLTP**

* OLAP stand for online analytical processing and OLTP stand for online transaction processing.
* OLAP is subject oriented and OLTP is application oriented
* OLAP is decision making and OLTP is day to day operations
* OLAP size is TB, PB and OLTP size is MB, GB
* OLAP data is used by CEO, MD, GM and OLTP data is used by clerks and managers.
* OLAP data is used to perform read operations and OLTP data is used to perform Read and Write operations.

**Entity Relation Diagram**

**Introduction and basics**

* Introduced by Dr. Peter Chen in 1976.
* A non-technical design method works on conceptual level based on the perception of real world.
* Consists of collection of objects called entities and relationships among these objects and attributes which defines their properties.
* Free from ambiguities and provides a standard and logical way of visualizing data.

****

**Basic Terminologies of ER Diagram:-**

* Entity: - An entity is a thing or an object in the real world that is distinguishable from other objects based on the values of the attributes it possess.
* Types of Entities: -
  + Tangible Entities: - Entities which physically exist such as Pen, Car etc.
  + Intangible Entities: - Entities which exists logically such as Bank accounts.
* Entity Set: - Collection of similar types of entities that share same attributes.
* Attributes: -
  + Describes the characteristics of entities.
  + For each attributes, there is a set of permitted values called domains.
  + Represented by ellipse or oval.
* Types of Attributes: -
  + Simple-composite: - Attributes which cannot be divided further is called simple attribute. Attributes which can be further divided in simple attributes are called composite attributes.
  + Single-Multivalued attributes: - Attributes which can have only one value at an instance of time. Attributes which can have more than one value at an instance of time.
  + Stored-derived: - stored attributes are the values which are stored in database. Derived attributes are not stored in database instead of that it is derived from stored attributes. Example Age is derived attribute.

**Relationship**

* Relationship: - It is an association between two or more entities of same or different entity set.
* Relationship type/set: - A set of similar type of relationship
* Every relationship type has three components: -
  + Name
  + Degree
  + Cardinality
* Degree of Relationship set: - means number of entity set associated in the relationship set. Degree can range from 1 to n.
* Mapping (cardinality ratio): - express the number of entities to which other entities can be related via a relationship.
* Types of Mapping: -
  + One to one Mapping (each entity can be related to **atmost** one entity of another entity set) (citizen with adhar card )
  + One to Many Mapping (every entity can be related to atmost n entities in another entity set but reverse is not allowed)(example person to phone number)
  + Many to One Mapping (phone number to person)
  + Many to Many Mapping