Part 2, DB Concepts Day 1

DB Life Cycle

* Analysis 🡺 System Analyst
  + Scope (Requirement Document).
* DB Design 🡺 Diagram
  + ERD (Entity Relationship Diagram) or Logical Representation.
  + Done By DB Designer.
* DB Mapping 🡺 Set of rules
  + DB Schema 🡺 Actual tables.
* DB Implementation 🡺 DB Online
  + Physical Schema.
  + Tool 🡺 RDBMS (Relational Database Management System).
    - Access.
    - SQL Server.
    - Oracle.
    - My SQL.
  + SQL (Structured Query Language).
  + DB Developer.
* GUI
  + Application.
  + User 🡺 Application Programmer.
  + Interface.
  + C#, HTML, JS, MVC.
* Client 🡺 Browser 🡺 URL.
* Client(Display) 🡺 App Server (SQL Query) 🡺 DB Server (Result) 🡺 App Server 🡺 Client.

File Based System

* Delimited File
  + 1.ahmed, 20
  + 2.ali,22
  + 3.omar,24
* Fixed Width (bytes)
  + 10 HR 20 SD 30 Admin 40 Finance

Issues with File Based System

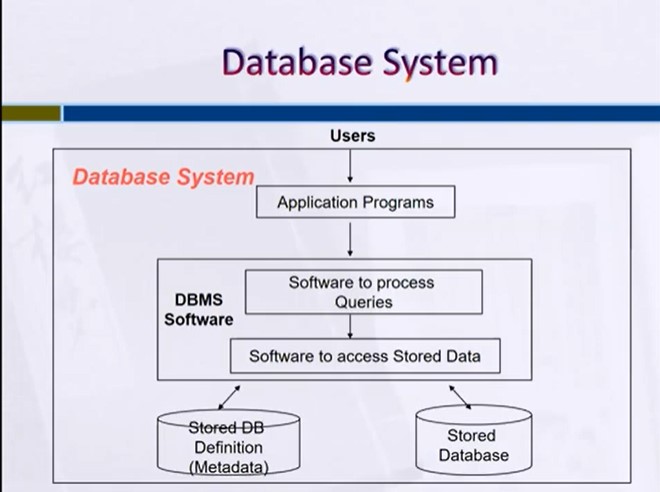
* Search and Sort.
* Performance.
* No Sharing.
* Modification.
* No Security.
* No Data Quality.
* Duplication or DB Redundancy.
* No Relationship.
* No DB Integrity.
* Manual Backup and Restore.

DB System

* Tables to contain my data.
* Set of rows and columns.
* Columns’ names are obvious, So DB Structure are also obvious.
* Any table should has Primary Key (PK) 🡺 Unique and Not Null 🡺like (ID).
* All tables have relationship between each other, any data in a table reflect in other tables.
* Each column has Data Type.

Basic Definitions

* **Database** 🡺 A collection of related data.
* **Database Management System (DBMS)** 🡺 A software package / system to facilitate the creation and maintenance of a computerized database (model introduced in 1970 IBM but RDBMS appears in 1980).
* **Database System** 🡺 The DBMS software together with the data itself.
  + Sometimes, the applications are also included. (Software + Database).



DBMS Advantages

* Standardization and better Data accessibility and response (SQL).
* Sharing data.
  + Different users get different views of the data.
* Enforcing Integrity Constrains.
* Improved Data Quality.
  + Constrains, Data validation rules.
* Inconsistency can be avoided.
* Restricting Unauthorized Access.
* Providing Backup and Recovery.
  + Disaster recovery is easier.
* Minimal Data Redundancy.
  + Leads to increased data integrity/consistency.
* Program-Data Independence.
  + Metadata stored in DBMS, so applications don’t worry about data formats.
  + Data queries / updates managed by DBMS.

Database Users

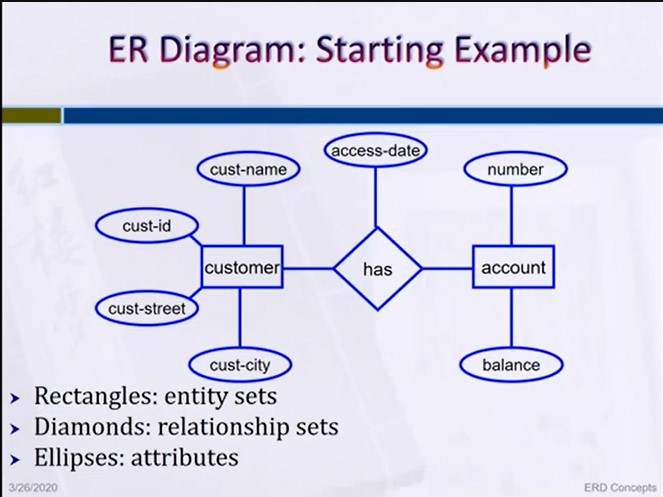
* Database Administrator (DBA)
* System Analysts.
* Database Designer.
* Database Developer.
* Application Programmer.
* End users.

Entity Relationship Diagram (ERD)

* Identifies information required by the business by displaying the relevant entities and the relationships between them.

Basic constructs of the ER Model

* **Entities** 🡺 person, place, object, event, concept (often corresponds to a real time object that is distinguishable from any other object).
* **Attributes** 🡺 property or characteristic of an entity type (often corresponds to a field in a table).
* **Relationships** 🡺 link between entities (corresponds to primary key / foreign key equivalencies in related tables).

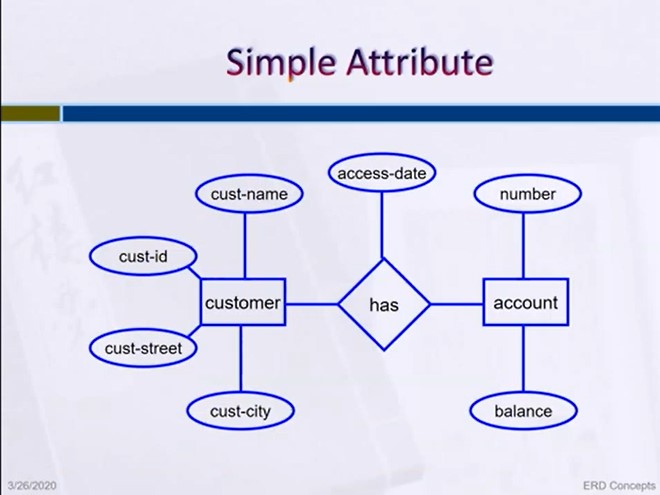


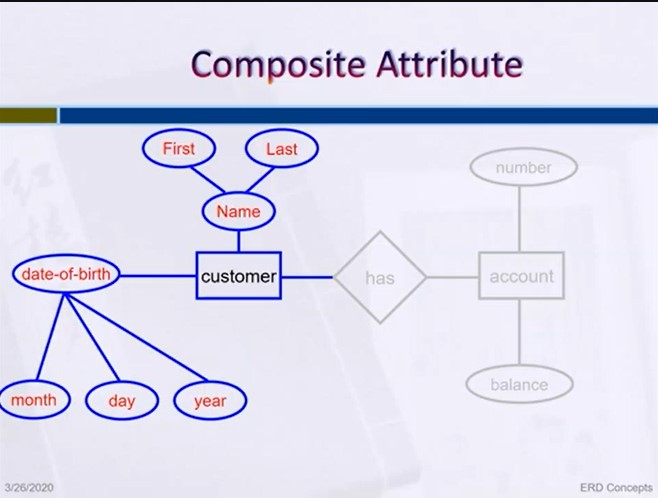
Strong Entity Vs Weak Entity

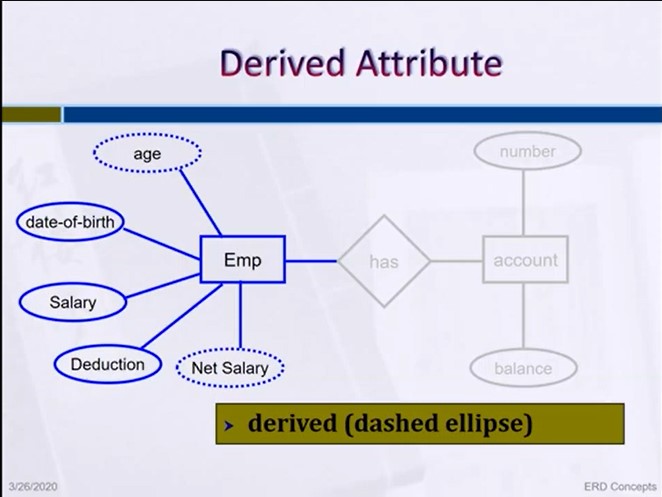
* **Strong Entity** 🡺 An entity set that has a primary key.
* **Weak Entity** 🡺 An entity set that don’t have sufficient attributes to form a primary key, has a partial key.
* **Partial Key** 🡺 A set of attributes that can associated with PK of an owner entity set to distinguish a weak entity.

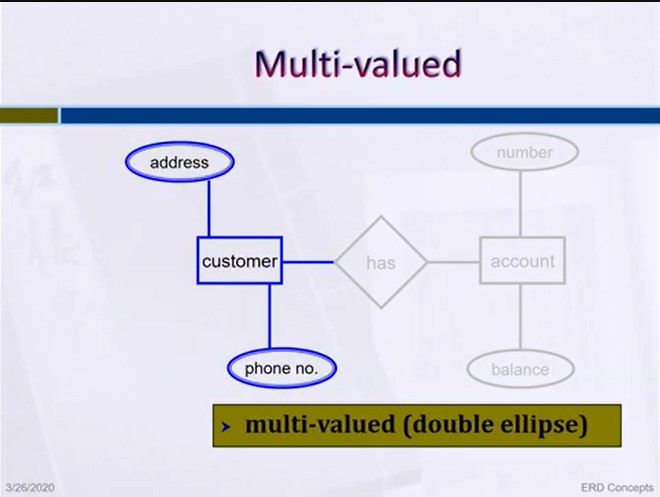
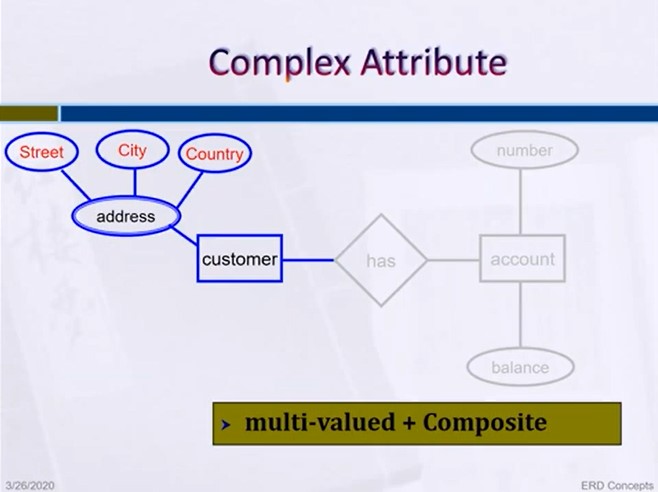
Types of Attributes

* Composite Attribute.
* Multi-valued Attribute.
* Derived Attribute.
* Complex Attribute.
* Simple Attribute.







Relationship

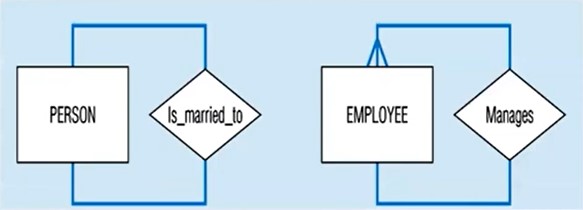
* A Relationship is an association among several entities.
* A relationship may also have attributes.
* Each relation has:
  + Degree of Relationships.
  + Cardinality Constraint.
  + Participation.

Degree of Relationships

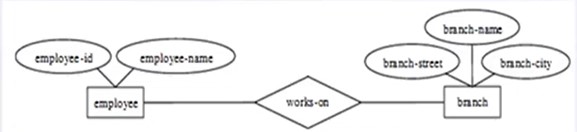
* Degree 🡺 number of entity types that participate in a relationship.
* Three cases
  + **Unary** 🡺 Between two instances of one entity type.
  + **Binary** 🡺 Between the instances of two entity types.
  + **Ternary** 🡺 Among the instances of three entity types.

**Recursive Relationship (Unary)**

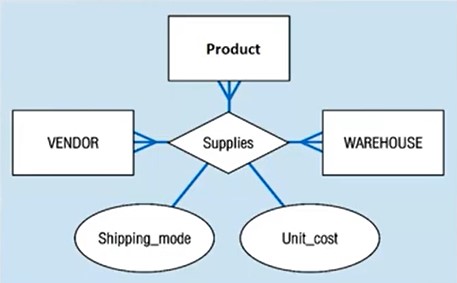
* Recursive Relationships 🡺 A relationship in which the same entity participates more than once.



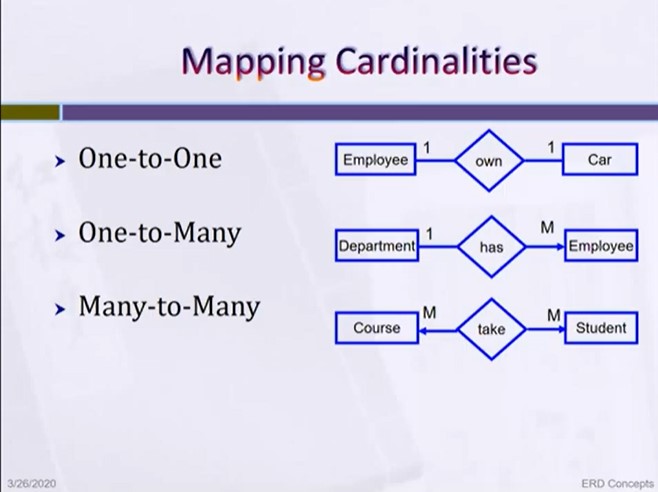
**Binary Relationship**

* A Binary Relationship set is of degree 2.

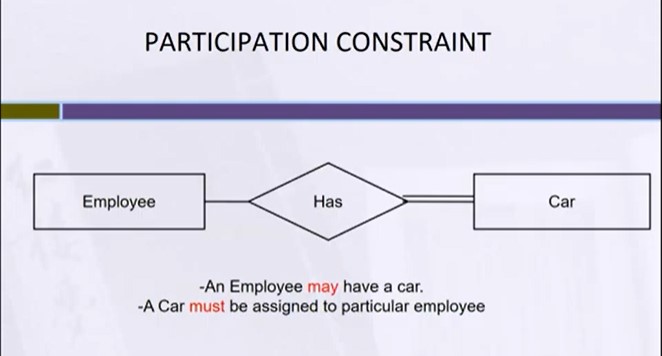
**Ternary Relationship**

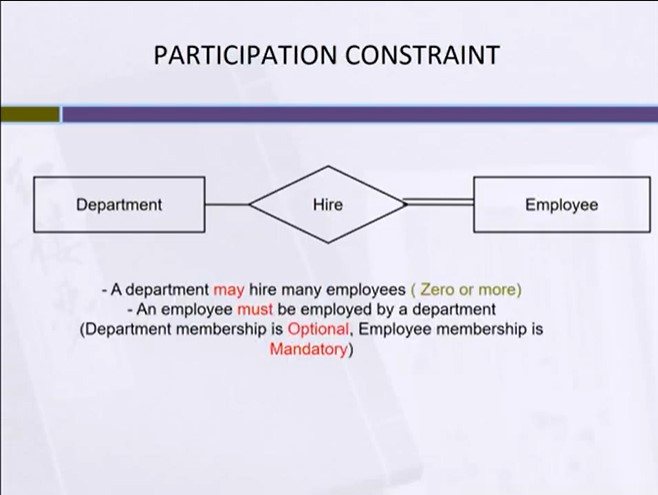
* A Ternary Relationship set is of degree 3.

Cardinality

* How many instances of one entity will or must be connected to a single instance from the other entities.
* Has three types:
  + One – One Relationship.
  + One – Many Relationship.
  + Many – Many Relationship.

Participation Constraint

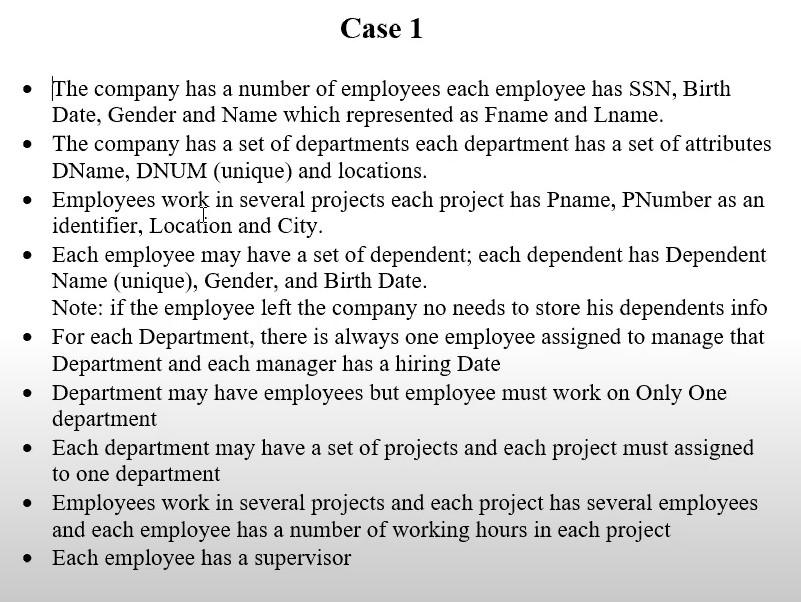
* An employee MUST work for a department.
* An employee entity can exist only if it participates in a WORKS\_FOR relationship instance.
  + **The participation is TOTAL.**
* Only some employees manage departments.
  + **The participation is PARTIAL.**



Keys

* Different Types of Keys:
  + Candidate Key
  + Primary Key
  + Foreign Key
  + Composite Key
  + Partial Key
  + Alternate Key
  + Super Key

Case Study



Solution of the Case

