DBMS - DataBase Management System

DBMS is a general-purpose system that facilitates the processes of defining, constructing, manipulating & sharing databases among various users & applications.

- Define a particular database in terms of its data types, structures, & constraints.
- Construct or load the initial dabase contents on a secondary storage medium.
- Manipulating the database:
  - **Retrieval:** Querying, generating reports.
  - **Modification:** Insertions, deletions & updates to its content.
  - **Accessing** the database through web applications.
- Processing & sharing by a set of concurrent users & application programs - yet, keeping all data valid & consistent.

⇒ Main Characteristics of Database

- Self-describing nature of a database system.
- Insulation bw programs & data, & data abstraction.
- support of multiple views of data.
- Sharing of data & multiuser transaction processing.

Khushal Agrawal
2015002 40

Q=> Define

a) Data: It is referred as known facts that can be recorded & have an implicit meaning; raw.
Ex:- name, Phone number of People You Know.

b) Database: It is a highly organised, interrelated, & structured set of data about a particular enterprise. Also, it is a collection of related data.

c) DataBase Management System (DBMS):

- A collection of Programs that enables users to create and maintain a database.
- Set of Programs to access the data.
- An environment that is both convenient & efficient to use.
- A Software package / system to facilitate the creation & maintenance of a computerized database.

d) Database System: The DBMS software together with the data itself. Sometimes, the applications are also included together called Database System.

e) Meta-data: The information stored in the catalog is called meta data. It describes the structure of the primary database.

Q2. Discuss main characteristic of database approach & how it is differs from traditional file systems.

Ans In traditional file Processing, each user defines & implements the files needed for a specific software application as part of programming the application. This redundancy in defining & storing data results in wasted storage space & in redendant efforts to maintain common data upto date.

Characteristics of database approach :-

- Self-describing nature of a database system.
- Insulation b/w programs & data, & data abstraction.
- Support of multiple views of the data.
- Sharing of data & multiuser transaction processing.

Q3⇒ Discuss the advantages of using the DBMS approach?

Ans⇒ Advantages :-

- Controlling Redundancy : It is necessary for improving the Performance of queries.
- Restricting Unauthorised Access : Restriction on unauthorised users to access information from the database is necessary.
- Providing Storage structures for efficient query Processing : It is responsible for choosing an efficient query execution Plan for each query based on the existing storage

Structures.

- Providing Backup & Recovery: DBMS Provide facility for recovering from hardware or software failures.
- Providing multiple user Interfaces: DBMS Provides a variety of user interfaces. Which includes ~~as~~ query language, Programming language, forms & command code & menu-driven interface.
- Representing complex Relationships among data: DBMS has the capability to represent a variety of complex relationships among the data as well as to retrieve & update related data easily & efficiently.
- Enforcing Integrity Constraints: Integrity constraint involves specifying a data type for each data item.
- Permitting Inferencing & Actions using Rules: Database Systems Provide capability for defining deduction rules for inferencing new information from the stored database facts.
- Flexibility: It is necessary to change the structure of a database as requirements change.

Q4) What are the responsibilities of the DBA & the database designers?

Ans:→ In database environment, the primary resource is the database itself, & the secondary resource is the DBMS & related software. These resources is the responsibility of the database administrator (DBA). DBA is responsible for authorizing access to the database, for co-ordinating & monitoring its use, & for acquiring software & hardware resources as needed. DBA is accountable for problems such as breach of security or poor system response time.

Database designers are responsible for identifying the data to be stored in the database & for choosing appropriate structures to represent & store this data. It is responsibility of database designers to communicate with all prospective database users in order to understand their requirements & to come up with a design that meets these requirements.

Database System - Concepts and Architecture

Database approach is that it provides some level of data abstraction.

Data abstraction : It refers to the suppression of details of data organisation & storage, & highlight only essential features for better understanding of Data.

⇒ Data Model

It is a collection of concepts that can be used to describe the structure of a database.

Structure of database means datatypes, relationships & constraints that should hold for the data.

Categories of data models

(i) Conceptual (high-level, semantic) data models

- Provide concepts that are close to the way many users perceive data.
- Also called entity-based or object-based data models.

(ii) Physical (low-level, internal) data models

- Provide concepts that describe details of how data is stored in the computer.

(iii) Implementation (representational) data models

- Provide concepts that fall between the above two.
- Provide concepts that may be easily understood by end users but that are not too far removed from the way data is organized in computer storage.
- Used by many commercial DBMS implementations (e.g. relational data models).

Conceptual Data Model

It use entities, attributes, & relationships.

- An entity represents a real-world object or concept, such as an employee or a project.
- A relationship among two or more entities represents an association among the entities,
- Entity-Relationship model - a popular high-level conceptual data model.

Representational data model

- This include the widely used relational data model, as well as legacy data models (network & hierarchical models).
- This represent data by using record structures & hence called record-based data models.

⇒ Schema

* **Database schema**

- It is the description of a database.
- Specified during database design & is not expected to change frequently.
- Includes descriptions of the database structure, data types, & the constraints on the database.

* **Schema construct**

It is a component of the schema or an object within the schema.

e.g.: STUDENT

Name	Student_number	Class	Major

⇒ Database State

- The actual data stored in a database at a particular moment in time. It include the collection of all the data in the database.
- Also called database instance.
- It refers to the content of a database at a moment in time.

* **Initial Database State:** Refers to the database state when it is initially loaded into the system.
* **Valid State:** It is a state that satisfies the structure & constraints of the database.

⇒ **Database Schema Vs Database State**

(i) The database schema changes very infrequently.
(ii) The database state changes every time the database is updated.
(iii) Schema is also called intension.
(iv) State is also called extension.

Three - Schema Architecture

- Proposed to support DBMS characteristics of:
  - Program data independence
  - Support of multiple views of data

End users
↓
External level
↓
Conceptual level
↓
Internal level
↓
Stored Database

- The goal of the three-schema architecture is to separate user applications from the Physical database.

- DBMS schemas at three levels :

* Internal schema : It is at the internal level to describe Physical storage structures & access paths (e.g indexes).
  - Typically uses a physical data model.

* Conceptual schema : It is at the conceptual level to describe the structure & constraints for the whole database for a community of users.
  - Hides the details of Physical storage structures.
  - uses an implementation data model.

* External schema : It is at the external level to describe the various user views.
  - It describes the part of the database that a Particular user group is interested in & hide rest.
  - uses representational data model.

- The process of transforming requests & results between levels are called mappings.

- Mappings among schema levels are needed to transform requests & data.
  - Programs refer to an external schema, & are

mapped by the DBMS to the internal schema for execution.

Data Independence

- It is the capacity to change the schema at one level of a database system without having to change the schema at the next higher level.

* **Logical Data Independence**

It is the capacity to change the conceptual schema without having to change the external schemas & their associated application Programs.

* **Physical Data Independence**

It is the capacity to change the internal schema without having to change the conceptual schema.

- When a schema at a lower level is changed, only the mappings between this schema & higher-level schemas need to be changed in a DBMS that fully supports data independence.

- The higher-level schemas themselves are unchanged.

DBMS Languages

- Data Definition Language (DDL).
- Data Manipulation Language (DML).

⇒ Data Definition Language (DDL)

- It is used by DBA & designers to specify the conceptual schema of a database.
- It is also used to define external schemas.

⇒ Data Manipulation Language (DML)

- It is used to specify database retrievals & updates.
- DML commands can be applied. Know as query language.

Types of DML

☐ High Level or Non-Procedural Language

- It specify "what data to retrieve".
- It is also called declarative languages.

Ex:- query in SQL.

☐ Low Level or Procedural Language

- Retrieve data one record at a time.
- include constructs such as looping are needed to retrieve

multiple records.

DBMS Interfaces

⇒ Stand alone query language interfaces

Entering SQL queries at the DBMS interactive SQL-interface. (SQL + oracle).

⇒ Programmer interfaces for embedding DML in a Programming languages

- Embedded Approach : embedded SQL (for C, C++), SQLJ (Java).
- Procedure Call Approach : JDBC for Java, ODBC for other languages.
- Database Programming Language Approach : ORACLE has PL/SQL, a Programming language based on SQL.
- Scripting Languages : Server-side scripting languages such as PHP & Python are used to write database Progras.

⇒ User friendly DBMS interfaces

- Menu Based interfaces for web clients or Browsing.
- App for mobile devices.
- Form-based, designed for users.
- GUI
- Natural Language.
- Keyword based Database search

⇒ Interfaces for the DBA

- Creating user accounts, granting authorizations
- Setting system Parameters.
- Changing Schemas or access Paths.

⇒ Interfaces for Parametric users

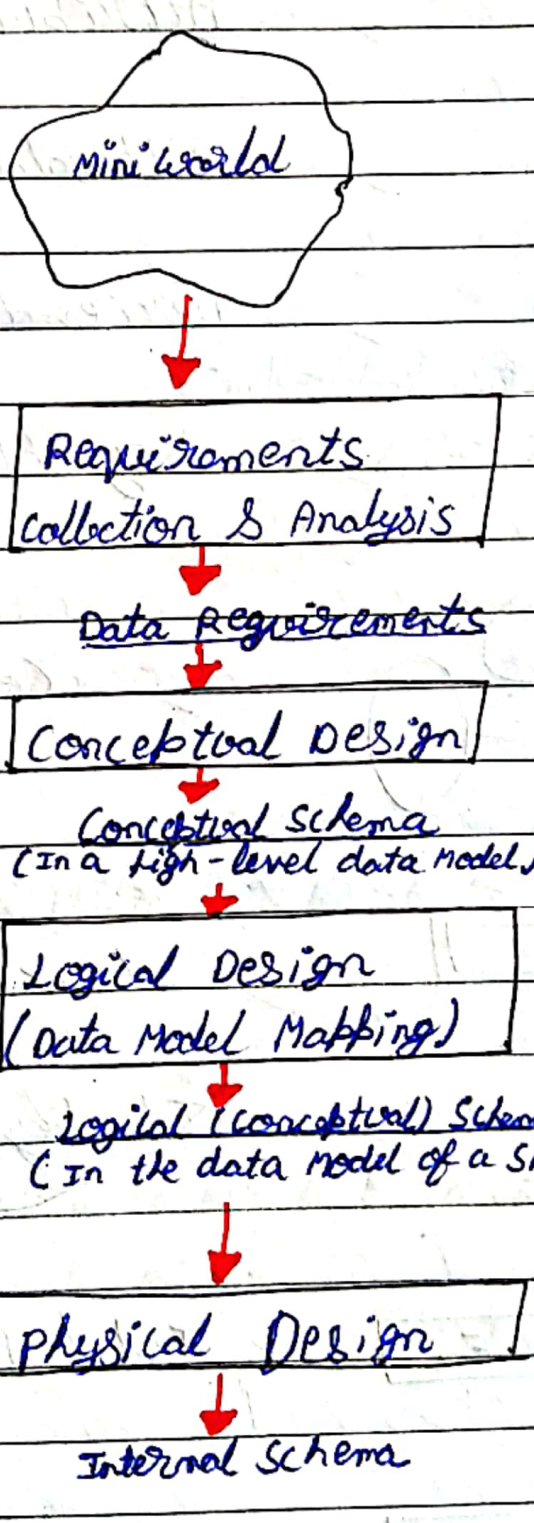
- Bank tellers, Small set of operations that they perform repeatedly.

Entity-Relationship (ER) Model

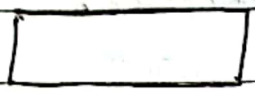
Overview of Database design Process.

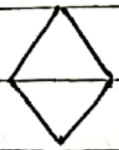
Two main activities :

- Database design
- Applications design



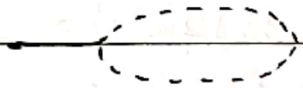
Notation for ER Diagram

*  → Entity

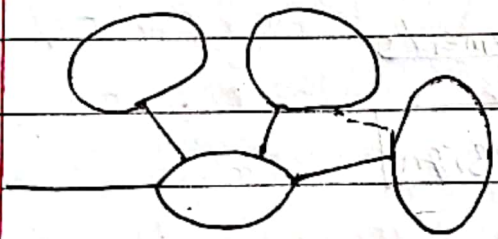
*  → Relationship

*  → Attribute

*  → Multivalued Attribute

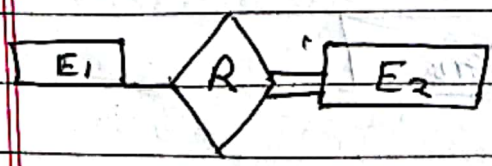
*  → Derived Attribute

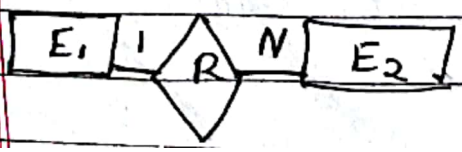
*  → Key Attribute

*  → Composite Attribute

*  → Weak Entity

*  → Identifying relationship

*  → Total Participation of E_2 in R

*  → Cardinality Ratio 1 : N

2A_201500340

Khushal Agarwal

Data Modeling Using the Entity-Relationship Model

=> Entity :- A table or an object that exists & is distinguishable from other objects. It is also defined as a set of entities of the same type.

=> Attribute :- It refers to a database component, such as table. It may also referred to as database field.

=> Relationship Instance :- The relation instance is a table, an instance of a relation is a set of tuples, also called records, in which each tuple has the same no. of fields as the relation schema.

=> Composite attribute :- It can be defined as those attributes which can be divided into sub-parts.

=> Multivalued Attribute :- It is an attribute that can have more than one value associated with the key of the entity.

=> Derived attribute :- The attribute that do not exist in the physical database, but their values are derieved from other attributes present in database.

=> Value Set :- It is a uniquely identifiable set of valid concept representations, where any concept representation can be tested to determine

whether or not it is a number of value set.

Difference b/w single-valued attribute & Multi-valued attribute.

Single valued attribute :
Attributes that can have single value at a particular instance of time are called Single valued. A person can't have more than one age value. Therefore, age of a person is a single-value attribute.

Multi valued attribute :
These are attributes that can have multiple values for a given entity. Ex:- a bank may limit the no. of addresses recorded for a single customer to two.

Degree of a relationship & types

The degree of relationship is the no. of entity types that participate connected to a relationship. The no. of an entity type that is connected to a relationship is the degree of relationship.
Ex:- If we have two entity type 'customer' & 'Account' & they are linked using the primary key & foreign key.

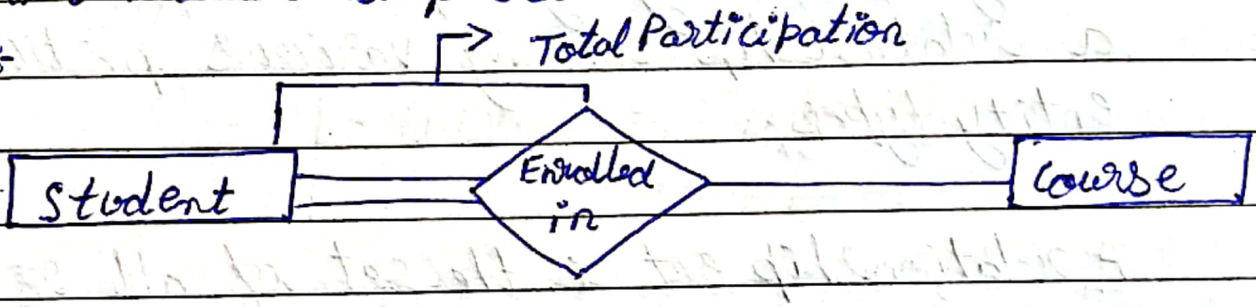
Participation Constraint & its types.

Participation constraint defines the least no. of relationship instances in which an entity must participate. These are two types of participation constraint :-

1> Total Participation

It specify that each entity set must ~~compute~~ compulsorily participate in at least one relationship instance in that relationship set.

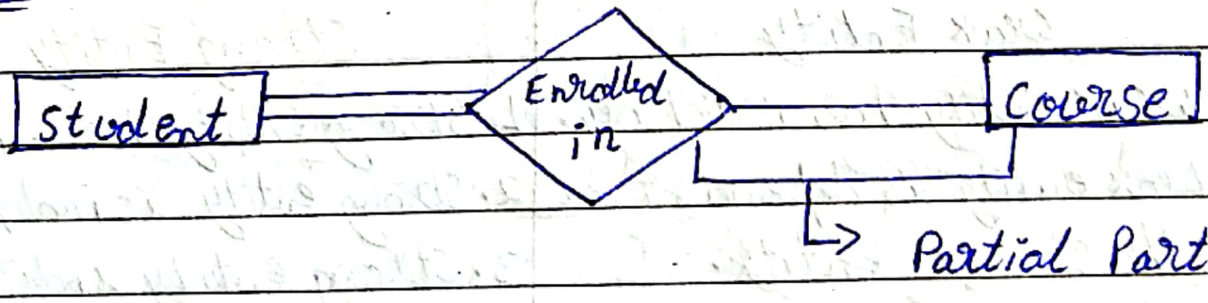
Ex:-



2> Partial Participation

It specify that each entity in the entity set may not participate in the relationship instance in that relationship set.

Ex:-



Name :
Date : / /

Difference b/w relationship type, relationship instances & relationship sets

A relationship type among N entity types defines the act of association among entities of N entities types via some relationship instance.

A relationship instance is the instance that associates an entity from an entity type to another entity of another type, in order to establish a relationship among various participating entity types.

A relationship set is the set of all relationship instances ~~but~~ that participates in any relationship so as to define a relationship b/w various participating entity types.

Difference b/w weak entity type & strong Entity type

	Weak Entity	Strong Entity
1.	Weak entity has Partial Key.	1. Strong entity has a primary key.
2.	Weak entity is dependent on the stronger entity.	2. Strong entity is independent.
3.	Weak entity indicated by double rectangle.	3. Strong entity indicated by a double rectangle.
4.	Weak entity always participate in entity relationships.	4. Strong entity may or may not participate in entity relationship.

Conventions used for displaying an ER Schema as an ER diagram.

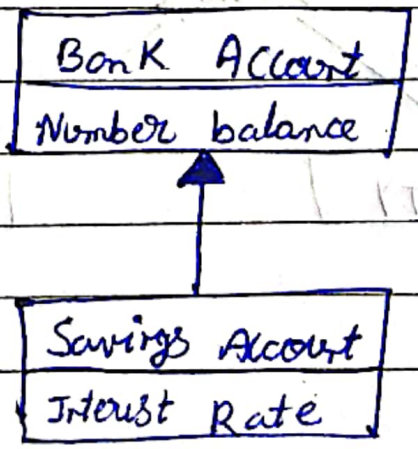
- Entity types are shown in rectangular boxes.
- Relationship types are shown in diamond boxes attached to the participating entity with straight lines.
- Attributes are shown in ovals & each attributes is attached by straight line to its entity type or a relation type.
- Component attributes are attached to the oval representing the composite attribute.
- Multivalued attribute are shown in double ovals & Key attributes have their names underlined.
- Derived attributes are shown in dotted ovals.

Super class & Sub class

Superclass :- It is a general class that is extended by one or more sub classes.

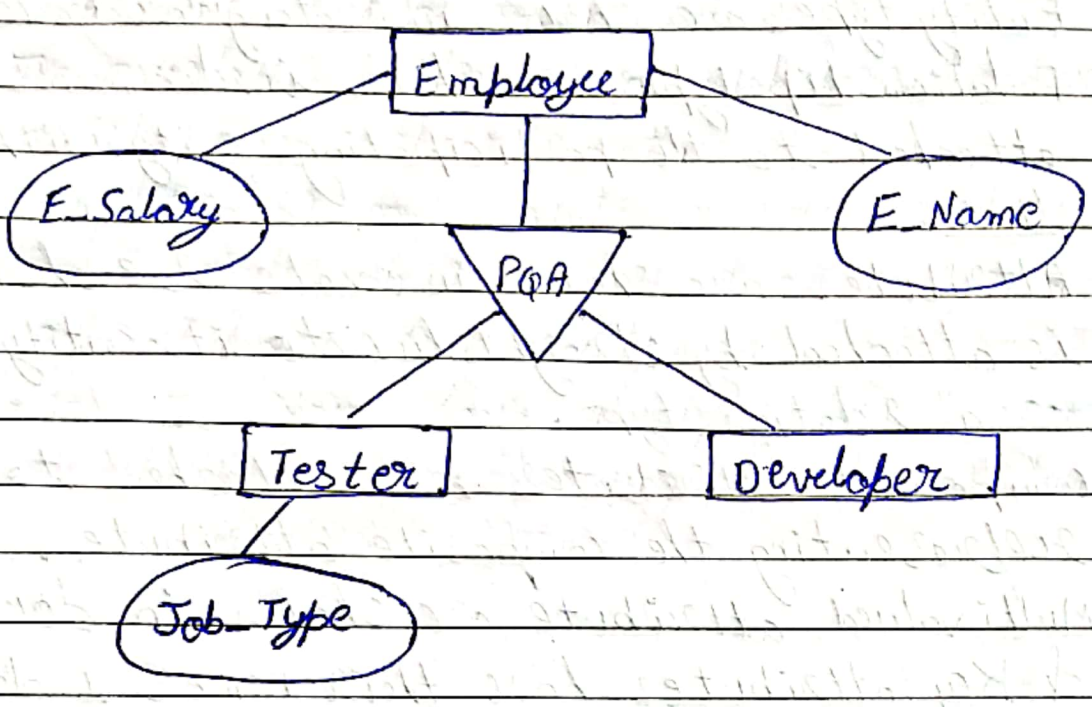
Subclass :- It is more specific class that extends a superclass by inheriting its methods & attributes & then adding its own methods & attributes.

Ex:-



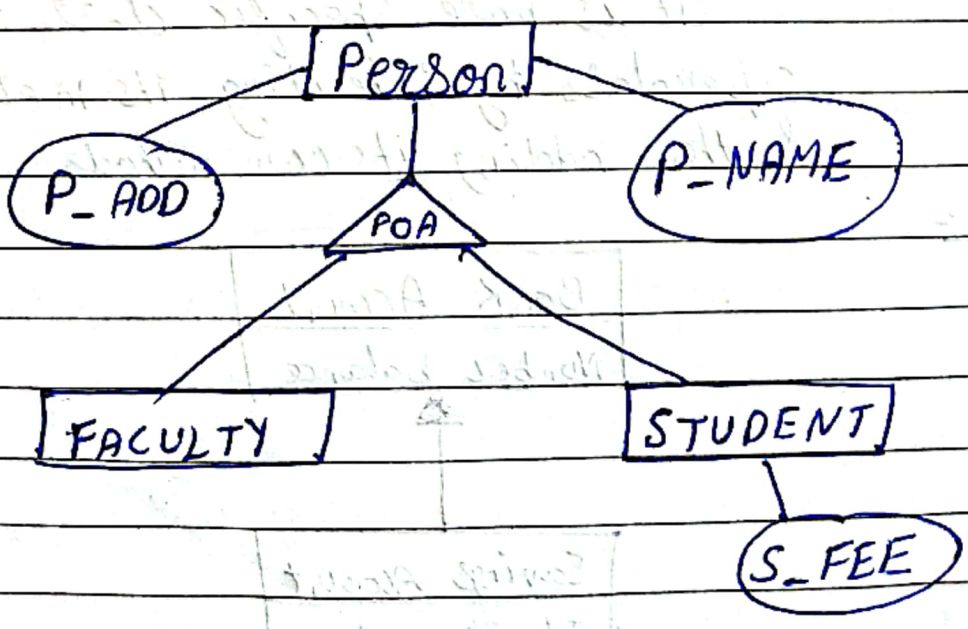
Define

=> Specialization :-



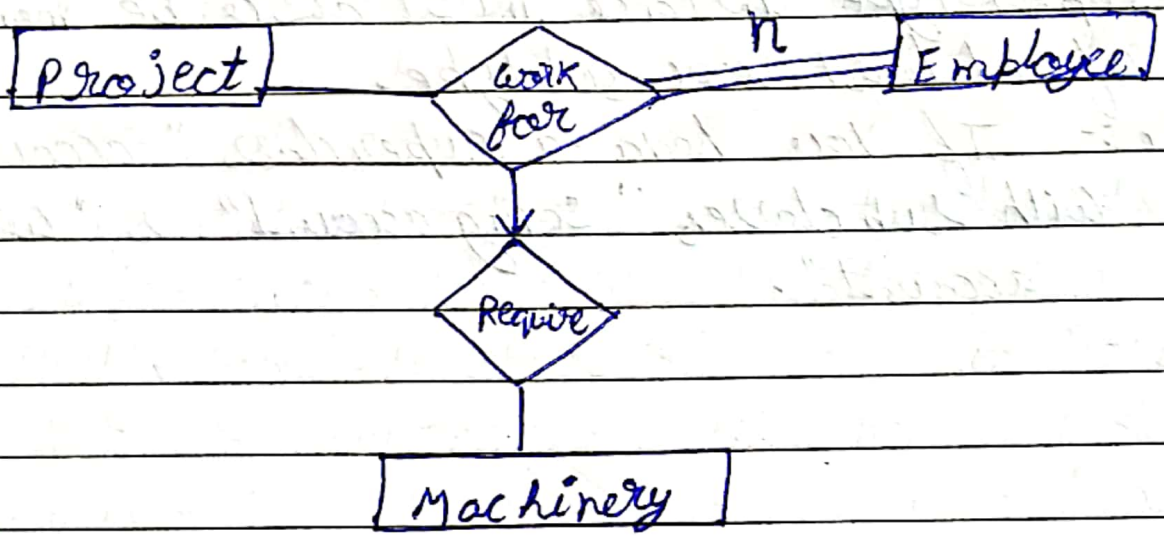
In this case, common attribute like E_Name, E_Sal etc. beomes part of the higer entity (Employee) & specialized attributes like Job_Type bcome part of specialized entity (Tester).

=> Generalisation :-



⇒ Category :- Data categories are tables of data, which are organised by data in rows & columns. Columns are known as data fields. A row of data has entries for one or more columns in the category.

⇒ Aggregation :-



⇒ Specific Relationship :-

Ex :- We have to create a table where details of our name, age, email, address, telephone, number & password needs to be stored. Now, all of this are sensitive information & we would not move to have them bundled together in a single table.

Disjoint, Completeness or Specialization

- The disjoint rule forced substances to have disjoint set of entities.
- The overlap rule forced a subclass to have overlapping sets of entities.
- Completeness constraints decide whether a super type instance must also be member of at least one subtype.

Ex:- If You have a superclass "account" with subclasses "saving account" & "current account".