Unit-3 Data Modeling Using the Entity-Relational Model

-Ankit Pangeni

DATE

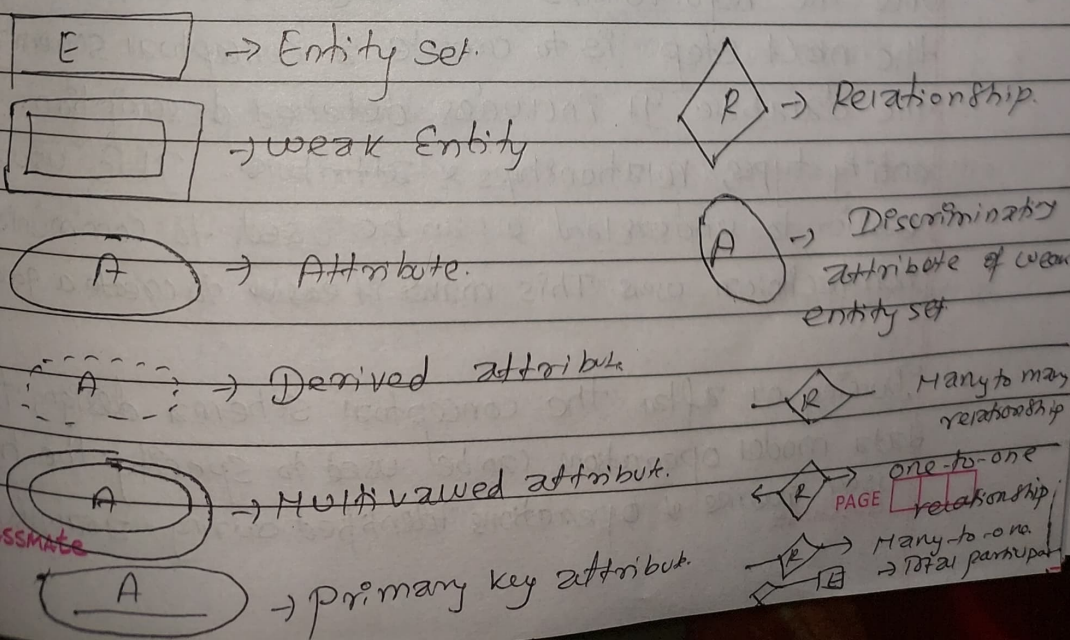
* Introduction to Entity-Relationship Diagram (ER).

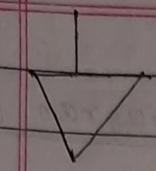
The ER model is based on the conceptual design of the database. It is the widely used data model in the database design. It describes the design of database in terms of entities and relationship among them. It employs three basic notations: Entity sets, Relationship sets and Attribute sets.

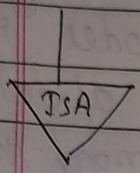
Once the entity types, relationship types & their corresponding attributes have been identified, the next step is to graphically represent these components using E-R diagram.

ER diagram is a graphical tool that demonstrates the relationships among various entities of database. It is used to design overall logical structure of database. While designing ER diagrams, the emphasis is on the schema of database and not the instances.

* Symbols used in ER diagram




→ Total generation



→ ISA (Specialization of generalization)

* Concept of conceptual design: Using High-Level Conceptual Data Models for database design.

Following figure shows a simplified overview of the database design process.

- The 1st step is requirements collection and analysis. Here, the database designers interview the db users to understand & document their data requirements. It is also useful to specify the known functional requirements of the application.

- Once the requirements have been collected & analyzed, the next step is to create a conceptual schema for the database. It includes detailed description of entity types, relationships & attributes. It is usually easier to understand & can be used to communicate with non technical users. This makes it easier to create a good conceptual design.

- During or after the conceptual schema design, the basic data model operations can be used to specify the high-level user queries & operations identified during analysis.

DATE

- The next step in db design is the actual implementation of database, using a commercial DBMS. Most commercial DBMSs use an implementation data model like relational or object-relational db model, so the conceptual schema is transformed from high-level model into the implementation data model. This step is called logical design or data model mapping

- The last step is physical design phase, during which the internal storage structures, file organizations, indexes, access paths, & physical design parameters for the db files are specified. In parallel, application programs are also designed & implemented as database transactions.

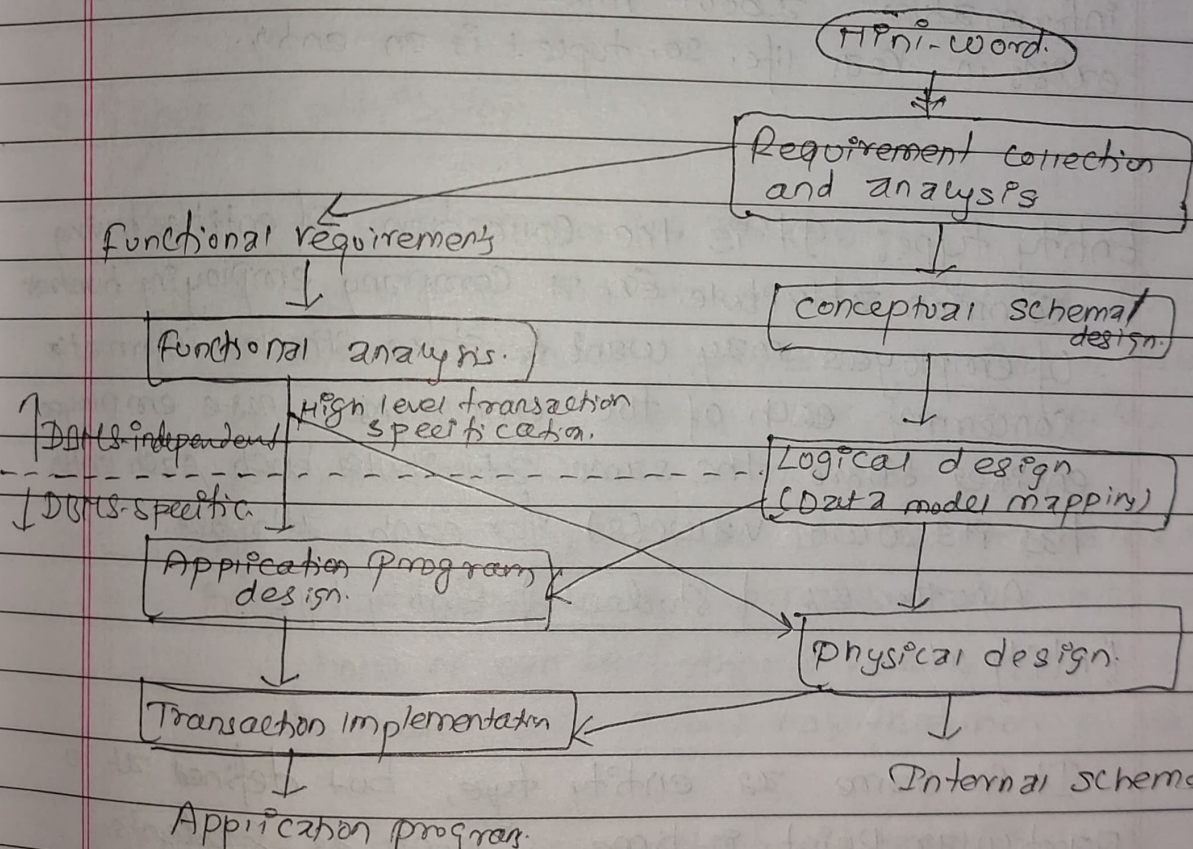


fig: Diagram showing the main phases of database design

DATE

* Elements of ER Diagram / ER Design issues

- **Entity:** Entity is a thing in real world that has physical existence. An entity can be a place, person, object, event or a concept, which stores data in the database. The characteristics of entities must have an attribute, and a unique key. Each entity is made up of some 'attributes' which represent that entity.

Ex: Let 'Ankit' is a particular member of entity type Student. [Ankit] Entity. Here, tuple 1 contains information about Ankit. (id, name & age) which exists in real life. So, tuple 1 is an entity.

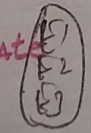
- **Entity type:** It is the collection of entities having common attribute. Ex, a company employing hundreds of employees may want to store similar information concerning each of the employees. These employee entities share the same attributes, buch each entity has its own value(s) for each attribute.

Another Ex. [Student] Entity type.

- **Entity set:**

It is same as entity type, but defined at a particular point in time, such as students enrolled in a class on the first day. Ex: customers who purchased last month, cars currently registered in Florida, etc.

Ex: Let E_1 is an entity having entity type Student
Set of all stds is entity set



Entity set

classmate

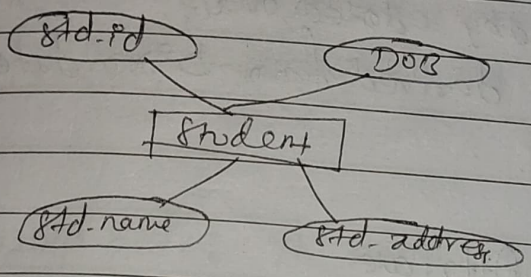
PAGE

DATE

o Attributes:

Attributes are the components of Entities, and are mostly table. They are the properties which define the entity type. It represents the descriptive properties possessed by each component/member of an entity set.
Ex: student_name, student_id, DOB, Age, Address, Mobile_No etc. are the attributes which define entity type student.

→ In ER diagram, attribute is represented by an oval.



o Types of attributes:

i) Atomic vs composite attributes

⇒ An attribute that cannot be divided into smaller independent attribute is Atomic attribute.
Ex: Name, age, address are the attributes of Entity student, which cannot be divided.

⇒ An attribute that can be divided into smaller independent attributes is composite. Ex: address
↓ phone no can be further divided into House no, City, town or mobile no, telephone no. etc.

ii) Single valued vs multivalued attributes.

⇒ An attribute that has only single value for an entity is known as single valued. ex: age of std.
It is represented by singe oval.

⇒ An attribute having multiple values ----
Ex: phone no, email, etc.
Represented by double oval.

classmate

PAGE

DATE

iii) Stored vs derived attribute

=> An attribute that cannot be derived from another attribute & we need to store their value in database is stored attribute. Ex: birth date cannot be derived from age of student.

=> An attribute that can be derived from another attribute & we do not need to store their value in the database. due to dynamic nature is derived. It is denoted by doted oval.

Ex: age is derived from birth date of student.

iv) Null valued Attribute.

=> An attribute, which has not ~~only~~ any value for an entity is known as null valued attribute.

Ex: Student is an entity & phone no. is an attribute. There may be chance that a student can't have any phone number.

v) Key attribute.

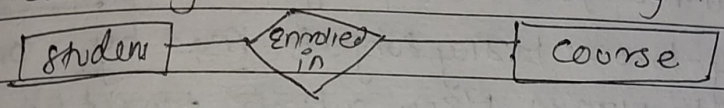
=> An attribute that has unique value of each entity is known as key attribute. Ex, every student has unique roll no. Here, roll no. is a key attribute.

Keys: It is used to uniquely identify any record or row of data from the table. It is also used to establish & identify relationship between tables.

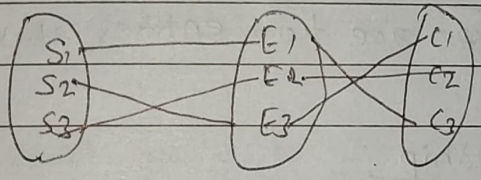
DATE

Concept of Relationship types, relationship sets, roles and constraints.

- **Relationship types:** A relationship type represents the association between entity types. For ex, "Enrolled in" is a relationship type that exists between entity type student & course. It is represented by diamond & connecting lines in ER diagra

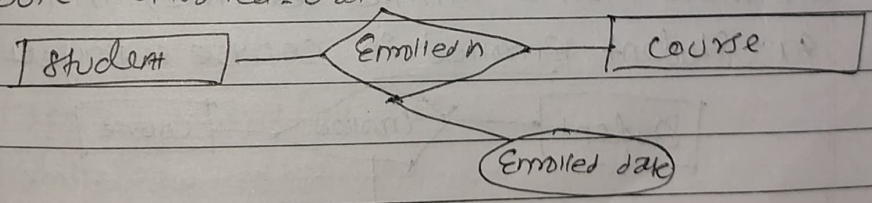


- **Relationship sets:** A set of relationships of same type is known as relationship set. The following relationship set depicts S_1 is enrolled in C_2, S_2 is enrolled in C_1 & S_3 is enrolled in C_3



In another way, the association betn two entity sets is ~~for~~ all.

→ A relationship set may also have attributes called descriptive attributes. Ex, Enrolled in relationship set between entity sets students & course may have attribute enrolled_date.



- **Degree of Relationship:-**

Number of entity sets that participate in a relationship set is called degree of relationship. It can be divided:-

PAGE

classmate

i) Unary Relationship.
ii) Binary Relationship
iii) N-ary Relationship.

- one to one (1:1)
- one to many (1:M)
- Many to one (M:1)
- Many to many (M:M)

Unary relationship (underlined are relationship)

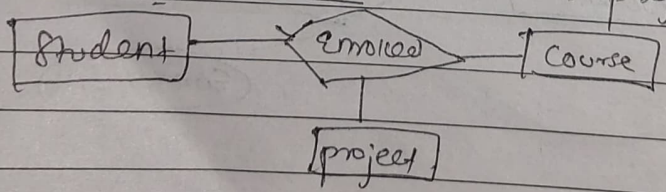
If only one entity set participate in a relation, then it
→ one-to-one: one person is married to one person
→ one-to-many: one employee may manage many workers
→ Many-to-one: Many subjects read by one student
→ Many-to-Many: Many subjects taught by many teachers

Binary relationship.

When there are two entities set participating in a relation, the relationship is called binary.
Ex: A student enrolled in a course.
Student & course are two entities set. & enrolled is relation

N-ary relationship.

When there are n entities participating in a relation, the relationship is called N-ary.
If n=2, then binary relationship. Generally there are more than 2 entities set (n>2) in N-ary relationship
ex: Student Enrolled in course & project



DATE

o Role names & Recursive Relationships.

Each entity type that participates in a relationship type plays a particular role in the relationship, the roll names specifies the role that a participating entity from the entity type plays in each relationships instance, & it helps to explain what the relationship means.

→ In some cases, the same entity type participates in more than one relationship type in different roles. In such cases, the role name is necessary for distinguishing the meaning of each roles. Such relationship types are called recursive relationship & self-referencing relationships.

o Constraints on ER Model / structural constraint in ER

Relationship sets in ER model usually have certain constraints that limit the possible combinations of entities that may involve in the corresponding relationship set. Database content must confirm these constraints. The most important structural constraints in ER are:

i) Mapping cardinalities
ii) Participant constraints.

Mapping cardinalities

The number of times an entity of an entity set participates in a relationship set is call

classmate Type:

PAGE

entity in each

* One-to-one : When each entity set can participate only once in the relationship, the cardinality is one-to-one. Ex: a male can marry to one female and vice-versa

* One-to-many or many to one
When entities in one entity set can take part only once in the relationship set & entities in other entity set can take part more than once in a relationship set, then cardinality is 0-to-m.
Ex: a student can take only one course, but one course can be taken by many students. So, cardinality will be many to 1 (n to 1). It means ~~for n students there can be 1 course~~ for one course there can be n students but for one std there is only one course

* Many to Many
When entities in all entity sets can take part more than once in the relationship, cardinality is n to n.
Ex: a student can take more than one course & one course can be taken by many stds. So, relationship is many to many.

Participation Constraints.

Participation constraints determine whether all or only some entity occurances participate in a relationship. There are two types:

DATE

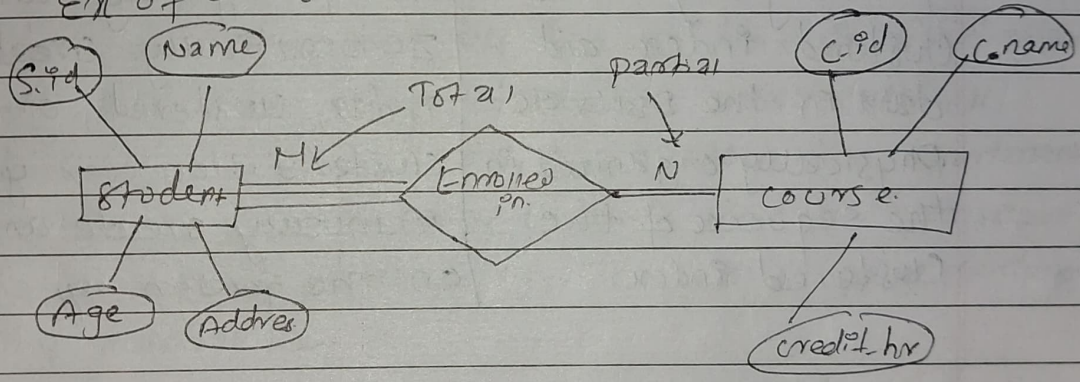
* Total participation Constraint

Here each (all) entity in the entity set must participate in the relationship. Ex: If each student must enroll in a course, then participation of student will be total. It is shown by double line in ER diagram.

* Partial participation Constraint

Here, the entity in the entity set may or may not participate in the relationship. Ex: If some courses are not enrolled by any of the student, then participation of course will be partial.

Ex of both:



o Keys in DBMS

- Super key
- o candidate key
- o Primary key
- o Composite key
- o Foreign key
- o Alternate key
- o compound key
- o Surrogate key

already explained before

PAGE

DATE

* Difference between primary key & Foreign key

Primary key	Foreign key
i) It helps us to uniquely identify a record in the table.	i) It is the field in the table that is the primary key of another table
ii) Primary key never accepts NULL value.	ii) A foreign key may accept multiple NULL values
iii) We can have a single primary key in the table.	iii) We can have multiple foreign keys in the table
iv) Primary key is a clustered index and data in the DBMS are physically organized in the sequence of the clustered index	iv) A foreign key cannot automatically create an index, clustered or non-clustered. However you can manually create an index on the foreign key

* Concept of weak entity types & partial keys.

→ Entity types that donot have key attributes of their own are called weak entity types. An entity type should have a key attribute which uniquely identifies each entity in entity set, but there exists some entity types for which key attribute can't be defined. These are called weak entity type.

CLASSMATE

PAGE

DATE

→ The entity sets which do not have sufficient attributes to form a primary key are known as weak entity sets.

→ And the entity sets which have a primary key are known as strong entity sets.

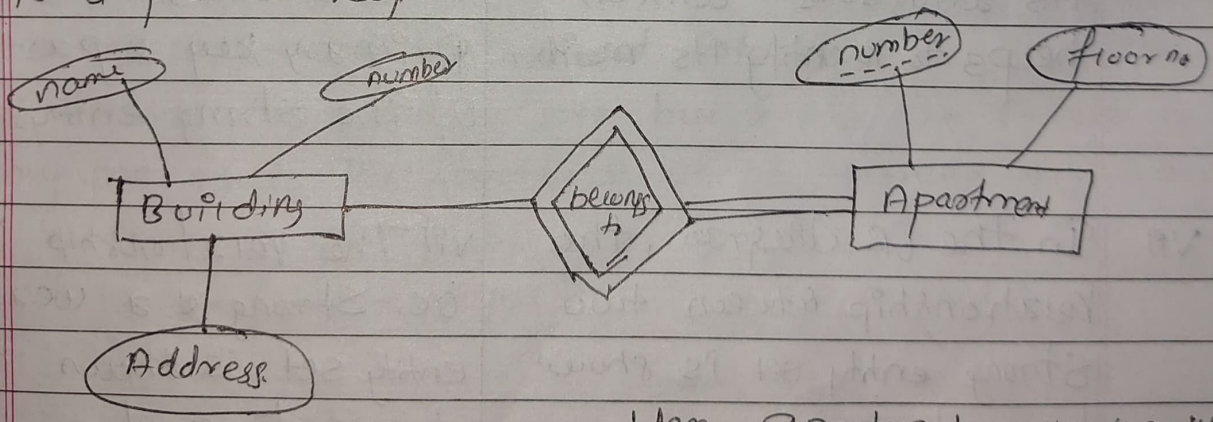
→ Weak entities always has total participation but strong entity may not have total participation.

Partial key: The set of attributes that are used to uniquely identify a weak entity set is called the partial key. It is also known as discriminator.

. It is partially unique & can be combined with other strong entity set to uniquely identify the tuples.

→ They are represented by dashed line in ER diagram

→ Ex: Let's take appointment as weak entity & buiding as strong entity type. Now, the apartment number is not globally unique but it is unique for a particular buiding. Thus, apartment number is a partial key



Here, apartment = weak entity

apartment no. = partial key.

DATE

* Difference between strong entity ~~set~~ and weak entity

	Strong Entity set	Weak entity set.
i)	Strong entity set always has a primary key.	i) It does not have enough attributes to build a primary key.
ii)	It is represented by a rectangle symbol.	ii) It is represented by a double rectangle symbol.
iii)	It contains a primary key represented by the underline symbol.	iii) It contains a partial key which is represented by a dashed underlined symbol.
iv)	The member of a strong entity set is called as dominant entity set.	iv) The member of a weak entity set is called a subordinate entity set
v)	Primary key is one of its attributes which helps to identify its member	v) In a weak entity set, it is a combination of primary key & partial key of the strong entity set.
vi)	In the ER diagram, the relationship between two strong entity set is shown by a diamond symbol.	vi) The relationship between one strong & a weak entity set is shown by a double diamond symbol.
vii)	The connecting line of the strong entity set with the relationship is single	vii) The connecting line of the ~~weak~~ relationship ~~set to~~ is double.

CLASSMATE

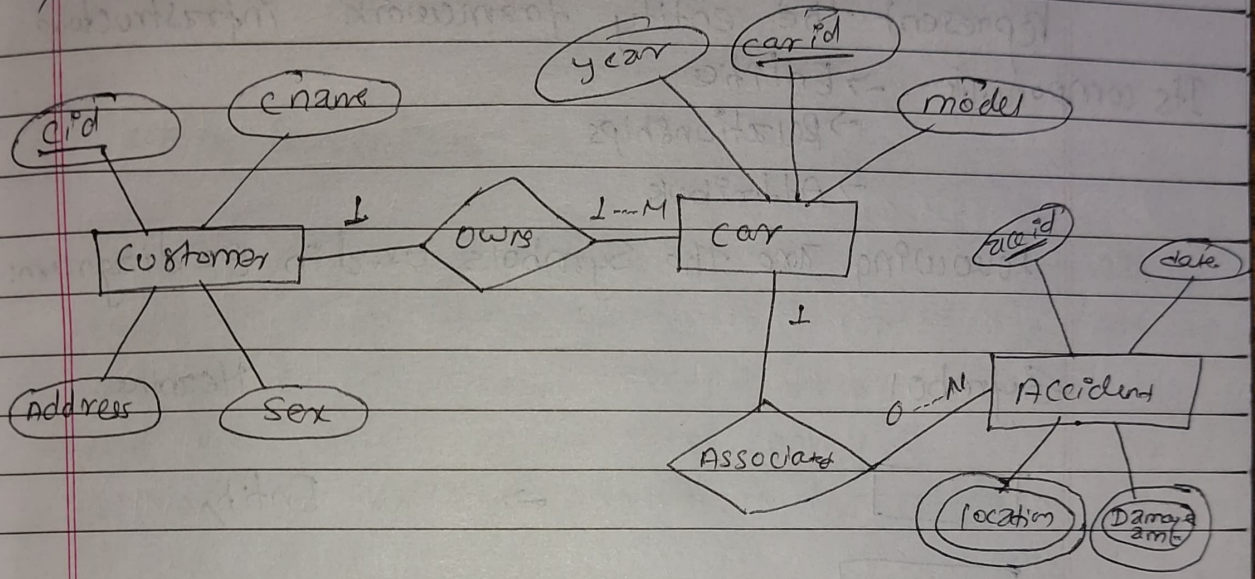
PAGE

DATE

Some examples of ER diagram:

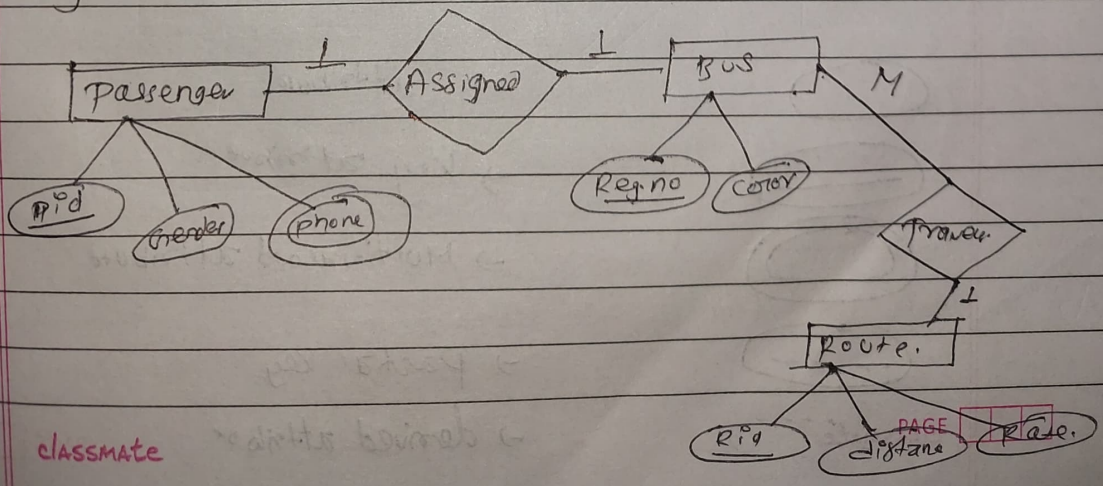
Ex. Construct an ER diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents. Assume attributes of your own interest.

=>



Ex. Consider a bus ticketing system that records information about the passenger, bus and route. Passenger is assigned to a bus travels to route. A bus contains many passengers & a passenger can be assigned into only one bus. Many bus travel in same route but a bus can travel in only one route. The attributes can be generated on your own.

=>



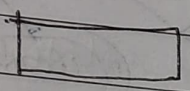
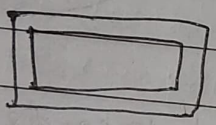
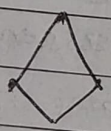
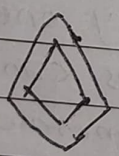
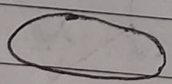
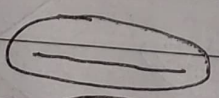
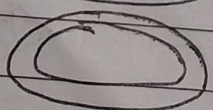
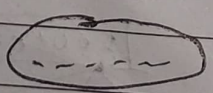
DATE

* Drawing ER diagrams using E-R notations, naming conventions and design issues

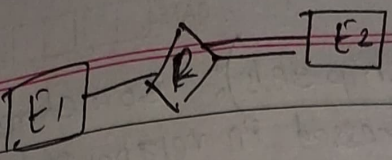
ER-diagram helps to explain the logical structure of database. It includes many specialized symbol, and its meanings make this model unique. The purpose of ER diagram is to represent the entity framework infrastructure

Its components: -> Entities
-> Relationships
-> Attribute.

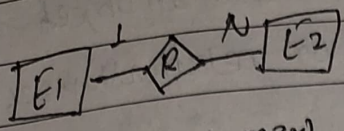
Following are the Symbols used in ER diagram:

Symbol	Meaning
	-> Entity
	-> Weak entity
	-> Relationship.
	-> Identifying relationship
	-> attribute
	-> key attribute.
	-> Multivalued attribute.
	-> Partial key
	-> derived attribute.

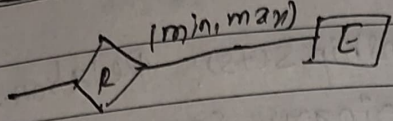
DATE



→ Total participation of E_2 in R.



→ cardinality ratio 1:N for E1:E2 in R.



→ Structural constraint (min, max) on participation of E in R.

* Naming Conventions

→ When designing a database schema, the choice of names for entity types, attributes, relationship types, and roles is not always straightforward.

→ One should choose names that convey, as much as possible, the meanings attached to the different constructs in the schema.

→ We choose to use singular names for entity types, rather than plural ones, because the entity type name applies to each individual entity belonging to that entity type.

→ In ER diagrams, we use the convention that entity type and relationship type names are uppercase letters, attribute names have initial letter capitalized & role names are lowercase letters.

* Design Issues:

o (Use of Entity set vs. Attribute): In the real world situation, it is difficult to select the property as an attribute or an entity set.

DATE

- (Use of Entity sets vs Relationship sets): Sometimes, an entity set can be better expressed in relationship set. Thus, it is not always clear whether an object is best expressed by an entity set or relationship set.

- (Binary vs n-ary relationship sets): Relationships in databases are often binary. Some relationships that appear to be non-binary could actually be better represented by several binary relationships.

* Concept of Enhanced ER (EER) Model:

→ The model that consists of all the modeling concepts of the ER-model and in addition includes consists of the concepts of subclass and superclass and the related concepts of specialization and generalization is called enhanced ER model.

→ It also includes the concept of a category or union type, which is used to represent a collection of objects (entities).

→ It is an improved or enhanced model of ER to handle complex application.

Features:

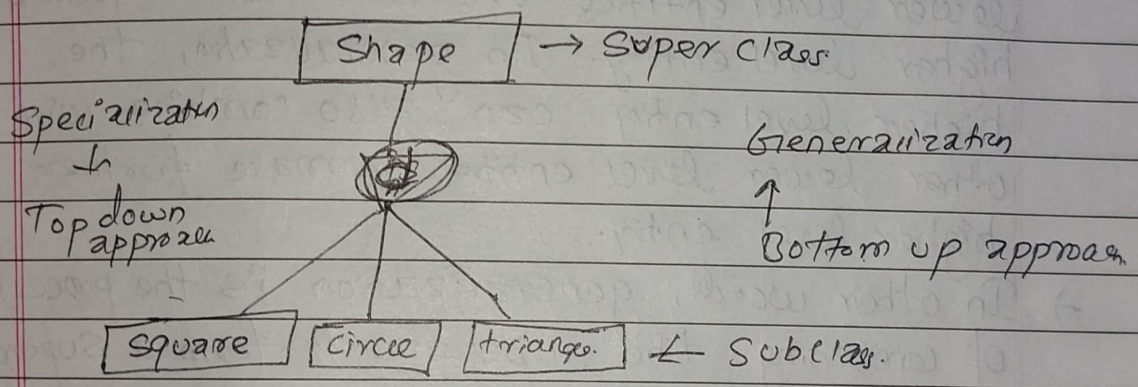
→ Creates more accurate design to database schemas
→ Reflects data properties & constraints more precisely
→ Includes all modeling concepts of ER model.
→ Includes the concept of specialization & generalization
→ used to represent a collection of objects ie. union of objects of different entity types.

DATE

* Subclasses and Superclasses

→ Superclass is an entity type that has a relationship with one or more subtypes. An entity cannot exist in database merely by being member of any super class. For example: shape superclass is having subgroups ~~and~~ as square, circle and Triangle.

→ Subclass is generated from superclass. It is a group of entities with unique attributes. Sub class inherits properties and attributes from its superclass. Ex: square, circle & triangle are the subclass of the super class shape.



* Inheritance

Inheritance is basically the process of building a class on a existing class. The new class contains all the features and functionalities of old class in addition of its own.
The class which is newly created is known as the subclass and the original class is the parent class or the superclass

DATE

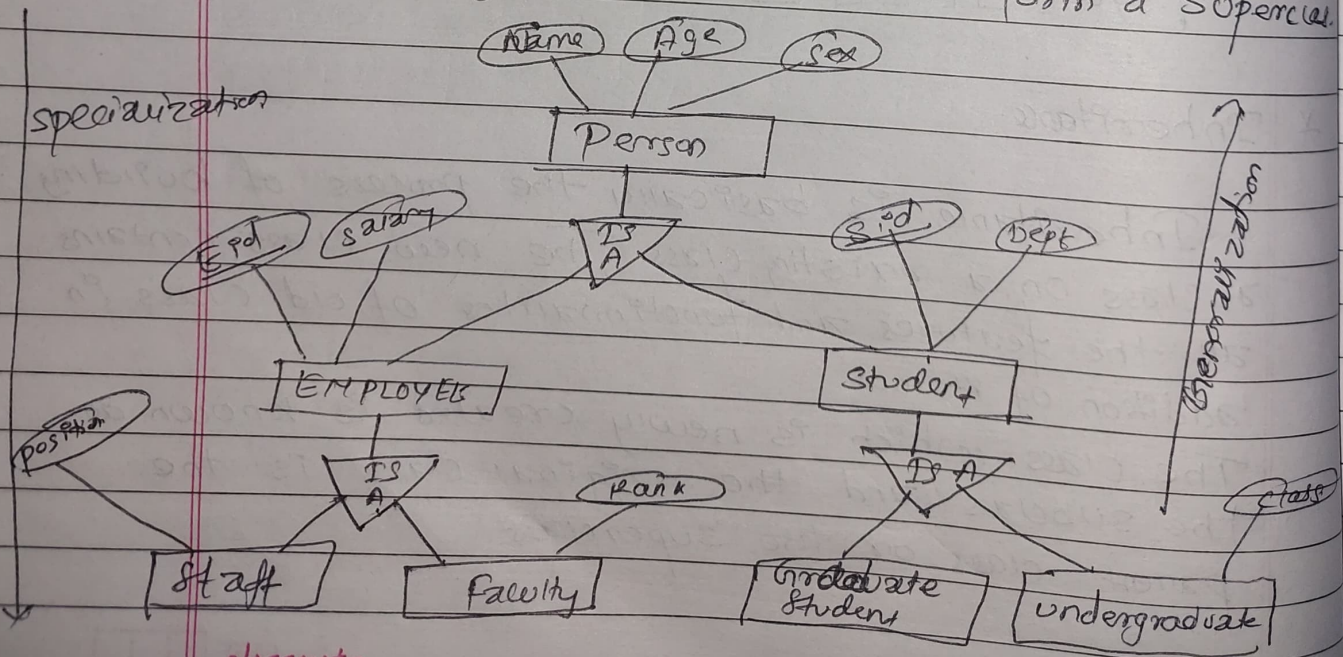
* Concept of Specialization and generalization

→ Specialization is a top-down approach in which one higher level entity can be broken down into two lower level entity.

→ It defines one or more subclass for the super class and also forms the superclass/ subclass relationship.

→ Generalization is opposite to specialization It is a bottom-top approch in which two lower level entities are combined to form a higher level entity. In generalization, the higher level entity can also combine with other lower level entities to make further higher level entity.

→ In other words, generalization is the process of combining the subclasses to form a superclass.



DATE

* Different constraints and characteristics of Specialization and Generalization.

To model real world more accurately by using ER diagram we need to keep certain constraints on it. There are three constraints that may apply to a Specialization / Generalization:

- Membership constraints
  - → condition defined membership constraint
  - → user defined membership constraint.
- Disjoint constraints
  - → Disjoint constraint
  - → overlapping constraint.
- Completeness Constraints
  - → Total Completeness constraint
  - → Partial Completeness constraint

Membership constraints.

* Condition defined membership constraint

→ If we can determine exactly those entities that will become members of each subclass by a condition, the subclasses are called predicate-defined or condition defined membership constraint.

→ Ex: If the employee entity set has an attribute job-type, we can specify the condition of membership in the Secretary subclass by a condition (jobtype = "secretary")

DATE

* User defined membership constraints

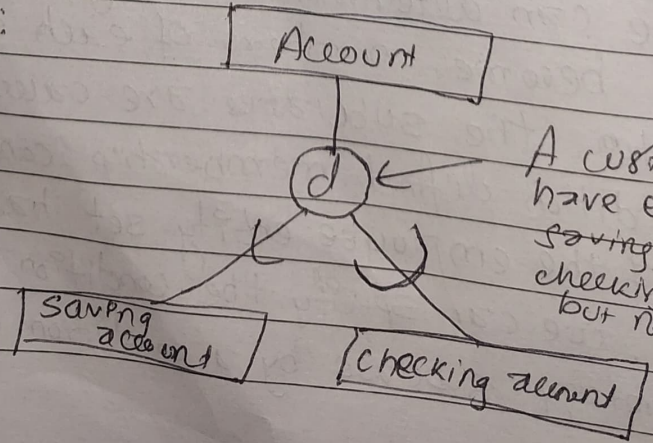
→ If no condition determines membership, the subclass is called user-defined.
→ Membership in a subclass is determined by the database users by applying an operation to add an entity to the subclass
→ Membership in the subclass is specified individually for each entity in the superclass by the user.

Disjoint Constraints.

* Disjoint constraint

→ It specifies that the subclass of the specialization must be disjoint.
→ Here an entity can be a member of at most one of the subclasses of the specialization and it is represented by d in EER diagram.
→ Simply, if an entity can be a member of at most one of the subclasses of the specialization, then the subclasses are disjoint.

Ex:



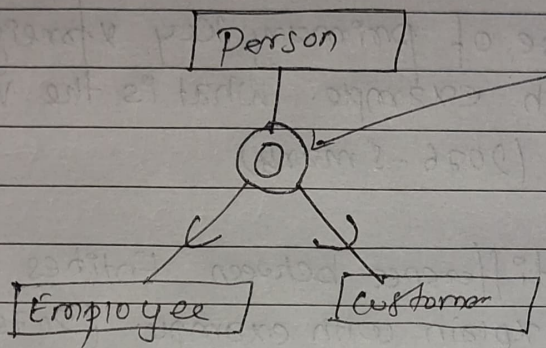
classmate

PAGE

DATE

* Overlapping constraint

-) It specifies that the subclasses are not constrained to be disjoint ie. the same entity may be a member of more than one subclass of the specialization & it is represented by o in EER diagram.



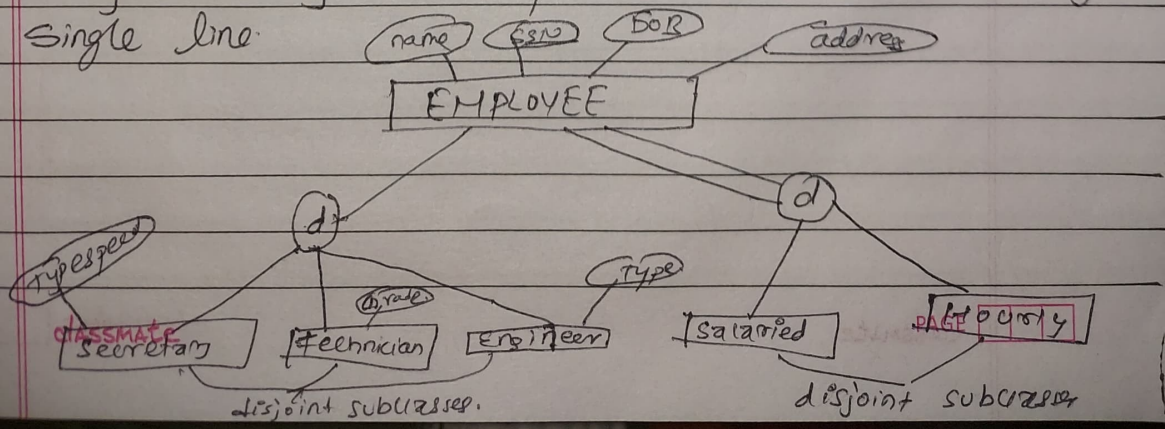
Completeness Constraints.

* Total participation constraint.

-) It specifies that every entity in the superclass must be a member of some subclass in the specialization/ generalization. It is represented by double line in EER diagram

* Partial Participation constraint

-) It specifies that entity in the superclass may or may not belong to a superclass & shown in EER by a single line



DATE

Questions asked from this chapter

Q. What are the components of ER diagram? Explain the functions of various symbols used in ER diagram. Construct an ER diagram to store data in a library of your college. (10 marks-2078)

Q. Explain the use of primary key & foreign key in DBMS with example. What is the role of foreign key? (2076 - 5 marks)

Q. What is the difference between Entities and Entity sets? Explain with example. (2076).

Q. Construct an ER diagram for online book store where customers buy book online. (2075)

Q. What do you mean by ER model? Explain ~~strong~~ & weak entity set. (2073).