Unit-3

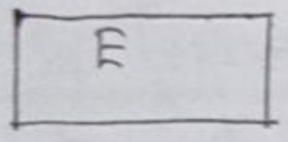
Data Modeling Using the Entity-Relational Model

⊛ Introduction to ER-Diagram:

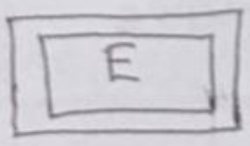
The E-R data model is based on a perception of real world that consist of a collection of basic objects called entities and relation among these objects. In an E-R model a database can be modeled as collection of entities, and relationship among entities.

Once the entity types, relationships types and their corresponding attributes have been identified the next step is to graphically represent these components using E-R diagram. E-R 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 E-R diagrams, the emphasis is on the schema of database and not on the instances.

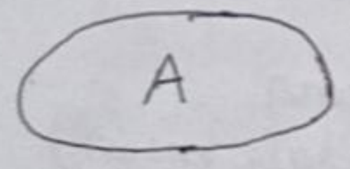
⊛ Symbols used in E-R diagram



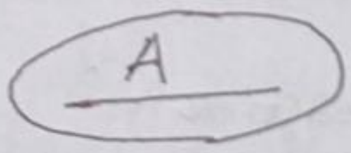
→ Entity set



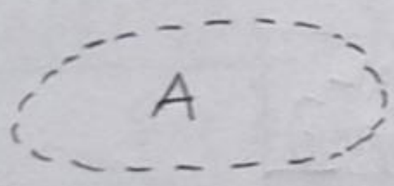
→ Weak entity.



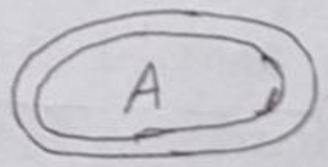
→ Attribute



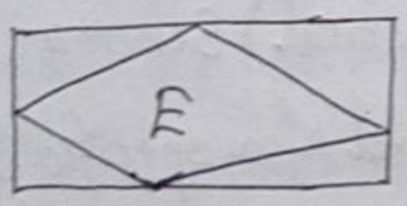
→ Primary key attribute



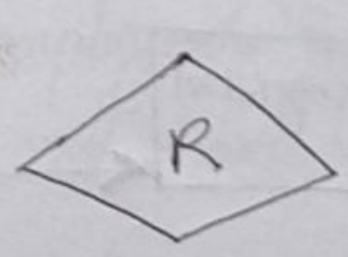
→ Derived attribute



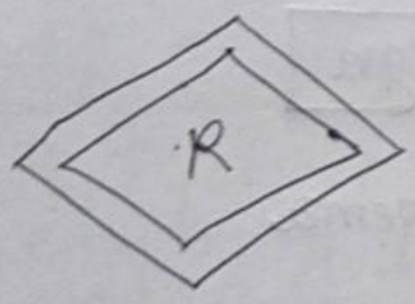
→ Multi-valued attribute



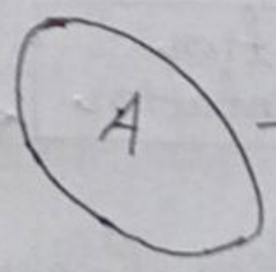
→ Associative entity



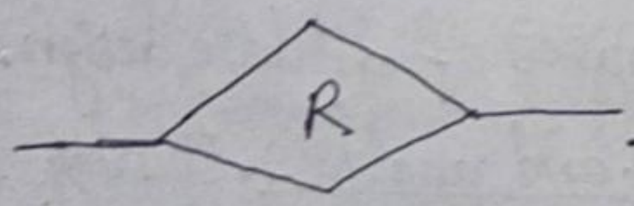
→ Relationship



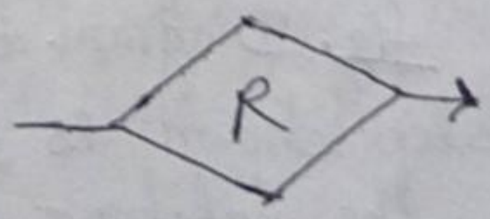
→ Identifying relationship



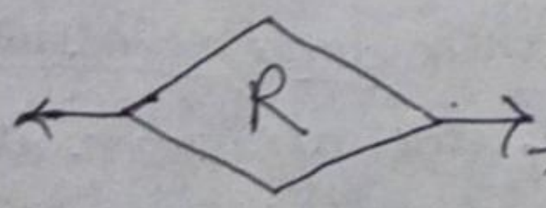
→ Discriminating attribute of weak entity set.



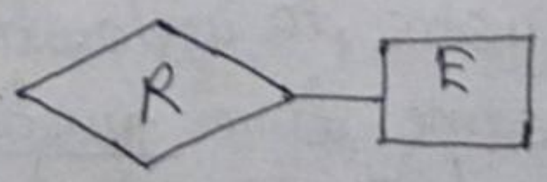
→ Many-to-many relationship



→ Many-to-one relationship

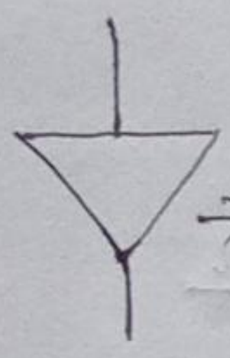


→ One-to-one relationship



→ Total participation.

Scanned by TapScanner



⊛. Concept of Conceptual Design: Using High-Level Conceptual Data Models for Database Design:

The below figure shows simplified overview of the database design process:-

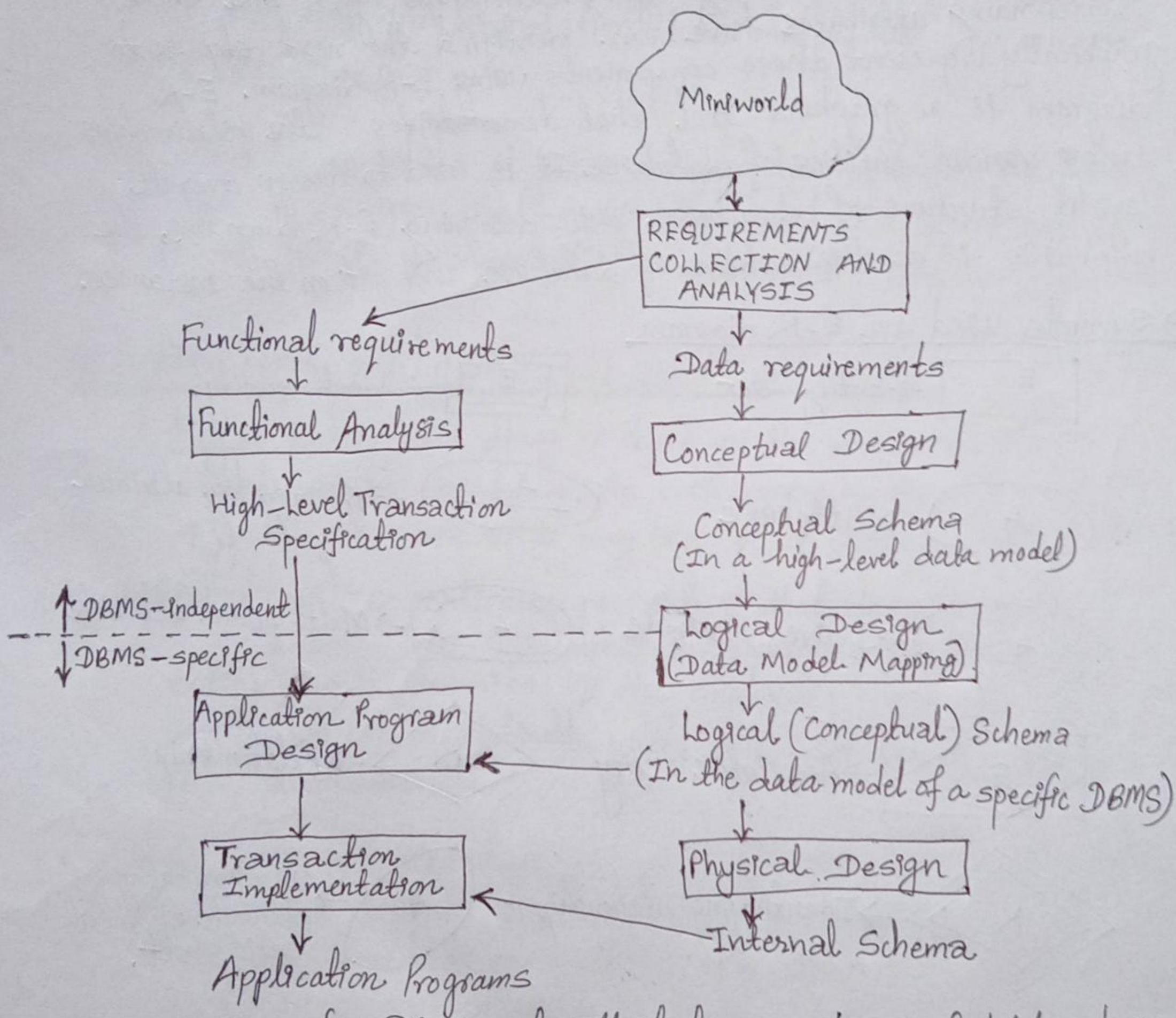


fig. Diagram to Illustrate main phases of database design.

→ The first step shown is requirements collection and analysis. During this step, the database designers interview with expected database users to understand and document their data requirements. At the same time functional requirements of the application are also analyzed. These consist of the user defined operations or transactions that will be applied to database.

Scanned by TapScanner

→ Once the requirements have been collected and analyzed, the next step is to create conceptual schema for the database, using high-level conceptual data model. This step is called conceptual design. This includes descriptions of the data requirements of the users and includes descriptions of the entity types, relationships and constraints.

→ The next step in database design is the actual implementation of the database, using a commercial DBMS. (for e.g. SQL). In this step conceptual schema is transformed from the high-level data model into the implementation data model. This step is called logical design.

→ The last step is the physical design, during which the internal storage structures, file organizations, indexes, access paths and physical design parameters for database files are specified.

⊛ Entity Types, Entity Sets, Attributes, and Keys:-

Entity → Entity is a thing or object in the real world with an independent existence. An entity may be an object with a physical existence or it may be an object with a conceptual existence.

Attributes → The particular properties that describe entity are known as attributes. For example, an EMPLOYEE entity may be described by the employee's name, age, address, salary, job etc. A particular entity will have value for each of its attributes.

Entity Type → Entity type defines a collection (or set) of entities that have same attributes. For example, a company employing hundreds of employees may want to store similar information concerning each of the employees. These employee entities share the same attributes, but each entity has its own value(s) for each attribute.

Entity set → The collection of entities of a particular entity type in the database at any point in time is called an entity set. The entity set is usually referred to use the same name as the entity type, even though they are two seperate

Scanned by TapScanner

concepts. For example, EMPLOYEE refers to both a type of entity as well as the current collection of all employee entities in the database.

Key → An entity attribute has value which is distinct for each individual entity, such a value is called key. It is an important constraint on the entities of an entity type. An entity type usually has one or more attributes whose values are distinct for each individual entity in the entity set, such an attribute is called a key attribute.

⊛. Concept of relationship types and relationship sets, roles and constraints:

A relation type R among n entity types $E_1, E_2, \dots, E_n$ defines a set of associations or a relationship set among entities from entity types. Similar to the case of entity types and entity sets, a relationship type and its corresponding relationship set are referred by same name R. Mathematically, the relationship set R is a set of relationship instances r_i, where each r_i associates n individual entities $(e_1, e_2, \dots, e_n)$, and each entity e_j in r_i is a member of entity set E_j, $1 \leq j \leq n$. Hence, a relationship set can be defined as a subset of the Cartesian product of the entity sets $E_1 \times E_2 \times \dots \times E_n$.

Degree of relationship type → The degree of relationship type is the number of participating entity types. A relationship type of degree two is called binary and degree three is called ternary.

Role names and Recursive Relationships → Each entity type that participates in a relationship type ~~plas~~ plays a particular role in the relationship. The role name signifies the role that a participating entity from the entity type plays in each relationship instance, and it helps to explain what the relationship means.

In some cases the same entity type participates more than once in a relationship type in different roles. In such cases the role name becomes essential for

Scanned by TapScanner

distinguishing the meaning of the role that each participating entity plays. Such relationship types are called recursive relationships and self-referencing relationships.

Constraints on Binary Relationship Types:→

There are two main types of binary relationship constraints: cardinality ratio and participation.

i) Cardinality Ratios for Binary relationships → The cardinality ratio for a binary relationship specifies the maximum number of relationship instances that an entity can participate.

ii) Participation constraint → The participation constraint specifies whether the existence of an entity depends on its being related to another entity through the relationship type. This constraint specifies the minimum number of relationship instances that each entity can participate in and is sometimes called the minimum cardinality constraint.

(*). Concept of Weak Entity Types and Partial Keys:-

Entity types that do not 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 type for which key attribute can't be defined. These are called weak entity type. 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. As the weak entities do not have any primary key, they cannot be identified on their own, so they depend on other entity known as owner entity. Weak entities always has total participation but strong entity may not have total participation. Weak entities are represented with double rectangular box in the ER diagram and the identifying relationships are represented with double diamond. Example: The existence of rooms is entirely dependent on existence of hotel. So room can be seen as weak entity of hotel.

Scanned by TapScanner

Partial key: The set of attributes that are used to uniquely identify a weak entity set is called the partial key. The partial key of the weak entity set is also known as discriminator. It is just a part of the key as only a subset of the attributes can be identified using it. It is partially unique and can be combined with other strong entity set to uniquely identify the tuples. ~~For exam~~ Partial keys are represented by dashed line in ER diagram as shown in figure below. For example:- Let us take a apartment as weak entity and building as a strong entity type. Now, the apartment number is not globally unique i.e, more than one apartment may have same number globally but it is unique for particular building. Thus apartment number is a partial key.

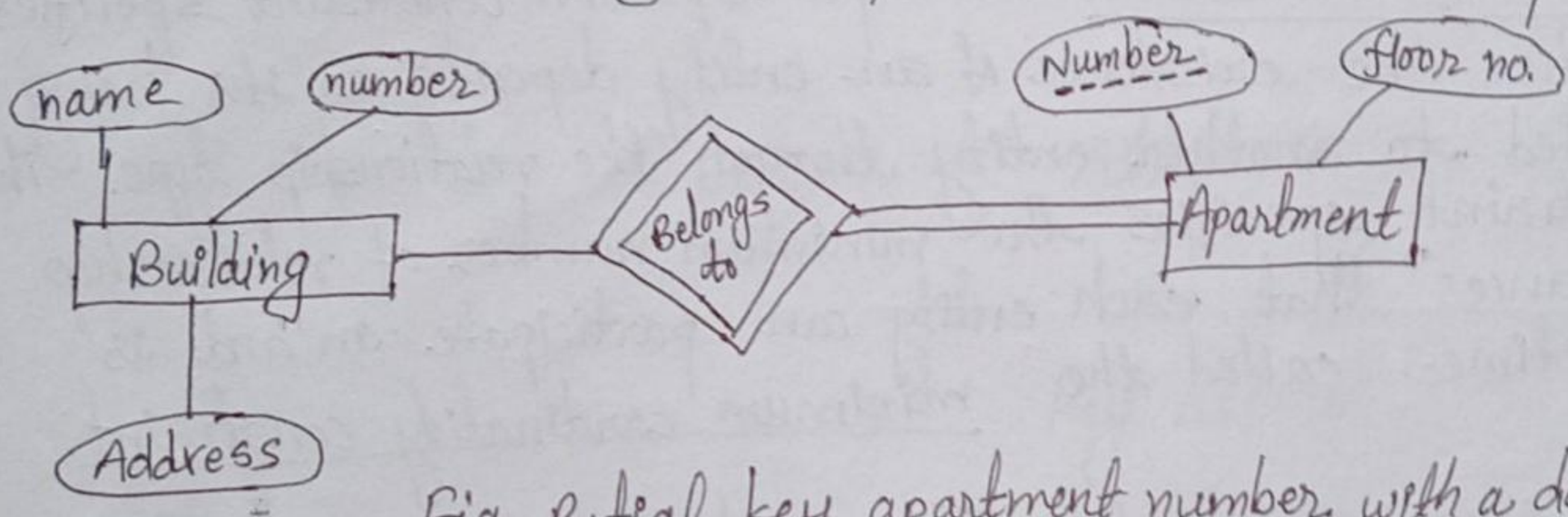


Fig. Partial key apartment number with a dashed line.

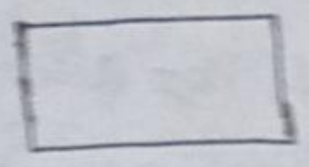
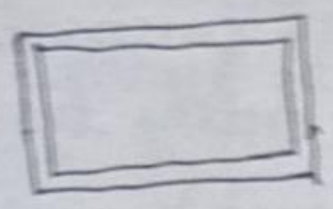
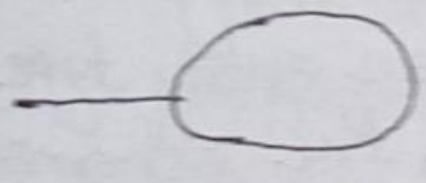
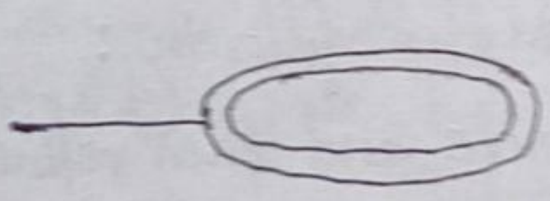
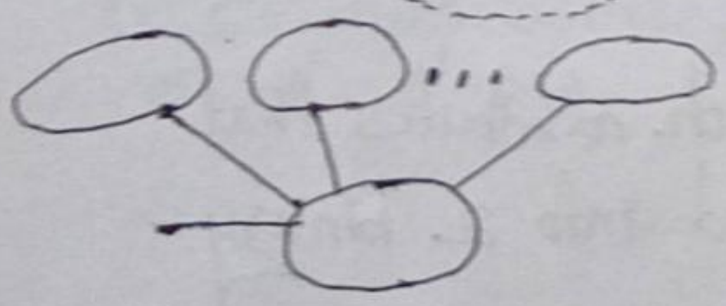
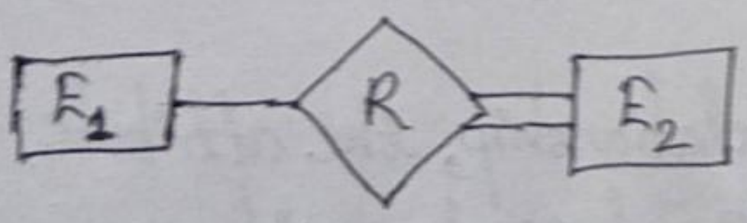
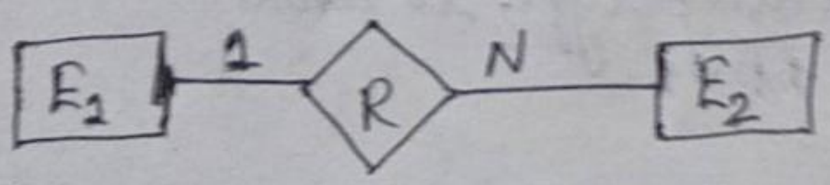
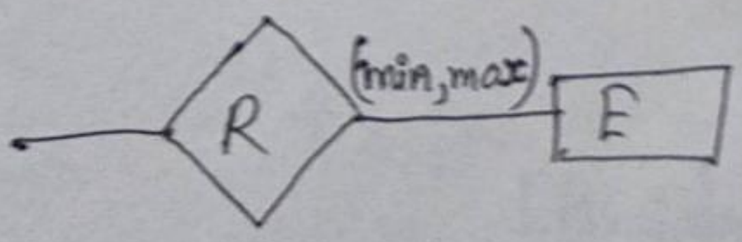
⊛ Drawing ER diagrams using E-R notations, naming conventions and design issues:

ER-diagram helps to explain the logical structure of database. It includes many specialized symbols, and its meanings make this model unique. The purpose of ER diagram is to represent the entity framework ~~inf~~ infastructure. The components of ER-diagram are:-

→ Entities
→ Attributes
→ Relationships.

The following are the different symbols/notations used to draw ER diagrams: (Among these some are already written in first page of note but we will also ~~use~~ write here with some additional notations). These are the design choises for concephual design.

Scanned by TapScanner

Symbol	Meaning
	→ Entity
	→ Weak Entity
	→ Relationship
	→ Identifying Relationship
	→ Attribute
	→ Key attribute
	→ Multivalued attribute
	→ Derived attribute.
	→ Composite attribute.
	→ Total Participation of E_2 in R
	→ Cardinality Ratio 1:N for E_1:E_2 in R
	→ Structural Constraint (min, max) on Participation of E in R.

Scanned by TapScanner

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 their initial letter capitalized, and role names are lowercase letters.

Design Issues:-

It is occasionally difficult to decide whether a particular concept in the miniworld should be modeled as an entity type, an attribute, or a relationship type. In general, the schema design process should be considered an iterative refinement process, where an initial design is created and then iteratively refined until the most suitable design is reached. Some of the refinements that are often used include the following:

→ A concept may be first modeled as an attribute and then refined into a relationship because it is determined that the attribute is a refrence to another entity type.

→ It is often the case that a pair of such attributes that are inverses of one another are refined into a binary relationship.

→ Once an attribute is replaced by a relationship, the attribute itself should be removed from the entity type to avoid duplication and redundancy.

Scanned by TapScanner

⊛. Concept of Enhanced ER (EER) Model:-

The model that consists all the modeling concepts of the ER-model and in addition includes/consists the concepts of subclass and superclass and the related concepts of specialization and generalization is called enhanced ER (i.e, EER) model. Also it includes concept of a category or union type, which is used to represent a collection of objects (entities). EER model is the improved or enhanced form of ER model to handle the complex applications better.

Features:

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

Subclasses and Superclasses:

Superclass is an entity type that has a relationship with one or more subtypes. An entity cannot exist in database ~~just~~ only by being member of any superclass. For example: Shape superclass is having sub groups as Square, Circle and Triangle in the figure below.

Subclass is a group of entities with unique attributes. Subclass inherits properties and attributes from its superclass. For example: Square, Circle and Triangle are the subclass of Shape superclass.

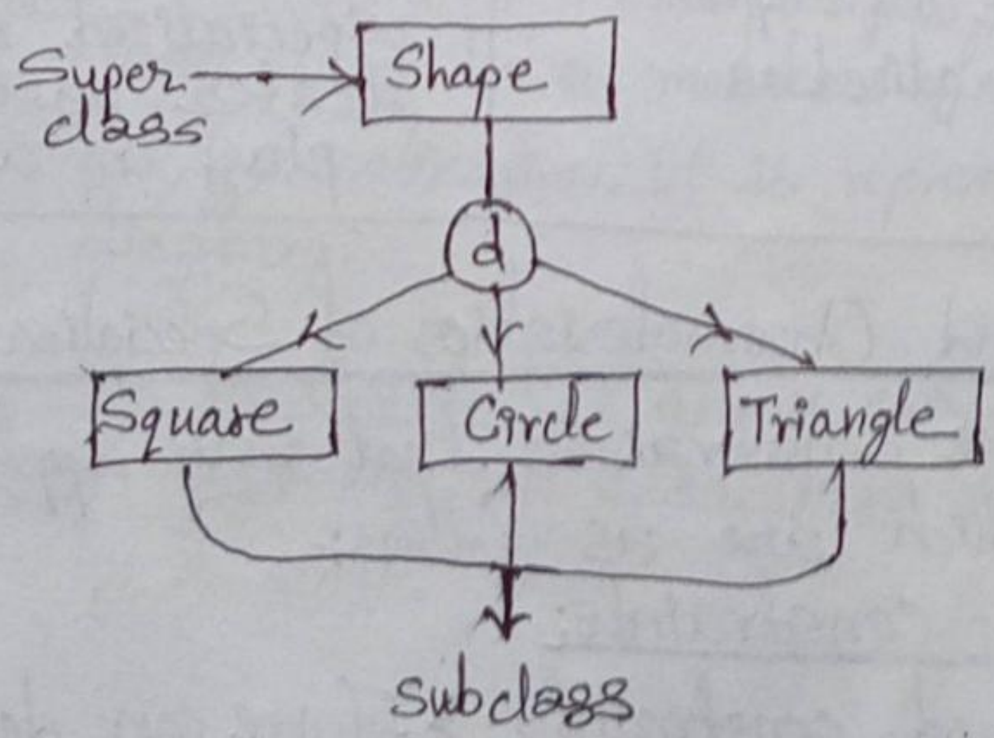


Fig: Subclass superclass relationship.

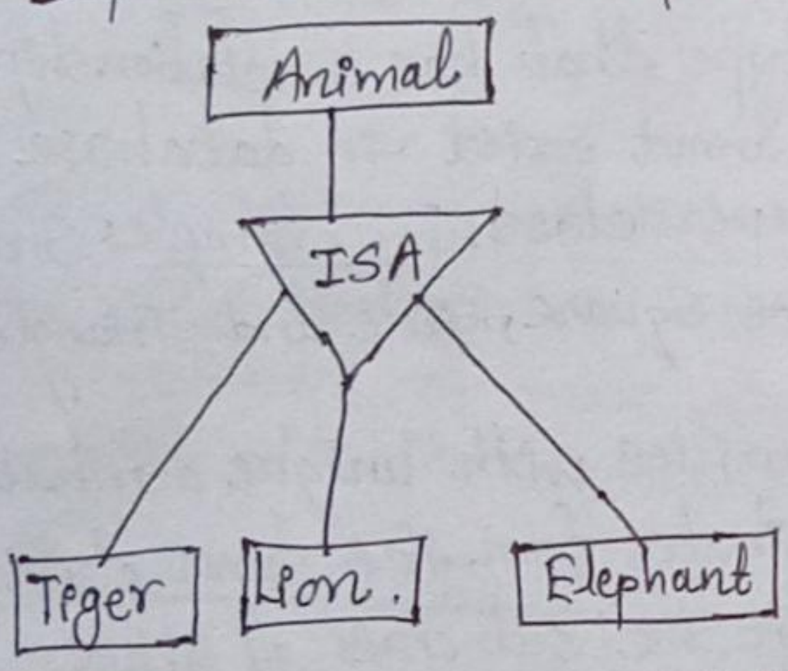
Scanned by TapScanner

⊛. Concept of Specialization and Generalization:

Specialization → It is a top-down approach in which one higher level entity can be broken down into two lower level entity. In specialization, a higher level entity may not have any lower-level entity sets. It maximizes the difference between the members of an entity by identifying the unique characteristic or attributes of each member. It defines one or more subclass for super class and also forms the super class/sub class relationship.

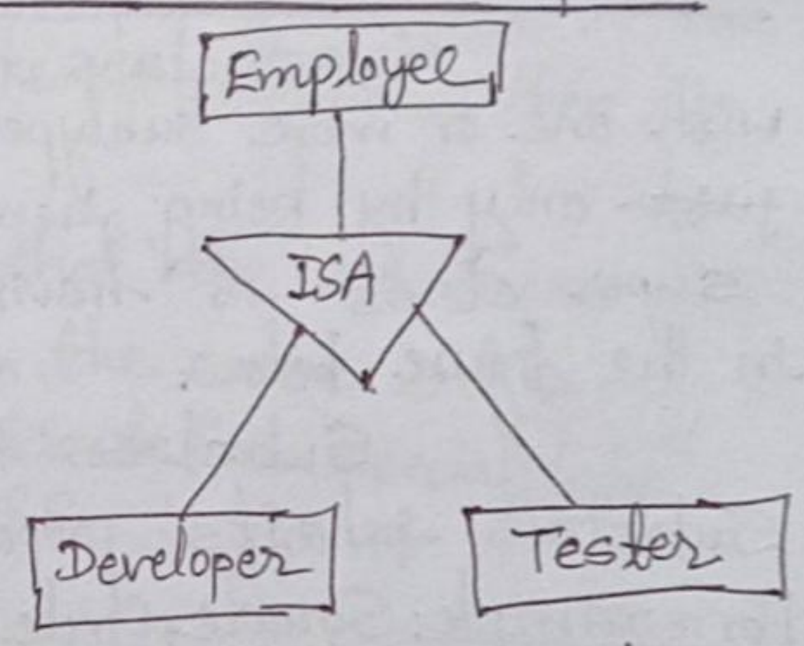
Generalization → Generalization is a bottom-up approach in which two lower level entities combine to form a higher level entity. In generalization, the higher level entity can also combine with other lower level entities to make further higher level entity. In this concept sub-classes are combined to form a super-class.

Generalization Example



Here; Tiger, Lion and Elephant can all be generalized as animals.

Specialization Example



Here; Employee can be Specialized as developer or tester base on what role they play in an organization.

⊛. Constraints and Characteristics of Specialization and Generalization:-

There are three constraints that may apply to specialization/generalization which are as follows:-

@. Membership constraints:

1) **Condition defined constraints** → If we can determine exactly those entities that ~~are~~ will become members of each subclass by a condition, the subclasses are called condition-defined classes.

Scanned by TapScanner

Whenever any tuple is inserted into the database, its membership in the various lower level entity-sets can be automatically decided by evaluating the respective membership predicates. Similarly, when a tuple is updated, its membership in the various entity sets can be re-evaluated automatically.

ii) User defined constraints → If no condition determines membership, the subclass is called user-defined. Membership in the 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 for the user.

(b) Disjoint constraints:

i) 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 subclass of specialization and it is represented by d in EER diagram. If an entity can be a member of at most one of the subclass of the specialization, then the subclass are called disjoint.

ii) Overlapping constraint → It specifies that the subclasses are not constrained to be disjoint i.e, the same entity may be member of more than one subclass of the specialization and it is represented by o in EER diagram.

(c) Completeness constraints:

i) Total participation constraint → It specifies that every entity in the super class must be a member of some subclass in the specialization/generalization. It is represented by double line in EER diagram.

ii) Partial participation constraint → It allows every entity in the super class may not belong to a subclass and shown in EER diagram by a single line.

Scanned by TapScanner

⊛. Differences between Primary key and foreign key:

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

⊛. Differences between strong entity set and weak entity set:

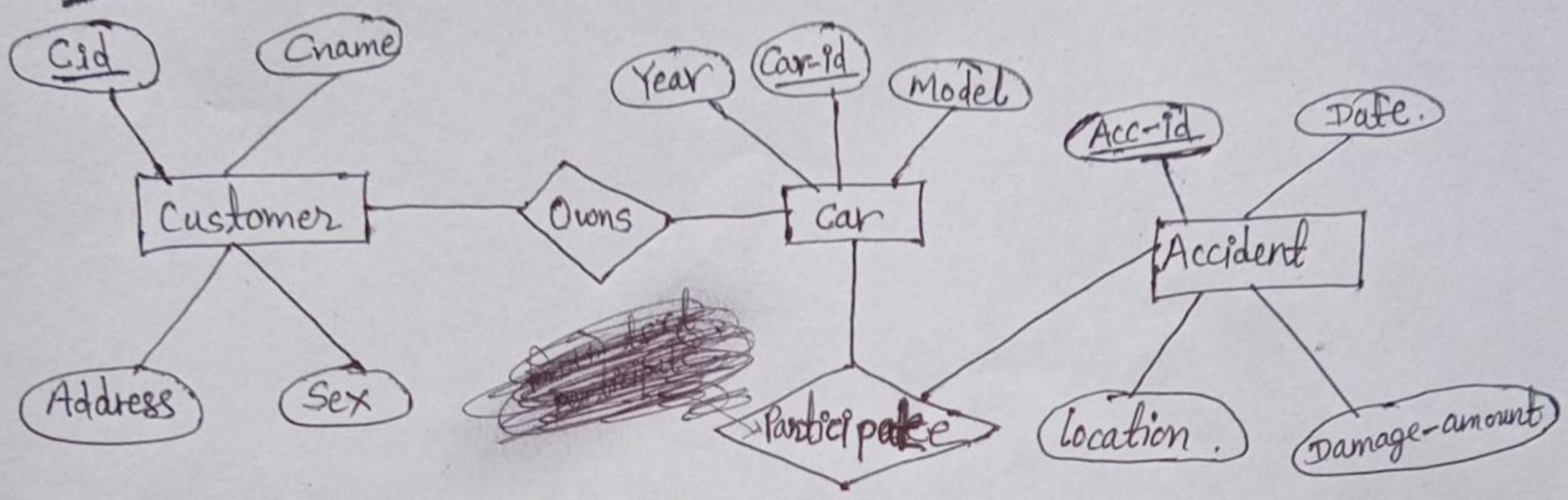
Strong Entity Set.	Weak Entity Set.
i) Strong entity set always has a primary key.	i) It does not have enough attributes to build a primary key.
ii) It is represented by a rectangle symbol.	ii) It is represented by a double rectangle symbol.
iii) It contains a primary key represented by underline symbol.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

Scanned by TapScanner

⊛. Examples of ER-diagram: [from kec book. page no.59]

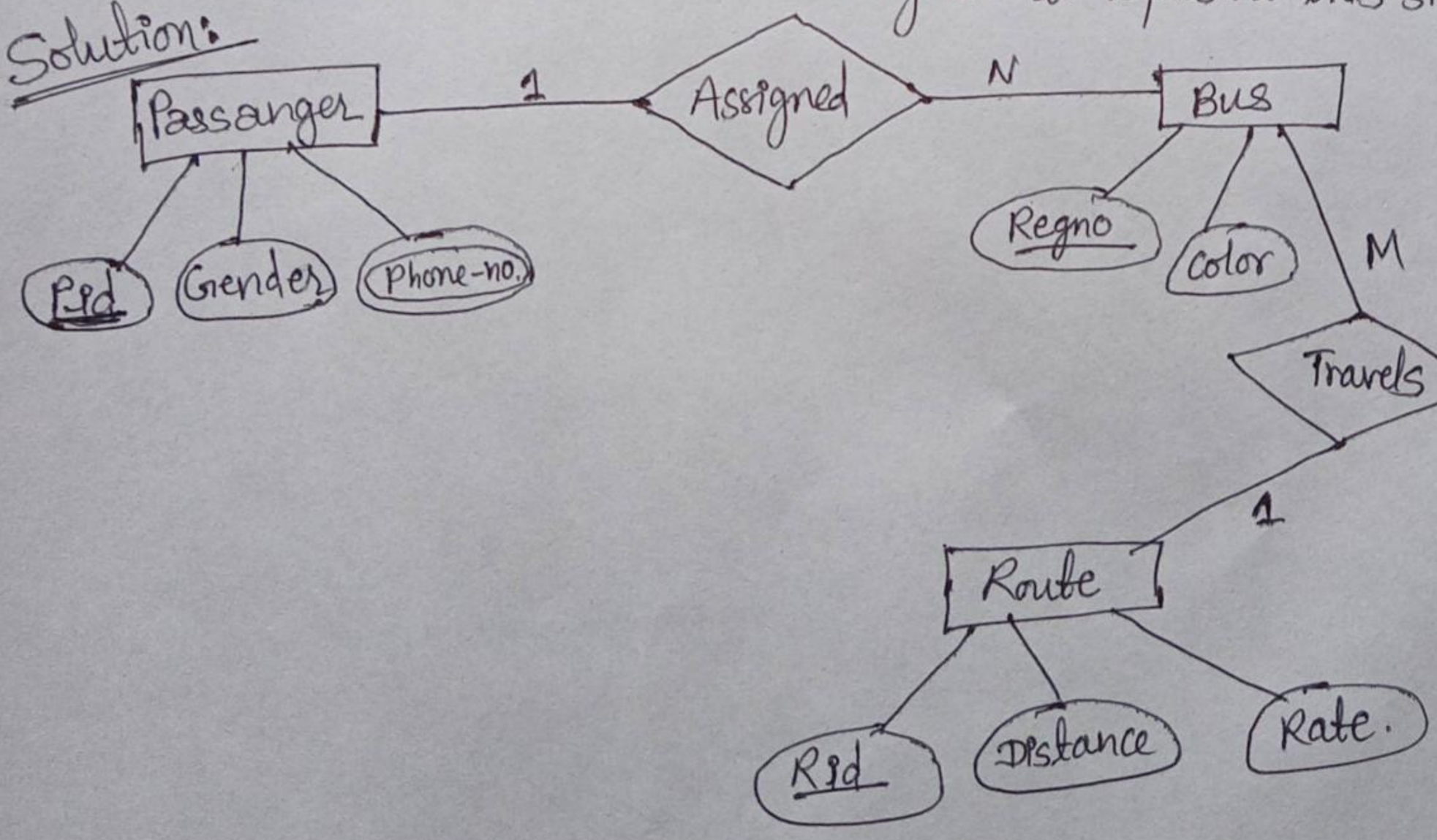
Example 1: Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents. Assume attributes of your own interest:

Solution:



Example 2: Consider a bus ticketing system that records information about the passanger, bus and route. Passanger is assigned to a bus travels to route. A bus contains many passangers and a passanger can be assigned into only one bus. Many buses travel in same route but a bus can travel in only one route. The attributes of passanger are ~~unique-pass~~ pid (unique), gender and telephone (multi-valued). Similarly bus contains rengo (unique) and color and route contains rid (unique), distance and rate (based on distance). Now draw E-R diagram to represent this situation.

Solution:



Scanned by TapScanner