UNIT - II

2 Relational Model

Syllabus

ER and EER diagrams : *Components of ER model, Conventions, Converting ER diagrams into tables.*

Relational Model : *Basic concepts, Attributes and Domains, Codd's rules.*

Relational Integrity : *Nulls, Entity, Referential integrities, Enterprise constraints, Views, Schema diagram.*

Contents

Part I : ER and EER Diagrams

2.1 Components of ER Model

SPPU : Aug 17, Marks 2

- Entity Relational model is a model for identifying entities tobe represented in the database and representation of how those entities are related.
- The ER data model specifies enterprise schema that represents the overall **logical structure** of a database graphically.
- E-R diagrams are used to model real-world objects like a person, a car, a company and the relation between these real-world objects.

Features of ER model

i) E-R diagrams are used to represent E-R model in a database, which makes them easy to be converted into relations (tables).

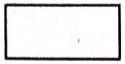
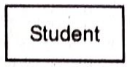
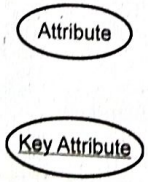
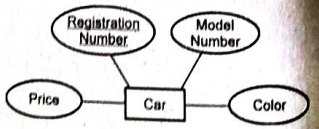
ii) E-R diagrams provide the purpose of real-world modeling of objects which makes them intently useful.

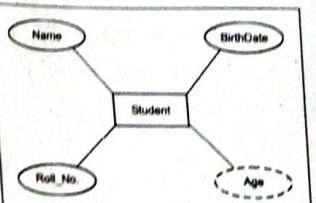
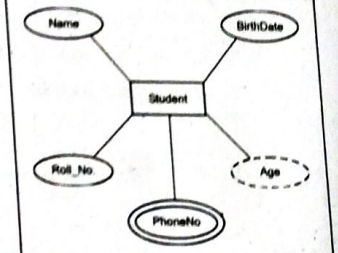
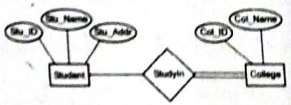
iii) E-R diagrams require **no technical knowledge** and no hardware support.

iv) These diagrams are very **easy to understand** and easy to create even by a naive user.

v) It gives a standard solution of **visualizing** the **data logically.**

Various Components used in ER Model are -

Component	Symbol	Example
Entity : Any real-world object can be represented as an entity about which data can be stored in a database. All the real world objects like a book, an organization, a product, a car, a person are the examples of an entity.		
Relationship : Rhombus is used to setup relationships between two or more entities.		
Attribute : Each entity has a set of properties. These properties of each entity are termed as attributes. For example, a car entity would be described by attributes such as price, registration number, model number, color etc		

Derived attribute : Derived attributes are those which are derived based on other attributes, for example, age can be derived from date of birth. To represent a derived attribute, another dotted ellipse is created inside the main ellipse		
Multivalued attribute : An attribute that can hold multiple values is known as multivalued attribute. We represent it with double ellipses in an E-R Diagram. E.g. A person can have more than one phone numbers so the phone number attribute is multivalued.		
Total participation : Each entity is involved in the relationship. Total participation is represented by double lines.		

Review Question

1. *List different components used in ER diagram with their meaning.* SPPU : Aug 17, Marks 2

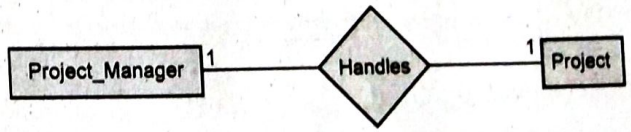
2.2 Conventions

SPPU : Oct.-19, Marks 2

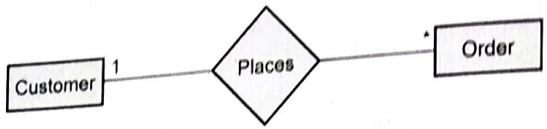
2.2.1 Mapping Cardinality Representation

There are four types of relationships that are considered for key constraints.

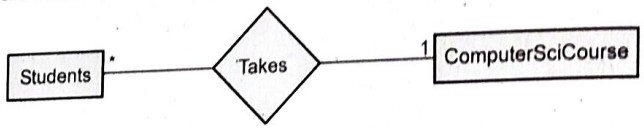
i) **One to one relation :** When entity A is associated with at the most one entity B then it shares one to one relation. For example - There is one project manager who manages only one project.



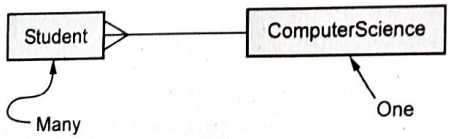
ii) One to many : When entity A is associated with more than one entities at a time then there is one to many relation. For example - One customer places order at time.



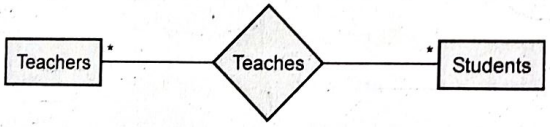
iii)Many to one : When more than one entities are associated with only one entity then there is is many to one relation. For example - Many student take ComputerSciCourse.



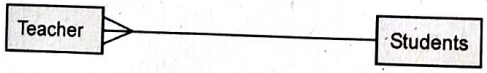
Alternate representation can be



iv) Many to many : When more than one entities are associated with more than one entities. For example -Many teachers can teach many students.

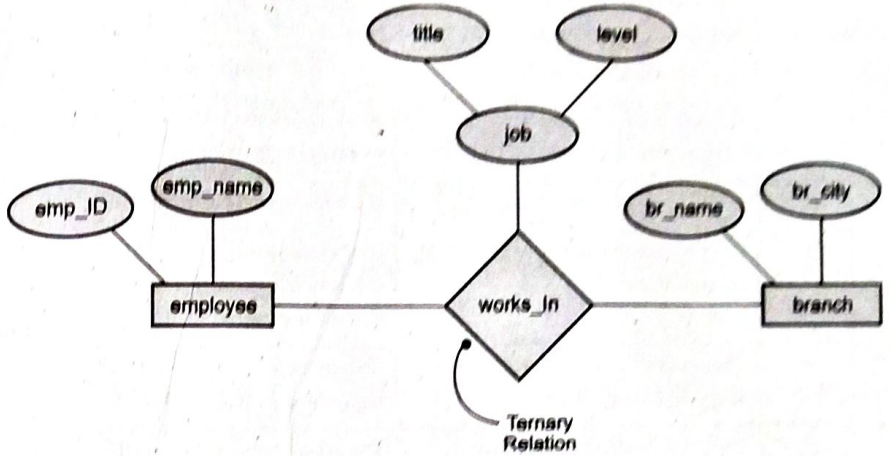


Alternate representation can be



2.2.2 Ternary Relationship

The relationship in which three entities are involved is called ternary relationship. For example -



2.2.3 Weak Entity Set

- A weak entity is an entity that cannot be uniquely identified by its attributes alone. The entity set which does not have sufficient attributes to form a primary key is called as weak entity set.

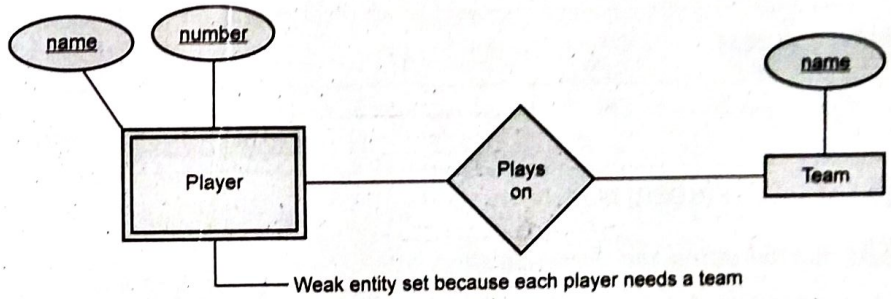


Fig. 2.2.1 : Weak entity set

- **Strong Entity Set**

  The entity set that has primary key is called as strong entity set

Weak entity rules

- A weak entity set has **one or more many-one** relationships to other (supporting) entity sets.
- The key for a weak entity set is its own underlined attributes and the keys for the supporting entity sets. For example - player-number and team-name is a key for Players.

Difference between Strong and Weak Entity Set

Sr. No.	Strong entity set	Weak entity set
1	It has its own primary key.	It does not have sufficient attribute to form a primary key on its own.
2.	It is represented by rectangle	It is represented by double rectangle.
3.	It represents the primary key which is underlined.	It represents the partial key or discriminator which is represented by dashed underline.
4.	The member of strong entity set is called as dominant entity set	The member of weak entity set is called subordinate entity set.
5.	The relationship between two strong entity sets is represented by diamond symbol.	The relationship between strong entity set and weak entity set is represented by double diamond symbol.
6.	The primary key is one of the attributes which uniquely identifies its member.	The primary key of weak entity set is a combination of partial key and primary key of the strong entity set.

Review Question

1. *Differentiate between – Strong entity set and Weak entity set*

SPPU : Oct.-19, (In Sem), Marks 2

2.3 Enhanced ER(EER) Model

2.3.1 Specialization and Generalization

- Some entities have relationships that form hierarchies. For instance, Employee can be an hourly employee or contracted employee.
- In this relationship hierarchies, some entities can act as superclass and some other entities can act as subclass.
- **Superclass :** An entity type that represents a general concept at a high level, is called superclass.
- **Subclass :** An entity type that represents a specific concept at lower levels, is called subclass.
- The subclass is said to inherit from superclass. When a subclass inherits from one or more superclasses, it inherits all their attributes. In addition to the inherited attributes, a subclass can also define its own specific attributes.

- The process of making subclasses from a general concept is called **specialization.** This is **top-down** process. In this process, the sub-groups are identified within an entity set which have attributes that are not shared by all entities.
- The process of making superclass from subclasses is called **generalization.** This is a **bottom up** process. In this process multiple sets are synthesized into high level entities.
- The symbol used for specialization/ Generalization is

- **For example –** There can be two subclass entities namely **Hourly_Emps** and **Contract_Emps** which are subclasses of **Empoyee** class. We might have attributes **hours_worked** and **hourly_wage** defined for **Hourly_Emps** and an attribute **contractid** defined for **ContractEmps.**

  Therefore, the attributes defined for an **Hourly_Emps** entity are the attributes for **Employees** plus **Hourly_Emps.** We say that the attributes for the entity set **Employees** are inherited by the entity set **Hourly_Emps** and that **Hourly-Emps ISA** (read is a) Employees. It can be represented by following Fig. 2.3.1.

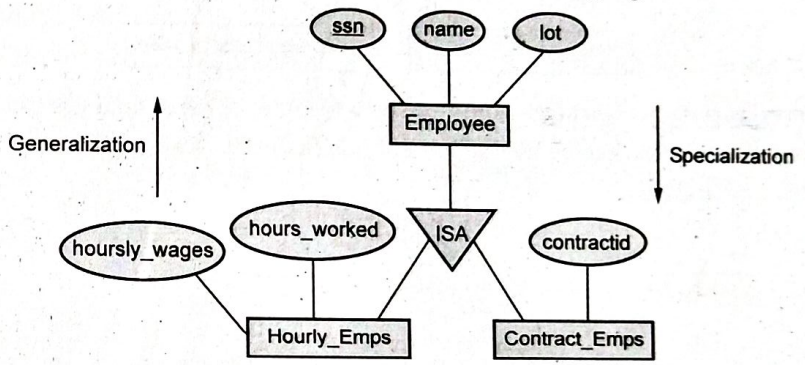


Fig. 2.3.1 Example of Generalization and Specialization

2.3.2 Constraints on Specialization/Generalization

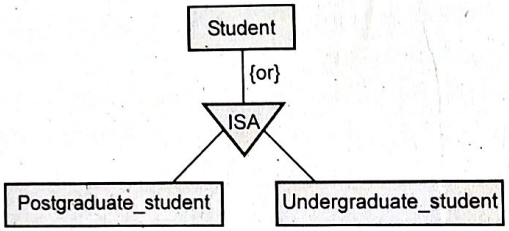
There are four types of constraints on specialization/generalization relationship. These are -

1) **Membership constraints :** This is a kind of constraints that involves determining which entities can be members of a given lower-level entity. There are two types of membership constraints -

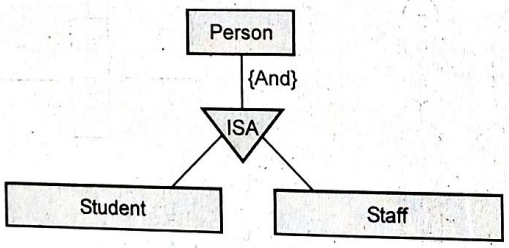
i) **Condition defined :** In condition-defined lower-level entity sets,membership is evaluated on the basis of whether or not an entity satisfies an explicit condition or predicate. For example - Consider the high-level entity Set **Employee** that has attribute **Employee_type**. All Employee entities are evaluated on defining **Employee_type** attribute. All entities that satisfy the condition student type = "ContractEmployee" are included in Contracted Employee. Since all the lower-level entities are evaluated on the basis of the same attribute this type of generalization is said to be **attribute-defined**.

ii) **User defined :** This is kind of entity set that in which the membership is manually defined.

2) **Disjoint constraints :** The disjoint constraint only applies when a superclass has more than one subclass. If the subclasses are disjoint, then an entity occurrence can be a member of only one of the subclasses. For entity Student has either **Postgraduate_Student** entity or **Undergraduate_Student**

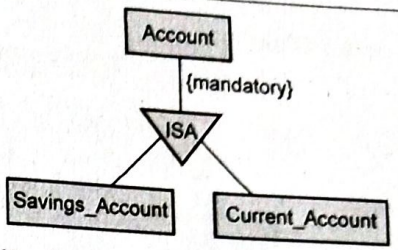


3) **Overlapping :** When some entity can be a member of more than one subclasses. For example - Person can be both a Student or a Staff. The **And** can be used to represent this constraint.

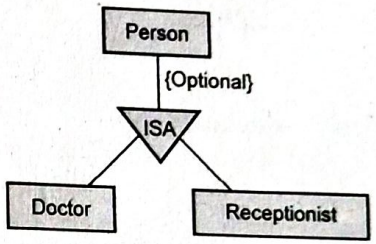


4) **Completeness :** It specifies whether or not an entity in the higher-level entity set must belong to at least one of the lower-level entity sets within the generalization/specialization. This constraint may be one of the following -

i) **Total generalization or specialization :** Each higher-level entity must belong to a lower-level entity set. For example - Account in the bank must either Savings account or Current Account. The **mandatory** can be used to represent this constraint.



ii) **Partial generalization or specialization :** Some higher-level entities may not belong to any lower-level entity set.



2.3.3 Aggregation

A feature of the entity relationship model that allows a relationship set to participate in another relationship set. This is indicated on an ER diagram by drawing a dashed box around the aggregation.

For example - We treat the relationship set work and the entity sets employee and project as a higher-level entity set called work.

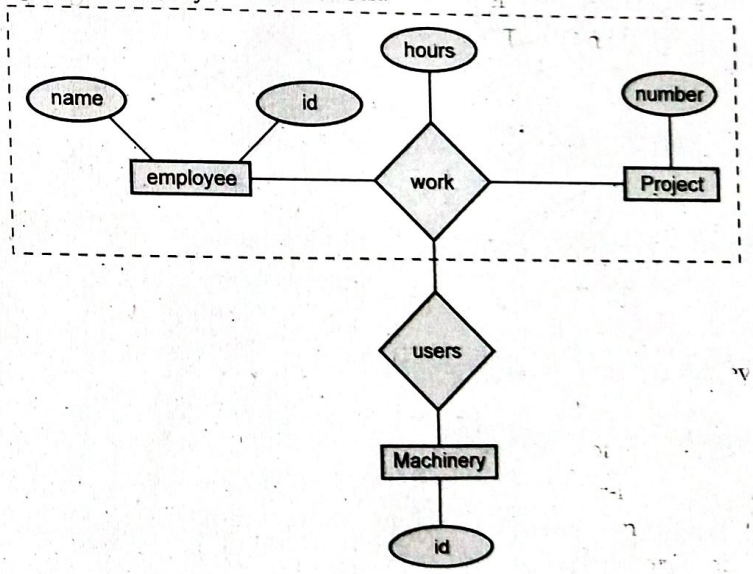


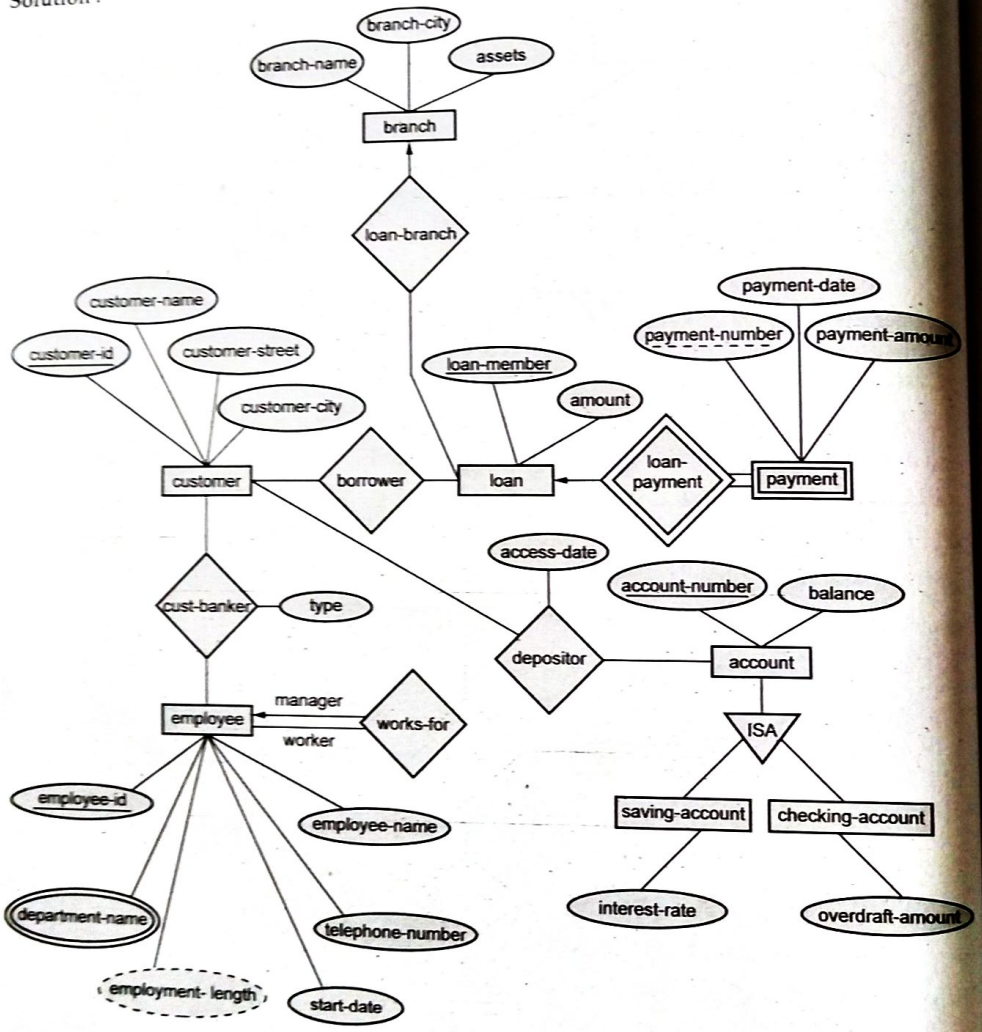
Fig. 2.3.2 : ER model with aggregation

2.4 Examples Based on ER Diagram

SPPU : Aug 17, Nov.-17, Oct.-19, Marks 6

Example 2.4.1 *Draw an ER diagram for banking system(Home-Loan Application)*

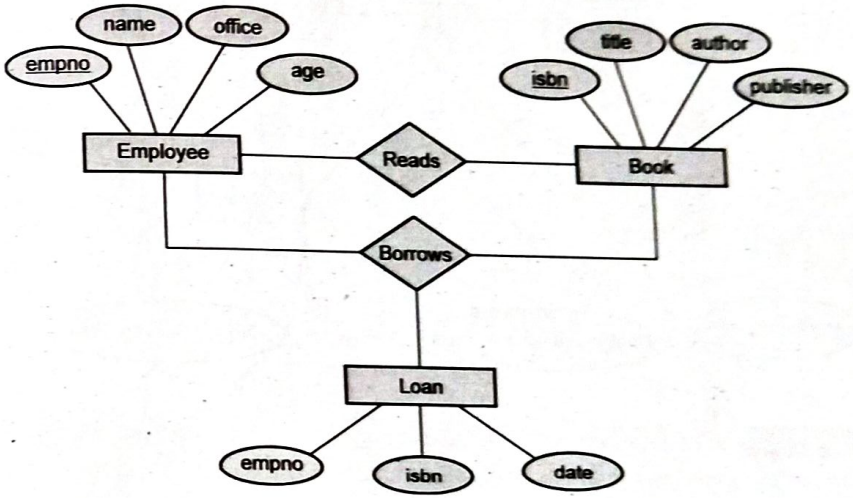
Solution :



Example 2.4.2 *Consider the relation schema given in following Figure. Design and draw an ER diagram that capture the information of this schema.*

Employee(empno,name,office,age)
Books(isbn,title,authors,publisher)
Loan(empno,isbn,date)

Solution :



Example 2.4.3 *A car rental company maintains a database for all vehicles in its current fleet. For all vehicles, it includes the vehicle identification number license number, manufacturer, model, date of purchase and color. Special data are included for certain types of vehicles.*

Trucks : Cargo capacity

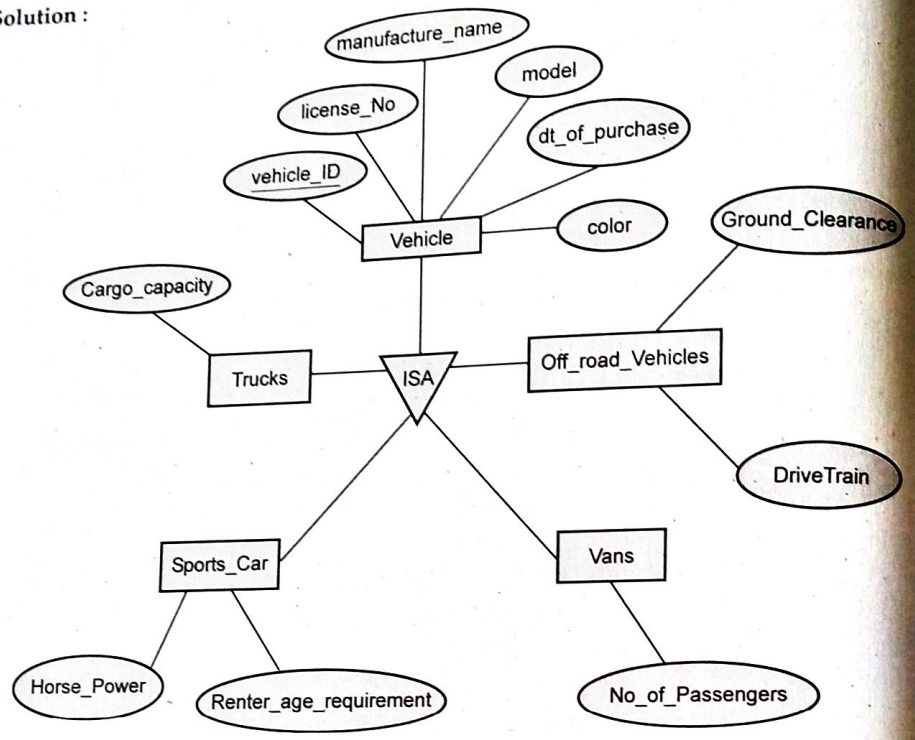
Sports cars : horsepower, renter age requirement

Vans : number of passengers

Off-road vehicles : ground clearance, drivetrain (four-or two-wheel drive)

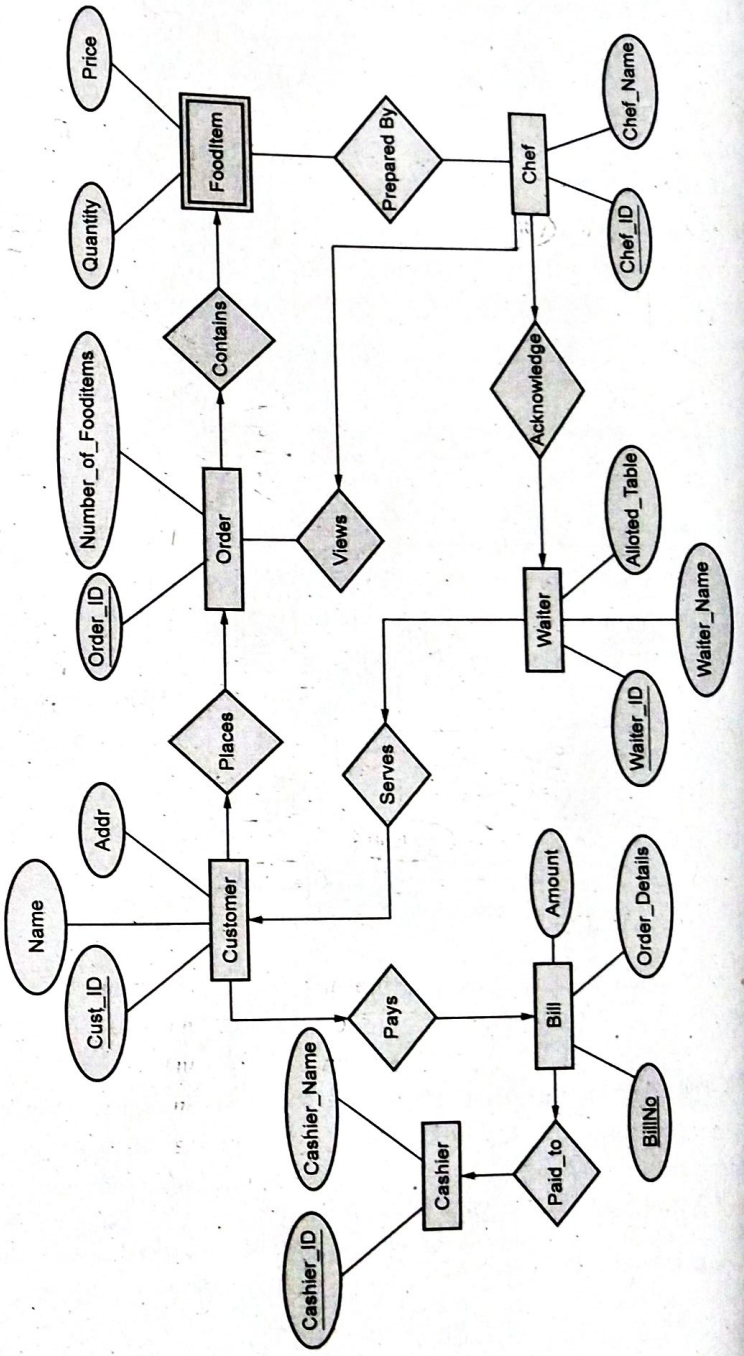
Construct an ER model for the car rental company database.

Solution :



Example 2.4.4 *Draw E-R diagram for the "Restaurant Menu Ordering System", which will facilitate the food items ordering and services within a restaurant. The entire restaurant scenario is detailed as follows. The customer is able to view the food items menu, call the waiter, place orders and obtain the final bill through the computer kept in their table. The Waiters through their wireless tablet PC are able to initialize a table for customers, control the table functions to assist customers, orders, send orders to food preparation staff (chef) and finalize the customer's bill. The Food preparation staffs (chefs), with their touch-display interfaces to the system, are able to view orders sent to the kitchen by waiters. During preparation they are able to let the waiter know the status of each item, and can send notifications when items are completed. The system should have full accountability and logging facilities, and should support supervisor actions to account for exceptional circumstances, such as a meal being refunded or walked out on.*

Solution :



Example 2.4.5 *A university registrar's office maintains data about the following entities :*

(1) courses, including number, title, credits, syllabus, and prerequisites;

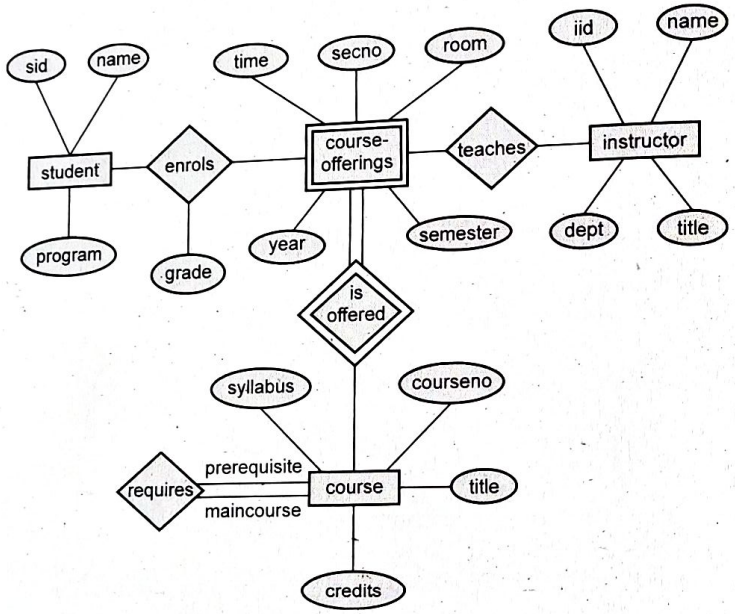
(2) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom;

(3) students, including student-id, name, and program; and

(4) instructors, including identification number, name, department, and title.

Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled. Construct an E-R diagram for the registrar's office. Document all assumptions that you make about the mapping constraints.

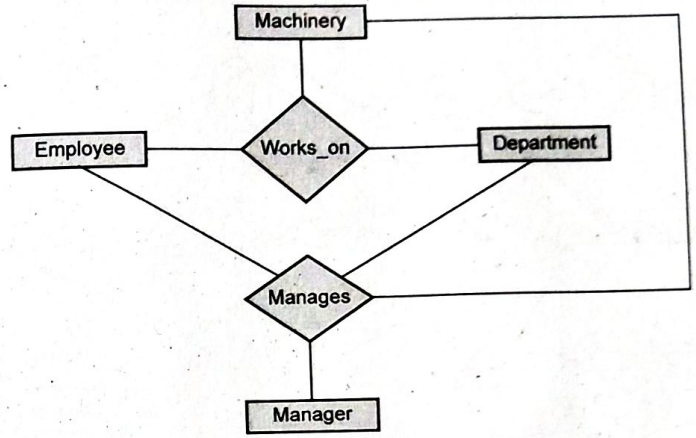
Solution :



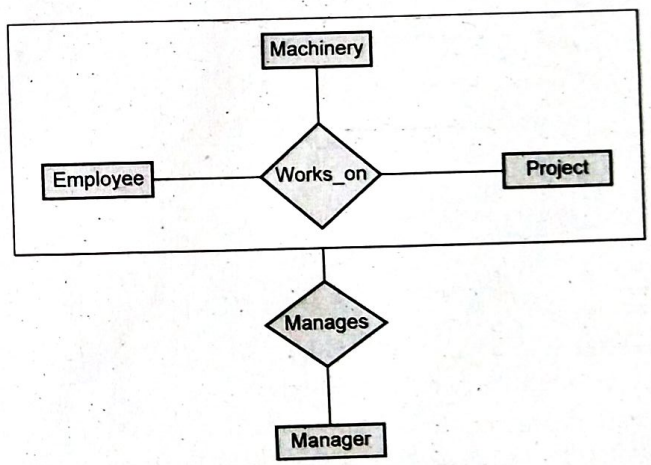
Example 2.4.6 *What is aggregation in ER model ? Develop an ER diagram using aggregation that captures following information : Employees work for projects. An employee working for particular project uses various machinery. Assume necessary attributes. State any assumptions you make. Also discuss about the ER diagram you have designed.*

Solution : Aggregation : Refer section 2.3.3.

ER Diagram : The ER diagram for above described scenario can be drawn as follows -



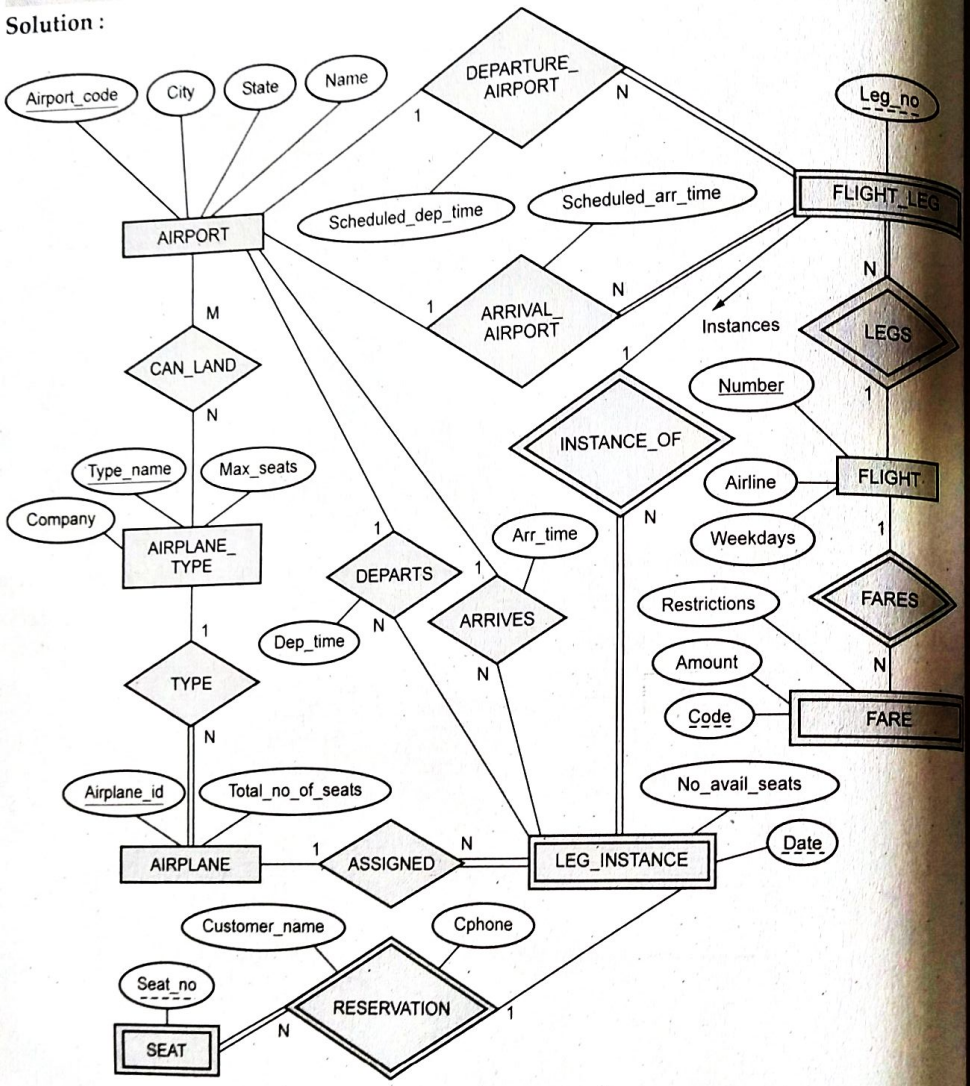
The above ER model contains the redundant information, because every Employee, Project, Machinery combination in **works_on** relationship is also considered in **manages** relationship. To avoid this redundancy problem we can make use of aggregation relationship in ER diagram as follows -



We can then create a binary relationship **manages** for between **Manager** and **(Employee, Project, Machinery).**

Example 2.4.7 *Draw an ER diagram of Airline reservation system taking into account at least five entities. Indicate all keys, constraints and assumptions that are made*

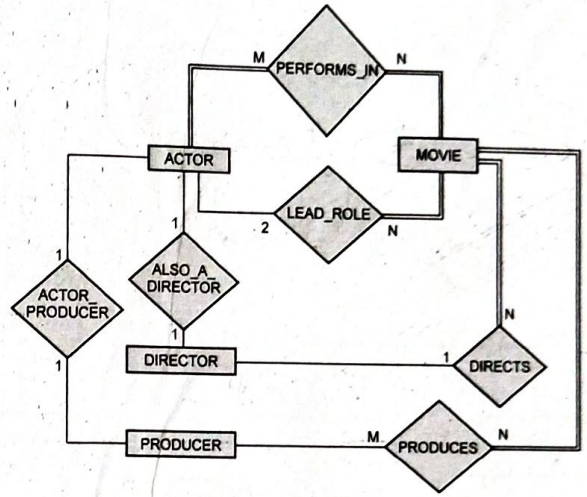
Solution :



A LEG is nonstop portion of flight. The LEG_INSTANCE is a particular occurrence of a LEG on particular date.

Example 2.4.8 *Draw an ER diagram of movie database. Assume your own entities (minimum 4) attributes and relationships.*

Solution :



Example 2.4.9 *Draw an ER diagram to represent the Election Information system based on the following description.*

In Indian national election, a state is divided into a number of constituencies depending upon the population of the state. Several candidates contest elections in each constituency. Record the number of votes obtained by each candidate. The system also maintains the voter list and a voter normally belongs to a particular constituency.

Note that the party details must also be taken care in the design.

Solution :

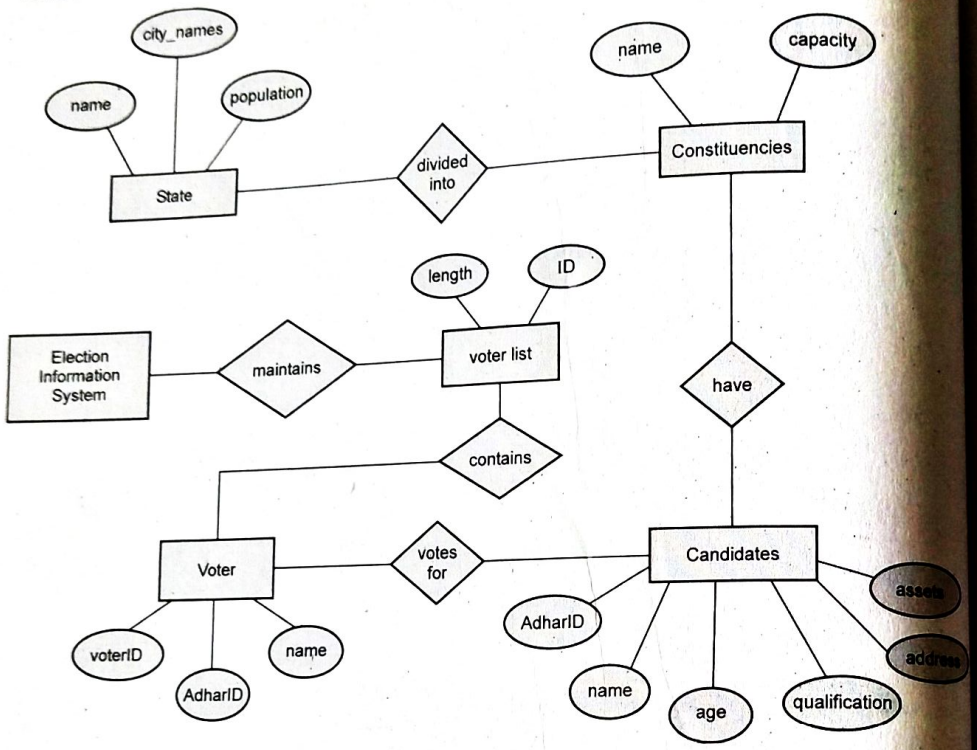
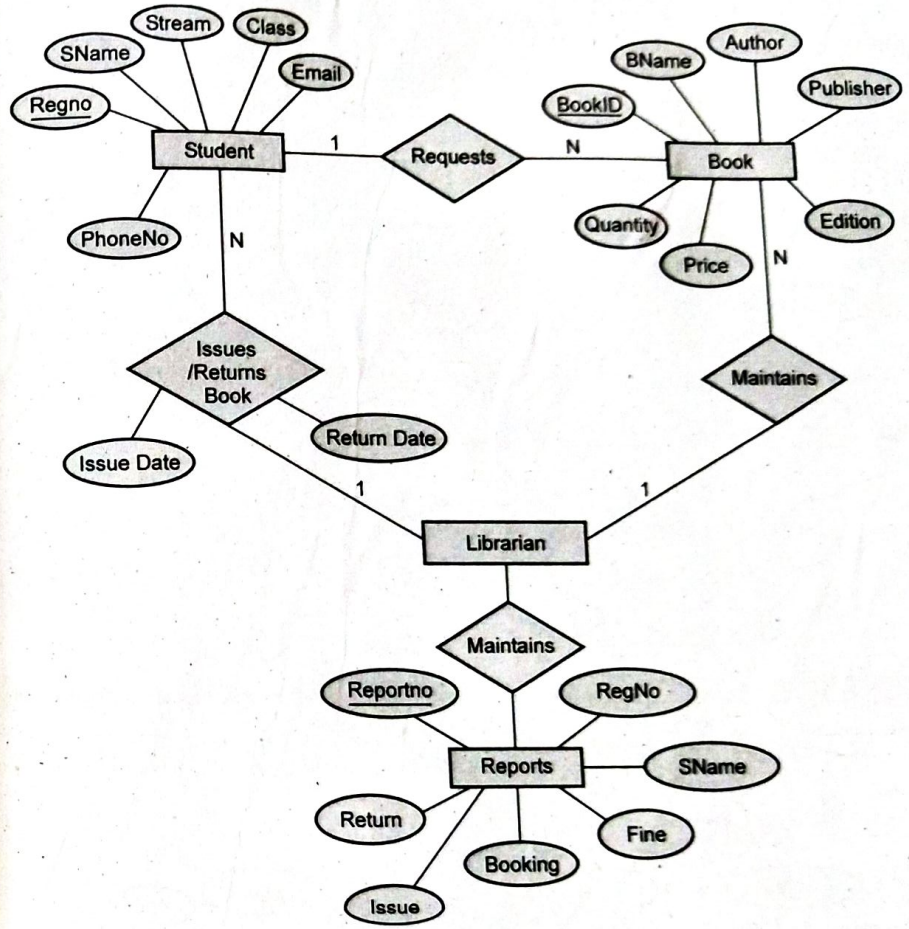


Fig. 2. 4.1 Election information system

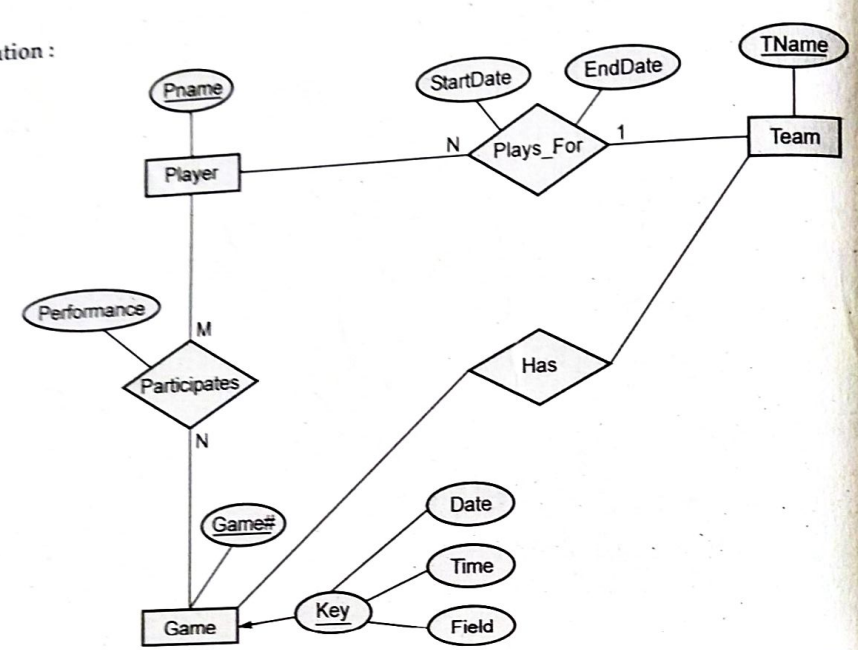
Example 2.4.10 *Construct an E R diagram for library management system*

Solution :



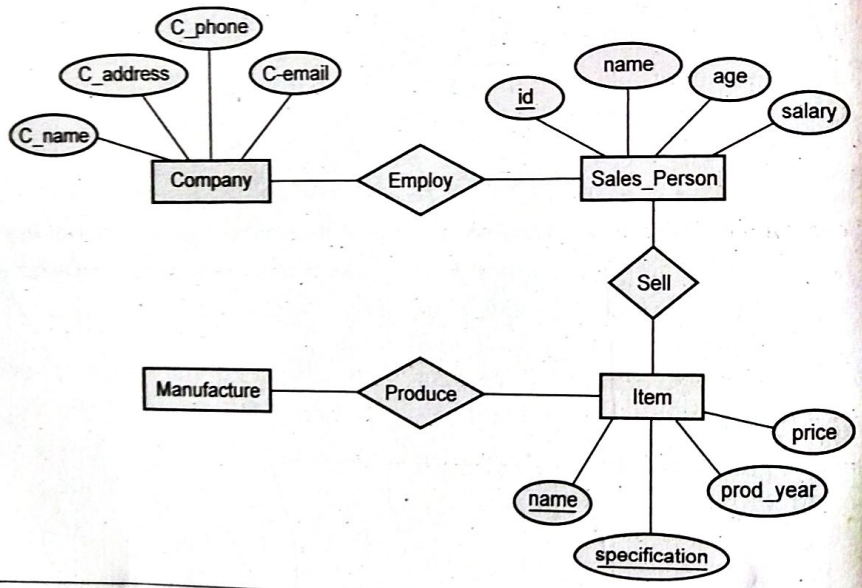
Example 2.4.11 *A database is to be constructed to keep track of the teams and games of a sport league. A team has number of players not all of whom participate in each game. It is desired to keep track of the players participating in each game for each team, the positions they played in that game, and the result of the game. Design an ER diagram completely with attributes, keys and constraints for the above description.*

Solution :



Example 2.4.12 *Draw an ER diagram for a small marketing company database. Assume suitable data.*

Solution :



Example 2.4.13 *A university database contains information about professors (identified by social security number, or SSN) and courses (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram -*

1. Professors can teach the same course in several semesters, and each ordering must be recorded.

2. Professors can teach the same course in several semesters, and only the most recent such ordering needs to be recorded (Assume this condition applies in all subsequent questions).

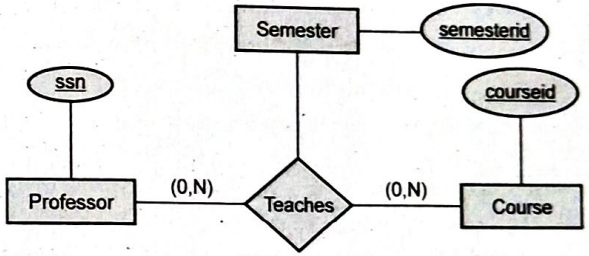
3. Every professor must teach some course.

4. Every professor teaches exactly one course.

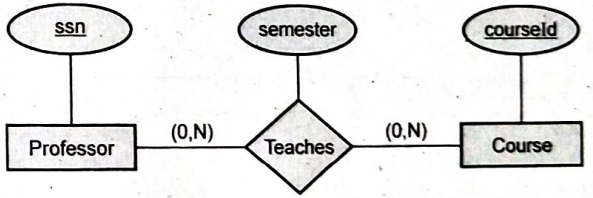
5. Every professor teaches exactly one course and every course must be taught by some professor.

Solution :

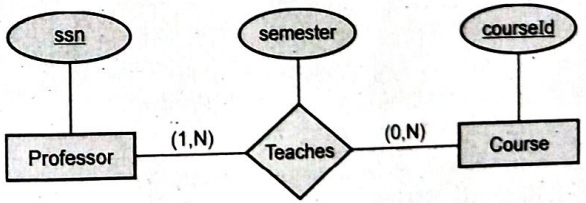
1.



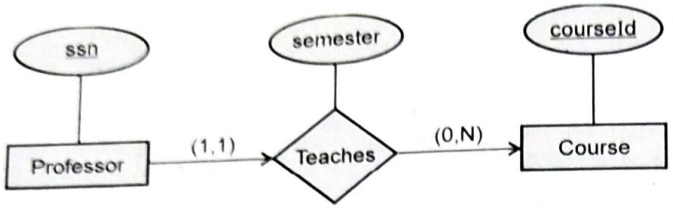
2.



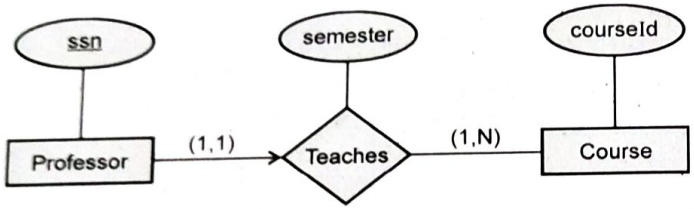
3.



4.

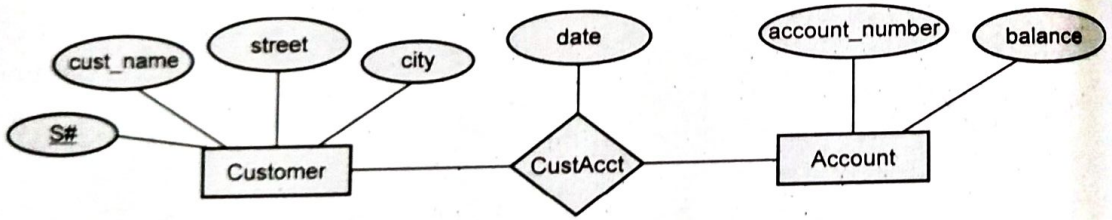


5.



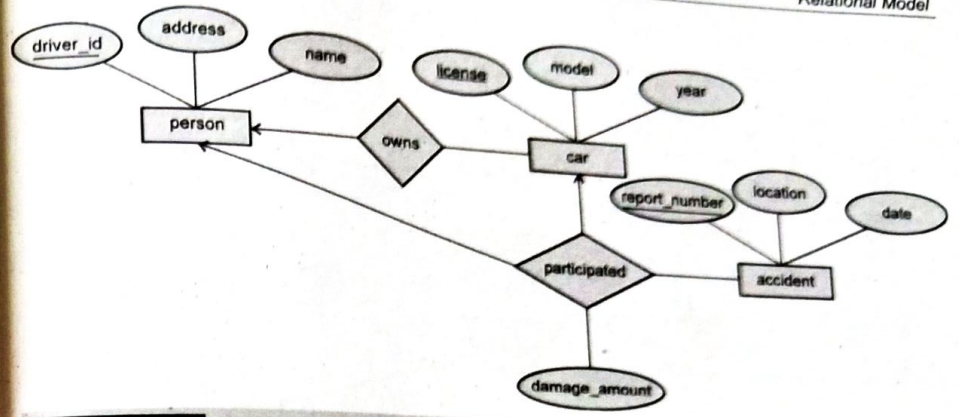
Example 2.4.13 *Construct an ER diagram of customer account relationship. Customer entity with attributes of S#, Customer name, street, customer city, and account entity with attributes account no, and balance. The customer account relationship with date attributes.*

Solution :



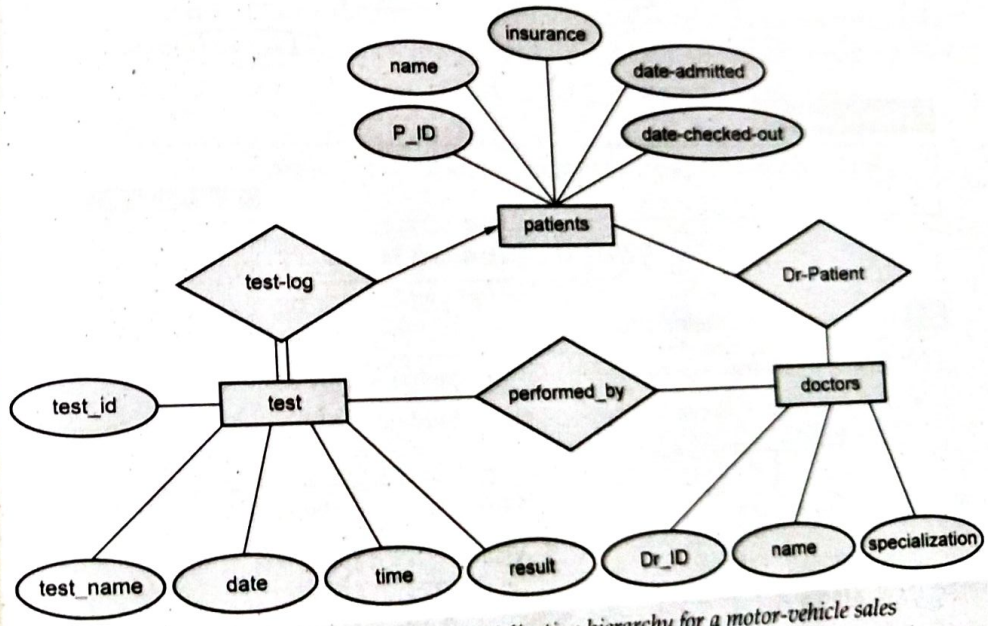
Example 2.4.14 *Construct an E-R diagram for a car insurance company whose customers own one or more cars each.Each car has associated with it zero to any number of recorded accidents. Each insurance policy covers one or more cars and has one or more premium payments associated with it. Each payment is for particular period of time and has an associated due date and date when the payment was received.* **SPPU : Aug 17, Marks 4**

Solution :



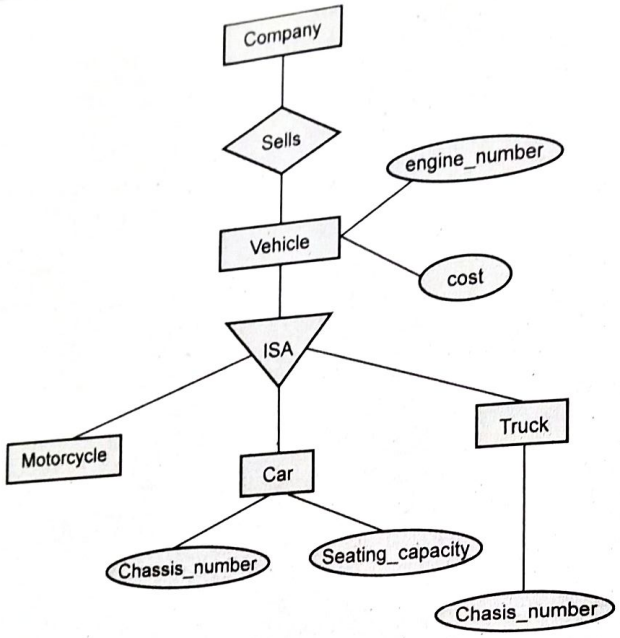
Example 2.4.15 *List E-R diagram symbol and draw an E-R diagram for a hospital management system with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examination conducted.* **SPPU : Nov -17, (End Sem), Marks 6**

Solution :



Example 2.4.16 *Design generalization-specialization hierarchy for a motor-vehicle sales company. The company sell motorcycles which have an engine number and cost. Cars which have chassis number, an engine number, seating capacity and cost; trucks which have chassis number an engine number and cost.* **SPPU : Oct.-19, (In Sem), Marks 6**

Solution :



Review Question

1. *Explain specialization, generalization and aggregation with example.* **SPPU : Aug 17, Marks 5**

Part II : Relational Model

2.5 Introduction to Relational Model

- Relation database is a **collection of tables** having unique names.
- For example – Consider the example of **Student table** in which the information about the student is stored.

RollNo	Name	Phone
001	AAA	1111111111
002	BBB	2222222222
003	CCC	3333333333

Fig. 2.5.1 Student table

The above table consists of three column headers **RollNo, Name** and **Phone**. Each row of the table indicates the information of each student by means of his Roll Number, Name and Phone number.

Similarly consider another table named **Course** as follows –

CourseID	CourseName	Credits
101	Mechanical	4
102	Computer Science	6
103	Electrical	5
104	Civil	3

Fig. 2.5.2 Course table

Clearly, in above table the columns are **CourseID, CourseName** and **Credits.** The **CourseID 101** is associated with the course named **Mechanical** and associated with the course of mechanical there are **4 credit points.** Thus the relation is represented by the table in the relation model. Similarly we can establish the relationship among the two tables by defining the third table. For example – Consider the table **Admission** as

RollNo	CourseID
001	102
002	104
003	101

Fig. 2.5.3 Admission Table

From this third table we can easily find out that the course to which the **RollNo 001** is admitted is **computer Science**.

2.6 Basic Terminologies

SPPU : Aug.-17, Marks 5

There are some commonly used terms in Relational Model and those are -

Table or relation : In relational model, table is a collection of data items arranged in rows and columns. The table cannot have duplicate data or rows. Below is an example of student table

Roll No	Name	Marks	Phone
001	AAA	88	1111111111
002	BBB	83	2222222222
003	CCC	98	3333333333
004	DDD	67	4444444444

Tuple or record or row : The single entry in the table is called tuple. The tuple represents a set of related data. In above **Student** table there are four tuples. One of the tuple can be represented as

001	AAA	88	1111111111

Attribute or columns : It is a part of table that contains several records. Each record can be broken down into several small parts of data known as attributes. For example the above table consists of four attributes such as **RollNo,Name,Marks** and **Phone**.

Relation schema : A relation schema describes the structure of the relation, with the name of the relation (i.e. name of table), its attributes and their names and type.

Relation Instance : It refers to specific instance of relation i.e. containing a specific set of rows. For example – the following is a relation instance – which contains the records with marks above 80.

RollNo	Name	Marks	Phone
001	AAA	88	1111111111
002	BBB	83	2222222222
003	CCC	98	3333333333

Domain : For each attribute of relation, there is a set of permitted values called domain. For example – in above table, the domain of attribute **Marks** is set of all possible permitted marks of the students. Similarly the domain of **Name** attribute is all possible names of students.

That means Domain of Marks attribute is (88,83,98)

Atomic : The domain is atomic if elements of the domain are considered to be indivisible units. For example in above **Student** table, the attribute **Phone** is non-atomic.

NULL attribute : A null is a special symbol, independent of data type, which means either unknown or inapplicable. It does not mean zero or blank. For example - Consider a salary table that contains NULL

Emp#	Job Name	Salary	Commission
E10	Sales	12500	32090
E11	Null	25000	8000
E12	Sales	44000	0
E13	Sales	44000	Null

Degree : It is nothing but total number of columns present in the relational database. In given Student table –

Roll No	Name	Marks	Phone
001	AAA	88	1111111111
002	BBB	83	2222222222
003	CCC	98	3333333333

The degree is 4.

Cardinality : It is total number of tuples present in the relational database. In above given table the cardinality is 3

Example 2.6.1 *Find out following for given Staff table*

i) No of Columns *ii) No of tuples*

iii) Different attributes *iv) Degree* *v) Cardinality*

StaffID	Name	Sex	Designation	Salary	DOJ
S001	John	M	Manager	50000	1 Oct. 2012
S002	Ram	M	Executive	20000	20 Jan. 2015
S003	Meena	F	Supervisor	40000	12 Aug. 2011

Solution :

i) No of Columns = 6

ii) No of Tuples= 3

iii) Different attributes are StaffID, Name,Sex, Designation, Salary, DOJ

iv) Degree= Total number of columns=6

v) Cardinality =Total number of rows = 3

Review Question

1. *What is schema definition? Explain the types of attributes with an example.* **SPPU : Aug.-17, Marks 5**

SPPU : May-19, Nov.-18, Marks 6

2.7 Codd's rules

Codd proposed 13 rules for relational database management system, which are popularly known as **Codd's 12 rule** : These rules are as follows –

Rule 0 : This rule states for a database to be relational, it must use its relational capabilities to manage the database.

Rule 1 : The Information rule - All information in an RDBMS is represented logically only by storing the values in tables.

Rule 2 : The Guaranteed Access rule - Each item of data in an RDBMS is guaranteed to be logically accessible by specifying the table name, primary key value, and column name.

Rule 3 : The Systematic Treatment of Null Values rule - Null values are supported in a fully relational DBMS for representing missing information or inapplicable information in a systematic way which is independent of the data type.

Rule 4 : The Dynamic Online Catalog Based on the Relational Model rule - Database dictionary which is called as catalog-is the structure description of the complete Database and it must be stored online. This Catalog must be governed by same rules as rest of the database. The same query language should be used on catalog as used to query database.

Rule 5 : The Comprehensive Data Sublanguage rule - At least one well structured, well-defined language must be there which can access all the data present in the database.

Rule 6 : The View Updating rule - All views of the data which are theoretically updatable must be updatable in practice by the DBMS.

Rule 7 : Relational level operation - The High-level Insert, Update, and Delete rule: There must be insert, delete and update operations at each level of relations.

Rule 8 : The Physical Data Independence rule - Physical storage should not matter the system. Whenever any changes are made in either storage representations or access methods then it should not affect the application.

Rule 9 : The Logical Data Independence rule - If any changes are made in table structure then the logical view of the user should not get affected. Fpr Rule example - if a table is split into two tables internally, the view of the table to the user should be an entire table and not the split tables.

Rule 10 : The Integrity Independence rule - The Integrity constraints must be defined by the RDBMS stored in the system and it should not be enforced by the external application programs.

Rule 11 : The Distribution Independence rule - An RDBMS must have distribution independence. That means, even if database is scattered geographically, user should get a feel as if it is stored in one piece at one location.

Rule 12 : The Non-sub-version rule - If low-level language is allowed to access the system, then that low-level language must not be able to subvert or bypass the integrity rules which are expressed in a higher-level language.

Review Question

1. *Specify Codd's Norms to be satisfied by RDBMS?*

SPPU : May-19, (End Sem), Marks 5; Nov.-18, (End Sem), Marks 6

Part III : Relational Integrity

2.8 Keys

SPPU : Aug.-17, Marks 4

Keys are used to specify the tuples distinctly in the given relation.

Various types of keys used in relational model are – Superkey, Candidate Keys, primary keys, foreign keys. Let us discuss them with suitable example

1) Super Key(SK):

It is a set of one or more attributes within a table that can uniquely identify each record within a table. For example – Consider the Student table as follows –

Reg No.	Roll No	Phone	Name	Marks
R101	001	1111111111	AAA	88
R102	002	2222222222	BBB	83
R103	003	3333333333	CCC	98
R104	004	4444444444	DDD	67

Fig. 2.8.1 Student

The superkey can be represented as follows

(RollNo, Phone, Name) Superkey

(Name, Marks) Not a Superkey

Superkey

RegNo.	RollNo	Phone	Name	Marks
R101	001	111111111	AAA	88
R102	002	2222222222	BBB	83
R103	003	3333333333	CCC	98
R104	004	4444444444	DDD	67

Clearly using the (RegNo) and (RollNo,Phone,Name) we can identify the records uniquely but (Name, Marks) of two students can be same, hence this combination not necessarily help in identifying the record uniquely.

2) Candidate Key(CK) :

The candidate key is a subset of superset. In other words candidate key is a single attribute or least or minimal combination of attributes that uniquely identify each record in the table. For example - in above given **Student** table, the candidate key is RegNo, (RollNo,Phone). The candidate key can be

Candidate key

Candidate key

RegNo.	RollNo	Phone	Name	Marks
R101	001	111111111	AAA	88
R102	002	2222222222	BBB	83
R103	003	3333333333	CCC	98
R104	004	4444444444	DDD	67

Thus every candidate key is a superkey but every superkey is not a candidate key.

3) Primary Key(PK) :

The primary key is a candidate key chosen by the database designer to identify the tuple in the relation uniquely. For example – Consider the following representation of primary key in the student table

Primary key

RegNo.	RollNo	Phone	Name	Marks
R101	001	111111111	AAA	88
R102	002	2222222222	BBB	83
R103	003	3333333333	CCC	98
R104	004	4444444444	DDD	67

Other than the above mentioned primary key, various possible primary keys can be **(RollNo), (RollNo,Name), (RollNo, Phone)**

The relation among super key, candidate key and primary can be denoted by

Candidate Key = Super Key – Primary Key

Rules for Primary Key

i) The primary key may have one or more attributes.

ii) There is **only one primary key** in the relation.

iii) The value of primary key attribute can not be NULL.

iv) The value of primary key attribute does not get changed.

4) Alternate key :

The alternate key is a candidate key which is not chosen by the database designer to uniquely identify the tuples. For example –

Primary key

Alternate key

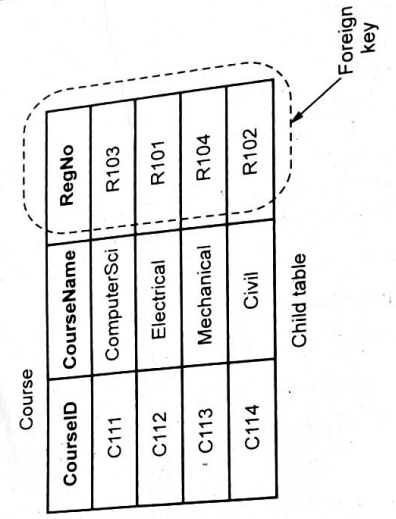
RegNo.	RollNo	Phone	Name	Marks
R101	001	111111111	AAA	88
R102	002	2222222222	BBB	83
R103	003	3333333333	CCC	98
R104	004	4444444444	DDD	67

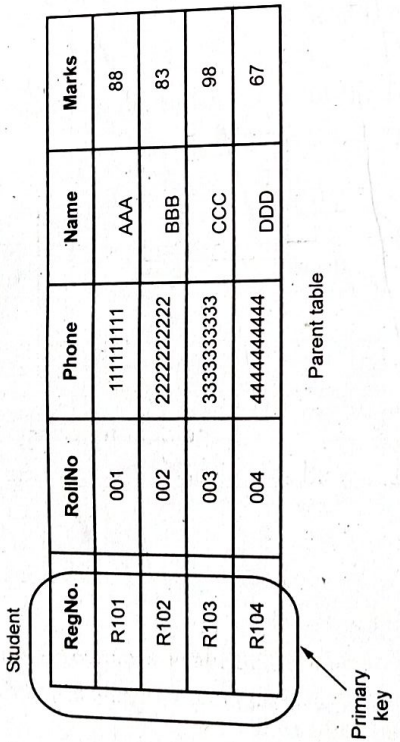
5) Foreign key :

Foreign key is a single attribute or collection of attributes in one table that refers to the primary key of other table.

- Thus foreign keys refer to primary key.
- The table containing the primary key is called **parent table** and the table containing foreign key is called **child table**.

- Example -





From above example, we can see that two tables are linked. For instance we could easily find out that the 'Student CCC has opted for ComputerSci course'

Review Question

1. *Explain the distinction among the terms primary key, candidate key and super key*
   SPPU : Aug.-17. Marks 4

2.9 Constraints

Constraints mean some rules or restrictions that are set on the database.

There are three main types of constraints.

1. Domain Constraint
2. Key Constraint or NULL Constraint
3. Integrity Constraint
   - i) Entity Integrity Constraint
   - ii) Referential Integrity Constraint

1. Domain Constraint

- Domain constraint defines the domain or set of values for an attribute.
- The data type of domain includes string, character, integer, time, date, currency, etc. The value of the attribute must be available in the corresponding domain.
- For example - Consider the Student table as follows.

Roll No	Name	Marks	Phone
001	AAA	88	111111111
001	BBB	83	2222222222
003	1234	98	3333333333
004	DDD	67	4444444444

Name cannot be numeric value.
It must be a string.
Hence this is not allowed !!!

The above relation does not satisfy the domain constraint.

2. Key Constraint or Null Constraint

- Keys are used to identify particular record from the table. Primary key is normally used to identify the record uniquely.
- Hence the key constraint can be stated as -
  - All values of primary key must be unique.
  - The value of primary key must not be NULL.
- For example - Consider the Student table as follows. For this relation, the **Roll No** is a primary key. It is expected to find the desired record using this primary key.

Roll No	Name	Marks	Phone
001	AAA	88	111111111
001	BBB	83	2222222222
003	CCC	98	3333333333
004	DDD	67	4444444444

Duplicate values ? This is not allowed !!!

The above relation does not satisfy key constraint as the primary key is not having unique value.

3. Integrity Constraint

- Integrity constraints are rules that are to be applied on database columns to ensure the validity of data.
- For example -
  i) The Employee ID and Department ID must consist of two digits.
  ii) Every Employee ID must start with letter.

i) Entity Integrity Constraint

- **This rule states that** "In the relations, the value of attribute of primary key can not be null".
- The NULL represents a value for an attribute that is currently unknown or is not applicable for this tuple. The Nulls are always **to deal with incomplete or exceptional data.**
- The primary key value helps in uniquely identifying every row in the table. Thus if the users of the database want to retrieve any row from the table or perform any action on that table, they must know the value of the key for that row. Hence it is necessary that the primary key should not have the NULL value.

- For example -

Roll No	Name	Marks	Phone
001	AAA	88	111111111
001	BBB	83	2222222222
003	CCC	98	3333333333
	DDD	67	4444444444

NULL is not allowed !!!

ii) Referential Integrity Constraint

- In relationships, data is linked between two or more tables.
- This is achieved by having the foreign key (in the associated table) reference a primary key value (in the primary - or parent - table). Because of this, we need to ensure that data on both sides of the relationship remain intact.
- **The referential integrity rule states that** "whenever **a foreign key** value is used it must **reference a valid, existing primary** key in the parent table".
- For example - Consider two tables

Primary key

RegNo	Roll No	Name	Marks	Phone
R101	001	AAA	88	111111111
R102	001	BBB	83	2222222222
R103	003	CCC	98	3333333333
R104	004	DDD	67	4444444444

Student table as parent table

Relation

Foreign key

CourseID	CourseName	RegNo
c111	ComputerSci	R103
c112	Electrical	R101
c113	Mechanical	R555
c114	Civil	R102

This value is not present in parent table

Course table as child table

In above relation, the registration no. R555 is not existing still if it is present in the course table, then we say that it is not following referential integrity constraint.

2.10 Enterprise Constraints

Enterprise constraints are also called as semantic constraints.

The enterprise constraints are basically the additional rules specified by users or database administrators. These constraints are normally based on multiple tables.

Examples of enterprise constraints are –

1) The salary of teacher should not exceed the salary of Principal.
2) A Student can not opt for more than two courses at a time.
3) A class can have maximum 50 students.

2.11 Views

- The view of the system help the user to retrieve data efficiently.
- A **view** is a virtual or logical table that allows to view or manipulate parts of the tables.
- A view also has rows and columns as they are in a real table in the database.
- We can create a view by selecting fields from one or more tables present in the database.
- A view can either have all the rows of a table or specific rows based on certain condition.
- For example

Consider table **Empolyee** and create a view **EmployeeDetails** whose Salary is <10000

Employee Table		
EmpID	**EName**	**Salary**
101	Archana	20000
102	Madhura	5000
103	Poonam	8000
104	Shruti	15000
105	Monika	7000

→

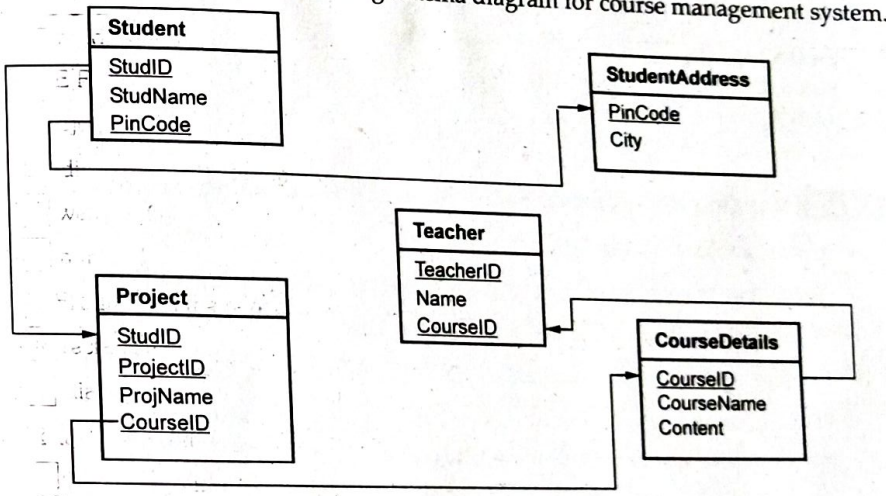
EmployeeDetails View	
EmpID	**Ename**
102	Madhura
103	Poonam
105	Monika

Role of Views in Database:

1) Views are valuable in the **context of security.**
2) We can define view that give a group of uses **access to only allowed part** of the database and **can restrict other part.**

2.12 Schema Diagrams

- Schema diagrams are used to represent the relationship among the various schemas.
- Each relation appears as a box, with the relation name at the top, and the attributes listed inside the box.
- Primary key attributes are shown underlined.
- Foreign key dependencies appear as arrows from the foreign key attributes of the referencing relation to the primary key of the referenced relation.
- For example – Consider following schema diagram for course management system.



2.13 Converting ER diagrams into tables

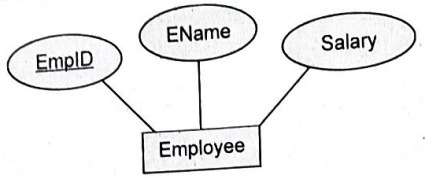
SPPU : Nov.-18, Marks 4

In this section we will discuss how to map various ER model constructs to Relational Model construct.

2.13.1 Mapping of Entity Set to Relationship

- An entity set is mapped to a relation in a straightforward way.
- Each attribute of entity set becomes an attribute of the table.
- The primary key attribute of entity set becomes an entity of the table.

- For example - Consider following ER diagram.



The converted employee table is as follows -

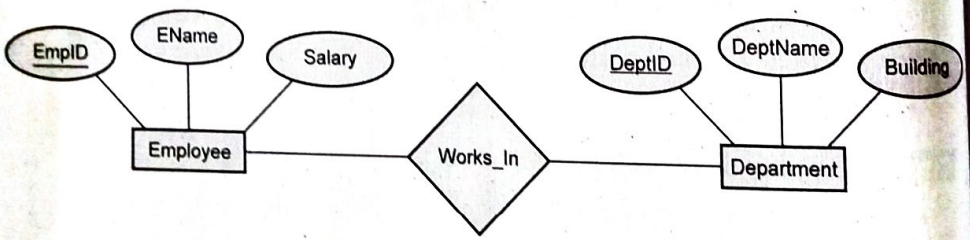
EmpID	EName	Salary
201	Poonam	30000
202	Ashwini	35000
203	Sharda	40000

The SQL statement captures the information for above ER diageam as follows -

```
CREATE TABLE Employee( EmpID CHAR(11),
    EName CHAR(30),
    Salary INTEGER,
PRIMARY KEY(EmpID))
```

2.13.2 Mapping Relationship Sets(without Constraints) to Tables

- Create a table for the relationship set.
- Add all primary keys of the participating entity sets as fields of the table.
- Add a field for each attribute of the relationship.
- Declare a primary key using all key fields from the entity sets.
- Declare foreign key constraints for all these fields from the entity sets.
  For example - Consider following ER model

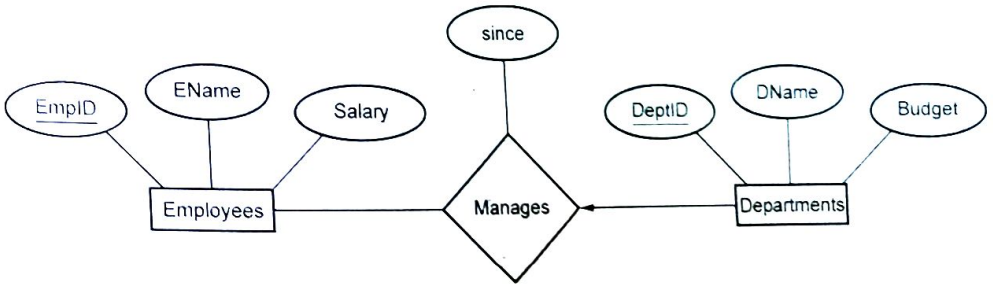


The SQL statement captures the information for relationship present in above ER diagram as follows -

```
CREATE TABLE Works_In  (EmpID CHAR(11),
                        DeptID CHAR(11),
                        EName CHAR(30),
                        Salary INTEGER,
                        DeptName CHAR(20),
                        Building CHAR(10),
                        PRIMARY KEY(EmpID,DeptID),
                        FOREIGN KEY (EmpID) REFERENCES Employee,
                        FOREIGN KEY (DeptID) REFERENCES Department
                        )
```

2.13.3 Mapping Relationship Sets(With Constraints) to Tables

- If a relationship set involves **n** entity sets and some **m** of them are linked via arrows in the ER diagram, the key for anyone of these **m** entity sets constitutes a key for the relation to which the relationship set is mapped.
- Hence we have **m** candidate keys, and one of these should be designated as the **primary key.**
- There are two approaches used to convert a relationship sets with key constraints into table.
- Approach 1 :
  - By this approach the relationship associated with more than one entities is separately represented using a table. For example - Consider following ER diagram. Each **Dept has at most one manager, according to the key constraint on Manages.**



Here the constraint is each department has at the most one manager to manage it. Hence no two tuples can have same DeptID. Hence there can be a separate table named **Manages** with DeptID as Primary Key. The table can be defined using following SQL statement

```
CREATE TABLE Manages(EmpID CHAR(11),
                    DeptID INTEGER,
                    Since DATE,
                    PRIMARY KEY(DeptID),
                    FOREIGN KEY (EmpID) REFERENCES Employees,
                    FOREIGN KEY (DeptID) REFERENCES Departments)
```

- Approach 2 :
  - In this approach , it is preferred **to translate a relationship set with key constraints**.
  - It is a superior approach because, it avoids creating a distinct table for the relationship set.
  - The idea is to include the information about the relationship set in the table corresponding to the entity set with the key, taking advantage of the key constraint.
  - This approach eliminates the need for a separate Manages relation, and queries asking for a department's manager can be answered without combining information from two relations.
  - The only drawback to this approach is that space could be wasted if several departments have no managers.
  - The following SQL statement, defining a **Dep_Mgr** relation that captures the information in both **Departments** and **Manages**, illustrates the second approach to translating relationship sets with key constraints :

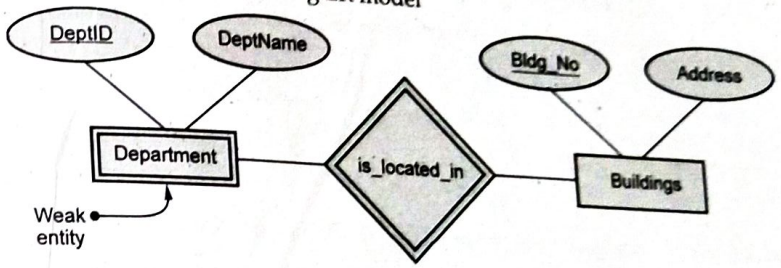
```
CREATE TABLE Dep_Mgr ( DeptID INTEGER,
                       DName CHAR(20),
                       Budget REAL,
                       EmpID CHAR (11),
                       since DATE,
                       PRIMARY KEY (DeptID),
                       FOREIGN KEY (EmpID) REFERENCES Employees)
```


2.13.4 Mapping Weak Entity Sets to Relational Mapping

A weak entity can be identified uniquely only by considering the primary key of another (owner) entity. Following steps are used for mapping Weka Entity Set to Relational Mapping

- Create a table for the weak entity set.
- Make each attribute of the weak entity set a field of the table.
- Add fields for the primary key attributes of the identifying owner.
- Declare a foreign key constraint on these identifying owner fields.
- Instruct the system to automatically delete any tuples in the table for which there are no owners

For example - Consider following ER model

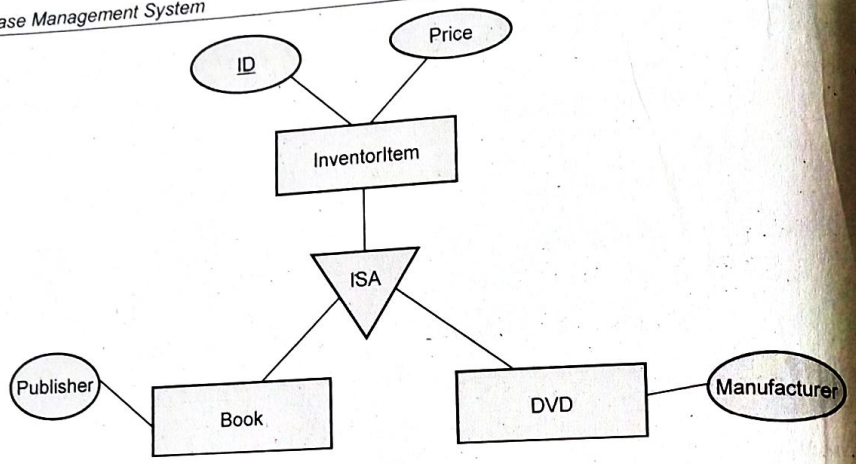


Following SQL Statement illustrates this mapping

```
CREATE TABLE Department(DeptID CHAR(11),
        DeptName CHAR(20),
        Bldg_No CHAR(5),
        PRIMARY KEY (DeptID,Bldg_No),
        FOREIGN KEY(Bldg_No) References Buildings on delete cascade
        )
```

2.13.5 Mapping of Specialization / Generalization(EER Construct) to Relational Mapping

The specialialization/Generalization relationship(Enhanced ER Construct) can be mapped to database tables(relations) using three methods. To demonstrate the methods, we will take the – InventoryItem, Book, DVD



Method 1 : All the entities in the relationship are mapped to individual tables

InventoryItem(ID , Price)
Book(ID,Publisher)
DVD(ID, Manufacturer)

Method 2 : Only subclasses are mapped to tables. The attributes in the superclass are duplicated in all subclasses. For example -

Book(ID, Price,Publisher)
DVD(ID, Price,Manufacturer)

Method 3 : Only the superclass is mapped to a table. The attributes in the subclasses are taken to the superclass. For example -

InventoryItem(ID , Price,Publisher,Manufacturer)

This method will introduce **null** values. When we insert a **Book** record in the table, the **Manufacturer** column value will be null. In the same way, when we insert a **DVD** record in the table, the **Publisher** value will be null.

Example 2.13.1 *Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted. Also construct appropriate tables for the ER diagram you have drawn.*

Solution :

ER Diagram - Refer example 2.4.15.

Relational Mapping

patients (P_id, name, insurance, date-admitted, date-checked-out)
doctors (Dr_id, name, specialization)
test (testid, testname, date, time, result)
doctor-patient (P_id, Dr_id)
test-log (testid, P_id) performed-by (testid, Dr_id)

Review Question

1. *Writing short note on – Mapping ISA relationship of E-R diagram to tables*

SPPU : Nov.-18, (End Sem), Marks 4

2.14 Multiple Choice Questions

Q.1 ER model was introduced by _____.

[a] E.F. Codd [b] P.P. Chen
[c] Bjarne Stroustrup [d] None of these

Q.2 ER diagram an ellipse represents _____.

[a] weak entity [b] relationship
[c] attribute [d] entity class

Q.3 In ER diagram the relationship is represented by _____.

[a] rectangle [b] ellipse
[c] diamond [d] circle

Q.4 Relationship among entities of a single class is called as _____.

[a] IS-A relationship [b] recursive relationship
[c] HAS -A relationship [d] none of these

Q.5 The relationship used for connecting entities of different types when identifiers are different _____.

[a] HAS-A relationship [b] IS-A relationship
[c] recursive relationship [d] none of these

Q.6 Which is not an example of strong entity type ?

[a] Employee [b] Department
[c] Emp_ID [d] Course

Q.7 If person is an entity then "Ankita" and "Prajkta" is the entity _____.
[a] characteristics [b] field
[c] identifier [d] instance

Q.8 Properties that describe the characteristics of entities are called :
[a] entities [b] attributes
[c] identifiers [d] relationships

Q.9 A student RollNumber, Name and Marks are all examples of _____.
[a] entities [b] attributes
[c] relationships [d] none of these

Q.10 An attribute which consists of a group of attributes is called as _____.
[a] composite attribute [b] single valued attribute
[c] multi-valued attribute [d] none of these

Q.11 The attribute "age" is calculated from "date_of_Birth". The attribute "age" is _____.
[a] single valued [b] multi valued
[c] composite [d] derived

Q.12 The most common type of relationship used in data modeling is _____.
[a] unary [b] binary
[c] ternary [d] none of these

Q.13 An entity type whose existence depends on another entity type is called ___ entity.
[a] strong [b] weak
[c] dependent [d] mutually dependent

Q.14 Every weak entity set can be converted into a strong entity set by :
[a] Using generalization [b] Adding appropriate attributes
[c] Using aggregation [d] None of the above

Q.15 In a one-to-many relationship, the entity that is on the one side of the relationship is called a(n) _____ entity.
[a] parent [b] child
[c] instance [d] subtype

Q.16 Which of the following is NOT a basic element of all versions of the E-R model ?
[a] Entities [b] Attributes
[c] Relationships [d] Primary keys

Q.17 The entity employee has three candidate keys : 1) EmpID 2) Email 3) Date_of_joining 4) Designation. Suggest best primary key for this entity :
[a] EmpID [b] Email
[c] Date_of_joining [d] Designation

Q.18 Foreign key is one in which the ____ of one relation is referenced in another relation.
[a] candidate key [b] constraint
[c] primary key [d] foreign key

Q.19 The relationship between the two tables are created using _____.
[a] primary key [b] check constraints
[c] candidate key [d] foreign key

Q.20 Constraints are preferred methods for enforcing _____.
[a] data abstraction [b] data access
[c] data inheritance [d] data integrity

Answer Keys for Multiple Choice Questions :

Q.1	b	Q.2	c	Q.3	c	Q.4	b
Q.5	a	Q.6	c	Q.7	d	Q.8	b
Q.9	b	Q.10	c	Q.11	d	Q.12	b
Q.13	b	Q.14	b	Q.15	a	Q.16	d
Q.17	a	Q.18	c	Q.19	d	Q.20	d

❑❑❑