Team HW Part 2

Do Review Questions 3, 13 from Modern Database Management Book (Page 101-102)
Do Problems and Exercises 8, 13, 16, 19 from Modern Database Management Book (Page 102-109)
Do Review Questions 4, 7, 8, 9, 16 from Modern Database Management Book (Page 143-144)
Do Problems and Exercises 12, 14, 15, 17 from Modern Database Management Book (Page 144-146)
Do Review Questions 1, 2, 3, 8, 12, 14, 21 from Modern Database Management Book (Page 194)
Do Problems and Exercises 1 (I, II, III)a, 2 (I, II, III)a, 3, 8, 12, 15, 25 from Modern Database Management Book (Page 195-202) (19, 26)

Do Review Questions 3, 13 from Modern Database Management Book (Page 101-102)

3.Contrast the following terms: a. stored attribute; derived attribute
b. simple attribute; composite attribute c. entity type; relationship type d. strong entity type; weak entity type e. degree; cardinality f. required attribute; optional attribute
g. composite attribute; multivalued attribute h. ternary relationship; three binary relationships

Solution:

a. stored attribute; derived attribute

Derived Attribute	Stored Attribute
An attribute that can be obtained using other stored attributes.	Attribute from which value of other attribute can be derived.
It is impossible to determine the value of the attribute using other attributes.	It is impossible to determine the value of the attribute using other attributes.
It is variable in nature.	It is fixed in nature.
It is not compulsory to store a derived attribute in the database.	It is compulsory to store the the stored attributes in the database.
Ex: Date_of_birth	Ex: age

b. simple attribute; composite attribute

Simple attribute	Composite attribute
Simple attributes are atomic values, which cannot be divided further.	Composite attributes are made of more than one simple attribute.
Attributes that can’t be divided into subparts are called Simple or Atomic attributes.	Composite attributes can be divided into smaller subparts. These subparts represent the basic attributes with independent meanings of their own.

Ex: phone_no	Ex: name_of_student

c. entity type; relationship type

Entity type	Relation type
Entity is a real-world object**.**	Relationship is an association between the entities.
Represented by a rectangle in the ER diagram.	Represented by a rhombus in the ER diagram.

d. strong entity type; weak entity type

Strong entity	Weak entity
The strong entity has a primary key.	The Weak entity has a partial discriminator key.
The strong entity is independent of any other entity in a schema.	Weak entity depends on the strong entity for its existence.
Denoted by a single rectangle.	Denoted by a double rectangle.
Strong entity may or may not have total participation in the relationship.	Weak entity always has total participation in the identifying relationship.

e. degree; cardinality

Degree: degree is the number of attributes in a relation (number of columns)

Cardinality: Cardinality is the number of tuples in a relation (number of rows).

f. required attribute; optional attribute

Required Attribute	Optional Attribute
Required attribute must have a value for every entity(or relationship) instance with which it is associated.	Optional attribute may not have a value for every entity(or relationship) instance with which it is associated.

g. composite attribute; multivalued attribute

Composite attribute	Multivalued attribute
Composite attributes are not atomic because they are assembled using some other atomic attributes.	A multivalued attribute may have one or more values for a particular entity.
Ex: A person’s address, which is	Ex: Location as the attribute of an entity

composed of atomic attributes, such as City, Zip, and Street.	called ENTERPRISE is multivalued, because each enterprise can have one or more locations.

h. ternary relationship; three binary relationships

Ternary Relationship
In ternary relationship three different entities takes part in a relationship.
Relationship degree = 3
Ex: Consider a Mobile manufacture company. Three different entities involved: Mobile - Manufactured by company. Part - Mobile Part which company get from Supplier. Supplier - Supplier supplies Mobile parts to Company.

13.What is the degree of a relationship? List the three types of relationship degrees described in the chapter and give an example of each

Solution:

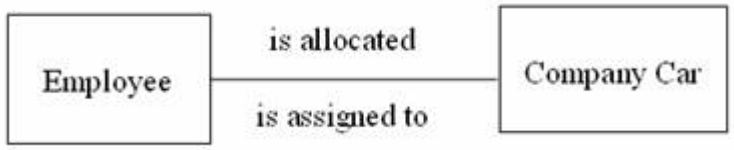
The degree of a relationship which is also known as cardinality is the number of occurrences in one entity which are associated or linked to the number of occurrence in another.

The three types of relationship degrees are:

1. One-to-one
2. One-to-many
3. Many-to-many

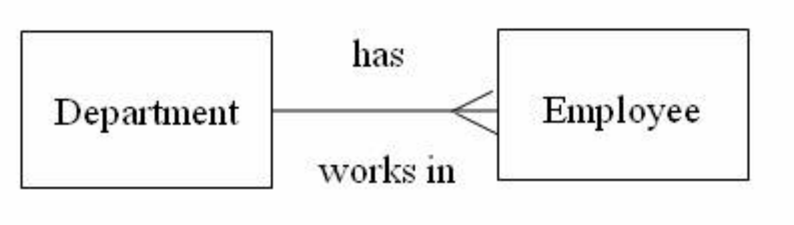
Examples for each :-

1. One-to-one: Where one occurrence of an entity relates to only one occurrence of the other entity.
   Ex: an employee is allocated a company car, which can only be driven by that employee.
   Therefore, there is a one-to-one relationship between employee and company car.

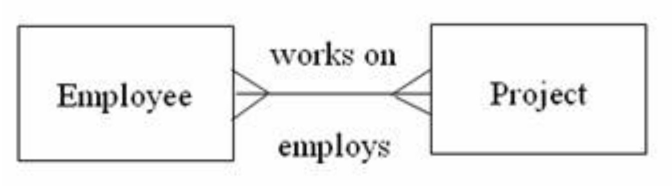


2. One-to-many: Where one occurrence of an entity relates to many occurrences of the other entity.

Ex: taking the employee and department entities, an employee works in one department but a department has many employees.
Therefore, there is a one-to-many relationship between department and employee.

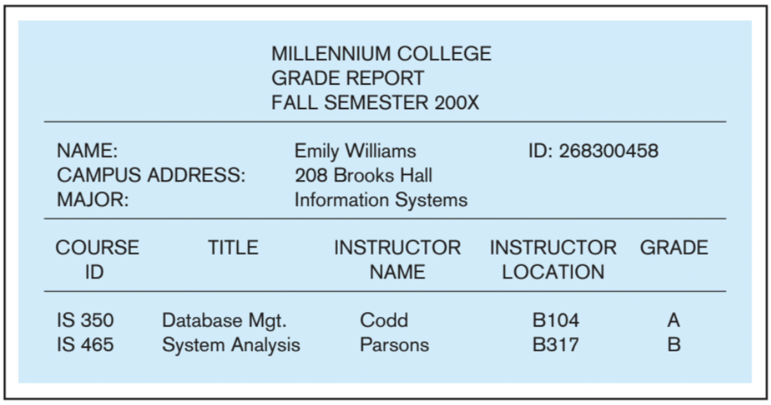


3. Many-to-many: Where many occurrences of an entity relates to many occurrences of the other entity.
Ex: an employee may work on several projects at the same time and a project has a team of many employees.
Therefore, there is a many-to-many relationship between employee and project.



Do Problems and Exercises 8, 13, 16, 19 from Modern Database Management Book (Page 102-109)

8. Figure 2-25 shows a grade report that is mailed to students at the end of each semester. Prepare an ERD reflecting the data contained in the grade report. Assume that each course is taught by one instructor. Also, draw this data model using the tool you have been told to use in the course. Explain what you chose for the identifier of each entity type on your ERD.

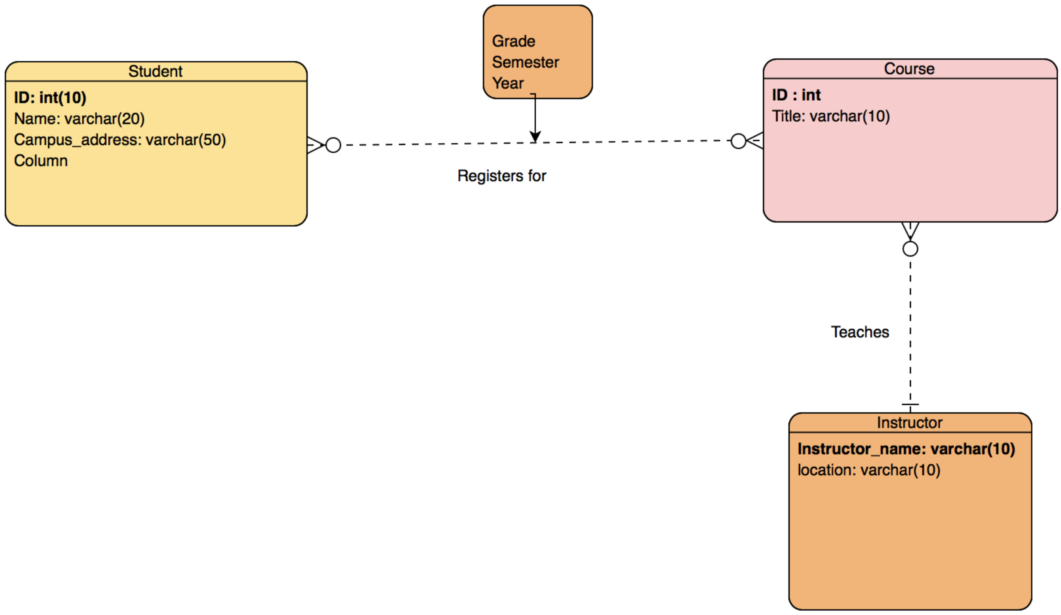

MILLENNIUM COLLEGE
GRADE REPORT
FALL SEMESTER 200X

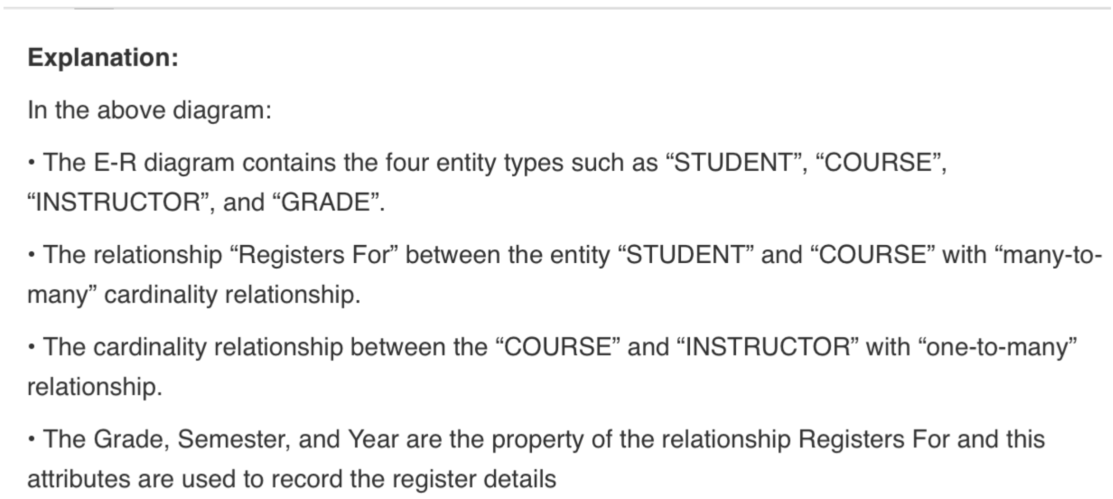
NAME:	Emily Williams	ID: 268300458
CAMPUS ADDRESS:	208 Brooks Hall	
MAJOR:	Information Systems	

COURSE ID	TITLE	INSTRUCTOR NAME	INSTRUCTOR LOCATION	GRADE
IS 350	Database Mgt.	Codd	B104	A
IS 465	System Analysis	Parsons	B317	B

FIGURE 2-25 Grade report

Solution:





Explanation:

In the above diagram:

• The E-R diagram contains the four entity types such as “STUDENT”, “COURSE”, “INSTRUCTOR”, and “GRADE”.

• The relationship “Registers For” between the entity “STUDENT” and “COURSE” with “many-to-many” cardinality relationship.

• The cardinality relationship between the “COURSE” and “INSTRUCTOR” with “one-to-many” relationship.

• The Grade, Semester, and Year are the property of the relationship Registers For and this attributes are used to record the register details

Identifier for each entity types:

The above E-R diagram tells clearly that it contains the four entity types such as “STUDENT”, “COURSE”, “INSTRUCTOR”, and “GRADE”.

For “STUDENT” entity type,

• The entity “STUDENT” chooses the “Student ID” as its identifier attribute because it is unique.

• The attributes of the entity “STUDENT” is “Name”, “Address”, and “Major”.

For “COURSE” entity type,

• The entity “COURSE” chooses the “Course ID” as its identifier attribute because it is unique.

• The attributes of the entity “COURSE” is “Title”.

For “INSTRUCTOR” entity type,

• The entity “INSTRUCTOR” chooses the “Instructor Name” as its identifier attribute because it is unique.

• During the analysis work, it can be proven that the composite identifier is combined of Instructor Name and Location or create a new identifier as Instructor ID will play as unique identifier.

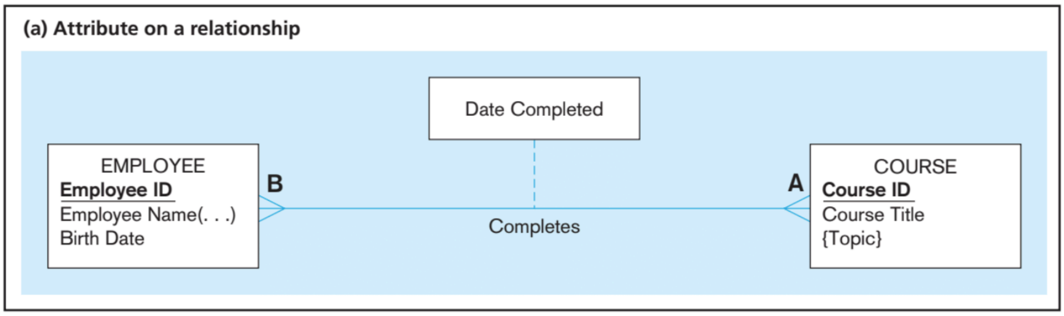
For “Registers For” relationship,

• The Grade, Semester, and Year are the property of the relationship Registers For.

• It is represented in rectangle box with dash line connected in relationship Registers For.

13.Modify Figure 2-11a to model the following additional information requirements: The training director decides, for each employee who completes each class, who (what employees) should be notified of the course completion. The training director needs to keep track of which employ-ees are notified about each course completion by a student. The date of notification is the only attribute recorded about this notification.

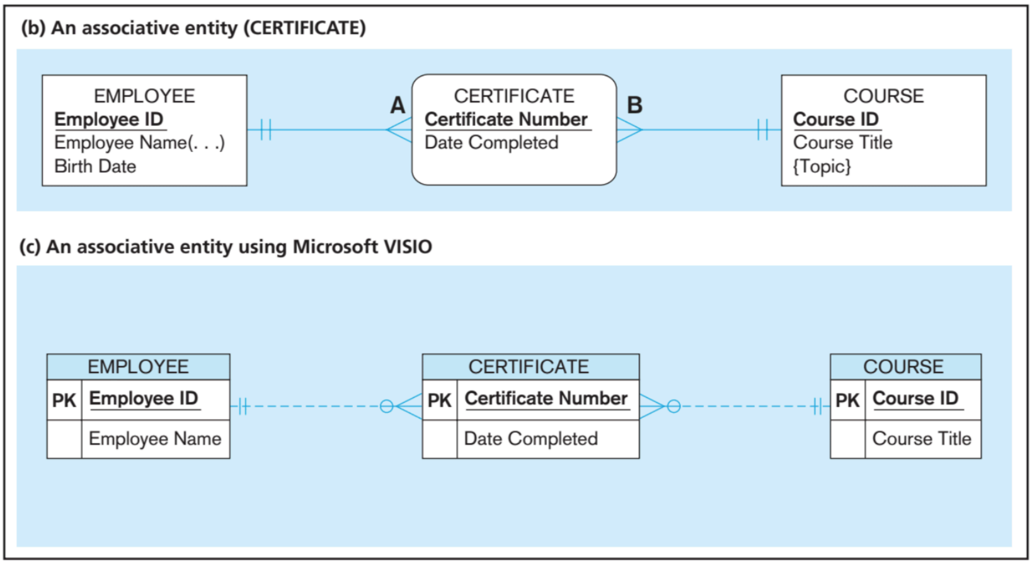
FIGURE 2-11 An associative entity



(a) Attribute on a relationship

(continued)

FIGURE 2-11 *(continued)*

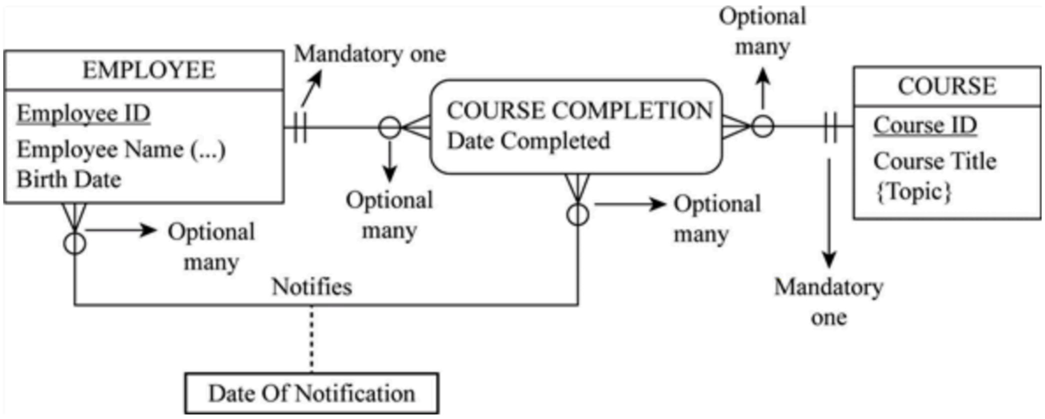


(b) An associative entity (CERTIFICATE)

(c) An associative entity using Microsoft VISIO

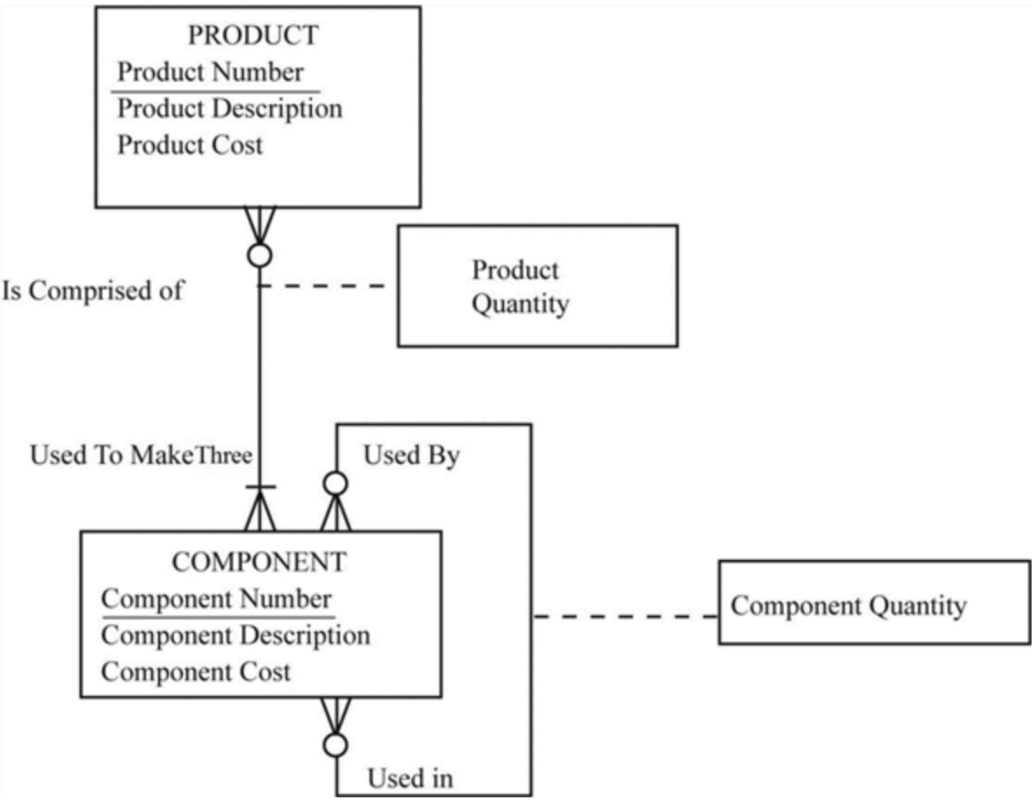
Solution:

The course completion should be notified to the employees by training director.
The Training director will track the notification and the attribute Date of Notification is used to record the notification.



16. Assume that at Pine Valley Furniture each product(described by product number, description, and cost) com-prises at least three components (described by component number, description, and unit of measure), and components are used to make one or many products. In addition, assume that components are used to make other components and that raw materials are also considered to be components. In both cases of components, we need to keep track of how many components go into making something else. Draw an ERD for this situation, and place minimum and maximum cardinalities on the diagram. Also, draw a data model for this situation using the tool you have been told to use in your course.

Solution:



Do Review Questions 4, 7, 8, 9, 16 from Modern Database Management Book (Page 143-144)

4. State two conditions when a database designer should consider using Supertype/subtype relationships.

   Solution: Database designer should consider supertype/subtype relationships if the following conditions are present:

   1. When there are attributes that apply to some instances of an entity type.
   2. And the instances of a subtype participate in a relationship unique to that subtype.
      Ex: Person has two subtype i.e. Student and Teacher.

7. What is attribute inheritance? Why is it important?
Solution:
This is when subtype entities inherit values of all attributes of supertype. When the instance of subtype is also an instance of supertype.
It is important because it makes the model look easier and we can identify which is supertype and which is subtype.

8. Give an example of each of the following:
a. a supertype/subtype relationship where the disjoint rule applies.
b. a supertype/subtype relationship where the overlap rule applies.

Solution:

a. The supertype PERSON contains subtypes MALE and FEMALE
   Here applying the disjoint rule, the person can be either male or female at a time but not both at the same time.
b. The supertype PERSON contains subtypes STUDENT and INSTRUCTOR
   The supertype PERSON may have both STUDENT and INSTRUCTOR subtypes at the same time.

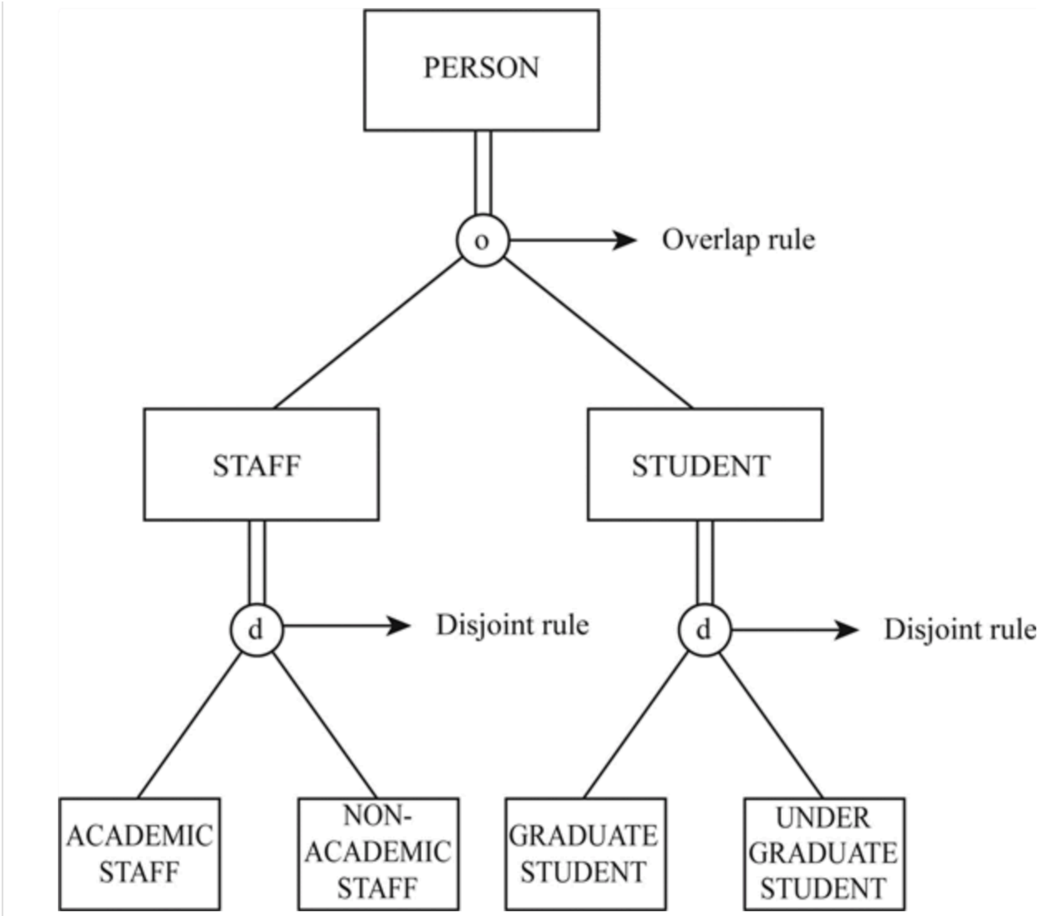
9. What type of business rules are captured in an EER diagram?
Solution: It is the statement used to define or limit some business feature. It is planned to declare the business structure or to organize the behaviour of the bunisess.
Types are:

1. Conditions
2. Relationship Constraints
3. Supertype or subtype relation

16. When might a supertype/subtype hierarchy be useful?

Solution: It is useful when there various different subtypes as well as super types.

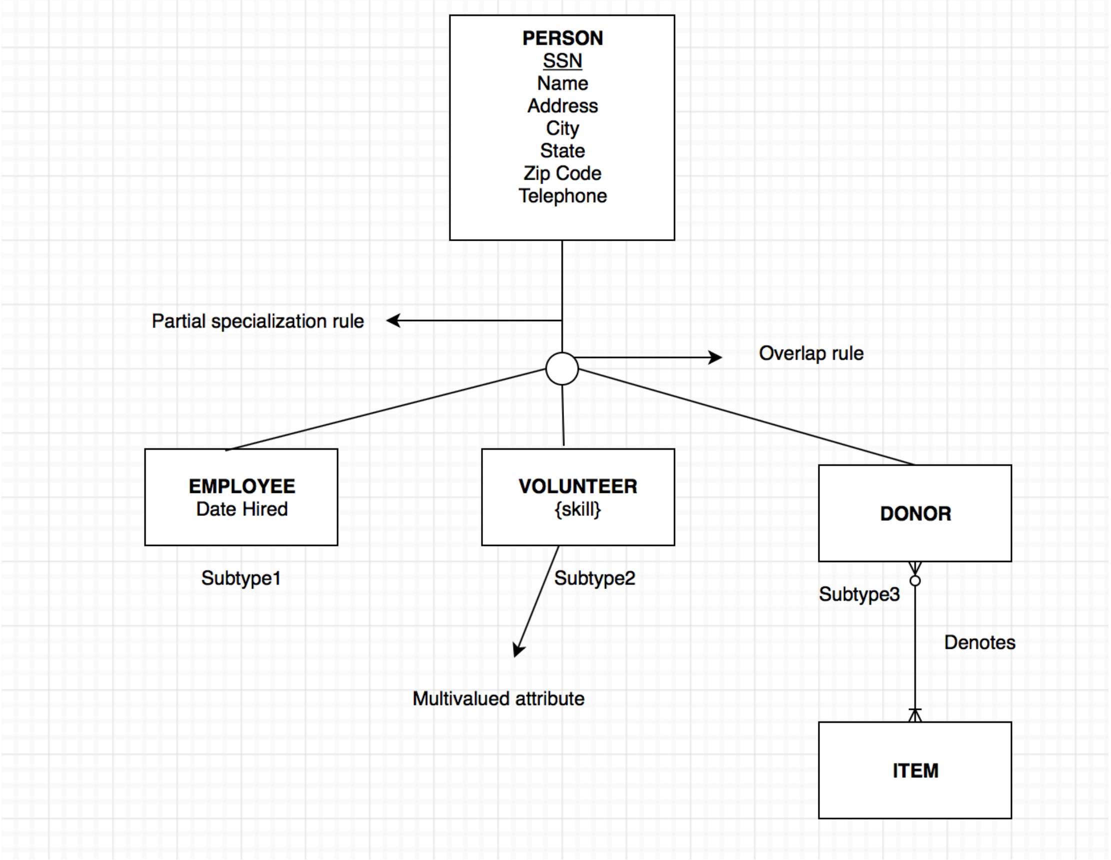
Example:


PERSON
o
Overlap rule
STAFF
STUDENT
d
Disjoint rule
d
Disjoint rule
ACADEMIC STAFF
NON-ACADEMIC STAFF
GRADUATE STUDENT
UNDER GRADUATE STUDENT

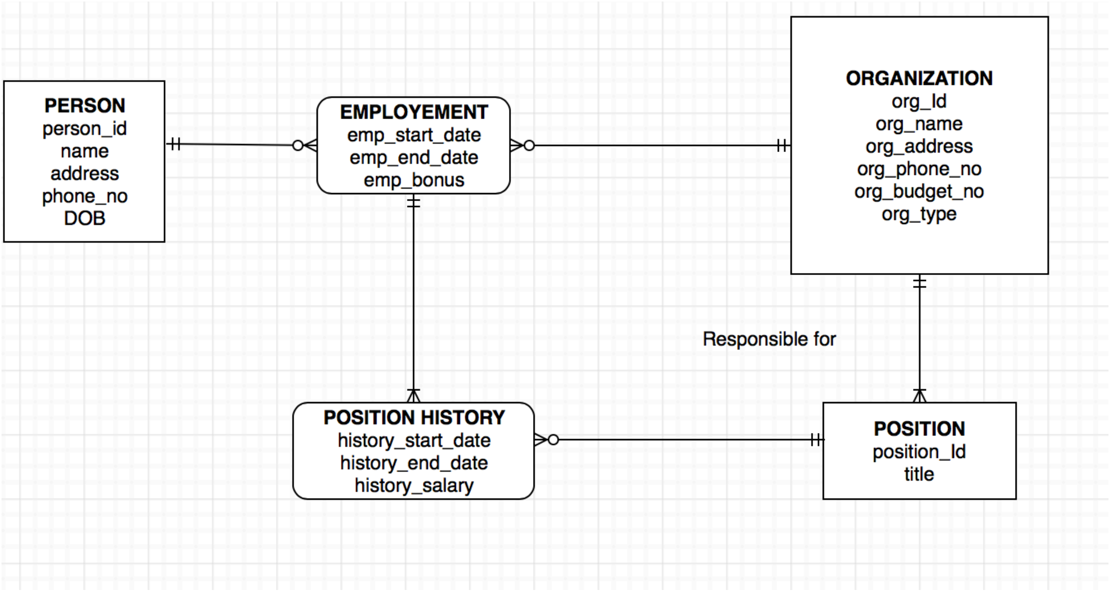
14, 15, 17

12.

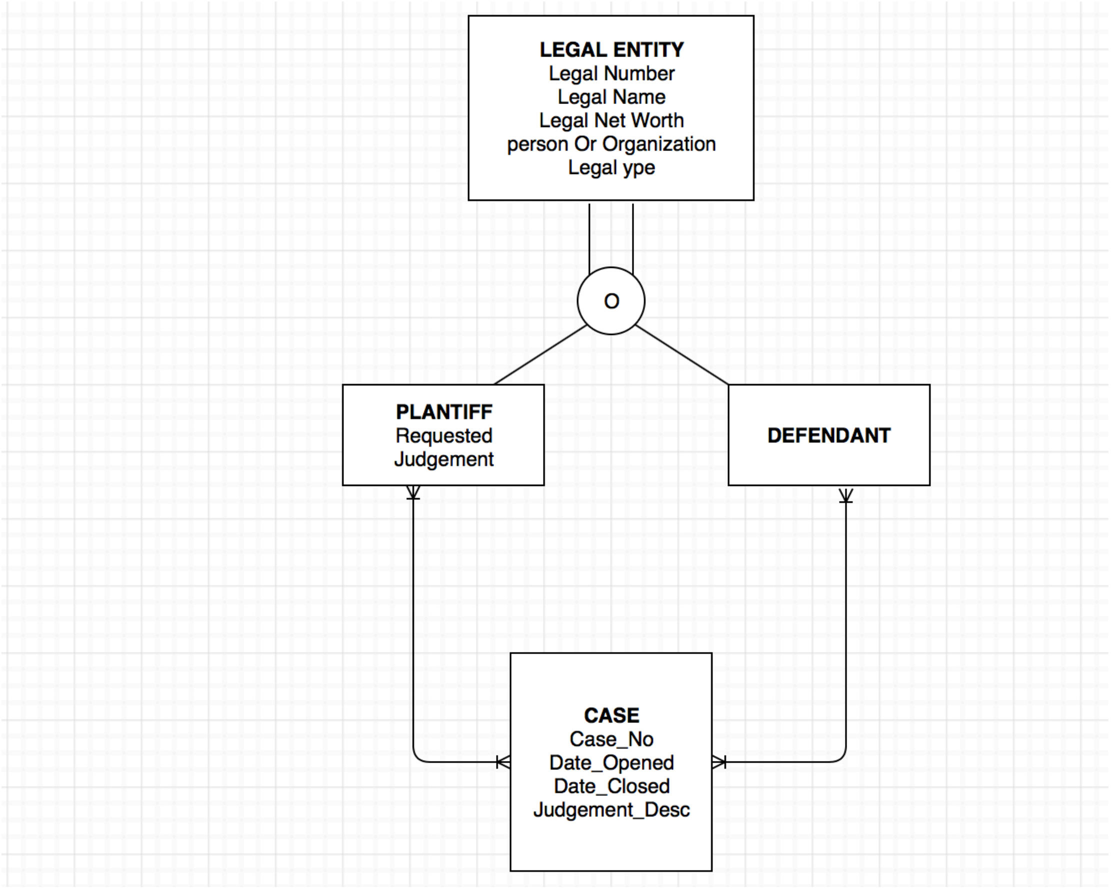
Solution:



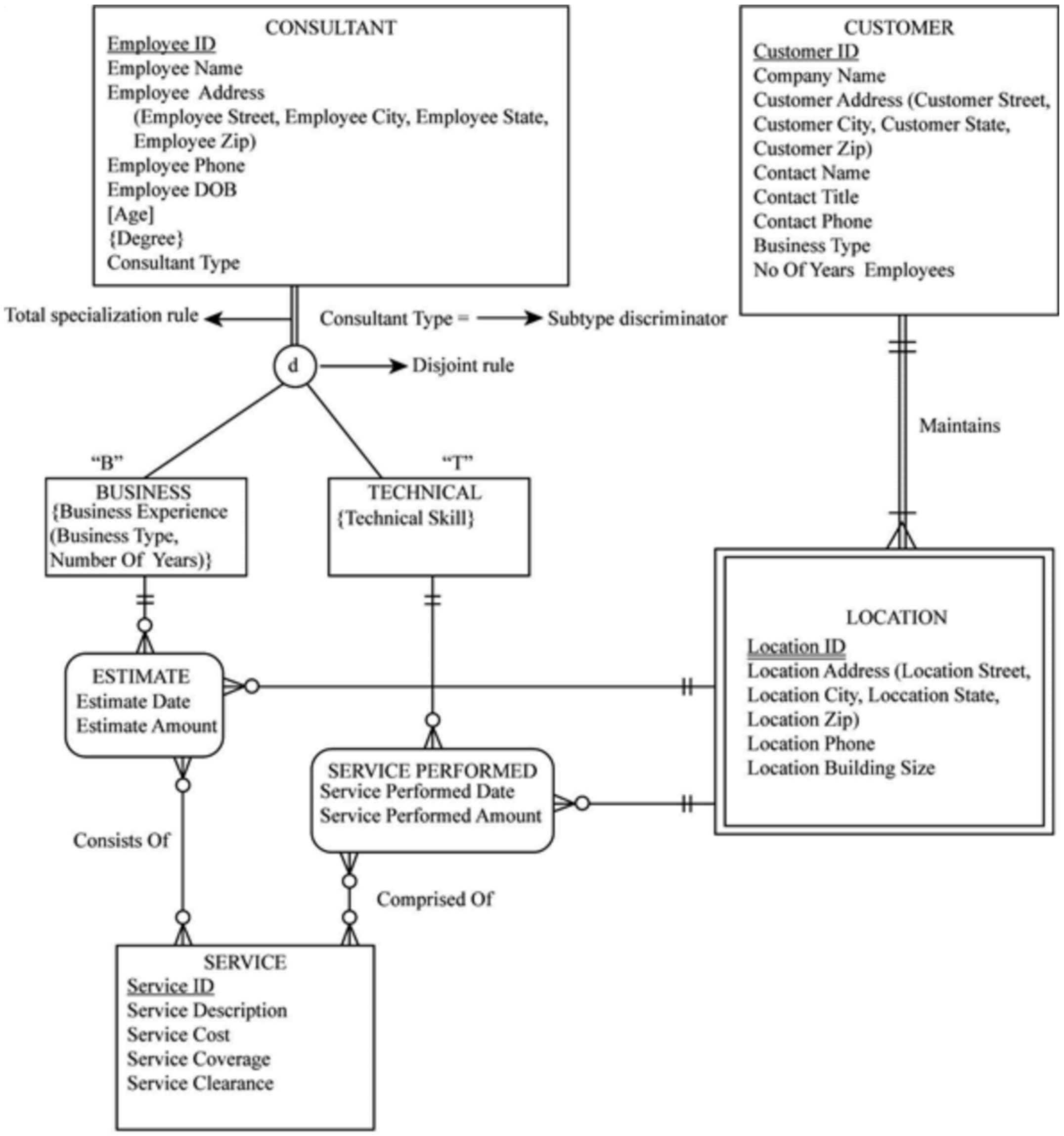
14. Solution:


PERSON
person_id
name
address
phone_no
DOB
EMPLOYEMENT
emp_start_date
emp_end_date
emp_bonus
ORGANIZATION
org_Id
org_name
org_address
org_phone_no
org_budget_no
org_type
Responsible for
POSITION HISTORY
history_start_date
history_end_date
history_salary
POSITION
position_Id
title

15.


LEGAL ENTITY
Legal Number
Legal Name
Legal Net Worth
person Or Organization
Legal ype
O
PLANTIFF
Requested
Judgement
DEFENDANT
CASE
Case_No
Date_Opened
Date_Closed
Judgement_Desc

17.



1. Define each of the following terms: a. determinant b. functional dependency c. transitive dependency d. recursive foreign key e. normalization f. composite key g. relation h. normal form i. partial functional dependency j. enterprise key k. surrogate primary key

Solution:

a. Determinant:
A determinant in a database table is any attribute that you can use to determine the values assigned to other attributes in the same row.

b. Functional dependency:
Functional dependency is a relationship that exists when one attribute uniquely determines another attribute. If R is a relation with attributes X and Y, a functional dependency between the attributes is represented as X->Y, which specifies Y is functionally dependent on X.

c. Transitive dependency: When an indirect relationship causes functional dependency it is called Transitive Dependency.

If P -> Q and Q -> R is true, then P-> R is a transitive dependency.

d. Recursive foreign key: It is the key which refers back to the same table i.e. the foreign key is in the same table. Parent and child table is the same.
e. Normalization: is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion and updation anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.
f. Composite key: A key that has more than one attributes. A composite key, in the context of relational databases, is a combination of two or more columns in a table that can be used to uniquely identify each row in the table.
g. Relation: defines a relationship between tables.
h. Normal form: To make the tables les vulnerable, higher normal form should be applied. Certain rules to determine the logical inconsistences and anomalies in the database is known as normal form.
i. Partial functional dependency: if one or more non-primary key attributes are partially dependent on the primary key attribute it is known as partial functional dependency.
j. Enterprise key: It is a type of a primary key, primary key is unique in a table whereas enterprise key is unique across the database.
k. Surrogate primary key: It is the deputy of a primary key, it acts as a primary key in the database but do not have any other significance.

2. Solution:
   1- f
   2- e
   3- a
   4- j
   5- g
   6- d
   7- h
   8- i
   9- c
   10- k
   11- b

3. Contrast the following terms: a. normal form; normalization b. candidate key; primary key c. partial dependency; transitive dependency d. composite key; recursive foreign key e. determinant; candidate key f. foreign key; primary key g. enterprise key; surrogate key

   Solution:

   a. normal form; normalization:

Normal form	Normalization
Certain rules to determine the logical inconsistences and anomalies in the database is known as normal form.	It is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause

	insertion, deletion and updation anomalies. So, it helps to minimize the redundancy in relations.
To make the tables les vulnerable, higher normal form should be applied.	Normal forms are used to eliminate or reduce redundancy in database tables.

b. candidate key; primary key:

Candidate key	Primary key
Candidate key is an attribute or set of attributes that uniquely identifies each row in a table.	It is the key that is unique of each row.
A relation can have more than one candidate key.	Can have only one unique key.

c. partial dependency; transitive dependency:

Partial dependency	Transitive dependency
If one or more non-primary key attributes are partially dependent on the primary key attribute it is known as partial functional dependency.	When an indirect relationship causes functional dependency it is called Transitive Dependency. If P -> Q and Q -> R is true, then P-> R is a transitive dependency.

d. composite key; recursive foreign key:

Composite key	Foreign key
A key that has more than one attributes. A composite key, in the context of relational databases, is a combination of two or more columns in a table that can be used to uniquely identify each row in the table.	A foreign key references the value of the primary key in other table.

e. determinant; candidate key:

Determinant	Candidate key
A determinant in a database table is any attribute that you can use to determine the values assigned to other attributes in the same row.	Candidate key is an attribute or set of attributes from which a primary key is selected.

f. foreign key; primary key:

Foreign key	Primary key
A foreign key references the value	It is the key that is unique of each

of the primary key in other table.	row.
	Can have only one unique key.

g. enterprise key; surrogate key:

Enterprise key	Surrogate key
It is a type of a primary key, primary key is unique in a table whereas enterprise key is unique across the database.	It is the deputy of a primary key, it acts as a primary key in the database but do not have any other significance.

8.
Solution:
a. Second Normal form
b. Third Normal form
c. First Normal form

12.
Solution:
The process used to minimize data redundancy and dependency in a relational database in known as Normalization.
It's purpose is to: avoid redundancy and inconsistent data.

14.
Solution: If the table does not have any duplicates fields or repeation values for any attributes then that table is in 1st normal form.

For any table to be in 2nd normal form, the table should be in 1st normal form and there should be no particial dependencies.

21. Solution:
A primary key is an attribute that uniquely indentifies a relation. The relationship between primary key and functional dependencies is that all the other attributes are fully dependent on the primary key itself and not on any other non prime attribute

Question1- 1,2,3a

The attributes of the relation Employee are Employee ID, Name, Address and date of Employment where Employee ID is the Primary key.
The attributes of the relation Skills are Skillname and Employee ID.

Functional Dependency:
Employee ID is the primary key of Employee relation and all the other attributes are Functionally dependent on EmployeeID.

EmployeeID —> EmployeeName, PayrollAddress, DateEmployed.

EmployeeID is the foreign key to refer the employee relation.
Skill name is a multivalued attribute as an employee can have multiple skills.

3NF

The employee relation is in 3rd normal form as it does not have any transitive dependency and no repeated attributes.
The skill relation is in 3rd normal form as it dos not have any transitive dependency and no repeated attributes.

Question 3

A) Class(CourseNo,SectionNo)
This relation is in 3rd normal form. The relation does not have any transitive dependency.

B) Class(CourseNo,SectionNo,Room)
The above relation is in 3NF

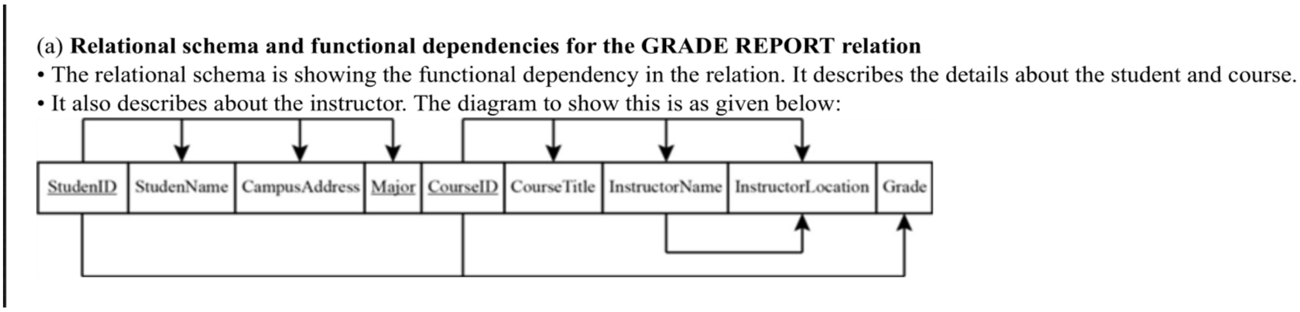
C) Class(CourseNo, SectionNo, Room,Capacity)
This relation is in 2nd Normal form as capacity is functionally dependent on the room.
Decomposition:
Class(CourseNo,SectionNo, Room)
Room(Room,Capacity)

D) Class(CourseNo, SectionNo, CourseName, Room, Capacity)
Functional dependency: CourseNo—> CourseName
Room—>Capacity

Decomposition:
Course (CourseNo, CourseName)
Class (CourseNo, SectionNo,Room)
Room(Room,Capacity)

Question 8

A)



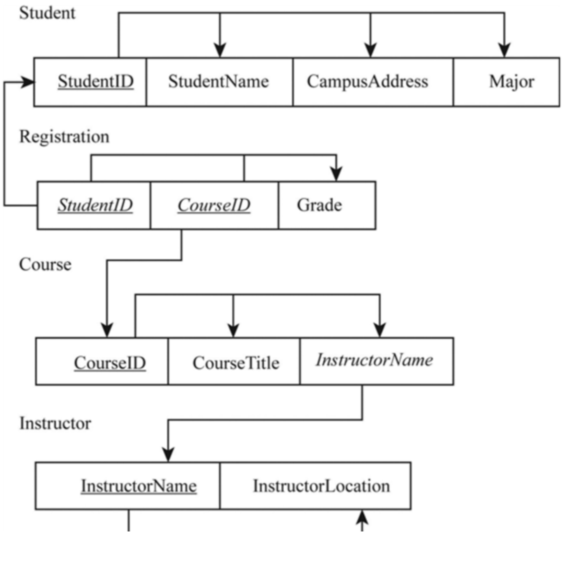
(a) **Relational schema and functional dependencies for the GRADE REPORT relation**

- The relational schema is showing the functional dependency in the relation. It describes the details about the student and course.
- It also describes about the instructor. The diagram to show this is as given below:

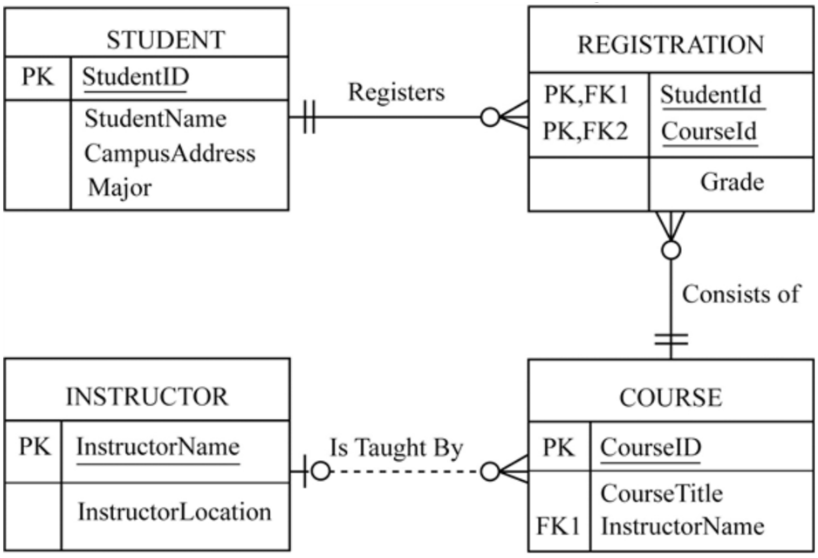
B) Normal form of the relation:
The above relation is in 1st normal form.

C) Convert the relation to 3NF
Student(StudentID, StudentName, CampusAddress, Major)
Registration(StudentID, CourseID, Grade)
Course(CourseID, CourseTitle, InstructorName)
Instructor(InstructorName, InstructorLocation)

D) Relational Schema for 3NF relations with referential integrity:



E) 3NF ERD:



Question 12:
3NF relations:

Course Relation:

CourseID is the primary key
In Prerequisite relation, CourseID and PrereqID are the Attributes.
CourseID is the foreign key for this relation.

Employee Relation:
Employee(EmployeeID, EmployeeName)
Possesses(EmployeeID, SkillCode)
Skill(SkillCode, SkillTitle, SkillType)

The attributes like skilltitle and skillType is not necessary to store value so it can be stored under one primary attribute which is SkillCode in skill relation.

Question 15

1)
Dependencies are
(Origin,Destination) —> Distance
Shipment —> (Origin,Destination)

2) The table is in 1st Normal form.

3) Converting into 3rd normal form:

Origin_Destination_Table(Origin_DestinationID, Origin, Destination, Distance)
Shipment_table(ShipmentNo, Origin_DestinationID)

Question 25

A) Function Dependencies:
ComputerSerialNbr —> PurchasePrice
VendorID —> VendorName,VendorPhone
VendorSupportID —> VendorSupportName, VendorSupportExtension
SoftwareID —> SoftwareName, SoftwareVendor, SoftwareLicenseExpires, SoftwareLicensePrice
UserID —> UserName, UserAuthorizationStarts, UserAuthorizationEnds, UserAuthorizationPassword

B) The relation is not in 3rd Normal form because all the attributes are not fully functionally dependent on ComputerSerialNbr.

C) 3rd Normal form:
Computer(ComputerSerialNbr, PurchasePrice)
Vendor(VendorID, VendorName, VendorPhone)
VendorSupport(VendorSupportID, VendorSupportName, VendorSupportExtension)
Software(SoftwareID, SoftwareName, SoftwareVendor, SoftwareLicenseExpires, SoftwareLicensePrice)
User(UserID, UserName, UserAuthorizationStarts, UserAuthorizationEnds, UserAuthorizationPassword)