

# Computer Science & Information Technology

## Databases

Comprehensive Theory

*with* Solved Examples and Practice Questions



**MADE EASY**  
Publications



### **MADE EASY Publications Pvt. Ltd.**

Corporate Office: 44-A/4, Kalu Sarai (Near Hauz Khas Metro Station), New Delhi-110016

E-mail: [infomep@madeeasy.in](mailto:infomep@madeeasy.in)

Contact: 011-45124660, 8860378007

Visit us at: [www.madeeasypublications.org](http://www.madeeasypublications.org)

### **Databases**

© Copyright by MADE EASY Publications Pvt. Ltd.

All rights are reserved. No part of this publication may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photo-copying, recording or otherwise), without the prior written permission of the above mentioned publisher of this book.

First Edition: 2015

Second Edition : 2016

Third Edition : 2017

Fourth Edition : 2018

Fifth Edition : 2019

Sixth Edition : 2020

Seventh Edition : 2021

**Eighth Edition : 2022**

### Chapter 1

#### The Relational Model ..... 3

1.1	Introduction.....	3
1.2	Database Design and ER Diagrams .....	3
1.3	Entity, Attributes, Entity Set .....	4
1.4	Relationship and Relationship Sets.....	4
1.5	Relationship Constraints .....	6
1.6	Minimization of ER Diagram .....	8
1.7	Self Referential Relationship .....	11
1.8	Weak Entity Set .....	14
1.9	Specialization and Generalization .....	17
1.10	Aggregation.....	17
1.11	Aggregation vs Ternary Relationships.....	18
	<i>Student Assignments</i> .....	20

### Chapter 2

#### Database Design and Normalization ....23

2.1	Definition of Keys and Attributes	
	Participating in Keys .....	23
2.2	Integrity Constraints .....	25
2.3	Closure of set of FDs .....	29
2.4	Attribute Closure.....	29
2.5	Membership Test.....	30
2.6	Closure of Set of Functional Dependencies	
	and Equivalence of Sets of Functional	
	Dependencies.....	31
2.7	Minimal Cover.....	33
2.8	Finding Candidate Keys.....	34
2.9	Problem Caused by Redundancy.....	35
2.10	Normalization of Relations .....	36
2.11	Properties of Decomposition .....	40
2.12	Multivalued Dependencies .....	41
2.13	Fourth Normal Form .....	42
	<i>Student Assignments</i> .....	47

### Chapter 3

#### Relational Algebra.....53

3.1	Introduction.....	53
3.2	Selection and Projection .....	53
3.3	Set Operations .....	55
3.4	The Rename Operation.....	56
3.5	Joins.....	57
3.6	Division.....	60
3.7	The Tuple Relational Calculus.....	66
3.8	The Domain Relational Calculus.....	71
	<i>Student Assignments</i> .....	76

### Chapter 4

#### Structured Query Language (SQL) ....83

4.1	Introduction.....	83
4.2	The Form of a Basic SQL Query.....	84
4.3	Union, Intersect, and Except.....	86
4.4	Nested Queries .....	87
4.5	Correlated Nested Queries .....	89
4.6	Set-Comparison Operators .....	89
4.7	Aggregate Operators.....	91
4.8	The Group by and Having Clauses.....	94
4.9	NULL Values .....	96
	<i>Student Assignments</i> .....	99

### Chapter 5

#### Transaction ..... 103

5.1	Introduction.....	103
5.2	ACID Properties .....	104
5.3	Types of Failures .....	106
5.4	Transaction States.....	107
5.5	Schedule .....	107
5.6	Concurrent Execution of Transaction .....	108
5.7	Problems because of Concurrent Execution....	109
5.8	Serializability .....	110
5.9	Uses of Serializability.....	111
5.10	Classification of Schedules .....	111
	<i>Student Assignments</i> .....	123

## Chapter 6

### Concurrency Control Techniques ..... 127

6.1	Introduction.....	127
6.2	Lock Granularity.....	127
6.3	Shared-Exclusive Locking .....	128
6.4	Two Phase Locking Protocol (2PL).....	128
6.5	Time Stamp based Concurrency Control .....	134
6.6	The Timestamp Ordering Algorithm.....	134
6.7	Multiversion Concurrency Control	
	Techniques.....	138
	<i>Student Assignments</i> .....	139

## Chapter 7

### File Organization and Indexing ..... 141

7.1	File Organization.....	141
7.2	Index Structure .....	143
7.3	Multilevel Index.....	149
7.4	B-Trees.....	151
7.5	Bulk Loading in B+ Tree .....	163
7.6	Join Algorithms .....	165
	<i>Student Assignments</i> .....	169

■■■■

# Databases

---

## Goal of the Subject

The main goal of Data Base management System is to make it possible for users to create, edit and update data in database files. Once created, the DBMS makes it possible to store and retrieve data from those database files.

More specifically, a DBMS provides the following functions:

- Concurrency: concurrent access (meaning 'at the same time') to the same database by multiple users
- Security: security rules to determine access rights of users
- Backup and recovery: processes to back-up the data regularly and recover data if a problem occurs
- Integrity: database structure and rules improve the integrity of the data
- Data descriptions: a data dictionary provides a description of the data

# Databases

---

## INTRODUCTION

---

Very often the subject *Data Base management System* is discussed vastly. But in this book we tried to keep it around the GATE syllabus. Each topic required for GATE is crisply covered with illustrative examples and each chapter is provided with Student Assignment at the end of each chapter so that the students get the thorough revision of the topics that he/she had studied. This subject is carefully divided into seven chapters as described below.

1. **The Relational Model:** In this chapter we discuss ER model and its constraints, types of relationships and we also the minimization of ER diagrams.
2. **Database design and Normalization:** In this chapter we discuss the types of keys in relational model, integrity constraints, functional dependencies of attributes and their properties, decomposition of relationship into 2NF, 3NF, BCNF etc.
3. **Relational Algebra:** In this chapter we discuss the Operators of relational algebra, operations on the set of records and finally we discuss the Tuple Relational Calculus (TRC).
4. **SQL:** In this chapter we discuss the basic form of SQL Query, operator, types of queries, group by and having clauses and finally we discuss the special properties of NULL value.
5. **Transaction:** In this chapter we discuss the ACID properties of a transaction, problem of concurrent execution, serializability etc.
6. **Concurrency Control Techniques:** In this chapter we discuss the various protocols used to achieve concurrent execution with any of the problem discussed in the previous chapter.
7. **File Organization and Indexing:** in this chapter we discuss concept of file organization, indexing techniques and B, B+ trees which form backbone of indexing techniques.



# The Relational Model

---

## 1.1 Introduction

Entity relationships (ER) model is high level database design allows us to describe the data involved in real-world enterprise in terms of objects and their relationships and is widely used to develop initial database design. In overall design process, the ER model is used in a phase called conceptual database design.

## 1.2 Database Design and ER Diagrams

The database design process can be divided into six steps. The ER model is most relevant to the first three steps.

### Requirements Analysis

The very first step in designing a database application is to understand what data is to be stored in the database, what applications must be built on top of it, and what operations are most frequent and subject to performance requirements. In other words, we must find out what the users want from the database. This is usually an informal process that involves discussions with user groups, a study of the current operating environment and how it is expected to change, analysis of any available documentation on existing applications that are expected to be replaced or complemented by the database, and so on.

### Conceptual Database Design

The information gathered in the requirements analysis step is used to develop a high-level description of the data to be stored in the database, along with the constraints known to hold over this data. This step is often carried out using the ER model and is discussed in the rest of this chapter. The ER model is one of several high-level, or semantic, data models used in database design. The goal is to create a simple description of the data that closely matches how users and developers think of the data

### Logical Database Design

We must choose a DBMS to implement our database design, and convert the conceptual database design into a database schema in the data model of the chosen DBMS. We will consider only relational DBMSs, and therefore, the task in the logical design step is to convert an ER schema into a relational database schema.

### 1.3 Entity, Attributes, Entity Set

- Entity is an object that exist and is distinguishable from other objects. For example a person with give UID is an entity as he can be uniquely identified as one particular person.
- An entity may be concrete (person) or abstract (job). It is represented by rectangle.
- An entity set is a collection of similar entities. (All persons having an account at a bank)
- Entity sets need not be disjoint. For example, the entity set employee (all employees of a bank) and entity set customer (all customers of the bank) may have members (Entity) in common.
- An entity is described using a set of attributes, all entities in a given entity set have same attributes, this is what we mean by similar.
- For each attribute associated with an entity set, we must identify domain of attribute which is the set of permitted values (e.g. if a company rates employees on a scale of 1 to 10 and stores rating in a field called rating, the associated domain consist of integers 1 through 10)
- Attribute is represented by an oval.
- Types of attributes and their representations:

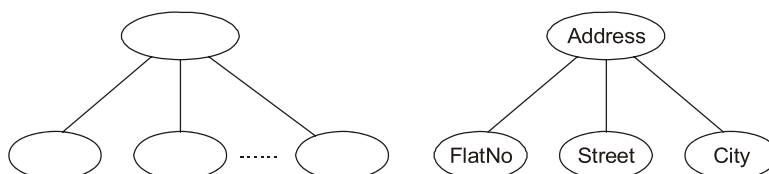
- 1. Key attribute:** The attribute which uniquely identifies each entity in the entity set is called key attribute. For example, account number of customer.

Represented as



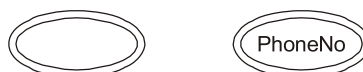
- 2. Composite attribute:** An attribute composed of many other attributes is called as composite attribute. For example, address of employee.

Represented as



- 3. Multivalued attributed:** An attribute consisting more than one value for a given entity. For example, phone number of bank employee.

Represented as



- 4. Derived attribute:** An attribute which can be derived from other attributes of the entity type is known as derived attribute. For example, age of employee (as it can be derived from date of birth).

Represented as



### 1.4 Relationship and Relationship Sets

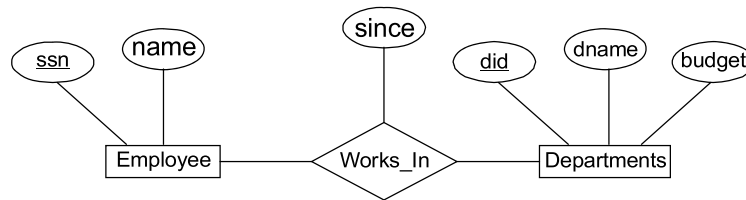
- Relationship is association between two or more entities.
- Relationship set is a set of relationships of same type i.e. relate two or more entity sets.

A relationship set can be thought of as a set of  $n$ -tuples:  $\{(e_1, \dots, e_n) \mid e_1 \in E_1, \dots, e_n \in E_n\}$

Each  $n$ -tuple denotes a relationship involving  $n$  entities  $e_1$  through  $e_n$ , where entity  $e_i$  is in entity set  $E_i$ . In figure, we show the relationship set Works\_In, in which each relationship indicates a department in which an

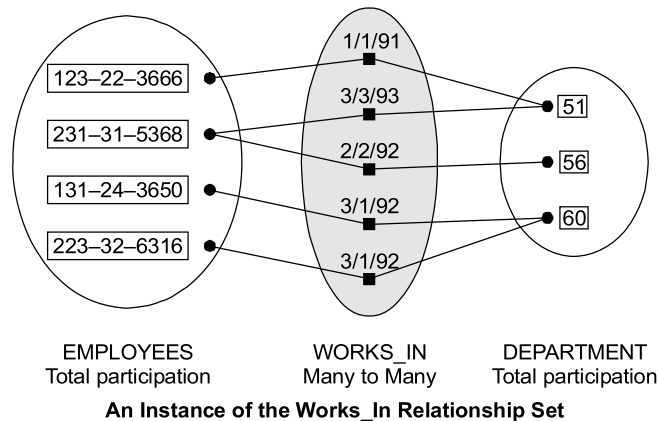


employee works. Note that several relationship sets might involve the same entity sets. For example, we could also have a Manages relationship set involving Employees and Departments.



A relationship can also have **descriptive attributes**. Descriptive attributes are used to record information about the relationship, rather than about any one of the participating entities; for example, we may wish to record that XYZ works in the pharmacy department as of January 1991. This information is captured in Figure by adding an attribute, *since*, to works\_In. A relationship must be uniquely identified by the participating entities, without reference to the descriptive attributes. In the works\_In relationship set, for example, each Works\_In relationship must be uniquely identified by the combination of employee *ssn* and department *did*. Thus, for a given Employee-Department pair, we cannot have more than one associated *since* value.

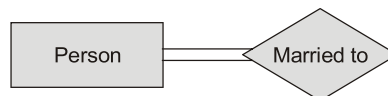
An **instance** of a relationship set is a set of relationships. Intuitively, an instance can be thought of as a 'snapshot' of the relationship set at some instant in time. An instance of the Works\_In relationship set is shown in Figure. Each Employees entity is denoted by its *ssn*, and each Departments entity is denoted by its *did*, for simplicity. The *since* value is shown beside each relationship.



### Degree of a Relationship Set

The number of different entity sets participating in a relationship set is called as degree of relationship set.

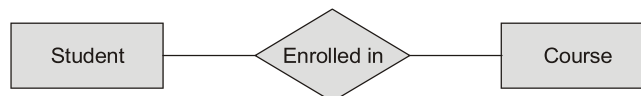
(i) **Unary relationship:** When there is only one entity set participating in a relation.



One person is married by other person.

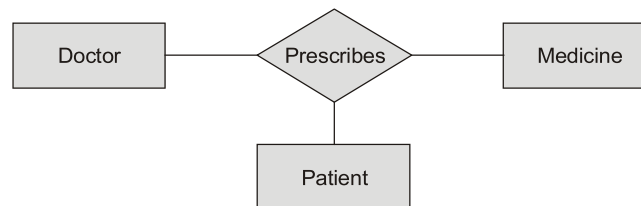
(ii) **Binary relationship:** When there are two entities set participating in a relation.

Eg: Student enroll in course



(iii) **Ternary relationship:** When there are three entity sets participating in a relation.

Eg: Patient visit doctor and doctor prescribes medicine



(iv) **n-ary:** When there are  $n$  number of entity sets participating in a relation.

## 1.5 Relationship Constraints

There are two types of relationship constraints

- Participation constraints
- Cardinality ratio

There are two types of participation constraints:

(i) **Total participation constraints (existence dependency):** The participation of an entity set  $E$  in a relationship set  $R$  is said to be total if every entity in  $E$  participates in at least one relationship in  $R$ . This participation is displayed as a double line connecting.

**Example:** If a company policy states that every employee must work for a department, then an employee entity can exist only if it participates in at least one “worksfor” relationship instance.



(ii) **Partial Dependency:** If only some entities in  $E$  participate in relationship in  $R$ , the participation of entity set  $E$  in relationship  $R$  is said to be partial. This participation is displayed as a single line connecting.

**Example:** Not every employee “Manages” a department.

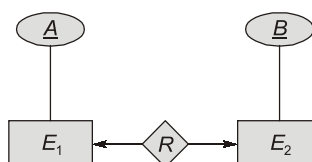


**Mapping Constraints:** The possible cardinality ratio for binary relationship are:

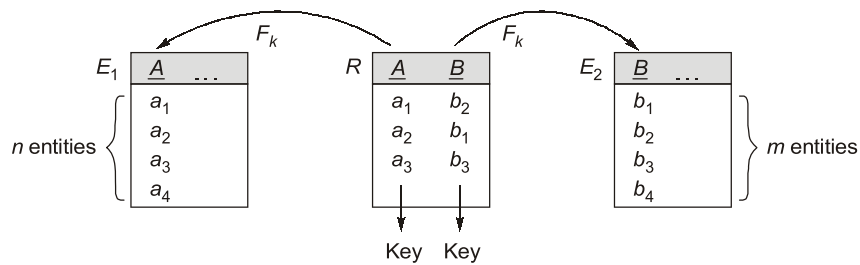
- One to one (1 : 1)
  - One to many (1 :  $M$ )
  - Many to one ( $M$  : 1)
  - Many to many ( $M$  :  $M$ )
- One is represented by ‘→’  
Many is represented by ‘—’

**One to One (1 : 1):** An entity (tuple) in  $E_1$  is associated with atmost one Entity (tuple) in  $E_2$ , and an entity in  $E_2$  is associated with atmost one entity in  $A$

**Example:**



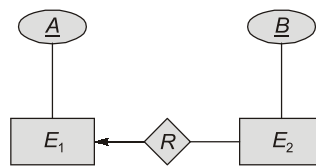
Candidate keys of relation  $R$  ( $A, B$ ) =  $A$ ,  $B$



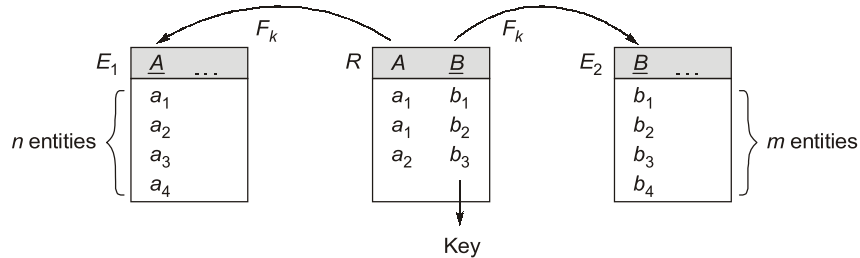
- Maximum possible records in relationship set  $R$  is  $\min(m, n)$ .

**One to Many (1 : M):** An entity (tuple) in  $E_1$  is associated with zero or more entities in  $E_2$  but an entity in  $E_2$  can be associated with at most one entity in  $E_1$

**Example :**



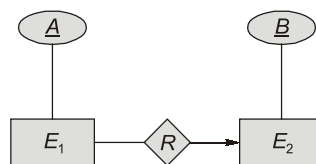
Candidate key of  $R(A, B) = \underline{B}$



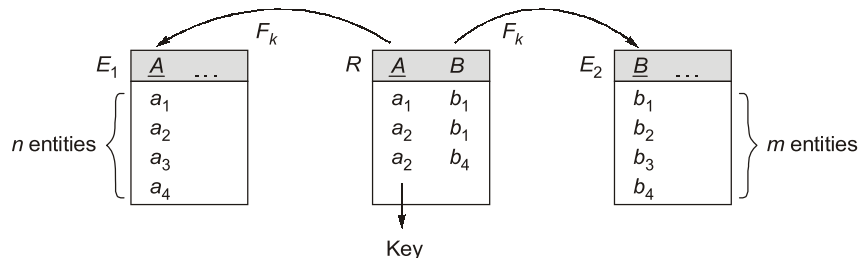
- Maximum possible records in relationship set  $R$  is  $m$  entities (or tuples).
- The relationship set  $R$  is same as right side they mapping.

**Many to one (M : 1):** An entity (tuple) in  $E_2$  is associated with zero or more entities in  $E_1$  but an entity in  $E_1$  can be associated with at most one entity in  $E_2$ .

**Example :**



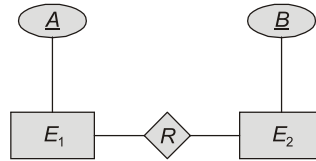
Candidate key of  $R(A, B) = \underline{A}$



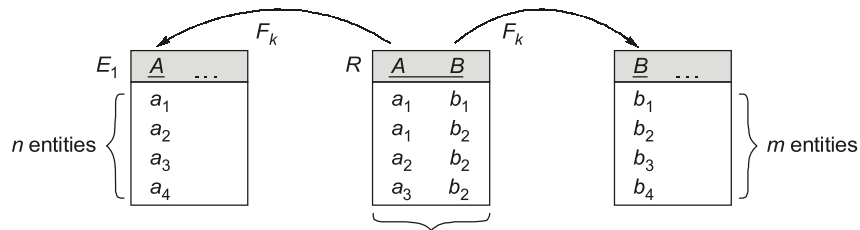
- Maximum possible records in relationship set  $R$  is  $n$  tuples.

**Many to many (M : M):** An entity (tuple) in  $E_2$  is associated with zero or more entities in  $E_1$  and An entity (tuple) in  $E_1$  is associated with zero or more entities in  $E_2$ .

**Example :**



Candidate key of  $R(A, B) = \underline{AB}$



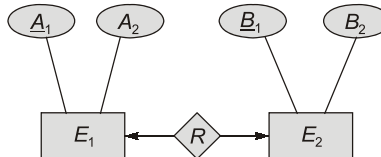
It can have null or duplicate values of A or B or both, so A or B can't be key individually.

- Maximum possible records in relationship set  $R$  is  $n \times m$  tuples.

## 1.6 Minimization of ER Diagram

- 1 : 1 cardinality** with partial participation at both end

**Example:**



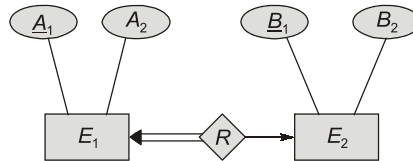
Consider the relation instances of relation  $E_1$ ,  $E_2$  and  $R$ .

$E_1$ :	<table><tr><th><math>A_1</math></th><th><math>A_2</math></th></tr><tr><td>1</td><td><math>P</math></td></tr><tr><td>2</td><td><math>P</math></td></tr><tr><td>3</td><td><math>q</math></td></tr></table>	$A_1$	$A_2$	1	$P$	2	$P$	3	$q$	$E_2$ :	<table><tr><th><math>B_1</math></th><th><math>B_2</math></th></tr><tr><td>11</td><td><math>P</math></td></tr><tr><td>21</td><td><math>q</math></td></tr><tr><td>31</td><td><math>R</math></td></tr></table>	$B_1$	$B_2$	11	$P$	21	$q$	31	$R$	$R$ :	<table><tr><th><math>A_1</math></th><th><math>B_1</math></th></tr><tr><td>1</td><td>11</td></tr><tr><td>3</td><td>21</td></tr></table>	$A_1$	$B_1$	1	11	3	21
$A_1$	$A_2$																										
1	$P$																										
2	$P$																										
3	$q$																										
$B_1$	$B_2$																										
11	$P$																										
21	$q$																										
31	$R$																										
$A_1$	$B_1$																										
1	11																										
3	21																										

$E_1$  and  $R$  can be combined to form a single table with  $A_1$  is primary key (Unique and not NULL) and  $B_1$  as alternate key as well as foreign key. Similarly  $R$  can be combined with  $E_2$ . But can't be combined to both  $E_1$  and  $E_2$  i.e. a single table  $E_1RE_2$  can't be formed.

- If  $R$  is combined with both  $E_1$  and  $E_2$  i.e. a single relation  $E_1RE_2$  then it has no primary key
- Minimum number of tables = 2**  
( $E_1R$ ) and  $E_2$  or  $E_1$  and ( $E_2R$ )

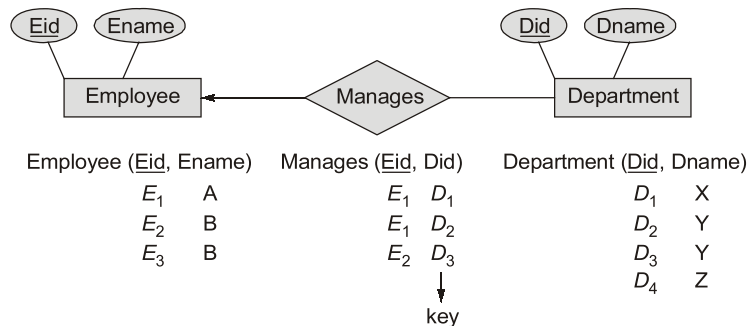
- 1 : 1 cardinality** with total participation atleast one side. The relationship set  $E_1RE_2 (A_1, A_2, \underline{B_1}, B_2)$  has  $B_1$  as primary key and  $A_1$  as alternate key.



Note that  $A_1$  can be null because there may be an entity in  $E_2$  which is not related to any entity of  $E_1$ .

- **Minimum number of tables** = 1 i.e.  $E_1 R E_2$

3. **1 : M Cardinality:** Consider relationship set called manages between the Employees and Departments entity sets such that each department has at most one manager, although a single employee is allowed to manage more than one department. The restriction implies that each department's entity appears in at most one manages relationship in any allowable instance of Manages. This restriction is indicated in the ER diagram of figure.



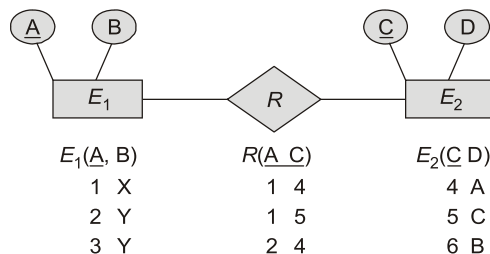
For 1 : M mapping, minimization is performed by combining the relationship set towards many side. Hence the resulting relations after minimization = 2.

Employee ( <u>Eid</u> , Ename) & Dept_manages ( <u>Did</u> , Dname Eid)		
$E_1$	A	$D_1$ X $E_1$
$E_2$	B	$D_2$ Y $E_1$
$E_3$	B	$D_3$ Y $E_2$
		$D_4$ Z Null

Here Eid of Dept\_manages foreign key refers to Eid of Employee.

- If foreign key attribute Eid allows null, that implies partial participation from Employee relation.
- If foreign key attribute Eid does not allow null values, that implies total participation from Employee entity.
- Minimum number of tables in 1 : M or M : 1 = 2.

4. **M : M Cardinality:** Consider the following ER model:



For  $M : M$  mapping, minimization is performed by maintaining separate tables for  $E_1$ ,  $E_2$ ,  $R$ .  $R$  can't be combined with  $E_1$  or  $E_2$  as it has key  $\underline{AC}$ .

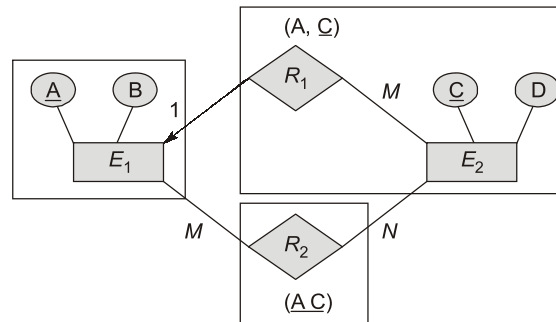
Minimum number of tables for  $M : M$  mapping = 3.

**Example - 1.1**

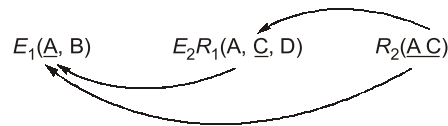
$E_1$  and  $E_2$  are entity sets and  $R_1$  and  $R_2$  are relationship sets related between  $E_1$  and  $E_2$  with  $1 : M$  and  $M : N$  mapping respectively. How many minimum relations are required to represent this database instance?

- (a) 2 (b) 3  
(c) 4 (d) 5

**Solution: b)**



Since  $R_1$  has  $1 : M$  mapping,  $R_1$  can be combined to  $E_2$ , so we get

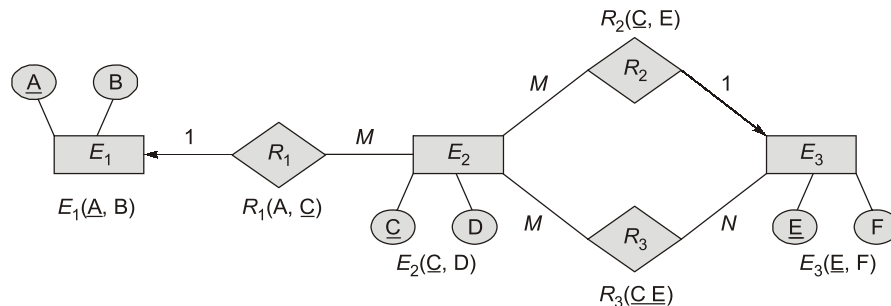


Hence, we have 3 relation and 3 foreign key.

**Example - 1.2**

$E_1$ ,  $E_2$ ,  $E_3$ , entity set.  $R_1$  relationship set between  $E_1$  and  $E_2$  with  $1 : M$  mapping  $R_2$ ,  $R_3$  relationship set between  $E_2$ ,  $E_3$  with  $M : 1$  and  $M : M$  mapping respectively. How many minimum relational table for given ERD?

**Solution:**



$R_1$  is in  $1 : M$  mapping, hence  $R_1$  and  $E_2$  can be combined.

$E_1(\underline{A}, B)$  and  $R_1 E_2(\underline{A}, \underline{C}, D)$

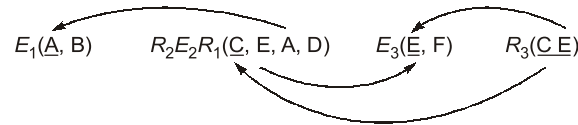
also  $R_2$  has  $M : 1$  mapping, hence  $R_2$  and  $E_2 R$ , can be combined

$E_3(\underline{E}, F)$  and  $R_2 E_2 R_1(\underline{C}, \underline{E}, A, D)$

$R_3$  has  $M : M$  mapping, hence it will be a separate table.

$R_3(\underline{C}, E)$

$\therefore$  final relations are:



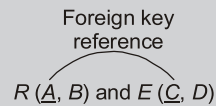
4 relations and 4 foreign key.

**NOTE**



When to minimize?

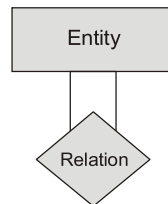
If relationship set  $R$  is having a key as  $A$  which is also foreign key referencing to entity set  $E$ , then  $R$  and  $E$  combined to a single entity set e.g.



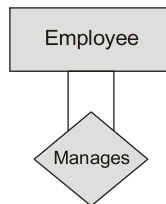
$R$  and  $E$  combined to a single table =  $RE(\underline{C}, B, D)$

## 1.7 Self Referential Relationship

Relationship set relates to same entity sets i.e. pair of entities relating to each other.

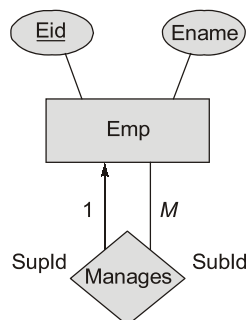


**Example:** Consider the relationship between manages and employees.



All people working in an organization are employees and few of the employees take the role of manager.

- Each employee manages more than one employee but a employee has only one manager, i.e.  $1 : M$  relation exist.



**Summary**

In this chapter we presented the modeling concepts of a high-level conceptual data model, the Entity-Relationship (ER) model. We started by discussing the role that a high-level data model plays in the database design process, we defined the basic ER model concepts of entities and their attributes. Which can be nested arbitrarily to produce complex attributes:

- Simple or atomic
- Composite
- Multivalued
- Derived

We also briefly discussed stored versus derived attributes. Then we discussed the ER model concepts at the schema or “intension” level:

- Entity types and their corresponding entity sets
- Key attributes of entity types
- Value sets (domains) of attributes
- Relationship types and their corresponding relationship sets
- Participation roles of entity types in relationship types

We presented two methods for specifying the structural constraints on relationship types.

The first method distinguished two types of structural constraints:

- Cardinality ratios (1:1, 1:N, M:N for binary relationships)
- Participation constraints (total, partial)

We discussed weak entity types and the related concepts of owner entity types, identifying relationship types, and partial key attributes.

The ER modeling concepts we have presented thus far—entity types, relationship types, attributes, keys, and structural constraints—can model many database applications. However, more complex applications—such as engineering design, medical information systems, and telecommunications—require additional concepts if we want to model them with greater accuracy.

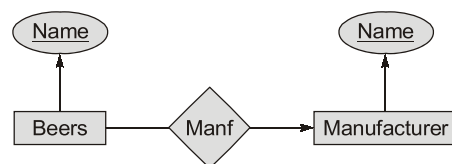
**Student's  
Assignment**

**Q.1** Which of the following statement above ER models is/are correct?

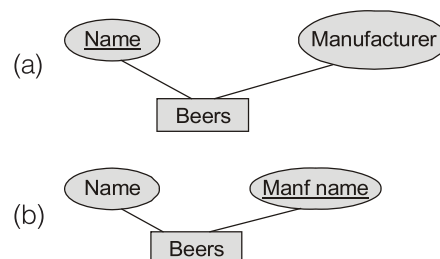
- I. Many-many relationships cannot be representation ERD.
- II. Relationship sets can have attributes of their own.
- III. All many to one relationships are represented by the relationships between a weak and a non weak entity set.

- (a) II only                      (b) III only  
(c) II and III only        (d) I and II only

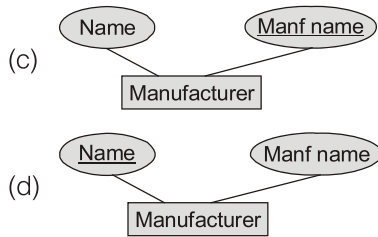
**Q.2** Figure shows an ER diagram:



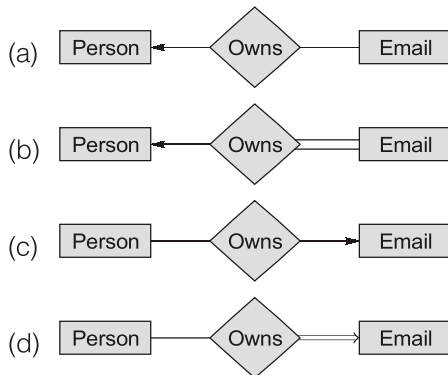
Which of the following best describes the above ER diagram?



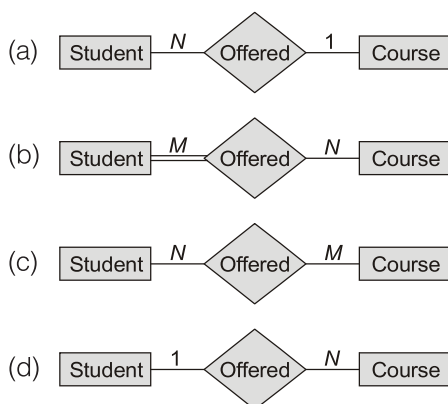




**Q.3** Suppose we have two entity sets person, E-mail and use a relationship Owns. A person own atmost one E-mail account but an email account can be owned by multiple persons. Which of the following is an ER diagram based on above description.



**Q.4** A student can take one or more courses and courses can be offered to any number of students. Which of the following represents given scenario in ER-model



**Q.5**  $R$  is relationship, with 1 : 1 cardinality 30% participation at  $E_1$  end and 70% participation at  $E_2$  end which is the best possible design?

- (a)  $E_1$  and  $E_2$  kept separate with foreign key at  $E_1$  end

- (b)  $E_1$  and  $E_2$  kept separate with foreign key at  $E_2$  end
- (c)  $E_1$  and  $E_2$  kept separate with foreign key at  $E_1$  as well as  $E_2$
- (d)  $E_1$  and  $E_2$  merges into a single table with no foreign key

**Q.6** For weak entity set to be meaningful it must be part of

- (a) one to one relationship
- (b) one to many relationship
- (c) many to one relationship
- (d) All of the above

**Q.7** In an E-R diagram,  $Y$  is the dominant entity and  $X$  is a subordinate entity. Then which of the following is incorrect:

- (a) Operationally, if  $Y$  is deleted, so  $X$  is also deleted
- (b)  $X$  existence is dependent on  $Y$
- (c) Operationally, if  $X$  is deleted, so  $Y$  is also deleted
- (d) Operationally, if  $X$  is deleted,  $Y$  remains the same

**Q.8** Map the following statements with True (T)/ False (F)

$S_1$ : Participation of the weak entity set in identifying relationship must be total.

$S_2$ : Multivalued attributes in E-R diagram require separate tables when converted into relational model with satisfy 2NF.

- (a) F T (b) T F
- (c) F F (d) T T

**Answers Key:**

1. (a) 2. (a) 3. (c) 4. (b) 5. (b)
6. (b) 7. (c) 8. (d)



**Student's  
Assignments**

**Explanations**

2. (a)

Manufacturer is a name and it is at the 'one' end of any relationship. Hence it should not be an entity set it can be a attribute of Beer entity option (a) is correct.

3. (c)

The relationship is many to one and the partial participation from both entity set.

4. (b)

Student entity will have total participation with offered relation because each student takes atleast one course (i.e. 1 or more).

6. (b)

Weak entity set always have total participation in the relationship. It must be merged with the relationship entity. So the relation must be 1 : M.



7. (c)

X foreign key references to Y if "X" deleted then "Y" need not deleted.

8. (d)

Both the given statements are true.

**S<sub>1</sub>:** Participation of the weak entity set in identifying relationship should be total because primary key of weak entity set gets defined only by relating it to strong entity and its primary key value.

**S<sub>2</sub>:** Multivalued attributes in E-R diagram require separate tables along with key attribute when converted into relational model.

◆◆◆◆◆