MODULE 8

LOGICAL DATABASE DESIGN

Contents

LOGICAL DATABASE DESIGN

MOTIVATION

When a DFD is developed we have knowledge of all data elements required by an application,data dictionary lists all data elements but does not say anything about relationships between data elements. Relationships are needed to logically group data elements into related sets or tables.
Such an organization reduces data duplication, simplifies adding, deleting and updating data and simplifies retrieval of desired data
Database management systems achieve the purpose of mapping the logical database to a physical medium which is transparent to an application program.

LEARNING GOALS

At the end of this module you will learn

1.The Entity-Relationship(ER) modelling to develop a conceptual model of data.
2.How to organize data required in an application as relations
3.The need for normalizing relations
4.The various normal forms and their relevance
5.How to normalize relations to successive higher normal forms to form a relational database
6.The need for an integrated database in organizations
7.The goals of Data Base Management systems (DBMS)
8.The structure and organization of DBMS.

LEARNING UNIT 1

Entity-relationship(E-R) modelling of data elements of an application.

LOGICAL DATABASE DESIGN-INTRODUCTION

Loosely one may call organization of related data, put in a table as a "RELATION". Systematization by which related data are put in a table is called "NORMALIZATION". A method called entity-relationship analysis facilitates creation of relations.

ENTITY-RELATIONSHIP MODEL

ENTITY: Specifies distinct real world items in an application

For example: vendor, item, student, course, teachers

RELATIONSHIP: Meaningful dependencies between entities

For example: vendor supplies items
teacher teaches courses

Relationships are underlined above

ENTITY SETS

An entity set is collection of similar entities

Examples : * Set of all vendors of an organization is a vendor set

* Set of all items in a store is an item set

Every member of an entity set is described by its attributes

ATTRIBUTES

Attributes specify properties of members of entity set & also specify properties of relationships

Examples:

Entity : Vendor

Attributes : **vendor code**, vendor name, address

Relationship : supplies

Attributes : **vendor code, item code**,order no., qty. supplied,date of supply,price/unit

ENTITES AND ATTRIBUTES

Example

Entity : Teacher

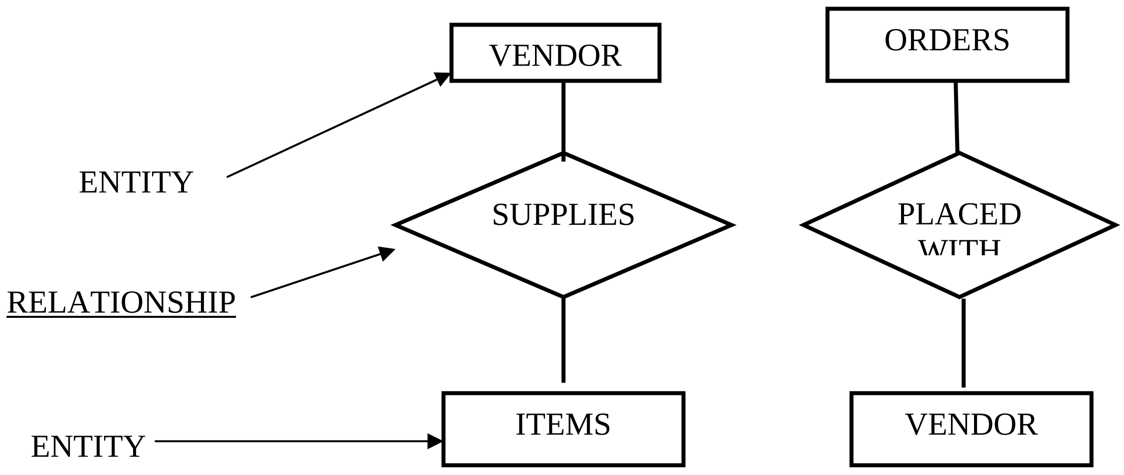
Attributes : **Teacher code**,teacher name,department,building,room no,phone no.

Relationship : Teaches

Attributes : **Teacher code,Course no**,course name,semester offered, credits, prerequisites

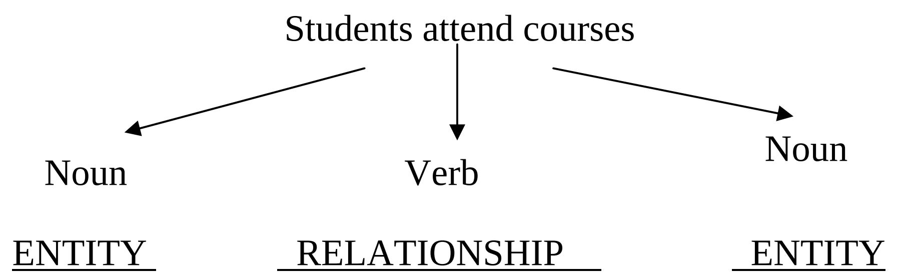
ENTITY-RELATIONSHIP DIAGRAM

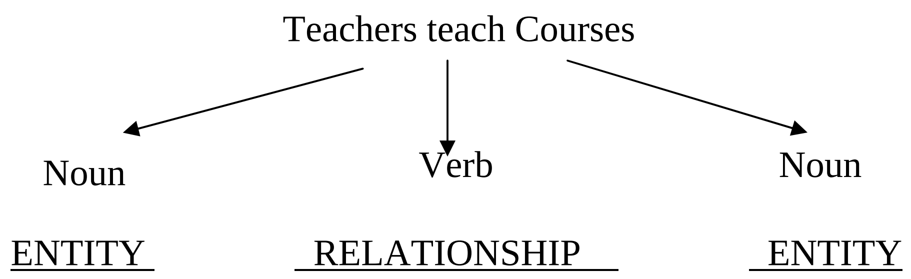
Some entities depend on one another, for example, entity vendor and entity items are related as vendors supply items. These relationships are described by entity-relationship diagrams (or ER diagrams). In an ER diagram entities are represented by rectangles and relationships by diamond shaped boxes



<u>HOW TO IDENTIFY ENTITIES AND RELATIONSHIPS</u>

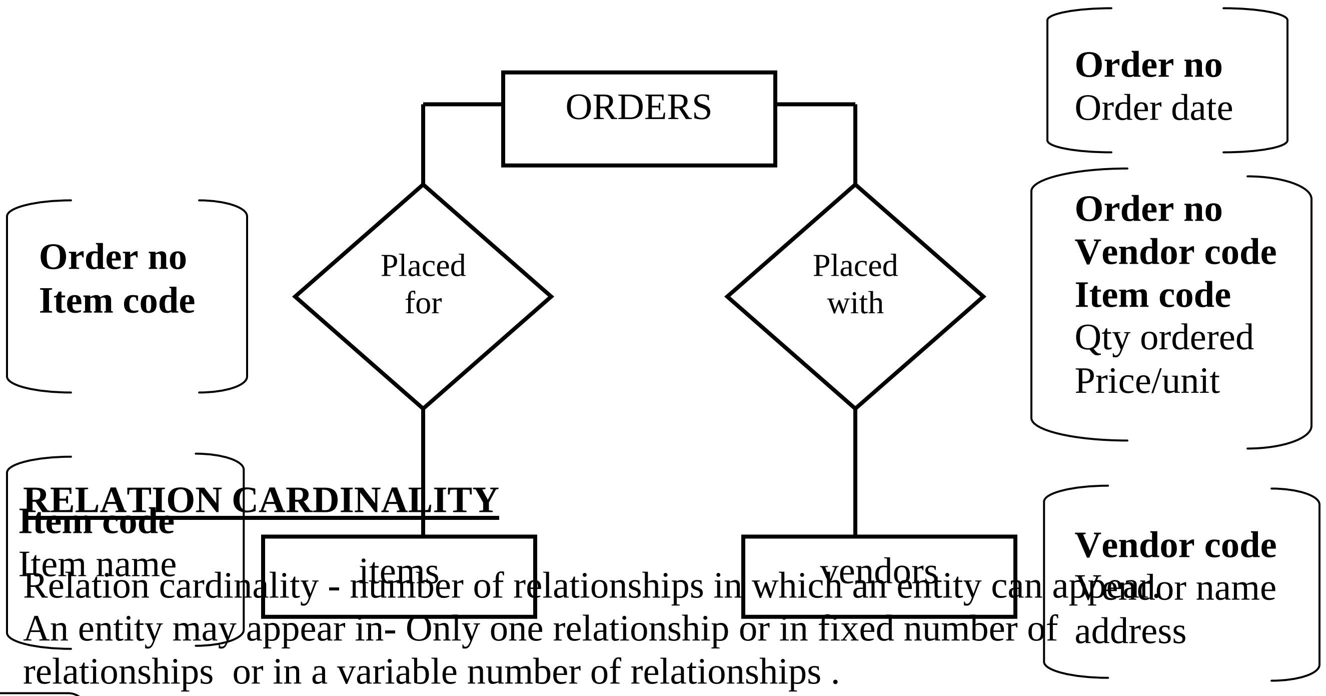
When a word statement is used to describe applications, <u>nouns</u> normally are entities and <u>verbs</u> relationships.





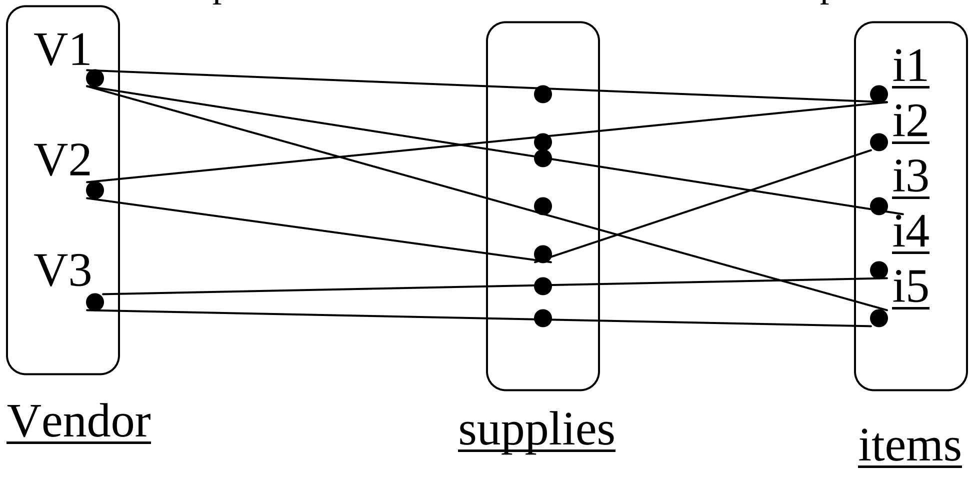
ENTITY-RELATIONSHIP DIAGRAMS

One entity may be related to many other entities by multiple relationships



RELATION CARDINALITY

Relation cardinality - number of relationships in which an entity can appear.
An entity may appear in- Only one relationship or in fixed number of relationships or in a variable number of relationships .



Vendor1 supplies items il,i3,i5

Vendor2 supplies items il and i2

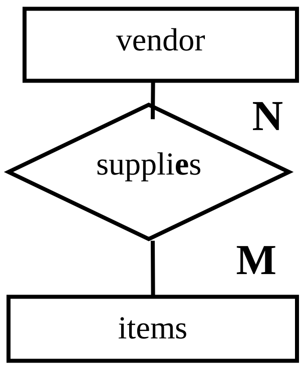
Vendor3 supplies items i4 and i5

Observe a vendor can supply many items, and also that many vendors can supply the same item

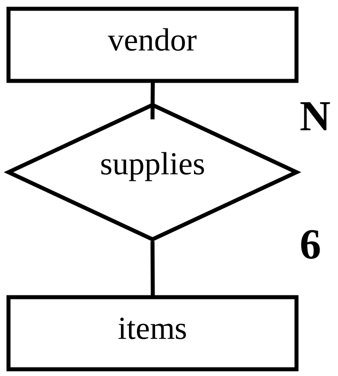
RELATION CARDINALITY REPRESENTATION

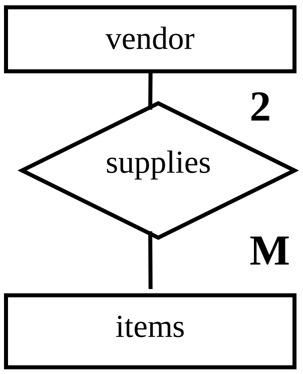
A vendor can supply upto M items

A vendor cannot supply more than 6 items
N vendors can supply a given item



N vendors can supply a given item





An item cannot be supplied by more than 2 vendors
A vendor cannot supply more than M items

WHY IS FINDING CARDINALITY NECESSARY

1. The identifier of the relationship will be composite if cardinality is N:M
2. It will be single if cardinality is 1:M
3. Will be useful later in designing data base

LEARNING UNIT 2

Organization of data as relations

RELATIONS

Entity sets and relationship sets are useful in designing data bases.
Entity - relationship sets may be put as a table of values. This is called a Relation. Relation name is entity name. A row of a relation has members of its attributes. The column headings are the names of the attributes.

EXAMPLES OF A RELATION

VENDOR CODE	VENDOR NAME	ADDRESS
1456	Ram & co	112, 1st cross Bangalore-12
1685	Gopal & sons	452,4th main, Delhi-8
1284	Sivaraj brother	368 M.G Road, Pune-8
1694	Gita ltd	495 N.S.C Road,Calicut

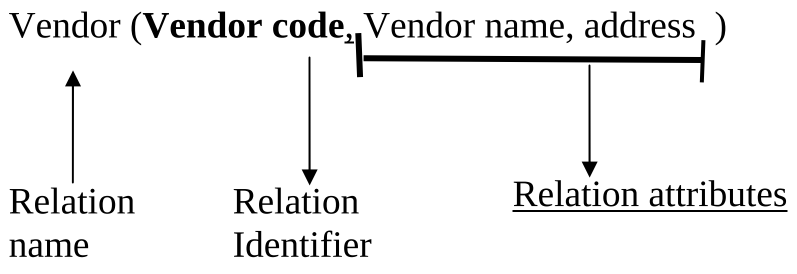
RELATION name:Vendor(same name as entity name)
RELATION ATTRIBUTES: vendor code, vendor name, address

Row of relation is called a <u>tuple.</u> In a RELATION rows and columns can be in any order. No two rows and two columns are identical.

RELATION NOTATION

Relation is an entire table

- Vendor relation:



WHY RELATION ?

Entity set can be easily stored as a flat file in a computer's storage
Sound theory of relations allows systematic design of relational data base, reduces duplication of data, tries to eliminate errors in adding, deleting, altering items in a data base and simplifies retrieval of data.

LEARNING UNIT 3

Normalization of relations

NORMALIZING RELATIONS

Normalizing is the process of restructuring relations to a form which: -

- Minimizes duplication of data in a database
- Operations of adding, deleting, modifying data in a database do not lead to inconsistent data in a database
- Retrieval of data simplified

WHY NORMALIZE ?

Relations are normalized to ensure that, collection of relations do not unnecessarily hold duplicate data. It is easy to modify a data item as it gets modified in all relations where it appears and hence no consistency is there. When data is deleted accidentally, required data does not get deleted. It also simplifies retrieval of required data.

HOW ARE RELATIONS NORMALIZED ?

UNNORMALIZED RELATION

Order no	order date	Item lines		
		Item code	Qty	Price/unit
1456	26021999	3687	52	50.40
		4627	38	60.20
		3214	20	17.50
1886	04031999	4629	45	20.25
		4627	30	60.20
1788	04111999	4627	40	60.20

1.Observe order for many items
2.Item lines has many attributes-called composite attributes
3.Each tuple has variable length
4.Difficult to store due to non-uniformity
5.Given item code difficult to find qty-ordered and hence called Unnormalized relation

FIRST NORMAL FORM

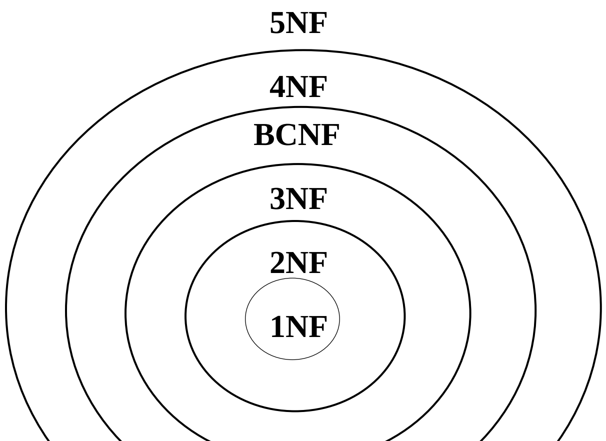
Identify the composite attributes, convert the composite attributes to individual attributes. Duplicate the common attributes as many times as lines in composite attribute. Every attribute now describes single property and not multiple properties, some data will be duplicated. Now this is called First normal form (1NF) also called flat file.

FIRST NORMAL FORM – 1NF

Order No	Order date	Item code	Qty	Price/unit
1456	26021999	3687	52	50.40
1456	26021999	4627	38	60.20
1456	26021999	3214	20	17.50
1886	04031999	4629	45	20.25
1886	04031999	4627	30	60.20
1788	04111999	4627	40	60.20

HIGHER NORMAL FORMS

First normal form is first essential step in normalization. Higher normal forms known as 2NF, 3NF, BCNF, 4NF, 5NF also exist. Each is an improvement of the preceding one. A higher normal form also satisfies requirements of a lower normal form



Higher normalization steps are based on :

- Detecting dependence between attributes
- Identifying key attributes
- Detecting multivalued dependency between attributes

FUNCTIONAL DEPENDENCY

Given X,Y as two attributes in a relation
Given X if only one value of Y corresponds to it then Y is functionally dependent on X

$$X \longrightarrow Y$$

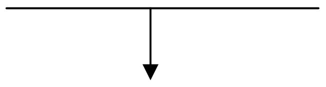
e.g. Given Item code - Item name known

Therefore Item code $\rightarrow$ Item name

Functional dependence may be based on a composite attribute

$$\begin{tikzcd} \underbrace{\mathbf{X,Z}} \arrow[r] \arrow[d] & \mathbf{Y} \\ \text{composite attribute} & \end{tikzcd}$$

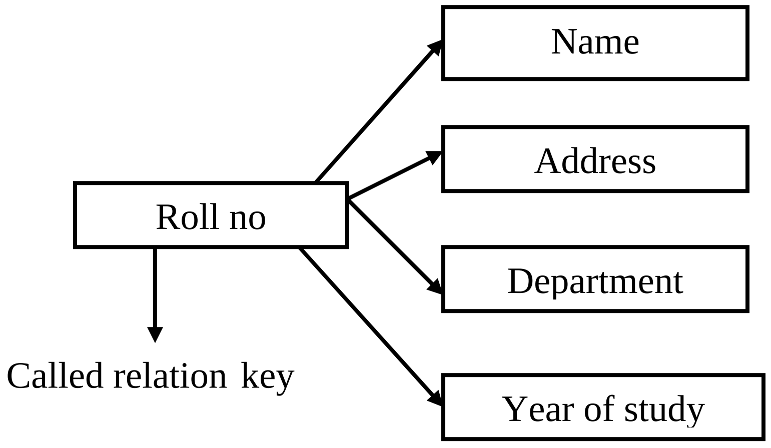
Order no. ,item code---------→ Qty , price



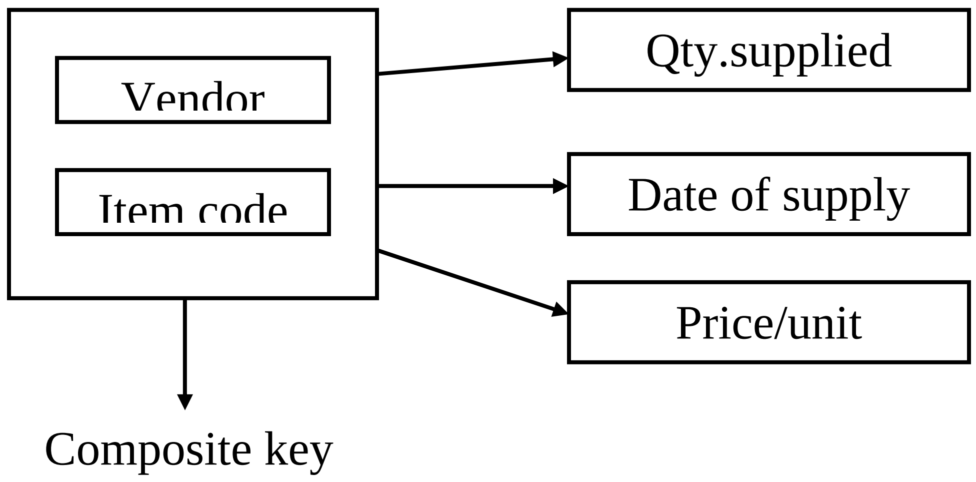
composite attribute

DEPENDENCY DIAGRAM

Student (Roll no, name, address, dept., year of study)



Roll no determines uniquely values of all other attributes in the relation, therefore it is called a key.



WHY NORMALIZE RELATIONS-REVISITED

We normalize relations to ensure the following:
While operating on data base we do not lose data or introduce inconsistencies. Insertion of new data should not force leaving blank fields for some attributes. We do not delete vital information during update. In a normalized relation it is possible to change the values of the attribute without exhaustively searching all tuples of the relation.

EXAMPLE TO SHOW NEED FOR NORMALIZATION

FIRST NORMAL FORM – 1NF

Order No	Order date	Item code	Qty	Price/unit
1456	26021999	3687	52	50.40
1456	26021999	4627	38	60.20
1456	26021999	3214	20	17.50
1886	04031999	4629	45	20.25
1886	04031999	4627	30	60.20
1788	04111999	4627	40	60.20

INSERTION : Enter new item with code 3945 and price 30.50 for which no order has been placed. Inserted tuple will have no values (i.e have to be left blank) for order no and order date

DELETION: If order no1886 is deleted the fact that item code 4629 costs 20.25 is lost

UPDATE: If price of item 4627 is changed, all instances of this item code have to be changed by exhaustive search-errors possible

IDEAL NORMALIZATION

At the end of normalization a normalized relation

- Should have no data values duplicated in rows
- Every attribute in a row must have a value
- Deletion of a row must not lead to accidental loss of information
- Adding a row should not affect other rows
- A value of an attribute in a row can be changed independent of other rows

SECOND NORMAL FORM (2NF)

A relation is in 2NF if it is in 1NF, non-key attributes are functionally dependent on key attribute and if there is a composite key then no non-key attribute is functionally depend on one part of the key.

2NF FORM

1 NF Orders Relation

Order No	Order date	Item code	Qty	Price/unit
1456	26021999	3687	52	50.40
1456	26021999	4627	38	60.20
1456	26021999	3214	20	17.50
1886	04031999	4629	45	20.25
1886	04031999	4627	30	60.20
1788	04041999	4627	40	60.20

2 NF Relations

ORDERS

Order No	Order date
1456	26021999
1886	04031999
1788	04041999

ORDER DETAILS

Order No	Item code	Qty
1456	3687	52
1456	4627	38
1456	3214	20
1886	4629	45

PRICES

Item code	Price/unit
3687	50.40
4627	60.20
3214	17.50
4629	20.25

NON KEY ATTRIBUTES WHOLLY DEPENDENT ON KEY

- Repetition of order date removed.
- If order 1886 for item 4629 is cancelled the price/unit is lost in INF as the whole tuple would be deleted.
- In 2NF item price not lost when order 1886 for item 4629 cancelled. Only row 4 in order details deleted.
- Duplication of data in a relation is not there.

<u>THIRD NORMAL FORM</u>

A Relation in 2NF may have functional dependency between some Non-key attributes. This needs further normalization as the non-keys being dependent leads to unnecessary duplication of data. Normalization to 3NF ensures that there is no functional dependency between non-key attributes.

<u>EXAMPLE</u>

Student (Roll no, name, dept, year, hostelname)

- If students in a given year are all put in one hostel then year and the hostel are functionally dependent
- Year implies hostel-hostel name unnecessarily duplicated
- If all students of year 1 are moved to another hostel many tuples need to be changed.

<u>NORMALIZATION TO 3NF</u>

Student(**Roll no**, name, dept, year)

Hostel (**year,** hostel)
This is in 3NF

<u>Example</u> :1
Employee (**empcode,**name,salary,project no,termination date of project)
*** termination date (non-key attribute)**
Dependent on project no. another non-key attribute
•Thus needs normalization
3NF relations : Employee(**empcode**,name,salary,projectno)
Project(**Projectno**_,termination date of project)

Example:2
Passenger(**Ticket code**,Passenger name,Train no,Departure time,Fare)
Train no. and departure time are non-key attributes and are functionally dependent
3NF Relations :
Passenger(**Ticket code** ,Passenger name,Train no, Fare)
Train details (**Train no**., departure time)

<u>BOYCE-CODD NORMAL FORM</u>

<u>Assume</u>
* Relation has more than 1 possible key
* Keys are composite
* Composite keys have common attribute
* Non-key attributes not dependent on one another
Thus though the relation is in 3NF, still there could be problems due to unnecessary duplication and loss of data accidentally.

<u>EXAMPLE</u>

Professor (**Prof code**, **Dept**, Head of Dept, Percent time)

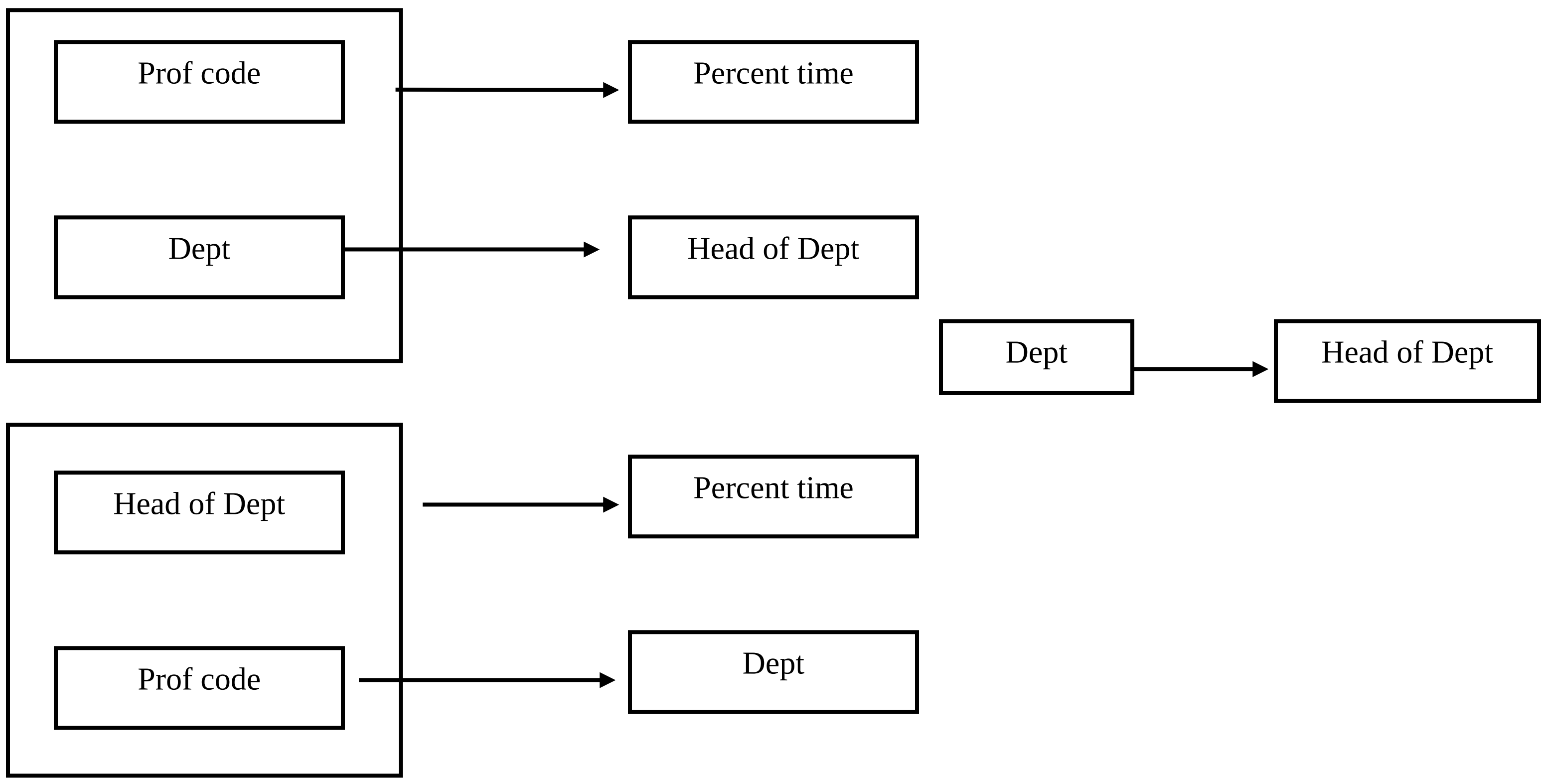
<u>RELATION</u>

Professor code	Dept	Head of dept	Percent time
P1	Physics	Ghosh	50
P1	Maths	Krishnan	50
P2	Chem	Rao	25

P2	Physics	Ghosh	75
P3	Maths	Krishnan	100
P4	Maths	Krishnan	30
P4	Physics	Ghosh	70

- Observe two possible composite keys (**Prof code, Dept**) or (**Prof code,Head of Dept**)
- Observe Head of dept name is repeated
- If professor P2 resigns the fact that Rao is Head of Chemistry is lost as lines 3 & 4 will be deleted

The dependency diagrams are:



▪Percentage time a Prof. spends in the department is dependent on Prof code and Department
▪ Head of Dept depends on department

<u>NEED FOR BCNF</u>

Observe the given relation is in 3NF as non key attributes are independent of one another and wholly dependent on key. However there are problems due to the fact that there are two possible composite keys, and attribute of on of the composite key depends on a attribute of other possible composite key

<u>NORMALIZING TO BCNF</u>

•Identify the dependent attributes in the possible composite keys
•Remove them and create a new relation

<u>EXAMPLE</u>

<u>Composite keys</u>
1.Prof code ,Dept 2. **Prof code,Head of Dept**

Dependency : Dept $\longrightarrow$ Head of dept

<u>New relations</u>
Professor (**Prof code, Dept**, Percent time)
Department (**Dept,** Head of Dept)

<u>NORMALIZED BCNF RELATIONS</u>

Professor code	**Dept**	**Percent time**
P1	Physics	50
P1	Maths	50
P2	Chem	25
P2	Physics	75
P3	Maths	100
P4	Maths	30

Dept	**Head of Dept**
Physics	Ghosh
Maths	Krishnan
Chem	Rao

FOURTH NORMAL FORM

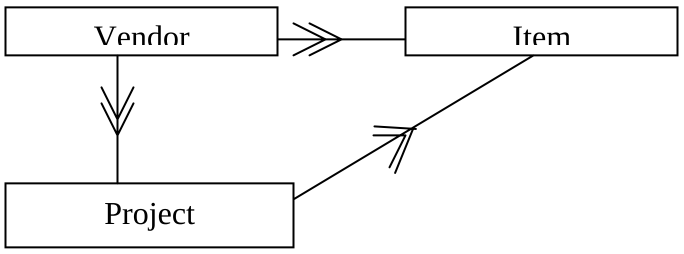
4NF is needed when there are multi-valued dependencies

- **Example :**

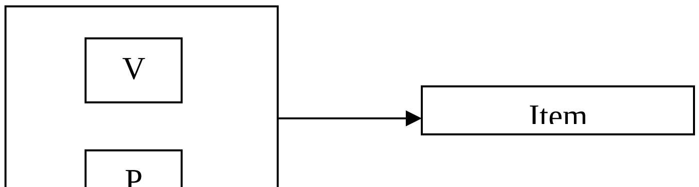
(Vendor, Project, Item) relations
Assumptions :

-A vendor capable of supplying many items to many projects
-A project needs many items
-Project may order the same item from many vendors

Vendor-Project-Item supply capability relation



$\twoheadrightarrow$ Multivalued dependency



Vendor code	Project code	Item code
VI	PI	I1
VI	PI	I2
VI	P3	I1
VI	P3	I2
V2	PI	I2
V2	PI	I3
V3	PI	I1
V3	P2	I1

Problems

•Item I1 duplicated for VI and also for V3

•If VI is to supply to project P2 but the item to be supplied is not decided there will be blank in item column

Solution:

•Split vendor-project-item relation into two relations
•Resulting relation must have not more than one independent multivalued dependency

RESULTING RELATIONS

Vendor	**Item**
VI	**I1**
VI	**I2**
V2	**I2**
V2	**I3**
V3	**I1**

Project	**Item**
P1	**I1**
P1	**I2**
P1	**I3**
P2	**I1**
P3	**I1**
P3	**I2**

Vendor Capability   Project needs

OBSERVE NO UNNECESSARY DUPLICATION

<u>NEED FOR 5NF</u>

▪In 4NF relations vendor capability to supply items and projects need for items are there.
▪They are obtained by splitting the given relation
▪Looking at relation project-item we see that project P2 requires item I1
▪From vendor item relation we see that I1 is supplied by V1.
▪This would lead us to infer that(V1,P1,I1)must be a tuple in the original relation but it is not.In other words V1 does not supply item I1 to project P2.
▪This spurious tuple has occurred because vendor V1 may not be allowed to supply item I1 to project P2
▪Similarly another spurious tuple is (V3, P3, I1)
▪We thus need a third relation which specifies the vendors who are allowed to supply to projects

Additional relation for 5NF

Vendor	Project
V1	P2
V1	P3
V2	P1
V3	P1
V3	P2

Vendors permitted for projects

The above relation is in addition to the two relations of 4NF.

LEARNING UNIT 4

Creation of logical relational database

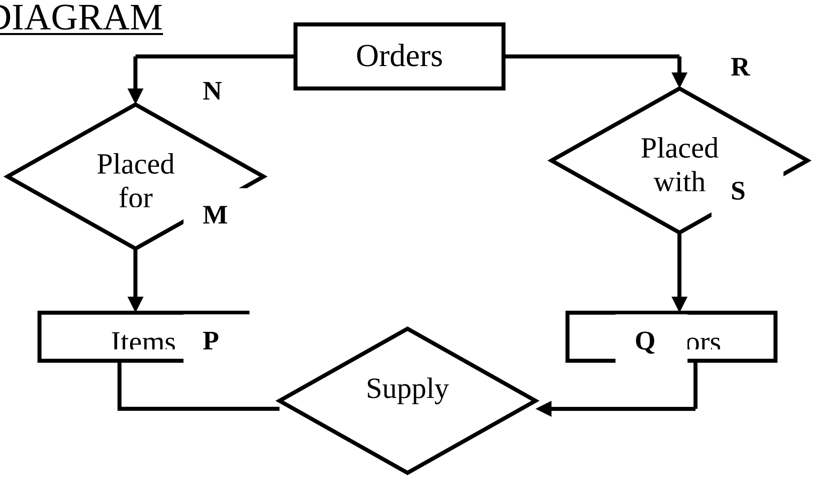
EXAMPLES OF DATA BASE DESIGN

ORDER - VENDOR - ITEMS ORDERED EXAMPLE IN CASE STUDY

Information on dependencies given :

•Orders for item placed with many vendors
•A given order no is only to one vendor
•Many items supplied against a given order no
•A vendor has capacity to supply many items but only some items maybe ordered from him at a particular time

ER - DIAGRAM



RELATIONS-UNNORMALIZED

EXAMPLES OF UNNORMALIZED RELATIONS

1.ORDERS(**Order no,**Order date)
2.ORDERS PLACED FOR(**Order no,item code,**qty ordered,delivery time allowed)
3.ORDERS PLACED WITH(**order no,vendor code,item code**)
4.VENDOR(**Vendor code**,vendor name,vendor address)
5.ITEM(**item code**,item name,price/unit)
6.SUPPLIES(**vendor code,item code,order no**,qty.supplied,date of supply)

NORMALIZATION:

Relation 1,4,5 are in 3NF and need no change
Relation 2 has a composite key,attributes of composite key not related.
Non key attributes dependent on composite key,need no change.
Relation 3: order no and item code have multivalued dependency.Relation2 already has **order no,item code** as composite key.

Relation 3 is reduced to:
7.ORDER PLACED WITH(**order no,vendor code**)

NORMALIZATION OF SUPPLIES RELATION

Consider relation 6 :
6.SUPPLIES (**vendor code, item code, order no**, qty supplied, date of supply)
•It has a composite key with three attributes
•Attributes **item code** and **order no** have multi-valued dependency as many items can be supplied in one order
• And hence need normalization to 4NF

Normalized to

8. ACTUAL SUPPLIES (**order no, item code,** qty supplied, date of supply)
9. VENDOR CAPABILITY (**vendor code, item code**)

The second relation may have items not yet ordered with a vendor but which could be supplied by vendor

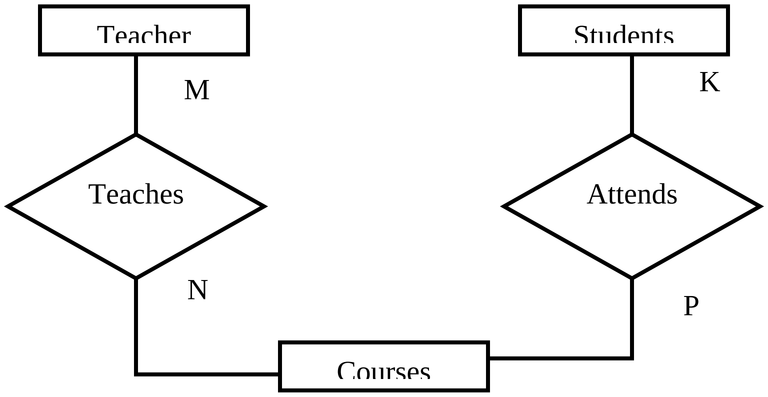
The Normalized relations are : 1,2,4,5,7,8,9

STUDENT-TEACHER-COURSES EXAMPLE

Information on dependence

- •A teacher may teach more than one course in a semester
- •A teacher belongs to only one dept.
- •A student may take many courses in a semester
- •A course may have many sections taught by different teachers

E-R Diagram



RELATION-UNNORMALIZED

1 TEACHER (**Teacher code,**teacher name, address, rank, dept)
2 TEACHER_COURSES (**Teacher code,Course no**,no of students, section no)
3 COURSE (**Course no , semester taught** ,Course name, credits)
4 STUDENT (**Student no,** student name, dept, year)
5 STUDENT COURSES (**Student no, Course no**, semester no)

a)Relations 1,3,4 in 3NF
b)Relations 2 and 5 have multi-attribute key which has multi-valued dependency but do not need normalization
c)However information on which teacher teaches a given student a specified course cannot be found from relations 1 to 5
Therefore Add relation
6TEACHER_STUDENT (**Teacher code, Student no, Course no**)

THIS SET IS NORMALIZED

CONCLUSIONS

▪ We have seen how data relevant to applications are organized logically into set of relations

▪ The process of normalization depends on the semantics, i.e, meanings of data and an understanding of how various data elements are related

▪It is thus a human intensive activity-it cannot be automated

▪ In most problems in practice one is satisfied with 3NF.Higher normal forms are theoretically important and in some cases becomes essential.

▪ There is a mathematical theory which underpins the idea of relations and normalization giving it a sound basis. We have not discussed it in this module.

▪ A full fledged course in Data Base will describe in detail the mathematical basis and methods of querying a database

LEARNING UNIT 5

Objectives of database management system(DBMS)

PROBLEMS WITH FILE BASED SYSTEMS

If programs and files independently developed for each application in the organization, it leads to the following problems

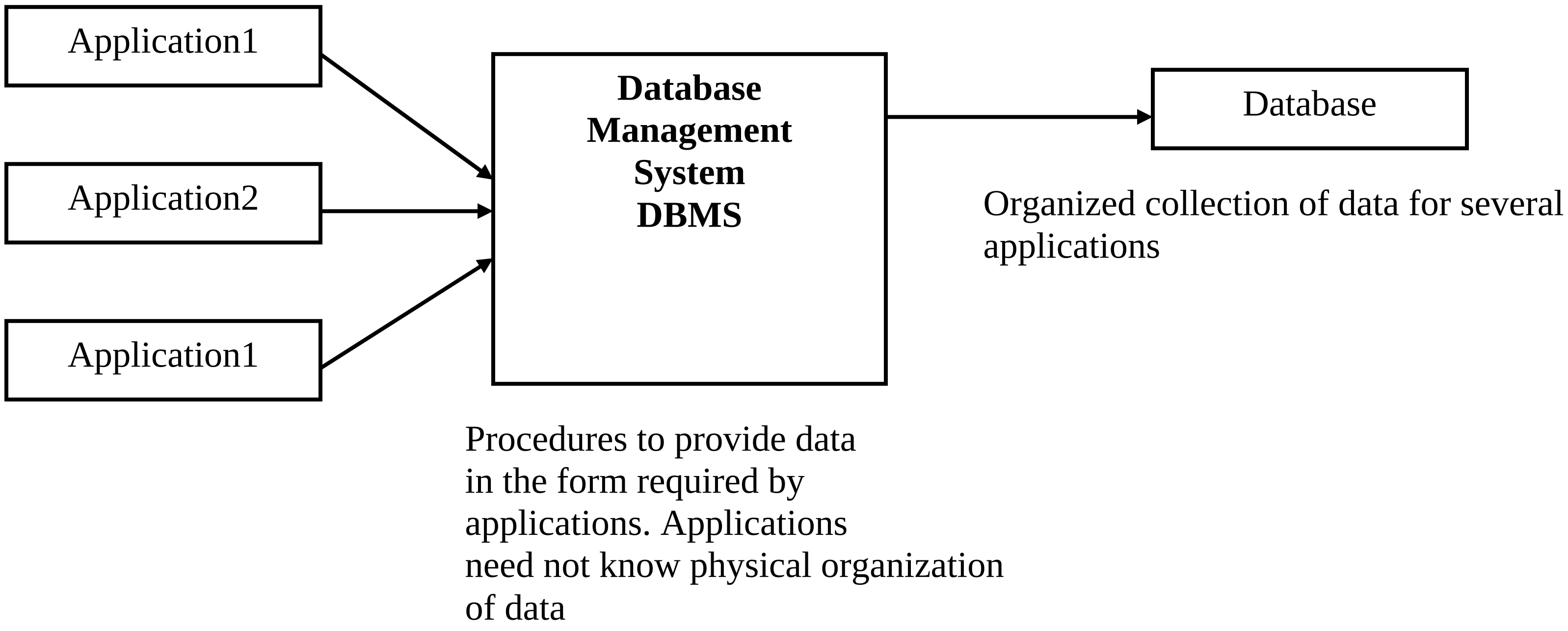
- DATA REDUNDANCY-Some data may be duplicated in many files.
  e.g.: Address of customer
- LACK OF DATA INTEGRITY- Duplicated data may be different in different files (New address in one file and old address in another file)
- DATA AVAILABILITY- May require search of number of files to access a specified data
- CONTROL BY MANAGEMENT-Difficult as data scattered across files.
  All files should be accessed to find specified data

Aim of data base management systems is to reduce above problems

DATABASE AND DATABASE MANAGEMENT SYSTEM

DATA BASE is a collection of related data necessary to manage an organization (Excludes transient data).It models a data resource and is independent of applications

DATA BASE MANAGEMENT-is a set of procedures that manage the database and provides access to the database in the form required by the application program


Application1
Application2
Application1
Database Management System DBMS
Database
Organized collection of data for several applications
Procedures to provide data in the form required by applications. Applications need not know physical organization of data

OBJECTIVES OF A DATABASE MANAGEMENT SYSTEM

▪Data is an organizational resource. It should be collected, validated, protected, logically organized and stored with controlled redundancy. This is the organizational database
▪ One of the main objectives of DBMS is to facilitate sharing of a database by current and future applications
▪ DBMS must ensure data independence for programs
▪Data independence to be provided to application programs
▪Data independence allows
    -Change of database without affecting application programs
    -Change of hardware or system software without affecting application programs
    -Sharing of data by different applications by providing views appropriate for the application
▪ Control of Redundancy - Avoid unnecessary duplication
▪ Relations between data items specified
▪ Data integrity - Preserve consistency of data values
▪ Data Security - Permit access to data to authorized users only

LEARNING UNIT 6

Overview of DBMS.

OVERVIEW OF DBMS

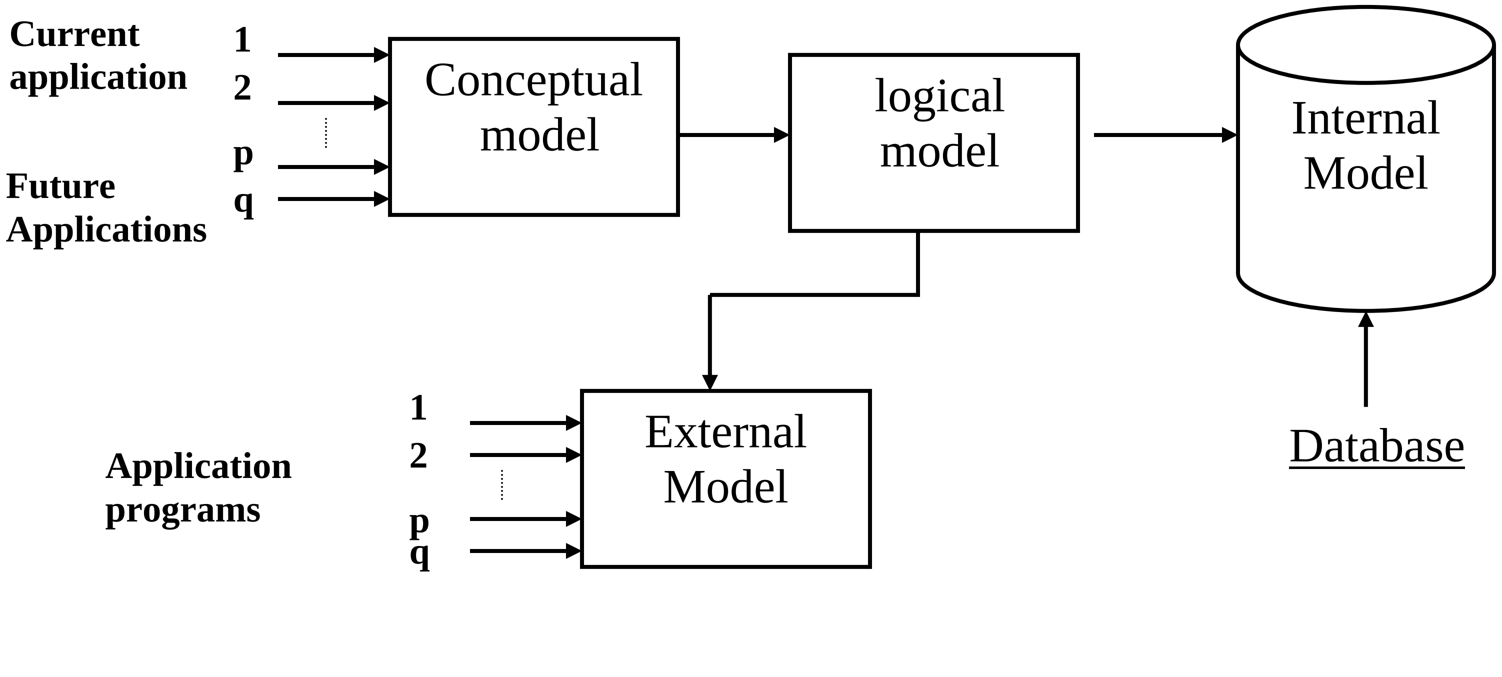
▪Data needs of current and possible future applications determined
▪Using E-R data modelling conceptual data model found
▪Converted to relations if Relational DBMS used
- called logical data model

▪Application programmers access subset of logical data model
- called external data model

•Logical model mapped to physical model for storage in disk store
- called internal model

•External data model kept invariant

VARIOUS TERMS USED IN DESCRIBING DBMS



COMPONENTS OF DBMS

- ▪Data Definition Languages (DDL) to define conceptual, logical and external models
- ▪Data manipulation and query language called Structured Query Language (SQL)
- ▪Features to implement security
- ▪Checkpointing and roll back recovery procedures
- ▪Record of accesses. Audit trail of changes

DATABASE ADMINISTRATOR

Database Administrator’s responsibilities are controlling of data recourse, to ensure integrity, security and privacy, maintenance of data dictionary, coordination of development and maintenance of data base and determining access rights

CHARACTERSTICS OF A GOOD DATA BASE DESIGN

- ▪Satisfy current and future needs of organization

- ▪Cater to unanticipated user requirements in the best possible way

- ▪Expandable with growth and changes in organization

- ▪Easy to change when hardware and software change

- ▪Ensure data security by allowing only authorized persons to access and modify database

REFERENCES

1. Most of the material in this module has been adapted from the book "Analysis and Design of Information Systems", 2nd Edition, by V.Rajaraman. Chapter 10 Logical Data Base Design, pp.130 to 153. Chapter 12 , Data Base Management Systems, pp. 171 –1719.

2. Good treatment of E-R Modelling and Normalization may be found in the book "Introduction to Systems analysis and Design", by I.T.Hawryszkiewycz, Prentice Hall of India, 1989.

3. For those interested in detailed study of Data Base Design and DBMS, the following books are standard texts:
   1, R.Elmasri and S.B.Navathe, Fundamentals of Data Base Systems, 4th Edition, Pearson Education Asia, New Delhi, 2004.
   2. T.Connolly and C.Begg, Dta Base Systems, 3rd Edition, Pearson Education Asia, New Delhi, 2003.