BACHELOR OF COMPUTER APPLICATIONS

SEMESTER 3

DCA2102

DATABASE MANAGEMENT SYSTEM

Unit 10

Normalization

Table of Contents

1. INTRODUCTION

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space, and an increase in the total size of the data stored. Relations are normalized so that when relations in a database are tobe altered during the lifetime of the database, we do not lose information or introduce inconsistencies. The type of alterations normally needed for relations are:

- **Insertion** of new data values to a relation. This should be possiblewithout being forced to leave blank fields for some attributes.
- **Deletion** of a tuple, namely, a row of a relation. This should be possiblewithout losing vital information unknowingly.
- **Updating** or changing a value of an attribute in a tuple. This should bepossible without exhaustively searching all the tuples in the relation.

1.1 Objectives

After going through this unit, the reader should be able to:

- ❖ *Discuss the different types of anomalies in a database.*
- ❖ *State the functional dependency.*
- ❖ *List the different forms of normalization.*
- ❖ *Differentiate among different types of normalization.*

2. FUNCTIONAL DEPENDENCY

As the concept of dependency is very important, it is essential that we first understand it well and then proceed to the idea of normalization. There is nofool-proof algorithmic method of identifying dependency. We have to use our commonsense and judgment to specify dependencies.

Let X and Y be the two attributes of a relation. Given the value of X, if there is only one value of Y corresponding to it, then Y is said to be functionally dependent on X. This is indicated by the notation:

$$X \rightarrow Y$$

For example, given the value of the item code, there is only one value of the item name for it. Thus item name is functionally dependent on the item code. This is shown as:

Item code → item name

Similarly in Table 10.1, given an order number, the date of the order is known.
Thus: Order no➔ Order date

Functional dependency may also be based on a composite attribute. Forexample, if we write

$$X, Z \rightarrow Y$$

It means that there is only one value of Y corresponding to given values ofX, Z. In other words, Y is functionally dependent on the composite X, Z. In Table 10.1 mentioned below, for example, Order no., and Item code together determine Qty. and Price.

Thus :

Order no., Item code → Qty., Price

As another example, consider the relation

Student (Roll no, Name, Address, Dept., Year of study)

Table 10.1: Normalized Form of the Relation

Order no.	Order date	Item code	Quantity	Price/unit
1456	260289	3687	52	50.40
1456	260289	4627	38	60.20
1456	260289	3214	20	17.50
1886	040389	4629	45	20.25
1886	040389	4627	30	60.20
1788	040489	4627	40	60.20

In this relation, Name is functionally dependent on Roll no. In fact, given thevalue of Roll no., the values of all other attributes can be uniquely

determined. Name and Department are not functionally dependent, because,given the name of a student, one cannot find his department uniquely. This is due to the fact that there may be more than one student with the same name. Name in this case is not a key. Department and Year of study are notfunctionally dependent, as Year of study pertains to a student, whereas Department is an independent attribute. The functional dependency in this relation is shown in the following figure as a dependency diagram. Such dependency diagrams shown in figure 10.1 are very useful in normalization.

Relation Key: Consider the relation of Table 10.1. Given the Vendor code, the Vendor name and Address are uniquely determined. Thus Vendor code is the relation key. Given a relation, if the value of an attribute X uniquely determines the value of all other attributes in a row, then X is said to be the key of that relation. Sometimes more than one attribute is needed to uniquely determine other attributes in a relation row. In that case, such a set of attributes is the key.

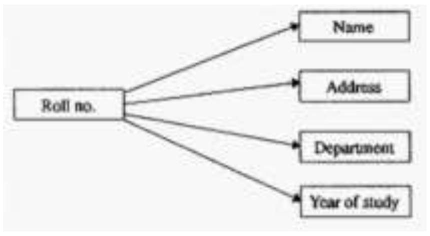

Fig 10.1: Dependency diagram for the relation "Student"

In table 10.1, Order no. and Item code together form the key. In the relation "Supplies" (Vendor code, Item code, Qty supplied, Date of supply, Price/unit) Vendor code and Item code together form the key. This dependency is shown in the following diagram (figure 10.2).

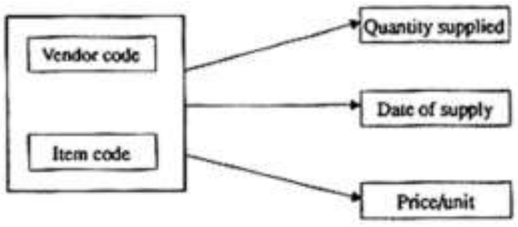

Fig 10.2: Dependency diagram for the relation "Supplies"

Observe that in the figure the fact that Vendor code and Item code together form a composite key is clearly shown by enclosing them together in a rectangle.

SELF ASSESSMENT QUESTIONS – 1

1. Let X and Y be the two attributes of a relation, The Functional Dependency can be written as_____________.
2. Functional dependency may also be based on a______attribute.

3. ANOMALIES IN A DATABASE

Consider the following relation scheme pertaining to the information about a student maintained by a university:

STDINF(Name, Course, Phone_No, Major, Prof, Grade)

Table 10.2 shows some tuples of a relation on the relation scheme STDINF(Name, Course, Phone_No, Major, Prof, Grade). The functionaldependencies among its attributes are shown in Figure 10.3. The key of the relation is Name Course and the relation has, in addition, the following functional dependencies {Name→ Phone_No, Name→ Major, Name Course→ Grade, Course→ Prof }.

Table 10.2: Student Data Representation in Relation STDINF

Name	Course	Phone_No	Major	Prof	Grade
Jones	353	237-4539	Comp Sci	Smith	A
Ng	329	427-7390	Chemistry	Turner	B
Jones	328	237-4539	Comp-Sci	Clark	B
Martin	456	388-5183	Physics	James	A
Dulles	293	371-6259	Decision Sci	Cook	C
Duke	491	823-7293	Mathematics	Lamb	B
Duke	356	823-7293	Mathematics	Bond	in prog
Jones	492	237-4539	Comp Sci	Cross	in prog
Baxter	379	839-0827	English	Broes	C

Here the attribute Phone_No, which is not in any key of the relation scheme STDINF, is not functionally dependent on the whole key, but only one part ofthe key, namely, the attribute Name. Similarly, the attributes Major and Prof,which are not in any key of the relation scheme STDINF either, are fully functionally dependent on the attributes Name and Course, respectively. Thus the determinants of these functional dependencies are again not the entire key, but only part of the key of the relation. Only the attribute Grade isfully functionally dependent on the key **Name Course**.

The relation scheme **STDINF** can lead to several undesirable problems: **Redundancy:** The aim of the database system is to reduce redundancy,meaning that information is to be stored only once. Storing information several times leads to the waste of storage space and an increase in the total size of the data stored.

Updates to the database with such redundancies have the potential ofbecoming inconsistent as explained below. In the relation of table 10.2, the Major and Phone_No. of a student are stored several times in the database:once for each course that is or was taken by a student.

Update Anomalies: Multiple copies of the same fact may lead to update anomalies or inconsistencies when an update is made, and only some of the multiple copies are updated. Thus, a change in the Phone_No. of Jones must be made, for consistency, in all tuples pertaining to the student Jones. If one of the three tuples of Figure 10.3 is not changed to reflect the new Phone_No. of Jones, there will be an inconsistency in the data.

Insertion Anomalies: If this is the only relation in the database showing theassociation between a faculty member and the course he or she teaches, the fact that a given professor is teaching a given course cannot be entered into the database unless a student is registered in the course. Also, if another relation also establishes a relationship between a course and a professor, who teaches that course, the information stored in these relations has to be consistent.

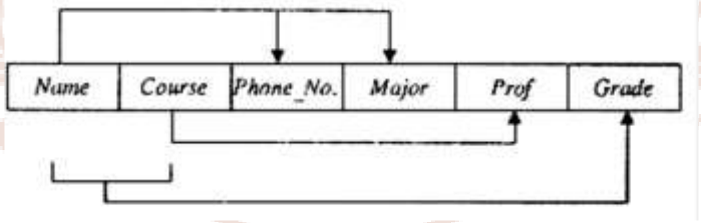

Fig 10.3: Function dependencies in STDINF

Deletion Anomalies: If the only student registered in a given course discontinues the course, the information as to which professor is offering thecourse will be lost, if this is the only relation in the database showing the association between a faculty member and the course she or he teaches. If another relation in the database also establishes the relationship between a course and a professor, who teaches that course, the deletion of the last tuple in STDINF for a given course will not cause the information about the course's teacher to be lost.

The problems of database inconsistency and redundancy of data are similarto the problems that exist in the hierarchical and network models. These problems are addressed in the network model by the introduction of virtual fields and in the hierarchical model by the introduction of virtual records. In the relational model, the above problems can be remedied by decomposition. We define decomposition as follows:

Definition: Decomposition

The decomposition of a relation scheme R = (A_1 ,A_2,...,An) is its replacement by a set of relation schemes {R_1,R_2,...,Rm} such that $R_1 \leq R$ for $1 \leq i \leq m$ and $R_1 \cup R_2 \cup R_m = R$.

A relation scheme R can be decomposed into a collection of relation schemes { R_1,R_2,R_3..., R_m } to eliminate some of the anomalities contained inthe original relation R. Here the relation schemes R1 ($1 \leq i \leq m$) are subsets of R and the intersection of $R_1 \cap R_j$ for i≠ j need not be empty. Furthermore, he union of R_j($1 \leq i \leq m$) is equal to R, i.e. R=R_1R_2... R_m.

The problems in the relation scheme STDINF can be resolved if we replace it with the following relation schemes:

STUDENT _ INFO (Name,Phone_No,Major)

TRANSCRIPT (Name,Course,Grade)

TEACHER (Course, Prof)

The first relation scheme gives the phone number and the major of each student, and such information will be stored only once for each student. Anychange in the phone number will thus require a change in only one tuple of this relation.

The second relation scheme stores the grade of each student in each course that the student is or was enrolled in. The third relation scheme records the teacher of each course. One of the disadvantages of replacing the original relation scheme STDINF with the three relation schemes is that the retrieval of certain information requires a natural join operation to be performed. For instance, to find the majors of a student who obtained a grade of A in course 353 requires a join to be performed: (STUDENT_INFO |x| TRANSCRIPT). The same information could be derived from the original relation STDINF by selection and projection.

When we replace the original scheme STDINF with the relation schemes STUDENT_INFO, TRANSCRIPT, and TEACHER, the consistency and referential integrity constraints have to be enforced. The referential integrity enforcement implies that if a tuple in the relation TRANSCRIPT exists, such as (Jones, 353, in prog), a tuple must exist in STUDENT_INFO with

Name =Jones, and furthermore, a tuple must exist in STUDENT_INFO with Course = 353. The attribute Name, which forms part of the key of the relation TRANSCRIPT, is a key of the relation STUDENT_INFO. Such an attribute (or a group of attributes), which establishes a relationship between specific tuples (of the same or two distinct relations), is called a foreign key. Notice that the attribute Course in relation to TRANSCRIPT is also a foreign key since it is a key of the relation TEACHER.

Note that the decomposition of STDINF into the relation schemes STUDENT (Name, Phone_No, Major, Grade) and COURSE (Course, Prof.) is a bad decomposition for the following reasons:

1. Redundancy and update anomaly, because the data for the attributesPhone_no and Major are repeated.
2. Loss of information, because we lose the fact that a student has a givengrade in a particular list.

Self-Assessment Questions – 2

3. The aim of the database system is to reduce redundancy, meaningthat information is to be stored________.
4. Multiple copies of the same fact may lead to update_____________.
5. In the relational model, the problem of redundancy and inconsistencycan be remedied by __________.

4. PROPERTIES OF NORMALIZED RELATIONS

Ideal relations after normalization should have the following properties so that the problems mentioned above do not occur for relations in the (ideal) normalized form:

1. No data value should be duplicated in different rows unnecessarily.
2. A value must be specified (and required) for every attribute in a row.
3. Each relation should be self-contained. In other words, if a row from arelation is deleted, important information should not be accidentally lost.
4. When a row is added to a relation, other relations in the databaseshould not be affected.
5. A value of an attribute in a tuple may be changed independently of othertuples in the relation and other relations.

The idea of normalizing relations to higher and higher normal forms is toattain the goal of having a set of ideal relations meeting the above criteria.

Self-Assessment Questions – 3

6. According to Properties of normalized relation, no data value should be__________ in different rows unnecessarily.
7. Each relation should be ____________

5. FIRST NORMALIZATION

The relation shown in table 10.1 is said to be in the **First Normal Form**, abbreviated as 1NF. This form is also called a **flat-file**. There are nocomposite attributes, and every attribute is single and describes one property. How do we convert an un-normalized table into the first normal form? Consider the following example:

Table 10.3: Department

Department_ID	Department_Name	Location
1	Production	Delhi, Kolkata
2	Sales	Mumbai
3	Marketing	Chennai
4	Research	Goa, Gurugram

Table 10.3 is not in the first normal form because the Location column contains multiple values. For example, the first row includes values "Delhi" and"Kolkata". To bring this table to its first normal form, we split the table into two tables and now we have the resulting tables:

Table 10.3a: Department_Name

Department_ID	Department_Name
1	Production
2	Sales
3	Marketing
4	Research

Table 10.3b : Department_Location

Department_ID	Location
1	Delhi
1	Kolkata
2	Mumbai
3	Chennai
4	Goa
4	Gurugram

Now the first normal form is satisfied, as the columns on each table all hold just one value. Converting a relation to the 1NF form is the first essential step in normalization. There are successive higher normal forms known as 2NF, 3NF, BNCF, 4NF, and 5NF. Each form is an improvement over the earlier form. In other words, 2NF is an improvement on 1NF, 3NF is an improvement on 2NF, and so on. A higher normal form relation is a subsetof a lower normal form as shown in the following figure 10.4. The higher normalization steps are based on three important concepts:

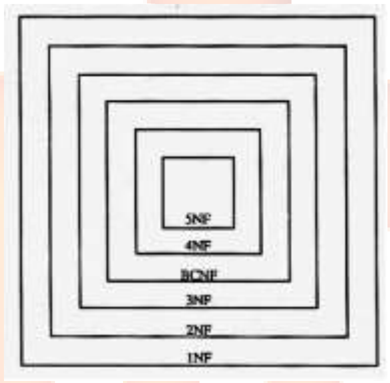

Fig 10.4: Illustration of successive normal forms of a relation

1. Dependencies among attributes in a relation
2. Identification of an attribute or a set of attributes as the key of a relation
3. Multivalued dependency between attributes.

SELF ASSESSMENT QUESTIONS – 4

8. First Normal Form, is also called a_______________file.
9. A higher normal; form relation is a subset of ___________normal form.

6. SECOND NORMAL FORM RELATION

We will now define a relation in the Second Normal Form (2NF). A relationis said to be in 2NF if it is in 1NF, and non-key attributes are functionally dependent on the key attribute(s). Further, if the key has more than one attribute then no non-key attributes should be functionally dependent upon apart of the key attributes. Consider, for example, the relation given in the table

This relation is in 1NF. The key is (Order no., Item code). The dependency diagram for attributes of this relation is shown in figure 10.5. The non-key attribute Price/Unit is functionally dependent on the Item code which is part of the relation key. Also, the non-key attribute Order date is functionally dependent on Order no. which is a part of the relation key. Thusthe relation is not in 2NF. It can be transformed to 2NF by splitting it into three relations as shown in table 10.4.

In table 10.4 the relation Orders has **Order no**. as the key. The relation **Order details** have the composite key **Order no**. and **Item code**.

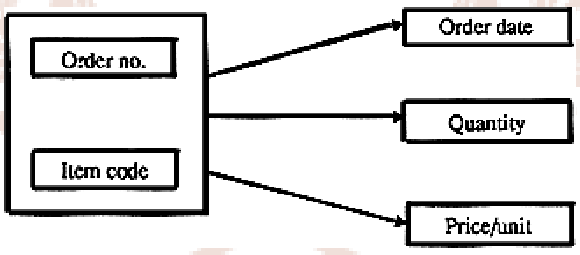

Fig 10.5: Dependency diagram for the relation given in a table

In both relations, the non-key attributes are functionally dependent on the whole key. Observe that by transforming to 2NF relations the repetition of the Order date (table 10.1) has been removed. Further, if an order for an item is cancelled, the price of an item is not lost. For example, if Order no.**1886** for item code **4629** is cancelled in table 10.1, then the fourth row will beremoved, and the price of the item is lost. In table 10.4 only the fourth row oftable 10.4(b) is omitted. The item price is not lost as it is available in table 10.4(c). The date of the order is also not lost as it is in table 10.4(a).

Table 10.4: Splitting of Relation given in table 10.1 into 2NF Relations

a) Orders

Order no.	***Order date***
1456	260289
1886	040389
1788	040489

b) Order Details

Order no.	***Item Code***	**Qty.**
1456	3687	52
1456	4627	38
1456	3214	20
1886	4629	45
1886	4627	30
1788	4627	40

c) Prices

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

These relations in 2NF form meet all the "ideal" conditions specified. Observe that the three relations obtained are self-contained. There is no duplication of data within a relation.

Self-Assessment Questions – 5

10. A relation is said to be in 2NF if it is in________and non-keyattributes are functionally dependent on the key attribute(s).
11. If the key has more than one attribute then no_________attributes should be functionally dependent upon a part of the key attributes.

7. THIRD NORMAL FORM

A Third Normal Form of normalization will be needed where all attributes in a relation tuple are not functionally dependent only on the key attribute. If two non-key attributes are functionally dependent, then there will be no unnecessary duplication of data. Consider the relation given in table 10.5. Here, Roll no. is the key, and all the other attributes are functionally dependent on it.

Table 10.5: A 2NF Form Relation

Roll no.	Name	Department	Year	Hostel name
1784	Raman	Physics	1	Ganga
1648	Krishnan	Chemistry	1	Ganga
1768	Gopalan	Mathematics	2	Kaveri
1848	Raja	Botany	2	Kaveri
1682	Maya	Geology	3	Krishna
1485	Singh	Zoology	4	Godavari

Thus it is in 2NF. If it is known that in the college all first-year students are accommodated in the Ganga hostel, all second-year students in Kaveri, all third-year students in Krishna, and all fourth-year students in the Godavari, then the non-key attribute Hostel name is dependent on the non-key attribute Year. This dependency is shown in figure 10.6.

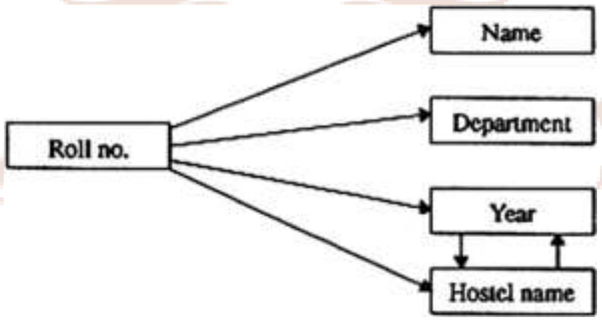

Fig 10.6: Dependency diagram for the relation

Observe that given the year of the student, his hostel is known and vice versa. The dependency of the hostel on year leads to duplication of data as is evident from table 10.5. If it is decided to ask all first-year students to move to Kaveri hostel, and all second-year students to Ganga hostel, this change should be made in many places in table 10.5. Also, when a student's year ofstudy changes, his hostel change also should be noted in table 10.5. This is undesirable. Table 10.5 is said to be in 3NF if it is in 2NF and no non-key attribute is functionally dependent on any other non-key attribute. Table 10.5is thus not in 3NF. To transform it to 3NF, we should introduce another relation that includes the functionally related non-key attributes.

Table 10.6: Conversion of table 10.5 into two 3NF relations

Roll no.	**Name**	**Department**	**Year**
1784	Raman	Physics	1
1648	Krishnan	Chemistry	1
1768	Gopalan	Mathematics	2
1848	Raja	Botany	2
1682	Maya	Geology	3
1485	Singh	Zoology	4

Year	**Hostel name**
1	Ganga
2	Kaveri
3	Krishna
4	Godavari

It should be stressed again that dependency between attributes is a semantic property and has to be stated in the problem specification. In this example, the dependency between Year and Hostel is clearly stated. In case hostel allocated to students does not depend on their year in college, then table 10.5 is already in 3NF.

Let us consider another example of a relation. The relation Employee is given below and its dependency diagram in figure 10.7.

Employee (Employee code, Employee name, Dept., Salary, Project no., Termination date of project).

As can be seen from the figure, the termination date of a project is dependent on Project no. Thus this relation is not in 3NF. The 3NF relations are:

Employee (**Employee code,** Employee name, Salary, Project no.) Project (**Project no**. Termination date)

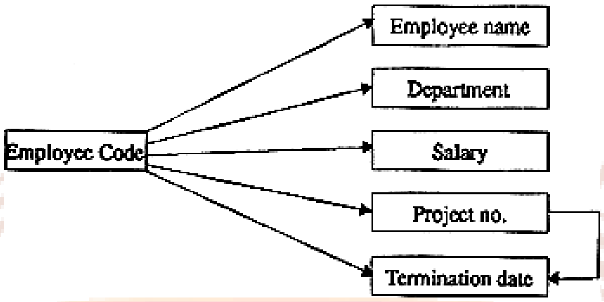

Fig 10.7: Dependency diagram of employee relation

SELF ASSESSMENT QUESTIONS – 6

12. A__________Form normalization will be needed where all attributes ina relation tuple are not functionally dependent only on the key attribute.
13. If two non-key attributes are functionally__________, then there will be no unnecessary duplication of data.
14. In 3 NF dependency between attributes is a______property and hasto be stated in the problem specification.

8. BOYCE-CODD NORMAL FORM (BCNF)

Assume that a relation has more than one possible key. Assume further thatthe composite keys have a common attribute. If an attribute of a composite key is dependent on an attribute of the other composite key, a normalizationcalled BCNF is needed. Consider, as an example, the relation **Professor**:

Professor (**Professor code**, Dept., Head of Dept., Parent time)

It is assumed that

1. A professor can work in more than one department
2. The percentage of the time he spends in each department is given.
3. Each department has only one Head of Department.

The relationship diagram for the above relation is given in figure 10.8. Table 10.7 gives the relation attributes. The two possible composite keys are Professor Code and Dept. or Professor Code and Head of Dept. Observe that department as well as Head of Dept. are not non-key attributes. They are a part of a composite key.

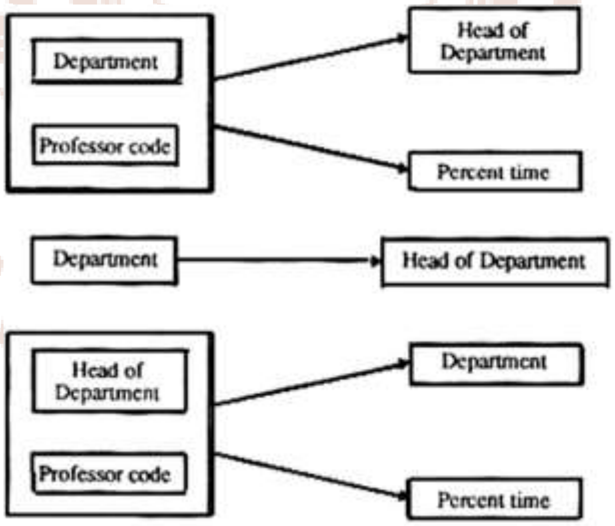

Fig 10.8: Dependency diagram of Professor relation Table 10.7: Normalization of Relation "Professor"

Table 10.7: Normalization of Relation "Professor"

Professor Code	Department	Head of Dept.	Parent
P1	Physics	Ghosh	50
P1	Mathematics	Krishnan	50
P2	Chemistry	Rao	25
P2	Physics	Ghosh	75
P3	Mathematics	Krishnan	100

The relation given in table 10.7 is 3NF. Observe, however, that the namesof Dept. and Head of Dept. are duplicated. Further, if Professor P2 resigns, rows 3 and 4 are deleted. We lose the information that Rao is the Head of the Department of Chemistry.

The normalization of the relation is done by creating new relation for Dept.and Head of Dept. and deleting Head of Dept. from Professor Relation. The normalized relations are shown in the following table 10.8 and the dependency diagrams for these new relations are in figure 10.8.

Table 10.8: Normalized Professor Relation in BCNF

a)

Professor Code	Department	Percent time
P1	Physics	50
P1	Mathematics	50
P2	Chemistry	25
P2	Physics	75
P3	Mathematics	100

b)

Department	Head of Dept.
Physics	Ghosh
Mathematics	Krishnan
Chemistry	Rao

The dependency diagram gives the important clue to this normalization stepas is clear from figures 10.8 and 10.9.

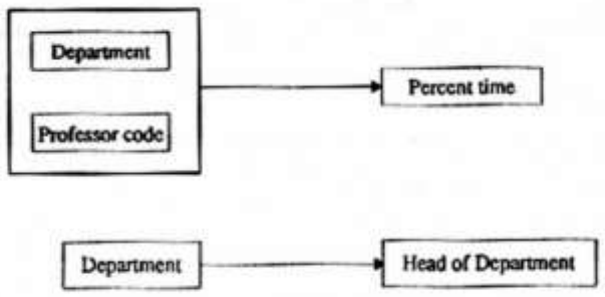

Fig 10.9: Dependency diagram of Professor relation

SELF ASSESSMENT QUESTIONS – 7

15. Boyce-Codd Normal Form is acronym for___________.
16. Removing more than one independent multivalued dependency from arelation by splitting relation is called______________.

9. FOURTH AND FIFTH NORMAL FORM

When attributes in relation have a multi-valued dependency, further Normalisation to 4NF and 5NF required. We will illustrate this with an example. Consider a vendor supplying many items to many projects in an organization. The following are the assumptions:

1. A vendor is capable of supplying many items.
2. A project uses many items.
3. A vendor supplies many projects.
4. An item may be supplied by many vendors.

Table 10.9 gives the relation for this problem and figure 10.10 the dependency diagram(s).

Table 10.9: Vendor-supply-projects Relation

Vendor Code	Item code	Project no.1
V1	I1	P1
V1	I2	P1
V1	I1	P3
V1	I2	P3
V2	I2	P1
V2	I3	P1
V3	I1	P2
V3	I1	P2

The relation given in table 10.9 has a number of problems. For example:

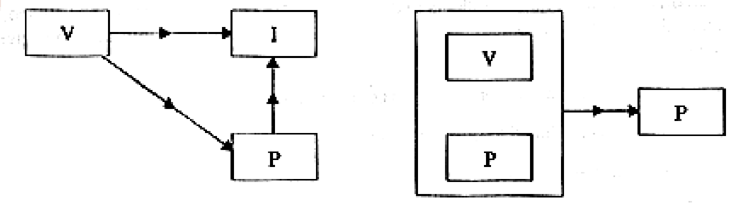

Fig 10.10: Dependency diagram of Professor Relation

If vendor V1 has to supply to project P2, but the item is not yet decided, then a row with a blank for item code has to be introduced. The information about item 1 is stored twice for vendor V3. Observe that the relation given inTable 10.8 is in 3NF and also in BCNF. It still has the problems mentioned above. The problem is reduced by expressing this relation as two relations in the Fourth Normal Form (4NF). A relation is in 4NF if it has no more than one independent multivalued dependency, or one independent multivalued dependency with a functional dependency.

Table 10.9 can be expressed as the two 4NF relations given in Table 10.10.The fact that vendors are capable of supplying certain items and that they are assigned to supply for some projects are independently specified in the 4NF relation.

Table 10.10: Vendor-supply-project Relations in 4NF

a) Vendor Supply

Vendor Code	Item code
V1	I1
V1	I2
V2	I2
V2	I3
V3	I1

b) Vendor project

Vendor Code	Project no.
V1	P1
V1	P3
V2	P1
V3	P1
V3	P2

These relations still have a problem. Even though vendor V1's capability to supply items and his allotment to supply for specified projects may not need it. We thus need another relation that specifies this. This is called the 5NF form. The 5NF relations are the relations in Table 10.10(a) and 10.10(b) together with the relation given in table 10.11.

Table 10.11: 5NF Additional Relation

Project no.	Item code
P1	I1
P1	I2
P2	I1
P3	I1
P3	I3

In table 10.12 we summarize the normalization steps already explained

Table 10.12: Summary of Normalization steps

Input relation	**Transformation**	**Output relation**
All relations	Eliminate variable-length records.	1NF
1NF relation	Remove dependency of non-key attribute on part of a multiattribute key	2NF
2NF	Remove dependency of non-key attribute on other non-key attributes	3NF
3NF	Remove dependency of an attribute ofa multiattribute key on an attribute of another (overlapping) multi-attribute key	BCNF
BCNF	Remove more than one independent multivalued dependency from relation by splitting relation	4NF
4NF	Add one relation relating attributes with multivalued dependency to the two relations with multivalued dependency	5NF

SELF ASSESSMENT QUESTIONS – 8

17. A relation is in 4NF if it has no more than one____________multivalued dependency or one independent multivalued dependency with a functional dependency.
18. In 4 NF, transformation is done by adding one relation relating attributes with multivalued dependency to the_____________relations withmultivalued dependency.

10. SUMMARY

In this unit, we discussed that there is no fool-proof algorithmic method of identifying dependency and hence we have to use our commonsense and judgment to specify dependencies. We dealt about the importance of having a consistent database without repetition of data and pointed out the anomalies that could be introduced in a database with an undesirable scheme. We also discussed the properties of normalized relations. Then we discussed the several forms of normalization that could help in removing these anomalies.

11. TERMINAL QUESTIONS

1. What is the basic purpose of 4NF?
2. What types of anomalies are found in relational database?
3. Define the term *functional dependency*.
4. Give a set of FDs for the relation schema *R(A,B,C,D)* with the primary key *AB* under which *R* is in 1NF but not in 2NF.
5. Consider the relation schema *R(A,B,C)*, which has the FD *B* ➔ *C*. If *A* is a candidate key for *R*, is it possible for *R* to be in BCNF? If so, under what conditions? If not, explain why not.
6. Suppose that we have a relation schema *R(A,B,C)* representing a relationship between two entity sets with keys *A* and *B*, respectively, andsuppose that *R* has (among others) the FDs *A* ➔ *B* and *B* ➔ *A*. Explain what such a pair of dependencies means (i.e., what they imply about therelationship that the relation models).
7. Consider a relation *R* with five attributes *ABCDE*. You are given the following dependencies: *A* ➔ *B*, *BC* ➔ *E*, and *ED* ➔ *A*.
   a. List all keys for *R*.
   b. Is *R* in 3NF?
   c. Is *R* in BCNF?

8. Consider the attribute set R = *ABCDEGH* and the FD set F = { $AB \rightarrow C$, $AC \rightarrow B$, $AD \rightarrow E$, $B \rightarrow D$, $BC \rightarrow A$, $E \rightarrow G$ }.
   a. For each of the following attribute sets, do the following:
      i. Compute the set of dependencies that hold over the set andwrite down a minimal cover.
      ii. Name the strongest normal form that is not violated by therelation containing these attributes.
      iii. Decompose it into a collection of BCNF relations if it is not inBCNF.
      a) *ABC* b) *ABCD* c) *ABCEG* d) *DCEGH* e) *ACEH*

   b. Which of the following decompositions of R = *ABCDEG*, with thesame set of dependencies F, is (a) dependency-preserving?
      a) lossless-join?
      b) {*AB, BC, ABDE, EG*}
      c) {*ABC, ACDE, ADG*}

12. ANSWERS

Self-Assessment Questions

1. X $\rightarrow$ Y
2. composite
3. only once
4. anomalies
5. decomposition
6. Duplicated
7. self-contained
8. flat
9. lower
10. 1NF
11. non-key
12. Third Normal
13. Dependent
14. Semantic
15. BCNF
16. Independent
17. two

Terminal Questions

1. Basic purpose of 4NF transformation is Normalization of data when attributes in a relation have multi-valued dependency. It is done byadding one relation relating attributes with multivalued dependency to the two relations with multivalued dependency. (Refer section 9)
2. Anomalies found in relational databases are Update Anomalies,Insertion Anomalies, and Deletion Anomalies (Refer section 3 for detail)
3. A functional dependency occurs when one attribute in a relation uniquely determines another attribute. This can be written X -> Y which would be the same as stating "Y is functionally dependent upon X".
4. Refer section 2 for detail
5. Yes, it is possible for R to be in BCNF if R has more than one possiblekey. (Refer section 8 for detail)
6. The given pair of dependencies means functional dependencies. (Refersection 10.2 for detail)
7. Refer whole unit for detail.
8. Refer whole unit for detail.
9. Refer whole unit for detail.