DATABASE MANAGEMENT SYSTEMS

Objectives:

To provide knowledge about the normalization techniques.

Learning Outcomes:

Students will be able to

- Understand various normalization techniques.
- Perform lossless decomposition and FD preserving on relations.
- To know how to overcome anomalies caused by redundancy.
- To know how to eliminate redundancy.
- Design the good database schema.

UNIT-III

Functional Dependencies: Partial, Full, Transitive and Trivial Dependencies, Axioms, Decomposition- Lossless join and Dependency Preserving Decomposition, Attribute closure, Normal forms- 1NF, 2NF, 3NF and BCNF.

Unit - III

Database Design

Functional Dependencies and Normalization for Relational Databases

1. Functional Dependencies

The single most important concept in relational schema design is a functional dependency.

A *Functional Dependency* describes a relationship between *attributes* within a single relation.

An attribute is *functionally dependent* on another if we can use the value of one attribute to determine the value of another. Formally, functional dependency is defined as;

Functional dependency – In agiven relation R, X and Y are attributes. Attribute Y is functionally dependent on attribute X if each value of X determines exactly one value of Y. This is represented as X → Y.

For example, consider the following relation

Reports(Student#, Course#, CourseName, Iname, Room#, Marks, Grade)

In this relation, {Student#, Course#} together can determine exactly one value of Marks. This can be symbolically represented as

{Student#, Course#} → Marks

This type of dependency is called as functional dependency. In the above example Marks is functionally dependent on {Student#, Course#}.

Other functional dependencies in the above example are:

- Course# →CourseName
- Course# →Iname (Assuming one course is taught by one and only one instructor)
- Iname→ Room# (Assuming each instructor has his/her own room)
- Marks → Grade
- Course# → Room#

Full Functional Dependency - In agiven relation R, X and Y are attributes. Attribute Y is fully functionally dependent on attribute X only if it is not functionally dependent on sub-set of X where X is composite in nature.

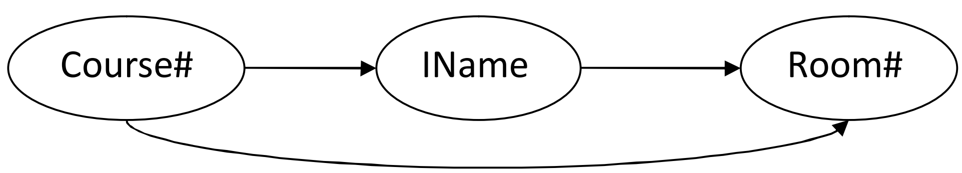
In the above example, Marks is fully functionally dependent on {Student#, Course#} and not on sub-set of {Student#, Course#}. This means Marks cannot be determined either by Student# or by Course#. It can be determined using Student# and Course# together. Hence, Marks is fully functionally dependent on {Student#, Course#}.

Partial Functional Dependency – In agiven relation R, X and Y are attributes. Attribute Y is partially dependent on attribute X only if it is dependent on sub-set of attribute X where X is composite in nature.

In the above example, CourseNmae, IName, Room# are partially dependent on {Student#, Course#} because Course# alone determines the CourseNmae, IName, Room#.

Transitive Dependency – In a given relation R, if attribute X determines attribute Y and attribute Y determines attribute Z, then attribute X determines attribute Z. Such a dependency is called transitive dependency.

Following example shows a transitive dependency.



Trivial Dependency-A FunctionalDependency X->Y is said to be trivial functional dependency if y is a subset of X(Y⊆X). In other words if R.H.S of some FD is the subset of L.H.S of FD is called Trivial Functional Dependency.

Example: AB->A

AB->B

AB->AB

2.Axioms: Armstrong's axioms are a set of rules, that when applied repeatedly generates a closure of functional dependencies.

1. Reflexive Rules: if X is a set of attributes and Y is a subset of X then X holds Y.

   $Y \subseteq X \Rightarrow X \rightarrow Y$

2. Augmentation Rule: if X hold Y and Z is a set of attributes then XZ holds YZ.

   X->Y then XZ->YZ

3. Transitive Rule: if X holds Y and Y holds Z , then X holds Z.

   X->Y and Y->Z then X->Z

4. Additive or Union Rule: if X holds Y and X holds Z ,then X holds YZ.

   X->Y and X->Z then X->YZ

5. Pseudo Transitive Rule: if X holds Y and YZ holds W, then XZ holds W.

   X->Y and YZ->W then XZ->W

6. Productive Rule or Decomposition Rule: if X holds YZ and X holds Y, then X holds Z.

   X->YZ then X->Y and X->Z

3.Decomposition:

- A functional decomposition is the process of breaking down the functions of an organization into progressively greater (finer and finer) levels of detail.
- In decomposition, one function is described in greater detail by a set of other supporting functions.
- The decomposition of a relation schema R consists of replacing the relation schema by two or more relation schemas that each contain a subset of the attributes of R and together include all attributes in R.
- Decomposition helps in eliminating some of the problems of bad design such as redundancy, inconsistencies and anomalies.

There are two types of decomposition :(Properties of Decomposition)

1. **Lossless Join Decomposition**
2. **Dependency Preserving Decomposition**

Lossless Join Decomposition :

- "The decomposition of relation R into R1 and R2 is lossless when the join of R1 and R2 yield the same relation as in R."
- A relational table is decomposed (or factored) into two or more smaller tables, in such a way that the designer can capture the precise content of the original table by joining the decomposed parts. This is called lossless-join (or non-additive join) decomposition.
- This is also referred as non-additive decomposition.
- The lossless-join decomposition is always defined with respect to a specific set F of dependencies.
- Consider that we have table STUDENT with three attribute roll_no ,sname and department.

  STUDENT :

Roll_no	Sname	Dept
111	parimal	COMPUTER
222	parimal	ELECTRICAL

This relation is decomposed into two relation Stu_name and Stu_dept :

Stu_name:

Roll_no	Sname
111	parimal
222	parimal

stu_dept

Roll_no	Dept
111	COMPUTER
222	ELECTRICAL

Now ,when these two relations are joined on the comman column 'roll_no' ,the resultant relation will look like stu_joined. stu_joined :

Roll_no	Sname	Dept
111	Parimal	COMPUTER
222	Parimal	ELECTRICAL

In lossless decomposition, no any spurious tuples are generated when a natural joined is applied to the relations in the decomposition.

Dependency Preservation Decomposition :

The dependency preservation decomposition is another property of decomposed relational database schema D in which each functional dependency X -> Y specified in F either appeared directly in one of the relation schemas R_i in the decomposed D or could be inferred from the dependencies that appear in some R_i.

Decomposition D = { R1 , R2, R3,,.., ,Rm} of R is said to be dependency-preserving with respect to F if the union of the projections of F on each R_i , in D is equivalent to F. In other words, R join of R1, R1 over X. The dependencies are preserved because each dependency in F represents a constraint on the database. If decomposition is not dependency-preserving, some dependency is lost in the decomposition.

Example:

Let a relation R(A,B,C,D) and set a FDs F = { A -> B , A -> C , C -> D} are given.

A relation R is decomposed into - R1 = (A, B, C) with FDs F1 = {A -> B, A -> C}, and

R2 = (C, D) with FDs F2 = {C -> D}.

F' = F1 ∪ F2 = {A -> B, A -> C, C -> D}

so, F' = F.

And so, F'+ = F+.

4.Attribute Closure:

The set of all those attributes which can be functionally determined from an attribute set is called as closure of that attribute set.

Closure of an attribute set {X} is denoted as $\{x\}^+$.

Steps to find closure of an attribute set:

1. Add the attributes contained in the attribute set for which closure is being calculated to the result set.
2. Recursively add the attributes to the result set which can be functionally determined from the attributes already contained in the result set.

Example: Consider a relation R(A,B,C,D,E,F,G) with the functional dependencies

A->BC

BC->DE

D->F

CF->G

Closure of Attribute A:

A+ ={A}

={ABC} ∵A->BC

={ABCDE} ∵BC->DE

={ABCDEF} ∵D->F

={ABCDEFG} ∵CF->G

Closure of Attribute set BC:

BC^+ ={BC}

={BCDE} ∵BC->DE

={BCDEF} ∵D->F

={BCDEFG} ∵CF->G

Normalization

Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion anomalies.

- These would include two properties
  - **Lossless join property**- which guarantees that the generation of spurious tuples will not occur.
  - **Dependency preservation property** - This ensures that each functional dependency is represented in some individual relation resulting after decomposition.
- **Prime attribute -** An attribute of relation schema R is called a prime attribute of R if it is a member of some candidate key of R.
- **Non prime attribute -** An attribute is called nonprime if it is not a prime attribute—that is, if it is not a member of any candidate key.

If a relation schema has more than one key, each is called a candidate key. One of the candidate keys is arbitrarily designated to be the primary key, and the others are called secondary keys.

Normal forms: The normal form is a relation refers to the highest normal form condition that it meets and hence indicates the degree to which it has been normalized.

Normal forms are used to eliminate or reduce redundancy in database tables.

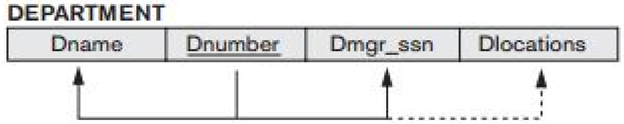
First Normal Form (1NF)

A relation R is said to be in the first normal form if and only if all the attributes of the relation R are atomic in nature.

For example, consider the Department table given in figure (b). It is not in 1NF because one of its attributes Dlocations is non-atomic as it contains more than one value in row 1. To make it 1NF compliant, create a separate row for each value of Dlocations of row 1 as shown in figure (c).

(a)

DEPARTMENT



(b)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

(c)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocation
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Figure (a) A relation schema (b) sample state of relation Department that is not in 1NF (c)1NF version of the same relation

Second Normal Form (2NF)

A Relation is said to be in Second Normal Form (2NF) if and only if:

- It is in First normal form (1NF)
- No partial dependency exists between non-key attributes and key attributes

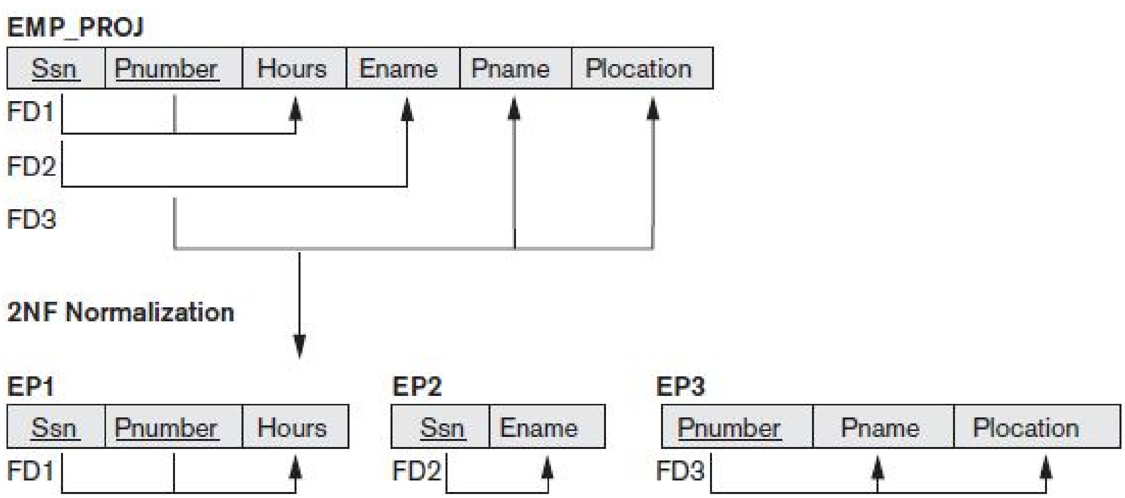
As an example, consider the EMP_PROJ schema shown below;

For this table, the key is {Ssn, Pnumber}

The functional dependencies are as follows;

- {Ssn, Pnumber} → Hours
- Ssn→Ename
- Pnumber→Pname
- Pnumber→Plocation

It is clear from these functional dependencies that the table has partial dependencies (Ssn→Ename, Pnumber→Pname, Pnumber→Plocation)



To make it 2NF compliant, remove all partial dependencies. For this, we need to split EMP_PROJ table into 3 tables as EP1, EP2 and EP3 as shown above.

Now these 3 tables do not contain partial dependencies and hence they are in 2NF.

Example2: Consider a relation R(A,B,C,D,E,F) with the functional dependencies

A->BCDEF

BC->ADEF

B->F

D->E

Solution:

Candidate keys are {A,BC}

So, prime attributes are {A,B,C}

Non-prime attributes are {D,E,F}

In the third functional dependency (B->F) partial dependency (non-prime attribute can't be derived from subset of candidate key) exists.

Therefore, the given relation R is not in 2NF

So, the relation R can be divided into two relations R1(A,B,C,D,E) and R2(B,F).

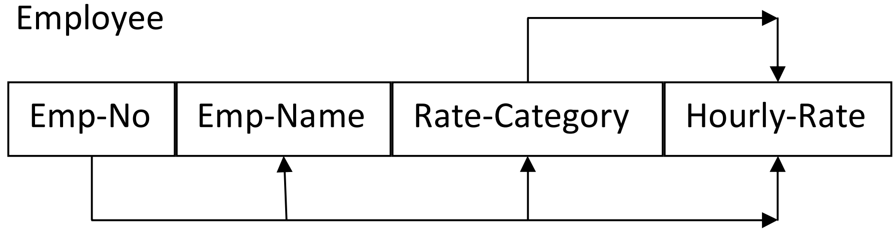
Third Normal Form (3NF):

A Relation is said to be in Third Normal Form (3NF) if and only if:

- It is in Second Normal Form (2NF)
- No transitive dependency exists between non-key attributes and key attributes

For example, consider the following Employee table.

Employee

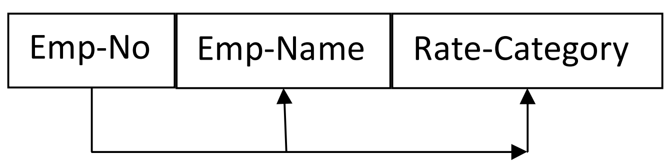


This table contains a transitive dependency as given below;

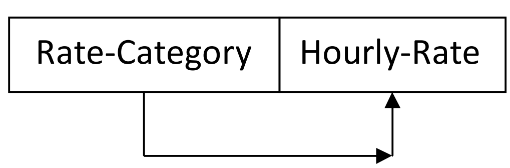
Emp-No → Rate-Category → Hourly-Rate

Hence, Employee table is not in 3NF. To make it 3NF compliant, we need to remove this transitive dependency. To do that, we need to split Employee table into two tables (Employee table and Rate table) as given below.

Employee Table



Rate Table



Now, both Employee and Rate tables are in 3NF as they do not have transitive dependencies.

Example2: Consider a relation R(A,B,C,D,E) with the functional dependencies

A->BCDE

BC->ADE

D->E

Solution:

Candidate keys are {A,BC}

So, prime attributes are {A,B,C}

Non-prime attributes are {D,E}

In the third functional dependency (D->E) Transitive dependency (non-prime attribute can't be derived from another non-prime attribute) exists.

Therefore, the given relation R is not in 3NF

So, the relation R can be divided into two relations R1(A,B,C,D) and R2(D,E).

Boyce-Codd Normal Form (BCNF):

A relation is said to be in BCNF if and only if all the determinants are candidate keys. BCNF relation is a strong 3NF relation. i.e. all BCNF relations are in 3NF but the reverse is not true.

For example, consider the following Result table.

Result Table

Student#	EmailID	Course#	Marks

This table has two candidate keys – {Student#, Course#} and {EmailID, Course#}.

This table is in 3NF because it has no partial and transitive dependencies between key attributes and non-key attributes. But this table is not in BCNF because all determinants are not candidate keys. This can be observed from the following FDs;

- {Student#, Course#} → Marks
- {EmailID, Course#} → Marks
- Student# →EmailID
- EmailID→ Student#

Though the determinants of first two FDs are candidate keys but the determinants of the last two FDs are not candidate keys. Thus it is violating the BCNF condition.

To make this table into BCNF compliant, we need to decompose the Result table into two tables as shown below;

Student Table

Student#	EmailID

Result Table

Student#	Course#	Marks

Example2: Consider a relation R(A,B,C,D) with the functional dependencies

A->BCD

BC->AD

D->B

Solution:

Candidate keys are {A,BC}

So, prime attributes are {A,B,C}

Non-prime attributes are {D}

The third functional dependency (D->B) Violates BCNF condition(because D is not a candidate key).

Therefore, the given relation R is not in BCNF

So, the relation R can be divided into two relations R1(A,C,D) and R2(B,D).

UNIT-III

Assignment-Cum-Tutorial Questions

SECTION-A

Objective Questions

1. The normalization of 1NF relations to 2NF involves: [ ]

   A)Removal of partial dependencies

   B) Removal of full dependencies

   C) Removal of transitive dependencies

   D) Removal of multi-valued dependencies

2. Why do we go for normalization of databases? [ ]

   A) To avoid the repetitions B) To prevent fragmentation

   C) To avoid redundancy D) To save memory

3. If a relation is in BCNF then it is in: [ ]

   A) 2 NF B) 3 NF C) 1 NF D) 1 NF and 2 NF

4. If a relation with a schema R is decomposed into two relations R1 and R2 such that (R1 ∪ R2) = R1 then which one of the following is to be satisfied for a lossless joint decomposition (→ indicates functional dependency) [ ]

   A) $R1 \cap R2 \rightarrow R1$ or $R1 \cap R2 \rightarrow R2$

   B) $R1 \cap R2 \rightarrow R1$

   C) $R1 \cap R2 \rightarrow R2$

   D) $R1 \cap R2 \rightarrow R1$ and $R1 \cap R2 \rightarrow R2$

5. Identify the minimal key for the relational scheme R(A, B, C, D, E) with functional dependencies F = {A→B, B→C, AC→D}. [ ]

   A) A B) AE C) BE D) CE

6. The best normal form of relation scheme R(A, B, C, D) along with the set of functional dependencies F = {AB→C, AB→D, C→A, D→B} is [ ]

   A) Boyce-Codd Normal form B) Third Normal form

   C) Second Normal form D) First Normal form

7. Match the following database terms to their functions: [ ]

List-I	List-II
(a) Normalization	(i) Enforces match of primary key to foreign key
(b) Data Dictionary	(ii) Reduces data redundancy in a database
(c) Referential Integrity	(iii) Define view(s) of the database for particular user(s).
(d) External Schema	(iv) Contains metadata describing database structure.

Codes:

	(a)	(b)	(c)	(d)
(A)	(iv)	(iii)	(i)	(ii)
(B)	(ii)	(iv)	(i)	(iii)
(C)	(ii)	(iv)	(iii)	(i)
(D)	(iv)	(iii)	(ii)	(i)

8. A relation R={A,B,C,D,E,F,G} is given with the following set of functional dependencies: F={AD→E, BE→F, B→C, AF→G}. [ ]
Which of the following is a candidate key?
A) A B) AB C) ABC D) ABD

9. Consider a relational schema R= (A, B, C, D, E, F, G, H) on which of the following functional dependencies hold: {A→B, BC→D, E→C, D→A}. [ ]

What are the candidates keys for R?
A) AE, BE B) AEH, BEH, DEH C) AEH, BEH, BCH D) AE, BE, DE

10. From the following instance of a relational schema R(A,B,C) we can conclude that

A	B	C
1	1	1
1	1	0
2	3	2
2	3	2

A) B does not functionally determines C

B) A does not functionally determine B and B does not functionally determines C

C) A functionally determine B and B functionally determines C

D) A functionally determine B and B does not functionally determines C

11. The relational schema student_performance(name, courseno, rollno, grade) has the following functional independencies. The highest normal form of this relation is______.

{name, courseno}→grade

{rollno, courseno}→grade

name→rollno

rollno→name

12. Given the following relation instance [ ]

X	Y	Z
1	4	2
1	5	3
1	6	3
3	2	2

Which of the following functional dependencies are satisfied by the instance?

A) XY→Z and Y→X B) YZ→X and X→Z

C) XY→Z and Z→Y D) YZ→X and Y→Z

13. Relation R with an associated set of functional dependencies F, is decomposed into BCNF. The redundancy in the resulting set of relations is [ ]

A) Zero

B) more than zero but less than that of an equivalent 3 NF decomposition

C) proportional to the size of F

D) Indeterminate

SECTION-B

SUBJECTIVE QUESTIONS

1. Outline the informal guidelines for relational schema.
2. What are the problems caused by redundancy? Explain.

3. Define decomposition and illustrate lossless and dependency preserving decompositions.
4. What is normalization? Explain 1NF,2NF,3NF and BCNF.
5. Differentiate between 3NF and BCNF.
6. Define FD, MVD and JD.
7. Describe 4NF and 5NF with an example.
8. Consider the relation schema R(ABCD) and the FDs {AB->C, B->D}.What is the highest normal form condition it satisfies?
9. Consider the relation schema R(ABC) and the FDs {AB->C, C->A}. What is the highest normal form that it satisfies?
10. Consider the relation schema R(ABC) and the following FDs {AB->C, C->B}. What is the highest normal form condition it satisfies?
11. Given R(A,B,C,D, E) with the set of FDs, F{AB→CD, A→E, C→D}. Is the decomposition of R into R1(A, B, C), R2(B, C, D) and R3(C, D, E) lossless? Prove.
12. Given R(A,B,C,D,E) with the set of FDs, F{AB→CD, ABC→E, C→A}
    (i) Find any two candidate keys of R
    (ii) What is the normal form of R? Justify.
13. Let R = (A, B, C, D) and F be the set of functional dependencies for R given by {A→B, A→C, BC→D}. Prove A→D.
14. Given R = ABCD with the FD set F = { A→B, B→C, C→D}. Determine all 3NF violations. Decompose the relation into relations which are in 3NF.
15. Determine a candidate key for R = ABCDEG with the FD set
    F = {AB→C, AC→ B, AD→E, B→D, BC→A, E→G}

SECTION-C

QUESTIONS AT THE LEVEL OF GATE

1. R(ABCD) is a relation. Which of the following does not have a lossless join, dependency preserving BCNF decomposition? **[GATE 2001]** [ ]
   A) A→B, B→C , C→D   B) A→B, B→CD
   C) AB→C, C→AD   D) A→BCD

2. The following functional dependencies are given below **[Gate 2005]**

   AB→CD, AF→D, DE→F, C→G, F→E, and G→A

   Which of the following option is false? [ ]

   A) {CF}+ = {ABCDEFG} B) {AF}+ = {ABCDEFG}

   C) {AB}+ = {ABCDFG} D) {BG}+ = {ABCDG}

3. Which of the following is TRUE? **[GATE 2012]**

   A) Every relation is 3NF is also in BCNF

   B) A relation R is in 3NF if every non-prime attribute of R is fully functionally dependent on every key of R

   C) Every relation in BCNF is also in 3NF

   D) No relation can be in both BCNF and 3NF

4. Relation R has eight attributes ABCDEFGH. Fields of R contain only atomic values.

   F = {CH → G, A→ BC, B →CFH, E → A, F →EG} is a set of functional dependencies (FDs) so that F+ is exactly the set of FDs that hold for R

   How many candidate keys does the relation R have? [**GATE 2013]** **[ ]**

   A) 3 B) 4 C) 5 D) 6