Unit-4: Relational Database Design Process	
Short Questions [1 Mark]	
1.	**What are pitfalls in relational database design?**
Ans:	➢ The pitfalls in relational database design are: 1. Repetition of information 2. Inability to represent certain information 3. Loss of information
2.	**What is decomposition?**
Ans:	Decomposition is the way of arranging the database in an organized manner using normalization in order to reduce the duplication (redundancy) of data.
3.	**Why decomposition is needed?**
Ans:	➢ Decomposition is the process of breaking one relation into more than one relation. ➢ We need decomposition to remove redundancy and avoid inconsistency. ➢ Relations are decomposed using functional dependency.
4.	**What do you meant by fully functional dependency?**
Ans:	Fully functional dependency indicates that if A and B are columns of a table, B is fully functionally dependent on A, if B is not dependent on any subset of A.
5.	**What is transitive dependency?**
Ans:	A functional dependency X → Y in relation schema R is transitive dependency, if there is a set of attributes Z, such that, Z is a non-prime attribute and both dependencies X → Z and Z → Y hold.
6.	**Define normalization.**
Ans:	Normalization of data is a process during which unsatisfactory relation schemas are decomposed by breaking up their attributes into smaller relation schemas that possess desirable properties. or Normalization is a process of decomposition of a relation (table) into more relations to avoid database anomalies (Insertion, updation, deletion). Normalization is based on Functional Dependency.
7.	**Define first normal form.**

Ans:

- 1NF states that the domains of attributes must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute.
- It disallows multivalued attributes, composite attributes and their combinations.

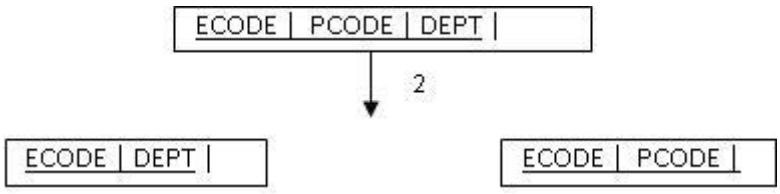
8. What is 2NF?

Ans:

- A relation schema R is in 2NF, if every nonprime attribute A in R is fully functionally dependent on the primary key.
- A functional dependency X->Y is a fully functional dependency if removal of any attribute A from X means that the dependency does not hold any more; that is, for any attribute A->X, (X-{A}) XY.

9. Give an example for a relation in 2NF.

Ans: Consider a relation in 2NF:



10. When can you say that the relation is in third normal form?

Ans:

- A relation schema R is in 3NF if it is in 2NF and no nonprime attribute of R is transitively dependent on the primary key.
- A functional dependency X -> Y in relation schema R is a transitive dependency if there is a set of attributes Z that is not a subset of any key of R, and both X->Y and Z->Y hold.

Short Questions [2 Marks]

1. What do you mean by redundancy? How this can be avoided?

Ans:

- Repeating of the same information in the table is called redundancy. Redundancy leads to wastage of memory. Redundant data introduces anomalies like insertion, updation and deletion anomaly. It also causes data inconsistency.
- Redundancy can be avoided by normalizing the table.

Class	Teacher	RoomNo	Size
BE(IT Sem-1)	Prof. Sharma	C301	350
BE(IT Sem-1)	Dr. Patel	C301	350
BE(IT Sem-1)	Prof. Desai	C301	350
BE(IT Sem-2)	Prof. Anjali	C302	250

- Here the information that “size of room no C301 is 350” is getting repeated in the first three rows. So this is redundancy.

2. What is deletion anomaly? Give example.

Ans:

- If we put attributes of more than one entity in a single relation then this problem will occur.
- For Example:

  Consider the following relation in which attributes of employee and department entities are present. If we delete the employee tuple that happens to represent the last employee working for a particular department, the information concerning that department is lost from the database.

Ename	Eno	Dob	Dno	Dname	Dmgrno

3. List the design goals of relational database.

Ans:

- The design goals of relational database are as follows:

1. Semantics of attributes:

   Design a relation schema so that it is easy to explain its meaning. Do not put attributes of multiple entities into a single relation.
2. No anomalies:

   Design the relation schemas so that no insertion, deletion, or modification anomalies occur in the relations.
3. Avoid null values:

   As for as possible, avoid placing attributes in a base relation whose values may be null. If nulls are unavoidable, make sure that they apply in exceptional
4. No spurious tuples:

   Design relation schemas so that they can be joined with equality conditions on attributes that are wither primary or foreign keys in a way that no spurious tuples are generated.

4. What are the desirable properties of decompositions?

Ans:

- Following are the two desirable properties of decomposition:

1. Lossless-join decomposition:

   Decomposition is lossless join decomposition if we are able to retrieve all information by joining the relations.

2. Dependency preservation:

   Decomposition is dependency preserving decomposition if all functional dependencies that exist in the original relation also exist in the decomposed relations.

5. Why normalization is needed?

Ans:

- To ensure that the update anomalies do not occur.

- Normal forms provide a formal frame work for analyzing relation schemas based on their keys and on the functional dependencies among their attributes.
- A series of tests that can be carried out on individual relation schemas so that the relation database can be normalized to any degree. When a test fails, the relation violating that test must be decomposed into relations that individually meet the normalization tests.

6. Difference between first normal form and second normal form.

Ans:

First normal form	Second normal form
1. 1NF states that the domains of attributes must include only atomic values.	**1.** A relation schema R is in 2NF, if every nonprime attribute A in R is fully functionally dependent on the primary key.
2. It disallows multivalued attributes, composite attributes and their combinations.	**2.** A functional dependency X->Y is a full functional dependency if removal of any attribute A from X means that the dependency does not hold any more; that is, for any attribute A->X, (X-{A}) X Y.

7. What do you mean by functional dependency?

Ans:

- A functional dependency is a constraint between two sets of attributes from the data base.
  A functional dependency, denoted by…

  X -> Y
- Between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuples that can form a relation instance r of R.
- R={ A1,A2,…,An}. The constraint states that, for any two tuples t1 and t2 in r such that t1[X]=t2[X] , we must also have t1[Y] = t2[Y].
  Therefore, We can also say that Y is functionally dependent on X.

8. Consider the following relation:

R (A, B, C, D, E)

The primary key of the relation is AB. The following functional dependencies hold:

A →C

B →D

AB → E

Is the above relation in second normal form?

Ans:

- A relation schema R is in 2NF, if every nonprime attribute A in R is fully functionally dependent on the primary key.
- The above relation is not in second normal form because non key attributes (C and D) are not fully functionally depends on the primary key (AB).

9. Consider the following relation:

R(A, B, C, D)

The primary key of the relation is A. The following functional dependencies hold:

A →B,C

B →D

Is the above relation in third normal form?

Ans:

- No, the above relation is not in third normal form.
- For the 3NF all non-key attributes are depend on the primary key only. But, here primary key is A and non-key attribute D is depend on the B. So, it is not in 3NF.

10. Which are the three design goals for relational databases? Why each is desirable.

Ans:

Goal 1: Avoid unnecessary redundant information:

Avoiding repetition of information helps to have a more concise database making it easy to update or check for corrupted information and make it easy to retrieve information.

Goal 2: Preserve information upon decomposition (lossless decomposition):

Avoiding loss of information is crucial for the integrity of the database. The database is as good as the information stored. Lose of information compromises the integrity and the functionality of the database.

Goal 3: Preserve functional dependencies:

Make it easy to check if updates in the database don't result in illegal relationships and make it easy to enforce constrains. Preserving functional dependencies on design make check dependencies easier during updates.

Scenario based Questions [5 Marks]

1. Explain what problems are caused by the design of the following table.

ID	name	salary	dept_name	building	budget
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

Ans:

1. If a department has more than one instructor, the building name and budget get repeated multiple times. Updates to the building name and budget may get performed on some of the copies but not others, resulting in an inconsistent state where it is not clear what is the actual building name and budget of a department.

2. A department needs to have at least one instructor in order for building and budget information to be included in the table. Nulls can be used when there is no instructor, but null values are rather difficult to handle.

3. If all instructors in a department are deleted, the building and budget information are also lost. Ideally, we would like to have the department information in the database irrespective of whether the department has an associated instructor or not, without resorting to null values.

2. Consider the following relation and answer the following questions.

CUSTOMER_ID	LAST_NAME	FIRST_NAME	PHONES
JS1	Smith	John	310-456-4022 (W) 310-444-8712 (H)
PH1	Pocahontas		310-432-2813

1. Is the relation in 1 NF? Justify your answer. If it is not in 1NF, convert it into 1NF.
2. Is the relation in 2 NF? Justify your answer. If it is not in 2NF, convert it into 2NF.
3. Is the relation in 3 NF? Justify your answer. If it is not in 1NF, convert it into 3NF.

Ans:

1. **The First Normal Form (1NF):**
   - A relation is said to be in 1NF if and only if every attribute of the relation contains atomic value.
   - The given relation is not in 1NF because phones contains multiple values for the same records.
   - To convert it into 1NF, we need to replace the attribute phones with two attributes

phone_w and phone_h.

CUSTOMER_ID	LAST_NAME	FIRST_NAME	PHONES_W	PHONES_H
JS1	Smith	John	310-456-4022	310-444-8712
PH1	Pocahontas		310-432-2813	

2. **The Second Normal Form (2NF):**
   - A relation is said to be in 2NF if and only if
     1. It is in 1NF
     2. Every non-prime attribute fully functionally depends on primary key.
   - Here customer_id is primary key and following FD exists.
   - Customer_id → last_name, first_name, phones_w, phones_h
   - So all the non-prime attributes fully functionally depends on primary key, therefore the above relation is in 2NF.

3. **The Third Normal Form (3NF):**
   - A relation is said to be in 3NF if and only if
     1. It is in 2NF
     2. No non-prime attribute functionally depends on another non-prime attribute.
   - From the set of FD [Customer_id → last_name, first_name, phones_w, phones_h]
   - It is clear that the above two conditions of 3NF holds, therefore the relation is in 3NF.

3. **Consider the following relation and answer the following questions.**

COURSE	SECTION	INSTRUCTOR	COURSE NAME
Cosc 250	1	Nguyen	Computer Science for Business
Cosc 250	2	Warford	Computer Science for Business
Cosc 250	3	Nguyen	Computer Science for Business
Cosc 250	4	Zimmerman	Computer Science for Business
Cosc 480	1	Nguyen	Programming Languages

1. Is the relation in 1 NF? Justify your answer. If it is not in 1NF, convert it into 1NF.
2. Is the relation in 2 NF? Justify your answer. If it is not in 2NF, convert it into 2NF.
3. Is the relation in 3 NF? Justify your answer. If it is not in 1NF, convert it into 3NF.

Ans: **1. The First Normal Form (1NF):**

- A relation is said to be in 1NF if and only if every attribute of the relation contains atomic value.
- It clear that from the above relation all attribute have an atomic value. So, it is in 1NF.

2. **The Second Normal Form (2NF):**

- A relation is said to be in 2NF if and only if
    1. It is in 1NF
    2. Every non-prime attribute fully functionally depends on primary key.
- Here course+section is primary key.
- In the above relation only the Instructor attribute is fully depends on the primary key and course name is only depend on the course.
- So, we can decomposite the relation into two relations.

Courses Table

COURSE	COURSE NAME
Cosc 250	Computer Science for Business
Cosc 480	Programming Languages

Sections Table

COURSE	SECTION	INSTRUCTOR
Cosc 250	1	Nguyen
Cosc 250	2	Warford
Cosc 250	3	Nguyen

3. The Third Normal Form (3NF):

- A relation is said to be in 3NF if and only if
    1. It is in 2NF
    2. No non-prime attribute functionally depends on another non-prime attribute.
- In the course relation the following FD is exist:

course → course name

- In the section relation the following FD is exist:

  course + section → instructor

- It is clears that the above relations hold the 3NF conditions. So, they are in 3NF.

4. Consider the following relation and answer the following questions.

EMP_ID	EMP_NAME	JOB_CODE	JOB_TITLE	DATE_HIRED	JOB_DESC
A120	Jones	1	Programmer	9/17/95	Write computer code.
A721	Harpo	1	Programmer	7/17/93	Write computer code.
B270	Garfunkel	2	Analyst	1/12/95	Perform cost analysis.
C273	Selsi	3	Designer	5/21/94	Design graphics.

1. Is the relation in 1 NF? Justify your answer. If it is not in 1NF, convert it into 1NF.
2. Is the relation in 2 NF? Justify your answer. If it is not in 2NF, convert it into 2NF.
3. Is the relation in 3 NF? Justify your answer. If it is not in 1NF, convert it into 3NF.

Ans: **1. The First Normal Form (1NF):**

- A relation is said to be in 1NF if and only if every attribute of the relation contains atomic value.
- In the above relation all attribute contain atomic value. So, it is in the 1NF.

2. The Second Normal Form (2NF):

- A relation is said to be in 2NF if and only if
  1. It is in 1NF
  2. Every non-prime attribute fully functionally depends on primary key.
- Here primary key is EMP_ID and the following FD is exist:

  EMP_ID → emp_name, job_code, job_title, date_hired, job_description.
- So, it is in 2NF.

3. The Third Normal Form (3NF):

- A relation is said to be in 3NF if and only if
  1. It is in 2NF

2. No non-prime attribute functionally depends on another non –prime attribute.

- Here primary key is emp_id.
- Job_title and job_description is not only depending on the primary key, it is also depending on the job_code.
- To convert into in 3NF, we can decomposite the relation into two relations.

Employees Table

EMP_ID	EMP_NAME	JOB_CODE	DATE_HIRED
A120	Jones	1	9/17/95
A721	Harpo	1	7/17/93
B270	Garfunkel	2	1/12/95
C273	Selsi	3	5/21/94

Jobs Table

JOB_CODE	JOB_TITLE	JOB_DESC
1	Programmer	Write computer code.
2	Analyst	Perform cost analysis.
3	Designer	Design graphics.

5. Consider the following relation and answer the following questions.

Album table

Album	Tracks	Artist	ArtistCountry
Abby Road	Here comes the sun, Octopus Garden, Something, etc	Beatles	UK
Blond on Blond	Rainy Day Woman, Sad eyed lady of the lowlands, Stuck in Memphis with the mobile blues again	Bob Dylan	US

1. **Is the relation in 1 NF? Justify your answer. If it is not in 1NF, convert it into 1NF.**
2. **Is the relation in 2 NF? Justify your answer. If it is not in 2NF, convert it into 2NF.**
3. **Is the relation in 3 NF? Justify your answer. If it is not in 1NF, convert it into 3NF.**

1. The First Normal Form (1NF):

- A relation is said to be in 1NF if and only if every attribute of the relation contains atomic value.
- So, the relation Album is not in 1NF as it contains attribute "Tracks" which is multi-valued attribute.

- **Solution:**

Album Table First Normal Form

Album	Artist	ArtistCountry
Abby Road	Beatles	UK
Blond on Blond	Bob Dylan	US

Track Table First Normal Form

Album	Tracks
Abby Road	Here comes the sun
Abby Road	Octopus's Garden
Abby Road	Something
Blond on Blond	Rainy Day Woman
Blond on Blond	Sad Eyed Lady of the lowlands
Blond on Blond	Stuck in Mobile with the Memphis blues again

2. The Second Normal Form (2NF):

- The Artist information depends on the Track.
- To conform to the Second Normal Form, the two functional dependencies must be broken into separate Entities.
- To relate the Album entity to the Track Entity, it is necessary to create a primary key for the Album entity that can be used to create a key—foreign key relationship with the Track entity. It is also a good idea to give the Track entity a Primary key.

- **Solution:**

Album Table Second Normal Form

AlbumKey	AlbumTitle
ABRD	Abby Road
BLBL	Blond On Blond

Track Table Second Normal Form

TrackKey	TrackTitle	AlbumKey	Artist	ArtistCountry
HCTS	Here Comes the Sun	ABRD	Beatles	UK
SMTH	Something	ABRD	Beatles	UK
OPGD	Octopus's Garden	ABRD	Beatles	UK
RDWM	Rainy Day Woman	BLBL	Bob Dylan	Us
SELL	Sad Eyed Lady of the Lowlands	BLBL	Bob Dylan	US
SMMB	Stuck in Memphis with the Mobile Blues	BLBL	Bob Dylan	US

3. The Third Normal Form (3NF):

- There is a transitive dependency in the table. ArtistCountry, doesn't describe the track; it describes the
- Artist. The solution, as usual, is to break out a separate table. Artist should be its own entity.

- **Solution:**

Album Table Third Normal Form

AlbumKey	AlbumTitle
ABRD	Abby Road
BLBL	Blond On Blond

Artist Table Third Normal Form

ArtistKey	ArtistName	ArtistCountry
BTLS	Beatles	UK
BDLN	Bob Dylan	US

Track Table Third Normal Form

TrackKey	TrackTitle	AlbumKey	ArtistKey
HCTS	Here Comes the Sun	ABRD	BTLS
SMTH	Something	ABRD	BTLS
OPGD	Octopus's Garden	ABRD	BTLS
RDWM	Rainy Day Woman	BLBL	BDLN
SELL	Sad Eyed Lady of the Lowlands	BLBL	BDLN
SMMB	Stuck in Memphis with the Mobile Blues	BLBL	BDLN

6. **A college maintains details of its lecturers' subject area skills. These details comprise:**

- **Lecturer Number**
- **Lecturer Name**
- **Lecturer Grade**
- **Department Code**
- **Department Name**
- **Subject Code**
- **Subject Name**
- **Subject Level**

- **Assume that each lecturer may teach many subjects but may not belong to more than one department.**
- **Subject Code, Subject Name and Subject Level are repeating fields.**
- **Normalize these data to Third Normal Form.**

Ans:

UNF:

LecturerNumber, LecturerName, LecturerGrade, DepartmentCode, DepartmentName, SubjectCode, SubjectName, SubjectLevel

1NF:

Table 1: LecturerNumber, LecturerName, LecturerGrade, DepartmentCode, DepartmentName

Table 2: LecturerNumber, SubjectCode, SubjectName, SubjectLevel

2NF:

Table 1: LecturerNumber, LecturerName, LecturerGrade, DepartmentCode, DepartmentName

Table 2: SubjectCode, SubjectName, SubjectLevel

Table 3: LecturerNumber, SubjectCode

3NF:

Table 1: LecturerNumber, LecturerName, LecturerGrade, *Department Code

Table 2: DepartmentCode, DepartmentName

Table 3: SubjectCode, SubjectName, SubjectLevel

Table 4: LecturerNumber, SubjectCode

7. **A software contract and consultancy firm maintains details of all the various projects in which its employees are currently involved. These details comprise:**

- **Employee Number**
- **Employee Name**
- **Date of Birth**
- **Department Code**
- **Department Name**
- **Project Code**
- **Project Description**
- **Project Supervisor**

Assume the following:

- **Each employee number is unique.**
- **Each department has a single department code.**
- **Each project has a single code and supervisor.**
- **Each employee may work on one or more projects.**
- **Employee names need not necessarily be unique.**
- **Project Code, Project Description and Project Supervisor are repeating fields.**

Normalize these data to Third Normal Form.

Ans:

UNF:

EmployeeNumber, EmployeeName, DateofBirth, DepartmentCode, DepartmentName, ProjectCode, ProjectDescription, ProjectSupervisor

1NF:

Table 1: EmployeeNumber, EmployeeName, DateofBirth, DepartmentCode, DepartmentName

Table 2: EmployeeNumber, ProjectCode, ProjectDescription, ProjectSupervisor

2NF:

Table 1: EmployeeNumber, EmployeeName, DateofBirth, DepartmentCode, DepartmentName

Table 2: ProjectCode, ProjectDescription, ProjectSupervisor

Table 3: EmployeeNumber, ProjectCode

3NF:

Table 1: EmployeeNumber, Employee Name, Date of Birth, *Department Code

Table 2: DepartmentCode, DepartmentName

Table 3: ProjectCode, ProjectDescription, ProjectSupervisor

Table 4: EmployeeNumber, ProjectCode

Long Questions [5 Marks]

1. **"A relational database should be designed with care." Justify.**

OR

What are the various problems or pitfalls introduced by the database which is poorly designed?

OR

Explain in Insertion, Updation and Deletion anomaly.

Ans:

- For any enterprise, database should be designed with a good quality, so that, the resultant system can work according to its requirements in an efficient way.
- A poorly designed database always results in a malfunctioning, inefficient system.

- Pit-falls in Relational Database Design:
  1. Redundancy
  2. Update Anomalies
  3. Insertion Anomalies
  4. Deletion Anomalies

- For Example:

  Consider the two relations **Account** and **Branch**. But, instead of using two separate relations, consider that only one single relation **Account_Branch** has been designed.

Account

ano	balance	branch
A01	5000	Surat
A02	6000	Navsari
A03	7000	Surat
A04	6000	Bardoli
A05	8000	Navsari

Branch

branch	baddress
Surat	Ring Road, Surat
Navsari	Mota bazaar, Navsari
Bardoli	Station Road, Bardoli

Account_Branch

ano	balance	branch	baddress
A01	5000	Surat	Ring Road, Surat
A02	6000	Navsari	Mota bazaar, Navsari
A03	7000	Surat	Ring Road, Surat
A04	6000	Bardoli	Station Road, Bardoli
A05	8000	Navsari	Mota bazaar, Navsari

1. **Redundancy:**
   - In this example, the branch address is stored repeatedly for each account of the same branch, occupying more amount of memory. And any branch will have thousand of accounts rather than just 1 or 2 accounts as shown.
   - Result of this is a data redundancy and memory wastage.

2. Update Anomalies:

- Consider that the branch address of one of the branches, say for 'Surat' branch, changes.
- In such case, branch address in all tuples, which belong to 'Surat' branch in Account_Branch relation must be changed. Failure in updating some of the tuples will result in inconsistent state of the database.
- The same branch will have different addresses. This must be avoided.
- Such type of problem is known as "Update anomaly".

3. Insertion Anomalies:

- What happens if the bank opens a new branch which doesn't have any accounts till now? Here, there is a need to insert a new tuple related to new branch storing its name and address.
- In case of Account_Branch relation, the only way to do this is to place null values in the attributes for account – account number and balance. But, this causes a problem.
- As account number is primary key, null value cannot be inserted. As a result, it is impossible to store data related to new branch.
- Such type of problem is known as "Insertion anomaly".

4. Deletion Anomalies:

- What happens if the tuple representing some account needs to be deleted from the Account_Branch relation, and, that account is the last account in that branch.
- The information concerning that branch will also be lost from the database.
- In our example, if tuple with account number 'A04' is deleted, the information regarding 'Bardoli' branch will also be lost.
- Such kind of problem is known as "Deletion anomaly".

2. Write a note on decomposition.

Ans:

- "Decomposition is the process of breaking down given relation into two or more relations."
- A relation R is replaced by two or more relations in such a way that –
  - Each new relation contains a subset of the attributes of R, and
  - Together, they all include all attributes of R.
- Relational database design process starts with a universal relation schema R = {A1, A2,…,An}, which includes all the attributes of the database.

- The universal relation states that every attribute name is unique.
- Using functional dependencies, this universal relation schema is decomposed into a set of relation schemas D = {R1, R2,…,Rm}.
- Now, D becomes the relational database schema and D is referred as decomposition of R.
- Generally, decomposition is used to eliminate the pitfalls of the poor database design during normalization process.

- For Example:

Account_Branch:

ano	balance	branch	baddress
A01	5000	Surat	Ring Road, Surat
A02	6000	Navsari	Mota bazaar, Navsari
A03	7000	Surat	Ring Road, Surat
A04	6000	Bardoli	Station Road, Bardoli
A05	8000	Navsari	Mota bazaar, Navsari

- The Account_Branch relation suffers from various database anomalies. This relation can be replaced with two different relations: **Account**(acc_num, balance, bname) and **Branch**(bname, baddress).

Account:

ano	balance	branch
A01	5000	Surat
A02	6000	Navsari
A03	7000	Surat
A04	6000	Bardoli
A05	8000	Navsari

Branch:

branch	baddress
Surat	Ring Road, Surat
Navsari	Mota bazaar, Navsari
Bardoli	Station Road, Bardoli

- These two relations eliminate many of the pitfalls of the relation Account_Branch.
- A Decomposition of a relation can be either lossy decomposition or lossless decomposition.

3. Explain Lossy join decomposition with suitable example.

Ans:

- "The decomposition of relation R into R1 and R2 is lossy when the join of R1 and R2 does not yield the same relation as in R."
- This is also referred as lossy decomposition.

➢ Disadvantage:

Some information is lost during retrieval of original relation. And so, such type of decomposition is referred as lossy decomposition.

➢ For Example:

- Relation Account is decomposed into two relations – Acc_Bal and Bal_Branch.

Account:

ano	balance	branch
A01	5000	Surat
A02	6000	Navsari

Acc_Bal:

ano	balance
A01	5000
A02	6000

Bal_Branch:

balance	branch
5000	Surat
6000	Navsari

Acc_Joined:

ano	balance	branch
A01	5000	Surat
A01	6000	Navsari
A02	5000	Surat
A02	6000	Navsari

- Now, when these two relations are joined on the common column 'balance', the resultant relation will look Acc_Joined relation. This relation contains rows in addition to those in original relation Account.
- Here, it is not possible to specify that in which branch account 'A01' or 'A02' belongs.
- So, information has been lost by this decomposition and then join operation.
- In lossy join decomposition, spurious tuples are generated when a natural join is applied to the relations in the decomposition.
- Decomposition is lossy if R is subset of join of R1 and R2, over X, where R is an original relation, R1 and R2 are decomposed relations, and X is a common attribute between these two relations.

4. Explain Lossless join decomposition by giving example.

Ans:

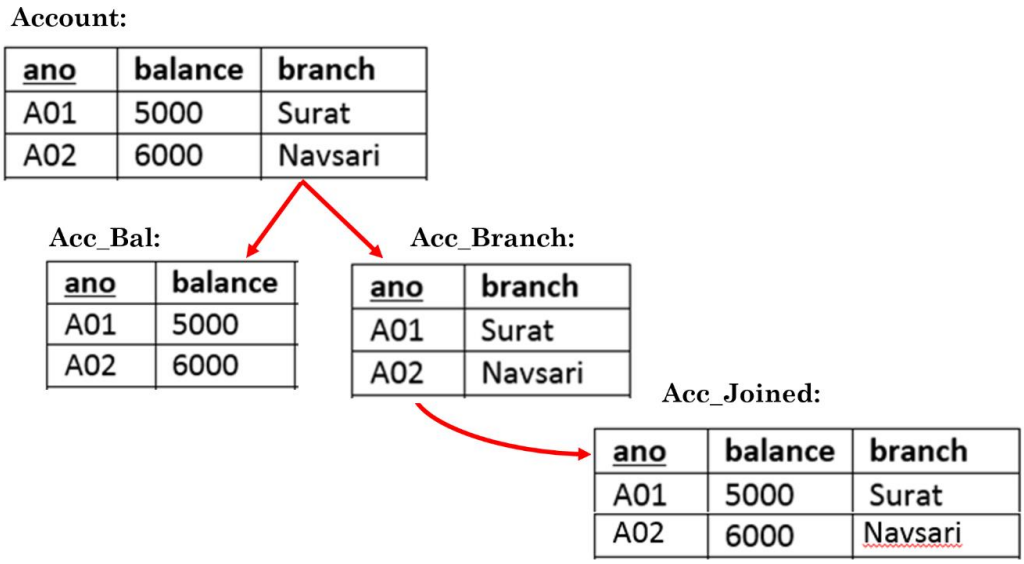
➢ "The decomposition of relation R into R1 and R2 is lossless when the join of R1 and R2 yields the same relation as in R."

➢ This is also referred as non-additive decomposition.

➢ All decompositions must be lossless. Here, the word 'loss' refers to the loss of information.

➢ For Example:

- The relation Account is decomposed into two relations – Acc_Bal and Acc_Branch.



- Now, when these two relations are joined on the common column 'ano', the resultant relation will look like Acc_Joined relation.
- This relation is exactly same as that of original relation Account. In other words, all the information of original relation is preserved here.
- In lossless decomposition, no any spurious tuples are generated when a natural join is applied to the relations in the decomposition.
- Decomposition is lossless if R = join of R1 and R2, over X, where R is an original relation, R1 and R2 are decomposed relations, and X is a common attribute between these two relations.

5. Write a note on Normalization.

Ans:

➢ Normalization is a process of decomposition of a relation (table) into more relations to avoid database anomalies (Insertion, updation, deletion).

➢ Normalization is based on Functional Dependency.

➢ "Normalization is the iterative process, which proceeds in a top-down fashion, by evaluating each relation against the criteria for normal forms, and decomposing relations if required."

➢ This process is first proposed by Dr. E. F. Codd in 1972.

Goals of Normalization:

➢ The main goals of the normalization are:

- To minimize the data redundancy, and
- To minimize update, insertion and deletion anomalies.

Normalization Process:

- In Normalization, given relation schema is analyzed based on its functional dependencies and primary key.
- If it does not satisfy the criteria specified by the normal form, it is decomposed into smaller relation schemas.
- Normalization is then applied on each of the new decomposed relation schema.
- If it satisfies the criteria, it is passed to next iteration, where it is tested against criteria for next higher normal form.

Normal Forms:

- Normal forms are state of a relation that results from applying some criteria (or rules, or conditions) on that relation. A relation is said to be in a normal form, if it satisfies the criteria for that normal form.

- Following are the levels of normal forms:
    1. First Normal Forms (1NF)
    2. Second Normal Forms (2NF)
    3. Third Normal Forms (3NF)
    4. Boyce-Codd Normal Forms (BCNF)
    5. Fourth Normal Forms (4NF)
    6. Fifth Normal Forms (5NF)

6. Explain 3NF in detail with proper example.

Ans:

- A relation is said to be in 3NF if and only if
    1. It is in 2NF
    2. No non-prime attribute functionally depends on another non-prime attribute.

- For Example:

To convert into in 3NF, we can decompose the relation into two relations.

- Here emp_id is a primary key.
- Job_title and job_description is not only depending on the primary key, it is depending on the job_code.

Employees Table

EMP_ID	EMP_NAME	JOB_CODE	DATE_HIRED
A120	Jones	1	9/17/95
A721	Harpo	1	7/17/93
B270	Garfunkel	2	1/12/95
C273	Selsi	3	5/21/94

Jobs Table

JOB_CODE	JOB_TITLE	JOB_DESC
1	Programmer	Write computer code.
2	Analyst	Perform cost analysis.
3	Designer	Design graphics.