Page No. _____
Date

Unit - III (Relational database design and Operation)

Pitfalls in database design

Pitfalls in database design refers to common mistakes and challenges that can lead to inefficient, insecure or difficult to maintain database.

- This issue often arises due to :+

1) Poor design / Planning
2) Ignoring normalization
3) Redundant Records
4) Poor naming standard
5) Lack of documentation
6) One table to hold all domain value
7) Ignoring purpose of data
8) Insufficient indexing
9) Lack of testing

Types
1) Redundancy
2) Null values

Dependencies

Dependencies in DBMS is a relation between two or more attributes. or tables ensuring data consistency and integrity.

Types of dependencies

1) functional dependencies

A functional dependency occur when one attribute uniquely determine another attribute within a relation.

- It is a constraint that describe how attribute in a table relate to each other.

Ex :-

Emp ID	Emp Age	Emp Name
E01	28	Amit
E02	31	Rohit

In above table, Emp Name is functionaly dependent on Emp ID because Emp Name can take only one value.

Types

↳ Trivial functional dependency

Those functional dependency which is always valid.

↳ Non - trivial functional dependency

Those functional dependency in which we check for valid or invalid.

Ex :-

S_Id	Name
1	Amit
2	Rohit

Valid

SId	Name
1	Amit
2	Amit

Valid

S_Id	Name
1	Amit
1	Rohit

Invalid

S_Id	Name
1	Amit
1	Amit

Invalid

Properties

→ Reflexivity

If Y is subset of X, $X \rightarrow Y$

→ Augmentation

If $X \rightarrow Y$ then $XZ \rightarrow YZ$ (for any Z)

→ Trasitivity

If $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

→ Union

If $X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow YZ$

→ Pseudotransitivity

If $X \rightarrow Y$ and $YZ \rightarrow W$ then $XZ \rightarrow W$

→ Composition

If $X \rightarrow Y$ and $Z \rightarrow W$ then $XZ \rightarrow YZ$

2) Multivalued dependency

Multivalued dependency (MVD) deals with complex attribute relationship in which an attribute may have many independent value while depending on another attribute or group of attributes.

- It improves database structure and consistency.
- MVD or multivalued dependency means that for a single value of attribute 'a' multiple values of attribute 'b' exist.

$$a \longrightarrow\!\!\!\!\rightarrow b$$

- When one attribute in a database depend on another attribute and has many independent values is said to have multivalued dependency (MVD).

- Ex:-

Car_Model	Manuf_Month	Colour
S2011	January	Yellow
S2001	February	Red
S3001	March	Yellow

- We use multivalued dependency when we want to test the relation or when we want to determine what limitation are there on the arrangement.

3) Join dependencies

Join dependency arises when the attributes in one relation are dependent on attribute in another relation which means certain raws will exist in table if there is same raw in another table.

- $\bowtie$ represent natural join operator

Types

→ Lossless join dependency
It means that whenever the join occurs between the tables, then no information should be lost the new table must have all the content.

→ Lossy join dependency
In this type of dependency, data loss may occur at some point which include absence of tuples.

Ex:- Student -

Std_Name	Std_Skill	Std_Job
Amit	Developer	Jk
Rohit	tester	GS

Student Skills -

Std_Name	Std_Skill
Amit	Developer
Rohit	tester

Student job -

Std_Name	Std_job
Amit	JK
Rohit	GS

Job Skills -

Std_Robe	Std_job
JK	Developer
GS	tester

Database Anomalies

Anomalies in the relational model refers to inconsistencies or error that can arise when working with relational database specifically in context of insertion, deletion and modification

- There are three types of data anomalies :-

1) Insertion Anomalies

These anomalies occur when it is not possible to insert data into a database because the required field are missing or because the data is incomplete.

2) Deletion Anomalies

These anomalies occur when deleting a record from a database and can result in unintentional losss of data.

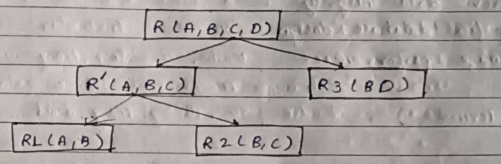
3) Update Anomalies

These anomalies occur when modifying data in a database and can result in inconsistencies or errors,

- These anomalies can be removed with the process of Normaliza-tion which generally splits the database which result in reducing the anomalies in the database.

Decomposition

When we divide a table into multiple tables or divider a relation into multiple relation than this process is termed as Decomposition in DBMS.



Types of decomposition

1) Lossless decomposition

The process in which we can regain the original relation R with help of join from multiple relation formed after decomposition.

2) Lossy decomposition

Lossy decomposition means when we perform join operation on sub - relation.

Normalization

Normalization is a systematic approach to organize data within a database to reduce redundancy and eliminate undesirable characteristics such as insertion, deletion and update anomalies

why is normalization important :-

1) Reduce Data Redundancy
2) Improve data integrity
3) Simplifies database design
4) Optimize performance

Normal forms

- 1st NF (first normal form)

A table is in 1NF if it satfies following condition

- All column contain atomic values (indivisible values)
- Each row is unique
- Each column has unique name
- Order of data stored is not matter

- 2nd NF (second normal form)

A table is in 2NF if it satfies the condition of 1NF Additionaly,

- No partial dependency exist means non-prime attribute depend on entire primary key.

- 3NF (Third Normal form)

A table is in 3NF if it satisfies the condition of 2NF Additionaly,

- No transitive dependencies means non-prime attributes does not depend on other non-prime attribute

- BCNF (Boyce-Codd Normal form)

A table is in BCNF if it satisfies the condition of 3NF Additionaly,

- Every non-trivial functional dependency mus be a superkey

- 4NF (fourth Normal form)

A table is in 4NF if it satisfies the condition of BCNF and has no multi-valued dependencies it occurs when one attributes determine other and both attributes are independent of all other attributes

- 5NF

A table is in 5NF if it satisfies the condition of 4NF and all join dependencies are removed. This eliminate redundancy and optimize data storage.

Advantages of normal form

→ Reduce data redundancy
→ Improved data consistency
→ Simplify database design
→ Improve query performance
→ Easier to maintain database

Example :- Unnormalised table (UNF)

Std_ID	Name	Courses	Instructor
101	Alice	Maths, Sci.	John, Mary
102	Bob	Maths	John
103	Charlie	Sci., History	Mary, david

First Normal form (1NF) :- Spliting into separate records

Std_ID	Name	Course	Instructor
101	Alice	Maths	John
101	Alice	Science	Mary
102	Bob	Maths	John
103	Charlie	Science	Mary
103	Charlie	History	david

Similarly, we can make 5 other (NF) tables.

Denormalization

Denormalization is a database optimization technique, in which we add redundant data to one or more tables. It is an optimization technique that is applied after normalization.

- The process of taking a normalized schema and making it non-normalized is called denormalization.

Advantages → Improved Query performance
→ Reduced complexity
→ Easier maintenance
→ Improved performance
→ Better scalability

Difference between normalization and denormalization

Normalization	Denormalization
1) Non-redundancy and consistency data are stored in schema	Data are combined to execute query quickly
2) Data redundancy and inconsistency is reduced	Redundancy is added for quick execution
3) Data integrity is maintained	Data integrity is not maintained
4) No. of tables is increased	No. of tables is decreased
5) Optimize the use of disk space	Do not optimize the use of disk space

Relational Algebra

- Relational algebra is a fundamental query language in database management system.
- It provide a mathematical framework fore querying database ensuring efficient data retrival and manipulation.

Key concept
- Relation
  A relation is a table that consist of row & column.
- Tuple
  A tuple is a single row in a relation.
- Attributes
  Attributes are column in a relation.

• Selection Operation (σ)

Selection operation is basically used to filter out rows from a given table based on certain condition.

Ex- In relation R we have attribute A, B and C we want to select tuples $C > 3$,

A	B	C
1	2	4
2	2	3
3	2	3
4	3	4

Output :-

A	B	C
1	2	4
4	3	4

We will select tuple which have C more than 3.

• Projection Operation (π)

While selection operation works on rows, similarly projection operation of relational algebra works on column.

Ex:- we want column B and C from R.

B	C
2	4
2	3
3	4

- By default, projection operation removes duplicate values.

• Union operation ($\cup$)

The union operator is basically used to combine the result of two queries into a single result.

Ex:- French :-

Name	Roll.No.
Ram	01
Mohan	02
Vivek	13
Geeta	17

German

Name	Roll No.
Vivek	13
Geeta	17
Shyam	21
Rohan	25

Output :- $\pi_{(student\ name)}$ French $\cup$ $\pi_{(std\_name}$ German

Ram
Mohan
Vivek
Geeta
Shyam
Rohan

- Minus operation or Set difference (−)

It basically provide the rows that are present in one table but not in another table.

Ex:- $\pi_{(std\_name)}(French) - \pi_{(std\_name)}(German)$

Std_name
Ram
Mohan
Shyam
Rahan

- Intersection operation (∩)

Basically fetches only those rows of data that are common between two relation

Ex:- $\pi_{(std\_name)}$ French ∩ $\pi_{(std\_name)}$ German

Std_name
Vivek
Geeta

- Cartesian product (X)

Cartesian product combines every row of one table with every row of another table producing every possible combination.

Ex:-

Name	Age	Religion
Ram	14	Hindu
Sona	15	Muslim

ID	Course
1	DS
2	DBMS

Output -

Name	Age	Religion	ID	Course
Ram	14	Hindu	1	DS
Ram	14	Hindu	2	DBMS
Sona	15	Muslim	1	DS
Sona	15	Muslim	2	DBMS

Join Operation

- Join operation in relational algebra combine data from two or more relation based on related attribute, allowing for more complex queries and data retrieval.

Types

Inner Join

An inner join combine rows from two relation based on matching condition and only return rows where there is match in both relation.

Outer join

An outer join return rows from one relation and matching rows from the other relation.

→ Left out join
→ Right outer join
→ full outer join