Roll No. 65

CLASSMATE
Date :
Page :

EXPERIMENT - 7

AIM: Study of Normalization in R-DBMS.

THEORY :

Normalization definition: Normalization is a procedure in which we transform database (decompose) from one form to another form so that our database would be free from all the anomalies which are insertion, updation and deletion.

A large database defined as a single relation may result in data duplication. The repetition of data may result in :

→ Making relations very large
→ It isn't easy to maintain and update data as it would involve searching many records in relation
→ Wastage and poor utilization of disk space and resources
→ The likelihood of errors and inconsistencies increases.

So to handle these problems, we should analyze and decompose the relations with redundant data into smaller, simpler, and well-structured relations that are to satisfy desirable properties. Normalization is a process of decomposing the relations into relations with fewer attributes.

CLASSMATE
Date :
Page :

Uses of Normalization:

1) Normalization is used to minimize the redundancy from a relation of set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update and Deletion Anomalies.
2) Normalization divides the larger table into smaller and links them using relationships.
3) The normal form is used to reduce redundancy from the database table.

Normalization is necessary for several reasons (needs):

1) To minimize the redundancy: When data is stored in multiple places, it can lead to inconsistencies and errors. Normalization helps to eliminate these redundancies by dividing data into smaller tables.
2) To improve data integrity: Normalization helps to ensure that the data in a database is consistent and accurate by reducing the potential for errors and inconsistencies.
3) To make the database more flexible: Normalization makes it easier to modify the database schema, add new tables or columns, and make changes to the database without affecting other parts of the system.

* Data modification anomalies can be categorized into three types :

→ 1) Insertion Anomaly : Insertion anomaly refers to when one cannot insert a new tuple into a relationship due to lack of data.
2) Deletion Anomaly : The delete anomaly refers to the situation where the deletion of data results in the unintended loss of some other important data
3) Updation Anomaly : The update anomaly is when an update of a single data value requires multiple rows of data to be updated.

* Advantages of Normalization :

→ 1) Normalization helps to minimize data redundancy.
2) Greater overall database organization.
3) Data consistency within the database.
4) Much more flexible database design.
5) Enforces the concept of relational integrity

* Disadvantages of Normalization :

→ 1) You cannot start building the database knowing what the user needs.
2) The performance degrades when normalizing the relations to higher normal forms i.e. 4NF, 5NF.

EXAMPLE OF 1NF :

(Multivalued attribute ∴ Not 1NF)

Sid	Sname	cname
1	Raju	C, C++
2	Rani	C, Java
3	Rakesh	Java

Decomposition into 1NF ⟹

(First Normal form)

Sid	Sname	cname
1	Raju	C
1	Raju	C++
2	Rani	C
2	Rani	Java
3	Rakesh	Java

(Composite Attribute ∴ Not 1NF)

sid	sname	Address
1	Raju	22-ABC-BOMBAY
2	Rani	39-XYZ-PUNE
3	Rakesh	67-NMT-DELHI

⇓ Decomposition into 1NF

(First Normal Form)

sid	sname	plot No.	streetname	city
1	Raju	22	ABC	BOMBAY
2	Rani	39	XYZ	PUNE
3	Rakesh	67	NMT	DELHI

3) It is very time-consuming and difficult to normalize relations of a higher degree.
4) Careless decomposition may lead to a bad database design, leading to serious problems.

---

* Types of Normal forms :

1) First Normal form (1NF) :

→ A relation will be 1NF if it contains an atomic value.
→ It states that an attribute of an table cannot hold multiple values. It must hold only single-valued attribute.

Example (← on blank page)

---

2) Second Normal Form : (2NF) :

→ To be in 2NF, a relation must be in 1 normal form and relation must not contain any partial dependency. A relation is in 2NF if it has No Partial Dependency. i.e no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.
Partial Dependency - If the proper subset of candidate key determines non-prime attribute, it is called partial dependency.

EXAMPLE (2NF) : Let's assume, a school can store the data of teachers and the subjects they teach. In a school, a teacher can teach more than one subject.

Teacher table

TEACHER_ID	SUBJECT	TEACHER_AGE
25	chemistry	30
25	Biology	30
47	English	35
83	Maths	38
83	Computer	38

In the given table, non-prime attribute TEACHER_AGE is dependent on TEACHER_ID which is a proper subset of a candidate key. That's why it violates the rule for 2NF.

To convert the given table into 2NF, we decompose it into two tables:

TEACHER_DETAIL TABLE

TEACHER ~~DETAIL~~ ID	TEACHER_AGE
25	30
47	35
83	38

TEACHER_SUBJECT Table

TEACHER_ID	SUBJECT
25	chemistry
25	Biology
47	English
83	Maths
83	Computer

3) Third Normal Form (3NF):

A relation will be in 3NF if it is in 2NF and not containing transitive partial dependency. 3NF is used to reduce the data duplication. It is also used to achieve data integrity. If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form. For 3NF, every non-trivial functional dependency $X \rightarrow Y$, X is a super key and Y is a prime attribute, i.e each element of Y is part of some candidate key.

EXAMPLE: <STUDENT>

Rollno	Game	Feestructure
1	Basketball	500
2	Basketball	500
3	Basketball	500
4	Cricket	600
5	Cricket	600
6	Cricket	600
7	Tennis	400

In this student table, it is in 1NF because there are no multivalued attributes. It is also in 2NF because all non-key attributes are fully functional dependent on the primary key (rollno). But, the table is not in 3NF because there is transitive dependency, i.e. game → feestructure. Feestructure has transitive/indirect dependency on rollno. via game.

To overcome these anomalies, the student table should be divided into smaller tables.

i) R_1

Rollno	Game
1	Basketball
2	Basketball
3	Basketball
4	Cricket
5	Cricket
6	Cricket
7	Tennis

ii) R_2

Game	Feestructure
Basketball	500
Cricket	600
Tennis	400

4) Boyce Codd normal form (BCNF)

BCNF is the advanced version of 3NF, A table is in BCNF if every functional dependency $X \rightarrow Y$, X is the super key of the table. For BCNF, the table should be in 3NF and for every functional dependency, LHS is super key.

EXAMPLE: Consider a Relation R with attributes (student, subject, teacher)

Student	Teacher	Subject
Jhansi	P. Naresh	Database
Jhansi	K. Das	C
Subbu	P. Naresh	Database
Subbu	R. Prasad	C

Candidate keys are (student, teacher) & (student, subject). The above relation is in 3NF (since there is no transitive dependency). But, it is not in BCNF, because in the FD (teacher → subject), teacher is not a key. This relation suffer with anomalies.

So, R is divided into two relations R_1 (Teacher, subject) and R_2 (Student, Teacher)

i) R_1

Teacher	Subject
P. Naresh	Database
K. Das	C
R. Prasad	C

ii) R_2

Student	Teacher
Jhansi	P. Naresh
Jhansi	K. Das
Subbu	P. Naresh
Subbu	R. Prasad

5) Fourth Normal Form (4NF):

A relation is said to be 4NF if it is in Boyce Codd normal form (BCNF) and has no multi-valued dependency (MVD). For a dependency A → B, if for a single value of A,

multiple values of B exists, then the relation will be a multi-valued dependency.

EXAMPLE: Consider the following table R.

Reg No.	Phone No.	Qualification
1	P_1	Diploma
1	P_2	B.Tech
1	P_1	M.Tech
1	P_2	Diploma
1	P_2	B.Tech
1	P_2	M.Tech

Here, Reg no → phone no
Reg no → qualification
Both are non trivial MVD.
The given relation is in BCNF (since no functional dependency exists). But, the above table is not in 4NF (since there is a non trivial MVD)

So, R has to be decomposed into 2 tables.

i) R_1

Reg No	Phone no
1	P_1
1	P_2

ii) R_2

Reg No.	Qualification
1	Diploma
1	B.Tech
1	M.Tech

6) Fifth Normal Form (5NF)

The 5NF is also known as project-join normal form (PJ/NF). A relation is in 5NF if it is in 4NF and contains any join dependency and joining should be lossless. 5NF is

satisfied when all the tables are broken into as many tables as possible in order to avoid redundancy.

EXAMPLE: <Employee>

EmpName	Empskills	EmpJob
David	Java	E145
John	Javascript	E146
Jamie	Jquery	E146
Emma	Java	E147

The relation violates the fifth normal form (5NF), so it can be decomposed into following 3 tables therefore it is not in 5NF.

i) <Employee skills>

EmpName	Empskills
David	Java
John	Javascript
Jamie	Jquery
Emma	Java

ii) <Employee Job>

EmpName	EmpJob
David	E145
John	E146
Jamie	E146
Emma	E147

iii) <Job Skills>

Empskills	EmpJob
Java	E145
Javascript	E146
Jquery	E146
Java	E147

CONCLUSION: Normalization and different types of forms has been studied in relational database management system (RDBMS).