Pune Vidyarthi Griha's
College of Engineering, Nashik
206, Behind Reliance Petrol Pump, Dindoriroad, Mhasrul, Nashik-422004
Phone: 0253-6480000 / 36 /44 Toll Free Number: 18002665330
Website: https://pvgcoenashik.org Email: admission@pvgcoenashik.org
Approved by AICTE, New Delhi, DTE, Mumbai and
Affiliated to Savitribai Phule Pune University, Pune.
DTE Code: EN5330

"Relational Database Design"

By,

Prof. Anand N. Gharu

Asst. Professor

Computer Department,

PVG COE, Nashik.

What is Anomaly?

"Anomalies are problems that can occur in poorly planned, un-normalised databases where all the data is stored in one table (a flat-file database)"

There are three anomalies :

1. Insert
2. Update
3. Delete

Example : https://youtu.be/qrTHlXgJyDQ

What is Anomaly?

1. Insertion anomaly:

An Insert Anomaly occurs when certain attributes cannot be inserted into the database without the presence of other attributes.

For example this is the converse of delete anomaly - we can't add a new course unless we have at least one student enrolled on the course.

What is Anomaly?

2. update anomaly:

An update anomaly is a data inconsistency that results from data redundancy and a partial update.

For example, each employee in a company has a department associated with them as well as the student group they participate in.

Employee_ID	Name	Department	Student_Group
123	J. Longfellow	Accounting	Beta Alpha Psi
234	B. Rech	Marketing	Marketing Club
234	B. Rech	Marketing	Management Club
456	A. Bruchs	CIS	Technology Org.
456	A. Bruchs	CIS	Beta Alpha Psi

What is Anomaly?

3. Delete anomaly :

A deletion anomaly is the unintended loss of data due to deletion of other data.

For example, if you try to delete information when student name = anand then it will delete all the record for name anand so it deletes unwanted record.

What is Anomaly?

emp_id	emp_name	emp_address	emp_dept
101	Rick	Delhi	D001
101	Rick	Delhi	D002
123	Maggie	Agra	D890
166	Glenn	Chennai	D900
166	Glenn	Chennai	D004

The above table is not normalized. We will see the problems that we face when a table is not normalized.

Update anomaly: In the above table we have two rows for employee Rick as he belongs to two departments of the company. If we want to update the address of Rick then we have to update the same in two rows or the data will become inconsistent. If somehow, the correct address gets updated in one department but not in other then as per the database, Rick would be having two different addresses, which is not correct and would lead to inconsistent data.

Insert anomaly: Suppose a new employee joins the company, who is under training and currently not assigned to any department then we would not be able to insert the data into the table if emp_dept field doesn't allow nulls.

Delete anomaly: Suppose, if at a point of time the company closes the department D890 then deleting the rows that are having emp_dept as D890 would also delete the information of employee Maggie since she is assigned only to this department.

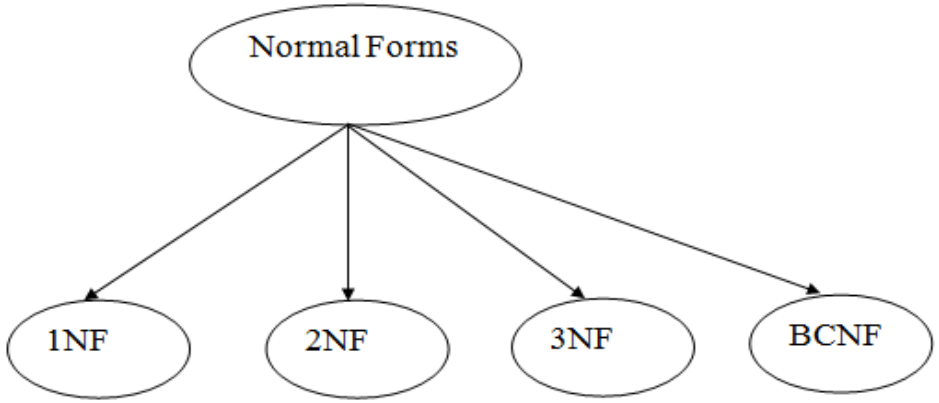
What is Normalization?

"Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion and updation anomalies. So, it helps to minimize the redundancy in relations".

Normal forms are used to reduce redundancy in database tables.

Types of Normal Form –

1. 1NF
2. 2NF
3. 3NF
4. NF
5. 5NF
6. BCNF



TYPES OF NORMAL FORM

Database Normal Forms

1. 1NF (First Normal Form)
2. 2NF (Second Normal Form)
3. 3NF (Third Normal Form)
4. BCNF (Boyce-Codd Normal Form)
5. 4NF (Fourth Normal Form)
6. 5NF (Fifth Normal Form)
7. 6NF (Sixth Normal Form)

1NF Normalization

Normal Form	Description
1NF	A relation is in 1NF if it contains an atomic(single value).
2NF	A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.
3NF	A relation will be in 3NF if it is in 2NF and no transition dependency exists.
4NF	A relation will be in 4NF if it is in Boyce Codd normal form and has no multi-valued dependency.
5NF	A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless.

1NF Normalization

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single(atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

FIRST NF EXAMPLE : https://youtu.be/mUtAPbb1ECM

1NF Normalization

In this example, table contains multiple values for attaributes

Course	Content
Programming	Java, c++
Web	HTML, PHP, ASP

We re-arrange the relation (table) as below, to convert it to First Normal Form

Course	Content
Programming	Java
Programming	c++
Web	HTML
Web	PHP
Web	ASP

2NF Normalization

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.
2. In the second normal form, all non-key attributes are fully functional dependent on the primary key.

SECOND NF EXAMPLE : https://youtu.be/R7UblSu4744

2NF Normalization

TEACHER table

TEACHER_ID	SUBJECT	TEACHER_AGE
25	Chemistry	30
25	Biology	30
47	English	35
83	Math	38
83	Computer	38

2NF Normalization

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

TEACHER_DETAIL table:

TEACHER_ID	TEACHER_AGE
25	30
47	35
83	38

TEACHER_SUBJECT table:

TEACHER_ID	SUBJECT
25	Chemistry
25	Biology
47	English
83	Math
83	Computer

3NF Normalization

For a relation to be in Third Normal Form, it must be in Second Normal form and the following must satisfy −

No non-prime attribute is transitively dependent on prime key attribute.

For any non-trivial functional dependency, X → A, then either −

X is a superkey or,

A is prime attribute.

Examaple : https://youtu.be/aAx_JoEDXQA

3NF Normalization

We find that in the above Student_detail relation, Stu_ID is the key and only prime key attribute. We find that City can be identified by Stu_ID as well as Zip itself. Neither Zip is a superkey nor is City a prime attribute. Additionally, Stu_ID → Zip → City, so there exists transitive dependency.

To bring this relation into third normal form, we break the relation into two relations as follows −

BCNF Normalization

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

R must be in 3rd Normal Form

and, for each functional dependency ($X \rightarrow Y$), X should be a super Key.

BCNF EXAMPLE : https://youtu.be/NNjUhvvwOrk

BCNF Normalization

Boyce-Codd Normal Form (BCNF) is an extension of Third Normal Form on strict terms. BCNF states that −

For any non-trivial functional dependency, X → A, X must be a super-key.

In the above image, Stu_ID is the super-key in the relation Student_Detail and Zip is the super-key in the relation ZipCodes. So,

Stu_ID → Stu_Name, Zip

and

Zip → City

Which confirms that both the relations are in BCNF.

3NF vs BCNF Normalization

SR. NO.	3NF	BCNF
1.	In 3NF there should be no transitive dependency that is no non prime attribute should be transitively dependent on the candidate key.	In BCNF for any relation A->B, A should be a super key of relation.
2.	It is less stronger than BCNF.	It is comparatively more stronger than 3NF.
3.	In 3NF the functional dependencies are already in 1NF and 2NF.	In BCNF the functional dependencies are already in 1NF, 2NF and 3NF.
4.	The redundancy is high in 3NF.	The redundancy is comparatively low in BCNF.
5.	In 3NF there is preservation of all functional dependencies.	In BCNF there may or may not be preservation of all functional dependencies.
6.	It is comparatively easier to achieve.	It is difficult to achieve.

THANK YOU!!!

My Blog : https://anandgharu.wordpress.com/

Email : gharu.anand@gmail.com