International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 3 Issue 5, August 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Impact of Normalization in Future

D. Gokila, S. BalaSubramani

Assistant Professor, Department of Computer Science,
Sri Krishna Adithya College of Arts and Science, Kovai Pudur, Coimbatore, Tamil Nadu, India

How to cite this paper: D. Gokila | S. BalaSubramani "Impact of Normalization in Future" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-5, August 2019, pp.153-156, **https://doi.org/10.31142/ijtsrd25128**

IJTSRD25128

ABSTRACT

Data and functionality are two primary aspects of systems. Unfortunately, there is a mental gap between these two aspects. Therefore, nowadays many are looking for the corresponding research and development fields as quite distinct with different terminology, tools, problems, processes,methods and best practices.

Normalization is a process of organizing the data in database to keep away from data redundancy, insertion anomaly, update abnormality & removingabnormality or it is the process of minimizing **redundancy** from a relation or set of relations. Redundancy in relation may cause insertion, deletion and updation of anomalies. So, it helps to minimize the redundancy in relations. **Normal** forms are used to remove or reduce redundancy in database tables.

It was first proposed by Edgar F. Codd (1970) as an integral part of his relational model.Codd defined the 2NF and 3NF in 1971.

This entails to organize the columns (attributes) and tables (relations) of a database to make sure that their dependencies are suitably imposed by database integrity constraints. It is proficient by applying some formal rules either by a process of *synthesis* (creating a new database design) or *decomposition* (improving an existing database design). Normalization has four basic thrusts today:

The first is for consciousness-raising. Normalization will help us dislodge some of the prejudices and biases that both we and the general society at large hold against people who are different. Unless we surface these massive, deeply held, often unconscious beliefs about differentness, as they are directed towards those labeled retarded in our society, we will make very slow headway in transforming social institutions

Normalization usually involves isolating a database into two or more tables and defining relationships between the tables. The objective is to isolate data so that adding, deleting, and modifying a field can be made in just one table and then propagated through the rest of the database via the defined relationships.

Here are the most commonly used normal forms:

- First normal form(1NF)
- Second normal form(2NF)
- Third normal form(3NF)
- Boyce & Codd normal form (BCNF)

First normal form (1NF)

As per the rule of first normal form, an attribute (column) of a table cannot hold many values. It should hold only atomic values, it violates first normal form or a relation is in first normal form if it does not contain any composite or multi-valued attribute. A relation is in first normal form if and if only every attribute in that relation is singled valued attribute.

Example: Company wants to store the names and contact details of employees.

emp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
102	Jon	Kanpur	8812121212
			9900012222
103	Ron	Chennai	7778881212
104	Lester	Bangalore	9990000123
			8123450987

Two employees (Jon & Lester) are having two mobile numbers so the company stored them in the same field as you can see in the table above.

This table is not in 1NF as the rule says “each attribute of a table must have atomic (single) values”, the emp_mobile values for employees Jon & Lester violates that rule.

To make the table to complies with 1NF:

emp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
102	Jon	Kanpur	8812121212
102	Jon	Kanpur	9900012222
103	Ron	Chennai	7778881212
104	Lester	Bangalore	9990000123
104	Lester	Bangalore	8123450987

Second normal form (2NF)

In 2NF the following conditions hold:

- Table is in 1NF (First normal form)
- No non-prime attribute is reliant on the proper subset of any candidate key of table.

An attribute that is nota part of any candidate key is known as non-prime attribute...

Partial Dependency – If proper subset of candidate key whichdetermine non-prime attribute, it is called partial dependency.

- Example 1 – In relation STUDENT_COURSE given in Table 3,
- FD set: {COURSE_NO->COURSE_NAME}
- Candidate Key: {STUD_NO, COURSE_NO}

In FD COURSE_NO->COURSE_NAME, COURSE_NO (proper subset of candidate key) is determining COURSE_NAME (non-prime attribute). Hence, it is partial dependency and relation is not in second normal form.

To convert it to second normal form, we will decompose the relation STUDENT_COURSE (STUD_NO, COURSE_NO, COURSE_NAME) as:
STUDENT_COURSE (STUD_NO, COURSE_NO)
COURSE (COURSE_NO, COURSE_NAME)

Note – This decomposition will be lossless join decomposition as well as dependency preserving.

- Example 2 – Consider following functional dependencies in relation R (A, B , C, D)
- AB -> C [A and B together determine C]

BC ->D [B and C together determine D]

In the above relation, AB is the only candidate key and there is no partial dependency, i.e., any proper subset of AB doesn't determine any non-prime attribute.

Example 3:A school wants to store the data of teachers and the subjects that they teach. Since a teacher can teach more than a subject, the table can have multiple rows for a teacher.

teacher_id	Subject	teacher_age
111	Maths	38
111	Physics	38
222	Biology	38
333	Physics	40
333	Chemistry	40

Candidate Keys: {teacher_id, subject}
Non prime attribute: teacher_age

The table is in 1 NF because each attribute has atomic values. However, it is not in 2NF because non prime attribute teacher_age is reliant on teacher_id alone which is a proper subset of candidate key. This violates the rule for 2NF as the rule says "no non-prime attribute is dependent on the proper subset of any candidate key of the table".

To make the table complies with 2NF we can break it in two tables like this:
teacher_details table:

teacher_id	teacher_age
111	38
222	38
333	40

Teacher_subject table:

teacher_id	subject
111	Maths
111	Physics
222	Biology
333	Physics
333	Chemistry

Now the tables comply with Second normal form (2NF).
Third Normal form (3NF)

In 3NF the following conditions holds:

- Table should be in 2NF
- Transitive functional dependency of non-prime attribute is on any super key should be removed.

An attribute which is not part of any candidate key is known as non-prime attribute.

In other words 3NF can be explained as: A table is in 3NF if it is in 2NF and for each functional dependency X-> Y have at least one of the following conditions:

- X is a super key of table
- Y is a prime attribute of table

An attribute which is a part of one of the candidate keys is known as prime attribute.

Transitive dependency – If A->B and B->C are two FDs then A->C is called transitive dependency.

- Example 1 – In relation STUDENT given in Table 4,

FD set: {STUD_NO -> STUD_NAME, STUD_NO -> STUD_STATE, STUD_STATE -> STUD_COUNTRY, STUD_NO -> STUD_AGE, and STUD_STATE -> STUD_COUNTRY}

Candidate Key: {STUD_NO}
For this relation in table 4, STUD_NO -> STUD_STATE and STUD_STATE -> STUD_COUNTRY are true. So STUD_COUNTRY is transitively dependent on STUD_NO. It violates third normal form. To convert it in third normal form, we will decompose the relation

STUDENT (STUD_NO, STUD_NAME, STUD_PHONE, STUD_STATE, STUD_COUNTRY_STUD_AGE) as:
STUDENT (STUD_NO, STUD_NAME, STUD_PHONE, STUD_STATE, STUD_AGE)
STATE_COUNTRY (STATE, COUNTRY)

- Example 2 – Consider relation R(A, B, C, D, E)

A -> BC,
CD -> E,
B -> D,
E -> A

All possible candidate keys in above relation are {A, E, CD, BC} All attribute are on right sides of all functional dependencies are prime.

Example 3: If a company wants to store the complete address of each and every employee.

emp_id	emp_name	empzip	emp_state	emp_city	emp_district
1001	John	282005	UP	Agra	Dayal Bagh
1002	Ajeet	222008	TN	Chennai	M-City
1006	Lora	282007	TN	Chennai	Sowkarpet
1101	Lilly	292008	UK	Pauri	Bhagwan
1201	Steve	222999	MP	Gwalior	Ratan

Super keys: {emp_id}, {emp_id, emp_name}, {emp_id, emp_name, emp_zip}…so on
Candidate Keys: {emp_id}
Non-prime attributes: all attributes except emp_id are non-prime as they are not part of any candidate keys.

Here, emp_state, emp_city & emp_district dependent on emp_zip. And, emp_zip is dependent on emp_id that makes non-prime attributes (emp_state, emp_city & emp_district) transitively dependent on super key (emp_id). This violates the rule of 3NF.

To make this table complies with 3NF we have to break the table into two tables to remove the transitive dependency:
Employee table:

emp_id	emp_name	emp_zip
1001	John	282005
1002	Ajeet	222008
1006	Lora	282007
1101	Lilly	292008
1201	Steve	222999

Employee_zip table:

emp_zip	**emp_state**	**emp_city**	**emp_district**
282005	UP	Agra	Dayal Bagh
222008	TN	Chennai	M-City
282007	TN	Chennai	Sowkarpet
292008	UK	Pauri	Bhagwan
222999	MP	Gwalior	Ratan

Boyce Codd normal form (BCNF)

It is aprogressadaptation of 3NF that's why it is also termed as 3.5NF. BCNF is stricter than 3NF. A table which complies with BCNF if it is in 3NF and for every functional dependency X->Y, X should be termed as super key of the table.

- Example 1 –For example consider relation R(A, B, C)

A -> BC,
B ->

A and B both are super keys so above relation is in BCNF.
Key Points –

1. BCNF is free of redundancy.
2. If a relation is in BCNF, then 3NF is also satisfy.
3. If all attributes of relation are the prime attribute, then the relation is alwaysconsider as 3NF.
4. A relation in a Relational Database is always bave at least in 1NF form.
5. Every Binary Relation (a Relation with only 2 attributes) always in BCNF.
6. If a Relation has only single candidate key, then the Relation is always in 2NF (because no Partial functional dependency possible).
7. Sometimes going for BCNF form may not maintain functional dependency. In such case move for BCNF only if the lost FD(s) is not required, else normalize till 3NF.
8. There are many more Normal forms that exist after BCNF, like 4NF and more. But in real world database systems generally not required to go further than BCNF.

Example 2: There is a company wherein employees work in more than a department.

emp_id	emp_nationality	emp_dept	dept_type	dept_no_of_emp
1001	Austrian	Production and planning	D001	200
1001	Austrian	Stores	D001	250
1002	American	design and technical support	D134	100
1002	American	Purchasing department	D134	600

Functional dependencies in the table above:
emp_id -> emp_nationality
emp_dept -> {dept_type, dept_no_of_emp}
Candidate key: {emp_id, emp_dept}

The table is not in BCNF as neither emp_id nor emp_dept alone are keys.
To make the table comply with BCNF we can break the table in three tables like this:

emp_nationality table:

emp_id	**emp_nationality**
1001	Austrian
1002	American

Emp_dept table:

emp_dept	dept_type	dept_no_of_emp
Production and planning	D001	200
stores	D001	250
design and technical suppo	D134	100
Purchasing department	D134	600

Emp_dept_mapping table:

emp_id	emp_dept
1001	Production and planning
1001	Stores
1002	Design and technical support
1002	Purchasing department

Functional dependencies:
emp_id -> emp_nationality
emp_dept -> {dept_type, dept_no_of_emp}
Candidate keys:
For first table: emp_id
for second table: emp_dept
for third table: {emp_id, emp_dept}

This is now in BCNF as in both the functional dependencies left side part is a key.

Should I Normalize?

While database normalization is a good idea, it's not an absolute requirement. In fact, there are many cases where purposely violating the rules of normalization is a good practice.

If you'd like to make sure your database is normalized, start with learning how to put your database into First Normal Form.

CONCLUSION:
Normalization is a fundamental tool to initially indoctrinate and train all potential human service workers … physicians, nurses, therapists, teachers, administrators, anybody in the human services embarking on their educational course. Technology must derive from the normalization concerns, and not vice versa. Sadly, technology today, as we know it, is so entrenched in attitudes and practices which dehumanize and devalue people served that normalization, taught apart from the core curriculum, becomes rhetoric to cloak business as usual. 4. Finally, normalization, or the socio-developmental model of growth, provides one of the most coherent and systematic ideologies to light the road for all human services: a guide, a direction in an era of turmoil, arbitrary scientific innovation, grass roots disenfranchisement and moral bankruptcy of so many of our professions.

Hence we have seen how normalization can decrease duplications, increase efficiencies by implementing the four levels of normalization forms.

The first three NF's are generallyenough for most small to medium size applications. Thus table structure is always in a certain normal form.

REFERENCES:

[1] CJ DATE" An Introdution to Database Systems"

[2] S. K. Singh "Database Systems"

[3] Henry K. Forth "Database Systems and Concepts"

REFERENCE LINKS:

1. http://www.bbc.com/future/story
2. https://mn.gov/mnddc/parallels2/
3. https://www.guru99.com/database-normalization.html
4. https://www.studytonight.com/dbms/database-normalization.php