

Normalization in Relational Databases

What is DB Normalization

- Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization rules divides larger tables into smaller tables and links them using relationships.
- The purpose of Normalisation in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.
- The inventor of the relational model Edgar Codd proposed the theory of normalization of data with the introduction of the First Normal Form, and he continued to extend theory with Second and Third Normal Form.
- Later he joined Raymond F. Boyce to develop the theory of Boyce-Codd Normal Form.

DB Normal Forms

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

Most of the times, Normalization is at its best in **3rd Normal Form**

Normalization by Case Study

A video library maintains a database of movies rented out. Without any normalization in database, all information is stored in one table!

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean, Clash of the Titans	Ms.
Robert Phil	3 rd Street 34	Forgetting Sarah Marshal, Daddy's Little Girls	Mr.
Robert Phil	5 th Avenue	Clash of the Titans	Mr.

What do you observe? Attribute (column) "Movies Rented" has multiple values. Let's transform it into 1st NF!

1st Normal Form

- Each attribute/column in the Table should contain a single value
- Each record/tuple should be unique

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3 rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3 rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5 th Avenue	Clash of the Titans	Mr.

What do you observe? What's the Primary Key?

We need a Primary Key

- A primary key is a single column-value used to identify a tuple (record) uniquely
 - It cannot be NULL
 - Its value must be UNIQUE
 - Its value should never change –or very very rarely
 - The primary key must be instantiated –be given a value- when a new record is inserted/created
- What is a Composite Key?
 - It is a Primary Key built out of “multiple” columns
 - i.e. Robert Phil3rd Street 34, Robert Phil5th Avenue
 - How clear is it, though?

2nd Normal Form

- Must be in 1st NF
- Single-column Primary Key is not functionally dependent on any subset of candidate key relation

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3 rd Street 34	Mr.
3	Robert Phil	5 th Avenue	Mr.

Foreign Key

MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

What is the solution? Splitting one table into two tables!!!

Functional Dependencies (FDs)

- A FD is a constraint that determines the relation of one attribute to another attribute in a Database Management System (DBMS).
- Functional Dependency assists in maintaining the quality of data in the database.
- It plays a vital role to find the difference between good and bad database design.
- The functional dependency of X on Y is represented as $X \rightarrow Y$

- If we know the value of Employee Number, we can obtain Employee name, city, salary.

EmpNum \rightarrow EmpName, Salary, City

- FDs could be:
 - Multivalued
 - Trivial
 - Non-Trivial
 - **Transitive**

Employee number	Employee Name	Salary	City
1	Dana	50000	San Francisco
2	Francis	38000	London
3	Andrew	25000	Tokyo

Transitive Functional Dependencies (TFDs)

- A TFD occurs when changing a non-key column, might cause any of the other non-key columns to change, too!

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3 rd Street 34	Mr.
3	Robert Phil	5 th Avenue	Mr.

Change in Name (under row 3, Full Names) → *May Change Salutation* (under row 3, Salutation)

Changing the non-key column Full Names may change Salutation

3rd Normal Form

- Must be in 2nd NF
- Has not transitive functional dependencies

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION ID
1	Janet Jones	First Street Plot No 4	2
2	Robert Phil	3 rd Street 34	1
3	Robert Phil	5 th Avenue	1

MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

SALUTATION ID	SALUTATION
1	Mr.
2	Ms.
3	Mrs.
4	Dr.

- Split tables again. Create one for Salutations
- There are not transitive functional dependencies, hence our tables are in 3rd NF
- Has not transitive functional dependencies

There is more

BCNF (Boyce-Codd Normal Form)

- Even when a database is in 3rd Normal Form, still there would be anomalies resulted if it has more than one Candidate Key.

4NF (Fourth Normal Form) Rules

- If no database table instance contains two or more, independent and multivalued data describing the relevant entity, then it is in 4th Normal Form.

5NF (Fifth Normal Form) Rules

- A table is in 5th Normal Form only if it is in 4NF and it cannot be decomposed into any number of smaller tables without loss of data.

6NF (Sixth Normal Form)

- 6th Normal Form is not standardized, yet however, it is being discussed by database experts for some time. Hopefully, we would have a clear & standardized definition for 6th Normal Form soon enough

Summarising

- Database designing is critical to the successful implementation of a database management system that meets the data requirements of an enterprise system.
- Normalization in DBMS is a process which helps produce database systems that are cost-effective and have better security models.
- Functional dependencies are a very important component of the normalize data process
 - Most database systems are normalized database up to the third normal forms in DBMS.
- A **primary key uniquely** identifies a record in a Table and **cannot be null**
- A foreign key helps connect tables and **it does reference** a primary key

References

- O. Appel. Instructor's material
- Jeffrey D. Ullman. Principles of Database and Knowledge-Base Systems
- C.J. Date. An Introduction to Database Systems
- Guru99 Web Site