DAMG 6210
Database Design

Nik Bear Brown

@NikBearBrown

Normal Forms

Topics

- Database Normalization
  - Data Anomalies Caused by:
    - Update, Insertion, Deletion
- Brief History/Overview
- 1st Normal Form
- 2nd Normal Form
- 3rd Normal Form
- Conclusion

Database Normalization

- The main goal of Database Normalization is to restructure the logical data model of a database to:
- Eliminate redundancy
- Organize data efficiently
- Reduce the potential for data anomalies.

Normal Forms

- 1st Normal Form
- 2nd Normal Form
- 3rd Normal Form
- BCNF
- 4th Normal Form
- 5th Normal Form
- Domain-Key Normal Form

Data Anomalies

- Data anomalies are inconsistencies in the data stored in a database as a result of an operation such as update, insertion, and/or deletion.
- Such inconsistencies may arise when have a particular record stored in multiple locations and not all of the copies are updated.

- We can prevent such anomalies by implementing 7 different level of normalization called Normal Forms (NF)
- We'll only look at the first three.

Brief History/Overview

- Database Normalization was first proposed by Edgar F. Codd.
- Codd defined the first three Normal Forms, which we'll look into, of the 7 known Normal Forms.

- In order to do normalization we must know what the requirements are for each of the three Normal Forms that we'll go over.
- One of the key requirements to remember is that Normal Forms are progressive. That is, in order to have 3rd NF we must have 2nd NF and in order to have 2nd NF we must have 1st NF.

Database Tables and Normalization

- Normalization (continued)
  - 2NF is better than 1NF; 3NF is better than 2NF
  - For most business database design purposes, 3NF is as high as we need to go in normalization process
  - Highest level of normalization is not always most desirable

The Normalization Process

- Each table represents a single subject
- No data item will be unnecessarily stored in more than one table
- All attributes in a table are dependent on the primary key

1st Normal Form - The Requirements

- The requirements to satisfy the 1st NF:
  - Each table has a primary key: minimal set of attributes which can uniquely identify a record
  - The values in each column of a table are atomic (No multi-value attributes allowed).
  - There are no repeating groups: two columns do not store similar information in the same table.

1st Normal Form

First Normal Form is violated if:

- The relation has no identifiable primary key.
- Any attempt has been made to store a multi-valued fact in a tuple.

Conversion to First Normal Form

- Repeating group
  - Derives its name from the fact that a group of multiple entries of same type can exist for any single key attribute occurrence
- Relational table must not contain repeating groups
- Normalizing table structure will reduce data redundancies
- Normalization is three-step procedure

Repeating groups

A repeating group is an array of data

A field fav games with “Grand Theft Auto V, Tekken 7, Assassin's Creed Unity”

Or three fields:

i. Fav_games_1
ii. Fav_games_2
iii. Fav_games_3

Conversion to First Normal Form

- Step 1: Eliminate the Repeating Groups
  - Present data in tabular format, where each cell has single value and there are no repeating groups
  - Eliminate repeating groups, eliminate nulls by making sure that each repeating group attribute contains an appropriate data value

Conversion to First Normal Form

- Step 2: Identify the Primary Key
  - Primary key must uniquely identify attribute value
  - New key must be composed

Conversion to First Normal Form

- Step 3: Identify All Dependencies
  - Dependencies can be depicted with help of a diagram
  - Dependency diagram:
    - Depicts all dependencies found within given table structure
    - Helpful in getting bird’s-eye view of all relationships among table’s attributes
    - Makes it less likely that will overlook an important dependency

Conversion to First Normal Form

- First normal form describes tabular format in which:
  - All key attributes are defined
  - There are no repeating groups in the table
  - All attributes are dependent on primary key
- All relational tables satisfy 1NF requirements
- Some tables contain partial dependencies
  - Dependencies based on only part of the primary key
  - Sometimes used for performance reasons, but should be used with caution
  - Still subject to data redundancies

Summary: 1NF

- A relation is in 1NF if it contains no repeating groups
- To convert an unnormalised relation to 1NF either:
- Flatten the table and change the primary key, or
- Decompose the relation into smaller relations, one for the repeating groups and one for the non-repeating groups.
- Remember to put the primary key from the original relation into both new relations.
- This option is liable to give the best results.

2nd Normal Form - The Requirements

- The requirements to satisfy the 2nd NF:
  - All requirements for 1st NF must be met.
  - Redundant data across multiple rows of a table must be moved to a separate table.
    - The resulting tables must be related to each other by use of foreign key.

2nd Normal Form

Second Normal Form is violated
if:

- First Normal Form is violated
- If there exists a non-key field(s) which is functionally dependent on a partial key.

partial key non-key

2nd Normal Form

- No calculated fields
- Non-key attributes must be dependent on the key(s) but NOT necessarily only on the key(s)

Conversion to Second Normal Form

- Relational database design can be improved by converting the database into second normal form (2NF)
- Two steps

Conversion to Second Normal Form

- Step 1: Write Each Key Component on a Separate Line
  - Write each key component on separate line, then write original (composite) key on last line
  - Each component will become key in new table

Conversion to Second Normal Form

- Step 2: Assign Corresponding Dependent Attributes
  - Determine those attributes that are dependent on other attributes
  - At this point, most anomalies have been eliminated

Conversion to Second Normal Form

- Table is in second normal form (2NF) when:
  - It is in 1NF and
  - It includes no partial dependencies:
    - No attribute is dependent on only portion of primary key

Normalisation 2NF: Second Normal Form Example

Normalisation 2NF: Second Normal Form Example:
https://youtu.be/8PwomfwMMyQ

Summary: 2NF

A relation is in 2NF if it contains no repeating groups and no partial key functional dependencies

- Rule: A relation in 1NF with a single key field must be in 2NF
- To convert a relation with partial functional dependencies to 2NF. create a set of new relations:
- One relation for the attributes that are fully dependent upon the key.
- One relation for each part of the key that has partially dependent attributes

3rd Normal Form - The Requirements

- The requirements to satisfy the 3rd NF:
  - All requirements for 2nd NF must be met.
  - Eliminate fields that do not depend on the primary key;
    - That is, any field that is dependent not only on the primary key but also on another field must be moved to another table.

3rd Normal Form

Third Normal Form is violated if:

- Second Normal Form is violated
- If there exists a non-key field(s) which is functionally dependent on another non-key field(s).

non-key → non-key

Note: A candidate key is not a non-key field.

Conversion to Third Normal Form

- Data anomalies created are easily eliminated by completing three steps
- Step 1: Identify Each New Determinant
  - For every transitive dependency, write its determinant as PK for new table
    - Determinant
      - Any attribute whose value determines other values within a row

Conversion to Third Normal Form

- Step 2: Identify the Dependent Attributes
  - Identify attributes dependent on each determinant identified in Step 1 and identify dependency
  - Name table to reflect its contents and function

Conversion to Third Normal Form

- Step 3: Remove the Dependent Attributes from Transitive Dependencies
  - Eliminate all dependent attributes in transitive relationship(s) from each of the tables that have such a transitive relationship
  - Draw new dependency diagram to show all tables defined in Steps 1–3
  - Check new tables as well as tables modified in Step 3 to make sure that each table has determinant and that no table contains inappropriate dependencies

Conversion to Third Normal Form

- A table is in third normal form (3NF) when both of the following are true:
  - It is in 2NF
  - It contains no transitive dependencies
  - Non-key attributes must be dependent on the key(s) but and only on the key(s)

Normalisation 3NF: Third Normal Form Example

- Normalisation 3NF: Third Normal Form Example: https://youtu.be/c7DXeY3aIJw

Summary: 3NF

- A relation is in 3NF if it contains no repeating groups, no partial functional dependencies, and no transitive functional dependencies
- To convert a relation with transitive functional dependencies to 3NF, remove the attributes involved in the transitive dependency and put them in a new relation
- Rule: A relation in 2NF with only one non-key attribute must be in 3NF
- In a normalised relation a non-key field must provide a fact about the key, the whole key and nothing but the key.
- Relations in 3NF are sufficient for most practical database design problems. However, 3NF does not guarantee that all anomalies have been removed

The Boyce-Codd Normal Form (BCNF)

- Every determinant in table is a candidate key
  - Has same characteristics as primary key, but for some reason, not chosen to be primary key
- When table contains only one candidate key, the 3NF and the BCNF are equivalent
- BCNF can be violated only when table contains more than one candidate key

The Boyce-Codd Normal Form (BCNF)

- Most designers consider the BCNF as special case of 3NF
- Table is in 3NF when it is in 2NF and there are no transitive dependencies
- Table can be in 3NF and fails to meet BCNF
  - No partial dependencies, nor does it contain transitive dependencies
  - A nonkey attribute is the determinant of a key attribute

Summary

- We have seen how Database Normalization can decrease redundancy, increase efficiency and reduce anomalies by implementing three of seven different levels of normalization called Normal Forms. The first three NF’s are usually sufficient for most small to medium size applications.

Summary

- Table is in 2NF when it is in 1NF and contains no partial dependencies
- Table is in 3NF when it is in 2NF and contains no transitive dependencies
- Table that is not in 3NF may be split into new tables until all of the tables meet 3NF requirements
- Normalization is important part—but only part—of design process

References

- SQL Normalization - The Basics - 1st, 2nd, 3rd Normal Form Software Engi...: https://youtu.be/ygfikznRjpw
- Implementing 1NF, 2NF, 3NF: https://youtu.be/G9SA0Yv-o28
- Normalisation 1NF: Understanding and applying First Normal Form: https://youtu.be/HHDH6N_qjm4
- Normalisation 2NF: Second Normal Form Example: https://youtu.be/8PwomfwMMyQ

References

- Hillyer Mike, MySQL AB. An Introduction to Database Normalization, http://dev.mysql.com/tech-resources/articles/intro-to-normalization.html, accessed October 17, 2006.
- Microsoft. Description of the database normalization basics, http://support.microsoft.com/kb/283878 , accessed October 17, 2006.
- Wikipedia. Database Normalization. http://en.wikipedia.org/wiki/Database_normalization.html , accessed October 17, 2006.