Normalization of Databases

By Krishnan Ramakrishnan
Senior Presentation (Jul 2017)
San Francisco State University

Outline

- **Introduction**
- First Normal Form
- Second Normal Form
- Third Normal Form
- Conclusion

Introduction

- Database normalization is a process of transforming a data model in such a way that performs and scales a physical implementation
- The main objective of normalization is as follows:
  - Eliminates redundancy of data elements (Storage efficiency)
  - Eliminates or reduces data anomalies
  - Enables efficient scaling of application functionality
- Database normalization was first proposed by Edgar F. Codd.
- In order to do normalization, we must understand the requirements in order to normalize a table.
- Database normalization is progressive. That is, in order to have a 3rd normal form, we must have a 2nd normal form and to have a 2nd normal form, we must have a 1st normal form.

Introduction (Ctd.)

- Redundancy:
  - The normalized design will ensure that the same data element is stored only once in one of the tables.
  - Example: An employee's first and last name can only be stored in the employee table and nowhere else
  - This will also result in an efficiency in space usage.
- Anomaly:
  - There are 3 types of anomalies: Insert, Update, and Delete
    - An insert anomaly occurs when some information cannot be inserted without the presence of others.
    - An update anomaly occurs when one or more instances of duplicated data are updated.
    - A delete anomaly occurs when certain attributes are lost due to the deletion of other attributes.

Introduction (Ctd.)

Student-Course Table

StudentNum	StudentName	CourseNum	CourseName
S15	James	80571	Java
S15	James	80595	Java
S37	Jason	80595	Python
S50	Brian	86927	C#
S50	Brian	89682	Databases

- Here, CourseName and StudentName are redundantly stored.
- Both student names and course names can be inconsistent across various records.
- New courses cannot be inserted unless a student is enrolled in it.
- New students cannot be entered unless they are enrolled in a course.
- Courses of students cannot be deleted without deleting the courses itself.

Outline

- Introduction
- **First Normal Form**
- Second Normal Form
- Third Normal Form
- Conclusion

First Normal Form

- In a relational database, an object is in first normal form if and only if it satisfies the following conditions:
- Three rules of first normal form:
  - Each table must contain a Primary Key that uniquely identifies each row in a table.
  - Values in each column has to be atomic, meaning one column cannot have multiple values.
  - There cannot be any repeating groups of attributes.
- If the above conditions are satisfied, the tables are considered to be in first normal form

First Normal Form - 1

- Original Table:

Customer ID	Customer Name	Phone Number	Shipping Address	Billing Address
5297	Christian	611-263-3631, 512-258-2611	1 First Ave CA 94544	2 First St CA 94549
7518	Paul	319-126-2418	53 Boscell Ln NV 89085	21 Lindell St WA 98081
1504	Isaac	328-479-1018	61 Lund Ln WA 98094	73 Berry St NV 89096
8362	James	998-121-3161	701 124st St IL 60062	718 179th Rd IL 60038
3158	Jane	832-123-4567, 928-391-2315	108 FDR Dr NY 10030	238 Queens Ave NY 10092
4179	Shannon	661-231-2812	212 Kobe Pl FL 32382	807 Bryant Ave FL 32719
6827	Alexander	141-687-3177	34 Surrey Rd NJ 07038	5 Borgata Way NJ 07312
3053	Ivan	177-238-4900	17 Lima Way PA 15038	318 Chance Way PA 15001
7397	David	434-212-2385, 418-237-5512	327 J St MA 01088	41 Harmon Ln MA 01182

  - This table is not normalized:
    - Multi-value violation: The phone number column violates this rule.
    - Both the Shipping Address and Billing Address columns violate repeating group rules.
    - At present, there is no Primary Key violation.

First Normal Form - 2

- Step 1: Normalizing multi-value columns.
- Multi-column value violations is addressed by normalizing each of the values in the multi-column into a separate row along with the rest of the attributes.

Customer ID	Customer Name	Phone Number
5297	Christian	611-263-3631
5297	Christian	512-258-2611
7518	Paul	319-126-2418
1504	Isaac	328-479-1018
8362	James	998-121-3161
3158	Jane	832-123-4567
3158	Jane	928-391-2315
4179	Shannon	661-231-2812
6827	Alexander	141-687-3177
3053	Ivan	177-238-4900
7397	David	434-212-2385
7397	David	418-237-5512

- The multi-value column violation has been fixed, however, we see that it has led to a Primary Key violation. We will resolve this in the next step.

First Normal Form – 3

- Step 1 (Ctd.): The Primary Key violation is resolved by removing the phone numbers to a separate table.

Customer Table

Customer ID	Customer Name
5297	Christian
7518	Paul
1504	Isaac
8362	James
3158	Jane
4179	Shannon
6827	Alexander
3053	Ivan
7397	David

Customer-Phone Table

Customer ID	Phone Type	Phone Number
5297	Home	611-263-3631
5297	Cell	512-258-2611
7518	Home	319-126-2418
1504	Home	328-479-1018
8362	Home	998-121-3161
3158	Home	832-123-4567
3158	Cell	928-391-2315
4179	Home	661-231-2812
6827	Home	141-687-3177
3053	Home	177-238-4900
7397	Home	434-212-2385
7397	Cell	418-237-5512

- The Primary Key violation of the Customer table has been resolved.
- We can now see that there's no constraint on defining any number of phone numbers for a given customer.

First Normal Form - 4

- Step 2: Take the repeating groups of elements and put it into a separate table to resolve the repeating group violation on Address

Customer-Address Table

Customer ID	Address Type	Address
5297	Shipping	1 First Ave CA 94544
5297	Billing	2 First St CA 94549
7518	Shipping	53 Boscell Ln NV 89085
7518	Billing	21 Lindell St WA 98081
1504	Shipping	61 Lund Ln WA 98094
1504	Billing	73 Berry St NV 89096
8362	Shipping	701 124st St IL 60062
8362	Billing	718 179th Rd IL 60038
3158	Shipping	108 FDR Dr NY 10030
3158	Billing	238 Queens Ave NY 10092
4179	Shipping	212 Kobe Pl FL 32382
4179	Billing	807 Bryant Ave FL 32719
6827	Shipping	34 Surrey Rd NJ 07038
6827	Billing	5 Borgata Way NJ 07312
3053	Shipping	17 Lima Way PA 15038
3053	Billing	318 Chance Way PA 15001
7397	Shipping	327 J St MA 01088
7397	Billing	41 Harmon Ln MA 01182

- We apply a similar logic to Step 1 to eliminate repeating groups of Billing and Shipping Addresses
- We have a one-to-many relationships in the Customer-Phone table, because 1 customer can have multiple phone numbers.
- Similar to the Customer-Phone table, we can clearly see the advantages of keeping the addresses organized in a separate table.

First Normal Form - 5

Customer Table

Customer ID	Customer Name
5297	Christian
7518	Paul
1504	Isaac
8362	James
3158	Jane
4179	Shannon
6827	Alexander
3053	Ivan
7397	David

Customer-Phone Table

Customer ID	Phone Type	Phone Number
5297	Home	611-263-3631
5297	Cell	512-258-2611
7518	Home	319-126-2418
1504	Home	328-479-1018
8362	Home	998-121-3161
3158	Home	832-123-4567
3158	Cell	928-391-2315
4179	Home	661-231-2812
6827	Home	141-687-3177
3053	Home	177-238-4900
7397	Home	434-212-2385
7397	Cell	418-237-5512

Customer-Address Table

Customer ID	Address Type	Address
5297	Shipping	1 First Ave CA 94544
5297	Billing	2 First St CA 94549
7518	Shipping	53 Boscell Ln NV 89085
7518	Billing	21 Lindell St WA 98081
1504	Shipping	61 Lund Ln WA 98094
1504	Billing	73 Berry St NV 89096
8362	Shipping	701 124st St IL 60062
8362	Billing	718 179th Rd IL 60038
3158	Shipping	108 FDR Dr NY 10030
3158	Billing	238 Queens Ave NY 10092
4179	Shipping	212 Kobe Pl FL 32382
4179	Billing	807 Bryant Ave FL 32719
6827	Shipping	34 Surrey Rd NJ 07038
6827	Billing	5 Borgata Way NJ 07312
3053	Shipping	17 Lima Way PA 15038
3053	Billing	318 Chance Way PA 15001
7397	Shipping	327 J St MA 01088
7397	Billing	41 Harmon Ln MA 01182

- Now, the original data is in first normal form, each table satisfies all 3 conditions.

Outline

- Introduction
- First Normal Form
- **Second Normal Form**
- Third Normal Form
- Conclusion

Second Normal Form

- The second normal form defines the rules of attribute dependence on the Primary Key. Data is suppose to be in the second normal form if it satisfies the following conditions:
  - The table must be in first normal form.
  - All Non-Key attributes are fully functional dependent on each element of the primary key.
- All attributes that make up the Primary Key are called key attributes. The rest of the attributes are called Non-Key attributes.
- The second level of normalization is also known as key dependency

Second Normal Form (Example)

CustomerID	StoreID	Purchase Location	Transaction Date
1	1	Seattle	10/5/2015
2	1	Seattle	12/20/2015
3	2	Bellevue	3/7/2015
1	2	Bellevue	4/30/2015
2	3	Redmond	5/24/2015
4	2	Bellevue	1/14/2015
5	3	Redmond	9/18/2015

- In this table, both CustomerID and StoreID are the Primary Key.
- The Non-Key attribute is Purchase Location which depends on StoreID that is only part of the Primary Key and violates the second normal form.
- The transaction date is dependent upon both CustomerID and StoreID and is in the second normal form.

Second Normal Form (Example)

- Converting data to second normal form:

Table Purchase

CustomerID	StoreID	Transaction Date
1	1	10/5/2015
2	1	12/20/2015
3	2	3/7/2015
1	2	4/30/2015
2	3	5/24/2015
4	2	1/14/2015
5	3	9/18/2015

Table Store

StoreID	Purchase Location
1	Seattle
2	Bellevue
3	Redmond

- We take the violating location name and create a separate table containing both the store location and StoreID.
- Now, the original data is in second normal form, each table satisfies the two conditions.

Outline

- Introduction
- First Normal Form
- Second Normal Form
- **Third Normal Form**
- Conclusion

Third Normal Form

- The third normal form defines the rules of transitive functional dependency. Data is suppose to be in the third normal form if it satisfies the following conditions:
  - The table must be in second normal form
  - There doesn’t exist a transitive functional dependency between any of the attributes in the table. Simply put, no two Non-Key attributes can have a dependency.

Third Normal Form (Example)

BookID	GenreID	Genre Type	Price
1	1	Engineering	$40.99
2	2	Mathematics	$57.99
3	3	Government	$60.00
4	2	Mathematics	$100.99
5	3	Government	$299.99
6	2	Mathematics	$25.00
7	1	Engineering	$80.95

- The Primary Key of this table is BookID.

The GenreID qualifies the BookID and is dependent on BookID.

- Price is also dependent on BookID
- However, the GenreType is dependent on only the GenreID which happens to be a Non-Key attribute.
- Moreover, this is a transitive dependency that violates the third normal form.

Third Normal Form (Example)

- How do we normalize the table using a 3NF?

Table Book

BookID	GenreID	Price
1	1	$40.99
2	2	$57.99
3	3	$60.00
4	2	$100.99
5	3	$299.99
6	2	$25.00
7	1	$80.95

Table Genre

GenreID	Genre Type
1	Engineering
2	Mathematics
3	Government

- As the GenreType has a transitive dependency violation, we separate that to a separate table.
- Now, both tables are in third normal form.

Outline

- Introduction
- First Normal Form
- Second Normal Form
- Third Normal Form
- **Conclusion**

Conclusion

- We have seen how database normalization can decrease redundancy, increase efficiency, and reduce anomalies.
- It is extremely important for any relational database design to conform to the basic normalization rules.
- Relational database designs that have not been optimally normalized has often ran into issues on scalability, performance, and storage.

References Used

- http://www.1keydata.com/database-normalization/second-normal-form-1nf.php
- http://www.1keydata.com/database-normalization/second-normal-form-2nf.php
- http://www.1keydata.com/database-normalization/second-normal-form-3nf.php
- https://www.sqa.org.uk/e-learning/MDBS01CD/page_24.htm
- https://www.sqa.org.uk/e-learning/MDBS01CD/page_22.htm
- https://www.sqa.org.uk/e-learning/MDBS01CD/page_23.htm