Institut Informatika & Bisnis DARMAJAYA
Yayasan Alfian Husin

SAINS DATA
Institut Informatika & Bisnis Darmajaya

Kampus Merdeka INDONESIA JAYA

MERDEKA BELAJAR

DATA SCIENCE DARMAJAYA
"YOUR BEST FUTURE IN DATA"

MEETING: [13 & 14]

NORMALIZATION

BY: HENDRA KURNIAWAN

[12-09-2023] | [KODE MK: SD23404 | MATAKULIAH: PENGANTAR BASIS DATA | SKS: 2/2 | VERSI: 01

Normalization

1. Functional Depedency
2. Determinat
3. Transitive Depedency
4. Partial Depedency
5. Normalization Steps

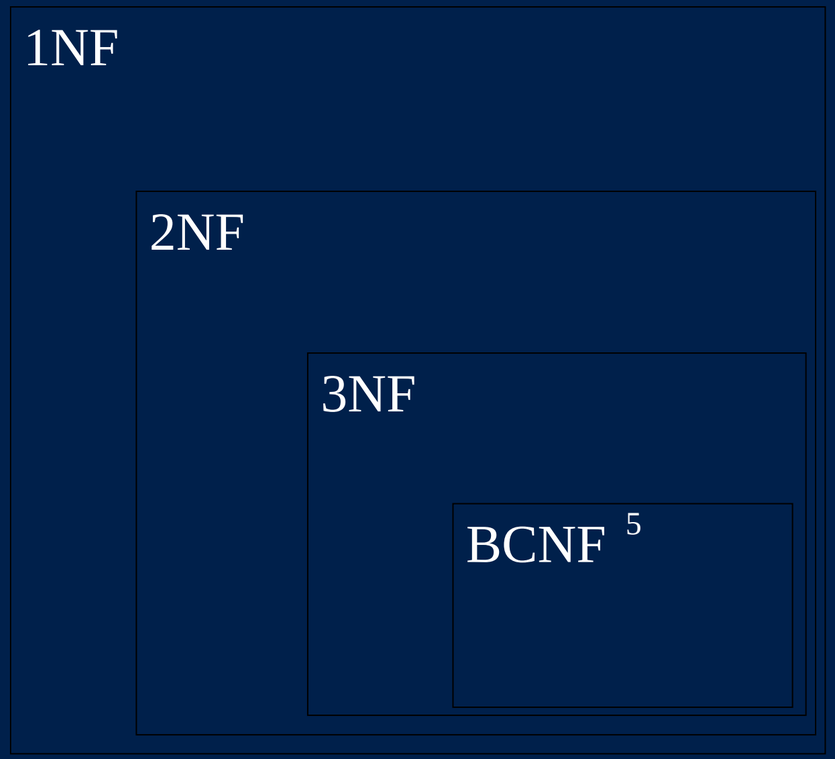
Learning Objectives

- The purpose of normalization.
- How normalization can be used when designing a relational database.
- The potential problems associated with redundant data in base relations.
- The concept of functional dependency, which describes the relationship between attributes.
- The characteristics of functional dependencies used in normalization.

Normalization is a process that “improves” a database design by generating relations that are of higher normal forms.

The *objective* of normalization:
“*to create relations where every dependency is on the key, the whole key, and nothing but the key*”.

Normalization



a relation in BCNF, is also in 3NF

a relation in 3NF is also in 2NF

a relation in 2NF is also in 1NF

Normalization

We consider a relation in BCNF to be fully normalized.

The benefit of higher normal forms is that update semantics for the affected data are simplified.

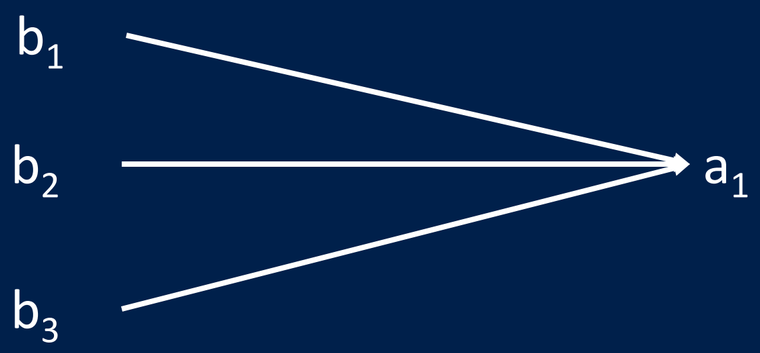
This means that applications required to maintain the database are simpler.

A design that has a lower normal form than another design has more redundancy. Uncontrolled redundancy can lead to data integrity problems.

First we introduce the concept of ***functional dependency***

Functional Dependence

An attribute A is functionally dependent on attribute(s) B if: given a value b for B there is one and only one corresponding value a for A (at a time).



Example: functional dependence

All sales representatives in a given pay class have the same commission rate.

Keys

Primary Key: a minimal set of attributes that form a candidate key

Any attribute or collection of attributes that functionally determine all attributes in a record is a Candidate Key.

Note: since no two rows in a relational table can be duplicates, the entire record is always a candidate key.

Primary Key (C)

- C determines all attributes
- No subset of the attributes in C is a candidate key

A key consisting of more than one attribute is called a "composite key."

Good Primary Keys

- Do not change over the life of the database
- Are not “intelligent keys”
- Are not too long
- Do not consist of too many attributes (3 or fewer is good)

Foreign Keys

A value in the “child” table that matches with the related value in the “parent” table.

SalesRep(SalesRepNumber, Name)

[03 | Mary Jones]

[124 | 03]

Customer(CustomerNumber, SalesRepNumber)

Foreign Keys

The foreign key in the child table has the same value as the primary key in the parent.

- The foreign key in a many-to-many relationship goes in the *many* table.
- In a many-to-many relationship, foreign keys from both tables go into an *associative entity.*
- In a 1-to-1 relationship the foreign key goes into *one* of the tables (usually the one most likely to change)

Normal Forms

- A set of conditions on table structure that improves maintenance. Normalization removes processing anomalies:
  - Update
  - Inconsistent Data
  - Addition
  - Deletion

Normal Forms

All attributes depend on the key, the whole key and nothing but the key.

1NF Keys and no repeating groups

2NF No partial dependencies

3NF All determinants are candidate keys

4NF No multivalued dependencies

1st Normal Form

- Table has a primary key
- Table has no repeating groups

A multivalued attribute is an attribute that may have several values for one record

A repeating group is a set of one or more multivalued attributes that are related

Example

- Multivalued attribute:

Orders(<u>OrderNumber</u>, OrderDate, {PartNumber})

[12491 | 9/02/2001 | BT04, BZ66]

- Repeating group:

Orders(OrderNumber, OrderDate, {PartNumber, NumberOrdered})

[12491 | 9/02/2001 | (BT04, 1), (BZ66, 1)]

Normalization: 1NF

- Every repeating group becomes a new table with the appropriate foreign key relationships preserved.
- Remove nested repeating groups from the outside in

Order(OrderNumber, OrderDate, {PartNumber, {Supplier}})

Example: 1NF

Order(OrderNumber, OrderDate, {PartNumber, {Supplier}})

Order(OrderNumber, OrderDate)

Order-Part(OrderNumber, PartNumber)

Part(PartNumber, {Supplier})

Example: 1NF (cont.)

Part(PartNumber, {Supplier})

Part(PartNumber)

Part-Supplier(PartNumber, SupplierNum)

Supplier(SupplierNum)

2nd Normal Form

- No partial dependencies

  No attribute depends on only some of the attributes of a concatenated key.

Order-Part

[OrderNumber | PartNumber | PartDescription]

Create a new table with PartNumber key.

3rd Normal Form / Boyce-Codd Normal Form

- 3rd Normal Form: no transitive dependencies

  Transitive dependency means that a non-key attribute depends on another non-key attribute(s).

  This definition says nothing about dependencies that involve the key.

3rd Normal Form / Boyce-Codd Normal Form

- BCNF: every determinant is a candidate key.

Determinant: any attribute(s) that functionally determine another attribute

BCNF means that there are no “transitive” dependencies involving key or non-key attributes.

3NF

- Pratt and Adamski use the BCNF definition as their definition of 3NF

BCNF was generated to deal with problems like:

Class(Section#, InstructorID, …) extra key attribute

(Student, Major, Advisor) wrong key

4th Normal Form

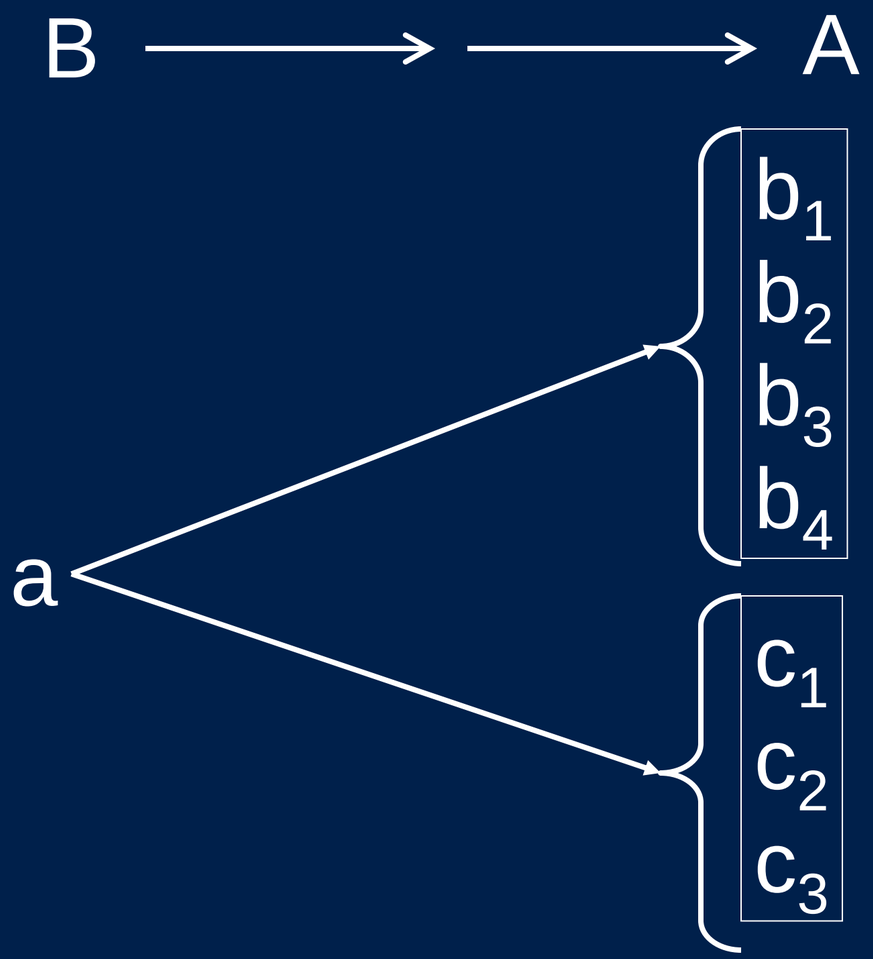
- No multivalued dependencies

  A multivalued dependency of column B on column A occurs when a table has a key with three or more attributes, (A, B, C) and

  – *each value of A is associated with a collection of values of B*

  – *this collection of values is independent of C*

B is multidependent on A



Normalization

- Improves maintenance for database changes

  Tends to slow down retrieval
- Better at finding problems than solving them

  Standard normalization procedures are subtle and may introduce BCNF or 4NF problems into tables

Normalization

1NF Keys & no repeating groups

2NF 1NF & all attributes depend on all key components

3NF 2NF & all determinants are candidate keys

4NF 3NF & no multivalued dependencies

Intuitive Normalization

1NF Tables represent entities

2NF Each table represents only one entity

3NF Tables do not contain attributes from embedded entities

4NF Triple relationships should not represent a pair of dual relationships

Intuitive Normalization

1NF Tables represent entities

2NF Each table represents only one entity

3NF Tables do not contain attributes from embedded entities

4NF Triple relationships should not represent a pair of dual relationships

CONCLUSION

Normalization is a crucial concept in database design that aims to eliminate redundancy, improve data integrity, and maintain data consistency within a relational database. It involves organizing data into separate related tables, each serving a specific purpose. By adhering to normalization principles, such as reducing data duplication and minimizing data anomalies, a database becomes more efficient, scalable, and less prone to errors.

In essence, normalization ensures that a database is structured in a way that minimizes data redundancy and enforces data integrity, resulting in a well-organized and maintainable database system. Learning and applying normalization principles is essential for designing effective and reliable database systems.

REFERENCES

1. Connoly. T., Begg. Carolyn. 2015. Database System: A Pratical Approach to Design, Implementation, and Management. Sixth Edition. Global Edition. Pearson.

Institut Informatika & Bisnis DARMAJAYA
Yayasan Alfian Husin

SAINS DATA
Institut Informatika & Bisnis Darmajaya

Kampus Merdeka INDONESIA JAYA

MERDEKA BELAJAR

THANK YOU!!

DATA SCIENCE DARMAJAYA “YOUR BEST FUTURE IN DATA”