
LUT
University
LAND
OF THE
CURIOUS

CT60A4304 - BASICS OF DATABASE SYSTEMS

NORMALIZATION

Lecture

Jiri Musto, D.Sc.

TABLE OF CONTENTS

WHAT IS NORMALIZATION?

- Normalization is the removal of redundant (overlapping, repetitive) data
- Basic rule of thumb:
  - Only store data from immediate proximity in one table/entity/relation
- Relational model made based on an ER-model is usually good enough
- Relational model based on an excel sheet most likely not if you do not have enough design experience

WHY NORMALIZE?

- Data is stored uniquely in their own locations
- Structure is logical and unambiguous
  - Every piece of data has a constant (unchanged) location
- Easy to query with SQL
- Normalization may fix problems in the database structure such as:
  - Extra attributes
  - Too many null values
  - Bad attribute names
  - Repetitive values

PROBLEMS WITH AN UNNORMALIZED TABLE

- Insertion anomalies
  - Cannot insert new data if all information is not available
- Deletion anomalies
  - Deletes all associated data even if you do not want to
- Update anomalies
  - May need to update multiple tuples even when changing just one piece of information

areaCode	areaName	salesCode	salesName	phoneNmbr	salesAddress	product1	quantity1	product2	Quantity2
A1	West-side	S1	Helen	04055750, 05011437	00660 hki	T1	120	T9	75
A1	West-side	S2	Jay	04055751	00630 hki	T2	99	T4	111
A1	West-side	S2	Jay	04055751	00630 hki	T5	234		
A2	East-side	S3	Simon	04059132	53100 lpr	T12	123		
A2	East-side	S4	Cathrine	04454974	55910 ima	T9	2234	T4	221

WHY NOT TO NORMALIZE?

- Overly normalized database may slow down queries
  - Need to join more tables together
  - Performance issues
- Normalizing existing database structure may take a considerable effort
  - Data warehouses
  - Large NoSQL databases
- Poorly done normalization may lead to loss of information
- Human error may cause issues
- If you have no need to enforce integrity rules or dependencies

THE MAIN CONCEPT OF NORMALIZATION: FUNCTIONAL DEPENDENCY (FD)

- X → Y
  - Y depends on X, X leads to Y, X determines Y
  - X is determinant, Y is dependent
- Trivial and non-trivial functional dependency
  - Trivial if X→Y and Y is a subset of X, non-trivial otherwise
- Transitive dependency
  - a→b & b→c = a→c
- Multivalued dependency
  - a → {b, c}, and not b→c

NORMAL FORMS, AS SEEN IN WIKIPEDIA

	UNF (1970)	1NF (1970)	2NF (1971)	3NF (1971)	EKNF (1982)	BCNF (1974)	4NF (1977)	ETNF (2012)	5NF (1979)	DKNF (1981)	6NF (2003)
Primary key (no duplicate tuples)[4]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Atomic columns (cells cannot have tables as values)[5]	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Every non-trivial functional dependency either does not begin with a proper subset of a candidate key or ends with a prime attribute (no partial functional dependencies of non-prime attributes on candidate keys)[5]	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Every non-trivial functional dependency either begins with a superkey or ends with a prime attribute (no transitive functional dependencies of non-prime attributes on candidate keys)[5]	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
Every non-trivial functional dependency either begins with a superkey or ends with an elementary prime attribute	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	N/A
Every non-trivial functional dependency begins with a superkey	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	N/A
Every non-trivial multivalued dependency begins with a superkey	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	N/A
Every join dependency has a superkey component[8]	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	N/A
Every join dependency has only superkey components	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	N/A
Every constraint is a consequence of domain constraints and key constraints	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Every join dependency is trivial	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

UNNORMALIZED FORM

- According to the relational model, the tuples need a primary key to be uniquely identify
- In unnormalized relation you may have:
  - Multivalued attributes or composite attributes
  - Repeating attribute names

Multivalued

Composite attribute

Repeating attributes

areaCode	areaName	salesCode	salesName	phoneNmbr	salesAddress	product1	quantity1	product2	Quantity2
A1	West-side	S1	Helen	04055750, 05011437	00660 hki	T1	120	T9	75
A1	West-side	S2	Jay	04055751	00630 hki	T2	99	T4	111
A1	West-side	S2	Jay	04055751	00630 hki	T5	234		
A2	East-side	S3	Simon	04059132	53100 lpr	T12	123		
A2	East-side	S4	Cathrine	04454974	55910 ima	T9	2234	T4	221

Primary key

FIRST NORMAL FORM

- In 1st normal form
  - Every attribute is atomic (no-composite attributes)
  - No multivalued attributes
  - Unique attribute names

salesCode	salesName	product	quantity
S1	Helen	T1	120
S1	Helen	T9	75
S2	Jay	T2	99
S2	Jay	T4	111
S2	Jay	T5	234
S3	Simon	T12	123
S4	Cathrine	T9	2234
S4	Cathrine	T4	221

areaCode	areaName	salesCode	phoneNmbr	salesName	postalCode	City
A1	West-side	S1	04055750	Helen	00660	Hki
A1	West-side	S1	05011437	Helen	00660	Hki
A1	West-side	S2	04055751	Jay	00630	Hki
A2	East-side	S3	04059132	Simon	53100	Ima
A2	East-side	S4	04454974	Cathrine	55910	Lpr

FIRST NORMAL FORM

- PhoneNumber should be put to its own table so area relation will now only need ***salesCode*** for the primary key

salesCode	salesName	product	quantity
S1	Helen	T1	120
S1	Helen	T9	75
S2	Jay	T2	99
S2	Jay	T4	111
S2	Jay	T5	234
S3	Simon	T12	123
S4	Cathrine	T9	2234
S4	Cathrine	T4	221

salesCode	phoneNmbr
S1	04055750
S1	05011437
S2	04055751
S3	04059132
S4	04454974

areaCode	areaName	salesCode	salesName	postalCode	City
A1	West-side	S1	Helen	00660	Hki
A1	West-side	S2	Jay	00630	Hki
A2	East-side	S3	Simon	53100	Ima
A2	East-side	S4	Cathrine	55910	Lpr

SECOND NORMAL FORM

- In 2nd normal form – Full functional dependency, {X} → Y and {X} cannot be divided without losing FD
  - Attributes should be functionally dependent on the key attribute(s) / prime attributes
  - If multiple attributes make up the primary key, all other attributes should be functionally dependent on all

salesCode	salesName	product	quantity
S1	Helen	T1	120
S1	Helen	T9	75
S2	Jay	T2	99
S2	Jay	T4	111
S2	Jay	T5	234
S3	Simon	T12	123
S4	Cathrine	T9	2234
S4	Cathrine	T4	221

areaCode	areaName	salesCode	salesName	postalCode	City
A1	West-side	S1	Helen	00660	Hki
A1	West-side	S2	Jay	00630	Hki
A2	East-side	S3	Simon	53100	Ima
A2	East-side	S4	Cathrine	55910	Lpr

salesCode	phoneNmbr
S1	04055750
S1	05011437
S2	04055751
S3	04059132
S4	04454974

SECOND NORMAL FORM

- {salesCode, product} is the primary key but **salesName** is only dependent on the **salesCode** → Remove the column from the relation

salesCode	product	quantity
S1	T1	120
S1	T9	75
S2	T2	99
S2	T4	111
S2	T5	234
S3	T12	123
S4	T9	2234
S4	T4	221

areaCode	areaName	salesCode	salesName	postalCode	City
A1	West-side	S1	Helen	00660	Hki
A1	West-side	S2	Jay	00630	Hki
A2	East-side	S3	Simon	53100	Ima
A2	East-side	S4	Cathrine	55910	Lpr

salesCode	phoneNmbr
S1	04055750
S1	05011437
S2	04055751
S3	04059132
S4	04454974

THIRD NORMAL FORM, NOW YOU ARE NORMALIZED

- Attributes can only be perfectly dependent on the key attributes
- There can be no transitive dependence between primary key and non-key attributes
  - E.g. Primary Key leads to Attribute 1, Attribute 1 leads to Attribute 2
    ←→ A2 is dependent on A1, A1 is dependent on PK
    ←→ A1 is transitively dependent on PK
  - Transitive dependency: PK → A1 & A1→ A2 = PK → A2
  - In other terms: No attribute should be identifiable based on a non-key attribute

salesCode	product	quantity
S1	T1	120
S1	T9	75
S2	T2	99
S2	T4	111
S2	T5	234
S3	T12	123
S4	T9	2234
S4	T4	221

areaCode	areaName	salesCode	salesName	postalCode	City
A1	West-side	S1	Helen	00660	Hki
A1	West-side	S2	Jay	00630	Hki
A2	East-side	S3	Simon	53100	Ima
A2	East-side	S4	Cathrine	55910	Lpr

salesCode	phoneNmbr
S1	04055750
S1	05011437
S2	04055751
S3	04059132
S4	04454974

THIRD NORMAL FORM

- In the example case,
  - City is possible to be identified based on the postal code
  - Area name can be identified based on area code
- To solve this, city and postal code need to be made into another relation as well as area code and area name

salesCode	product	quantity
S1	T1	120
S1	T9	75
S2	T2	99
S2	T4	111
S2	T5	234
S3	T12	123
S4	T9	2234
S4	T4	221

areaCode	areaName	salesCode	salesName	postalCode	City
A1	West-side	S1	Helen	00660	Hki
A1	West-side	S2	Jay	00630	Hki
A2	East-side	S3	Simon	53100	Ima
A2	East-side	S4	Cathrine	55910	Lpr

salesCode	phoneNmbr
S1	04055750
S1	05011437
S2	04055751
S3	04059132
S4	04454974

THIRD NORMAL FORM

areaCode	salesCode	salesName	postalCode
A1	S1	Helen	00660
A1	S2	Jay	00630
A2	S3	Simon	53100
A2	S4	Cathrine	55910

areaCode	areaName
A1	West-side
A2	East-side

salesCode	product	quantity
S1	T1	120
S1	T9	75
S2	T2	99
S2	T4	111
S2	T5	234
S3	T12	123
S4	T9	2234
S4	T4	221

postalCode	City
00660	Hki
00630	Hki
53100	Ima
55910	Lpr

salesCode	phoneNmbr
S1	04055750
S1	05011437
S2	04055751
S3	04059132
S4	04454974

THIRD NORMAL FORM EXTRAORDINAIRE

- Boyce/Codd normal form
- Extension of 3rd normal form
  - Attributes can only be perfectly dependent on the key attributes
- In BCNF, the attributes can only be dependent on the super keys
  - Superkey = set of attributes that uniquely identify an entity/tuple

R(A, B, C, D)
{A,B} → C
{A,B} → D
{B,C} → D

R(A, B, C,)
{A} → B
{A} → C
{C} → B

Buyer depends on the **product** and **salesCode**, but there is also a dependency from **buyer** to **product**. However, **buyer** is not a superkey.

https://www.studytonight.com/dbms/boyce-codd-normal-form.php

salesCode	product	quantity	buyer
S1	T1	120	B1
S1	T9	75	B4
S2	T2	99	B7
S2	T4	111	B2
S2	T5	234	B3
S3	T12	123	B5
S4	T9	2234	B4
S4	T4	221	B6

salesCode	buyer	quantity
S1	B1	120
S1	B4	75
S2	B6	99
S2	B2	111
S2	B3	234
S3	B5	123
S4	B4	2234
S4	B6	221

buyer	product
B1	T1
B2	T4
B3	T5
B4	T9
B5	T12
B6	T2
B7	T4

FOURTH NORMAL FORM

- There should be no multi-valued dependencies
  - Requires at least three attributes per relation

saleID	product	quantity
1	T1	120
2	T9	75
3	T2	99
4	T4	111
5	T5	234
6	T12	123
7	T9	2234
8	T4	221

saleID	quantity
1	120
2	75
3	99
4	111
5	234
6	123
7	2234
8	221

saleID	product
1	T1
2	T9
3	T2
4	T4
5	T5
6	T12
7	T9
8	T4

Dep	empl	Skills
1	Jay	SQL
1	Jay	Java
1	Helen	SQL
1	Helen	Java
2	Mark	HTML
2	Mark	React

Dep → empl
Dep → skills
But Dep is not a super key

Dep	Skills
1	SQL
1	Java
2	HTML
2	React

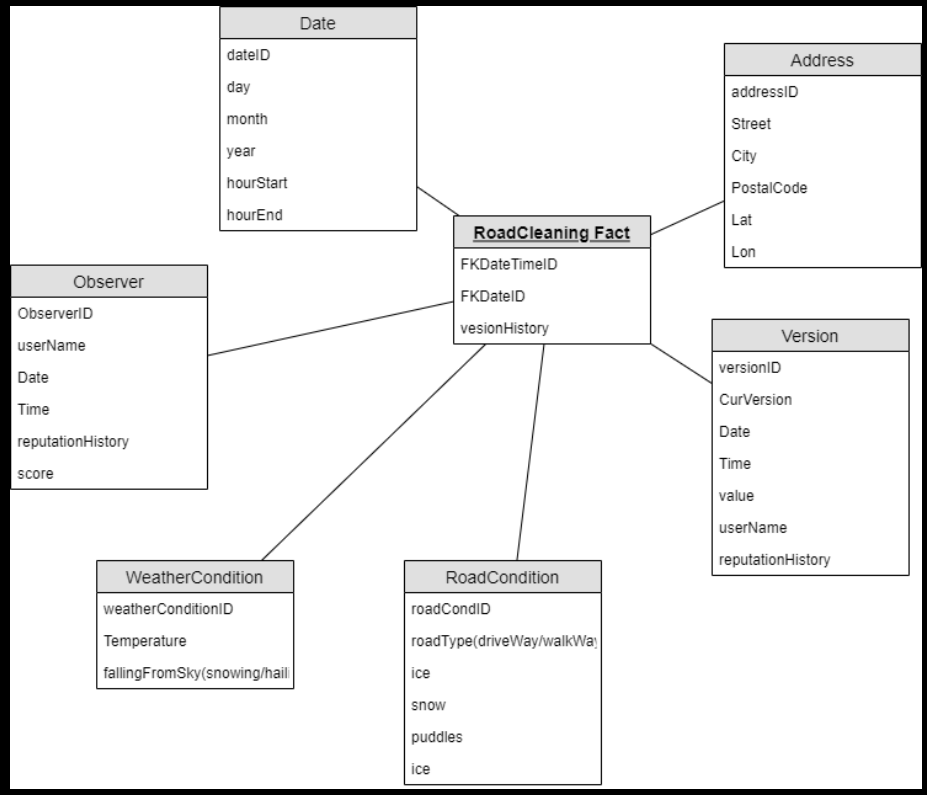
Dep	Empl
1	Jay
1	Helen
2	Mark

WHAT IS ENOUGH NORMALIZATION?

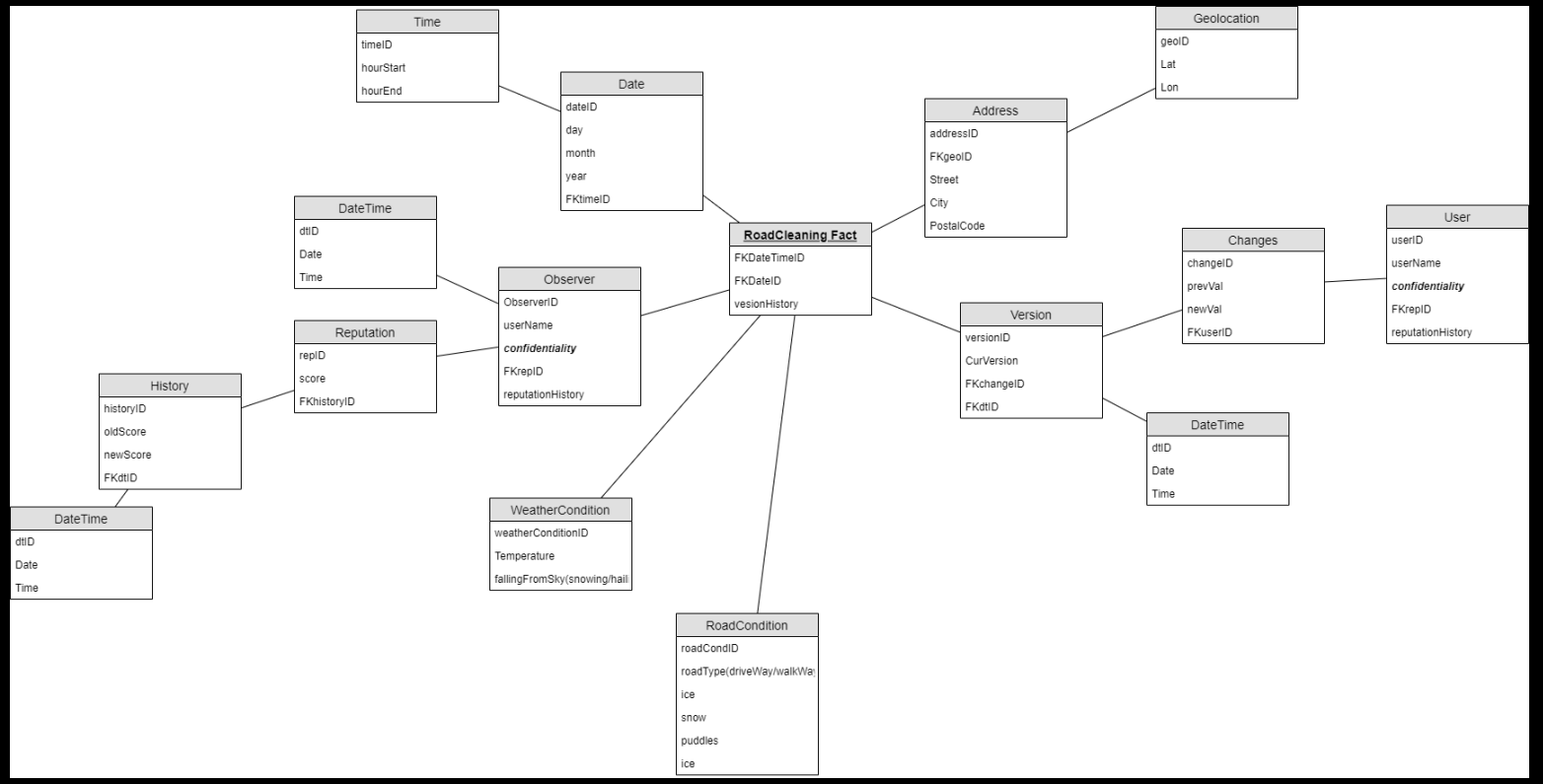
- In general, third normal form is considered to be enough
  - Everything beyond 3rd normal form should be carefully considered if it is necessary
  - Normalizing everything into relations with only two attributes (one key attribute) seems like an overkill and waste of resources
    - Normalizing some into this state may be beneficial

- Normalization is situational
  - Sometimes you may need to normalize all relations to 3rd form
  - Sometimes you need to leave some relations to 2nd form

STAR SCHEMA


Date
dateID
day
month
year
hourStart
hourEnd
Address
addressID
Street
City
PostalCode
Lat
Lon
RoadCleaning Fact
FKDateTimeID
FKDateID
vesionHistory
Observer
ObserverID
userName
Date
Time
reputationHistory
score
Version
versionID
CurVersion
Date
Time
value
userName
reputationHistory
WeatherCondition
weatherConditionID
Temperature
fallingFromSky(snowing/hail
RoadCondition
roadCondID
roadType(driveWay/walkWa
ice
snow
puddles
ice

SNOWFLAKE SCHEMA




LUT
University