3NF

Université Grenoble Alpes

30/03/2023

Bahareh Afshinpour

bahareh.afshinpour@univ-grenoble-alpes.fr

Université Grenoble Alpes

Main reference:
A First Course in Database Systems (and associated material) by
J. Ullman and J. Widom, Prentice-Hall

- Still we have some redundancy in our database if our tables are in the 2NF. Because of that redundancy, some anomalies may exist right now.
- The third normal form is an adequate normal form for your database.
- It means if your database is in the 3NF then you can say that it is a good database design.

What kind of anomaly we have in the 2NF

- The only candidate key is the PassportID (primary-key)

- Since here the candidate key has only one attribute (simple candidate key), then this relation is in the 2NF.

In the 2NF there should be no partial dependency.

What is partial functional dependency (PD) :
- A proper subset of candidate key → non-prim attributes(NPA)

If only one attribute we have for the candidate key (PassportID) obviously no proper subset is possible. So we will never have a partial dependency. (This relation is in 2NF)

PasspostID	Name	Family	country	City	Postal-code	Degree
58459435	David	Mixam	USA	x	9011	1
21476984	Mark	Smith	USA	y	9011	2
74632543	Franck	Thury	USA	z	9011	1
93268547	Sara	Victory	canada	v	91761	3

Redundancy

Update anomaly

-But This relation still has a redundancy
By using postal code we can determine country and city. (postal-code → country,city)

NPA
Is not candidate key or partial candidate key

NPA

Non prime attribute → Non prime attribute
Transitive dependency

- The relation is in the 3NF if and only if :

  1- It is 2NF

  2- It does not contain any transitive dependency (for non-prime attributes)

  Non prime attribute → Non prime attribute

R(A,B,C,D) FD={A→B, B→C, C→D}

Superkey
Set of attributes whose closure contains all attributes of a given relation

- Find out candidate key ABCD$^+$ ={A,B,C,D} We have all the attributes and it is a super key.

- We start to discard it since the candidate key is a minimal super key. A~~BCD~~$^+$ A$^+$ ={ABCD} SK=YES

CK= Is a superkey whose proper subset is **not** a superkey.

A has no PRPOPER SUBSET, so CK=YES

Prime attributes (PA) are those which are part of candidate keys.

PA=A

R(A,B,C,D) FD={A→B, B→C, C→D}   PA=A

Short trick: If prime attributes are present on the right-hand side of any functional dependency, then there would be more candidate keys. If not, there would be no more candidate keys.

In this example, we have no more prime attributes. The only candidate key is A

Non prime attribute → Non prime attribute

If this kind of dependency is present in the relation, this relation is not in the 3NF.

A→B , A is the prime attributes - no problem

B→C , B and C are the non- prime attributes - transitive dependency

C→D , C and D are the non- prime attributes - transitive dependency

So this relation is not in the 3NF.

3.2.9 Exercises for Section 3.2

Exercise 3.2.1: Consider a relation with schema $R(A, B, C, D)$ and FD's $AB \rightarrow C$, $C \rightarrow D$, and $D \rightarrow A$.

a) What are all the nontrivial FD's that follow from the given FD's? You should restrict yourself to FD's with single attributes on the right side.

b) What are all the keys of R?

c) What are all the superkeys for R that are not keys?

Example 3.8: Let us consider a relation with attributes A, B, C, D, E, and F. Suppose that this relation has the FD's $AB \rightarrow C$, $BC \rightarrow AD$, $D \rightarrow E$, and $CF \rightarrow B$. What is the closure of $\{A, B\}$, that is, $\{A, B\}^+$?

Decomposition

- Suppose a given relation is in the 2NF, we have to divide(decompose) the relation to have a higher normal form.
- For decomposition must follow some properties.
  - Dependency preserving decomposition
  - Lossless join decomposition

- R(A,B,C)
- FD=G

A	B	C
1	5	7
2	5	2
3	9	7
4	9	2

?

A	B

B	C

R1 R2 R3 R4 .... Rn

F1 U F2 U F3 U F4 U .... Fn F=G

Example of Dependency preserving

- Suppose we divide R into two sub relations R1 and R2. Is this dependency preserving decompositions?

R(A,B,C,D,E) FD= {A→B, B→C, C→D, D→A}

R1(A,B,C) R2(C,D,E)

First, we need to find the FD for R1 and R2.

- Find out the closure of all the attributes in the relation.

For example for R1(A,B,C) , one member, two member , … :

We do not have D in R1

A^+ ={A,B,C,D} → A→BC

B^+ ={B,C,D,A} → B→CA

C^+ ={C,D,A,B} → C→AB

AB^+={A,B,C,D} → ~~AB→C~~ But it is a duplicate FD

Since only using A→BC means A→C so AB→C . So we are not getting sth new

F1 ={A→BC, B→CA, C→AB}

Second, R2(C,D,E) , we find closure of each single attributes

C^+={C,D,A,B} C→D

D^+={D,A,B,C} D→C

E^+={E}

CD^+ ={CDAB}

DE^+ ={DE,AB,C} ~~DE→C~~ It would be duplicate because we have D→C

CE^+ ={CEDAB} ~~CE→D~~ It would be duplicate because we have C→D

F2 ={D→C, C→D}

F1 U F2=
{A→BC, B→CA, C→AB, D→C, C→D}

- If every FD of F is a member of G and every FD of G is a member of F then we can say they are equivalent.

F1 U F2 == G ????

G= {A→B, B→C, C→D, D→A}

F1 U F2=F={A→BC, B→CA, C→AB, D→C, C→D}

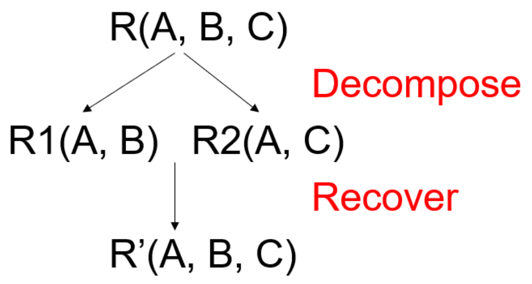
D^{+}={DC**A**B}

It means D→A is in the F

Yes. This example is a dependency-preserving decomposition

Lossless join decomposition

- Decomposition of a relation R is lossless if it is feasible to reconstruct the relation from decomposed sub relations by using joints.



Thus, R’ = R

A decomposition {R1, R2,…, Rn} of a relation R is called a lossless decomposition for R if the natural join of R1, R2,…, Rn produces exactly the relation R.

A	B	C
α	1	A
β	2	B

r

A	B
α	1
β	2

$\Pi_{A,B}(r)$

B	C
1	A
2	B

$\Pi_{B,C}(r)$

$$\Pi_{A}(r) \bowtie \Pi_{B}(r)$$

A	B	C
α	1	A
β	2	B

Project- part 2

printed version till 20/27 April

Your project counts for 33% of the final mark.

Propose a good database schema for a relational DBMS. This schema has to include the schema of the relations and their integrity constraints. Create the schema in the Oracle DBMS.

Insert pertinent data in your database.

Choose 5 queries to be evaluated on your data. Write the algebraic expressions and the SQL queries. Execute the SQL queries on your database.

Choose 3 more queries requiring sub-queries and/or the *group* by operator. Write and execute the SQL queries on your database.

The assignment must be returned in class.

Regarding part one, part 2 has to include the text of the description and the UML/ER diagram. Also, part two must contain the queries in natural language (English) and in algebra, all SQL programs (schema creation, data insertion, and queries).

Constraints

- A *constraint* is a relationship among data elements that the DBMS is required to enforce.
- Several kind of constraints
  - Example: primary key constraints.

Most used kinds of constraints

- Keys
  - Foreign-key, or referential integrity.
- Value-based constraints.
  - Constrain values of a particular attribute.

Actions Taken

- Suppose *R* = Sells, *S* = Beers.
- An insert or update to Sells that introduces a nonexistent beer must be rejected.
- A deletion or update to Beers that removes a beer value found in some tuples of Sells can be handled in three ways (next slide).

Actions Taken

1. Default : Reject the modification.
2. Cascade : Make the same changes in Sells.
   - Deleted beer: delete Sells tuple.
   - Updated beer: change value in Sells.
3. Set NULL : Change the beer to NULL.

Example: Cascade

- Delete the Bud tuple from Beers:
  - Then delete all tuples from Sells that have beer = ’Bud’.
- Update the Bud tuple by changing ’Bud’ to ’Budweiser’:
  - Then change all Sells tuples with beer = ’Bud’ so that beer = ’Budweiser’.

Example: Set NULL

- Delete the Bud tuple from Beers:
  - Change all tuples of Sells that have beer = ’Bud’ to have beer = NULL.
- Update the Bud tuple by changing ’Bud’ to ’Budweiser’:
  - Same change.