NORMALIZATION

Relational Database Design

_Part 2a

Outline

Features of Good Relational Design

Atomic Domains and First Normal Form

Functional Dependencies

Normal Forms

Functional Dependency Theory

Decomposition Using Functional Dependencies

Algorithms for Decomposition using Functional Dependencies

Decomposition Using Multivalued Dependencies

More Normal Form

To understand the Normal Forms and their importance in Relational Design .

We will cover 4 Normal Forms

- INF
- 2NF
- 3NF
- BCNF

Is this a good schema ?

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

- **Data Redundancy**
- **Need to decompose into smaller schemas**

Normalisation or Schema Refinment

Normalisation or Schema Refinement is a technique of organizing the data in a Database.

A systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like

Insertion Anomaly

Update Anomaly

Deletion Anomaly

Normalisation is used mainly for two purposes

- Eliminate redundant data
- Ensure Data dependencies logically stored

Desirable properties of Decomposition

Lossless join decomposition property

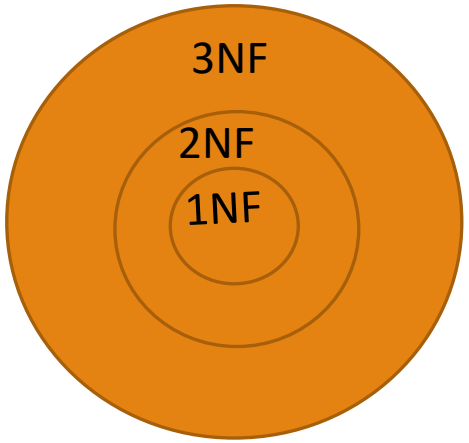
- It should be possible to reconstruct the original data

Dependency preserving property

- No functional dependency or constraints should be violated

From original schema step by step we can add constraints and split it into different schemas which are in

- INF
- 2NF
- 3Nf



Normalisation and Normal Forms

A normal form specifies a set of conditions that the relational schema must satisfy in terms of its constraints

- Offers varied levels of guarantee for the design

Normalisation rules are divided into various normal forms. Most common normal forms are.

- INF
- 2NF
- 3NF and
- BCNF

  Informally a relational database relation is said to be normalized if it meets third normal form . Most 3NF are free of insertion, updation and deletion anomalies.

Normalisation and Normal Forms

Additional Normal Forms:

- **BCNF**
- **Multivalued dependencies and 4NF**
- Joined dependencies and 5NF
- Sixth Normal Form
- DCNF(Domain Key Normal Form)

Goals of Normalization

Let R be a relation schema with a set F of functional dependencies.

Decide whether a relation schema R is in "good" form.

In the case that a relation schema R is not in "good" form, how to decompose it into a set of relation scheme $\{R_1, R_2, ..., R_n\}$ such that:

- Each relation schema is in good form
- The decomposition is a lossless decomposition
- Preferably, the decomposition should be dependency preserving.

- Based on

  Functional dependencies

  Multi-valued dependencies

First Normal Form

A relational schema R is in first normal form if the domains of all attributes of R are atomic

Domain is **atomic** if its elements are considered to be indivisible units

- Examples of non-atomic domains:
  - Set of names, composite attributes
  - Identification numbers like CS101 that can be broken up into parts

In other words a relation is in 1NF , if it does not have any composite or multi valued attributes .

Composite attribute ---->split to atomic unit (simple attributes)

Muli valued --→either put in different columns

or insert separate rows

leads to data redundancy

First Normal Form (Cont.)

Possible redundancy

Drawbacks

- Deletion Anomaly
- Insertion Anomaly
- Updation Anomaly

When LHS is not a superkey

- Let X→Y is a non trivial FD over R , with X is not a super key over R , then redundancy exists between X and Y attribute set.
- Hence in order to identify the redundancy, we need not to look at the actual data, it can be identified by given functional dependency.

When LHS is a superkey

- Let X→Y is a non trivial FD over R , with X is a super key over R , then redundancy does not exists between X and Y attribute set.

  X cannot duplicate as it is the candidate key.
- Y may or may not duplicate

Second Normal Form

For a Relation R to be in second normal form (2NF)

- R ishould be in 1NF
- R should not contain any partial dependency.

K is a super key for relation schema R if and only if $K \rightarrow R$

K is a candidate key (minimal super key) for R if and only if

- $K \rightarrow R$, *and*
- *for no* $\alpha \subset K$, $\alpha \rightarrow R$

Partial dependency

Let R be a relational Schema and(X,Y,A) be the attribute sets over R where X: Any candidate key , Y : Any proper subset of X and A : non key attribute

then Y->A is a partial dependency

Second Normal Form

Consider the ins_dep schema

Ins_dep (ID, name, salary, dept_name, building, budget).

(ID,dept_name) is the super key

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

Second Normal Form

Functional Dependencies

(ID,dept_name) -→ name (ID,dept_name) -→ salary

(ID,dept_name) -→ building (ID,dept_name) -→ budget

(ID,dept_name) -→ dept_name (trivial dependency)

Other FD's

ID → name *dept_name → budget*

ID → salary *dept_name → building*

Partial dependencies as ID

is a proper subset of super key ??

and name and salary are non key

attributes

Second Normal Form

Redundancy due to partial dependency. So split the Relation to

Instructor (ID,name,salary,dept_name) and

department(dept_name,building,budget)

Is this a lossless decomposition?

Will 2NF remove all data redundancy?