Normalization to 3NF

Design Theory for Relational Databases

- There are a lot of choices among attribute sets of a relational schema
- Some choices are better than others
- We will study
  - Desirable properties
  - How to obtain these desirable properties

2

Basic Concepts

- Study following relation

SP (Supp-Name, Supp-Addr, Item, Price)

John	10 Main	Apple	$2.00
John	10 Main	Orange	2.50
Jane	20 State	Grape	1.25
Jane	20 State	Apple	2.25
Frank	30 Elm	Mango	6.00

3

Basic Concepts

Do you see any problems with this relation?

- Study following relation

SP (Supp-Name, Supp-Addr, Item, Price)

John	10 Main	Apple	$2.00
John	10 Main	Orange	2.50
Jane	20 State	Grape	1.25
Jane	20 State	Apple	2.25
Frank	30 Elm	Mango	6.00

4

Basic Concepts

- What happens if John moves?

SP (Supp-Name, Supp-Addr, Item, Price)

John	10 Main	Apple	$2.00
John	10 Main	Orange	2.50
Jane	20 State	Grape	1.25
Jane	20 State	Apple	2.25
Frank	30 Elm	Mango	6.00

5

Basic Concepts

Every tuple of Johns would have to be changed.

- What happens if John moves?

SP (Supp-Name, Supp-Addr, Item, Price)

John	10 Main	Apple	$2.00
John	10 Main	Orange	2.50
Jane	20 State	Grape	1.25
Jane	20 State	Apple	2.25
Frank	30 Elm	Mango	6.00

6

Basic Concepts

- What happens to Frank's Address if we delete his tuple because he is temporarily not supplying us?

SP (Supp-Name, Supp-Addr, Item, Price)

John	10 Main	Apple	$2.00
John	10 Main	Orange	2.50
Jane	20 State	Grape	1.25
Jane	20 State	Apple	2.25
Frank	30 Elm	Mango	6.00

7

Basic Concepts

We would lose all the information about Frank and his address.

- What ha ns to Frank's Address if we delete hi ple because he is temporari ot supplying us?

SP (Supp-Na e, Supp-Addr, Item, Price)

John	10 Main	Apple	$2.00
John	10 Main	Orange	2.50
Jane	20 State	Grape	1.25
Jane	20 State	Apple	2.25
Frank	30 Elm	Mango	6.00

8

Basic Concepts

- What if we wish to keep track of a new supplier and the address but do not know what they will be supplying?

SP (Supp-Name, Supp-Addr, Item, Price)

John	10 Main	Apple	$2.00
John	10 Main	Orange	2.50
Jane	20 State	Grape	1.25
Jane	20 State	Apple	2.25
Frank	30 Elm	Mango	6.00

9

Basic Concepts

We cannot insert a new tuple with information about a supplier until we know what item they will be supplying!! Remember the Entity Integrity Constraint

SP (Supp-Name, Supp-Addr, Item, Price)

John	10 Main	Apple	$2.00
John	10 Main	Orange	2.50
Jane	20 State	Grape	1.25
Jane	20 State	Apple	2.25
Frank	30 Elm	Mango	6.00

10

Redundancy

- When attribute values are repeated unnecessarily
- Notice that address is repeated for each item that is supplied

11

Anomalies

- Update Anomaly
  - caused by redundant information
  - must find all copies of information in order to prevent inconsistencies when updating

12

Anomalies

- Deletion Anomaly
  - occurs when data is lost during a deletion that we do not wish to be lost
  - occurs when there are attributes within a tuple that are logically related to only part of the primary key

13

Anomalies

- Insertion Anomaly
  - occurs when we cannot insert some information into a tuple because of a violation of a relational constraint
  - occurs when a multiple attribute key cannot be fully completed as necessary for insertion

14

Fixing Anomalies

- Decompose the relation

SP (Supp-Name, Supp-Addr, Item, Price)

Supp(Supp-Name, Supp-Addr)

John	10 Main
Jane	20 State
Frank	30 Elm

SP-Item (Supp-Name, Item, Price)

John	Apple	$2.00
John	Orange	2.50
Jane	Grape	1.25
Jane	Apple	2.25
Frank	Mango	6.00

15

Fixing Anomalies: Decomposing Relations

Notice there is not any redundancy on supplier and supplier address. Update anomaly eliminated.

We can insert new supplier and its address and keep this information without knowing the item that will be supplied. Insertion anomaly eliminated.

We can delete the fact that Frank supplies mangos without losing Frank's address. Deletion anomaly eliminated

Supp(Supp-Name, Supp-Addr)

John	10 Main
Jane	20 State
Frank	30 Elm

SP-Item (Supp-Name, Item, Price)

John	Apple	$2.00
John	Orange	2.50
Jane	Grape	1.25
Jane	Apple	2.25
Frank	Mango	6.00

16

Decomposition of Relations

- Disadvantages
  - it is more expensive to solve queries
    - example: Get the address of suppliers supplying grapes.

SP (Supp-Name, Supp-Addr, Item, Price)
Only need ONE relation with the original

Supp(Supp-Name, Supp-Addr)
SP-Item (Supp-Name, Item, Price)
with decomposed schema - need to join two relations

17

Relational Design Methodology

- How do we tell whether one relation is better than another?
  - Check for anomalies
  - Normalization (also called functional dependency theory)

18

Functional Dependency

- Functional dependency
  - constraints in the data that depend upon NOT on the values within a given tuple BUT on whether or not two tuples agree on certain components.

19

Functional Dependency

- Functional dependency
  - Let R be a relation and let X and Y be subsets of the attributes (one or more) of R we say
    X → Y
    - X functionally determines Y
    - y is functionally dependent on X
  - if for all the tuples of R it is NOT possible that two tuples agree on X but disagree on Y

20

Functional Dependency

- Functional dependency
  - Given a unique value of X, we can ALWAYS determine a value of Y

21

Functional Dependency

- Person (SSN, Age, Gender)
  FD = { SSN → Age
  SSN → Gender
  Age ↛ Gender }
- Since two people of the same age can be of different genders

22

Functional Dependency

- FDs are assertions about the real world which cannot be proven
- FDs are established by the database designer by considering the meaning of the attributes
- FDs ***MUST*** hold for all possible data values
- FDs can be enforced during insertion if programmed and told to do so by the DBA

23

Functional Dependency

Emp-Proj (Emp-ID, Project, Supv, Dept, Case)

e1	p1	s1	d1	c1
e2	p2	s2	d2	c2
e1	p3	s1	d1	c3
e3	p3	s1	d1	c3

FD = { Emp-ID,Project →Project, Supv, Dept, Case
Emp-ID → Supv, Dept
Supv → Dept }

24

What do you see that is wrong with this relation?

Insert Anomaly:
We cannot enter a new employee say e4 who works for supervisor s4 until e4 works on some project.

Emp-Proj (Emp-ID, Project, Supv, Dept, Case)

e1	p1	s1	d1	c1
e2	p2	s2	d2	c2
e1	p3	s1	d1	c3
e3	p3		d1	c3

Update Anomaly:
If e1 has a new supervisor, then more than one tuple has to be changed.

Deletion Anomaly:
If e2 does not work on project p2 and we delete this tuple, we will loose ALL information about e2's supervisor, etc.

25

Before Normal Forms

- Unnormalized
  - repeating groups
  - **R (A, B, C, D (d1, d2, d3)*, E, F)**

FD = { A → B, C, E, F
A, d1, → d2, d3 }

R1 (A, B, C, E, F)
FD = { A → B, C, E, F }

D (A, d1, d2, d3)
FD2 = { A, d1 → d2, d3 }

26

Normal Forms

- 1NF (First Normal Form)
  - all values are atomic

27

First Normal Form

Emp-Proj (Emp-ID, Project, Supv, Dept, Case)

This relation is in 1NF. All values are atomic!

e1	p1	s1	d1	c1
e2	p2	s2	d2	c2
e1	p3	s1	d1	c3
e3	p3	s1	d1	c3

FD = { Emp-ID,Project → Emp-ID, Project, Supv, Dept,Case
Emp-ID → Supv, Dept
Supv → Dept }

28

Definitions

- Remember definitions of key
  - Super Key:
    - an attribute or set of attributes that uniquely identify a tuple (can be > 1 in a relation)
  - Candidate Key:
    - a minimum set of attributes that uniquely identify a tuple (can be > 1 in a relation)
    - a minimal super key
  - Primary Key:
    - one and only one per relation.
    - a chosen candidate key

29

Definitions

- Remember definitions of key
  - A candidate key of a relation functionally determines ALL attributes of the relation.

30

Definitions

- Fully Dependent
  - an attribute set Y is fully dependent on attribute set X if
    X → Y
    and Y cannot be determined by any subset of X
  - In Emp-Proj,
    - Case is fully dependent on Emp-ID, Project
    - Supv and Dept are NOT fully dependent on Emp-ID, Project

31

Definitions

- Prime Attribute
  - if an attribute appears in a key of a relation, then it is a prime attribute.
    - In Emp-Proj, Emp-ID is prime
- Non-Prime Attribute
  - an attribute not appearing in a key of a relation.
    - In Emp-Proj, Supv is non-prime

32

Normal Forms

- 1NF (First Normal Form)
  - all values are atomic
- 2NF (Second Normal Form)
  - a relation is in 2NF if it is in 1NF and each of its *non-prime* attributes are *fully dependent* upon its entire primary key.

33

Second Normal Form

NOT in 2NF because Supv and Dept are NOT fully dependent on Emp-ID, Project

Emp-Proj (Emp-ID, Project, Supv, Dept, Case)

	p1	s1	d1	c1
e	p2	s2	d2	c2
e1	p3	s1	d1	c3
e3	p3	s1	d1	c3

This FD violates 2NF!

FD = { Emp-ID, Project →Project, Supv, Dept, Case

Emp-ID → Supv, Dept

Supv → Dept }

34

Second Normal Form

Emp-Proj (Emp-ID, Project, Supv, Dept, Case)

FD = { Emp-ID, Project →Project, Supv, Dept, Cas

Emp-ID → Supv, Dept

Supv → Dept }

EP1 (Emp-ID, Supv, Dept)

FD1 = { Emp-ID → Supv, Dept
Supv → Dept }

EP2 (Emp-ID, Project, Case)

FD2 = { Emp-ID, Project → Case }

35

Second Normal Form

EP1 (Emp-ID, Supv, Dept)

FD1 = { Emp-ID → Supv, Dept
Supv → Dept }

EP2 (Emp-ID, Project, Case)

FD2 = { Emp-ID, Project → Case }

36

Problems with Second Normal Form

EP1 (Emp-ID, Supv, Dept)

e1	s1	d1
e2	s2	d2
e3	s1	d1

EP2 (Emp-ID, Project, Case)

e1	p1	c1
e2	p2	c2
e1	p3	c3
e3	p3	c3

37

Insertion Anomaly:
We cannot insert the information that supervisor s6 works for dept d6 unless at least one employee works for s6.

EP1 (Emp-ID, Supv, Dept)

e1	s1	d1
e2	s2	d2
e3	s1	d1

EP2 (Emp-ID, Project, Case)

e1	p1	c1
e2	p2	c2
e1	p3	c3
e3	p3	c3

Update Anomaly:
If supervisor s1 is moved to d4, we have to make changes in more than one tuple.

Deletion Anomaly:
If e2 is fired, information that s2 works for d2 is lost.

38

Definitions

- Transitively Dependent
  - A non-prime attribute is ***transitively dependent*** upon the primary key of a relation if there is also a non-prime (non key) attribute that functionally determines the attribute.
  - In EP1, Dept is transitively dependent upon Emp-ID since
    - Emp-ID → Supv
    - Supv → Dept

39

Normal Forms

- 1NF (First Normal Form)
  - all values are atomic
- 2NF (Second Normal Form)
  - a relation is in 2NF if it is in 1NF and each of its *non-prime* attributes are *fully dependent* upon its key.
- 3NF (Third Normal Form)
  - a relation is in 3NF if it is in 2NF and none of its non-prime attributes are *transitively dependent* on its key.

40

Third Normal Form

EP1 (Emp-ID, Supv, Dept)

FD1 = { Emp-ID → Supv, Dept

Supv → Dept }

EP1-1(Supv, Dept)

FD1-1 = { Supv → Dept }

EP1-2 (Emp-ID, Supv)

FD1-2 = { Emp-ID → Supv }

41

Third Normal Form Relations

Emp-Proj (Emp-ID, Project, Supv, Dept, Case)

FD = { Emp-ID,Project →Project, Supv, Dept, Case

Emp-ID → Supv, Dept

Supv → Dept }

Final Result of Normalization

EP2 (Emp-ID, Project, Case)

FD2 = { Emp-ID, Project → Case }

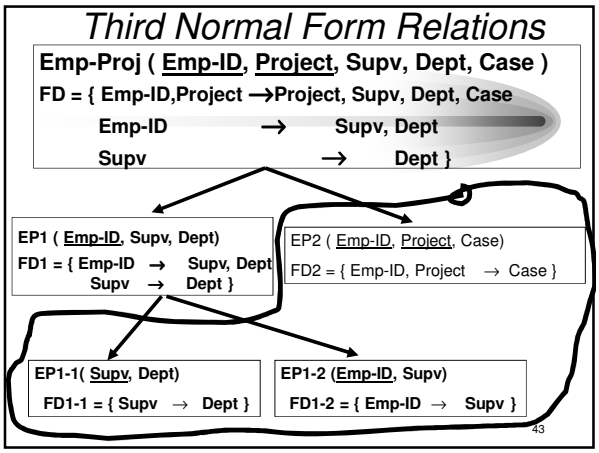
EP1-1(Supv, Dept)

FD1-1 = { Supv → Dept }

EP1-2 (Emp-ID, Supv)

FD1-2 = { Emp-ID → Supv }

42


Third Normal Form Relations
Emp-Proj (Emp-ID, Project, Supv, Dept, Case)
FD = { Emp-ID,Project →Project, Supv, Dept, Case
Emp-ID → Supv, Dept
Supv → Dept }
EP1 (Emp-ID, Supv, Dept)
FD1 = { Emp-ID → Supv, Dept
Supv → Dept }
EP2 (Emp-ID, Project, Case)
FD2 = { Emp-ID, Project → Case }
EP1-1(Supv, Dept)
FD1-1 = { Supv → Dept }
EP1-2 (Emp-ID, Supv)
FD1-2 = { Emp-ID → Supv }
43