CO226: Database Systems

Normalization

Sampath Deegalla
dsdeegalla@pdn.ac.lk

26th August 2014

Approaches for designing a relational database

- Designing a conceptual model and then map the conceptual design into a set of relations
- Designing the relations based on external knowledge derived from an existing implementation of files or forms or reports
- What are the criteria for good base relations?

Design Guidelines for Relational Databases

- Informal guidelines for good relational design
  - Semantics of the attributes
  - Reducing the redundant values in tuples
  - Reducing the null values in tuples
  - Disallowing the possibility of generating false tuples

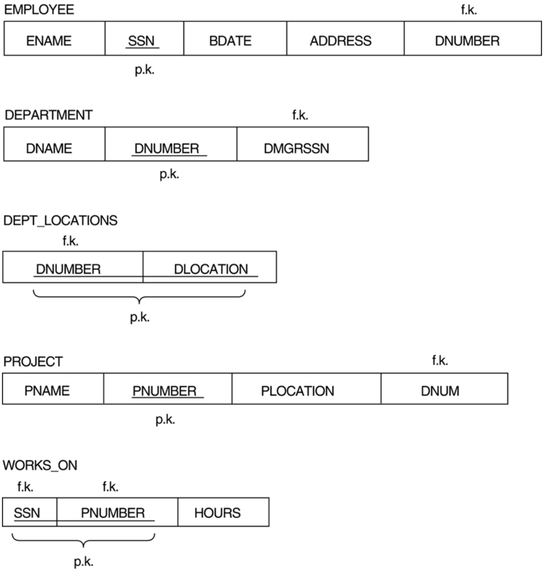
Design Guidelines for Relational Databases

- Formal concepts of functional dependencies, multivalued dependencies, join dependencies and normal forms
  - 1NF (First Normal Form)
  - 2NF (Second Normal Form)
  - 3NF (Third Normal Form)
  - BCNF (Boyce-Codd Normal Form)
  - 4NF (Forth Normal Form)
  - 5NF (Fifth Normal Form)

Semantics of the Relation Attributes

- GUIDELINE 1: Each tuple in a relation should represent one entity or relationship instance
  - Attributes of different entities (EMPLOYEEs, DEPARTMENTs, PROJECTs) should not be mixed in the same relation
  - Only foreign keys should be used to refer to other entities
  - Entity and relationship attributes should be kept apart as much as possible.
  - Design a schema that can be explained easily relation by relation.

A simplified COMPANY relational database schema



EMPLOYEE

ENAME	SSN	BDATE	ADDRESS	DNUMBER
	p.k.			f.k.

DEPARTMENT

DNAME	DNUMBER	DMGRSSN
	p.k.	f.k.

DEPT_LOCATIONS

DNUMBER	DLOCATION
f.k.	

p.k.

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM
	p.k.		f.k.

WORKS_ON

SSN	PNUMBER	HOURS
f.k.	f.k.	

p.k.

Redundant Information in Tuples and Update Anomalies

- Mixing attributes of multiple entities may cause problems
- Information is stored redundantly wasting storage
- Problems with update anomalies
    - Insertion anomalies
    - Deletion anomalies
    - Modification anomalies

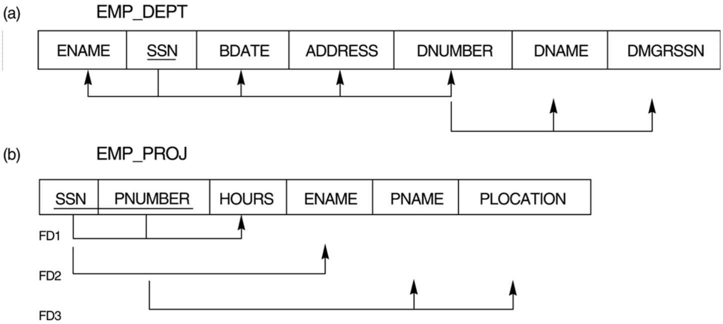
EXAMPLE OF AN UPDATE ANOMALY

- Consider the relation:
  `EMP_PROJ (Ssn, Pnumber, Hours, Ename, Pname, Plocation)`
- Update Anomaly: Changing the name of project number P1 from "Billing" to "Customer- Accounting" may cause this update to be made for all 100 employees working on project P1.

EXAMPLE OF AN UPDATE ANOMALY

- Insert Anomaly: Cannot insert a project unless an employee is assigned to .
  Inversely - Cannot insert an employee unless an he/she is assigned to a project.
- Delete Anomaly: When a project is deleted, it will result in deleting all the employees who work on that project.
  Alternately, if an employee is the sole employee on a project, deleting that employee would result in deleting the corresponding project.

Two relation schemas suffering from update anomalies



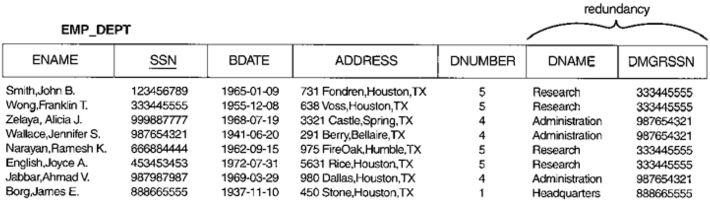
(a) EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN

(b) EMP_PROJ

SSN	PNUMBER	HOURS	ENAME	PNAME	PLOCATION

Example States for EMP_DEPT and EMP_PROJ



redundancy

EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
Smith,John B.	123456789	1965-01-09	731 Fondren,Houston,TX	5	Research	333445555
Wong,Franklin T.	333445555	1955-12-08	638 Voss,Houston,TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle,Spring,TX	4	Administration	987654321
Wallace,Jennifer S.	987654321	1941-06-20	291 Berry,Bellaire,TX	4	Administration	987654321
Narayan,Ramesh K.	666884444	1962-09-15	975 FireOak,Humble,TX	5	Research	333445555
English,Joyce A.	453453453	1972-07-31	5631 Rice,Houston,TX	5	Research	333445555
Jabbar,Ahmad V.	987987987	1969-03-29	980 Dallas,Houston,TX	4	Administration	987654321
Borg,James E.	888665555	1937-11-10	450 Stone,Houston,TX	1	Headquarters	888665555

Guideline to Redundant Information in Tuples and Update Anomalies

- GUIDELINE 2: Design a schema that does not suffer from the insertion, deletion and update anomalies. If there are any present, then note them so that applications can be made to take them into account

Null Values in Tuples

GUIDELINE 3: Relations should be designed such that their tuples will have as few NULL values as possible

- Attributes that are NULL frequently could be placed in separate relations (with the primary key)
- Reasons for nulls:
    - attribute not applicable or invalid
    - attribute value unknown (may exist)
    - value known to exist, but unavailable

Spurious Tuples

- Bad designs for a relational database may result in erroneous results for certain JOIN operations
- The "lossless join" property is used to guarantee meaningful results for join operations

GUIDELINE 4: The relations should be designed to satisfy the lossless join condition. No spurious tuples should be generated by doing a natural-join of any relations.

Functional Dependencies

- Functional dependencies (FDs) are used to specify formal measures of the “goodness” of relational designs
- FDs and keys are used to define normal forms for relations
- FDs are constraints that are derived from the meaning and interrelationships of the data attributes
- A set of attributes X functionally determines a set of attributes Y if the value of X determines a unique value for Y

Functional Dependencies

- $X \rightarrow Y$ holds if whenever two tuples have the same value for X, they must have the same value for Y
- For any two tuples t_1 and t_2 in any relation instance $r(R)$: If $t_1[X] = t_2[X]$, then $t_1[Y] = t_2[Y]$
- $X \rightarrow Y$ in R specifies a constraint on all relation instances $r(R)$
- Written as $X \rightarrow Y$; can be displayed graphically on a relation schema as in Figure **??**.
- FDs are derived from the real-world constraints on the attributes

Examples of FD constraints

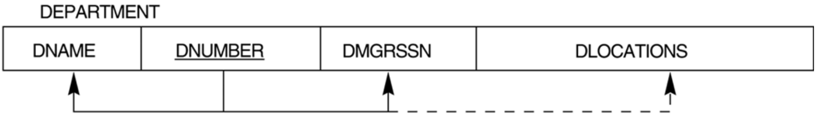
- social security number determines employee name
  `SSN` → `ENAME`
- project number determines project name and location
  `PNUMBER` → `{PNAME, PLOCATION}`
- employee ssn and project number determines the hours per week that the employee works on the project
  `{SSN, PNUMBER}` → `HOURS`

First Normal Form

- Disallows composite attributes, multivalued attributes, and nested relations; attributes whose values for an individual tuple are non-atomic
- Considered to be part of the definition of relation

Normalization into 1NF

(a)



(b)

DEPARTMENT

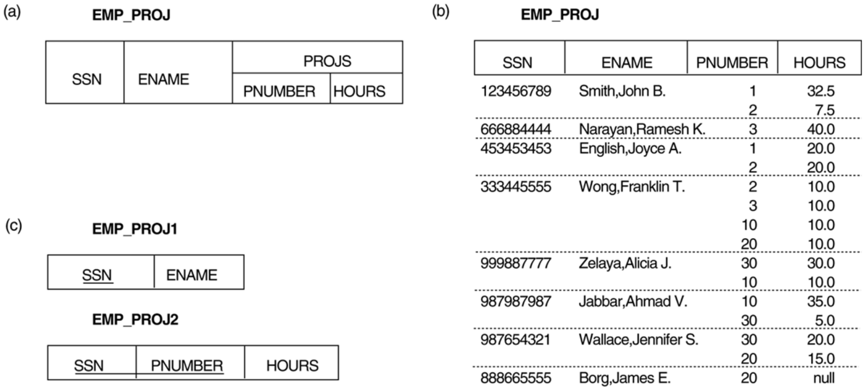
DNAME	DNUMBER	DMGRSSN	DLOCATIONS
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

(c)

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATION
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Normalization nested relations into 1NF



(a) **EMP_PROJ**

SSN	ENAME	PROJS	
		PNUMBER	HOURS

(b) **EMP_PROJ**

SSN	ENAME	PNUMBER	HOURS
123456789	Smith,John B.	1	32.5
		2	7.5
666884444	Narayan,Ramesh K.	3	40.0
453453453	English,Joyce A.	1	20.0
		2	20.0
333445555	Wong,Franklin T.	2	10.0
		3	10.0
		10	10.0
		20	10.0
999887777	Zelaya,Alicia J.	30	30.0
		10	10.0
987987987	Jabbar,Ahmad V.	10	35.0
		30	5.0
987654321	Wallace,Jennifer S.	30	20.0
		20	15.0
888665555	Borg,James E.	20	null

(c) **EMP_PROJ1**

SSN	ENAME

EMP_PROJ2

SSN	PNUMBER	HOURS

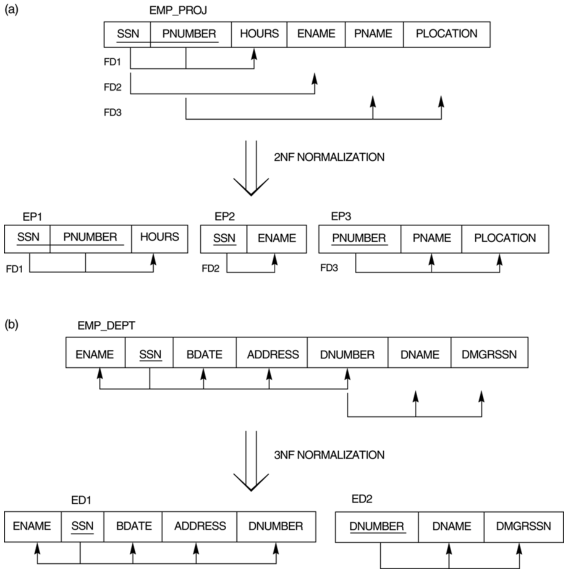
Second Normal Form

- Uses the concepts of FDs, primary key Definitions:
- Prime attribute -attribute that is member of the primary key K
- Full functional dependency - a FD $Y \rightarrow Z$ where removal of any attribute from Y means the FD does not hold any more Examples:
  - {SSN, PNUMBER} → HOURS is a full FD since neither SSN → HOURS nor PNUMBER → HOURS hold
  - {SSN, PNUMBER} → ENAME is not a full FD (it is called a partial dependency) since SSN → ENAME also holds

Second Normal Form

- A relation schema R is in second normal form (2NF) if every non-prime attribute A in R is fully functionally dependent on the primary key
- R can be decomposed into 2NF relations via the process of 2NF normalization

Normalizing into 2NF and 3NF



(a)

EMP_PROJ

SSN	PNUMBER	HOURS	ENAME	PNAME	PLOCATION

FD1

FD2

FD3

2NF NORMALIZATION

EP1

SSN	PNUMBER	HOURS

FD1

EP2

SSN	ENAME

FD2

EP3

PNUMBER	PNAME	PLOCATION

FD3

(b)

EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN

3NF NORMALIZATION

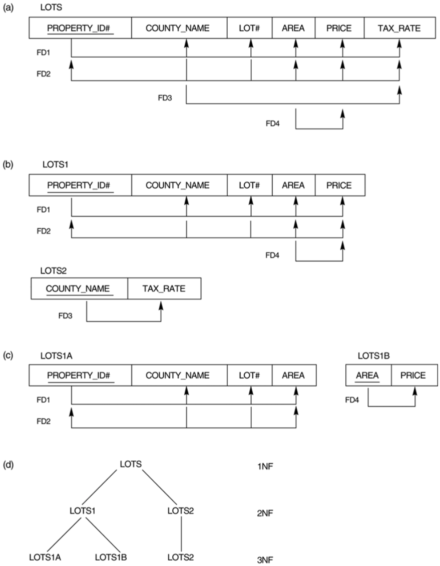
ED1

ENAME	SSN	BDATE	ADDRESS	DNUMBER

ED2

DNUMBER	DNAME	DMGRSSN

Normalization into 2NF and 3NF



(a) LOTS

PROPERTY_ID#	COUNTY_NAME	LOT#	AREA	PRICE	TAX_RATE

FD1

FD2

FD3

FD4

(b) LOTS1

PROPERTY_ID#	COUNTY_NAME	LOT#	AREA	PRICE

FD1

FD2

FD4

LOTS2

COUNTY_NAME	TAX_RATE

FD3

(c) LOTS1A

PROPERTY_ID#	COUNTY_NAME	LOT#	AREA

FD1

FD2

LOTS1B

AREA	PRICE

FD4

(d)

Third Normal Form

- Definition: Transitive functional dependency - a FD $X \to Y$ that can be derived from two FDs $X \to Z$ and $Z \to Y$
- Examples:
  - SSN → DMGRSSN is a transitive FD since SSN → DNUMBER and DNUMBER → DMGRSSN hold
  - SSN → ENAME is non-transitive since there is no set of attributes X where SSN → X and X → ENAME

Third Normal Form

- A relation schema R is in third normal form (3NF) if it is in 2NF and no non-prime attribute A in R is transitively dependent on the primary key
- R can be decomposed into 3NF relations via the process of 3NF normalization

NOTE:
In X → Y and Y → Z, with X as the primary key, we consider this a problem only if Y is not a candidate key. When Y is a candidate key, there is no problem with the transitive dependency.
E.g., Consider EMP (SSN, Emp#, Salary). Here, SSN → Emp# → Salary and Emp# is a candidate key.

General Normal Form Definitions (For Multiple Keys)

- The above definitions consider the primary key only
- The following more general definitions take into account relations with multiple candidate keys
- A relation schema R is in second normal form (2NF) if every non-prime attribute A in R is fully functionally dependent on every key of R

General Normal Form Definitions

Definition:

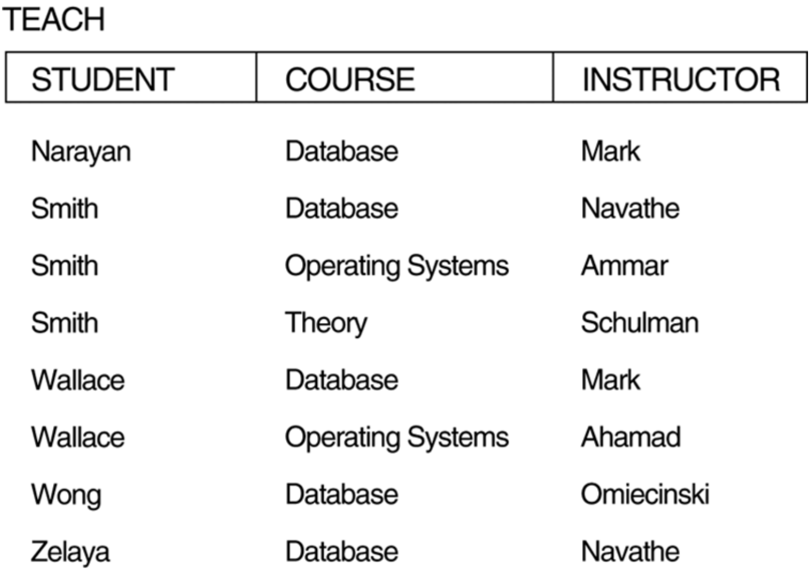
- Superkey of relation schema R - a set of attributes S of R that contains a key of R
- A relation schema R is in third normal form (3NF) if whenever a FD X → A holds in R, then either:
  1. X is a superkey of R, or
  2. A is a prime attribute of R

  NOTE: Boyce-Codd normal form disallows condition (2) above

BCNF (Boyce-Codd Normal Form)

- A relation schema R is in Boyce-Codd Normal Form (BCNF) if whenever an FD X → A holds in R, then X is a superkey of R
- Each normal form is strictly stronger than the previous one
    - Every 2NF relation is in 1NF
    - Every 3NF relation is in 2NF
    - Every BCNF relation is in 3NF
- There exist relations that are in 3NF but not in BCNF
- The goal is to have each relation in BCNF (or 3NF)

A relation TEACH that is in 3NF but not in BCNF



TEACH

STUDENT	COURSE	INSTRUCTOR
Narayan	Database	Mark
Smith	Database	Navathe
Smith	Operating Systems	Ammar
Smith	Theory	Schulman
Wallace	Database	Mark
Wallace	Operating Systems	Ahamad
Wong	Database	Omiecinski
Zelaya	Database	Navathe

Achieving the BCNF by Decomposition

- Two FDs exist in the relation TEACH:

  fd1 {student, course} → instructor

  fd2 instructor → course

- {student, course} is a candidate key for this relation and that the dependencies shown follow the pattern in Figure **??**. So this relation is in 3NF but not in BCNF
- A relation NOT in BCNF should be decomposed so as to meet this property, while possibly forgoing the preservation of all functional dependencies in the decomposed relations.

Achieving the BCNF by Decomposition

- Three possible decompositions for relation TEACH

$$\{\underline{\text{student}}, \underline{\text{instructor}}\} \text{ and } \{\underline{\text{student}}, \underline{\text{course}}\}$$
$$\{\text{course}, \underline{\text{instructor}}\} \text{ and } \{\underline{\text{course}}, \underline{\text{student}}\}$$
$$\{\underline{\text{instructor}}, \text{course}\} \text{ and } \{\underline{\text{instructor}}, \underline{\text{student}}\}$$

- All three decompositions will lose fd1. We have to settle for sacrificing the functional dependency preservation. But we cannot sacrifice the non-additivity property after decomposition.
- Out of the above three, only the 3rd decomposition will not generate spurious tuples after join.