A Deeper Look at Data Modeling

Shan-Hung Wu & DataLab
CS, NTHU

Outline

- More about ER & Relational Models
  - Weak Entities
  - Inheritance
- Avoiding redundancy & inconsistency
  - Functional Dependencies
  - Normal Forms

Outline

- More about ER & Relational Models
  - Weak Entities
  - Inheritance
- Avoiding redundancy & inconsistency
  - Functional Dependencies
  - Normal Forms

Modeling Users Addresses

- Street, city, etc.
- Each user may have multiple addresses
  - Home, office, etc.

users

id	name	karma
729	Bob	35
730	John	0

posts

id	text	ts	authorId
33981	‘Hello DB!’	1493897351	729
33982	‘Show me code’	1493904323	812

Modeling Users Addresses

- How to reflect:
  - Home and office addresses?
  - Address exists only when it owner (user) exists?

users

id	name	karma
729	Bob	35
730	John	0

addresses

id	userId	street	city
4356	729	‘X Rd.’	‘New York’
4357	729	‘Y Rd.’	‘LA’

posts

id	text	ts	authorId
33981	‘Hello DB!’	1493897351	729
33982	‘Show me code’	1493904323	812

Modeling Users Addresses

- How to reflect:
  - ***Home and office addresses?***
  - Address exists only when it owner (user) exists?

users

id	name	karma
729	Bob	35
730	John	0

addresses

userId	type	street	city
729	‘home’	‘X Rd.’	‘New York’
729	‘office’	‘Y Rd.’	‘LA’

```
CREATE TABLE addresses (
  userId        serial NOT NULL,
  type          text NOT NULL,
  ...
  PRIMARY KEY   (userId, type)
);
```

Modeling Users Addresses

- How to reflect:
  - Home and office addresses?
  - ***Address exists only when it owner (user) exists?***

```
CREATE TABLE addresses (
   userId            serial NOT NULL,
   type              text NOT NULL,
   ...
   PRIMARY KEY        (userId, type),
   FOREIGN KEY       userId
                     REFERENCES users ON DELETE CASCADE
);
```

Outline

- More about ER & Relational Models
  - Weak Entities
  - Inheritance
- Avoiding redundancy & inconsistency
  - Functional Dependencies
  - Normal Forms

Modeling Inheritance

- Suppose you have employees in your model
- How to model special types of employees?
  - Contracted: contractId
  - Hourly: wage, workHours

Modeling Inheritance (1/2)

employees

id	name	department	type	wage	workHours	contractId
729	Bob	‘R&D’	Hourly	$10	4	NULL
730	John	‘Sales’	Hourly	$20	16	NULL
834	Steven	‘R&D’	Contract	NULL	NULL	3004
878	Chris	‘Sales’	Contract	NULL	NULL	2045

- Union columns
- Cons:
  - Null values
  - Schema changes when defining new emp. types

Modeling Inheritance (2/2)

employees

id	name	department
729	Bob	‘R&D’
730	John	‘Sales’

contractEmployees

eId	contractId
834	$10
878	$20

hourlyEmployees

eId	wage	workHours
729	$10	4
730	$20	16

- No nulls; less schema changes
- Cons:
  - Join queries
  - If a superclass tuple is deleted, needs cascade deleting subclass tuple

Outline

- More about ER & Relational Models
  - Weak Entities
  - Inheritance
- Avoiding redundancy & inconsistency
  - Functional Dependencies
  - Normal Forms

How Good Are Your Data?

- Let’s say, if you want to track the topics of a blog page
- Is this a good table?

blog_pages

blogId	url	created	authorId	topic	topicAdmin
33981	ms.com/…	2012/10/31	729	programming	5638
33981	ms.com/…	2012/10/31	729	db	5649
33982	apache.org/…	2012/11/15	4412	programming	5638
33982	apache.org/…	2012/11/15	4412	os	7423

Insertion Anomaly

blog_pages

blogId	url	created	authorId	topic	topicAdmin
33981	ms.com/…	2012/10/31	729	programming	5638
33981	ms.com/…	2012/10/31	729	db	5649
33982	apache.org/…	2012/11/15	4412	programming	5638
33982	apache.org/…	2012/11/15	4412	os	7423

33983	apache.org/…	2013/02/15	7412

?

- A blog cannot be inserted without knowing all fields of topics (except setting them to null)

Update Anomaly

blog_pages

blogId	url	created	authorId	topic	topicAdmin
33981	ms.com/…	2012/10/31	729	***win prog.***	5638
33981	ms.com/…	2012/10/31	729	db	5649
33982	apache.org/…	2012/11/15	4412	programming **X**	5638
33982	apache.org/…	2012/11/15	4412	os	7423

- If you forget to update all duplicated cells, you get inconsistent data

Deletion Anomaly

blog_pages

blogId	url	created	authorId	topic	topicAdmin
33981	ms.com/…	2012/10/31	729	***programming***	***5638*** X
33981	ms.com/…	2012/10/31	729	db	5649
33982	apache.org/…	2012/11/15	4412	programming	5638
33982	apache.org/…	2012/11/15	4412	os	7423

- Deleting topics force you to delete the blog fields too

Outline

- More about ER & Relational Models
  - Weak Entities
  - Inheritance
- Avoid redundancy & inconsistency
  - Functional Dependencies
  - Normal Forms

Functional Dependency (FD)

- FD: X □ Y
  - If two tuples agree in X, then they agree in Y
- What are the FDs for blog_pages?
  - blogId □ ... (key-based)
  - ***topic □ topicAdmin (non key-based)***

blog_pages

<u>blogId</u>	url	created	authorId	topic	topicAdmin
33981	ms.com/a…	2012/10/31	729	programming	5638
33982	ms.com/b…	2012/11/31	732	db	5649
33983	apache.org/…	2012/12/15	1312	programming	5638
33984	wiki.org/…	2013/1/15	4345	os	7423

Non Key-based FDs

- The root cause of anomalies
- Data redundancy
- Inconsistency

blog_pages

blogId	url	created	authorId	topic	topicAdmin
33981	ms.com/a…	2012/10/31	729	***win prog.***	5638
33982	ms.com/b…	2012/11/31	732	os	5649
33983	apache.org/…	2012/12/15	1312	programming	5638
33984	wiki.org/…	2013/1/15	4345	os X	7423

Outline

- More about ER & Relational Models
  - Weak Entities
  - Inheritance
- Avoid redundancy & inconsistency
  - Functional Dependencies
  - Normal Forms

Keys

- ***Super key***: an attribute or set of attributes that uniquely identifies a tuple within a relation
- ***Candidate key***: a super key such that no proper subset is a super key within the relation
  - An attribute that does not occur in any candidate key is called a ***non-prime attribute***
- ***Primary key***: the candidate key that is selected to identify tuples uniquely within the relation
  - Candidate keys which are not selected as PK are called alternate keys

Example

- Candidate keys

blog_pages

blogId	url	created	authorId	topic	topicAdmin
33981	ms.com/a…	2012/10/31	729	programming	5638
33982	ms.com/b…	2012/11/31	732	db	5649
33983	apache.org/…	2012/12/15	1312	programming	5638
33984	wiki.org/…	2013/1/15	4345	os	7423

Normal Forms

- 1st normal form:
  - Single-valued columns
- 2nd normal form:
  - All fields depends on the primary key
- BCNF normal form:
  - For every FD X □ Y, X is a super key
- 3rd normal form:
  - For every FD X □ Y, X is a super key ***or Y is a prime attribute***
  - Weaker than BCNF

3rd Normal Form?

blog_pages

blogId	url	created	authorId	topic	topicAdmin
33981	ms.com/a…	2012/10/31	729	programming	5638
33982	ms.com/b…	2012/11/31	732	db	5649
33983	apache.org/…	2012/12/15	1312	programming	5638
33984	wiki.org/…	2013/1/15	4345	os	7423

- FD: topic □ topicAdmin
  - Topic is not a superkey
  - TopicAdmin is not a prime attribute
- No!

Solution

blog_pages

blogId	url	created	authorId	topicId
33981	ms.com/a…	2012/10/31	729	123
33982	ms.com/b…	2012/11/31	732	456
33983	apache.org/…	2012/12/15	1312	123
33984	wiki.org/…	2013/1/15	4345	456

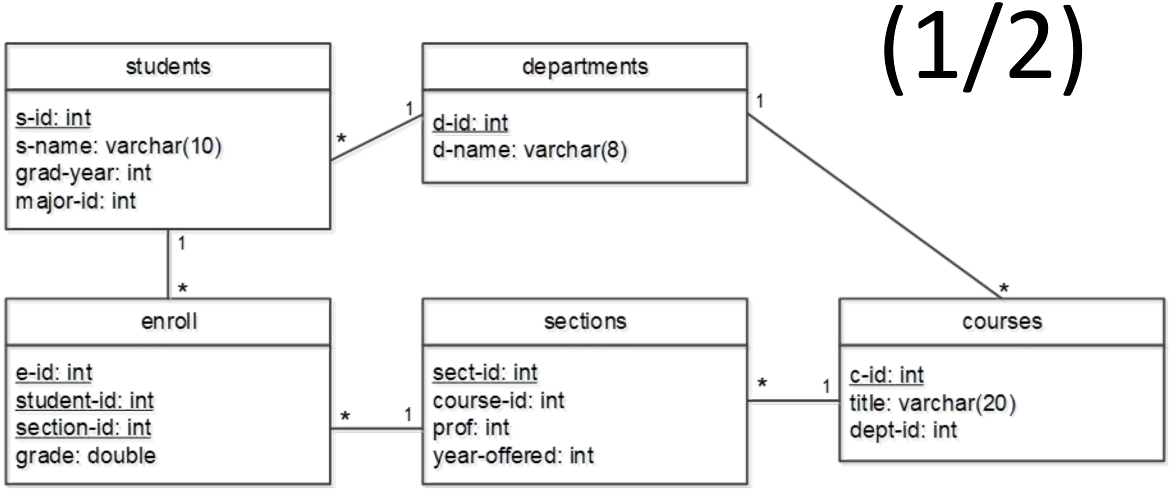
topics

topicId	name	admin
123	programming	5638
234	os	7423
456	db	5649
789	alg	7324

- Move non key-based FDs to new tables
- Avoids redundancy & inconsistency

BCNF Normal Form (1/2)

- Recall student DB:



students
s-id: int
s-name: varchar(10)
grad-year: int
major-id: int

departments
d-id: int
d-name: varchar(8)

enroll
e-id: int
student-id: int
section-id: int
grade: double

sections
sect-id: int
course-id: int
prof: int
year-offered: int

courses
c-id: int
title: varchar(20)
dept-id: int

- Let’s modify “sections” relation like this:

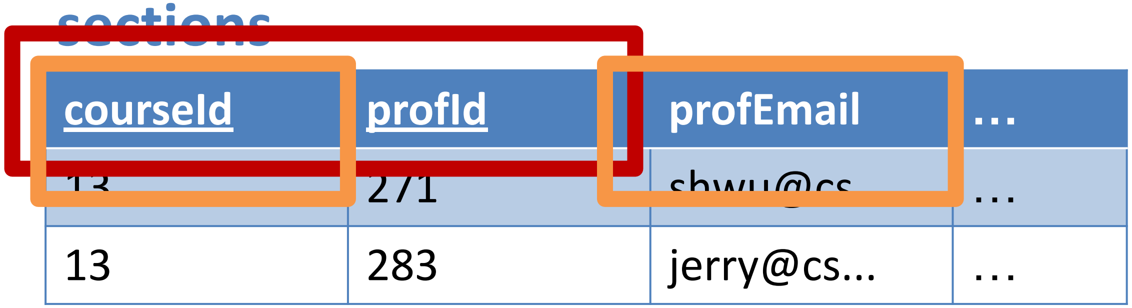
sections

courseId	profId	profEmail	…
13	271	shwu@cs…	…
13	283	jerry@cs…	…

  – Suppose each course needs to be taught by different professors in different years

BCNF Normal Form (2/2)

- Candidate keys:



sections

courseId	profId	profEmail	…
13	271	shwu@cs…	…
13	283	jerry@cs...	…

- “sections” is in 3rd normal form
  - FDs:
    - profId □ profEmail, and profEmail is a prime attribute
    - profEmail □ profId, and profId is a prime attribute
- But ***not*** in BCNF normal form!
  - profId/proEmail is not a super key

Solution

sections

courseId	profId	…
13	271	…
13	283	…

professors

profId	profEmail	…
271	shwu@cs...	…
283	jerry@cs...	…

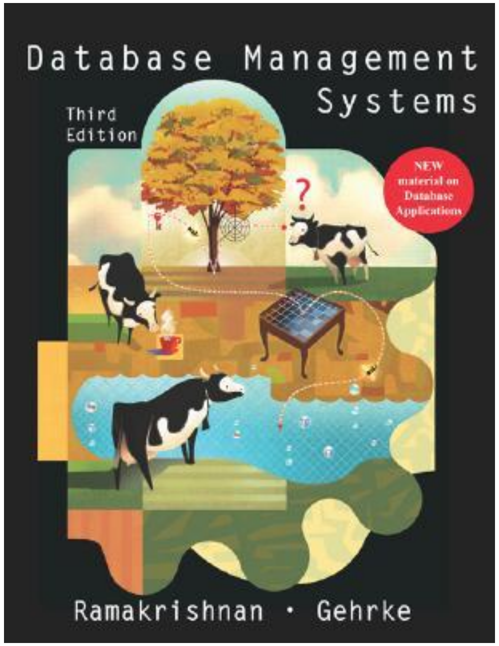
- BCNF normal form makes the 1-1 mapping between profId and profEmail explicit

Normalized ≠ Well-Designed

- Norm forms help reducing redundancy & avoiding inconsistency
- Costs
  - Slower query speed due to Joins
  - Hard-to-partition data on multiple machines
- In practice, it's common to to deliberately ***denormalize*** a schema
  - Will be covered in NoSQL lecture

Assigned Reading

- Chaps 2 and 3 on ER & relational models
- Chap 19 on FDs and normal forms


Database Management Systems
Third Edition
NEW material on Database Applications
Ramakrishnan · Gehrke