RELATIONAL DESIGN THEORY

6.830 / 6.814 LECTURE 4

TIM KRASKA

RECAP

Physical Independence

Logical Independence

Simplified Zoo Relations

animals

name	age	species	cageno	keptby	feedtime
mike	13	giraffe	1	1	10:00am
sam	3	salam	2	1	11:00am
sally	1	student	1	2	1:00pm

keepers

keeper	name
1	jenny
2	joe

Primary Key
Foreign Key

cages

cageno	bldg
1	2
2	3

KEYS AND RELATIONS

Kinds of keys

- Superkeys:
  set of attributes of table for which every row has distinct set of values
- Candidate keys:
  "minimal" superkeys
- Primary keys:
  DBA-chosen candidate key (marked in schema by underlining)

ISBN	Title	Author	Edition	Publisher	Price
0439708184	Harry Potter	J.K. Rowling	1	Scholastic	$6.70
0545663261	Mockingjay	Suzanne Collins	1	Scholastic	$7.39

Relational Design Theory

- Assess the quality of a schema
  - redundancy
  - integrity constraints
  - **Quality seal: normal forms (1-3, BCNF/3.5)**

- Improve the quality of a schema
  - **synthesis algorithm**
  - **decomposition algorithm**

- Construct a (high-quality) schema
  - start with universal relation
  - apply synthesis or decomposition algorithms

Bad Schemas

ProfLecture						
PersNr	Name	Level	Room	NB	Title	CP
2125	Tim	AP	G914	814	DB Systems	4
2125	Tim	AP	G914	049	Algorithms	2
2125	Tim	AP	G914	052	Logik	4
...	...	...	...	...	...	...
2132	Raul	AP	10	5259	German	2
2137	Mike	FP	100	4630	ML	4

- Update-Anomaly
  - What happens when Tim moves to a different room?
- Insert-Anomaly
  - What happens if Raul is elected as a new professor?
- Delete-Anomaly
  - What happens if Tim does not teach this semester?

Functional Dependencies

- Schema: $\mathcal{R} = \{A{:}D_A,\ B{:}D_B,\ C{:}D_C,\ D{:}D_D\}$
- Instance: R

- Let $\alpha \subseteq \mathcal{R},\ \ \beta \subseteq \mathcal{R}$
- $\alpha \rightarrow \beta$ iff $\forall r, s \in R$: $r.\alpha = s.\alpha \Rightarrow r.\beta = s.\beta$
- (There is a function f: $D_\alpha \rightarrow D_\beta$)

R			
A	*B*	*C*	*D*
a4	b2	c4	d3
a1	b1	c1	d1
a1	b1	c1	d2
a2	b2	c3	d2
a3	b2	c4	d3

$\{A\} \rightarrow \{B\}$

$\{C, D\} \rightarrow \{B\}$

Not: $\{B\} \rightarrow \{C\}$

Convention:

$CD \rightarrow B$

Example

Family Tree				
Child	Father	Mother	Grandma	Grandpa
Sofie	Alfons	Susan	Zoe	Kevin
Sofie	Alfons	Susan	Isabella	Mike
Mark	Alfons	Susan	Zoe	Kevin
Mark	Alfons	Susan	Isabella	Mike
...	...	...	Zoe	Martha
...	...	...	...	...

Example

Family Tree				
Child	Father	Mother	Grandma	Grandpa
Sofie	Alfons	Susan	Zoe	Kevin
Sofie	Alfons	Susan	Isabella	Mike
Mark	Alfons	Susan	Zoe	Kevin
Mark	Alfons	Susan	Isabella	Mike
...	...	...	Zoe	Martha
...	...	...	...	...

- Child→ Father,Mother
- Child, Grandpa → Grandma
- Child, Grandma → Grandpa

Analogy to functions

- f1 : Child → Father
  - E.g., f1(Mark) = Alfons
- f2: Child → Mother
  - E.g., f2(Mark) = Susan
- f3: Child x Grandpa → Grandma
- FD: Child → Father, Mother
  - represents two functions (f1, f2)
  - Comma on right side indicates multiple functions
- FD: Child, Grandpa → Grandma
  - Comma on the left side indicates Cartesian product

Decomposition of Relations

- Bad relations combine several concepts
  - decompose them so that each concept in one relation
  - $\mathcal{R} \rightarrow \mathcal{R}_1, .., \mathcal{R}_n$

1. Lossless Decomposition

   $\mathcal{R} = \mathcal{R}_{1.} \bowtie \mathcal{R}_{2.} \bowtie \ \ldots \bowtie \mathcal{R}_n$

2. Preservation of Dependencies

   $FD(\mathcal{R}) = (FD(\mathcal{R}_1) \cup \ldots \cup FD(\mathcal{R}_n))$

Example

Drinker		
Pub	*Guest*	*Beer*
Kowalski	Kemper	Pils
Kowalski	Eickler	Hefeweizen
Innsteg	Kemper	Hefeweizen

Lossy Decomposition

Drinker		
Pub	*Guest*	*Beer*
Kowalski	Kemper	Pils
Kowalski	Eickler	Hefeweizen
Innsteg	Kemper	Hefeweizen

$\Pi_{\text{Pub, Guest}}$

$\Pi_{\text{Guest, Beer}}$

Visitor	
Pub	*Guest*
Kowalski	Kemper
Kowalski	Eickler
Innsteg	Kemper

Drinks	
Guest	*Beer*
Kemper	Pils
Eickler	Hefeweizen
Kemper	Hefeweizen

Drinker		
Kneipe	*Gast*	*Bier*
Kowalski	Kemper	Pils
Kowalski	Eickler	Hefeweizen
Innsteg	Kemper	Hefeweizen

$\Pi_{\ldots}$

Visitor	
Pub	*Guest*
Kowalski	Kemper
Kowalski	Eickler
Innsteg	Kemper

Drinks	
Guest	*Beer*
Kemper	Pils
Eickler	Hefeweizen
Kemper	Hefeweizen

$\bowtie$

$\neq$

Visitor⋈Drinks		
Pub	*Guest*	*Beer*
Kowalski	Kemper	Pils
Kowalski	Kemper	Hefeweizen
Kowalski	Eickler	Hefeweizen
Innsteg	Kemper	Pils
Innsteg	Kemper	Hefeweizen

Preservation of Dependencies

- Let $\mathcal{R}$ be decomposed into $\mathcal{R}1$, ..., $\mathcal{R}n$
- $F_{\mathcal{R}} = (F_{\mathcal{R}1} \cup \ldots \cup F_{\mathcal{R}n})$

- ZipCodes: {[Street, City, State, Zip]}

- Functional dependencies in ZipCodes
  - {Zip} → {City, State}
  - {Street, City, State} → {Zip}

- What about this decomposition?
  - Streets: {[Zip, Street]}
  - Cities: {[Zip, City, State]}

- Clicker:
  Is it lossless? Does it preserve functional dependencies?

  Answer A: Yes, Yes Answer C: No, Yes

  Answer B: Yes, No Answer D: No, No

http://clicker.csail.mit.edu/6.814/

Decomposition of ZipCodes

ZipCodes			
City	***State***	***Street***	*Zip*
Cambridge	MA	Vassar St	02139
Cambridge	MA	Main St	02142
Cambridge	TX	Vassar St	75076

$\Pi_{\text{Zip,Street}}$

$\Pi_{\text{City,State,Zip}}$

Streets	
Zip	***Street***
75076	Vassar St
02139	Vassar St
02142	Main St

Cities		
City	*State*	***Zip***
Cambridge	MA	02139
Cambridge	MA	02142
Cambridge	TX	75076

{Street, City, State} → {Zip} not checkable in decomp. schema

It is possible to insert inconsistent tuples

Violation of City,State,Street→Zip

ZipCodes			
City	**State**	**Street**	*Zip*
Cambridge	MA	Vassar St	02139
Cambridge	MA	Main St	02142
Cambridge	TX	Vassar St	75076

$\Pi_{\text{Zip,Street}}$

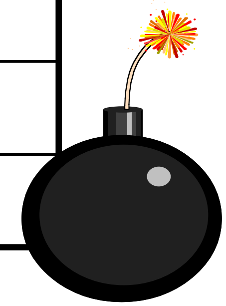
$\Pi_{\text{City,State,Zip}}$

Streets	
Zip	**Street**
75076	Vassar St
02139	Vassar St
02142	Main St
75078	Vassar St

Cities		
City	*State*	**Zip**
Cambridge	MA	02139
Cambridge	MA	02142
Cambridge	TX	75076
Cambridge	TX	75078

Violation of City,State,Street→Zip

ZipCodes			
City	***State***	***Street***	*Zip*
Cambridge	MA	Vassar St	02139
Cambridge	MA	Main St	02142
Cambridge	TX	Vassar St	75076
Cambridge	TX	Vassar St	75078



$\bowtie$

Streets	
Zip	***Street***
75076	Vassar St
02139	Vassar St
02142	Main St
75078	Vassar St

Cities		
City	*State*	***Zip***
Cambridge	MA	02139
Cambridge	MA	02142
Cambridge	TX	75076
Cambridge	TX	75078

First Normal Form

- Only atomic domains (as in SQL 92)

Parents		
Father	*Mother*	*Children*
Johann	Martha	{Else, Lucie}
Johann	Maria	{Theo, Josef}
Heinz	Martha	{Cleo}

vs.

Parents		
Father	*Mother*	*Child*
Johann	Martha	Else
Johann	Martha	Lucie
Johann	Maria	Theo
Johann	Maria	Josef
Heinz	Martha	Cleo

Second Normal Form

- No **non-prime attribute** is dependent on any subset of any candidate key of the relation

StundentAttends			
Student-ID	**Course-Nb**	Name	Semester
26120	5001	Fichte	10
27550	5001	Schopenhauer	6
27550	4052	Schopenhauer	6
28106	5041	Carnap	3
28106	5052	Carnap	3
28106	5216	Carnap	3
28106	5259	Carnap	3
...	...	...	...

Second Normal Form

- No **non-prime attribute** is dependent on any subset of any candidate key of the relation

StundentAttends			
Student-ID	**Course-Nb**	Name	Semester
26120	5001	Fichte	10
27550	5001	Schopenhauer	6
27550	4052	Schopenhauer	6
28106	5041	Carnap	3
28106	5052	Carnap	3
28106	5216	Carnap	3
28106	5259	Carnap	3
...	...	...	...

- StudentAttends is not in 2NF!!!
  - {Student-ID} → {Name, Semester}

Third Normal Form

- A table is in 3NF if and only if, for each of its functional dependencies $X \rightarrow A$, at least one of the following conditions holds:
  - $X \rightarrow A$ is trivial: X contains A
  - X is a superkey
  - Each attribute in $A - X$ (i.e, the set difference of between A and X) is contained in some candidate key

Third Normal Form

- A table is in 3NF if and only if, for each of its functional dependencies $X \to A$, at least one of the following conditions holds:
  - $X \to A$ is trivial: X contains A
  - X is a superkey
  - Each attribute in $A - X$ (i.e, the set difference of between A and X) is contained in some candidate key

Alternative:

- The relation R (table) is in second normal form (2NF)
- Every non-prime attribute of R is **non-transitively** dependent on every key of R.

Is The Following Table in 3NF?

Drinker		
Pub	*Guest*	*Beer*
Kowalski	Kemper	Hefeweizen
Kowalski	Eickler	Pils
Innsteg	Kemper	Hefeweizen

{Pub}→{Guest}
{Guest}→{Beer}

IS The Following Table in 3NF?

Drinker		
Pub	*Guest*	*Beer*
Kowalski	Kemper	Hefeweizen
Kowalski	Eickler	Pils
Innsteg	Kemper	Hefeweizen

{Pub,Guest}→{Beer}

Boyce-Codd-Normal Form (BCNF)

- $\mathcal{R}$ is in BCNF iff for all $\alpha \rightarrow \mathrm{B}$ in $\mathcal{R}$ at least one condition holds:
  - B $\in \alpha$ (i.e., $\alpha \rightarrow \mathrm{B}$ is trivial)
  - α is a superkey of $\mathcal{R}$

- $\mathcal{R}$ in BCNF implies $\mathcal{R}$ in 3NF
  - Proof trivial from definition

Decomposition Algorithm (BCNF)

While some relation R is not in BCNF:

Find an FD $F = X \rightarrow Y$ that violates BCNF on R

Split R into:

$R1 = (X \cup Y)$

$R2 = R - Y$

BCNFify Example for Hobbies

Iter 1

S = SSN, H = Hobby, N = Name, A = Addr, C = Cost

Schema	FDs
(S,H,N,A,C)	S,H $\rightarrow$ N,A,C S $\rightarrow$ N, A H $\rightarrow$ C

violates bcnf

key

Iter 2

Schema	FDs
(S, N,A)	S $\rightarrow$ N, A

Schema	FDs
(S,H, C)	S,H $\rightarrow$ C H $\rightarrow$ C

violates bcnf

Iter 3

Schema	FDs
(H, C)	H $\rightarrow$ C

Schema	FDs
(S,H)	

ZipCodes(<u>Street, State, City,</u> Zip)

- ZipCodes is not in BCNF
  - {Zip} → {State, City} // evil
  - {Street, State, City} → {Zip} // okay
- Redundancy in ZipCodes
  - (Vassar St, MA, Cambridge, 02139)
  - (Main St, MA, Cambridge, 02139)
  - (Mass. Ave, MA, Cambridge, 02139)
  - stores several times that 02139 belongs to MA

Decomposition of ZipCodes

- ZipCodes: {[Street, City, State, Zip]}
  - {Zip} → {City, State}   // evil
  - {Street, City, State} → {Zip}   // okay

- Applying the decomposition algorithm...
  - Street: {[Zip, Street]}
  - Cities: {[Zip, City, State]}

- Assessment
  - decomposition is lossless
  - decomposition does not preserve dependencies

Boyce-Codd-Normal Form (BCNF)

- $\mathcal{R}$ is in BCNF iff for all $\alpha \rightarrow \mathrm{B}$ in $\mathcal{R}$ at least one condition holds:
  - B $\in \alpha$ (i.e., $\alpha \rightarrow \mathrm{B}$ is trivial)
  - α is a superkey of $\mathcal{R}$

- $\mathcal{R}$ in BCNF implies $\mathcal{R}$ in 3NF
  - Proof trivial from definition

- Result
  - any schema can be decomposed losslessly into BCNF
  - but, preservation of dependencies cannot be guaranteed
  - need to trade „correctness“ for „efficiency“
  - that is why 3NF is so important in practice