Normal Forms for Relational Databases

Normal Forms for Relational Databases

- criteria for a good database design (i.e., to resolve update anomalies)
- formalized by functional (or other) dependencies

Normal Forms for Relational Databases(cont)

Normal Forms:

- 1NF, 2NF, 3NF (Codd 1972)
- Boyce-Codd NF (1974)
- Multivalued dependencies and 4NF (Zaniolo 1976 and Fagin 1977)
- Join dependencies (Rissanen 1977) and 5NF (Fagin 1979)

First Normal Form (1NF)

This simply means that attribute values are *atomic*, and is part of the definition of the relational model.

Atomic: multivalued attributes, composite attributes, and their combinations are disallowed.

There is currently a lot of interests in non-first normal form databases, particularly those where an attribute value can be a table (nested relations).

Consider the table below, adapted from Desai.

First Normal Form (1NF) (cont)

Fac_Dept	Prof	Course Preferences	
		Course	Course_Dept
Comp Sci	Smith	353 379 221	Comp Sci Comp Sci Decision Sci
	Clark	353 351 379 456	Comp Sci Comp Sci Comp Sci Mathematics
Chemistry	Turner	353 456 272	Comp Sci Mathematics Chemsitry
Mathematics	Jameison	353 379 221 456 469	Comp Sci Comp Sci Decision Sci Mathematics Mathematics

This can be transformed into:

First Normal Form (1NF) (cont)

CRS_PREF			
Prof	Course	Fac_Dept	Crs_Dept
Smith	353	Comp Sci	Comp Sci
Smith	379	Comp Sci	Comp Sci
Smith	221	Comp Sci	Decision Sci
Clark	353	Comp Sci	Comp Sci
Clark	351	Comp Sci	Comp Sci
Clark	379	Comp Sci	Comp Sci
Clark	456	Comp Sci	Mathematics
Turner	353	Chemistry	Comp Sci
Turner	456	Chemistry	Mathematics
Turner	272	Chemistry	Chemistry
Jamieson	353	Mathematics	Comp Sci
Jamieson	379	Mathematics	Comp Sci
Jamieson	221	Mathematics	Decision Sci
Jamieson	456	Mathematics	Mathematics
Jamieson	469	Mathematics	Mathematics

First Normal Form (1NF) (cont)

The representation in the figure above has the following drawbacks:

- the fact that a given professor is in a given department may be repeated,
- the association between professor and department will not be recorded unless the professor has some course references,
- the fact that a given course is offered by a given department may be repeated,
- again, this is not recorded unless someone has a preference for the course.

First Normal Form (1NF) (cont)

Suppose the FD's for these attributes are

$$F = \{Prof \rightarrow Fac\_Dept,\ Course \rightarrow Crs\_Dept\}.$$

Notice that a superkey is just a set of attributes S such that

$$S \rightarrow \{Prof,\ Course,\ Fac\_Dept,\ Crs\_Dept\} \in F^+$$

Thus the only candidate key here is $\{Prof,\ Course\}$.

First Normal Form (1NF) (cont)

These problems arise because *Fac_Dept* depends only on *Prof* and not on *Course*, and similarly *Crs_Dept* depends only on *Course* and not on *Prof*.

We can recognize and avoid these problems using functional dependencies.

Second Normal Form (2NF)

A *prime* attribute is one that is part of a candidate key. Other attributes are *non-prime*.

Definition: In an FD $X \rightarrow Y$, Y is *fully functionally dependent* on X if there is no $Z \subset X$ such that $Z \rightarrow Y$. Otherwise Y is *partially* dependent on X.

Proper Subset

Definition (*Second Normal Form*): A relation scheme is in second normal form (2NF) if all non-prime attributes are fully functionally dependent on the candidate keys.

A database scheme is in 2NF if all its relations are in 2NF.

Second Normal Form (2NF) (cont)

Possible 2NF decomposition of the relation above is:

COURSE_PREF	
Prof	Course
Smith	353
Smith	379
Smith	221
Clark	353
Clark	351
Clark	379
Clark	456
Turner	353
Turner	456
Turner	272
Jamieson	353
Jamieson	379
Jamieson	221
Jamieson	456
Jamieson	469

COURSE	
Course	Dept
353	Comp Sci
379	Comp Sci
221	Decision Sci
351	Comp Sci
456	Mathematics
272	Chemistry
469	Mathematics

FACULTY	
Prof	Dept
Smith	Comp Sci
Clark	Comp Sci
Turner	Chemistry
Jamieson	Mathematics

Second Normal Form (2NF) (cont)

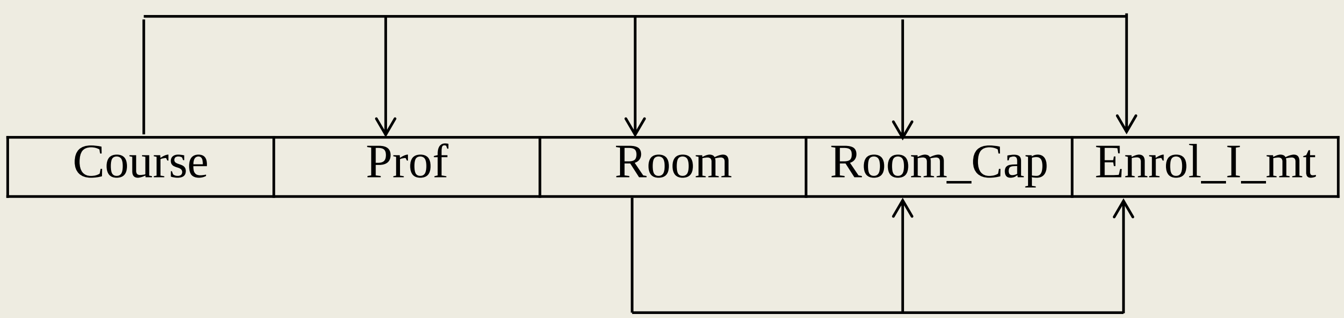
Question: What relational algebra expression recovers *CRS_PREF* from these relations?

Answer: Join

Second Normal Form (2NF) (cont)

2NF does not completely eliminate the kind of anomaly we saw before:

TEACHES				
Course	Prof	Room	Room_Cap	Enrol_I_mt
353	Smith	A532	45	40
351	Smith	C320	100	60
355	Clark	H940	400	300
456	Turner	B278	50	45
459	Jamieson	D110	50	45



Second Normal Form (2NF) (cont)

This is in 2NF but:

If another course uses say Room A532, then the fact that A532 has *Room_Cap* of 45 and *Enrol_Lmt* of 40 will be stored twice.

If course 355 is deleted, then the fact that H940 has *Room_Cap* of 400 and *Enrol_Lmt* of 300 will be lost.

This we can also fix by adding further restrictions on functional dependencies.

Third Normal Form (3NF)

Definition: An FD X→Y is a transitive dependency if there is a Z that is not a subset of any key, such that $X \rightarrow Z$ and $Z \rightarrow Y$ and $Z \nrightarrow X$ hold.

The attributes of Y are transitively dependent on X.

e.g. Room_Cap is transitively dependent on {*Course*}, since {*Course*} → {*Room*} and {*Room*} → {*Room_Cap*} hold, and {*Room*} is not a subset of any key.

Third Normal Form (3NF) (cont)

Definition (Third Normal Form): A relation scheme is in *third normal form (3NF)* if for all non-trivial FD's of the form X→A that hold, either X is a superkey or A is a prime attribute.

Note: a FD X → Y is trivial iff Y is a subset of X.

Alternative definition: A relation scheme is in third normal form if every non-prime attribute is fully functionally dependent on the keys and not transitively dependent on any key.

A database scheme is in 3NF if all its relations are in 3NF.

Third Normal Form (3NF) (cont)

TEACHES can be decomposed into 3NF:

ROOM_DETAILS		
Room	Room_Cap	Enrol_I_mt
A532	45	40
C320	100	60
B278	50	45
D110	50	45
H940	400	300

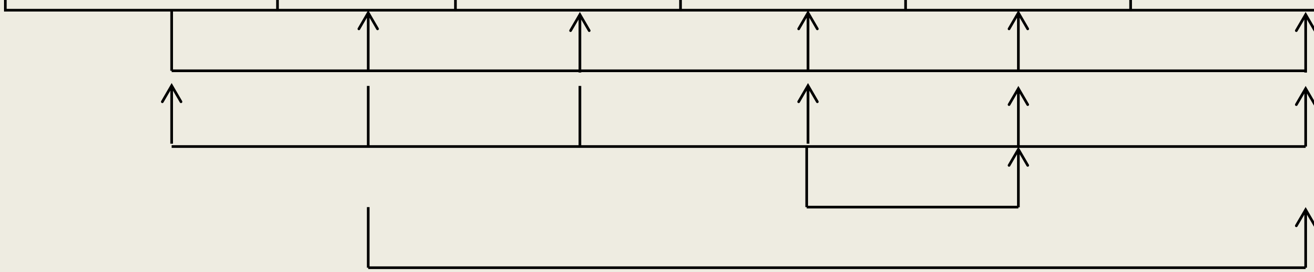
COURSE_DETAILS		
Course	Prof	Room
353	Smith	A532
351	Smith	C320
456	Turner	B278
459	Jamieson	D110
355	Clark	H940

Third Normal Form (3NF) (cont)

Another example:

LOTS

Property_Id	City	Lot_No	Area	Price	Tax_Rate

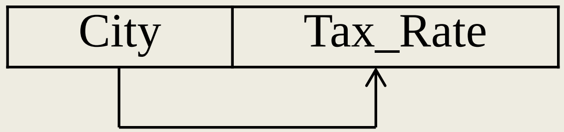


This is not in 2NF since *City*→*Tax_Rate*, *Tax_Rate* is not prime, and {*City*,*Lot_No*} is a key, making *Tax_Rate* partially dependent on a key.

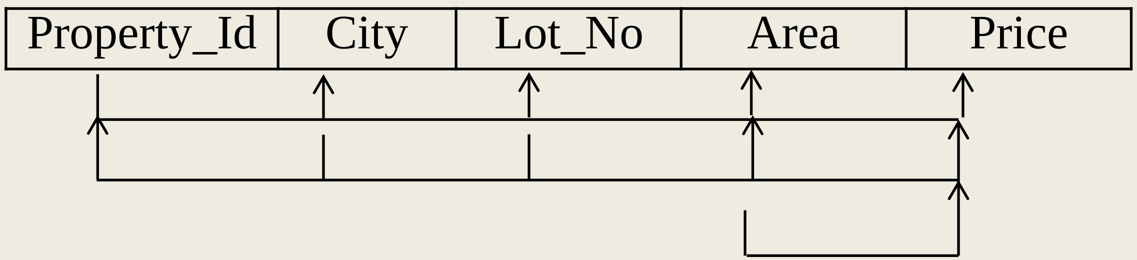
We could fix this:

Third Normal Form (3NF) (cont)

LOTS1



LOTS2



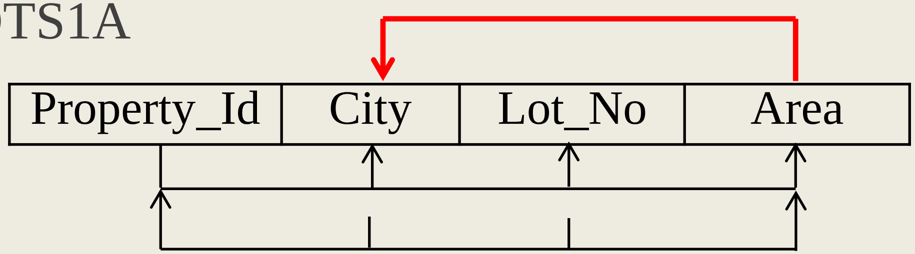
Now we have 2NF but not 3NF, since *Area* → *Price*, {*Area*} is not a superkey and Price is not prime.

Note: the transitive dependency : *Property_Id*→ *Area* → *Price*.

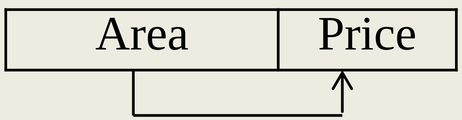
We could fix this too:

Third Normal Form (3NF) (cont)

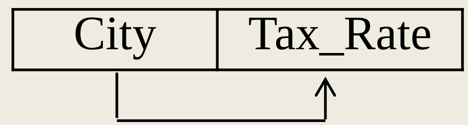
LOTS1A



LOTS1B



LOTS2



Suppose also that *Area*→*City*. The relations schemes are still in 3NF since *City* is a prime attribute. However, there can be anomalies, just as before. We need more restrictions still to fix these.

Boyce-Codd Normal Form (BCNF)

Definition (*Boyce-Codd Normal Form*):

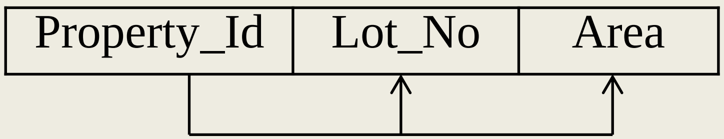
A relation scheme is in *Boyce-Codd* Normal Form (BCNF) if whenever $X \rightarrow A$ holds and $X \rightarrow A$ is non-trivial, X is a superkey.

A database scheme is in BCNF if all its relations are in BCNF.

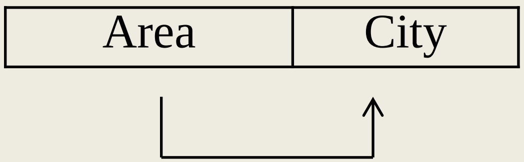
We can make our example into BCNF:

Boyce-Codd Normal Form (BCNF)(cont)

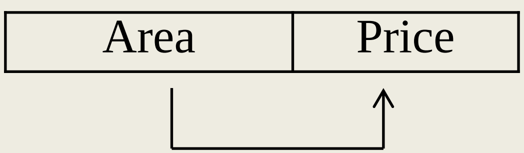
LOTS1AA



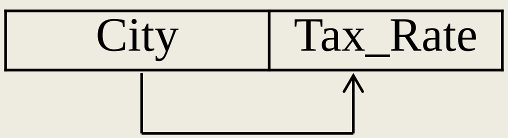
LOTS1AB



LOTS1B



LOTS2



Boyce-Codd Normal Form (BCNF)(cont)

LOTS1AA

Property_Id	Lot_No	Area

LOTS1AB

Area	City	Price

LOTS2

City	Tax_Rate

Learning Outcomes

- Definitions for 1NF, 2NF, 3NF and BCNF

- Determine the highest NF that a given relation is in