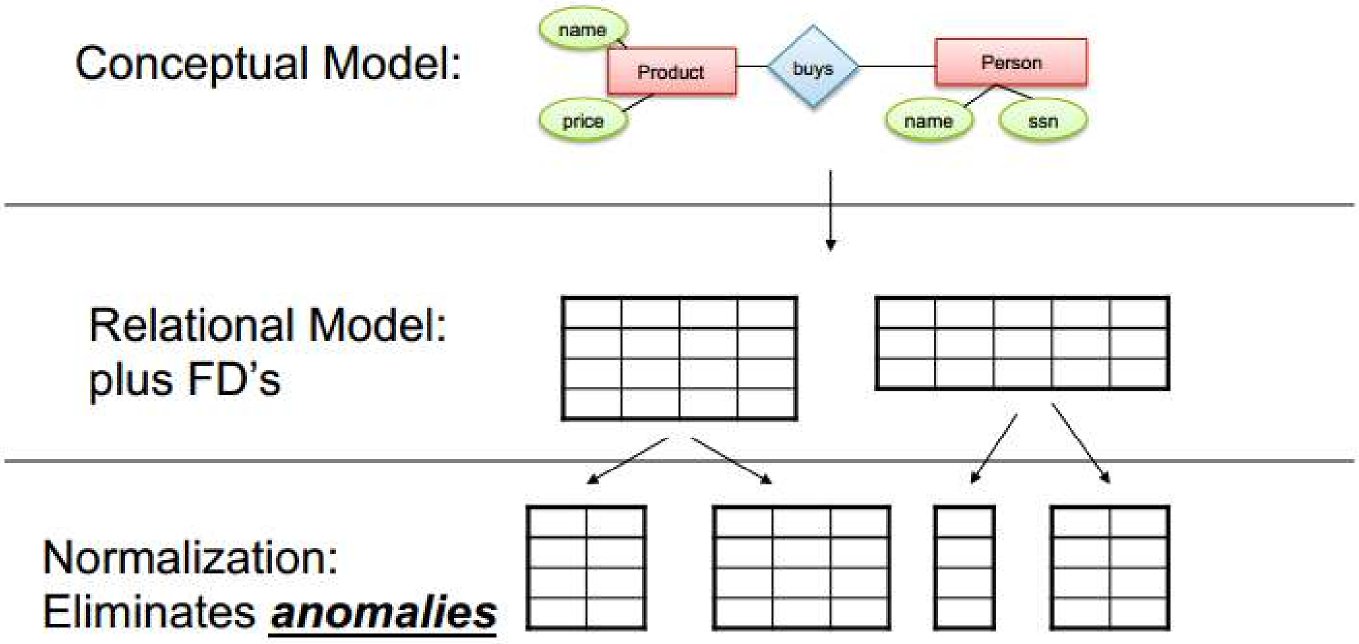
CSE 344
section 7

2014 Fall

Part I --- Conceptual Design



Anomalies(redundancy, update/deletion anomalies), functional dependencies, attribute closures, BCNF decomposition

* BCNF

Problem 1.

R(A,B,C,D,E,F,G) with functional dependencies:

$A \rightarrow D$

$D \rightarrow C$

$F \rightarrow E,G$

$D,C \rightarrow B,F$

Decompose R into BCNF.

Problem 1 -- Solution.

R(A,B,C,D,E,F,G)

A → D

D → C F → E,G

D,C → B,F

From A--> D, {A}+ = {A,B,C,D,E,F,G}, it is useless.

From D-->C, {D}+ = {D,C,B,F,E,G}, we can decompose R to R_1={D,C,B,F,E,G} and R_2={A,D}

From F-->E,G, {F}+ = {F,E,G}, we can further decompose R1 to R_{11}={E,F,G} and R_{12}={C,D,B,F}

Problem 2.

Relation R(A,B,C,D,E,F) and functional dependencies:

$A \rightarrow BC$ and $D \rightarrow AF$

Decompose R into BCNF.

Problem 2 -- Solution.

Relation R(A,B,C,D,E,F) and FD's A → BC and D → AF

A→BC violates BCNF since A+ = ABC ≠ ABCDEF. So we split R into R1(<u>A</u>BC) and R2(<u>A</u>DEF).

The only non-trivial FD in R1 is A→BC, and A+ = ABC, so R1 is in BCNF.

R2 has a non-trivial dependency D→AF that violates BCNF because D+ = ADF ≠ ADEF. So we split R2 into R21(<u>D</u>AF) and R22(<u>D</u>E). Both of these are in BCNF since they have no non-trivial dependencies that are not superkeys.

Part II -- Lossless-join decomposition

Consider the relation R(A,B,C,D,E)

with FDs: {AB → C, BC → D, AD → E}. We want to check whether the decomposition {ABC, BCD, ADE} is a lossless-join decomposition.

Part II -- Lossless-join decomposition

Consider the relation R(A,B,C,D,E)
with FDs: {AB → C, BC → D, AD → E}. We want to check whether the decomposition {ABC, BCD, ADE} is a lossless-join decomposition.

Start by constructing a tableau as follows:

A	B	C	D	E
a	b	c	d_1	e_1
a_2	b	c	d	e_2
a	b_3	c_3	d	e

Part II -- Lossless-join decomposition

A	B	C	D	E
a	b	c	d_1	e_1
a_2	b	c	d	e_2
a	b_3	c_3	d	e

BC -> D

A	B	C	D	E
a	b	c	d	e_1
a_2	b	c	d	e_2
a	b_3	c_3	d	e

Notice that we use a common distinguished variable (a,b,c,...) if the variable is a key, otherwise we use a non-distinguished symbol (e^1, e^2, b^3,...) We next start applying the fd's! Notice that the 1st and 2nd row have the same distinguished B and C attributes. Hence, D must be the same by the fd BC -> D. This results in unifying d^1 = d.

Part II -- Lossless-join decomposition

But now rows 1 and 3 agree on A and D. Because AD -> E, we unify e1 = e. Now, we have:

A	B	C	D	E
a	b	c	d	e_1
a_1	b	c	d	e_2
a	b_1	c_1	d	e

AD -> E

A	B	C	D	E
a	b	c	d	e
a_1	b	c	d	e_2
a	b_1	c_1	d	e

Part II -- Lossless-join decomposition

```
A  |  B  |  C  |  D  |  E
-----------------------------
 a  |  b  |  c  |  d  |  e
 a1 |  b  |  c  |  d  |  e2
 a  |  b1 |  c1 |  d  |  e
```

Row 1 contains only distinguished symbols, hence the algorithm terminates and the answer is YES, the decomposition is lossless. If we could not apply any fd and no row had only distinguished symbols, we would terminate with NO.

This algorithm is called "chase".

Problem 3

The relation is R (A, B, C, D, E) and the FDs :

$A \rightarrow E$

$B,C \rightarrow A$

$D,E \rightarrow B$

Decompose R into BCNF.

Problem 3 – solution 1

The relation is R (A, B, C, D, E) and the

FDs : $A \rightarrow E$, $B,C \rightarrow A$, and $D,E \rightarrow B$

Notice that {A}+ = {A,E}, violating the BCNF condition. We split R to R_1(A,E) and R_2(A,B,C,D).

R_1 satisfies BCNF now, but R_2 not because of: {B,C}+ = {B,C,A}. Notice that the fd D,E --> B has now disappeared and we don't need to consider it! Split R_2 to: R_{2A}(B,C,A) and R_{2B}(B,C,D).

Problem 3 – solution 2

The relation is R (A, B, C, D, E) and the
FDs : A → E, BC → A, and DE → B

Can we split differently? Let's try with the violation {B,C}+ = {B,C,A,E}. We initially split to R_1(B,C,A,E) and R_2(B,C,D). Now we need to resolve for R_1 the violation {A}+ = {A,E}. So we split again R_1 to R_1A(A,E) and R_1B(A,B,C). The same!

We can also start splitting by considering the BCNF violation {D,E}+ = {D,E,B}. Which is the resulting BCNF decomposition in this case? (it will be a different one)

Part III - 3rd Normal Form

- Relation R:
  - $R = (J, K, L)$

    $F = \{JK \rightarrow L,\ L \rightarrow K\}$

  - BCNF?

    R1=(L,K), R2=?

Dependency Preserving

- Let F_i be the set of dependencies F^+ that include only attributes in R_i.
  - A decomposition is dependency preserving, if

    $$(F_1 \cup F_2 \cup \ldots \cup F_n)^+ = F^+$$

  - If it is not, then checking updates for violation of functional dependencies may require computing joins, which is expensive.

Third Normal Form

- A relation schema *R* is in third normal form (3NF) if for all:

  $$\alpha \rightarrow \beta \text{ on R}$$

  at least one of the following holds:
  - $\alpha \rightarrow \beta$ is trivial (i.e., $\beta \in \alpha$)
  - α is a superkey for *R*
  - Each attribute *A* in $\beta - \alpha$ is contained in a candidate key for *R.*

    (**NOTE***:* each attribute may be in a different candidate key)
- If a relation is in BCNF it is in 3NF
  - since in BCNF one of the first two conditions above must hold.
- Third condition is a minimal relaxation of BCNF that ensures dependency preservation.

Redundancy in 3NF

- There is some redundancy in this schema
- Example of problems due to redundancy in 3NF
  - $R = (J, K, L)$
    $F = \{JK \rightarrow L, L \rightarrow K\}$

J	L	K
j_1	l_1	k_1
j_2	l_1	k_1
j_3	l_1	k_1
null	l_2	k_2

- repetition of information (e.g., the relationship l_1, k_1)
- need to use null values (e.g., to represent the relationship l_2, k_2 where there is no corresponding value for J).

Questions?