Third Normal Form

- Compromise – Not all redundancy removed, but dependency preserving decompositions are <u>always</u> possible
- 3NF decomposition is based on a *minimal cover*

51

Minimal Cover

- A *minimal cover* of a set of dependencies, *T*, is a set of dependencies, *U*, such that:
  - *U* is equivalent to *T* ($T^+ = U^+$)
  - All FDs in *U* have the form $X \rightarrow A$ where *A* is a single attribute
  - It is not possible to make *U* smaller by
    - Deleting an FD
    - Deleting an attribute from an FD (either from LHS or RHS)

    FDs and attributes that can be deleted are called *redundant*

52

Computing Minimal Cover

- **Example**: $T = \{ABH \rightarrow C, A \rightarrow D, C \rightarrow E, BGH \rightarrow F, F \rightarrow AD, E \rightarrow F, BH \rightarrow E\}$
- **step 1**: Make RHS of each FD into a single attribute
  - *Algorithm*: Use the decomposition inference rule for FDs
  - Example: $F \rightarrow AD$ replaced by $F \rightarrow A$, $F \rightarrow D$
- **step 2**: Eliminate redundant attributes from LHS.
  - *Algorithm*: If FD $XB \rightarrow A \in T$ (where *B* is a single attribute) and $X \rightarrow A$ is entailed by *T*, then *B* was unnecessary
  - Example: Can an attribute be deleted from $ABH \rightarrow C$?
    - Compute AB^+_T, AH^+_T, BH^+_T.
    - Since $C \in (BH)^+_T$, $BH \rightarrow C$ is entailed by *T* and *A* is redundant in $ABH \rightarrow C$.

53

Computing Minimal Cover (con't)

- **step 3**: Delete redundant FDs from *T*
  - *Algorithm*: If $T - \{f\}$ entails *f*, then *f* is redundant
    - If *f* is $X \rightarrow A$ then check if $A \in X^+_{T-\{f\}}$
  - Example: $BGH \rightarrow F$ is entailed by $E \rightarrow F$, $BH \rightarrow E$, so it is redundant
- Note: Steps 2 and 3 cannot be reversed!! See the textbook for a counterexample

54

Synthesizing a 3NF Schema

Starting with a schema **R** = (*R*, *T*)

- **step 1**: Compute a minimal cover, *U*, of *T*. The decomposition is based on *U*, but since $U^+ = T^+$ the same functional dependencies will hold
  - A minimal cover for

    $T=\{ABH \rightarrow C, A \rightarrow D, C \rightarrow E, BGH \rightarrow F, F \rightarrow AD, E \rightarrow F, BH \rightarrow E\}$

    is

    $U=\{BH \rightarrow C, A \rightarrow D, C \rightarrow E, F \rightarrow A, E \rightarrow F\}$

55

Synthesizing a 3NF schema (con't)

- **step 2**: Partition *U* into sets $U_1, U_2, \ldots U_n$ such that the LHS of all elements of U_i are the same
  - $U_1 = \{BH \rightarrow C\}$, $U_2 = \{A \rightarrow D\}$, $U_3 = \{C \rightarrow E\}$, $U_4 = \{F \rightarrow A\}$, $U_5 = \{E \rightarrow F\}$

    (*in general* U_i *can have more than one FD*)

56

Synthesizing a 3NF schema (con't)

- **step 3**: For each U_i form schema $\mathbf{R_i} = (R_i, U_i)$, where R_i is the set of all attributes mentioned in U_i
  - Each FD of U will be in some $\mathbf{R_i}$. Hence the decomposition is *dependency preserving*
  - $\mathbf{R_1} = (BHC;\ BH \rightarrow C)$, $\mathbf{R_2} = (AD;\ A \rightarrow D)$, $\mathbf{R_3} = (CE;\ C \rightarrow E)$, $\mathbf{R_4} = (FA;\ F \rightarrow A)$, $\mathbf{R_5} = (EF;\ E \rightarrow F)$

Synthesizing a 3NF schema (con't)

- **step 4**: If no R_i is a superkey of **R,** add schema $\mathbf{R_0} = (R_0,\{\})$ where R_0 is a key of **R**.
  - $\mathbf{R_0} = (BGH, \{\})$
    - $\mathbf{R_0}$ might be needed when not all attributes are necessarily contained in $R_1 \cup R_2 \ldots \cup R_n$
      - A missing attribute, A, must be part of all keys (since it's not in any FD of U, deriving a key constraint from U involves the augmentation axiom)
    - $\mathbf{R_0}$ might be needed even if all attributes are accounted for in $R_1 \cup R_2 \ldots \cup R_n$
      - Example: $(ABCD; \{A \rightarrow B, C \rightarrow D\})$. Step 3 decomposition: $R_1 = (AB; \{A \rightarrow B\})$, $R_2 = (CD; \{C \rightarrow D\})$. Lossy! Need to add (AC; { }), for losslessness
  - Step 4 guarantees lossless decomposition.

3NF Synthesis

- The resulting decomposition, $\mathbf{R_0, R_1, \ldots R_n}$, is
  - Dependency preserving (since every FD in U is a FD of some schema)
  - Lossless (although this is not obvious)
  - In 3NF (although this is not obvious)
- Strategy for decomposing a relation
  - Use 3NF decomposition first to get lossless, dependency preserving decomposition
  - If any resulting schema is not in BCNF, split it using the BCNF algorithm (but this may yield a non-dependency preserving result)

Normalization

- By limiting redundancy, normalization helps maintain consistency and saves space
- But performance can suffer because related information that was stored in a single relation is now distributed among several
- **Example**: A join is required to get the names and grades of all students taking CS305 in S2002.

```
SELECT S.Name, T.Grade
FROM Student S, Transcript T
WHERE S.Id = T.StudId AND
      T.CrsCode = 'CS305' AND T.Semester = 'S2002'
```

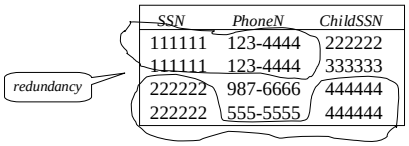

Denormalization

- **Tradeoff**: *Judiciously* introduce redundancy to improve performance
- **Example**: Add attribute *Name* to Transcript

```
SELECT T.Name, T.Grade
FROM Transcript' T
WHERE T.CrsCode = 'CS305' AND T.Semester = 'S2002'
```

  - Join is avoided
  - If queries are asked more frequently than Transcript is modified, added redundancy might improve average performance
  - But, Transcript' is no longer in BCNF since key is (*StudId, CrsCode, Semester*) and *StudId* $\rightarrow$ *Name*

Fourth Normal Form



- Relation has redundant data
- Yet it is in BCNF (since there are no non-trivial FDs)
- Redundancy is due to set valued attributes (in the E-R sense), not because of the FDs

Multi-Valued Dependency

- **Problem**: multi-valued (or binary join) dependency
  - **Definition**: If every instance of schema **R** can be (losslessly) decomposed using attribute sets (*X, Y*) such that:

$$\mathbf{r} = \Pi_X(\mathbf{r}) \bowtie \Pi_Y(\mathbf{r})$$

then a *multi-valued dependency*

$$\mathbf{R} = \Pi_X(\mathbf{R}) \bowtie \Pi_Y(\mathbf{R})$$

holds in **r**

Ex: Person = $\Pi_{SSN,PhoneN}$(Person) $\bowtie$ $\Pi_{SSN,ChildSSN}$(Person)

63

Fourth Normal Form (4NF)

- A schema is in *fourth normal form* (4NF) if for every non-trivial multi-valued dependency:

$$R = X \bowtie Y$$

either:

- $X \subseteq Y$ or $Y \subseteq X$ (trivial case); or
- $X \cap Y$ is a superkey of R (*i.e.*, $X \cap Y \rightarrow R$)

64

Fourth Normal Form (Cont'd)

- *Intuition*: if $X \cap Y \rightarrow R$, there is a unique row in relation **r** for each value of $X \cap Y$ (hence no redundancy)
  - Ex: *SSN* does not uniquely determine *PhoneN* or *ChildSSN*, thus Person is not in 4th normal form.
- *Solution*: Decompose *R* into *X* and *Y*
  - Decomposition is lossless – but not necessarily dependency preserving (since 4NF implies BCNF)

65

4NF Implies BCNF

- Suppose *R* is in 4NF and $X \rightarrow Y$ is an FD.
  - *R1 = XY, R2 = R-Y* is a lossless decomposition.
  - Then R has the multi-valued dependency:

$$R = R_1 \bowtie R_2$$

  - Since *R* is in 4NF, one of the following must hold :
    - $XY \subseteq R - Y$ (an impossibility)
    - $R - Y \subseteq XY$ (i.e., $R = XY$ and *X* is a superkey)
    - $XY \cap R - Y$ (= *X*) is a superkey
  - Hence $X \rightarrow Y$ satisfies BCNF condition

66