CS2102 Database Systems

SCHEMA REFINEMENT: NORMAL FORMS

Normal Forms

* A normal form restricts the set of data dependencies that are allowed to hold on a schema to avoid certain undesirable redundancy and update problems in the database.

Normal Forms

- There are several normal forms, each providing guidance on good schema designs
- We focus on two normal forms that are based on FDs:
 - Boyce-Codd Normal Form (BCNF)
 - Third Normal Form (3NF)
- ❖ Definitions of BCNF and 3NF assume that each FD is of the form $X \rightarrow A$ where A is a single attribute.

Boyce-Codd Normal Form (BCNF)

- * A relation schema R (with FDs F) is in Boyce-Codd normal form if for every non-trivial FD $X \rightarrow A$ in F, X is a superkey.
- * A non-trivial FD $X \rightarrow A$ that holds on R is said to violate BCNF if X is not a superkey of R

Example

- ❖ Consider the MovieList schema with FDs F = { title → director, address → phone, {address, time} → title }
- Recall that the only key is {address, time}
- ❖ FDs in F that violate BCNF are
 - title \rightarrow director
 - address \rightarrow phone
- Thus, MovieList is not in BCNF

Decomposition into BCNF

- ❖ Let $X \rightarrow A$ be an FD in F that causes violation of BCNF
- Decompose R into

$$R_1 = XA$$

$$R_2 = R - A$$

❖ If R₁ or R₂ is not in BCNF, then decompose them further as described.

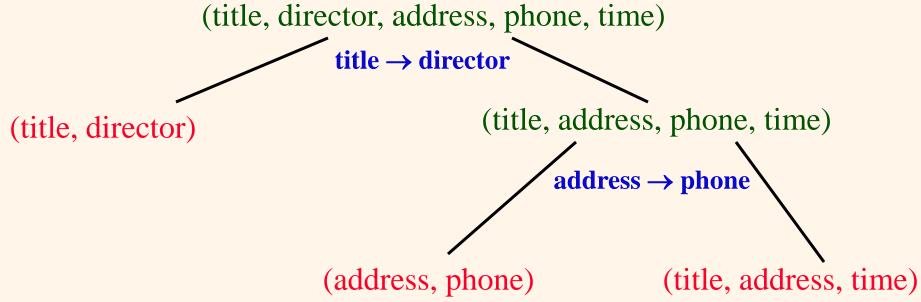
Decomposition into BCNF

- \bullet Let X \rightarrow A be an FD in F that causes violation of BCNF
- Decompose R into

$$R_1 = XA$$

$$R_2 = R - A$$

 \bullet If R_1 or R_2 is not in BCNF, then decompose them further as described.



Decomposition into BCNF

- * Decomposition $\{R_1, R_2, ..., R_n\}$ is in BCNF if each R_i is in BCNF (w.r.t. F_{Ri})
- BCNF decompositions are lossless join decomposition
- But, not all schema has a dependencypreserving BCNF decomposition

Example

- ❖ Consider R (course, prof, time) with FDs $F = \{ course \rightarrow prof, \{ prof, time \} \rightarrow course \}$
- Keys are {course, time} and {prof, time}
- ❖ R is not in BCNF because course is not a superkey
- ❖ Decomposition into R_1 (course, prof) and R_2 (course, time) is a lossless join but does not preserve the FD {prof, time} → course

Third Normal Form (3NF)

- ❖ 3NF is a less restrictive normal form that always guarantees a lossless join decomposition that preserves dependencies.
- ❖ A relation schema R (with FDs F) is in third normal form if for every non-trivial FD X → A in F (where A is a single attribute), either X is a superkey or A is a prime attribute.
- * A non-trivial FD $X \rightarrow A$ that holds on R is said to violate 3NF if X is not a superkey of R and A is a nonprime attribute
- * R in BCNF \Rightarrow R in 3NF

Example

- ❖ Consider again R (course, prof, time) with FDs { course \rightarrow prof, {prof, time} \rightarrow course }
- Keys are {course, time} and {prof, time}
- * R is in 3NF because both prof and course are prime attributes

Instance of R

prof	time	course
Codd	Tue 3pm	DB101
Codd	Thur 9am	DB101
Gray	Tue 4pm	CS323
Gray	Fri 10am	IT201

Decomposition into 3NF

- Synthesis Approach
- Input: Schema R with FDs F which is a minimal cover
- Output: A dependency preserving, lossless join 3NF decomposition of R

Decomposition into 3NF (cont'd)

- * Initialize D = ϕ
- Apply union rule to combine FDs with same LHS into a single FD.
 - Let $F = \{f_1, f_2, ..., f_n\}$ be the resultant set of FDs
- For each f_i of the form $X_i \rightarrow A_i$ do
 - Create a relation schema $R_i(X_i, A_i)$ for FD f_i
 - Insert the schema R_i into D
- * Choose a key K of R and insert a relation schema $R_{n+1}(K)$ into D
- Remove redundant schema from D
 - Delete R_i from D if $R_i \subseteq R_j$ where $R_i \in D$
- * Return D

Example

- ❖ Consider R(A, B, C, D, E) with FDs F = {ABCD → E, E → D, A → B, AC → D}
- * A minimal cover of F is $\{AC \rightarrow E, E \rightarrow D, A \rightarrow B\}$
- Only key is AC
- * R is not in 3NF because $A \rightarrow B$ violates 3NF (A is not a superkey and B is not a prime attribute)
- 3NF decomposition of R
 - Create a schema for each FD: R_1 (A, C, E), R_2 (E, D), R_3 (A, B)
 - Create a schema for a key of R: R_4 (A, C)
 - Remove redundant schema: R_4 is redundant because $R_4 \subseteq R_1$
 - 3NF decomposition is R_1 (A, C, E), R_2 (E, D), R_3 (A, B)

Remarks on 3NF Decomposition

- * A decomposition $\{R_1, R_2, ..., R_n\}$ is in 3NF if each R_i is in 3NF (w.r.t. F_{Ri})
- The 3NF decomposition produced by synthesis approach may not be unique
 - Choice of minimal cover
 - Choice of redundant relation schema being removed

BCNF vs. 3NF

- ❖ BCNF is lossless join (may not be dependency preserving
- * 3NF is lossless join and dependency preserving
- ❖ Recall R(course, prof, time) with FDs { course → prof, {prof, time} → course }
 - Keys are {course, time} and {prof, time}
 - R is in 3NF but not in BCNF
 - BCNF decomposition { R₁(course, prof), R₂(course, time) } is lossless but not dependency preserving

Another Example

- Consider schema Contract (contractid, supplierid, projectid, deptid, partid, qty, value)
- CSJDPQV for short
- Contract C is an agreement that supplier S will supply
 Q items of part P to project J associated with department
 D; value of this contract is V
 - Contract id C is a key: C → CSJDPQV
 - A project purchase a part using a single contract:
 JP → C
 - A department purchase at most one part from a supplier: SD → P
 - Each project deals with a single supplier: $J \rightarrow S$

Example – BCNF Decomposition

- * FDs F = { $C \rightarrow CSJDPQV, JP \rightarrow C, SD \rightarrow P, J \rightarrow S$ }
- ❖ From JP → C, C → CSJDPQV and transitivity, we have JP → CSJDPQV
- ❖ SD → P violates BCNF since SD is not a key, decompose CSJDPQV into CSJDQV and SDP
- ❖ From $J \rightarrow S$, decompose CSJDQV into JS and CJDQV
- Decomposition is lossless
- ❖ Decomposition does not preserve FD JP \rightarrow C
 - Need to join the two relations to check that the FD is not violated.
 - Can add a relation CJP to the decomposition if CJP is in BCNF

Example – 3NF Synthesis

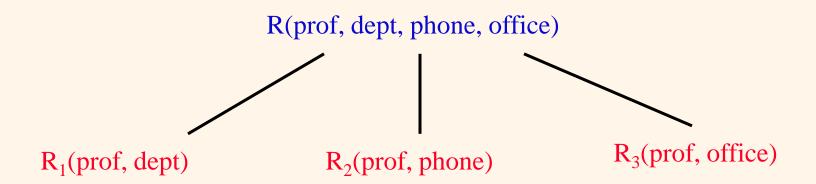
- * FDs F = { $C \rightarrow CSJDPQV, JP \rightarrow C, SD \rightarrow P, J \rightarrow S$ }
- * F is not a minimal cover.
 - Replace $C \to CSJDPQV$ with $\{C \to S, C \to J, C \to D, C \to P, C \to Q, C \to V\}$
 - Remove $C \rightarrow P$ from F since it is implied by $C \rightarrow S$, $C \rightarrow D$ and $SD \rightarrow P$
 - Remove $C \rightarrow S$ from F since it is implied by $C \rightarrow J$ and $J \rightarrow S$
- * Minimal cover $F' = \{C \rightarrow J, C \rightarrow D, C \rightarrow Q, C \rightarrow V, JP \rightarrow C, SD \rightarrow P, J \rightarrow S\}$

Example – 3NF Synthesis

- * Minimal cover $F' = \{C \rightarrow J, C \rightarrow D, C \rightarrow Q, C \rightarrow V, JP \rightarrow C, SD \rightarrow P, J \rightarrow S\}$
- ❖ Combine FDs with same LHS $F' = \{C \rightarrow JDQV, JP \rightarrow C, SD \rightarrow P, J \rightarrow S\}$
- Create relations CJDQV, CJP, SDP, JS
- * Remark: You can combine relations with C as key
 - e.g., CJDQV and CJP to CJDQVP

Remarks on Decomposition

- Decomposition is a last resort to solve problems of redundancy and anomalies
- Too much decomposition can be harmful
- ❖ Example: R(prof, dept, phone, office) with FD { prof → dept, phone, office }



 Consider de-normalization for performance reasons