Relational Normalization Theory

Chapter 8

Limitations of E-R Designs

- Provides a set of guidelines, does not result in a unique database schema
- Does not provide a way of evaluating alternative schemas
- Normalization theory provides a mechanism for analyzing and refining the schema produced by an E-R design

Redundancy

- Dependencies between attributes cause redundancy
  - Ex. All addresses in the same town have the same zip code

SSN	*Name*	*Town*	*Zip*
1234	Joe	Stony Brook	11790
4321	Mary	Stony Brook	11790
5454	Tom	Stony Brook	11790
	…….		

Redundancy

Redundancy and Other Problems

- Set valued attributes in the E-R diagram result in multiple rows in corresponding table
- Example: Person (*SSN, Name, Address, Hobbies*)
  - A person entity with multiple hobbies yields multiple rows in table Person
    - Hence, the association between *Name* and *Address* for the same person is stored redundantly
  - *SSN* is key of entity set, but (*SSN*, *Hobby*) is key of corresponding relation
    - The relation Person can't describe people without hobbies

Example

ER Model

SSN	*Name*	*Address*	*Hobby*
1111	Joe	123 Main	{biking, hiking}

Relational Model

SSN	*Name*	*Address*	*Hobby*
1111	Joe	123 Main	biking
1111	Joe	123 Main	hiking
	…….		

Redundancy

Anomalies

- Redundancy leads to anomalies:
  - **Update anomaly**: A change in *Address* must be made in several places
  - **Deletion anomaly**: Suppose a person gives up all hobbies. Do we:
    - Set Hobby attribute to null? No, since *Hobby* is part of key
    - Delete the entire row? No, since we lose other information in the row
  - **Insertion anomaly**: *Hobby* value must be supplied for any inserted row since *Hobby* is part of key

Decomposition

- **Solution**: use two relations to store Person information
  - Person1 (*SSN, Name, Address*)
  - Hobbies (*SSN, Hobby*)
- The decomposition is more general: people with hobbies can now be described
- No update anomalies:
  - Name and address stored once
  - A hobby can be separately supplied or deleted

7

Normalization Theory

- Result of E-R analysis need further refinement
- Appropriate decomposition can solve problems
- The underlying theory is referred to as *normalization theory* and is based on *functional dependencies* (and other kinds, like *multivalued dependencies*)

8

Functional Dependencies

- **Definition:** A *functional dependency* (FD) on a relation schema **R** is a constraint $\boldsymbol{X \rightarrow Y}$, where *X* and *Y* are subsets of attributes of **R.**
- **Definition**: An FD $\boldsymbol{X \rightarrow Y}$ is *satisfied* in an instance **r** of **R** if for every pair of tuples, *t* and s: if *t* and *s* agree on all attributes in *X* then they must agree on all attributes in *Y*
  - Key constraint is a special kind of functional dependency: all attributes of relation occur on the right-hand side of the FD:
    - $SSN \rightarrow SSN, Name, Address$

9

Functional Dependencies

- $Address \rightarrow ZipCode$
  - Stony Brook's ZIP is 11733
- $ArtistName \rightarrow BirthYear$
  - Picasso was born in 1881
- $Autobrand \rightarrow Manufacturer, Engine\ type$
  - Pontiac is built by General Motors with gasoline engine
- $Author, Title \rightarrow PublDate$
  - Shakespeare's Hamlet published in 1600

10

Functional Dependency - Example

- Brokerage firm allows multiple clients to share an account, but each account is managed from a single office and a client can have no more than one account in an office
  - HasAccount (*AcctNum, ClientId, OfficeId*)
    - keys are (*ClientId, OfficeId*), (*AcctNum, ClientId*)
  - $Client, OfficeId \rightarrow AcctNum$
  - $AcctNum \rightarrow OfficeId$
    - Thus, attribute values need not depend only on key values

11

Entailment, Closure, Equivalence

- **Definition**: If *F* is a set of FDs on schema **R** and f is another FD on **R**, then *F entails* f if every instance **r** of **R** that satisfies every FD in *F* also satisfies f
  - Ex: $F = \{A \rightarrow B, B \rightarrow C\}$ and f is $A \rightarrow C$
    - If $Streetaddr \rightarrow Town$ and $Town \rightarrow Zip$ then $Streetaddr \rightarrow Zip$
- **Definition**: The *closure* of *F*, denoted F^+, is the set of all FDs entailed by *F*
- **Definition**: *F* and *G* are *equivalent* if *F* entails *G* and *G* entails *F*

12

Entailment (cont'd)

- Satisfaction, entailment, and equivalence are *semantic* concepts – defined in terms of the actual relations in the "real world."
  - They define *what these notions are*, **not** how to compute them
- How to check if F entails f or if F and G are equivalent?
  - Apply the respective definitions for all possible relations?
    - *Bad idea*: might be infinite in number for infinite domains
    - Even for finite domains, we have to look at relations of *all* arities
  - **Solution**: find algorithmic, *syntactic* ways to compute these notions
    - *Important*: The syntactic solution must be "correct" with respect to the semantic definitions
    - Correctness has two aspects: *soundness* and *completeness* – see later

13

Armstrong's Axioms for FDs

- This is the *syntactic* way of computing/testing the various properties of FDs
- **Reflexivity**: If $Y \subseteq X$ then $X \rightarrow Y$ (trivial FD)
  - *Name, Address* $\rightarrow$ *Name*
- **Augmentation**: If $X \rightarrow Y$ then $XZ \rightarrow YZ$
  - If *Town* $\rightarrow$ *Zip* then *Town, Name* $\rightarrow$ *Zip, Name*
- **Transitivity**: If $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

14

Soundness

- Axioms are *sound*: If an FD $f: X \rightarrow Y$ can be derived from a set of FDs F using the axioms, then f holds in every relation that satisfies every FD in F.
- Example: Given $X \rightarrow Y$ and $X \rightarrow Z$ then

$X \rightarrow XY$ *Augmentation by X*
$YX \rightarrow YZ$ *Augmentation by Y*
$X \rightarrow YZ$ *Transitivity*

  - Thus, $X \rightarrow YZ$ is satisfied in every relation where both $X \rightarrow Y$ and $X \rightarrow Y$ are satisfied
  - Therefore, we have derived the *union rule* for FDs: we can take the union of the RHSs of FDs that have the same LHS

15

Completeness

- Axioms are *complete:* If F entails f, then f can be derived from F using the axioms
- A consequence of completeness is the following (naïve) way of determining if F entails f:
  - Use the axioms in all possible ways to generate F^+ (the set of possible FD's is finite so this can be done) and see if f is in F^+

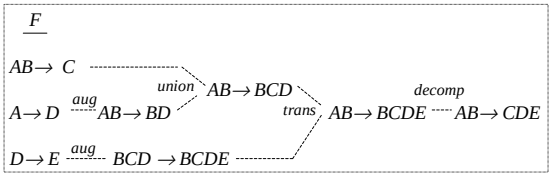
16

Correctness

- The notions of *soundness* and *completeness* link the syntax (Armstrong's axioms) with semantics (the definitions in terms of relational instances)
- This is the formal way of saying that the algorithm for entailment based on the axioms is "correct" with respect to the semantic (or w.r.t. the definitions)

17

Generating F^+



Thus, $AB \rightarrow BD$, $AB \rightarrow BCD$, $AB \rightarrow BCDE$, and $AB \rightarrow CDE$ are all elements of F^+

18