COSC 4820

Algebra and constraints

Kim Buckner

University of Wyoming

Jan. 25, 2023

Finish Chapter 2

Constraints

- Restrict data stored in the database
- *Referential integrity* constraint
- A value in one context must also appear in another
- example: Model number in **pc** must also appear as a model number in **product**.

Constraints (2)

- *Key* constraints
- Uniqueness of keys
- Then base constraints on those keys
- example: If two tuples of the **product** table have the same model value then they must be the same tuple.

Constraints (3)

- Use the algebra to express the constraints
- Add
  - $R = \emptyset$ means that expression R is empty
  - $R \subseteq S$ means that every tuple in R must also be in S
- example: $\textsc{select}$(model_R $=$ model_S AND maker_R $\neq$ maker_S)$\{$R,S$\}$ $= \emptyset$

Constraints (4)

- *Domain* constraints
- Limit the set of valid values for an attribute
- example: $\textsc{select}$(color $\neq$ 'true' AND color $\neq$ 'false'){laptop} $= \emptyset$

Chapter 3

Functional Dependencies

- Usually just FD.
- $A_1 A_2 \cdots A_n \longrightarrow B_1 B_2 \cdots B_n$
- If two tuples of a relation agree on values in some set of attributes then they must also agree on the values in another set of attributes.

more . . .

- The $\longrightarrow$ reads “functionally determine”
- Sets need not have an size greater than one (1).
- Adjacency is not required, the A's and B's can appear in any order in the relation.
- The FD must apply to all possible instances.

Keys

- $\{ A_1, A_2, \cdots, A_n \}$ is a key if
  - "These attributes *functionally determine* all other attributes of a relation."
  - In another way "No two distinct tuples can agree on $\{ A_1, A_2, \cdots, A_n \}$ (the key)."
  - (What does this mean?)

more . . .

- and if
  - “No proper subset of the key *functionally determines* all other attributes.”
  - In another way “The key must be minimal”.
  - (What does this really mean?).

More on keys

- “Key” vs “Primary Key”.
- Database engines make a differentiation.
- Helps in optimizing storage.
- BUT no difference in FD theory.

Superkey

- Set of attributes which contain a key.
- Remember what a key is.
- Superkey is not necessarily minimal.
- If key is say ”model” from the **pc** table.
- Superkey might be ”model,speed”.

FD Rules

- What FD's mean
- If $R(A, B, C)$ satisfies $A \rightarrow B$ and $B \rightarrow C$ then $A \rightarrow C$
- Two sets of FD are *equivalent* means
  - relation instances satisfying one are exactly the same as those satisfying the other

more . . .

- A set of FD's *follows* from another if
  - every relation satisfying the second
  - also satisfies the first
- If S and T are equivalent then
  - T follows from S and
  - S follows from T

Splitting/Combining Rule

- The single FD $A_1A_2\cdots A_n \to B_1B_2\cdots B_m$
- can be replaced with the set $A_1A_2\cdots A_n \to B_i, i = 1, 2\cdots m$
- This is the *splitting rule*

more . . .

- The set $A_1A_2\cdots A_n \to B_i, i = 1, 2\cdots m$
- can be replaced with the single FD $A_1A_2\cdots A_n \to B_1B_2\cdots B_m$
- This is the *combining rule*

more . . .

- No splitting rule for the left-hand side
- For example
  - From the **outcomes** table
  - ship,battle $\rightarrow$ result
  - ship $\not\rightarrow$ result
  - battle $\not\rightarrow$ result

Trivial FD

- A constraint is said to be *trivial* if it holds for every instance of the relation regardless what other constraints are assumed
- If the constraint is a FD then
  - The FD's $A_1A_2\cdots A_n \rightarrow B_1B_2\cdots B_m$ where
  - $\{B_1, B_2, \cdots, B_m\} \subset \{A_1, A_2, \cdots, A_n\}$
  - are trivial.

more . . .

- If some (but not all) of the attributes on right are on the left
- It is not trivial but can be simplified
- $A_1A_2\cdots A_n \rightarrow B_1B_2\cdots B_m$ is equivalent to $A_1A_2\cdots A_n \rightarrow C_1C_2\cdots C_k$ where the C's are all those B's not also A's
- This is the *trivial-dependency rule*