

## Functional Dependencies

R&G Chapter 19

Science is the knowledge of consequences, and dependence of one fact upon another.

Thomas Hobbes  
(1588-1679)



### Review: Database Design

- Requirements Analysis
  - user needs; what must database do?
- Conceptual Design
  - high level descr (often done w/ER model)
- Logical Design
  - translate ER into DBMS data model
- Schema Refinement
  - **consistency, normalization**
- Physical Design - indexes, disk layout
- Security Design - who accesses what



### The Evils of Redundancy

- **Redundancy** is at the root of several problems associated with **relational schemas**:
  - *redundant storage, insert/delete/update anomalies*
- Integrity constraints, in particular **functional dependencies**, can be used to identify schemas with such problems and to suggest refinements.
- Main refinement technique: **decomposition**
  - replacing ABCD with, say, AB and BCD, or ACD and ABD.
- **Decomposition should be used judiciously**:
  - Is there reason to decompose a relation?
  - What problems (if any) does the decomposition cause?



### Functional Dependencies (FDs)

- A functional dependency  $X \twoheadrightarrow Y$  holds over relation schema  $R$  if, for every allowable instance  $r$  of  $R$ :
$$t1 \twoheadrightarrow r, t2 \twoheadrightarrow r, \text{ if } \text{proj}_X(t1) = \text{proj}_X(t2) \text{ implies } \text{proj}_Y(t1) = \text{proj}_Y(t2)$$

(where  $t1$  and  $t2$  are tuples;  $X$  and  $Y$  are sets of attributes)
- In other words:  $X \twoheadrightarrow Y$  means
  - Given any two tuples in  $r$ , if the  $X$  values are the same, then the  $Y$  values must also be the same. (but not vice versa)
- Can read " $\twoheadrightarrow$ " as "determines"



### FD's Continued

- An FD is a statement about **all** allowable relations.
  - Must be identified based on semantics of application.
  - Given some instance  $r1$  of  $R$ , we can check if  $r1$  violates some FD  $f$ , but we cannot determine if  $f$  holds over  $R$ .
- **Question: How related to keys?**
- if " $K \twoheadrightarrow$  all attributes of  $R$ " then  $K$  is a **superkey** for  $R$ 
  - (does not require  $K$  to be *minimal*.)
- FDs are a generalization of keys.



### Example: Constraints on Entity Set

- Consider relation obtained from Hourly\_Emps:  
Hourly\_Emps ( $ssn, name, lot, rating, wage\_per\_hr, hrs\_per\_wk$ )
  - We sometimes denote a relation schema by listing the attributes: e.g., SNLRWH
  - This is really the *set* of attributes  $\{S, N, L, R, W, H\}$ .
  - Sometimes, we refer to the set of *all attributes* of a relation by using the relation name. e.g., "Hourly\_Emps" for SNLRWH
- What are some FDs on Hourly\_Emps?**
- $ssn$  is the key:  $S \twoheadrightarrow$  SNLRWH
  - $rating$  determines  $wage\_per\_hr$ :  $R \twoheadrightarrow W$
  - $lot$  determines  $lot$ :  $L \twoheadrightarrow L$  ("trivial" dependency)



## Problems Due to R ⊆ W

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

Hourly\_Emps

- **Update anomaly:** Can we modify W in only the 1st tuple of SNLRWH?
- **Insertion anomaly:** What if we want to insert an employee and don't know the hourly wage for his or her rating? (or we get it wrong?)
- **Deletion anomaly:** If we delete all employees with rating 5, we lose the information about the wage for rating 5!



## Detecting Reduncancy

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

Hourly\_Emps

Q: Why was R ⊆ W problematic, but S ⊆ W not?



## Decomposing a Relation

- Redundancy can be removed by "chopping" the relation into pieces.
- FD's are used to drive this process.

R ⊆ W is causing the problems, so decompose SNLRWH into what relations?

S	N	L	R	H
123-22-3666	Attishoo	48	8	40
231-31-5368	Smiley	22	8	30
131-24-3650	Smethurst	35	5	30
434-26-3751	Guldu	35	5	32
612-67-4134	Madayan	35	8	40

R	W
8	10
5	7

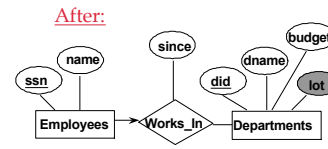
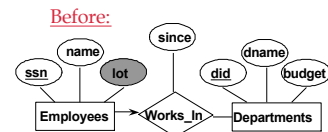
Wages

Hourly\_Emps2



## Refining an ER Diagram

- 1st diagram becomes:  
Workers(S,N,L,D,Si)  
Departments(D,M,B)  
– Lots associated with workers.
- Suppose all workers in a dept are assigned the same lot: D ⊆ L
- Redundancy; fixed by:  
Workers2(S,N,D,Si)  
Dept\_Lots(D,L)  
Departments(D,M,B)
- Can fine-tune this:  
Workers2(S,N,D,Si)  
Departments(D,M,B,L)



## Reasoning About FDs

- Given some FDs, we can usually infer additional FDs:  
 $title \sqsubseteq studio, star$  implies  $title \sqsubseteq studio$  and  $title \sqsubseteq star$   
 $title \sqsubseteq studio$  and  $title \sqsubseteq star$  implies  $title \sqsubseteq studio, star$   
 $title \sqsubseteq studio, studio \sqsubseteq star$  implies  $title \sqsubseteq star$

But,

$title, star \sqsubseteq studio$  does NOT necessarily imply that  $title \sqsubseteq studio$  or that  $star \sqsubseteq studio$

- An FD  $f$  is **implied by** a set of FDs  $F$  if  $f$  holds whenever all FDs in  $F$  hold.
- $F^+$  = **closure of  $F$**  is the set of all FDs that are implied by  $F$ . (includes "trivial dependencies")



## Rules of Inference

- **Armstrong's Axioms** ( $X, Y, Z$  are sets of attributes):
  - **Reflexivity:** If  $X \supseteq Y$ , then  $X \sqsubseteq Y$
  - **Augmentation:** If  $X \sqsubseteq Y$ , then  $XZ \sqsubseteq YZ$  for any  $Z$
  - **Transitivity:** If  $X \sqsubseteq Y$  and  $Y \sqsubseteq Z$ , then  $X \sqsubseteq Z$
- These are **sound** and **complete** inference rules for FDs!
  - i.e., using AA you can compute all the FDs in  $F^+$  and only these FDs.
- Some additional rules (that follow from AA):
  - **Union:** If  $X \sqsubseteq Y$  and  $X \sqsubseteq Z$ , then  $X \sqsubseteq YZ$
  - **Decomposition:** If  $X \sqsubseteq YZ$ , then  $X \sqsubseteq Y$  and  $X \sqsubseteq Z$



## Example

- **Contracts**(*cid,sid,jid,did,pid,qty,value*), and:
    - C is the key:  $C \twoheadrightarrow CSJDPQV$
    - Proj purchases each part using single contract:  $JP \twoheadrightarrow C$
    - Dept purchases at most 1 part from a supplier:  $SD \twoheadrightarrow P$
  - **Problem: Prove that SDJ is a key for Contracts**
  - $JP \twoheadrightarrow C, C \twoheadrightarrow CSJDPQV$  imply  $JP \twoheadrightarrow CSJDPQV$   
(by transitivity) (shows that JP is a key)
  - $SD \twoheadrightarrow P$  implies  $SDJ \twoheadrightarrow JP$  (by augmentation)
  - $SDJ \twoheadrightarrow JP, JP \twoheadrightarrow CSJDPQV$  imply  $SDJ \twoheadrightarrow CSJDPQV$   
(by transitivity) thus SDJ is a key.
- Q: can you now infer that SD  $\twoheadrightarrow$  CSDPQV (i.e., drop J on both sides)?**

**No! FD inference is not like arithmetic multiplication.**



## Attribute Closure

- **Computing the closure of a set of FDs can be expensive. (Size of closure is exponential in # attrs!)**
- **Typically, we just want to check if a given FD  $X \twoheadrightarrow Y$  is in the closure of a set of FDs  $F$ . An efficient check:**
  - Compute *attribute closure* of  $X$  (denoted  $X^+$ ) wrt  $F$ .  
 $X^+ =$  Set of all attributes  $A$  such that  $X \twoheadrightarrow A$  is in  $F^+$ 
    - $X^+ := X$
    - Repeat until no change: if there is an fd  $U \twoheadrightarrow V$  in  $F$  such that  $U$  is in  $X^+$ , then add  $V$  to  $X^+$
  - Check if  $Y$  is in  $X^+$
  - Approach can also be used to find the keys of a relation.
    - If all attributes of  $R$  are in the closure of  $X$  then  $X$  is a superkey for  $R$ .
    - Q: How to check if  $X$  is a "candidate key"?



## Attribute Closure (example)

- $R = \{A, B, C, D, E\}$
- $F = \{B \twoheadrightarrow CD, D \twoheadrightarrow E, B \twoheadrightarrow A, E \twoheadrightarrow C, AD \twoheadrightarrow B\}$
- **Is  $B \twoheadrightarrow E$  in  $F^+$ ?**
  - $B^+ = B$
  - $B^+ = BCD$
  - $B^+ = BCDA$
  - $B^+ = BCDAE$  ... Yes!  
and  $B$  is a key for  $R$  too!
- **Is  $D$  a key for  $R$ ?**
  - $D^+ = D$
  - $D^+ = DE$
  - $D^+ = DEC$
  - ... Nope!
- **Is  $AD$  a key for  $R$ ?**
  - $AD^+ = AD$
  - $AD^+ = ABD$  and  $B$  is a key, so Yes!
- **Is  $AD$  a candidate key for  $R$ ?**
  - $A^+ = A, D^+ = DEC$
  - ...  $A, D$  not keys, so Yes!
- **Is  $ADE$  a candidate key for  $R$ ?**
  - ... No!  $AD$  is a key, so  $ADE$  is a superkey, but not a cand. key



## Next Class...

- **Normal forms and normalization**
- **Table decompositions**