

# Database Management Systems

## Part III: Database Design

### Lecture 9

## Functional Dependencies

# Contents

- Basic Definitions
- Trivial and Nontrivial Dependencies
- Closure of a Set of Dependencies
- Closure of a Set of Attributes
- Irreducible Sets of Dependencies

# Basic Definitions

- A functional dependency is **a constraint between two sets of attributes** from the database.
- Basically, a functional dependence is a **many-to-one relationship** from one set of attributes to another within a given relation.
- Functional dependency determines **the relation of one attribute to another attribute** in a database management system (DBMS).

## Basic Definitions (Cont.)

- Functional dependency is represented by an arrow sign ( $\rightarrow$ ).
- For example, the functional dependency of X on Y is represented by  $X \rightarrow Y$ .
- The left-hand side of an FD is called the **determinant** and the right-hand side is called the **dependent**.
- The determinant and dependent are both **sets of attributes**.
- The **left-hand side** attributes **determine the values** of attributes **on the right-hand side**.

## Basic Definitions (Cont.)

- The concept of FD is first defined as it applies to **individual relation values** of a given relation at a given point in time.
- Let  $R$  be a relation and let  $X$  and  $Y$  be arbitrary subsets of the set of attributes of  $R$ .
- Then, we say that  $Y$  is functionally dependent on  $X$  ( $X \rightarrow Y$ ) if and only if each  $X$ -value in  $R$  has associated with it precisely one  $Y$ -value in  $R$ .
- In other words, whenever two tuples of  $R$  agree on their  $X$ -value, they also agree on their  $Y$ -value.

# Basic Definitions (Cont.)

- In the table, a relation SCP satisfies the FD

$$\{ S\# \} \rightarrow \{ CITY \}$$

because every SCP tuple with a given S# value also has the same CITY value.

S#	CITY	P#	QTY
S1	London	P1	100
S1	London	P2	100
S2	Paris	P1	200
S2	Paris	P2	200
S3	Paris	P2	300
S4	London	P2	400
S4	London	P4	400
S4	London	P5	400

- Indeed, it also satisfies several more FDs, the following among them:

$$\{ S\#, P\# \} \rightarrow \{ QTY \}$$

$$\{ S\#, P\# \} \rightarrow \{ CITY \}$$

$$\{ S\#, P\# \} \rightarrow \{ CITY, QTY \}$$

$$\{ S\#, P\# \} \rightarrow \{ S\# \}$$

$$\{ S\#, P\# \} \rightarrow \{ S\#, P\#, CITY, QTY \}$$

$$\{ S\# \} \rightarrow \{ QTY \}$$

$$\{ QTY \} \rightarrow \{ S\# \}$$

## Basic Definitions (Cont.)

- Then, the concept of FD is defined as it applies to **all possible values** of a given relation.
- Let R be a relation variable and let X and Y be arbitrary subsets of the set of attributes of R.
- Then, we say that Y is functionally dependent on X ( $X \rightarrow Y$ ) if and only if, **in every possible legal value of R**, each X-value has associated with it precisely one Y-value.
- In other words, **in every possible legal value of R**, whenever two tuples of R agree on their X-value, they also agree on their Y-value.

## Basic Definitions (Cont.)

- Here are some time-independent FDs that apply to the relation variable SCP:

$\{ S\#, P\# \} \rightarrow QTY$

$\{ S\#, P\# \} \rightarrow CITY$

$\{ S\#, P\# \} \rightarrow \{ CITY, QTY \}$

$\{ S\#, P\# \} \rightarrow S\#$

$\{ S\#, P\# \} \rightarrow \{ S\#, P\#, CITY, QTY \}$

$\{ S\# \} \rightarrow CITY$

- Notice that the following FDs **do not hold for all time**:

$S\# \rightarrow QTY$        $QTY \rightarrow S\#$

# Trivial and Nontrivial Dependencies

- The FD is **trivial** if and only if the right-hand side **is a subset** of the left-hand side.
- For example, the FD  $\{ S\#, P\# \} \rightarrow \{ S\# \}$  is **trivial**.
- One obvious way to reduce the size of the set of FDs is **to eliminate the trivial dependencies**.
- Nontrivial dependencies are the ones that correspond to **genuine integrity constraints**.
- Nontrivial dependency occurs when  $A \rightarrow B$  holds true where attribute B **is not a subset** of attribute A.

# Closure of a Set of Dependencies

- The set of all FDs that are implied by a given set  $S$  of FDs is called **the closure of  $S$** , and is denoted  $S^+$ .
- To compute  $S^+$  from  $S$ , a set of rules of inference by which new FDs can be inferred from given ones can be stated in a variety of equivalent ways.
- One of the simplest rule is **Armstrong's inference rules**.

# Closure of a Set of Dependencies (Cont.)

## Armstrong's inference rules

- Let A, B, and C be arbitrary subsets of the set of attributes of the given relation R, and AB means the union of A and B. Then,
  1. **Reflexivity**: If B is a subset of A, then  $A \rightarrow B$ .
  2. **Augmentation**: If  $A \rightarrow B$ , then  $AC \rightarrow BC$ .
  3. **Transitivity**: If  $A \rightarrow B$  and  $B \rightarrow C$ , then,  $A \rightarrow C$ .

# Closure of a Set of Dependencies (Cont.)

## Armstrong's inference rules

- These additional rules derived from the first three can be used to simplify the practical task of computing from S.
4. **Self-determination**:  $A \rightarrow A$ .
  5. **Decomposition**: If  $A \rightarrow BC$ , then  $A \rightarrow B$  and  $A \rightarrow C$ .
  6. **Union**: If  $A \rightarrow B$  and  $A \rightarrow C$ , then,  $A \rightarrow BC$ .
  7. **Composition**: If  $A \rightarrow B$  and  $C \rightarrow D$ , then  $AC \rightarrow BD$ .

# Closure of a Set of Dependencies (Cont.)

## General Unification Theorem

▪ Darwen proves the following rule.

8. If  $A \rightarrow B$  and  $C \rightarrow D$ , then  $A \cup (C - B) \rightarrow BD$  (where “ $\cup$ ” is union and “ $-$ ” is set difference).

# Closure of a Set of Dependencies (Cont.)

## Example

- Suppose we are given relation R with attributes A, B, C, D, E, F, and the FDs

$$A \rightarrow BC$$

$$B \rightarrow E$$

$$CD \rightarrow EF$$

- Show that the FD  $AD \rightarrow F$  holds in R and it is a member of the closure of the given set.

# Closure of a Set of Dependencies (Cont.)

## Example

■ The solution is as follows.

1.  $A \rightarrow BC$  (given)
2.  $A \rightarrow C$  (1, decomposition)
3.  $AD \rightarrow CD$  (2, augmentation)
4.  $CD \rightarrow EF$  (given)
5.  $AD \rightarrow EF$  (3 and 4, transitivity)
6.  **$AD \rightarrow F$**  (5, decomposition)

# Closure of a Set of Attributes

- An effective way of determining whether a given FD is in that closure is **the notion of a superkey**.
- A superkey for a relation  $R$  is a set of attributes of  $R$  that **includes at least one candidate key** of  $R$  as a subset  $K$ .
- To determine whether  $K$  is a superkey, we need to determine whether the set of all attributes functionally dependent on  $K$  is the set of all attributes of  $R$ .

## Closure of a Set of Attributes (Cont.)

- Given a set  $S$  of FDs that hold in  $R$ , we need a way of determining the set of all attributes of  $R$  that are functionally dependent on  $K$  called **closure  $K^+$  of  $K$  under  $S$** .
- A simple algorithm for computing this closure is given in figure.

```
CLOSURE[K,S] := K ;
do "forever" ;
  for each FD  $X \rightarrow Y$  in  $S$ 
    do ;
      if  $X$  is a subset of CLOSURE[K,S]
        then CLOSURE[K,S] := CLOSURE[K,S] UNION  $Y$  ;
      end ;
    if CLOSURE[K,S] did not change on this iteration
      then /* computation complete */ leave loop ;
  end ;
```

# Closure of a Set of Attributes (Cont.)

## Example

- Suppose we are given relation R with attributes A, B, C, D, E, F, and FDs

$$A \rightarrow BC$$

$$E \rightarrow CF$$

$$B \rightarrow E$$

$$CD \rightarrow EF$$

- Compute the closure  $\{A,B\}^+$  of the set of attributes  $\{A,B\}$  under this set of FDs.

# Closure of a Set of Attributes (Cont.)

## Example

- The solution by using the algorithm is as follows.
  1. Initialize the result  $\text{CLOSURE}[K,S]$  to  $\{A,B\}$ .
  2. Round the inner loop four times, once for each of the given FDs. On the first iteration (for the FD  $A \rightarrow BC$ ), the left-hand side is indeed a subset of  $\text{CLOSURE}[K,S]$  as computed so far, so we add attributes (B and) C to the result.  $\text{CLOSURE}[K,S]$  is now the set  $\{A,B,C\}$ .

# Closure of a Set of Attributes (Cont.)

## Example

3. On the second iteration (for the FD  $E \rightarrow CF$ ), the left-hand side is not a subset of the result as computed so far, which thus remains unchanged.
4. On the third iteration (for the FD  $B \rightarrow E$ ), we add  $E$  to  $\text{CLOSURE}[K,S]$ , which now has the value  $\{A,B,C,E\}$ .
5. On the fourth iteration (for the FD  $CD \rightarrow EF$ ),  $\text{CLOSURE}[K,S]$  remains unchanged.

# Closure of a Set of Attributes (Cont.)

## Example

6. Round the inner loop four times again. On the first iteration, the result does not change; on the second, it expands to  $\{A,B,C,E,F\}$ ; on the third and fourth, it does not change.
  7. Round the inner loop four times again.  $CLOSURE[K,S]$  does not change, and so the whole process terminates, with  $\{A,B\}^+ = \{A,B,C,E,F\}$ .
- Therefore,  $\{A,B\}^+ = \{A,B,C,E,F\}$ .

# Irreducible Sets of Dependencies

- A set  $S$  of FDs is **irreducible** if and only if it **satisfies** the following **three properties**:
  1. The **right-hand side** of every FD in  $S$  involves **just one attribute**.
  2. The **left-hand side** of every FD in  $S$  is **irreducible**.
  3. **No FD in  $S$  can be discarded** from  $S$  without changing the closure  $S^+$ .

# Irreducible Sets of Dependencies (Cont.)

- For example, consider the parts relation P which holds the following FDs:

$P\# \rightarrow PNAME$

$P\# \rightarrow WEIGHT$

$P\# \rightarrow COLOR$

$P\# \rightarrow CITY$

P	P#	PNAME	COLOR	WEIGHT	CITY
	P1	Nut	Red	12	London
	P2	Bolt	Green	17	Paris
	P3	Screw	Blue	17	Rome
	P4	Screw	Red	14	London
	P5	Cam	Blue	12	Paris
	P6	Cog	Red	19	London

- This set of FDs is easily **seen to be irreducible**.
- The right-hand side is a single attribute in each case, the left-hand side is obviously irreducible in turn, and none of the FDs can be discarded without changing the closure.

# Irreducible Sets of Dependencies (Cont.)

- By contrast, the following sets of FDs are not irreducible:

$P\# \rightarrow \{ PNAME, COLOR \}$  : The right-hand side of this FD is not

$P\# \rightarrow WEIGHT$  a singleton set

$P\# \rightarrow CITY$

$\{ P\#, PNAME \} \rightarrow COLOR$  : This FD can be simplified by

$P\# \rightarrow PNAME$  dropping PNAME from the left-hand

$P\# \rightarrow WEIGHT$  side without changing the closure

$P\# \rightarrow CITY$  (i.e., it is not left-irreducible)

# Irreducible Sets of Dependencies (Cont.)

- The following sets of FDs are also not irreducible:

$P\# \rightarrow P\#$  : This FD can be discarded without

$P\# \rightarrow PNAME$  changing the closure

$P\# \rightarrow COLOR$

$P\# \rightarrow WEIGHT$

$P\# \rightarrow CITY$

# Summary

- Functional Dependency is when **one attribute determines another attribute** in a DBMS system.
- The FD is trivial if and only if the right-hand side (the dependent) **is a subset** of the left-hand side (the determinant).
- The FD is nontrivial if and only if the right-hand side **is not a subset** of the left-hand side.
- The set of FDs is **irreducible** if the right-hand side involves just one attribute, the left-hand side is irreducible and none of the FDs can be discarded without changing the closure.

# Next Lecture

## Part III: Database Design

### Normalization

- 1NF
- 2NF
- 3NF
- BCNF

# Textbook and References

## Textbook

- C. J. Date, “An Introduction to Database Systems”, 6th Edition, 1994.

## Additional References

- Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database System Concepts”, 6th Edition, 2011.
- Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, 6th Edition, 2010.
- <https://www.guru99.com>
- [https:// www.tutorialspoint.com](https://www.tutorialspoint.com)