# An Introduction to Database System

## Part III Database Design



spark@dblab.sogang.ac.kr

# Database Design

- chapter 11. Functional Dependencies
- chapter 12. Further Normalization I
- chapter 13. Further Normalization II
- chapter 14. Semantic Modeling



spark@dblab.sogang.ac.kr

# Database Design

- "Given some body of data to be represented in a database, how do we decide on a suitable logical structure for that data ?"
  - how do we decide what base relations should exist and what attributes those relations should have?
  - relational database design

(a) logical design only

  - physical design can be treated as a separate, follow-on activity
  - physical design : DBMS-specific
  - logical design : DBMS-independent
  - getting the logical design right first

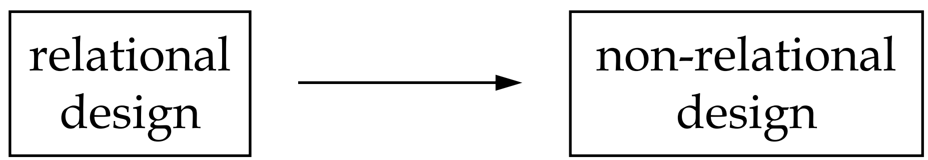


spark@dblab.sogang.ac.kr

# Database Design

**(b) relational design**

- **relevant to the design of non-relational database**




**(c) database design**

- **still very much of an art, not a science**
- **some specific principles(chapter 11 - 14)**

# Database Design

## (d) a couple of assumptions

- Database design is not just a question of getting the data structures right - data integrity is a key ingredient also
- abstract logical design
  - *application-independent* design
  - design that will be *robust*
  - to design the *conceptual* schema

## (e) what domains or types should exist

- we will have little to say on this topic



spark@dblab.sogang.ac.kr

# An Introduction to Database Systems

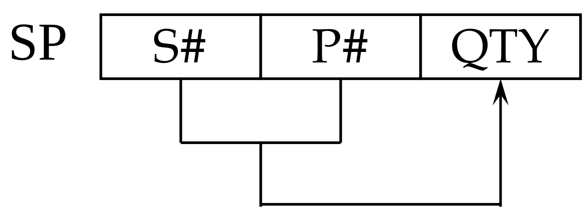
## chapter 11. Functional Dependencies



spark@dblab.sogang.ac.kr

# 11.1 Introduction

- the concept of functional dependence (FD)
  - a many-to-one relationship from one set of attributes to another within a given relation
  - satisfied all times



# 11.2 Basic Definitions

SCP

| S# | CITY | P# | QTY |
|---|---|---|---|
| S1 | London | P1 | 100 |
| S1 | London | P2 | 100 |
| | | | |

{S#, P#} → {S#}
{S#, P#} → {P#}
{S#, P#} → {QTY}
{S#, P#} → {CITY}
.
.
.
{S#} → {CITY}

- **Let R be a relation variable, and let X and Y be subset of the set of attributes of R. Then we say that Y is functionally dependent (FD) on X iff, in every possible legal value of R, each X-value has associated with it precisely one Y-value**

**X → Y** *functionally determine*

- **time-independent meaning**
- **an integrity constraint for SCP**

**S#→ CITY**
**CONSTRAINT S# _CITY_FD**
**COUNT ( SCP {S#} ) = COUNT (SCP {S#, CITY} ) ;**



spark@dblab.sogang.ac.kr

# 11.2 Basic Definition

- **If X is a candidate key of relation R - if it is the primary key, then all attributes of Y of R must necessarily be functionally dependent on X.**
  - **definition of candidate key**
  - **example**

    **P# $\rightarrow$ { P#, PNAME, COLOR, WEIGHT, CITY }**

- **If R satisfies FD A $\rightarrow$ B and A is not a candidate key, then R will involve some *redundancy***
  - **SCP : S# $\rightarrow$ CITY**
- **Complete set of FDs**



spark@dblab.sogang.ac.kr

# 11.2 Basic Definition

- Objective : The set of FDs satisfied by all legal values of R : very large $\rightarrow$ reducing that set to a manageable size

- Why is this objective desirable ?

  - FDs represent integrity constraints, and hence DBMS needs to check them when updates are performed.

  - Given a set S of FDs, it is desirable to find some other T that is much smaller than S and has the property that every FD in S is implied by the FDs in T.

  - If such a set T can be found, it is sufficient that the DBMS enforce the FDs in T, and the FDs in S will then be enforced automatically



spark@dblab.sogang.ac.kr

# 11.3 Trivial and Nontrivial Dependencies

- **One obvious way to reduce the size of the set of FDs is to eliminate the trivial dependencies**

- **An FD is *trivial* if and only if right-hand side is a subset (not necessarily proper subset) of the left-hand side.**
  - **example**

    **SCP : {S#, P#} $\rightarrow$ S#**



spark@dblab.sogang.ac.kr

# 11.4 Closure of a Set of Dependencies

- Simple example of FD
  - $\{S\#, P\#\} \rightarrow \{CITY, QTY\}$
  - implies both the following FDs :

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

    $\{S\#, P\#\} \rightarrow QTY$
- more complex example of FD
  - relation R with three attributes A,B, and C : R(A,B,C)
  - functional dependencies : $A \rightarrow B$, $B \rightarrow C$
  - it is easy to see that the FD

    $A \rightarrow C$



spark@dblab.sogang.ac.kr

# 11.4 Closure of a Set of Dependencies

- **Closure of S ( $S^+$ )**
  - **the set of all FDs that are implied by a set S of FDs is called the closure of S ($S^+$)**
- **Armstrong's axioms( rules of inference )**
  - **$S \rightarrow S^+$**
  - **a way of computing $S^+$ from S**
- **Armstrong's inference rules**
  - **Let A,B and C be subset of the set of attributes of R**

    **(1)Reflexivity:If B is a subset of A, them $A \rightarrow B$**

    **(2)Augmentation:If $A \rightarrow B$, then $AC \rightarrow BC$**

    **(3)Transitivity:If $A \rightarrow B$ and $B \rightarrow C$, then $A \rightarrow C$**



spark@dblab.sogang.ac.kr

# 11.4 Closure of a Set of Dependencies

- The rules are ***complete***, in the sense that, given a set of FDs, all FDs implied by S and be derived from s using the rules.
- The rules are ***sound***, in the sense that no additional FDs
- (i.e., FDs not implied by S) can be so derived.
  - The rules can be used to derive precisely the closure $S^+$

- **additional rules**

  **(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**



spark@dblab.sogang.ac.kr

# 11.4 Closure of a Set of Dependencies

- **General Unification Theorem ( Darwen )**
  - **(8) If $A \rightarrow B$ and $C \rightarrow D$, then $A \cup (C-B) \rightarrow BD$**
    - **(where "$\cup$" is union and "-" is set difference)**
- **example**

**R -** ◆ **FD : $A \rightarrow BC$, $B \rightarrow E$, $CD \rightarrow EF$**

$S \rightarrow S^{+} \ni AD \rightarrow F$ ?

◆ **FD : AD F hold in R? (yes)**

| | |
|---|---|
| 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 |



spark@dblab.sogang.ac.kr

# 11.5 Closure of a Set of Attributes

- **Compute the closure $S^+$ of a given set S of FDs**
- **How to compute a certain subset of the closure**

  (that subset consisting of all FDs with a set Z as the left-hand side)

- **Closure $Z^+$ of Z under S**
  - **Given a relation R, a set Z of attributes in R, a set of S of FDs that holds for R, we can determine the set of all attributes of R that are functionally dependent of Z**



spark@dblab.sogang.ac.kr

# 11.5 Closure of a Set of Attributes

R : S
↓
CK

- **definition of superkey**

S: a set of FDs
↓
$S^+$

Z: a set of attr.
under S
↓
$Z^+$

- **A superkey for R is a set of attributes of R that includes at least one candidate key of R as a subset - not necessarily a proper subset.**

- **To determine whether Z is a superkey, we need to determine whether the set of all attributes functionally dependent on Z is the set of all attributes of R.**

⇩

- **Given a set S of FDs, a way of determining the set of all attributes of R that are functionally dependent on Z(closure $Z^+$ of Z under S)**

⇩



**spark@dblab.sogang.ac.kr**

# 11.5 Closure of a Set of Attributes

- **Algorithm for Computing the closure $Z^+$ of Z under S**

```
CLOSURE [Z, S] := Z;
do "forever";
        for each FD X → Y in S
            do ;
                    if  X  is a subset of CLOSURE [Z, S]
                    then CLOSURE [Z, S] : = CLOSURE [Z, S]  ∪ Y;
            end ;
           if CLOSURE [Z, S] did not change on this iteration
           then  leave loop ;
end ;
```



spark@dblab.sogang.ac.kr

# 11.5 Closure of a Set of Attributes

- example
  - Relation R with attributes A,B,C,D,E and F : R(A,B,C,D,E,F)

$S \rightarrow$

  - FD : $A \rightarrow BC$, $E \rightarrow CF$, $B \rightarrow E$, $CD \rightarrow EF$
  - compute the $(AB)^+$ = ?
  - First iteration

    $AB \cup BC$

    $A \rightarrow BC$ : $(AB)^+ = \{A,B,C\}$

    $E \rightarrow CF$ : $(AB)^+$ : unchanged

    $B \rightarrow E$ : $(AB)^+ = \{A,B,C,E\}$

    $CD \rightarrow EF$ : unchanged
  - second iteration

    $E \rightarrow CF$ : $(AB)^+ = \{A,B,C,E,F\}$
  - closure of {AB}={A,B,C,E,F} is not a superkey



spark@dblab.sogang.ac.kr

# 11.5 Closure of a Set of Attributes

- **a FD $X \rightarrow Y$ hold if and only if Y is a subset of $X^+$ of X under S.**
- **Given a set S of FDs, we can easily tell whether a specific FD $X \rightarrow Y$ follows from S**
  **( a simple way of determining whether a FD $X \rightarrow Y$ is in the closure $S^+$ of S)**

$S \ni X \rightarrow Y$ ? ⇨ $Y \leq X^+$ ?



spark@dblab.sogang.ac.kr

# 11.6 Irreducible Sets of Dependencies

- **Cover**
  - **Let S1 and S2 be two sets of FDs. If every FD implied by S1 is implied by the FDs in S2 (i.e., if $S1^+$ is a subset of $S2^+$), we say S2 is a cover for S1**
    - **If the DBMS enforces the constraints represented by the FDs in S2, then it will automatically be enforcing the FDs in S1.**
- **equivalent**
  - **If S2 is a cover for S1and S1 is a cover for S2($S1^+=S2^+$), we say S1 and S2 are equivalent.**
    - **If the DBMS enforces the containts represented by the FDs in S2, it will automatically be enforcing the FDs in S1 and vice versa.**



spark@dblab.sogang.ac.kr

# 11.6 Irreducible Sets of Dependencies

- **Irreducible**

$X \rightarrow Y$

- **A set S of FDs are irreducible(minimal) if and only if it satisfies the following three conditions**

**1) the right-hand side(the dependent) of every FD in S involves just one attribute(singleton set)**

**2) the left-hand side(the determinant) of every FD in S is irreducible(no attribute can be discarded from the determinant without changing the closure $S^+$). Such a FD is left-irreducible.**

**3) No FD in S can be discarded from S without changing the closure $S^+$**



spark@dblab.sogang.ac.kr

# 11.6 Irreducible Sets of Dependencies

- **example**
  - **relation P : P(P#, PNAME, COLOR, WEIGHT, CITY)**
  - **irreducible**

    **FD : P# → PNAME, P# → CLOLR, P# → WEIGHT, P#→ CITY**
  - **reducible**

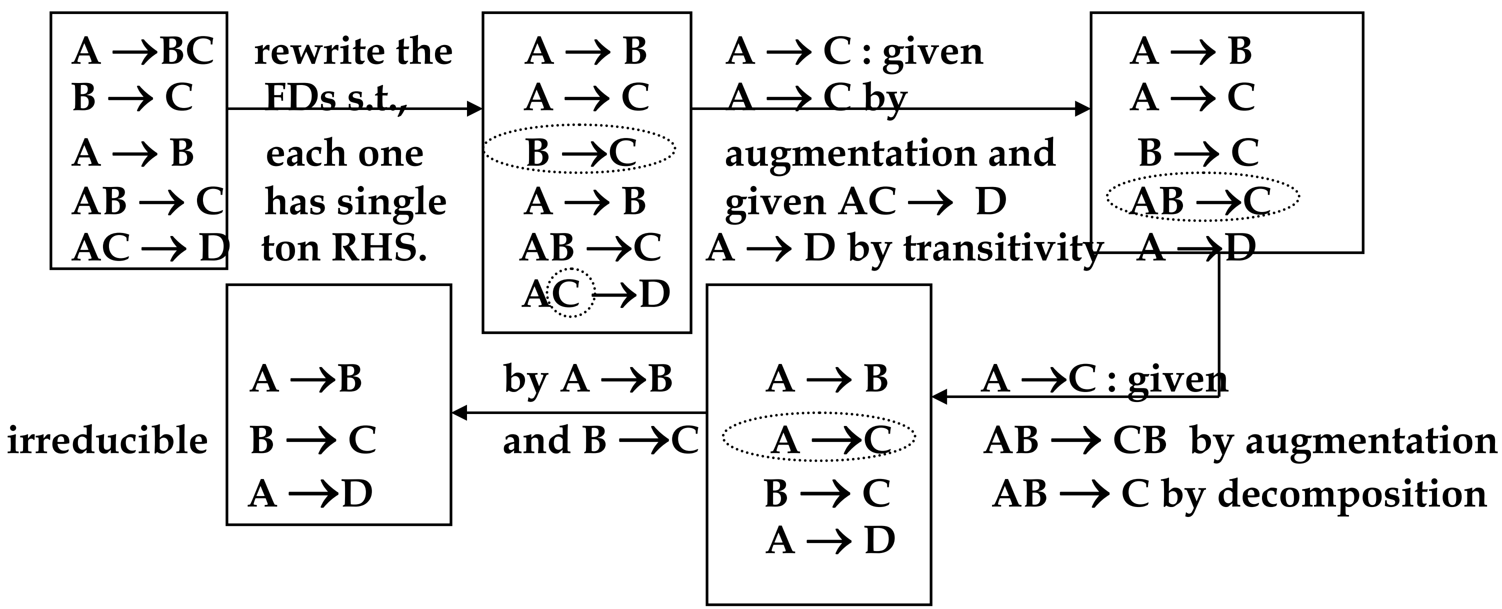
| | | |
|---|---|---|
| 1)<br>P# → {PNAME,COLOR}<br>P# → WEIGHT<br>P# → CITY | 2)<br>{P#,PNAME} → COLOR<br>P# → PNAME<br>P# → WEIGHT<br>P# → CITY | 3)<br>P# → P#<br>P# → PNAME<br>P# → WEIGHT<br>P# → CITY |



**spark@dblab.sogang.ac.kr**

# 11.6 Irreducible Sets of Dependencies

- For every set of FDs, there exists at least one equivalent set that is irreducible
- Example
  - relation R : R(A,B,C,D)
  - FD : { A $\rightarrow$BC, B $\rightarrow$ C, A $\rightarrow$ B, AB $\rightarrow$ C, AC $\rightarrow$ D }



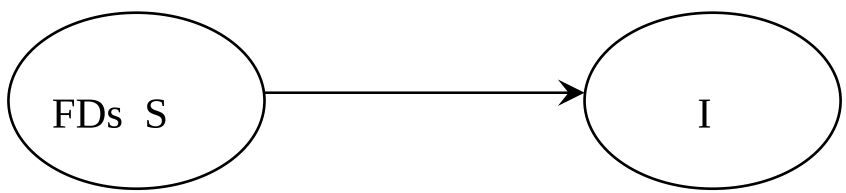



spark@dblab.sogang.ac.kr

# 11.6 Irreducible Sets of Dependencies

- A Set I of FDs that is irreducible and is equivalent to some other set S of FDs is said to be an irreducible cover of S

**(minimal cover)**




not unique



spark@dblab.sogang.ac.kr