Lezione 20 – Algoritmo di decomposizione – Esercizi

Prof.ssa Maria De Marsico
demarsico@di.uniroma1.it

Parte del materiale è tratto dal materiale preparato dalla prof.ssa Moscarini per il canale A-L

Introduzione

- In questa lezione mostreremo degli esempi in cui dato uno schema di relazione R e un insieme di dipendenze funzionali F su R troviamo una decomposizione $\rho=\{R_1, R_2,\ldots, R_k\}$ di R tale che:
  - per ogni i, $i=1,\ldots,k$, R_i è in 3NF
  - ρ preserva F
  - ρ ha un join senza perdita

Algoritmo per la decomposizione di uno schema

SAPIENZA
Università di Roma
Dipartimento di Informatica

Algoritmo – decomposizione di uno schema di relazione

Input uno schema di relazione R e un insieme F di dipendenze funzionali su R, che è una **copertura minimale;**

Output una decomposizione ρ di R che preserva F e tale che per ogni schema di relazione in ρ è in 3NF;

begin

$S:=\varnothing$;

for every $A \in R$ tale che <u>A non è coinvolto in nessuna dipendenza funzionale in</u> F **do**

$S:=S \cup \{A\}$;

if $S \neq \varnothing$ **then**

begin

$R:=R-S$;

$\rho:=\rho \cup \{S\}$

end

if esiste una dipendenza funzionale in F <u>che coinvolge tutti gli attributi in R</u>

then $\rho:=\rho \cup \{R\}$

else for every $X \rightarrow A \in F$ **do** $\rho:=\rho \cup \{XA\}$

end

R residuo **dopo** aver eventualmente **eliminato** gli **attributi** inseriti prima in **S**

in questo caso **ci fermiamo** anche se la copertura minimale contiene **anche altre dipendenze; in altre parole la copertura minimale potrebbe contenere anche altre dipendenze**

Completiamo

- Sia R uno schema di relazione, F un insieme di dipendenze funzionali su R, che è una copertura minimale e ρ la decomposizione di R prodotta dall'Algoritmo di decomposizione. La decomposizione $\sigma = \rho \cup \{K\}$, dove K è una chiave per R, è tale che:
- ogni schema di relazione in σ è in 3NF
- σ preserva F
- σ ha un join senza perdita.

Esempio 1

Sapienza Università di Roma
Dipartimento di Informatica

Dato il seguente schema di relazione

$$R = (A, B, C, D, E, H)$$

e il seguente insieme di dipendenze funzionali

$$F = \{ AB \rightarrow CD, C \rightarrow E, AB \rightarrow E, ABC \rightarrow D \}$$

- Verificare che ABH è una chiave per R.
- Sapendo che ABH è l'unica chiave per R, verificare che R non è in 3NF.
- Trovare una copertura minimale G di F.
- Trovare una decomposizione ρ di R tale che preserva G e ogni schema in ρ è in 3NF
- Trovare una decomposizione σ di R tale che preserva G, ha un join senza perdita e ogni schema in σ è in 3NF.

Esempio 1: chiave

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = (A, B, C, D, E, H)

F = { AB $\rightarrow$ CD, C $\rightarrow$ E, AB $\rightarrow$ E, ABC $\rightarrow$ D }

Verificare che ABH è una chiave significa verificare due condizioni

- ABH deve **determinare funzionalmente l'intero** schema
- **Nessun sottoinsieme di ABH deve determinare funzionalmente l'intero schema**

- Per verificare la prima condizione si controlla se la chiusura dell'insieme di attributi ABH contiene tutti gli attributi dello schema. Applicando l'algoritmo per il calcolo della chiusura di un insieme di attributi, al primo passo inseriamo nella chiusura di ABH gli attributi C, D ed E grazie alle dipendenze funzionali in cui la parte sinistra e' contenuta in ABH, cioè AB $\rightarrow$ CD e AB $\rightarrow$ E, quindi possiamo anche fermarci perché abbiamo inserito tutti gli attributi dello schema, cioè abbiamo verificato $(ABH)^+$ = {A,B,C,D,E,H}.

Esempio 1: chiave

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = (A, B, C, D, E, H)

F = { $AB \rightarrow CD$, $C \rightarrow E$, $AB \rightarrow E$, $ABC \rightarrow D$ }

Verificare che ABH è una chiave significa verificare due condizioni

- ABH deve **determinare funzionalmente l'intero** schema
- **Nessun sottoinsieme di ABH deve determinare funzionalmente l'intero schema**

- Per verificare la seconda condizione, dobbiamo controllare **che la chiusura di nessun sottoinsieme di ABH** contenga tutti gli attributi dello schema. A questo proposito notiamo che **H deve comparire in ogni caso in una chiave** dello schema, perché non viene determinato da altri attributi. Quindi possiamo evitare di calcolare le chiusure degli attributi singoli A e B che non potrebbero determinare H neppure per transitività. Inoltre H non compare mai a sinistra delle dipendenze, quindi da solo non determina alcun attributo. Restano da controllare le chiusure di AH e di BH, ma è banale verificare che in entrambi i casi l'algoritmo terminerebbe subito perché non ci sono dipendenze con a sinistra sottoinsiemi di AH o di BH.
- Possiamo quindi concludere che **ABH è chiave dello schema dato**.

Esempio 1: chiave

$R = (A, B, C, D, E, H)$ $F = \{ AB \rightarrow CD, C \rightarrow E, AB \rightarrow E, ABC \rightarrow D \}$

- Per verificare che lo schema non è in terza forma normale, basta osservare la presenza delle dipendenze parziali $AB \rightarrow CD$ e $AB \rightarrow E$.

Esempio 1: copertura minimale di F

$F = \{ AB \rightarrow CD, C \rightarrow E, AB \rightarrow E, ABC \rightarrow D \}$

- Per trovare la copertura minimale, prima di tutto riduciamo le parti destre a singleton

  $F = \{AB \rightarrow C, AB \rightarrow D, C \rightarrow E, AB \rightarrow E, ABC \rightarrow D\}$

- Passo 2. Ora dobbiamo verificare se nelle dipendenze ci sono ridondanze nelle parti sinistre. Cominciamo dalla dipendenza $AB \rightarrow C$ e controlliamo se $A \rightarrow C$ appartiene a F^+, cioè se C appartiene a $(A)^+_F$. Provando ad applicare l'algoritmo non potremmo inserire nessun attributo nella chiusura di A, in quanto non ci sono dipendenze che abbiano nella parte sinistra il solo attributo A, quindi $(A)^+_F = \{A\}$. Lo stesso si verifica per B, cioè $(B)^+_F = \{B\}$, quindi la parte sinistra della dipendenza non può essere ridotta. Lo stesso si verifica per le dipendenze AB® D e $AB \rightarrow E$. Proviamo allora a ridurre $ABC \rightarrow D$. Poiché nell'insieme di dipendenze esiste $AB \rightarrow D$, possiamo non solo eliminare l'attributo C ma anche tutta la dipendenza risultante che è un duplicato. Alla fine di questo passo abbiamo un insieme $G = \{ AB \rightarrow C, AB \rightarrow D, C \rightarrow E, AB \rightarrow E\}$ di dipendenze equivalente ad F.

Esempio 1: copertura minimale di F

SAPIENZA
Università di Roma
Dipartimento di Informatica

$G = \{AB \to C, AB \to D, C \to E, AB \to E\}$

- Passo 3. Vediamo ora se questo insieme contiene delle dipendenze ridondanti. Intanto possiamo considerare che C viene determinato unicamente da AB, quindi eliminando la dipendenza $AB \to C$ non riusciremmo più ad inserirlo nella chiusura di AB rispetto al nuovo insieme di dipendenze. Lo stesso vale per D. Proviamo allora ad eliminare la dipendenza $C \to E$. Rispetto al nuovo insieme di dipendenze $G = \{ AB \to C, AB \to D, AB \to E\}$ abbiamo che $(C)^+_G = \{C\}$ in cui non compare E. La dipendenza dunque deve rimanere. Proviamo infine ad eliminare $AB \to E$. Rispetto a $G = \{ AB \to C, AB \to D, C \to E\}$ abbiamo che $(AB)^+_G = \{A,B,C,D,E\}$ in cui E viene aggiunto al secondo passo dell'algoritmo per il calcolo della chiusura. Ciò significa che E rientra comunque nella chiusura di AB perché la dipendenza $AB \to E$, pur non comparendo in G, si trova in G^+, e quindi può essere eliminata. La copertura minimale di F sarà allora

$G = \{AB \to C, AB \to D, C \to E\}$

Esempio 1: decomposizione

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

$R = (A, B, C, D, E, H)$ $F = \{AB \rightarrow C, AB \rightarrow D, C \rightarrow E\}$

- Applichiamo l'algoritmo per la decomposizione dello schema di relazione che non è 3NF dato l' insieme di dipendenze G che è una copertura minimale.
- Al primo passo dobbiamo inserire in un elemento della decomposizione gli attributi che non compaiono nelle dipendenze di G. È il caso dell'attributo H, quindi inizialmente avremo $\rho = \{H\}$ e lo schema R diventa (A,B,C,D,E). Passiamo poi a verificare che **non ci sono** dipendenze che coinvolgono tutti gli attributi dello schema, per cui eseguiamo il passo alternativo. Abbiamo alla fine $\rho = \{H, ABC, ABD, CE\}$
- per avere una decomposizione con join senza perdita, aggiungiamo alla decomposizione precedente un sottoschema che contenga la chiave ABH (che non è già contenuta in alcuno degli schemi ottenuti). Avremo quindi

$$\sigma = \{\{H\}, \{A,B,C\}, \{A,B,D\}, \{C,E\}, \{A,B,H\}\}$$

che si può scrivere anche

$$\sigma = \{H, ABC, ABD, CE, ABH\}$$

Esempio 2

Dato il seguente schema di relazione

$$R = (A, B, C, D, E)$$

e il seguente insieme di dipendenze funzionali

$$F = \{ AB \rightarrow C, B \rightarrow D, D \rightarrow C \}$$

- Verificare che R non è in 3NF.
- Fornire una decomposizione di R tale che:
  - ogni schema della decomposizione è in 3NF,
  - la decomposizione preserva F,
  - la decomposizione ha un join senza perdite.

Esempio 2: chiave

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = (A, B, C, D, E)

F = { AB $\rightarrow$ C, B $\rightarrow$ D, D $\rightarrow$ C }

- Per verificare se lo schema è 3NF dobbiamo prima di tutto identificare la/le chiavi
- Notiamo che **l'attributo E deve far parte della chiave**, perché non compare nelle dipendenze e quindi può essere determinato funzionalmente solo per riflessività.
- **C non compare mai** a sinistra (non determina nessun altro attributo) né da solo ne con altri attributi, quindi **non farà parte** della chiave
- Calcoliamo $(AB)^+_F$ = {A,B,C,D} . Manca solo la E e infatti $(ABE)^+_F$ = {A,B,C,D, E} =R
  - Vediamo se ABE è minimale: $(AE)^+_F$ = {A, E} , $(BE)^+_F$ = {B,C,D, E} → OK ABE è minimale
- Calcoliamo le chiusure delle altre parti sinistre per valutare se sarebbe possibile trovare un'altra chiave . Inutile provare con B … E deve esserci e abbiamo già provato BE. Proviamo DE: $(DE)^+_F$ = {D, E, C}
- ABE è l'unica chiave
- In F ci sono le dipendenze parziali AB $\rightarrow$ C, B $\rightarrow$ D e la dipendenza transitiva D $\rightarrow$ C … quindi **F non è 3NF**

Esempio 2: copertura minimale

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

$R = (A, B, C, D, E)$ $F = \{ AB \rightarrow C, B \rightarrow D, D \rightarrow C \}$

- Cerchiamo una copertura minimale di F
- Passo 1 - tutte le parti destre sono già singleton
- Passo 2 - proviamo a ridurre $AB \rightarrow C$: $(A)^+_F = \{A\}$, $(B)^+_F = \{B,D,C\}$ che contiene C quindi la dipendenza si può ridurre in $B \rightarrow C$

$G = \{ B \rightarrow C, B \rightarrow D, D \rightarrow C \}$

- Passo 3 - verifichiamo se ci sono dipendenze ridondanti: cominciamo da $B \rightarrow C$ e vediamo che considerando $G = \{B \rightarrow D, D \rightarrow C \}$ abbiamo $(B)^+_G = \{B,D,C\}$ quindi la dipendenza si può eliminare

$G = \{ B \rightarrow D, D \rightarrow C \}$

- Inutile provare con le altre perché le parti destre compaiono una volta sola, quindi abbiamo trovato la copertura minimale

$G = \{ B \rightarrow D, D \rightarrow C \}$

Esempio 2: decomposizione

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

$R = (A, B, C, D, E)$

$G = \{ B \to D, D \to C \}$

- Applichiamo l'algoritmo per la decomposizione dello schema di relazione che non è 3NF dato l' insieme di dipendenze G che è una copertura minimale.

- Al primo passo dobbiamo inserire in un elemento della decomposizione gli attributi che non compaiono nelle dipendenze di G. È il caso degli attributi A ed E, quindi inizialmente avremo $\rho = \{AE\}$ e lo schema R diventa (B,C,D). Passiamo poi a verificare che **non ci sono** dipendenze che coinvolgono tutti gli attributi dello schema, per cui eseguiamo il passo alternativo. Abbiamo alla fine $\rho = \{AE, BD, DC\}$

- per avere una decomposizione con join senza perdita, aggiungiamo alla decomposizione precedente un sottoschema che contenga la chiave ABE (che non è già contenuta in alcuno degli schemi ottenuti). Avremo quindi

$$\sigma = \{\{A, E\}, \{B,D\}, \{D, C\}, \{A, B, E\}\}$$

che si può scrivere anche

$$\sigma = \{AE, BD, DC, ABE\}$$

Esempio 3

Dato lo schema di relazione **R = ABCDEH** e l'insieme di dipendenze funzionali

F = { D → H, B → AC, CD → H, C → AD }

- Determinare l'unica chiave di R
- Dire perché R con l'insieme di dipendenze funzionali F non è in 3NF
- Trovare una decomposizione **ρ** di **R** tale che:
  - ogni schema in ρ è in 3NF
  - ρ preserva F
  - ρ ha un join senza perdita

Esempio 3: chiave

$R = ABCDEH$

$F = \{ D \rightarrow H, B \rightarrow AC, CD \rightarrow H, C \rightarrow AD \}$

- Per verificare se lo schema è 3NF dobbiamo prima di tutto identificare la chiavi
- Notiamo che **l'attributo E deve far parte della chiave**, perché non compare nelle dipendenze e quindi può essere determinato funzionalmente solo per riflessività.
- **A non compare mai** a sinistra (non determina nessun altro attributo) né da solo ne con altri attributi, quindi **non farà parte** della chiave
- Calcoliamo $(D)^+_F = \{D, H\}$. Anche aggiungendo E non abbiamo lo schema R
- Calcoliamo $(B)^+_F = \{B, A, C, D, H\}$. Aggiungendo E abbiamo lo schema R quindi BE è una possibile chiave
  - BE è minimale in quanto alla chiusura di B manca la E e la chiusura di E contiene solo E
- L'esercizio ci dice che BE è l'unica chiave
- $B \rightarrow A$ e $B \rightarrow C$ in F^+ sono dipendenze parziali, $C \rightarrow A$ e $C \rightarrow D$ in F^+ sono dipendenze transitive come le altre dipendenze in F ... quindi **F non è 3NF**

Esempio 3: copertura minimale

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = ABCDEH

$F = \{ D \rightarrow H, B \rightarrow AC, CD \rightarrow H, C \rightarrow AD \}$

- Cerchiamo una copertura minimale di F
- Passo 1 - parti destre diventano singleton

$G = \{ D \rightarrow H, B \rightarrow A, B \rightarrow C, CD \rightarrow H, C \rightarrow A, C \rightarrow D \}$

- Passo 2 - proviamo a ridurre $CD \rightarrow H$: la possiamo eliminare del tutto perché in F c'è $D \rightarrow H$

$G = \{ D \rightarrow H, B \rightarrow A, B \rightarrow C, C \rightarrow A, C \rightarrow D \}$

- Passo 3 - verifichiamo se ci sono dipendenze ridondanti
  - consideriamo che è inutile provare ad eliminare $D \rightarrow H$, $B \rightarrow C$ e $C \rightarrow D$ perché le parti destre compaiono solo in queste dipendenze
  - $B \rightarrow A$: considerando $G = \{D \rightarrow H, B \rightarrow C, C \rightarrow A, C \rightarrow D\}$ abbiamo $(B)^+_G = \{B, C, A, D, H\}$ quindi la dipendenza si può eliminare

$G = \{D \rightarrow H, B \rightarrow C, C \rightarrow A, C \rightarrow D\}$

- Ora tuttel le parti destre compaiono una volta sola, quindi abbiamo trovato la copertura minimale

$G = \{D \rightarrow H, B \rightarrow C, C \rightarrow A, C \rightarrow D\}$

Esempio 3: decomposizione

R = ABCDEH

$G = \{D \rightarrow H, B \rightarrow C, C \rightarrow A, C \rightarrow D\}$

- Applichiamo l'algoritmo per la decomposizione dello schema di relazione che non è 3NF dato l' insieme di dipendenze G che è una copertura minimale.

- Al primo passo dobbiamo inserire in un elemento della decomposizione gli attributi che non compaiono nelle dipendenze di G. È il caso dell'attributo E, quindi inizialmente avremo $\rho = \{E\}$ e lo schema R diventa (A, B, C, D, H). Passiamo poi a verificare che **non ci sono** dipendenze che coinvolgono tutti gli attributi dello schema, per cui eseguiamo il passo alternativo. Abbiamo alla fine $\rho = \{E, DH, BC, CA, CD\}$

- per avere una decomposizione con join senza perdita, aggiungiamo alla decomposizione precedente un sottoschema che contenga la chiave BE (che non è già contenuta in alcuno degli schemi ottenuti). Avremo quindi

$$\sigma = \{E, DH, BC, CA, CD, BE\}$$

Esempio 4

Dato lo schema di relazione **R = ABCDEHI** e l' insieme di dipendenze funzionali

F = { A → C , C → D , BI → H , H → I }

- Trovare le due chiavi dello schema **R** e spiegare perché sono chiavi
- Dire se **R** è in 3NF e giustificare la risposta
- Fornire una decomposizione ρ di R tale che:

  - ogni schema in ρ è in 3NF
  - ρ preserva F
  - ρ ha un join senza perdita

Esempio 4: chiavi

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = ABCDEHI

F = { $A \rightarrow C$, $C \rightarrow D$, $BI \rightarrow H$, $H \rightarrow I$ }

- Per verificare se lo schema è 3NF dobbiamo prima di tutto identificare la chiavi
- Notiamo che **l'attributo E deve far parte delle chiavi** perché non compare nelle dipendenze e quindi può essere determinato funzionalmente solo per riflessività.
- Anche **A e B devono far parte delle chiavi** perché non vengono determinati
- **D non compare mai** a sinistra (non determina nessun altro attributo) né da solo ne con altri attributi, quindi **non farà parte** della chiave
- Calcoliamo $(AB)^+_F$ = {A, B, C, D}. Anche aggiungendo E non abbiamo lo schema R
- Calcoliamo $(C)^+_F$ = {C, D}. Anche aggiungendo E non abbiamo lo schema R
- Calcoliamo $(BI)^+_F$ = {B, I, H}. Anche aggiungendo E non abbiamo lo schema R
- Calcoliamo $(H)^+_F$ = {H, I}. Anche aggiungendo E non abbiamo lo schema R
- Osservando le chiusure (cosa manca e come potrebbe essere inserito) notiamo che potremmo provare ABEI e ABEH infatti $(ABEI)^+_F$ = {A, B, E, I, C, D, H } e $(ABEH)^+_F$ = {A, B, E, H, C, D, I }
- La minimalità è implicita nelle prove che abbiamo già fatto: ABE devono esserci ma non bastano
- $C \rightarrow D$ è transitiva e tutte le altre sono parziali … quindi **F non è 3NF**

Esempio 4: copertura minimale

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = ABCDEHI

F = { A → C , C → D , BI → H , H → I }

- Cerchiamo una copertura minimale di F
- Passo 1 - parti destre sono già singleton
- Passo 2 - proviamo a ridurre BI → H : $(B)^+_F = \{B\}$ e $(I)^+_F = \{I\}$ quindi no
- Passo 3 - verifichiamo se ci sono dipendenze ridondanti
  - consideriamo che è inutile provare ad eliminarle per tutte vale che la parte destra compare una sola volta
- F è già minimale

F = { A → C , C → D , BI → H , H → I }

Esempio 4: decomposizione

R = ABCDEHI

$F = \{ A \rightarrow C, C \rightarrow D, BI \rightarrow H, H \rightarrow I \}$

- Applichiamo l'algoritmo per la decomposizione dello schema di relazione che non è 3NF dato l' insieme di dipendenze G che è una copertura minimale.

- Al primo passo dobbiamo inserire in un elemento della decomposizione gli attributi che non compaiono nelle dipendenze di G. È il caso dell'attributo E, quindi inizialmente avremo ρ = {E} e lo schema R diventa (A, B, C, D, H, I). Passiamo poi a verificare che **non ci sono** dipendenze che coinvolgono tutti gli attributi dello schema, per cui eseguiamo il passo alternativo. Abbiamo alla fine ρ = {E, AC, CD, BIH, HI}
- per avere una decomposizione con join senza perdita, aggiungiamo alla decomposizione precedente un sottoschema che contenga **una** delle chiavi (ABEI e ABEH)(che non sono già contenute in alcuno degli schemi ottenuti). Avremo quindi **due possibili** decomposizioni σ

σ1 = {E, AC, CD, BIH, HI, ABEI}

σ2 = {E, AC, CD, BIH, HI, ABEH}

Esempio 5

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Dato lo schema di relazione **R = ABCD** e l' insieme di dipendenze funzionali
$F = \{ A B \rightarrow C, AD \rightarrow BC, AC \rightarrow B, B \rightarrow D \}$

- Trovare le chiavi dello schema **R** e spiegare perché sono chiavi
- Dire se **R** è in 3NF e giustificare la risposta
- Fornire una decomposizione ρ di R tale che:
  - ogni schema in ρ è in 3NF
  - ρ preserva F
  - ρ ha un join senza perdita

Esempio 5: chiavi

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = ABCD

F = {$A B \rightarrow C$, $AD \rightarrow BC$, $AC \rightarrow B$, $B \rightarrow D$**}**

- Per verificare se lo schema è 3NF dobbiamo prima di tutto identificare la chiavi
- A non viene mai determinato quindi deve appartenere a ogni chiave
- Calcoliamo $(AB)^+_F = \{A, B, C, D\}$ e AB è minimale perché $(A)^+_F = \{A\}$ e $(B)^+_F = \{B, D\}$
- Calcoliamo $(AD)^+_F = \{A, D, B, C\}$.e AD è minimale perché $(D)^+_F = \{D\}$
- Calcoliamo $(AC)^+_F = \{A, C, B, D\}$ e AC è minimale perché $(C)^+_F = \{C\}$

- $B \rightarrow D$ è non crea problemi perché D è primo (fa parte della chiave AD) ma supponiamo di voler comunque decomporre lo schema.

Esempio 5: copertura minimale

R = ABCD

F = {$A\,B \rightarrow C,\ AD \rightarrow BC,\ AC \rightarrow B,\ B \rightarrow D$**}**

- Abbiamo già calcolato la copertura minimale di questo insieme, e anzi ne abbiamo trovate due

$$F1 = \{AD \rightarrow C,\ AC \rightarrow B,\ B \rightarrow D\}$$

$$F2 = \{A\,B \rightarrow C,\ AD \rightarrow B,\ AC \rightarrow B,\ B \rightarrow D\}$$

Esempio 5: decomposizione/1

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = ABCD

F1 = $\{AD \rightarrow C, AC \rightarrow B, B \rightarrow D\}$

- Applichiamo l'algoritmo per la decomposizione dello schema di relazione che non è 3NF dato l' insieme di dipendenze G che è una copertura minimale.

- Al primo passo dobbiamo inserire in un elemento della decomposizione gli attributi che non compaiono nelle dipendenze di F1. . NON CE NE SONO
- Passiamo poi a verificare che **non ci sono** dipendenze che coinvolgono tutti gli attributi dello schema, per cui eseguiamo il passo alternativo. Abbiamo alla fine $\rho 1 = \{ADC, ACB, BD\}$
- per avere una decomposizione con join senza perdita, dovremmo aggiungere alla decomposizione precedente un sottoschema che contenga **una** delle chiavi (AB, AD o AC)(se non sono già contenute in alcuno degli schemi ottenuti). Ma in effetti OGNUNA delle chiavi è già contenuta in qualche sottoschema di ρ quindi $\sigma 1 = \rho 1$

Esempio 5: decomposizione/2

R = ABCD

F2 = {A B $\rightarrow$ C, AD $\rightarrow$ B, AC $\rightarrow$ B, B $\rightarrow$ D}

- Applichiamo l'algoritmo per la decomposizione dello schema di relazione che non è 3NF dato l' insieme di dipendenze G che è una copertura minimale.
- Al primo passo dobbiamo inserire in un elemento della decomposizione gli attributi che non compaiono nelle dipendenze di F2. . NON CE NE SONO
- Passiamo poi a verificare che **non ci sono** dipendenze che coinvolgono tutti gli attributi dello schema, per cui eseguiamo il passo alternativo. Abbiamo alla fine ρ2 = {ABC, ADB, ACB, BD}
- per avere una decomposizione con join senza perdita, dovremmo aggiungere alla decomposizione precedente un sottoschema che contenga **una** delle chiavi (AB, AD o AC)(se non sono già contenute in alcuno degli schemi ottenuti). Ma in effetti OGNUNA delle chiavi è già contenuta in qualche sottoschema di ρ2 quindi σ2= ρ2
- Abbiamo ottenuto due decomposizioni con le caratteristiche richieste

σ1 = {ADC, ACB, BD}

σ2 ={ABC, ADB, ACB, BD}