

Corso di Basi di Dati

Esercitazioni in classe

A.A. 2014/2015

Ultima Esercitazione (Cap. 5,9)

e

soluzioni della simulazione secondo parziale

Esercizio normalizzazione

Dato lo schema di relazione $R(A,B,C,D,E,F,G)$, con dipendenze $AC \rightarrow GD$, $D \rightarrow AEF$, $GE \rightarrow D$, $FG \rightarrow AD$, $BD \rightarrow CE$, $A \rightarrow E$

1. Trovare una copertura canonica.
2. Trovare tutte le chiavi.
3. Dire quali sono, se ci sono, le dipendenze che violano la 3NF.
4. Normalizzare lo schema in 3NF.

Esercizio normalizzazione

Dato lo schema di relazione $R(A,B,C,D,E,F,G)$, con dipendenze $AC \rightarrow GD$, $D \rightarrow AEF$, $GE \rightarrow D$, $FG \rightarrow AD$, $BD \rightarrow CE$, $A \rightarrow E$

1. Trovare una copertura canonica
2. Trovare tutte le chiavi

Algoritmo per la copertura ridotta (Reminder)

I passi per calcolare la copertura ridotta di una relazione sono i seguenti:

1. Sostituzione l'insieme di dipendenze funzionali con un insieme equivalente che ha i secondi membri costituiti da un singolo attributo
2. Per ogni dipendenza verifica dell'esistenza di attributi eliminabili dal primo membro
3. Eliminazione delle dipendenze ridondanti

Esercizio normalizzazione

Dato lo schema di relazione $R(A,B,C,D,E,F,G)$, con dipendenze
 $AC \rightarrow GD$, $D \rightarrow AEF$, $GE \rightarrow D$, $FG \rightarrow AD$, $BD \rightarrow CE$, $A \rightarrow E$

1. Trovare copertura ridotta
 $AC \rightarrow G$, $D \rightarrow A$, $D \rightarrow F$, $GE \rightarrow D$, $FG \rightarrow D$, $BD \rightarrow C$, $A \rightarrow E$
2. Trovare tutte le chiavi
 BD
3. Dire se ci sono (ed eventualmente quali) dipendenze che violano 3NF.
Ricordiamo che 3NF richiede che,
per ogni FD $X \rightarrow Y$ sia soddisfatto:
 - a. X contiene una chiave K di r
oppure
 - b. ogni attributo in Y è contenuto in almeno una chiave di r
4. Normalizzare lo schema in 3NF

Esercizio normalizzazione

Ho:

$AC \rightarrow G,$

$D \rightarrow A,$

$D \rightarrow F$

$GE \rightarrow D,$

$FG \rightarrow D,$

$BD \rightarrow C,$

$A \rightarrow E$

Con chiavi: BD

Dire se ci sono e quali dipendenze violano 3NF, per ogni FD $X \rightarrow Y$:

- X contiene chiave K di r
- ogni attributo in Y è contenuto in almeno una chiave di r

Esercizio normalizzazione

Ho:

$AC \rightarrow G,$	viola 3NF
$D \rightarrow A,$	viola 3NF
$D \rightarrow F,$	viola 3NF
$GE \rightarrow D,$	non viola
$FG \rightarrow D,$	non viola
$BD \rightarrow C,$	non viola
$A \rightarrow E$	viola 3NF

Con chiavi: BD

Non 3NF

Esercizio normalizzazione

4. Normalizzare lo schema in 3NF

$AC \rightarrow G,$
 $D \rightarrow A,$
 $D \rightarrow F,$
 $GE \rightarrow D,$
 $FG \rightarrow D,$
 $BD \rightarrow C,$
 $A \rightarrow E$

Con chiavi: BD

Sintesi di schema in 3NF (Reminder)

Dati uno schema $R(U)$ e un insieme di dipendenze F su U

1. Viene calcolata una copertura ridotta G di F
2. G viene partizionato in sottoinsiemi tali che a ogni insieme appartengono dipendenze che hanno primi membri con la stessa chiusura
3. Viene costruito un insieme \mathbf{U} di sottoinsiemi di U , uno per ciascuna partizione di dipendenze, con tutti gli attributi coinvolti nella partizione
4. Se un elemento di \mathbf{U} è propriamente contenuto in un altro, allora esso viene eliminato da \mathbf{U}
5. Viene costruito uno schema di relazione $R_i(U_i)$ per ciascun elemento $U_i \in \mathbf{U}$ con associate le dipendenze in G i cui attributi sono tutti contenuti in U_i
6. Se nessuno degli U_i è chiave per $R(U)$, allora viene calcolata una chiave K di $R(U)$ e viene aggiunto allo schema generato uno schema di relazione sugli attributi K , senza dipendenze.

Esercizio normalizzazione

4. Normalizzare lo schema in 3NF

Calcolo la chiusura dei primi membri:

$AC \rightarrow G,$ $AC^+ = \{ A, C, E, G, D, F \}$

$D \rightarrow A,$ $D^+ = \{ A, D, F, E \}$
 $D \rightarrow F$

$GE \rightarrow D,$ $GE^+ = \{ G, E, D, A, F \}$

$FG \rightarrow D,$ $FG^+ = \{ F, G, D, A, E \}$

$BD \rightarrow C,$ $BD^+ = \{ B, D, C, A, F, E, G \}$

$A \rightarrow E$ $A^+ = \{ A, E \}$

Esercizio normalizzazione

4. Normalizzare lo schema in 3NF

Calcolo la chiusura dei primi membri:

$AC \rightarrow G,$ $AC^+ = \{ A, C, E, G, D, F \}$

$D \rightarrow A,$ $D^+ = \{ A, D, F, E \}$
 $D \rightarrow F$

$GE \rightarrow D,$ $GE^+ = \{ \mathbf{G, E, D, A, F} \}$

$FG \rightarrow D,$ $FG^+ = \{ \mathbf{F, G, D, A, E} \}$

$BD \rightarrow C,$ $BD^+ = \{ B, D, C, A, F, E, G \}$

$A \rightarrow E$ $A^+ = \{ A, E \}$

Le chiusure di GE e FG coincidono, quindi vanno considerati nella stessa partizione

Esercizio normalizzazione

4. Normalizzare lo schema in 3NF

R1 (A, C, G) $AC \rightarrow G$

R2 (A, D, F) $D \rightarrow A,$
 $D \rightarrow F$

R3 (D, G, E, F) $GE \rightarrow D,$
 $FG \rightarrow D$

R4 (B, D, C) $BD \rightarrow C$

R5 (A, E) $A \rightarrow E$

Esercizio normalizzazione

4. Normalizzare lo schema in 3NF

R1 (A, C, G) $AC \rightarrow G$

R2 (A, D, F) $D \rightarrow A,$
 $D \rightarrow F$

R3 (D, G, E, F) $GE \rightarrow D,$
 $FG \rightarrow D$

R4 (B, D, C) $BD \rightarrow C$

R5 (A, E) $A \rightarrow E$

**R4 CONTIENE LA
CHIAVE DELLA
RELAZIONE ORIGINALE**

Esercizio 5.12

Dato lo schema relazionale:

IMPIEGATO (Nome, Salario, DipNum)

DIPARTIMENTO (DipNum, NomeManager)



Definire le seguenti regole attive (TRIGGER):

1. una regola, che quando il dipartimento è cancellato, mette ad un valore di default (99) il valore di DipNum degli impiegati appartenenti a quel dipartimento
2. una regola che cancella tutti gli impiegati appartenenti a un dipartimento quando quest'ultimo è cancellato
3. una regola che, ogni qual volta il salario di un impiegato supera il salario del suo manager, pone tale salario uguale al salario del manager
4. una regola che, ogni qual volta vengono modificati i salari, verifica che non vi siano dipartimenti in cui il salario medio cresce più del tre per cento, e in tal caso annulla la modifica.

Esercizio 5.12

Dato lo schema relazionale:

IMPIEGATO (Nome, Salario, DipNum)

DIPARTIMENTO (DipNum, NomeManager)

Definire le seguenti regole attive (TRIGGER):

1. una regola, che quando il dipartimento è cancellato, mette ad un valore di default (99) il valore di DipNum degli impiegati appartenenti a quel dipartimento



Esercizio 5.12

Dato lo schema relazionale:

IMPIEGATO (Nome, Salario, DipNum)

DIPARTIMENTO (DipNum, NomeManager)



Definire le seguenti regole attive (TRIGGER):

1. una regola, che quando il dipartimento è cancellato, mette ad un valore di default (99) il valore di DipNum degli impiegati appartenenti a quel dipartimento

SOLUZIONE:

```
create trigger T1
after delete on DIPARTIMENTO
for each row
    update IMPIEGATO set DipNum = 99 where DipNum=old.DipNum
```


Esercizio 5.12

Dato lo schema relazionale:

IMPIEGATO (Nome, Salario, DipNum)

DIPARTIMENTO (DipNum, NomeManager)

Definire le seguenti regole attive (TRIGGER):

2. una regola che cancella tutti gli impiegati appartenenti a un dipartimento quando quest'ultimo è cancellato



Esercizio 5.12

Dato lo schema relazionale:

IMPIEGATO (Nome, Salario, DipNum)

DIPARTIMENTO (DipNum, NomeManager)



Definire le seguenti regole attive (TRIGGER):

2. una regola che cancella tutti gli impiegati appartenenti a un dipartimento quando quest'ultimo è cancellato

SOLUZIONE:

```
create trigger T2
after delete on DIPARTIMENTO
for each row
    delete from IMPIEGATO where DipNum=Old.DipNum
```

Esercizio 5.12

Dato lo schema relazionale:

IMPIEGATO (Nome, Salario, DipNum)

DIPARTIMENTO (DipNum, NomeManager)

Definire le seguenti regole attive (TRIGGER):

3. una regola che, ogni qual volta il salario di un impiegato supera il salario del suo manager, pone tale salario uguale al salario del manager



Esercizio 5.12

3. una regola che, ogni qual volta il salario di un impiegato supera il salario del suo manager, pone tale salario uguale al salario del manager

SOLUZIONE:

```
delimiter |  
create trigger T3  
after update on Impiegato  
for each row  
begin  
    declare sal INT;  
    select I.Salario into sal  
    from Impiegato as I join Dipartimento as D  
        on I.Nome = D.NomeManager  
    Where D.DipNum = New.DipNum;  
    if (new.Salario > sal) then  
        update Impiegato set Salario = sal  
        where Nome = New.Nome;  
    end if;  
end|  
delimiter ;
```



Esercizio 5.12

Dato lo schema relazionale:

IMPIEGATO (Nome, Salario, DipNum)

DIPARTIMENTO (DipNum, NomeManager)

Definire le seguenti regole attive (TRIGGER):

4. una regola che, ogni qual volta vengono modificati i salari, verifica che non vi siano dipartimenti in cui il salario medio cresce più del tre per cento, e in tal caso annulla la modifica.



Esercizio 5.12

4. una regola che, ogni qual volta vengono modificati i salari, verifica che non vi siano dipartimenti in cui il salario medio cresce più del tre per cento, e in tal caso annulla la modifica.

SOLUZIONE:

```
Delimiter |
create trigger T4
after update on Impiegato
for each row
begin
    declare x INT;
    declare y INT;
    declare l INT;
    select avg(Salario), count(*) into x,l
    from Impiegato
    where DipNum=new.DipNum;

    set y=((x*l)-new.Salario+old.Salario)/l;

    if (x>(y*1.03)) then
        update IMPIEGATO set Salario=old.Salario
        where DipNum=new.DipNum;
    end if;
end|
Delimiter ;
```



Simulazione secondo parziale

```
CREATE TABLE Clienti (  
    codice bigint PRIMARY KEY,  
    nome varchar(20) NOT NULL,  
    indirizzo varchar(40) NOT NULL,  
    p_iva char(11) unique  
);
```

```
CREATE TABLE Fatture (  
    codice bigint PRIMARY KEY,  
    cliente integer NOT NULL,  
    data date NOT NULL,  
    FOREIGN KEY (cliente) REFERENCES  
    Clienti(codice)  
);
```

```
CREATE TABLE Prodotti (  
    codice bigint PRIMARY KEY,  
    nome varchar(20) NOT NULL,  
    descrizione varchar(256) NOT NULL,  
    prezzo decimal(8,2) NOT NULL  
);
```

```
CREATE TABLE RigheFatture (  
    codice bigint PRIMARY KEY,  
    fattura integer NOT NULL,  
    prodotto integer NOT NULL,  
    quantita' integer NOT NULL,  
    prezzo decimal (8,2),  
    FOREIGN KEY (fattura)  
        REFERENCES Fatture  
        (codice),  
    FOREIGN KEY (prodotto)  
        REFERENCES Prodotti  
        (codice)  
);
```

Simulazione secondo parziale

Si scriva il codice SQL che implementi le seguenti interrogazioni o regole attive

- a. Ricercare tutti i prodotti acquistati da 'pippo', mostrando la data della fattura, il nome del prodotto e il prezzo come da fattura
- b. Ricercare tutti i prodotti con le rispettive fatture, mostrando il nome del prodotto, la data della fattura e il prezzo a cui e' stato venduto e avendo cura di visualizzare anche i prodotti che non sono mai stati venduti
- c. Calcolare il totale delle vendite sommando il prezzo di tutte le righe di tutte le fatture. Supponendo che l'azienda guadagni all'incirca il 10% su ogni vendita si calcoli anche il guadagno atteso.
- d. Visualizzare per ogni fattura, il codice, la data, il numero di righe e il totale ottenuto sommando il prezzo associato a ciascuna riga
- e. Ripetere l'interrogazione precedente (d) ordinando i dati per data
- f. Visualizzare il nome di ogni cliente con il numero totale di fatture che gli sono state emesse includendo nel conto solo le fatture del 2009, visualizzando solo i clienti che hanno almeno 3 fatture e ordinando le fatture per nome del cliente (N.B. usare funzione YEAR sugli attributi di tipo DATE)
- g. Verificare che una fattura non superi mai le 100 righe (in caso contrario generare un errore)

Simulazione secondo parziale

- A. Ricercare tutti i prodotti acquistati da 'pippo', mostrando la data della fattura, il nome del prodotto e il prezzo come da fattura

```
SELECT fatture.data, prodotti.nome, righefattura.prezzo
FROM clienti, fatture, righefattura, prodotti
WHERE clienti.codice=fatture.cliente
      AND fatture.codice=righefattura.fattura
      AND righefattura.prodotto=prodotti.codice
      AND clienti.nome='pippo'
```

Possibile anche usando dei JOIN ... ON ...

Simulazione secondo parziale

B) Ricercare tutti i prodotti con le rispettive fatture, mostrando il nome del prodotto, la data della fattura e il prezzo a cui e' stato venduto e avendo cura di visualizzare anche i prodotti che non sono mai stati venduti

```
SELECT fatture.data, prodotti.nome, righefatture.prezzo
FROM prodotti LEFT JOIN righefatture ON
            righefatture.prodotto=prodotti.codice
LEFT JOIN fatture ON
            fatture.codice=righefatture.fattura
```

Simulazione secondo parziale

C) Calcolare il totale delle vendite sommando il prezzo di tutte le righe di tutte le fatture. Supponendo che l'azienda guadagni all'incirca il 10% su ogni vendita si calcoli anche il guadagno atteso.

```
SELECT SUM(righefatture.prezzo) AS "Tot. Vendite",  
       SUM(righefatture.prezzo)*0.1 AS "Guadagno Atteso"  
FROM righefatture
```

Simulazione secondo parziale

D) Visualizzare per ogni fattura, il codice, la data, il numero di righe e il totale ottenuto sommando il prezzo associato a ciascuna riga

```
SELECT fatture.codice, fatture.data,  
       SUM(righefatture.prezzo) AS "Tot. Fattura"  
FROM righefatture, fatture  
WHERE righefatture.fattura=fatture.codice  
GROUP BY fatture.codice, fatture.data
```

Simulazione secondo parziale

E) Ripetere l'interrogazione precedente (D) ordinando i dati per data

```
SELECT fatture.codice, fatture.data,  
       SUM(righefatture.prezzo) AS "Tot. Fattura"  
FROM righefatture, fatture  
WHERE righefatture.fattura=fatture.codice  
GROUP BY fatture.codice, fatture.data  
ORDER BY fatture.data
```

Simulazione secondo parziale

F) Visualizzare il nome di ogni cliente con il numero totale di fatture che gli sono state emesse includendo nel conto solo le fatture del 2009, visualizzando solo i clienti che hanno almeno 3 fatture e ordinando le fatture per nome del cliente (N.B. usare funzione YEAR sugli attributi di tipo DATE)

```
SELECT clienti.nome, COUNT(*) AS "Numero Righe"  
  FROM clienti, fatture  
 WHERE fatture.cliente=clienti.codice AND YEAR(fatture.  
data)=2009  
 GROUP BY clienti.nome  
 HAVING count(*) > 1  
 ORDER BY clienti.nome
```

Simulazione secondo parziale

G) Verificare che una fattura non superi mai le 100 righe (in caso contrario generare un errore)

```
DELIMITER |
CREATE TRIGGER maxRighe
BEFORE INSERT ON RigheFattura
FOR EACH ROW
BEGIN
    DECLARE numRighe INT;
    SELECT COUNT(*)
    FROM RigheFattura
    WHERE fattura=NEW.fattura
    INTO numRighe;
    IF (numRighe > 99) THEN
        INSERT INTO RigheFattura SELECT * FROM RigheFattura
        LIMIT 1;
    END IF;
END|
DELIMITER ;
```

Simulazione secondo parziale

Dato lo schema di relazione $R(A,B,C,D,E,F)$ con le dipendenze

$BD \rightarrow F$, $FE \rightarrow BA$, $F \rightarrow C$, $B \rightarrow E$, $C \rightarrow A$, $E \rightarrow CD$

- Trovare la copertura ridotta
- Trovare tutte le chiavi di R
- Dire se e quali dipendenze violano la 3NF
- Normalizzare lo schema in 3NF

Simulazione secondo parziale

Dato lo schema di relazione $R(A,B,C,D,E,F)$ con le dipendenze

$BD \rightarrow F$, $FE \rightarrow BA$, $F \rightarrow C$, $B \rightarrow E$, $C \rightarrow A$, $E \rightarrow CD$

SOLUZIONE

a. Trovare la copertura ridotta:

$B \rightarrow F$, $FE \rightarrow B$, $F \rightarrow C$, $B \rightarrow E$, $C \rightarrow A$, $E \rightarrow C$, $E \rightarrow D$

b. Trovare tutte le chiavi di R:

B , FE

c. Dire se e quali dipendenze violano la 3NF

$B \rightarrow F$, non viola

$FE \rightarrow B$, non viola

$F \rightarrow C$, VIOLA

$B \rightarrow E$, non viola

$C \rightarrow A$, VIOLA

$E \rightarrow C$, VIOLA

$E \rightarrow D$ VIOLA

d. Normalizzare lo schema in 3NF

$R_1(B, F, E)$ $R_2(C, A)$ $R_3(E, C, D)$ $R_4(F, C)$