

Esonero 14 07 2021

1. Esonero SQL V1

Le seguenti relazioni definiscono una base di dati “ILoveMyDAD” per la gestione delle videolezioni. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

CORSO(Codice, Nome, CFU)

VIDEOLEZIONE(Argomento, NomeDocente, Durata, CFU, Formato)

PROGRAMMA(Corso, Videolezione, Ordine, Laboratorio)

Vincoli di integrità referenziale:

PROGRAMMA(Corso) referencia CORSO(Codice),

PROGRAMMA(Videolezione) referencia VIDEOLEZIONE(Argomento).

Significato delle relazioni e degli attributi (CFU sta per crediti formativi universitari). *CORSO* contiene le informazioni sui corsi insegnati (gli attributi sono autoesplicativi), *VIDEOLEZIONE* contiene le informazioni sugli spezzoni delle videolezioni: *Argomento* contiene il titolo della videolezione; *NomeDocente* è il nome del docente che tiene la videolezione (può essere NULL se il video è un contenuto audiovisivo esterno); *Durata* contiene la durata della videolezione in minuti; *Formato* può assumere i valori *audioslide*, *registrazione*, *interattivo*. *PROGRAMMA* associa gli spezzoni delle videolezioni ai corsi, in un ordine preferenziale di visualizzazione determinato dall'attributo *Ordine*. *Laboratorio* è un attributo booleano che assume il valore TRUE se la videolezione è parte del laboratorio del corso, FALSE se fa parte della teoria.

Infine, vale il seguente vincolo: il valore dell'attributo CFU in CORSO è uguale alla somma dei CFU delle sue videolezioni.

Esprimere in SQL tutte e tre le seguenti interrogazioni indicando a quale query si sta rispondendo (A, B, C):

(A - Bassa complessità) Elencare, senza duplicati e ordinate per nome del docente crescente, le videolezioni di laboratorio i cui CFU (i CFU della singola videolezione) sono più di un terzo dei CFU totali del corso.

(B - Media complessità) Trovare i corsi da almeno 6 CFU con più di 3 CFU di laboratorio. Mostrare il nome del corso, il numero di lezioni di laboratorio e il numero dei CFU di teoria del corso.

(C - Alta complessità) Trovare i corsi senza laboratorio che hanno più videolezioni interattive che audioslide.

Soluzioni.

A.

```
select distinct v.argomento, v.nomedocente
from corso c join programma p on c.codice=p.corso
join videolezione v on v.argomento=c.videolezione
where v.cfu > (1/3)*c.cfu and p.laboratorio=True
order by v.nomedocente;
```

B.

```
select c.nome, count(*) as NumVL, c.cfu-sum(v.cfu) as CFUTeoria
from corso c join programma p on c.codice=p.corso
join videolezione v on v.argomento=c.videolezione
where c.cfu >= 6 and p.laboratorio=True
group by c.nome
having sum(v.cfu) > 3;
```

C.

```
with corsiNoLabLI as (
select p.corso, count(*) as NLI
```

```

from programma p join videolezione v on p.videolezione=v.argomento
where v.formato='interattiva' and not exists (
select *
from programma p2
where p2.corso=p.corso and p2.laboratorio=True)
group by p.corso),
corsiNoLabAS as(
select p.corso, count(*) as NAS
from programma p join videolezione v on p.videolezione=v.argomento
where v.formato='audioslide' and not exists (
select *
from programma p2
where p2.corso=p.corso and p2.laboratorio=True)
group by p.corso)
select c1.corso
from corsiNoLabLI c1 join corsiNolabAS c2 on c1.corso=c2.corso
where c1.NLI > c2.NAS;

```

2. Esonero SQL V2

Le seguenti relazioni definiscono una base di dati “ILoveMyDAD” per la gestione delle videolezioni. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

CORSO(Codice, Nome, CFU)

VIDEOLEZIONE(Argomento, NomeDocente, Durata, CFU, Formato)

PROGRAMMA(Corso, Videolezione, Ordine, Laboratorio)

Vincoli di integrità referenziale:

PROGRAMMA(Corso) riferenzia CORSO(Codice),

PROGRAMMA(Videolezione) riferenzia VIDEOLEZIONE(Argomento).

Significato delle relazioni e degli attributi (CFU sta per crediti formativi universitari). *CORSO* contiene le informazioni sui corsi insegnati (gli attributi sono autoesplicativi), *VIDEOLEZIONE* contiene le informazioni sugli spezzoni delle videolezioni: *Argomento* contiene il titolo della videolezione; *NomeDocente* è il nome del docente che tiene la videolezione (può essere NULL se il video è un contenuto audiovisivo esterno); *Durata* contiene la durata della videolezione in minuti; *Formato* può assumere i valori *audioslide*, *registrazione*, *interattivo*. *PROGRAMMA* associa gli spezzoni delle videolezioni ai corsi, in un ordine preferenziale di visualizzazione determinato dall'attributo *Ordine*. *Laboratorio* è un attributo booleano che assume il valore TRUE se la videolezione è parte del laboratorio del corso, FALSE se fa parte della teoria.

Infine, vale il seguente vincolo: il valore dell'attributo CFU in CORSO è uguale alla somma dei CFU delle sue videolezioni.

Esprimere in SQL tutte e tre le seguenti interrogazioni indicando a quale query si sta rispondendo (A, B, C):

(A - Bassa complessità) Elencare, senza duplicati e ordinate per durata decrescente, le videolezioni di teoria i cui CFU (i CFU della singola videolezione) non eccedono la metà dei CFU totali del corso.

(B - Media complessità) Trovare i corsi da meno di 5 CFU per cui lo stesso docente abbia erogato tutte le videolezioni utilizzando almeno due formati di lezione diversi. Mostrare il nome del corso e la quantità di ore di lezione erogate. La query non deve essere legata agli specifici nomi di formato ma deve funzionare anche nel caso in cui i formati cambino.

(C - Alta complessità) Trovare i corsi senza audioslide che hanno meno lezioni di teoria che di laboratorio.

Soluzioni.

A.

```

select distinct v.argomento, v.nomedocente
from corso c join programma p on c.codice=p.corso join videolezione v on
v.argomento=c.videolezione
where v.cfu <= (1/2)*c.cfu and p.laboratorio=False
order by v.durata desc;

```

B.

```

select c.nome, sum(v.durata) as DurataLezioni
from corso c join programma p on c.codice=p.corso
join videolezione v on v.argomento=p.videolezione
where c.cfu < 5
group by c.codice
having count(distinct v.nomedocente) = 1 and count(distinct v.formato) >= 2;

```

C.

```

with corsiNoASLab as (
select p.corso, count(*) as NASL
from programma p join videolezione v on p.videolezione=v.argomento
where p.laboratorio=True and not exists (
select *
from videolezione v2
where v2.corso=v.corso and v2.formato='audioslide' )
group by p.corso ),

```

```

corsiNoASTeo as (
select p.corso, count(*) as NAST
from programma p join videolezione v on p.videolezione=v.argomento
where p.laboratorio=False and not exists (
select *
from videolezione v2
where v2.corso=v.corso and v2.formato='audioslide' )
group by p.corso )

```

```

select c1.corso
from corsiNoASLab c1 join corsiNoASTeo c2 on c1.corso=c2.corso
where c1.NASL > c2.NAST);

```

3. Esonero Algebra V1

Le seguenti relazioni definiscono una base di dati “ILoveMyDAD” per la gestione delle videolezioni. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

CORSO(Codice, Nome, CFU)

VIDEOLEZIONE(Argomento, NomeDocente, Durata, CFU, Formato)

PROGRAMMA(Corso, Videolezione, Ordine, Laboratorio)

Vincoli di integrità referenziale:

PROGRAMMA(Corso) riferenzia CORSO(Codice),

PROGRAMMA(Videolezione) riferenzia VIDEOLEZIONE(Argomento).

Significato delle relazioni e degli attributi (CFU sta per crediti formativi universitari). *CORSO* contiene le informazioni sui corsi insegnati (gli attributi sono autoesplicativi), *VIDEOLEZIONE* contiene le informazioni sugli spezzoni delle videolezioni: *Argomento* contiene il titolo della videolezione; *NomeDocente* è il nome del docente che tiene la videolezione (può essere NULL se il video è un contenuto audiovisivo esterno); *Durata* contiene la durata della videolezione in minuti; *Formato* può assumere i valori *audioslide*, *registrazione*, *interattivo*. *PROGRAMMA* associa gli spezzoni delle

videolezioni ai corsi, in un ordine preferenziale di visualizzazione determinato dall'attributo *Ordine*. *Laboratorio* è un attributo booleano che assume il valore TRUE se la videolezione è parte del laboratorio del corso, FALSE se fa parte della teoria.

Infine, vale il seguente vincolo: il valore dell'attributo CFU in CORSO è uguale alla somma dei CFU delle sue videolezioni.

Esprimere tutte e due le seguenti interrogazioni indicando a quale query si sta rispondendo (per facilità di scrittura delle formule, si possono eseguire gli esercizi con carta e penna e fare l'upload delle foto prestando attenzione che lo svolgimento sia chiaramente leggibile):

(A - Algebra Relazionale) Trovare la più lunga videolezione introduttiva (cioè usata all'inizio di un corso). Mostrarne argomento e numero di CFU.

(B - Calcolo Relazionale su tuple con dichiarazione di range) Trovare i corsi con più di 6 CFU composti da sole videolezioni nel formato registrazione.

Soluzioni.

A. $\pi_{argomento, cfu}(\sigma_{Ordine=1}(\text{programma} \bowtie_{videolezione=argomento} videolezione)) -$
 $\pi_{V1.argomento, V1.cfu}(\sigma_{Ordine=1}(\rho_{P1 \leftarrow \text{programma}}(\text{programma})$
 $\bowtie_{P1.videolezione=V1.argomento} \rho_{V1 \leftarrow videolezione}(videolezione))$
 $\bowtie_{V1.durata < V2.durata} (\sigma_{Ordine=1}(\rho_{P2 \leftarrow \text{programma}}(\text{programma})$
 $\bowtie_{P2.videolezione=V2.argomento} \rho_{V2 \leftarrow videolezione}(videolezione))))$

B. $\{c.codice, c.nome | c(corso) | c.cfu > 6 \wedge$
 $\forall p(\text{programma})((p.corso = c.codice) \Rightarrow$
 $(\exists v(videolezione)(p.videolezione = v.argomento \wedge v.formato = 'registrazione')))\}$

4. Esonero Algebra V2

Le seguenti relazioni definiscono una base di dati "ILoveMyDAD" per la gestione delle videolezioni. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

CORSO(Codice, Nome, CFU)

VIDEOLEZIONE(Argomento, NomeDocente, Durata, CFU, Formato)

PROGRAMMA(Corso, Videolezione, Ordine, Laboratorio)

Vincoli di integrità referenziale:

PROGRAMMA(Corso) referencia CORSO(Codice),

PROGRAMMA(Videolezione) referencia VIDEOLEZIONE(Argomento).

Significato delle relazioni e degli attributi (CFU sta per crediti formativi universitari). *CORSO* contiene le informazioni sui corsi insegnati (gli attributi sono autoesplicativi), *VIDEOLEZIONE* contiene le informazioni sugli spezzoni delle videolezioni: *Argomento* contiene il titolo della videolezione; *NomeDocente* è il nome del docente che tiene la videolezione (può essere NULL se il video è un contenuto audiovisivo esterno); *Durata* contiene la durata della videolezione in minuti; *Formato* può assumere i valori *audioslide*, *registrazione*, *interattivo*. *PROGRAMMA* associa gli spezzoni delle videolezioni ai corsi, in un ordine preferenziale di visualizzazione determinato dall'attributo *Ordine*. *Laboratorio* è un attributo booleano che assume il valore TRUE se la videolezione è parte del laboratorio del corso, FALSE se fa parte della teoria.

Infine, vale il seguente vincolo: il valore dell'attributo CFU in CORSO è uguale alla somma dei CFU delle sue videolezioni.

Esprimere tutte e due le seguenti interrogazioni indicando a quale query si sta rispondendo (per facilità di scrittura delle formule, si possono eseguire gli esercizi con carta e penna e fare l'upload delle foto prestando attenzione che lo svolgimento sia chiaramente leggibile):

(A - Algebra Relazionale) Trovare i corsi con più di 6 CFU composti da sole videolezioni nel formato interattivo. Mostrare nome del corso. Rappresentare la query come albero sintattico.

(B - Calcolo Relazionale su tuple con dichiarazione di range) Trovare la più lunga videolezione introduttiva (cioè usata all'inizio di un corso). Mostrarne argomento e numero di CFU.

Soluzioni.

A. $\rho_{\text{corso} \leftarrow \text{nome}}(\pi_{\text{nome}}(\sigma_{\text{cfu} > 6}(\text{corso}))) -$

$\pi_{\text{corso}}(\sigma_{\text{formato} \neq \text{'interattivo'}}(\text{programma} \bowtie_{\text{videolezione}=\text{argomento}} \text{videolezione}))$

B. $\{v.\text{argomento}, v.\text{cfu} | v(\text{videolezione}) | \exists p(\text{programma})(v.\text{argomento} = p.\text{videolezione} \wedge p.\text{ordine} = 1 \wedge \nexists v'(\text{videolezione})(v'.\text{durata} > v.\text{durata} \wedge \exists(p'(\text{programma})(v'.\text{argomento} = p'.\text{videolezione} \wedge p'.\text{ordine} = 1))))\}$

5. Esonero Ottimizzazione V1

Le seguenti relazioni definiscono una base di dati “ILoveMyDAD” per la gestione delle videolezioni. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

CORSO(Codice, Nome, CFU)

VIDEOLEZIONE(Argomento, NomeDocente, Durata, CFU, Formato)

PROGRAMMA(Corso, Videolezione, Ordine, Laboratorio)

Vincoli di integrità referenziale:

PROGRAMMA(Corso) referencia CORSO(Codice),

PROGRAMMA(Videolezione) referencia VIDEOLEZIONE(Argomento).

Significato delle relazioni e degli attributi (CFU sta per crediti formativi universitari). *CORSO* contiene le informazioni sui corsi insegnati (gli attributi sono autoesplicativi), *VIDEOLEZIONE* contiene le informazioni sugli spezzoni delle videolezioni: *Argomento* contiene il titolo della videolezione; *NomeDocente* è il nome del docente che tiene la videolezione (può essere NULL se il video è un contenuto audiovisivo esterno); *Durata* contiene la durata della videolezione in minuti; *Formato* può assumere i valori *audioslide*, *registrazione*, *interattivo*. *PROGRAMMA* associa gli spezzoni delle videolezioni ai corsi, in un ordine preferenziale di visualizzazione determinato dall'attributo *Ordine*. *Laboratorio* è un attributo booleano che assume il valore TRUE se la videolezione è parte del laboratorio del corso, FALSE se fa parte della teoria.

Infine, vale il seguente vincolo: il valore dell'attributo CFU in CORSO è uguale alla somma dei CFU delle sue videolezioni.

Data la seguente query:

$\sigma_{\text{formato}=\text{'registrazione'} \wedge \text{CFU} < 1.2 \wedge \text{laboratorio}=\text{False}}(\text{videolezione} \bowtie_{\text{argomento}=\text{videolezione}} \text{programma})$

disegnare gli alberi sintattici prima e dopo l'ottimizzazione logica e calcolare il numero di tuple “mosse” prima e dopo l'ottimizzazione logica.

Si svolgano i calcoli sapendo che:

$\text{CARD}(\text{videolezione}) = 540$

$\text{CARD}(\text{programma}) = 600$

$\text{VAL}(\text{videolezione}, \text{programma}) = \text{CARD}(\text{videolezione}) = 540$

$\text{MIN}(\text{cfu}, \text{videolezione}) = 0.2$

$\text{MAX}(\text{cfu}, \text{videolezione}) = 2.2$

Soluzioni.

La query ottimizzata dividendo la selezione e portandola verso le foglie è

$\sigma_{\text{formato}=\text{'registrazione'} \wedge \text{CFU} < 1.2}(\text{videolezione}) \bowtie_{\text{argomento}=\text{videolezione}} \sigma_{\text{laboratorio}=\text{False}}(\text{programma})$

Prima dell'ottimizzazione:

- Costo $r_1 = (\text{videolezione} \bowtie_{\text{argomento}=\text{videolezione}} \text{programma}) : 540 \cdot 600 = 324\,000$.
- Cardinalità di $|r_1| = \text{CARD}(\text{programma}) = 600$ (equijoin attraverso la chiave esterna)
- Costo della selezione: $|r_1|$
- Costo totale = $324\,000 + 600 \approx 325\,000$.

Dopo l'ottimizzazione:

- Costo $\sigma_1 = \sigma_{formato='registrazione' \wedge CFU < 1.2}(videolezione) = CARD(videolezione) = 540$
- Costo $\sigma_2 = \sigma_{laboratorio=False}(programma) = 600$
- Tuple prodotte dalla selezione $|\sigma_1| = \frac{1}{VAL(formato, videolezione)} \cdot \frac{1.2 - MIN(cfu, videolezione)}{MAX(cfu, videolezione) - MIN(cfu, videolezione)} \cdot CARD(videolezione) = \frac{1}{3} \cdot \frac{1}{2} \cdot 540 = 90$
- Tuple prodotte dalla selezione $|\sigma_2| = \frac{1}{VAL(laboratorio, programma)} \cdot CARD(programma) = \frac{1}{2} \cdot 600 = 300$
- Costo join $r = \sigma_1 \bowtie_{argomento=videolezione} \sigma_2$: $|\sigma_1| \cdot |\sigma_2| = 90 \cdot 300 = 27\,000$.
- Costo totale = $27\,000 + 540 + 600 \approx 28\,000$

6. Esonero Ottimizzazione V2

Le seguenti relazioni definiscono una base di dati “ILoveMyDAD” per la gestione delle videolezioni. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

CORSO(Codice, Nome, CFU)

VIDEOLEZIONE(Argomento, NomeDocente, Durata, CFU, Formato)

PROGRAMMA(Corso, Videolezione, Ordine, Laboratorio)

Vincoli di integrità referenziale:

PROGRAMMA(Corso) referencia CORSO(Codice),

PROGRAMMA(Videolezione) referencia VIDEOLEZIONE(Argomento).

Significato delle relazioni e degli attributi (CFU sta per crediti formativi universitari). *CORSO* contiene le informazioni sui corsi insegnati (gli attributi sono autoesplicativi), *VIDEOLEZIONE* contiene le informazioni sugli spezzoni delle videolezioni: *Argomento* contiene il titolo della videolezione; *NomeDocente* è il nome del docente che tiene la videolezione (può essere NULL se il video è un contenuto audiovisivo esterno); *Durata* contiene la durata della videolezione in minuti; *Formato* può assumere i valori *audioslide*, *registrazione*, *interattivo*. *PROGRAMMA* associa gli spezzoni delle videolezioni ai corsi, in un ordine preferenziale di visualizzazione determinato dall'attributo *Ordine*. *Laboratorio* è un attributo booleano che assume il valore TRUE se la videolezione è parte del laboratorio del corso, FALSE se fa parte della teoria.

Infine, vale il seguente vincolo: il valore dell'attributo CFU in CORSO è uguale alla somma dei CFU delle sue videolezioni.

Data la seguente query:

$\sigma_{laboratorio=True \wedge formato \neq 'audioslide' \wedge CFU > 1.7}(videolezione \bowtie_{argomento=videolezione} programma)$

disegnare gli alberi sintattici prima e dopo l'ottimizzazione logica e calcolare il numero di tuple “mosse” prima e dopo l'ottimizzazione logica.

Si svolgano i calcoli sapendo che:

$CARD(videolezione) = 540$

$CARD(programma) = 600$

$VAL(videolezione, programma) = CARD(videolezione) = 540$

$MIN(cfu, videolezione) = 0.2$

$MAX(cfu, videolezione) = 2.2$

Soluzioni.

La query ottimizzata dividendo la selezione e portandola verso le foglie è

$\sigma_{formato \neq 'audioslide' \wedge CFU > 1.7}(videolezione) \bowtie_{argomento=videolezione} \sigma_{Laboratorio=True}(programma)$

Prima dell'ottimizzazione:

- Costo $r_1 = (\text{videolezione} \bowtie_{\text{argomento}=\text{videolezione}} \text{programma} : 540 \cdot 600 = 324\,000$.
- Cardinalità di $|r_1| = \text{CARD}(\text{programma}) = 600$ (equijoin attraverso la chiave esterna)
- Costo della selezione: $|r_1|$
- Costo totale $= 324\,000 + 600 \approx 325\,000$.

Dopo l'ottimizzazione:

- Costo $\sigma_1 = \sigma_{\text{formato} \neq \text{'audioslide'} \wedge \text{CFU} > 1.7}(\text{videolezione}) = \text{CARD}(\text{videolezione}) = 540$
- Costo $\sigma_2 = \sigma_{\text{Laboratorio}=\text{True}}(\text{programma}) = 600$
- Tuple prodotte dalla selezione $|\sigma_1| = (1 - \frac{1}{\text{VAL}(\text{formato}, \text{videolezione})}) \cdot \frac{\text{MAX}(\text{cfu}, \text{videolezione}) - 1.7}{\text{MAX}(\text{cfu}, \text{videolezione}) - \text{MIN}(\text{cfu}, \text{videolezione})} \cdot \text{CARD}(\text{videolezione}) = \frac{2}{3} \cdot \frac{0.5}{2} \cdot 540 = 90$
- Tuple prodotte dalla selezione $|\sigma_2| = \frac{1}{\text{VAL}(\text{laboratorio}, \text{programma})} \cdot \text{CARD}(\text{programma}) = \frac{1}{2} \cdot 600 = 300$
- Costo join $r = \sigma_1 \bowtie_{\text{argomento}=\text{videolezione}} \sigma_2$: $|\sigma_1| \cdot |\sigma_2| = 90 \cdot 300 = 27\,000$.
- Costo totale $= 27\,000 + 540 + 600 \approx 28\,000$

7. Esonero Teoria 3NF V1

Dati:

$R(\text{Bu}, \text{Ca}, \text{E}, \text{N}, \text{O}, \text{Va}, \text{Ze})$ e

$F = \{\text{Bu} \text{ N O} \rightarrow \text{E}, \text{E} \rightarrow \text{N O}, \text{N} \rightarrow \text{Ca Va}, \text{N Va} \rightarrow \text{Ze}\}$

dire, motivando la risposta, se R è in 3FN e se non lo è decomporla in relazioni in 3FN esplicitando tutti i passaggi. Il risultato è BCNF? Perché?

Soluzione.

Per prima cosa, occorre individuare la o le chiavi della relazione R. Ogni chiave deve contenere Bu, perché non compare a destra in nessuna dipendenza funzionale. La chiave non può contenere Ca, Va e Ze, perché compaiono sempre e solo a destra. Una possibile superchiave è quindi: $\{\text{Bu E N O}\}^+ = \{\text{Bu E N O} | \text{Ca} | \text{Va} | \text{Ze}\}$, ma non è chiave, infatti i suoi sottoinsiemi $\{\text{Bu E}\}^+ = \{\text{Bu E} | \text{N O} | \text{Ca} | \text{Va} | \text{Ze}\}$ e $\{\text{Bu N O}\}^+ = \{\text{Bu N O} | \text{E} | \text{Ca} | \text{Va} | \text{Ze}\}$ sono superchiavi minimali e quindi chiavi candidate.

La relazione non è in 3NF, infatti per esempio $\text{N Va} \rightarrow \text{Ze}$ non è riflessiva, non ha a sinistra una superchiave né a destra attributi primi.

Troviamo quindi la copertura minimale di F, partendo dalla sua forma canonica:

$F' = \{\text{Bu} \text{ N O} \rightarrow \text{E}, \text{E} \rightarrow \text{N}, \text{E} \rightarrow \text{O}, \text{N} \rightarrow \text{Ca}, \text{N} \rightarrow \text{Va}, \text{N Va} \rightarrow \text{Ze}\}$

Va è estraneo in $\text{N Va} \rightarrow \text{Ze}$, quindi:

$F' = \{\text{Bu} \text{ N O} \rightarrow \text{E}, \text{E} \rightarrow \text{N}, \text{E} \rightarrow \text{O}, \text{N} \rightarrow \text{Ca}, \text{N} \rightarrow \text{Va}, \text{N} \rightarrow \text{Ze}\}$

non ci sono dipendenze funzionali ridondanti.

Possiamo ora scomporre in 3NF e otteniamo le seguenti relazioni dopo aver accorpato le d.f. con lo stesso antecedente:

$R1(\text{Bu}, \text{O}, \text{N}, \text{E})$

$R2(\underline{\text{E}}, \text{N}, \text{O})$

$R3(\text{Va}, \text{Ca}, \underline{\text{N}}, \text{Ze})$

R1 contiene una chiave. È ancora possibile includere R2 in R1, quindi:

$R1(\text{Bu}, \text{O}, \text{N}, \text{E})$ con $F_1 = \{\text{E} \rightarrow \text{N O}\}$

$R3(\text{Va}, \text{Ca}, \underline{\text{N}}, \text{Ze})$

Il risultato non è BCNF perché R1 contiene una dipendenza funzionale che non è né superchiave né banale.

8. Esonero Teoria 3NF V2

Dati:

$R(A, B, C, D, E, F, G)$ e

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

dire, motivando la risposta, se R è in 3FN e se non lo è decomporla in relazioni in 3FN esplicitando tutti i passaggi. Il risultato è BCNF? Perché?

Soluzione.

Per prima cosa, occorre individuare la o le chiavi della relazione R. Ogni chiave deve contenere A, perché non compare a destra in nessuna dipendenza funzionale. La chiave non può contenere B, F e G, perché compaiono sempre e solo a destra. Una possibile superchiave è quindi: $\{A C D E\}^+ = \{A C D E | B | F | G\}$, ma non è chiave, infatti i suoi sottoinsiemi $\{A C\}^+ = \{A C | D E | B | F | G\}$ e $\{A D E\}^+ = \{A D E | C | B | F | G\}$ sono superchiavi minimali e quindi chiavi candidate.

La relazione non è in 3NF, infatti per esempio $D F \rightarrow G$ non è riflessiva, non ha a sinistra una superchiave né a destra attributi primi.

Troviamo quindi la copertura minimale di F, partendo dalla sua forma canonica:

$F' = \{A D E \rightarrow C, C \rightarrow D, C \rightarrow E, D \rightarrow B, D \rightarrow F, D F \rightarrow G\}$

F è estraneo in $D F \rightarrow G$, quindi:

$F' = \{A D E \rightarrow C, C \rightarrow D, C \rightarrow E, D \rightarrow B, D \rightarrow F, D \rightarrow G\}$

non ci sono dipendenze funzionali ridondanti.

Possiamo ora scomporre in 3NF e otteniamo le seguenti relazioni dopo aver accorpato le d.f. con lo stesso antecedente:

$R1(\underline{A}, E, D, C)$

$R2(\underline{C}, D, E)$

$R3(\underline{D}, F, B, G)$

R1 contiene una chiave. È ancora possibile includere R2 in R1, quindi:

$R1(\underline{A}, E, D, C)$ con $F_1 = \{C \rightarrow D E\}$

$R3(\underline{D}, F, B, G)$

Il risultato non è BCNF perché R1 contiene una dipendenza funzionale che non è né superchiave né banale.

9. Esonero Teoria Armstrong V1

Data $R(A, B, C, D, E, F, G)$ con $F = \{C \rightarrow FG, D \rightarrow A, A \rightarrow F, E \rightarrow BG\}$

dimostrare che $R1(A, B, C, D, E)$ e $R2(C, D, E, F, G)$ è una decomposizione senza perdita di informazione usando la teoria di Armstrong. Suggerimento: partire dalla definizione di decomposizione senza perdita di informazione.

Soluzione. Affinché R1 ed R2 siano il risultato di una decomposizione senza perdita di informazione è necessario che $\{A, B, C, D, E\} \cap \{C, D, E, F, G\} = \{C D E\}$ sia una superchiave per R1 o per R2, cioè che $C D E \rightarrow A B C D E$ sia deducibile da F oppure che $C D E \rightarrow C D E F G$ sia deducibile da F.

Per l'assioma di riflessività $C D E \rightarrow C D E$. Per la regola del prodotto da $C \rightarrow F G, D \rightarrow A, E \rightarrow B G$ otteniamo $C D E \rightarrow A B F G$.

Per l'assioma dell'unione da $C D E \rightarrow C D E$ e $C D E \rightarrow A B F G$ otteniamo $C D E \rightarrow A B C D E F G$.

Infine per la regola di decomposizione da $C D E \rightarrow A B C D E F G$ otteniamo $C D E \rightarrow A B C D E$.

10. Esonero Teoria Armstrong V2

Data $R(A, B, C, D, E, F, G)$ con $F = \{A \rightarrow DE, B \rightarrow F, F \rightarrow D, C \rightarrow EG\}$

dimostrare che $R_1(A, B, C, D, E)$ e $R_2(A, B, C, F, G)$ è una decomposizione senza perdita di informazione usando la teoria di Armstrong. Suggerimento: partire dalla definizione di decomposizione senza perdita di informazione.

Soluzione. Affinché R_1 ed R_2 siano il risultato di una decomposizione senza perdita di informazione è necessario che $\{A, B, C, D, E\} \cap \{A, B, C, F, G\} = \{ABC\}$ sia una superchiave per R_1 o per R_2 , cioè che $ABC \rightarrow ABCDE$ sia deducibile da F oppure che $CDE \rightarrow ABCFG$ sia deducibile da F .

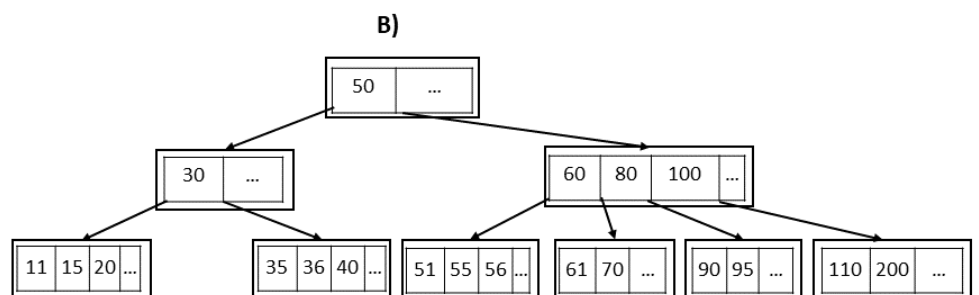
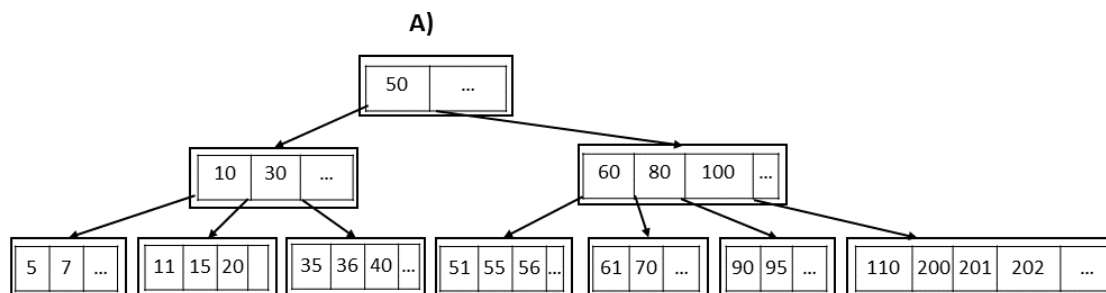
Per l'assioma di riflessività $ABC \rightarrow ABC$. Per la regola del prodotto da $A \rightarrow DE, B \rightarrow F, C \rightarrow EG$ otteniamo $ABC \rightarrow DEFG$.

Per l'assioma dell'unione da $ABC \rightarrow ABC$ e $ABC \rightarrow DEFG$ otteniamo $ABC \rightarrow ABCDEFG$.

Infine per la regola di decomposizione da $ABC \rightarrow ABCDEFG$ otteniamo $ABC \rightarrow ABCDE$.

11. Esonero Teoria B+-Tree V1

Elencare, se esistono, tutti i valori di m che rendono i seguenti B-tree entrambi contemporaneamente legali, motivando la risposta.



Soluzione.

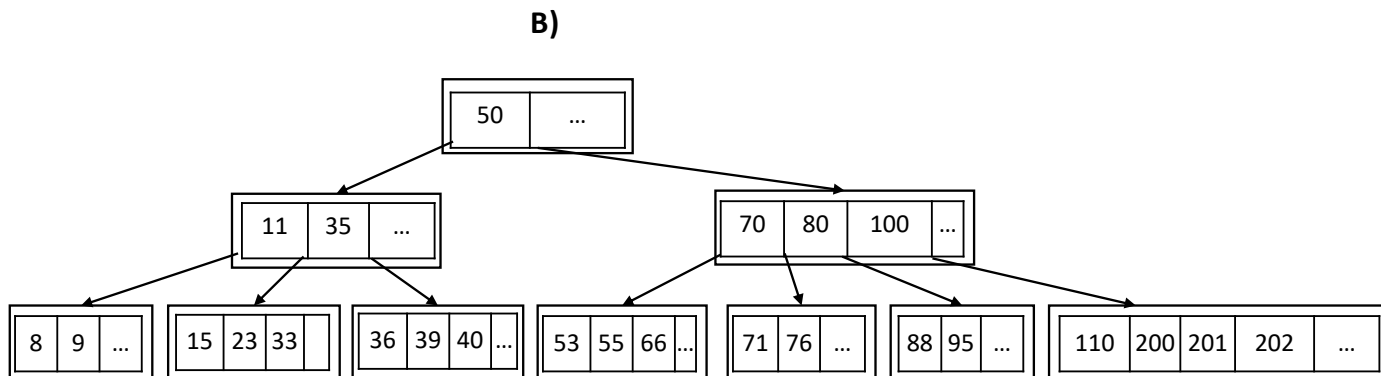
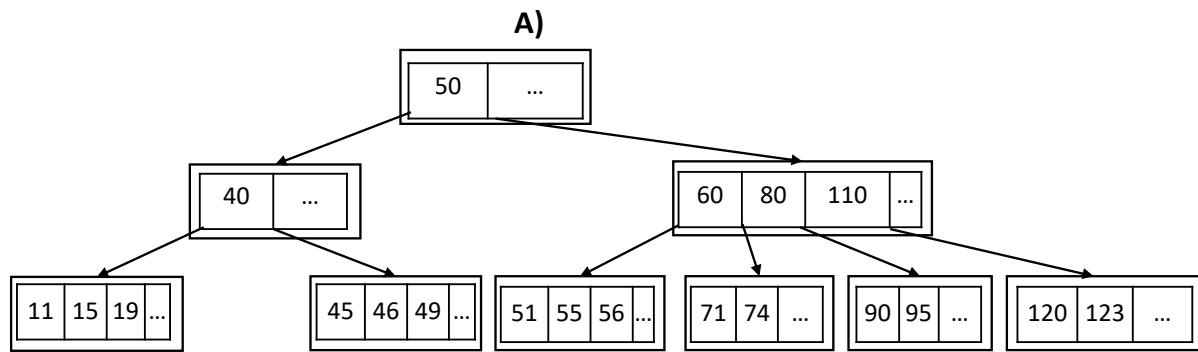
Escludendo la radice, un B-tree con $m = 4$ può avere da 1 a 3 chiavi, con $m = 5$ da 2 a 4 chiavi, con $m = 6$ da 3 a 5 chiavi.

A ha da 2 a 4 chiavi per nodo, quindi è legale solo con $m = 5$. B ha da 1 a 3 chiavi per nodo, quindi è legale solo con $m = 4$.

Quindi non esiste nessun valore di m che renda contemporaneamente legali entrambi i B-tree.

12. Esonero Teoria B+-Tree V2

Elencare, se esistono, tutti i valori di m che rendono i seguenti B-tree entrambi legali, motivando la risposta.



Soluzione.

Escludendo la radice, un B-tree con $m = 4$ può avere da 1 a 3 chiavi, con $m = 5$ da 2 a 4 chiavi, con $m = 6$ da 3 a 5 chiavi.

A ha da 1 a 3 chiavi per nodo, quindi è legale solo con $m = 4$. B ha da 2 a 4 chiavi per nodo, quindi è legale solo con $m = 5$.

Quindi non esiste nessun valore di m che renda contemporaneamente legali entrambi i B-tree.

13. Esonero Teoria Log V1

Si consideri una memoria stabile contenente un unico file di log con il seguente contenuto in seguito a un evento che ha causato il danneggiamento fisico della periferica di storage della memoria standard:

```

<T1, START>;
<T1, BS(t1[A], 1), AS(t1[A], 3)>;
<T2, START>;
<T2, BS(t2[B], 10), AS(t2[B], 2)>;
<T2, ABORT>
<T3, START>;
<T1, COMMIT>;
– CHECKPOINT –
<T3, BS(t3[C], 13), AS(t3[C], 17)>;
<T3, COMMIT>

```

crash!

Indicare, con il massimo dettaglio, le operazioni da effettuare per ripristinare la base di dati.

Soluzione. Si tratta di una situazione di ripresa a freddo, dovuta ad un guasto di un dispositivo. Bisogna quindi effettuare un'operazione di restore della base di dati, copiando il file di dump nel

nuovo dispositivo. Successivamente, si esegue un ripristino a caldo, eseguendo una REDO(T1,T3). Il checkpoint viene ovviamente ignorato.

14. Esonero Teoria Log V2

Si consideri una memoria stabile contenente un unico file di log con il seguente contenuto in seguito a un evento che ha causato il danneggiamento fisico della periferica di storage della memoria standard:

$\langle T1, START \rangle;$
 $\langle T1, BS(t1[B], 9), AS(t1[B], 11) \rangle;$
 $\langle T2, START \rangle;$
 $\langle T2, BS(t2[A], 11), AS(t2[A], 19) \rangle;$
 $\langle T3, START \rangle;$
 $\langle T2, ABORT \rangle$
 $\langle T1, COMMIT \rangle;$
– CHECKPOINT –
 $\langle T3, BS(t3[A], 19), AS(t3[A], 23) \rangle;$
 $\langle T3, COMMIT \rangle$

crash!

Indicare, con il massimo dettaglio, le operazioni da effettuare per ripristinare la base di dati.

Soluzione. Si tratta di un guasto “hard” dovuta ad un guasto di un dispositivo di storage per cui bisogna effettuare una ripresa a freddo. Bisogna quindi ripristinare i dati a partire dai backup e effettuare un ripristino a caldo, eseguendo una redo delle transazioni committed, cioè REDO(T1,T3). Il checkpoint viene ovviamente ignorato.