

1. Esonero SQL V1

Le seguenti relazioni definiscono una base di dati “AllYouCanFit” per gestire esercizi della palestra. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

UTENTE(Nome, Cognome, Età, Sesso)

VOCESCHEDA(Nome, Cognome, Ordine, Esercizio, NumRipetizioni, Durata)

ESERCIZIO(Nome, Tipo, Attrezzo)

Vincoli di integrità referenziale:

VOCESCHEDA(Nome, Cognome) referencia UTENTE(Nome, Cognome),

VOCESCHEDA(Esercizio) referencia ESERCIZIO(Nome).

Significato delle relazioni e degli attributi: *VOCESCHEDA* contiene le voci della scheda dell'utente con i vari esercizi, *Ordine* indica l'ordine in cui l'esercizio deve essere eseguito (1, 2, 3, ...) nella scheda, *Tipo* può assumere i valori *aerobico*, *posturale*, *tonicità*; *Attrezzo* può assumere i valori *cyclette*, *manubrio*, *elastico* ecc. e può essere NULL. Le rimanenti relazioni e attributi sono autoesplicativi.

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, i tipi di esercizi fatti da donne che hanno tra 20 e 30 anni.

(B - Media complessità) Trovare gli utenti e la durata media degli esercizi aerobici nella loro scheda per gli utenti la cui durata complessiva degli esercizi aerobici è superiore a 60 minuti.

(C - Alta complessità) Trovare l'utente che, nell'ultimo esercizio senza attrezzi della propria scheda, ha fatto più ripetizioni.

Soluzioni.

A.

```
select distinct e.tipo
from utente u join vocescheda s on u.nome=s.nome and u.cognome=s.cognome
join esercizio e on s.esercizio=e.nome
where sesso='femmina' and età>=20 and età<=30;
```

B.

```
select s1.nome, s1.cognome, avg(s1.durata)
from vocescheda s1 join esercizio e on s1.esercizio=e.nome
where s1.tipo='aerobico'
group by s1.nome, s1.cognome
having sum(s1.durata)>60;
```

C.

```
with UltimoEsercizio as ( select nome,cognome,numripetizioni
from vocescheda
where (nome,cognome,ordine)=(
select nome,cognome,max(ordine)
from vocescheda s join esercizio e on s.esercizio=e.nome
where attrezzo is null
group by nome,cognome))
select nome, cognome from UltimoEsercizio where numripetizioni=( select max(numripetizioni) from
UltimoEsercizio);
```

2. Esonero SQL V2

Le seguenti relazioni definiscono una base di dati “AllYouCanFit” per gestire esercizi della palestra. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

UTENTE(Nome, Cognome, Età, Sesso)

VOCESCHEDA(Nome, Cognome, Ordine, Esercizio, NumRipetizioni, Durata)

ESERCIZIO(Nome, Tipo, Attrezzo)

Vincoli di integrità referenziale:

VOCESCHEDA(Nome, Cognome) referencia UTENTE(Nome, Cognome),

VOCESCHEDA(Esercizio) referencia ESERCIZIO(Nome).

Significato delle relazioni e degli attributi: *VOCESCHEDA* contiene le voci della scheda dell'utente con i vari esercizi, *Ordine* indica l'ordine in cui l'esercizio deve essere eseguito (1, 2, 3, ...) nella scheda, *Tipo* può assumere i valori *aerobico*, *posturale*, *tonicità*; *Attrezzo* può assumere i valori *cyclette*, *manubrio*, *elastico* ecc. e può essere NULL. Le rimanenti relazioni e attributi sono autoesplicativi.

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

(A - Bassa complessità) Elencare, in ordine alfabetico decrescente e senza duplicati, gli attrezzi usati da uomini nei primi tre esercizi delle loro schede.

(B - Media complessità) Considerando gli attrezzi che, negli esercizi di tipo posturale, non hanno più di 20 ripetizioni, mostrare il nome dell'attrezzo e la durata media degli esercizi.

(C - Alta complessità) Trovare l'attrezzo usato più a lungo (cioè, la cui somma delle durate di utilizzo è massima) negli esercizi di tipo aerobico presenti in schede in cui era l'unico attrezzo usato.

Soluzioni.

A.

```
select distinct e.attrezzo
from utente u join vocescheda s on u.nome=s.nome and u.cognome=s.cognome
join esercizio e on s.esercizio=e.nome
where e.attrezzo is not null and sesso='maschio' and ordine<=3
order by e.attrezzo desc;
```

B.

```
select e.attrezzo, avg(e.durata)
from vocescheda s join e esercizio on s.esercizio=e.nome
where e.tipo='posturale' and e.attrezzo is not null
group by e.attrezzo
having max(s.numripetizioni)<=20;
```

C.

```
with UnicoAttrezzoAerobico as (
select e.attrezzo, sum(s.durata) as SommaDurata
from vocescheda s join esercizio e on s.esercizio=e.nome
where e.attrezzo is not null and e.tipo='aerobico'
and not exists (
select *
from vocescheda s1 join esercizio e1 on s1.esercizio=e1.nome
where s1.nome=s.nome and s1.cognome=s.cognome and e1.attrezzo<>e.attrezzo)
group by e.attrezzo)
select attrezzo
```

```

from UnicoAttrezzoAerobico
where SommaDurata=(
select max(SommaDurata)
from UnicoAttrezzoAerobico);

```

3. Esonero Algebra V1

Le seguenti relazioni definiscono una base di dati “AllYouCanFit” per gestire esercizi della palestra. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

UTENTE(Nome, Cognome, Età, Sesso)

VOCESCHEDA(Nome, Cognome, Ordine, Esercizio, NumRipetizioni, Durata)

ESERCIZIO(Nome, Tipo, Attrezzo)

Vincoli di integrità referenziale:

VOCESCHEDA(Nome, Cognome) referencia UTENTE(Nome, Cognome),

VOCESCHEDA(Esercizio) referencia ESERCIZIO(Nome).

Significato delle relazioni e degli attributi: *VOCESCHEDA* contiene le voci della scheda dell’utente con i vari esercizi, *Ordine* indica l’ordine in cui l’esercizio deve essere eseguito (1, 2, 3, ...) nella scheda, *Tipo* può assumere i valori *aerobico*, *posturale*, *tonicità*; *Attrezzo* può assumere i valori *cyclette*, *manubrio*, *elastico* ecc. e può essere NULL. Le rimanenti relazioni e attributi sono autoesplicativi.

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) Elencare gli utenti che hanno fatto più di un esercizio aerobico.

(B - Calcolo Relazionale su tuple con dichiarazione di range) Elencare nome, cognome e età degli utenti che hanno fatto solo esercizi aerobici.

Soluzioni.

A. $\pi_{S1.nome, S1.cognome} ($
 $(\rho_{S1 \leftarrow VOCESCHEDA}(vocescheda) \bowtie_{S1.esercizio=E1.nome} \rho_{E1 \leftarrow ESERCIZIO}(\sigma_{tipo='aerobico'}(esercizio)))$
 $\bowtie_{S1.nome=S2.nome \wedge S1.cognome=S2.cognome \wedge S1.esercizio <> S2.esercizio}$
 $(\rho_{S2 \leftarrow VOCESCHEDA}(vocescheda) \bowtie_{S2.esercizio=E2.nome} \rho_{E2 \leftarrow ESERCIZIO}(\sigma_{tipo='aerobico'}(esercizio))))$

B. $\{u.nome, u.cognome, u.eta | u(utente) | \forall e(esercizio) (\exists s(vocescheda) ((s.esercizio = e.nome \wedge e.nome = u.nome \wedge e.cognome = u.cognome) \Rightarrow e.tipo = 'aerobico'))\}$

4. Esonero Algebra V2

Le seguenti relazioni definiscono una base di dati “AllYouCanFit” per gestire esercizi della palestra. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

UTENTE(Nome, Cognome, Età, Sesso)

VOCESCHEDA(Nome, Cognome, Ordine, Esercizio, NumRipetizioni, Durata)

ESERCIZIO(Nome, Tipo, Attrezzo)

Vincoli di integrità referenziale:

VOCESCHEDA(Nome, Cognome) referencia UTENTE(Nome, Cognome),

VOCESCHEDA(Esercizio) referencia ESERCIZIO(Nome).

Significato delle relazioni e degli attributi: *VOCESCHEDA* contiene le voci della scheda dell’utente con i vari esercizi, *Ordine* indica l’ordine in cui l’esercizio deve essere eseguito (1, 2, 3, ...) nella scheda, *Tipo* può assumere i valori *aerobico*, *posturale*, *tonicità*; *Attrezzo* può assumere i valori *cyclette*, *manubrio*, *elastico* ecc. e può essere NULL. Le rimanenti relazioni e attributi sono autoesplicativi.

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) Elencare gli attrezzi che non sono mai usati da utenti con più di 60 anni.

(B - Calcolo Relazionale su tuple con dichiarazione di range) Elencare nome, cognome e età degli utenti che hanno fatto solo esercizi senza attrezzi.

Soluzioni.

A. $\pi_{attrezzo}(\sigma_{attrezzo \text{ IS NOT NULL } (esercizio)})$

—

$\pi_{attrezzo}(\sigma_{utente.eta > 60}((utente \bowtie_vocesc\text{hed}\text{a}) \bowtie_{vocesc\text{hed}\text{a}.esercizio=esercizio.nome} esercizio))$

B. $\{u.nome, u.cognome, u.eta | u(utente) | \forall e(esercizio) (\exists s(vocesc\text{hed}\text{a}) ((s.esercizio = e.nome \wedge e.nome = u.nome \wedge e.cognome = u.cognome) \Rightarrow e.attrezzo \text{ IS NULL}))\}$

5. Esonero Ottimizzazione V1

Le seguenti relazioni definiscono una base di dati “AllYouCanFit” per gestire esercizi della palestra. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

UTENTE(Nome, Cognome, Età, Sesso)

VOCESCHEDA(Nome, Cognome, Ordine, Esercizio, NumRipetizioni, Durata)

ESERCIZIO(Nome, Tipo, Attrezzo)

Vincoli di integrità referenziale:

VOCESCHEDA(Nome, Cognome) referencia UTENTE(Nome, Cognome),

VOCESCHEDA(Esercizio) referencia ESERCIZIO(Nome).

Significato delle relazioni e degli attributi: *VOCESCHEDA* contiene le voci della scheda dell'utente con i vari esercizi, *Ordine* indica l'ordine in cui l'esercizio deve essere eseguito (1, 2, 3, ...) nella scheda, *Tipo* può assumere i valori *aerobico*, *posturale*, *tonicità*; *Attrezzo* può assumere i valori *cyclette*, *manubrio*, *elastico* ecc. e può essere NULL. Le rimanenti relazioni e attributi sono autoesplicativi.

Data la seguente query:

$\sigma_{NumRipetizioni \geq 200 \wedge NumRipetizioni < 300 \wedge attrezzo = 'elastico'}(vocesc\text{hed}\text{a} \bowtie_{esercizio=esercizio.nome} esercizio)$

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(utente) = 100

CARD(vocesc\text{hed}\text{a}) = 1 000

CARD(esercizio) = VAL(Esercizio, vocesc\text{hed}\text{a}) = 20

MIN(NumRipetizioni, vocesc\text{hed}\text{a}) = 0

MAX(NumRipetizioni, vocesc\text{hed}\text{a}) = 1 000

VAL(Attrezzo, esercizio) = 10

Soluzioni.

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

$\sigma_{NumRipetizioni \geq 200 \wedge NumRipetizioni < 300}(vocesc\text{hed}\text{a}) \bowtie_{esercizio=esercizio.nome} \sigma_{attrezzo = 'elastico'}(esercizio)$

Prima dell'ottimizzazione:

- Costo $r_1 = vocesc\text{hed}\text{a} \bowtie_{esercizio=esercizio.nome} esercizio$: $1\,000 \cdot 20 = 20\,000$.

- Cardinalità di $|r_1| = CARD(vocescheda) = 100$ (equijoin attraverso la chiave esterna)
- Costo della selezione: $|r_1|$
- Costo totale = $20\,000 + 100 \approx 20\,000$

Dopo l'ottimizzazione:

- Costo $\sigma_1 = \sigma_{NumRipetizioni \geq 200 \wedge NumRipetizioni < 300}(vocescheda) = 1\,000$
- Tuple prodotte dalla selezione $|\sigma_1| = \frac{300-200}{MAX(NumRipetizioni, vocescheda) - MIN(NumRipetizioni, vocescheda)} \cdot CARD(vocescheda) = \frac{300-200}{1000-0} \cdot 1\,000 = \frac{1}{10} \cdot 1\,000 = 100$
- Costo $\sigma_2 = \sigma_{attrezzo='elastico'}(esercizio) = 20$
- Tuple prodotte dalla selezione $|\sigma_2| = \frac{1}{VAL(Attrezzo, esercizio)} \times CARD(esercizio) = \frac{1}{10} \times 20 = 2$
- Costo join $r = \sigma_{NumRipetizioni \geq 200 \wedge NumRipetizioni < 300}(vocescheda) \bowtie_{esercizio=esercizio.nome} \sigma_{attrezzo='elastico'}(esercizio)$
 $|\sigma_1| \cdot |\sigma_2| = 100 \cdot 2 = 200.$
- Costo totale = $1\,000 + 20 + 200 \approx 1\,220$

6. Esonero Ottimizzazione V2

Le seguenti relazioni definiscono una base di dati “AllYouCanFit” per gestire esercizi della palestra. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

UTENTE(Nome, Cognome, Età, Sesso)

VOCESCHEDA(Nome, Cognome, Ordine, Esercizio, NumRipetizioni, Durata)

ESERCIZIO(Nome, Tipo, Attrezzo)

Vincoli di integrità referenziale:

VOCESCHEDA(Nome, Cognome) referencia UTENTE(Nome, Cognome),

VOCESCHEDA(Esercizio) referencia ESERCIZIO(Nome).

Significato delle relazioni e degli attributi: *VOCESCHEDA* contiene le voci della scheda dell'utente con i vari esercizi, *Ordine* indica l'ordine in cui l'esercizio deve essere eseguito (1, 2, 3, ...) nella scheda, *Tipo* può assumere i valori *aerobico*, *posturale*, *tonicità*; *Attrezzo* può assumere i valori *cyclette*, *manubrio*, *elastico* ecc. e può essere NULL. Le rimanenti relazioni e attributi sono autoesplicativi.

Data la seguente query:

$\sigma_{Eta \geq 20 \wedge Eta < 30 \wedge esercizio='plank'}(vocescheda \bowtie_{vocescheda.nome=utente.nome \wedge vocescheda.cognome=utente.cognome} utente)$

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 esiste una scheda per ogni utente e che:

$CARD(utente) = 100$

$CARD(vocescheda) = 1\,000$

$VAL(esercizio, vocescheda) = 20$

$MIN(Età, utente) = 15$

$MAX(Età, utente) = 65$

Soluzioni.

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

$\sigma_{esercizio='plank'}(vocescheda) \bowtie_{vocescheda.nome=utente.nome \wedge vocescheda.cognome=utente.cognome} \sigma_{Eta \geq 20 \wedge Eta < 30}(utente)$

Prima dell'ottimizzazione:

- Costo $r_1 = \text{vocescheda} \bowtie_{\text{vocescheda.nome}=\text{utente.nome} \wedge \text{vocescheda.cognome}=\text{utente.cognome}} \text{utente}$: $1\,000 \cdot 100 = 100\,000$.
- Cardinalità di $|r_1| = \text{CARD}(\text{vocescheda}) = 1\,000$ (equijoin attraverso la chiave esterna)
- Costo della selezione: $|r_1|$
- Costo totale = $100\,000 + 1\,000 \approx 100\,000$

Dopo l'ottimizzazione:

- Costo $\sigma_1 = \sigma_{\text{esercizio}='plank'}(\text{vocescheda}) = 1\,000$
- Tuple prodotte dalla selezione $|\sigma_1| = \frac{1}{\text{VAL}(\text{Esercizio}, \text{vocescheda})} \cdot \text{CARD}(\text{vocescheda}) = \frac{1}{20} \cdot 1000 = 50$
- Costo $\sigma_2 = \sigma_{\text{Eta} \geq 20 \wedge \text{Eta} < 30}(\text{utente}) = 100$
- Tuple prodotte dalla selezione $|\sigma_2| = \frac{30-20}{\text{MAX}(\text{Eta}, \text{utente}) - \text{MIN}(\text{Eta}, \text{utente})} \cdot \text{CARD}(\text{utente}) = \frac{30-20}{65-15} \cdot 100 = \frac{1}{5} \cdot 100 = 20$
- Costo join $r = \sigma_{\text{esercizio}='plank'}(\text{vocescheda}) \bowtie_{\text{vocescheda.nome}=\text{utente.nome} \wedge \text{vocescheda.cognome}=\text{utente.cognome}} \sigma_{\text{Eta} \geq 20 \wedge \text{Eta} < 30}(\text{utente})$: $|\sigma_1| \cdot |\sigma_2| = 50 \cdot 20 = 1\,000$.
- Costo totale = $1\,000 + 100 + 1\,000 \approx 1\,000$

7. Esonero Teoria 3NF V1

Dati:

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

$F = \{ABC \rightarrow DEF, BC \rightarrow FG, F \rightarrow G\}$

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 ABH, perché A, B e H non compaiono a destra in nessuna dipendenza funzionale (e quindi devono fare parte della chiave), ma $ABH^+ = \{A, B, H\}$, e quindi non è chiave; $ABCH^+ = \{A, B, C, H | D, E, F | G\}$ è chiave, $ABDH^+ = \{A, B, D, H\}$ non è chiave, $ABEH^+ = \{A, B, E, H\}$ non è chiave, $ABFH^+ = \{A, B, F, H | G\}$ non è chiave, $ABGH^+ = \{A, B, G, H\}$ non è chiave.

La relazione non è in 3NF, infatti nessuna delle tre dipendenze funzionali è superchiave, riflessiva o attributi primi.

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

$F' = \{ABC \rightarrow D, ABC \rightarrow E, ABC \rightarrow F, BC \rightarrow F, BC \rightarrow G, F \rightarrow G\}$.

A è estraneo in $ABC \rightarrow F$ perché F si può ottenere da BC tramite $BC \rightarrow F$, quindi:

$F' = \{ABC \rightarrow D, ABC \rightarrow E, BC \rightarrow F, BC \rightarrow F, BC \rightarrow G, F \rightarrow G\} = \{ABC \rightarrow D, ABC \rightarrow E, BC \rightarrow F, BC \rightarrow G, F \rightarrow G\}$.

$BC \rightarrow G$ è ridondante perché $BC^+_{\{ABC \rightarrow D, ABC \rightarrow E, BC \rightarrow F, F \rightarrow G\}} = BC | F | G \dots$, quindi

$F' = \{ABC \rightarrow D, ABC \rightarrow E, BC \rightarrow F, F \rightarrow G\}$

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

$R1(A, B, C, D, E)$

$R2(B, C, F)$

$R3(F, G)$

cui bisogna aggiungere anche una relazione $R4(\underline{A,B,C,H})$ contenente la chiave della relazione R.

Il risultato è BCNF perché ogni relazione ha la dipendenza funzionale (implicita) di chiave primaria, cioè di tipo superchiave.

8. Esonero Teoria 3NF V2

Dati:

$R(M, N, O, P, Q, R, S, T)$ e

$F = \{RST \rightarrow MNO, ST \rightarrow OP, O \rightarrow P\}$

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 QRST, perché Q, R, S e T non compaiono a destra in nessuna dipendenza funzionale (e quindi devono fare parte della chiave), inoltre $QRST^+ = \{Q, R, S, T | M, N, O | P\}$ è superchiave e si dimostra che è anche minimale. Non ci sono altre chiavi.

La relazione non è in 3NF, infatti nessuna delle tre dipendenze funzionali è superchiave, riflessiva o attributi primi.

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

$F' = \{RST \rightarrow M, RST \rightarrow N, RST \rightarrow O, ST \rightarrow O, ST \rightarrow P, O \rightarrow P\}$.

M è estraneo in $RST \rightarrow O$ perché O si può ottenere da ST tramite $ST \rightarrow O$, quindi:

$F' = \{RST \rightarrow M, RST \rightarrow N, ST \rightarrow O, ST \rightarrow P, O \rightarrow P\} =$
 $\{RST \rightarrow M, RST \rightarrow N, ST \rightarrow O, ST \rightarrow P, O \rightarrow P\}$.

$ST \rightarrow P$ è ridondante perché deducibile da $ST \rightarrow O$ e $O \rightarrow P$ per transitività.

$F' = \{RST \rightarrow M, RST \rightarrow N, ST \rightarrow O, O \rightarrow P\}$

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

$R1(\underline{R,S,T}, M, N)$

$R2(\underline{S,T}, O)$

$R3(\underline{O}, P)$

cui bisogna aggiungere anche una relazione $R4(\underline{Q,R,S,T})$ contenente la chiave della relazione R.

Il risultato è BCNF perché ogni relazione ha la dipendenza funzionale (implicita) di chiave primaria, cioè di tipo superchiave.

9. Esonero Teoria Armstrong V1

Dimostrare che si può derivare F' da F usando le regole di Armstrong, dove $F = \{AB \rightarrow CD, A \rightarrow DE, F \rightarrow DG\}$ e $F' = \{AB \rightarrow CDE, AB \rightarrow B, A \rightarrow DE, F \rightarrow D, F \rightarrow G\}$.

Soluzione.

Partendo da $F = \{AB \rightarrow CD, A \rightarrow DE, F \rightarrow DG\}$ applicando la regola del prodotto otteniamo

$\{AB \rightarrow CDE, A \rightarrow DE, F \rightarrow DG\}$

Per riflessività otteniamo

$\{AB \rightarrow CDE, AB \rightarrow B, A \rightarrow DE, F \rightarrow DG\}$

E infine per decomposizione abbiamo

$\{AB \rightarrow CDE, AB \rightarrow B, A \rightarrow DE, F \rightarrow D, F \rightarrow G\}$.

10. Esonero Teoria Armstrong V2

Dimostrare che si può derivare F' da F usando le regole di Armstrong (indicando quali), dove $F = \{AB \rightarrow CD, A \rightarrow DE, A \rightarrow DG\}$ e $F' = \{AB \rightarrow CDE, A \rightarrow DG, A \rightarrow EG\}$.

Soluzione.

Partendo da $F = \{AB \rightarrow CD, A \rightarrow DE, A \rightarrow DG\}$ applicando la regola del prodotto otteniamo $\{AB \rightarrow CDE, A \rightarrow DE, A \rightarrow DG\}$

Per l'assioma dell'unione otteniamo

$\{AB \rightarrow CDE, A \rightarrow DEG\}$

E infine per decomposizione abbiamo

$\{AB \rightarrow CDE, A \rightarrow DG, A \rightarrow EG\}$.

11. Esonero Teoria B+-Tree V1

A. Si discuta se è opportuno usare un indice, se sì quale struttura dati sarebbe opportuno usare e caratterizzare il tipo di indice per l'attributo NumRipetizioni della relazione VOCESCHEDA tenendo conto delle seguenti informazioni.

VOCESCHEDA(Nome, Cognome, Ordine, Esercizio, NumRipetizioni, Durata)

con

CARD(vocescheda) = 1 000

MIN(NumRipetizioni, vocescheda) = 0

MAX(NumRipetizioni, vocescheda) = 1 000

su query del tipo $\sigma_{NumRipetizioni \geq 100 \wedge NumRipetizioni < 130}(vocescheda)$

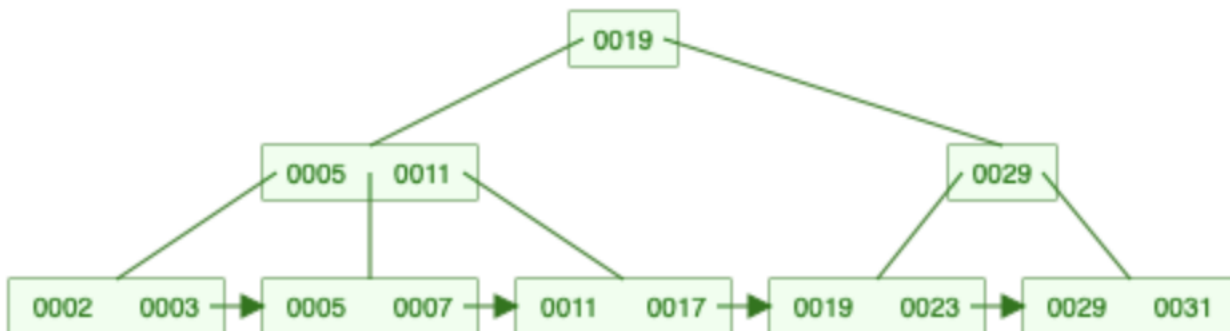
B. Rappresentare due possibili B+-tree contenenti le chiavi (2, 3, 5, 7, 11, 17, 19, 23, 29, 31) per i casi con $m=4$ e $m=7$. Non si richiede di simulare le singole operazioni di inserimento, ma di mostrare un possibile B+-tree con le caratteristiche indicate.

Soluzione.

A. L'attributo NumRipetizioni presenta valori ripetuti, quindi l'eventuale indice sarà di tipo clusterizzato o secondario a seconda che l'ordinamento fisico sia sull'attributo. L'attributo presenta un alto grado di selettività, quindi può essere opportuno definire un indice. La query di esempio effettua una selezione su un range di valori, che può essere eseguito in modo efficiente da un B+-tree.

B. Caso $m=4$. Ogni nodo ha al massimo $m - 1 = 3$ chiavi, la radice ha almeno 1 chiave, gli altri nodi hanno almeno $\lceil \frac{m}{2} - 1 \rceil = 1$ chiave. Le foglie devono contenere tutte le chiavi e essere linkate.

Un possibile B+-tree è in figura.



Caso $m=7$. Ogni nodo ha al massimo $m - 1 = 6$ chiavi, la radice ha almeno 1 chiave, gli altri nodi hanno almeno $\lceil \frac{m}{2} - 1 \rceil = 3$ chiavi.



12. Esonero Teoria B+-Tree V2

A. Si discuta se è opportuno usare un indice, se sì quale struttura dati sarebbe opportuno usare e caratterizzare il tipo di indice per l'attributo Durata della relazione VOCESCHEDA tenendo conto delle seguenti informazioni.

VOCESCHEDA(Nome, Cognome, Ordine, Esercizio, NumRipetizioni, Durata)

con

CARD(vocescheda) = 1 000

MIN(Durata,vocescheda) = 30

MAX(Durata,vocescheda) = 180

su query del tipo $\sigma_{Durata \geq 60 \wedge Durata < 120}(vocescheda)$

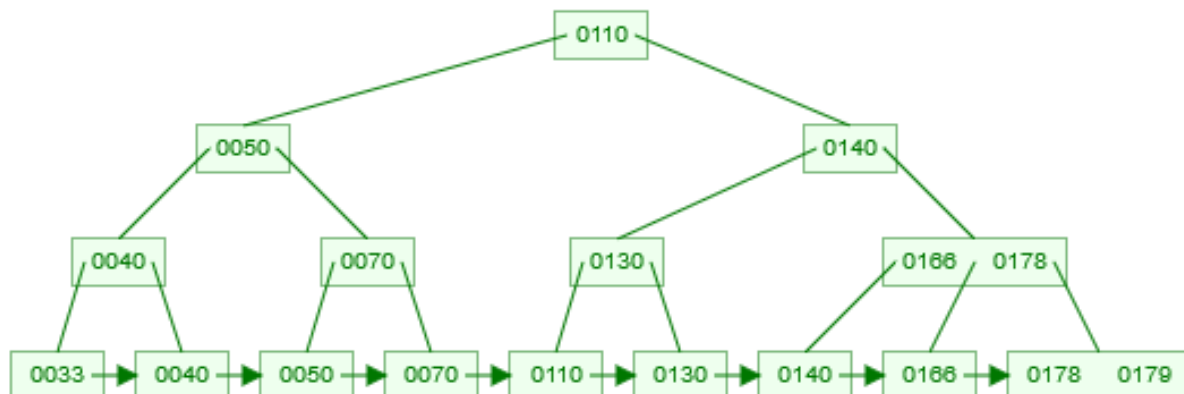
B. Rappresentare due possibili B+-tree contenenti le chiavi (33, 40, 50, 70, 110, 130, 140, 166, 178, 179) per i casi con $m=3$ e $m=6$. Non si richiede di simulare le singole operazioni di inserimento, ma di mostrare un possibile B+-tree con le caratteristiche indicate.

Soluzione.

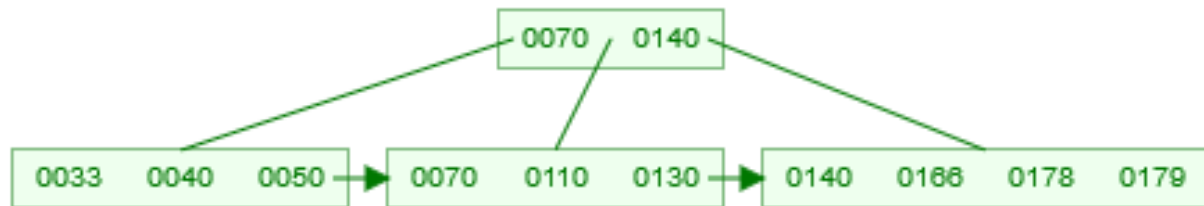
A. L'attributo Durata presenta valori ripetuti, quindi l'eventuale indice sarà di tipo clusterizzato o secondario a seconda che l'ordinamento fisico sia sull'attributo. L'attributo presenta un alto grado di selettività, quindi può essere opportuno definire un indice. La query di esempio effettua una selezione su un range di valori, che può essere eseguito in modo efficiente da un B+-tree.

B. Caso $m=3$. Ogni nodo ha al massimo $m - 1 = 2$ chiavi, la radice ha almeno 1 chiave, gli altri nodi hanno almeno $\lceil \frac{m}{2} - 1 \rceil = 1$ chiave. Le foglie devono contenere tutte le chiavi e essere linkate.

Un possibile B+-tree è in figura.



Caso $m=6$. Ogni nodo ha al massimo $m - 1 = 5$ chiavi, la radice ha almeno 1 chiave, gli altri nodi hanno almeno $\lceil \frac{m}{2} - 1 \rceil = 2$ chiavi.



13. Esonero Teoria 2PL V1

Considerare la seguente storia interfogliata

$$S = r1(x), r2(y), w2(y), r2(x), w1(x), w1(y)$$

La storia S è compatibile con il protocollo 2PL? Giustificare la risposta.

Soluzione.

La storia S è compatibile con 2PL perché esiste almeno un modo di aggiungere i lock (lock condivisi o esclusivi per le letture e lock esclusivi per le scritture) rispettando il 2PL (cioè nessuna transazione acquisisce lock dopo avere rilasciato un lock precedentemente acquisito). Un esempio di aggiunta dei lock 2PL è il seguente:

$$S = LS1(x), r1(x), LS2(y), r2(y), LX2(y), w2(y), LS2(x), r2(x), UL2(x), UL2(y), LX1(x), w1(x), LX1(y), w1(y), UL1(x), UL1(y)$$

14. Esonero Teoria 2PL V2

Considerare la seguente storia interfogliata

$$S = r1(x), r2(x), w2(y), w2(x), w1(x), w1(y)$$

La storia S è compatibile con il protocollo 2PL? Giustificare la risposta.

Soluzione.

La storia S **non è compatibile** con 2PL perché, prima di richiedere il lock esclusiva per scrivere x , la transazione T1 deve eseguire un unlock per permettere alla transazione T2 di scrivere x . Questo viola il protocollo 2PL, che richiede che, dopo il primo unlock, una transazione non possa più richiedere un lock.