

## Esonero 15 12 2020

### 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, Eta, 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 maschi che non hanno nessuna vocale nel cognome.

**(B - Media complessità)** Tra gli utenti che hanno tra 50 e 60 anni e hanno fatto in media tra 20 e 30 ripetizioni, elencare gli utenti e la durata massima dei loro esercizi. N.B.: non usare sottoquery.

**(C - Alta complessità)** Tra gli utenti che hanno fatto solo esercizi con più di 30 ripetizioni (tranne che con la cyclette, per cui il numero di ripetizioni può essere qualunque), trovare l'utente più vecchio e il numero totale delle ripetizioni dei suoi esercizi. Si presti attenzione al fatto che non tutti gli esercizi usano attrezzi e che possono esserci più utenti con la stessa età.

### Soluzioni.

#### A.

```
select distinct e.tipo
from utente u join vocescheda v on u.nome=v.nome and u.cognome=v.cognome
join esercizio e on v.esercizio=e.nome
where sesso='maschio' and not(u.cognome like '%a%' or u.cognome like '%A%' or u.cognome like '%e%' or u.cognome like '%E%' or u.cognome like '%i%' or u.cognome like '%I%' or u.cognome like '%o%' or u.cognome like '%O%' or u.cognome like '%u%' or u.cognome like '%U%');
```

#### B.

```
select v.nome, v.cognome, max(v.durata)
from utente u join vocescheda v on u.nome=v.nome and u.cognome=v.cognome
where u.eta>=50 and u.eta<=60
group by v.nome, v.cognome
having avg(v.numripetizioni)>=20 and avg(v.numripetizioni)<=30;
```

#### C.

```
with utenti30rip as ( select nome,cognome,eta,numripetizioni
from utente u join vocescheda v1 on u.nome=v1.nome and u.cognome=v.cognome
where not exists (
select *
from vocescheda v2 join esercizio e on v2.esercizio=e.nome
where v2.nome=u.nome and v2.cognome=u.cognome and
(attrezzo<>'cyclette' or attrezzo is null) and numripetizioni<=30))
select nome, cognome, sum(numripetizioni)
from utenti30rip
```

```

where eta=(
select max(eta)
from utenti30rip)
group by nome, cognome;

```

## 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, Eta, 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, gli attrezzi utilizzati da femmine il cui cognome inizia per “Azz” e finisce per “ina” ma non contiene “oli”.

**(B - Media complessità)** Tra gli utenti che hanno meno di 30 anni o più di 50 anni e hanno fatto esercizi con durata media compresa tra 5 e 10 minuti, elencare utenti e il minimo numero di ripetizioni tra i loro esercizi. N.B.: non usare sottoquery.

**(C - Alta complessità)** Tra gli utenti che hanno fatto solo esercizi della durata di meno di 7 minuti (tranne che con il TRX, per cui la durata può essere qualunque), trovare l’utente più giovane e la durata media dei suoi esercizi. Si presti attenzione al fatto che non tutti gli esercizi usano attrezzi e che possono esserci più utenti con la stessa età.

### Soluzioni.

#### A.

```

select distinct e.attrezzo
from utente u join vocescheda v on (u.nome=v.nome and u.cognome=v.cognome)
join esercizio e on v.esercizio=e.nome
where sesso='femmina' and cognome like 'Azz%ina' and not(cognome like '%oli%');

```

#### B.

```

select v.nome, v.cognome, min(v.numripetizioni)
from utente u join vocescheda v on u.nome=v.nome and u.cognome=v.cognome
where u.eta<30 or u.eta>50
group by v.nome, v.cognome
having avg(v.durata)>=5 and avg(v.durata)<=10;

```

#### C.

```

with utenti7min as ( select nome,cognome,eta,durata
from utente u join vocescheda v1 on u.nome=v1.nome and u.cognome=v1.cognome
where not exists (
select *
from vocescheda v2 join esercizio e on v2.esercizio=e.nome
where v2.nome=u.nome and v2.cognome=u.cognome and
(attrezzo<>'TRX' or attrezzo is null) and durata>=7))

```

```

select nome, cognome, avg(durata)
from utenti7min
where eta=(
select min(eta)
from utenti7min)
group by nome, cognome;

```

### 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, Eta, 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 nome, cognome e età degli utenti che hanno fatto l'esercizio che è durato di più. Scrivere la query sotto forma di albero sintattico.

**(B - Calcolo Relazionale su tuple con dichiarazione di range)** Elencare nome, cognome e sesso degli utenti che hanno fatto solo esercizi con il manubrio e con più di 30 ripetizioni.

**Soluzioni.**

**A.**  $\pi_{utente.nome, utente.cognome, eta}(utente \bowtie_{utente.nome=V1.nome \wedge utente.cognome=V1.cognome} (\rho_{V1 \leftarrow VOCESCHEDA}(vocescheda) - \pi_{V1.*}(\rho_{V1 \leftarrow VOCESCHEDA}(vocescheda) \bowtie_{V1.durata < V2.durata} \rho_{V2 \leftarrow VOCESCHEDA}(vocescheda))))$

**B.**  $\{u.nome, u.cognome, u.sesso | u(utente) | \forall v(vocescheda)((v.nome = u.nome \wedge v.cognome = u.cognome) \Rightarrow (numripetizioni > 30 \wedge \exists e(esercizio)(e.nome = v.esercizio \wedge e.attrezzo = 'manubrio')))\}$

### 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, Eta, 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 il nome dell'attrezzo per cui è stato fatto il maggior numero di ripetizioni in un singolo esercizio. Scrivere la query sotto forma di albero sintattico.

**(B - Calcolo Relazionale su tuple con dichiarazione di range)** Elencare nome e tipo degli esercizi eseguiti solo da utenti con più di 35 anni per più di 10 minuti.

**Soluzioni.**

**A.**  $\pi_{esercizio.attrezzo}(\rho_{V1 \leftarrow VOCESCHEDA}(vocescheda) \bowtie_{esercizio.nome=v1.esercizio} \rho_{V2 \leftarrow VOCESCHEDA}(vocescheda)))$

**B.**  $\{e.nome, e.tipo | e(esercizio) \wedge \forall v(vocescheda)((v.esercizio = e.nome) \Rightarrow (v.durata > 10 \wedge \exists u(utente)(u.nome = v.nome \wedge u.cognome = v.cognome \wedge u.eta > 35)))\}$

## 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, Eta, 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:

$\sigma_{Durata \geq 10 \wedge Durata < 30 \wedge Tipo = 'aerobico'}(esercizio \bowtie_{esercizio.nome=esercizio} vocescheda)$

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) = 1 000

CARD(vocescheda) = 10 000

CARD(esercizio) = VAL(Esercizio,vocescheda) = 20

MIN(Durata,vocescheda) = 0

MAX(Durata,vocescheda) = 100

VAL(Tipo,esercizio) = 5

**Soluzioni.**

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

$\sigma_{Tipo='aerobico'}(esercizio) \bowtie_{esercizio.nome=esercizio} \sigma_{Durata \geq 10 \wedge Durata < 30}(vocescheda)$

Prima dell'ottimizzazione:

- Costo  $r_1 = esercizio \bowtie_{esercizio.nome=esercizio} vocescheda$ :  $20 \cdot 10\,000 = 200\,000$ .
- Cardinalità di  $|r_1| = CARD(vocescheda) = 10\,000$  (equijoin attraverso la chiave esterna)
- Costo della selezione:  $|r_1|$
- Costo totale =  $200\,000 + 10\,000 \approx 200\,000$

Dopo l'ottimizzazione:

- Costo  $\sigma_1 = \sigma_{Tipo='aerobico'}(esercizio) = 20$
- Tuple prodotte dalla selezione  $|\sigma_1| = \frac{1}{VAL(Tipo, esercizio)} \cdot CARD(erercizio) = \frac{1}{5} \cdot 20 = 4$
- Costo  $\sigma_2 = \sigma_{Durata \geq 10 \wedge Durata < 30}(vocescheda) = CARD(vocescheda) = 10\,000$
- Tuple prodotte dalla selezione  $|\sigma_2| = \frac{30-10}{\frac{MAX(Durata, vocescheda) - MIN(Durata, vocescheda)}{100-0}} \cdot CARD(vocescheda) = \frac{30-10}{100-0} \cdot 10\,000 = \frac{20}{100} \cdot 10\,000 = 2\,000$
- Costo join  $r = \sigma_1 \bowtie_{esercizio.nome=esercizio} \sigma_2$ :  $|\sigma_1| \cdot |\sigma_2| = 4 \cdot 2\,000 = 8\,000$ .
- Costo totale =  $20 + 10\,000 + 8\,000 \approx 18\,000$

## 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, Eta, 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:

$\sigma_{NumRipetizioni \geq 7 \wedge NumRipetizioni < 12 \wedge Tipo='posturale'}(esercizio \bowtie_{esercizio.nome=esercizio} vocescheda)$

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:

$CARD(utente) = 1\,000$

$CARD(vocescheda) = 10\,000$

$CARD(erercizio) = VAL(Esercizio, vocescheda) = 30$

$MIN(NumRipetizioni, vocescheda) = 5$

$MAX(NumRipetizioni, vocescheda) = 30$

### Soluzioni.

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

$\sigma_{Tipo='posturale'}(esercizio) \bowtie_{esercizio.nome=esercizio} \sigma_{NumRipetizioni \geq 7 \wedge NumRipetizioni < 12}(vocescheda)$

Prima dell'ottimizzazione:

- Costo  $r_1 = \text{esercizio} \bowtie_{\text{esercizio.nome}=\text{esercizio}} \text{vocescheda}$ :  $30 \cdot 10\,000 = 300\,000$ .
- Cardinalità di  $|r_1| = \text{CARD}(\text{vocescheda}) = 10\,000$  (equi-join attraverso la chiave esterna)
- Costo della selezione =  $|r_1|$
- Costo totale =  $300\,000 + 10\,000 \approx 300\,000$

Dopo l'ottimizzazione:

- Costo  $\sigma_1 = \sigma_{\text{Tipo}='posturale'}(\text{esercizio}) = 30$
- Tuple prodotte dalla selezione  $|\sigma_1| = \frac{1}{\text{VAL}(\text{Tipo}, \text{esercizio})} \cdot \text{CARD}(\text{esercizio}) = \frac{1}{3} \cdot 30 = 10$
- Costo  $\sigma_2 = \sigma_{\text{NumRipetizioni} \geq 7 \wedge \text{NumRipetizioni} < 12}(\text{vocescheda}) = \text{CARD}(\text{vocescheda}) = 10\,000$
- Tuple prodotte dalla selezione  $|\sigma_2| = \frac{12-7}{\text{MAX}(\text{NumRipetizioni}, \text{vocescheda}) - \text{MIN}(\text{NumRipetizioni}, \text{vocescheda})} \cdot \text{CARD}(\text{vocescheda}) = \frac{12-7}{30-5} \cdot 10\,000 = \frac{5}{25} \cdot 10\,000 = 2\,000$
- Costo join  $r = \sigma_1 \bowtie_{\text{esercizio.nome}=\text{esercizio}} \sigma_2$ :  $|\sigma_1| \cdot |\sigma_2| = 10 \cdot 2\,000 = 20\,000$ .
- Costo totale =  $30 + 10\,000 + 20\,000 \approx 30\,000$

## 7. Esonero Teoria 3NF V1

Dati:

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

$F = \{AB \rightarrow BC, CD \rightarrow BC, AC \rightarrow D\}$

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 ed E, perché non compaiono a destra in nessuna dipendenza funzionale.  $ABE^+ = \{A, B, E | C | D\}$ , e quindi è chiave; inoltre  $ACE^+ = \{A, C, E | D | B\}$  è chiave, mentre  $ADE^+ = \{A, D, E\}$  non è chiave.

La relazione non è in 3NF, infatti  $AC \rightarrow D$  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:

$F' = \{AB \rightarrow B, AB \rightarrow C, CD \rightarrow B, CD \rightarrow C, AC \rightarrow D\}$ .

A è estraneo in  $AB \rightarrow B$  perché B ovviamente si può ottenere da se stesso e allo stesso modo D è estraneo in  $CD \rightarrow C$ , quindi:

$F' = \{B \rightarrow B, AB \rightarrow C, CD \rightarrow B, C \rightarrow C, AC \rightarrow D\}$ .

$B \rightarrow B$  e  $C \rightarrow C$  sono ridondanti perché riflessive, quindi la copertura minimale è:

$F' = \{AB \rightarrow C, CD \rightarrow B, AC \rightarrow D\}$ .

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

$R1(A, B, C)$

$R2(C, D, B)$

$R3(A, C, D)$

cui bisogna aggiungere anche una relazione  $R4(A, B, E)$  contenente una 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:

$R(Di, Ez, Ma, Ra, Do, Na)$  e

$F = \{MaRa \rightarrow RaDo, DoNa \rightarrow RaDo, MaDo \rightarrow NaEz\}$

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  $Ma$  e  $Di$ , perché non compaiono a destra in nessuna dipendenza funzionale.  $MaRaDi^+ = \{Ma, Ra, Di | Do | Na, Ez\}$  e quindi è chiave; inoltre  $MaDoDi^+ = \{Ma, Do, Di | Na, Ez | Ra\}$  è chiave, mentre  $MaNaDiE^+ = \{Ma, Na, Di\}$  non è chiave.

La relazione non è in 3NF, infatti  $MaDo \rightarrow NaEz$  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:

$F' = \{MaRa \rightarrow Ra, MaRa \rightarrow Do, DoNa \rightarrow Ra, DoNa \rightarrow Do, MaDo \rightarrow Na, MaDo \rightarrow Ez\}$ .

$Ma$  è estraneo in  $MaRa \rightarrow Ra$  perché  $Ra$  ovviamente si può ottenere da se stesso e allo stesso modo  $Na$  è estraneo in  $DoNa \rightarrow Do$ , quindi:

$F' = \{Ra \rightarrow Ra, MaRa \rightarrow Do, DoNa \rightarrow Ra, Do \rightarrow Do, MaDo \rightarrow Na, MaDo \rightarrow Ez\}$ .

$Ra \rightarrow Ra$  e  $Do \rightarrow Do$  sono ridondanti perché riflessive, quindi la copertura minimale è:

$F' = \{MaRa \rightarrow Do, DoNa \rightarrow Ra, MaDo \rightarrow Na, MaDo \rightarrow Ez\}$ .

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

$R1(\underline{Ma}, \underline{Ra}, Do)$

$R2(\underline{Do}, Na, Ra)$

$R3(\underline{Ma}, \underline{Do}, Na, Ez)$

cui bisogna aggiungere anche una relazione  $R4(\underline{Ma}, \underline{Ra}, \underline{Di})$  contenente una 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

Derivare  $F'$  da  $F$  usando le regole di Armstrong, dove  $F = \{AB \rightarrow CD, A \rightarrow DE, F \rightarrow DG\}$  e  $F' = \{AB \rightarrow CDE, AB \rightarrow B, F \rightarrow D, F \rightarrow G\}$ . Si può concludere che  $F$  ed  $F'$  sono equivalenti?

**Soluzione.**

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

Per riflessività otteniamo

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

E infine per decomposizione abbiamo

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

Si noti che i due insiemi non sono equivalenti perché la d.f.  $A \rightarrow CD$  in  $F$  non è deducibile da  $F'$ .

## 10. Esonero Teoria Armstrong V2

Derivare  $F'$  da  $F$  usando le regole di Armstrong, dove  $F = \{AB \rightarrow DE, A \rightarrow CD, F \rightarrow DG\}$  e  $F' = \{AB \rightarrow CDE, AB \rightarrow A, F \rightarrow D, F \rightarrow G\}$ . Si può concludere che  $F$  ed  $F'$  sono equivalenti?

**Soluzione.**

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

Per riflessività otteniamo

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

E infine per decomposizione abbiamo

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

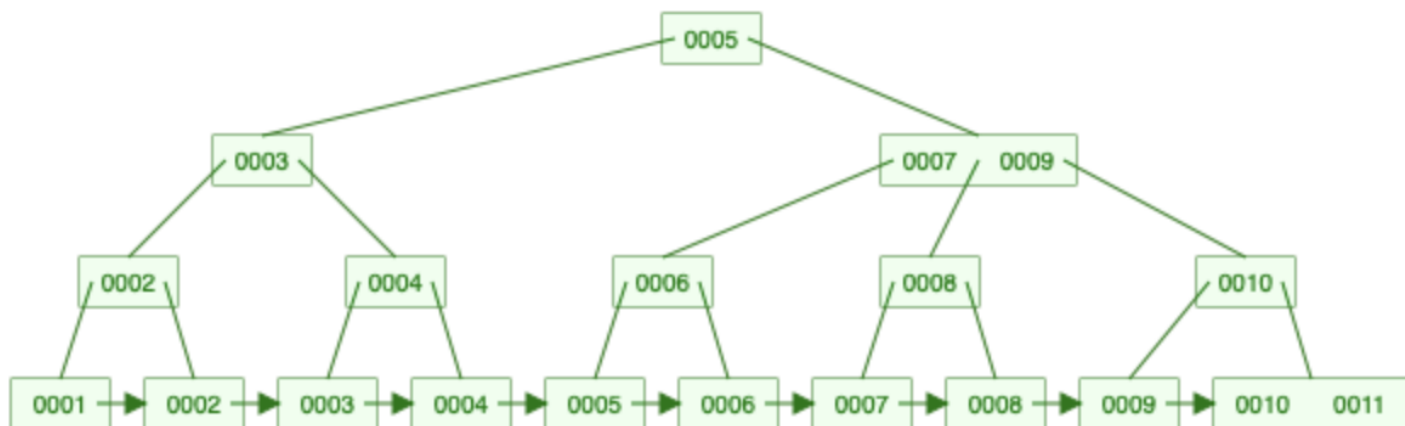
Si noti che i due insiemi non sono equivalenti perché la d.f.  $A \rightarrow CD$  in  $F$  non è deducibile da  $F'$ .

## 11. Esonero Teoria B+-Tree V1

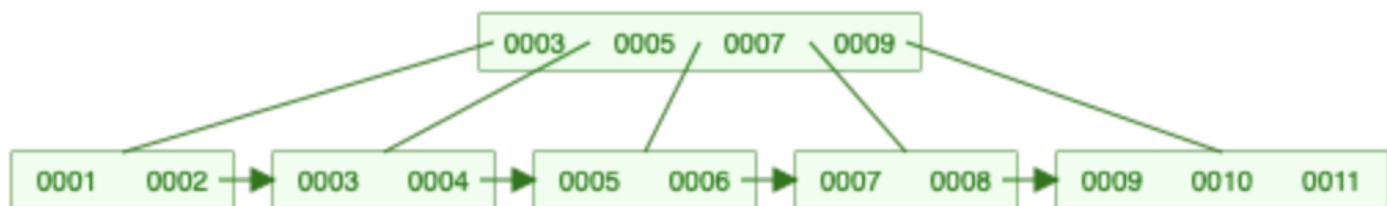
Rappresentare due possibili B+-tree contenenti le chiavi  $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)$  per i casi con  $m=3$  e  $m=5$ . Non si richiede di simulare le singole operazioni di inserimento, ma di mostrare un possibile B+-tree con le caratteristiche indicate.

**Soluzione.** 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  $m=5$ . Ogni nodo ha al massimo  $m - 1 = 4$  chiavi, la radice ha almeno 1 chiave, gli altri nodi hanno almeno  $\lceil \frac{m}{2} - 1 \rceil = 2$  chiavi.



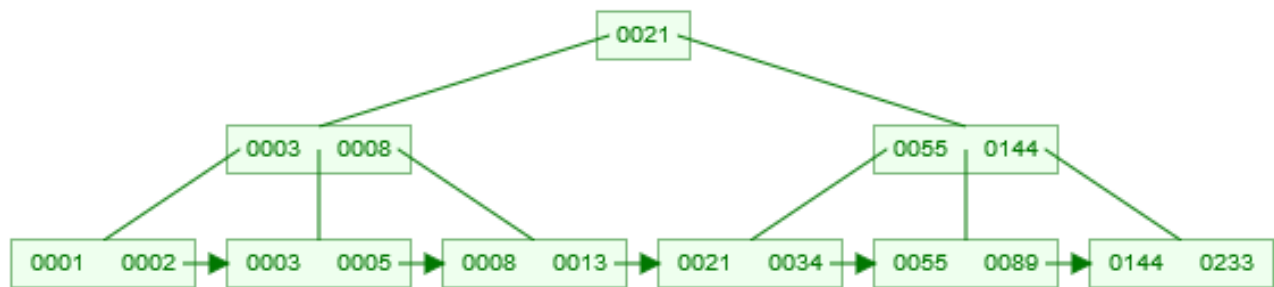
## 12. Esonero Teoria B+-Tree V2

Rappresentare due possibili B+-tree contenenti le chiavi  $(1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233)$  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.

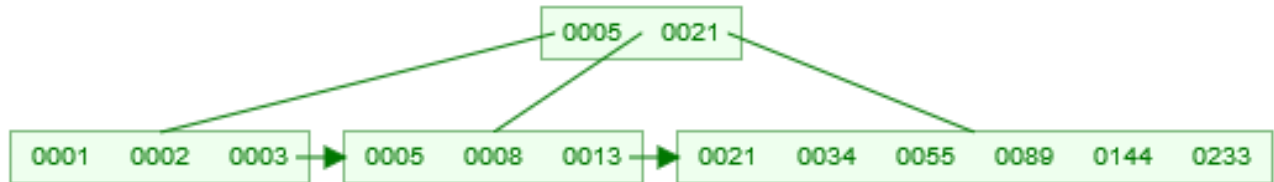
**Soluzione.** Caso  $m=4$ . Ogni nodo ha al massimo  $m - 1 = 3$  chiavi, la radice ha almeno 1 chiave, gli altri nodi hanno almeno  $\lceil \frac{m}{2} - 1 \rceil = 1$  chiave. Le foglie devono contenere tutte le chiavi e essere linkate.

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.



### 13. Esonero Teoria 2PL V1

Considerare la seguente storia interfogliata

$S = r1(x), r1(y), r2(x), r3(y), r2(y), w2(x)$  La storia  $S$  è compatibile con il protocollo 2PL? Giustificare la risposta.

**Soluzione.**

La storia  $S$  è compatibile con 2PL perché banalmente esiste c'è un'unica scrittura ed è l'ultima operazione della storia, quindi sicuramente le altre transazioni possono rilasciare i loro lock senza avere bisogno di richiederli nuovamente. Un esempio di aggiunta dei lock 2PL è il seguente:

$S = LS1(x), r1(x), LS1(y), r1(y), LS2(x), r2(x), LS3(y), r3(y), LS2(y), r2(y), UL1(x), LX2(x), w2(x)$

### 14. Esonero Teoria 2PL V2

Considerare la seguente storia interfogliata

$S = w1(x), r2(x), w3(y), r1(y), r2(y), r3(x)$

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

**Soluzione.**

La storia  $S$  non è compatibile con 2PL perché la transazione T1 dovrebbe riacquisire un lock su  $y$  dopo aver rilasciato il lock su  $x$ , in violazione al protocollo.