

1. Esonero SQL V1

Le seguenti relazioni definiscono una base di dati “MustEat” per gestire le ordinazioni e le consegne di pasti a domicilio. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

UTENTE(Codice, Nome, Email, Città, IndirizzoPreferito)

INDIRIZZO(Utente, Indirizzo, Note)

ORDINE(Utente, Ristorante, DataOrdine, Prezzo, IndirizzoConsegna)

RISTORANTE(Codice, Nome, Tipo, Indirizzo, Città)

Vincoli di integrità referenziale:

INDIRIZZO(Utente) referencia UTENTE(Codice),

UTENTE(Codice, IndirizzoPreferito) referencia INDIRIZZO(Utente, Indirizzo),

ORDINE(Ristorante) referencia RISTORANTE(Codice) e

ORDINE(Utente, IndirizzoConsegna) referencia INDIRIZZO(Utente, Indirizzo).

Significato degli attributi: *IndirizzoPreferito* è l'indirizzo di consegna preferito per un dato utente; *Tipo* può assumere i valori *pizzeria*, *paninoteca*, *cucina etnica* e *cucina vegetariana*; *Prezzo* è il prezzo totale di un ordine. I rimanenti attributi sono autoesplicativi. Gli attributi sono tutti NOT NULL.

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

(A - Bassa complessità) Indicare la data e il prezzo degli ordini (effettuati presso pizzerie) consegnati presso un indirizzo diverso da quello preferito dagli utenti che li hanno effettuati.

(B - Media complessità) Indicare nome e città degli utenti che hanno ordinato pasti ognuno di almeno 50 euro da almeno due ristoranti di tipo diverso.

(C - Alta complessità) Trovare i ristoranti che hanno totalizzato più ricavi rispetto agli altri ristoranti del loro stesso tipo (ovvero, i ristoranti per cui la somma totale del prezzo degli ordini è maggiore di quella degli altri ristoranti del loro stesso tipo).

Soluzioni.

A.

```
select o.DataOrdine, o.Prezzo
from utente u join ordine o on (u.Codice=o.Utente) join ristorante r on (o.Ristorante=r.Codice)
where o.IndirizzoConsegna<>u.IndirizzoPreferito and r.Tipo='pizzeria';
```

B.

```
select u.Nome, u.Città
from utente u join ordine o on (u.Codice=o.Utente) join ristorante r on (r.Codice=o.Ristorante)
where o.Prezzo>= 50
group by u.Codice
having (distinct r.Tipo)>=2;
```

C.

```
with RicaviRistorante as (
select r.Codice, r.Nome, r.Tipo, sum(o.Prezzo) TotRicavi
from ristorante r join ordine o on (r.Codice=o.Ristorante)
group by r.Codice )
select *
from RicaviRistorante r1
where r1.TotRicavi >= all ( select r2.TotRicavi
```

from RicaviRistorante r2
where r2.Tipo=r1.Tipo and r2.Codice<>r1.Codice);

2. Esonero SQL V2

Le seguenti relazioni definiscono una base di dati “MustEat” per gestire le ordinazioni e le consegne di pasti a domicilio. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

UTENTE(Codice, Nome, Email, Città, IndirizzoPreferito)

INDIRIZZO(Utente, Indirizzo, Note)

ORDINE(Utente, Ristorante, DataOrdine, Prezzo, IndirizzoConsegna)

RISTORANTE(Codice, Nome, Tipo, Indirizzo, Città)

Vincoli di integrità referenziale:

INDIRIZZO(Utente) referencia UTENTE(Codice),

UTENTE(Codice, IndirizzoPreferito) referencia INDIRIZZO(Utente, Indirizzo),

ORDINE(Ristorante) referencia RISTORANTE(Codice) e

ORDINE(Utente, IndirizzoConsegna) referencia INDIRIZZO(Utente, Indirizzo).

Significato degli attributi: *IndirizzoPreferito* è l'indirizzo di consegna preferito per un dato utente; *Tipo* può assumere i valori *pizzeria*, *paninoteca*, *cucina etnica* e *cucina vegetariana*; *Prezzo* è il prezzo totale di un ordine. I rimanenti attributi sono autoesplicativi. Gli attributi sono tutti NOT NULL.

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

(A - Bassa complessità) Riportare, ordinate per ordine alfabetico inverso, le email degli utenti che hanno ordinato pasti in pizzerie o in paninoteche di Torino.

(B - Media complessità) Trovare i tipi dei ristoranti di Nichelino che hanno servito almeno due clienti di Torino.

(C - Alta complessità) Tra i ristoranti che hanno ogni ordine al di sotto di 50 euro, trovare il ristorante con il ricavo maggiore nella fascia di ricavi totali tra 80 mila euro e 100 mila euro.

Soluzioni.

A.

```
select email
from utente u join ordine o on u.codice=o.utente join ristorante r on o.ristorante=r.codice
where (tipo='pizzeria' or tipo='paninoteca') and r.città='Torino'
order by email desc;
```

B.

```
select distinct tipo
from ristorante r join ordine o1 on r.codice=o1.ristorante join ordine o2 on r.codice=o2.ristorante join
utente u1 join o1.utente=u1.codice join utente u2 on o2.utente=u2.codice
where r.città='Nichelino' and u1.città='Torino' and u2.città='Torino' and u1.codice<>u2.codice;
```

C.

```
with risticavi as (
select ristorante, sum(prezzo) as totale
from ordine o
where ristorante not in (
select ristorante
from ordine
where prezzo ≥ 50)
```

```

group by ristorante
having sum(prezzo) ≥ 800000 and sum(prezzo) ≤ 100000
select ristorante
from risticavi
where totale ≥ all (
select totale
from risticavi);

```

3. Esonero Algebra V1

Le seguenti relazioni definiscono una base di dati “MustEat” per gestire le ordinazioni e le consegne di pasti a domicilio. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

UTENTE(Codice, Nome, Email, Città, IndirizzoPreferito)

INDIRIZZO(Utente, Indirizzo, Note)

ORDINE(Utente, Ristorante, DataOrdine, Prezzo, IndirizzoConsegna)

RISTORANTE(Codice, Nome, Tipo, Indirizzo, Città)

Vincoli di integrità referenziale:

INDIRIZZO(Utente) referencia UTENTE(Codice),

UTENTE(Codice, IndirizzoPreferito) referencia INDIRIZZO(Utente, Indirizzo),

ORDINE(Ristorante) referencia RISTORANTE(Codice) e

ORDINE(Utente, IndirizzoConsegna) referencia INDIRIZZO(Utente, Indirizzo).

Significato degli attributi: *IndirizzoPreferito* è l’indirizzo di consegna preferito per un dato utente; *Tipo* può assumere i valori *pizzeria*, *paninoteca*, *cucina etnica* e *cucina vegetariana*; *Prezzo* è il costo totale di un ordine. I rimanenti attributi sono autoesplicativi. Gli attributi sono tutti NOT NULL.

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 ordinato almeno un pasto per ogni tipologia di ristorante.

(B - Calcolo Relazionale su tuple con dichiarazione di range) Elencare gli utenti che hanno ordinato almeno un pasto in ogni ristorante della propria città.

Soluzioni.

A. $\pi_{Utente, Tipo}(\text{ordine} \bowtie_{Ristorante=Codice} ristorante) \div \pi_{Tipo}(ristorante)$

B. $\{u.Citta | u(utente) | \forall r(ristorante) (r.Citta = u.Citta \Rightarrow \exists o(ordine) (o.Utente = u.Codice \wedge r.Codice = o.Ristorante))\}$

4. Esonero Algebra V2

Le seguenti relazioni definiscono una base di dati “MustEat” per gestire le ordinazioni e le consegne di pasti a domicilio. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

UTENTE(Codice, Nome, Email, Città, IndirizzoPreferito)

INDIRIZZO(Utente, Indirizzo, Note)

ORDINE(Utente, Ristorante, DataOrdine, Prezzo, IndirizzoConsegna)

RISTORANTE(Codice, Nome, Tipo, Indirizzo, Città)

Vincoli di integrità referenziale:

INDIRIZZO(Utente) referencia UTENTE(Codice),

UTENTE(Codice, IndirizzoPreferito) referencia INDIRIZZO(Utente, Indirizzo),

ORDINE(Ristorante) referencia RISTORANTE(Codice) e
 ORDINE(Utente, IndirizzoConsegna) referencia INDIRIZZO(Utente, Indirizzo).

Significato degli attributi: *IndirizzoPreferito* è l'indirizzo di consegna preferito per un dato utente; *Tipo* può assumere i valori *pizzeria*, *paninoteca*, *cucina etnica* e *cucina vegetariana*; *Prezzo* è il costo totale di un ordine. I rimanenti attributi sono autoesplicativi. Gli attributi sono tutti NOT NULL.

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 almeno due ordini in pizzerie diverse.

(B - Calcolo Relazionale su tuple con dichiarazione di range) Elencare le città con almeno un ristorante per cui ogni ordine è di almeno 50 euro (escludendo i ristoranti senza ordini).

Soluzioni.

A. $\pi_{O1.Utente}(\sigma_{R1.Tipo='Pizzeria'\wedge R2.Tipo='Pizzeria'}(\rho_{O1\leftarrow ORDINE, R1\leftarrow RISTORANTE}(\text{ordine} \bowtie_{Ristorante=Codice} ristorante) \bowtie_{O1.Utente=O2.Utente\wedge R1.Codice\neq R2.Codice} \rho_{O2\leftarrow ORDINE, R2\leftarrow RISTORANTE}(\text{ordine} \bowtie_{Ristorante=Codice} ristorante)))$

B. $\{r.Citta | r(ristorante) | \exists o(ordine)(o.Ristorante = r.Codice) \wedge \forall o(ordine)(o.Ristorante = r.Codice \Rightarrow o.Prezzo \geq 50)\}$

5. Esonero Algebra V1

Le seguenti relazioni definiscono una base di dati “MustEat” per gestire le ordinazioni e le consegne di pasti a domicilio. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

UTENTE(Codice, Nome, Email, Città, IndirizzoPreferito)

INDIRIZZO(Utente, Indirizzo, Note)

ORDINE(Utente, Ristorante, DataOrdine, Prezzo, IndirizzoConsegna)

RISTORANTE(Codice, Nome, Tipo, Indirizzo, Città)

Vincoli di integrità referenziale:

INDIRIZZO(Utente) referencia UTENTE(Codice),

UTENTE(Codice, IndirizzoPreferito) referencia INDIRIZZO(Utente, Indirizzo),

ORDINE(Ristorante) referencia RISTORANTE(Codice) e

ORDINE(Utente, IndirizzoConsegna) referencia INDIRIZZO(Utente, Indirizzo).

Significato degli attributi: *IndirizzoPreferito* è l'indirizzo di consegna preferito per un dato utente; *Tipo* può assumere i valori *pizzeria*, *paninoteca*, *cucina etnica* e *cucina vegetariana*; *Prezzo* è il costo totale di un ordine. I rimanenti attributi sono autoesplicativi. Gli attributi sono tutti NOT NULL.

Data la seguente query sulla base di dati MustEat:

$\sigma_{Nome='Robert Hope'\wedge Tipo='Pizzeria'}((\text{utente} \bowtie_{utente.Codice=Utente} \text{ordine}) \bowtie_{Ristorante=ristorante.Codice} \text{ristorante})$

disegnare gli alberi sintattici prima e dopo l'ottimizzazione logica. Per semplicità non considerare le varianti di parsificazione date dalla proprietà associativa dei join.

Inoltre calcolare il numero di tuple “mosse” prima e dopo l'ottimizzazione logica.

Si svolgano i calcoli sapendo che:

CARD(ordine) = 100 000

CARD(utente) = VAL(Utente, ordine) = 10 000

CARD(ristorante) = VAL(Ristorante, ordine) = 1 000

VAL(Nome, utente) = 5 000

Soluzioni.

$(\sigma_{Nome='Robert Hope'}(utente) \bowtie_{utente.Codice=Utente} ordine) \bowtie_{Ristorante=ristorante.Codice} \sigma_{Tipo='Pizzeria'}(ristorante))$

Prima dell'ottimizzazione:

- Costo $r_1 = utente \bowtie_{utente.Codice=Utente} ordine$: $10\,000 \times 100\,000 = 10^9$.
- Cardinalità di $|r_1| = |utente \bowtie_{utente.Codice=Utente} ordine| = CARD(ordine) = 100\,000$ (equijoin attraverso la chiave esterna)
- Costo join $r_2 = r_1 \bowtie_{Ristorante=ristorante.Codice} ristorante = 100\,000 \times 1\,000 = 10^8$
- Cardinalità di $|r_2| = |r_1 \bowtie_{Ristorante=ristorante.Codice} ristorante| =$
 $= \min\{\frac{1}{VAL(Ristorante,r_1)}, \frac{1}{VAL(Codice,ristorante)}\} \times CARD(r_1) \times CARD(ristorante) = \min\{\frac{1}{1\,000}, \frac{1}{1\,000}\} \times$
 $100\,000 \times 1\,000 = 10^5$
- Costo della selezione: $|r_2|$
- Costo totale = $10^9 + 10^8 + 10^5 \approx 10^9$

Dopo l'ottimizzazione:

- Costo $\sigma_{Nome='Robert Hope'}(utente) = 10^4$
- Tuple prodotte dalla selezione $|\sigma_1| = |\sigma_{Nome='Robert Hope'}(utente)| = \frac{1}{VAL(Nome,utente)} \times CARD(utente) = 2$
- Costo $\sigma_{Tipo='Pizzeria'}(ristorante) = 10^3$
- Tuple prodotte dalla selezione $|\sigma_2| = |\sigma_{Tipo='Pizzeria'}(ristorante)| =$
 $= \frac{1}{VAL(Tipo,ristorante)} \times CARD(ristorante) = \frac{1}{4} \times 1\,000 = 250$
- Costo $r_1 = \sigma_1 \bowtie_{utente.Codice=Utente} ordine$: $2 \times 100\,000 = 2 \times 10^5$.
- Cardinalità di $|r_1| = |utente \bowtie_{utente.Codice=Utente} ordine| =$
 $= \min\{\frac{1}{\min\{VAL(Codice,utente), CARD(\sigma_1)\}}, \frac{1}{VAL(Utente,ordine)}\} \times CARD(\sigma_1) \times CARD(ordine) =$
 $= \min\{\frac{1}{\min\{1\,000, 2\}}, \frac{1}{10\,000}\} \times 2 \times 100\,000 = \min\{\frac{1}{2}, \frac{1}{10\,000}\} \times 2 \times 100\,000 = 20$
- Costo join $r_2 = r_1 \bowtie_{Ristorante=ristorante.Codice} \sigma_2 = 20 \times 250 = 5 \times 10^3$
- Costo totale = $10^4 + 10^3 + 2 \times 10^5 + 5 \times 10^3 \approx 2 \times 10^5$

6. Esonero Algebra V2

Le seguenti relazioni definiscono una base di dati “MustEat” per gestire le ordinazioni e le consegne di pasti a domicilio. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

UTENTE(Codice, Nome, Email, Città, IndirizzoPreferito)

INDIRIZZO(Utente, Indirizzo, Note)

ORDINE(Utente, Ristorante, DataOrdine, Prezzo, IndirizzoConsegna)

RISTORANTE(Codice, Nome, Tipo, Indirizzo, Città)

Vincoli di integrità referenziale:

INDIRIZZO(Utente) referencia UTENTE(Codice),

UTENTE(Codice, IndirizzoPreferito) referencia INDIRIZZO(Utente, Indirizzo),

ORDINE(Ristorante) referencia RISTORANTE(Codice) e

ORDINE(Utente, IndirizzoConsegna) referencia INDIRIZZO(Utente, Indirizzo).

Significato degli attributi: *IndirizzoPreferito* è l'indirizzo di consegna preferito per un dato utente; *Tipo* può assumere i valori *pizzeria*, *paninoteca*, *cucina etnica* e *cucina vegetariana*; *Prezzo* è il costo totale di un ordine. I rimanenti attributi sono autoesplicativi. Gli attributi sono tutti NOT NULL.

Data la seguente query sulla base di dati MustEat:

$\sigma_{Nome='Joseph \ Count'} \wedge Tipo='paninoteca' ((utente \bowtie_{utente.Codice=Utente \ ordine}) \bowtie_{Ristorante=ristorante.Codice \ ristorante})$

disegnare gli alberi sintattici prima e dopo l'ottimizzazione logica. Per semplicità non considerare le varianti di parsificazione date dalla proprietà associativa dei join.

Inoltre calcolare il numero di tuple “mosse” prima e dopo l'ottimizzazione logica.

Si svolgano i calcoli sapendo che:

$$CARD(ordine) = 200\ 000$$

$$CARD(utente) = VAL(Utente, ordine) = 20\ 000$$

$$CARD(ristorante) = 2\ 000$$

$$VAL(Nome, utente) = 10\ 000$$

Soluzioni.

Query ottimizzata:

$(\sigma_{Nome='Joseph \ Count'}(utente) \bowtie_{utente.Codice=Utente \ ordine}) \bowtie_{Ristorante=ristorante.Codice} \sigma_{Tipo='paninoteca'}(ristorante)$

Prima dell'ottimizzazione:

- Costo r_1 : $utente \bowtie_{utente.Codice=Utente \ ordine}$: $20\ 000 \cdot 200\ 000 = 4 \cdot 10^9$.
- Cardinalità di r_1 : $|r_1| = |utente \bowtie_{utente.Codice=Utente \ ordine}| = CARD(ordine) = 200\ 000$ (equijoin attraverso la chiave esterna)
- Costo join r_2 : $r_1 \bowtie_{Ristorante=ristorante.Codice} ristorante = 200\ 000 \cdot 2\ 000 = 4 \cdot 10^8$
- Cardinalità di r_2 : $|r_2| = |r_1 \bowtie_{Ristorante=ristorante.Codice} ristorante| = \min\{\frac{1}{VAL(Ristorante, r_1)}, \frac{1}{VAL(Codice, ristorante)}\} \cdot CARD(r_1) \cdot CARD(ristorante) = \min\{\frac{1}{200\ 000}, \frac{1}{2\ 000}\} \cdot 200\ 000 \cdot 2\ 000 = 2 \cdot 10^5$
- Costo della selezione: $|r_2|$
- Costo totale = $4 \cdot 10^9 + 4 \cdot 10^8 + 2 \cdot 10^5 \approx 4 \cdot 10^9$

Dopo l'ottimizzazione:

- Costo σ_1 : $\sigma_{Nome='Joseph \ Count'}(utente) = 2 \cdot 10^4$
- Tuple prodotte dalla selezione $|\sigma_1| = |\sigma_{Nome='Joseph \ Count'}(utente)| = \frac{1}{VAL(Nome, utente)} \cdot CARD(utente) = 2$
- Costo σ_2 : $\sigma_{Tipo='paninoteca'}(ristorante) = 2 \cdot 10^3$
- Tuple prodotte dalla selezione $|\sigma_2| = |\sigma_{Tipo='paninoteca'}(ristorante)| = \frac{1}{VAL(Tipo, ristorante)} \cdot CARD(ristorante) = \frac{1}{4} \cdot 2\ 000 = 500$
- Costo $r_1 = \sigma_1 \bowtie_{utente.Codice=Utente \ ordine}$: $2 \cdot 200\ 000 = 4 \cdot 10^5$.
- Cardinalità di $|r_1| = |\sigma_{Nome='Joseph \ Count'}(utente) \bowtie_{utente.Codice=Utente \ ordine}| = \min\{\frac{1}{VAL(Codice, \sigma_1)}, \frac{1}{VAL(Utente, ordine)}\} \cdot CARD(\sigma_1) \cdot CARD(ordine) = \min\{\frac{1}{2}, \frac{1}{20\ 000}\} \cdot 2 \cdot 200\ 000 = 20$
- Costo join $r_2 = r_1 \bowtie_{Ristorante=ristorante.Codice} \sigma_2 = 20 \cdot 500 = 10^4$
- Costo totale = $2 \cdot 10^4 + 2 \cdot 10^3 + 4 \cdot 10^5 + 5 \cdot 10^4 \approx 4 \cdot 10^5$

7. Esonero Teoria 3NF V1

Dati:

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

$$F = \{AC \rightarrow FG, C \rightarrow DEG, E \rightarrow F\}$$

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

Soluzione.

Per prima cosa, occorre individuare la o le chiavi della relazione R. Una chiave possibile è ABC, infatti B non compare in nessuna dipendenza funzionale (quindi deve far parte della chiave) e abbiamo che: $ABC^+ = \{A, B, C|F, G|D, E\}$.

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: $F' = \{AC \rightarrow F, AC \rightarrow G, C \rightarrow D, C \rightarrow E, C \rightarrow G, E \rightarrow F\}$

A è estraneo sia in $AC \rightarrow F$ che in $AC \rightarrow G$. Infatti sia F che G sono determinabili partendo dal solo C, quindi:

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

La dipendenza funzionale $C \rightarrow F$ è ridondante per transitività, quindi:

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

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

R1(C,D,E,G)

R2(E,F)

cui bisogna aggiungere anche una relazione R3(A,B,C) contenente la chiave della relazione R.

8. Esonero Teoria 3NF V2

Dati:

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

$$F = \{DF \rightarrow BC, F \rightarrow GAC, A \rightarrow B\}$$

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

Soluzione.

Per prima cosa, occorre individuare la o le chiavi della relazione R. Una chiave possibile è DEF, infatti E non compare in nessuna dipendenza funzionale (quindi deve far parte della chiave) e abbiamo che: $DEF^+ = \{D, E, F|B, C|G, A\}$.

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

Troviamo quindi la copertura minimale di F partendo dalla sua forma canonica: $F' = \{DF \rightarrow B, DF \rightarrow C, F \rightarrow G, F \rightarrow A, F \rightarrow C, A \rightarrow B\}$

D è estraneo sia in $DF \rightarrow B$ che in $DF \rightarrow C$. Infatti sia B che C sono determinabili partendo dal solo F, quindi:

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

La dipendenza funzionale $F \rightarrow B$ è ridondante per transitività, quindi:

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

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

$R1(\underline{F}, G, A, C)$

$R2(\underline{A}, B)$

cui bisogna aggiungere anche una relazione $R3(\underline{D}, \underline{E}, F)$ contenente la chiave della relazione R.

9. Esonero Teoria Armstrong V1

Derivare la regola dell'unione partendo dalla regola dell'espansione e applicando le altre regole di Armstrong.

Soluzione.

Vogliamo dimostrare che, date due dipendenze funzionali $X \rightarrow Y$ e $X \rightarrow Z$, allora vale $X \rightarrow YZ$.

Sfruttiamo la proprietà dell'espansione, applicandola alla d.f. $X \rightarrow Y$ con un insieme X . Abbiamo: $X \rightarrow XY$. Appliciamo ora la stessa proprietà a $X \rightarrow Z$ espandendola con un insieme Y . Abbiamo: $XY \rightarrow YZ$.

Per la transitività, quindi, dato che $X \rightarrow XY$ e $XY \rightarrow YZ$, vale $X \rightarrow YZ$, come volevasi dimostrare.

10. Esonero Teoria Armstrong V2

Derivare la regola del prodotto partendo dalla regola dell'espansione e applicando le altre regole di Armstrong.

Soluzione.

Vogliamo dimostrare che, date due dipendenze funzionali $X \rightarrow Y$ e $W \rightarrow Z$ allora vale $XW \rightarrow YZ$.

Sfruttiamo la proprietà dell'espansione, applicandola alla d.f. $X \rightarrow Y$ con un insieme W . Abbiamo: $XW \rightarrow WY$. Appliciamo ora la stessa proprietà a $W \rightarrow Z$, espandendola con un insieme Y . Abbiamo: $WY \rightarrow YZ$.

Per la transitività, quindi, dato che $XW \rightarrow WY$ e $WY \rightarrow YZ$, vale $XW \rightarrow YZ$, come volevasi dimostrare.

11. Esonero Teoria B+-Tree V1

A. Spiegare in modo succinto ma preciso le principali differenze tra i vari tipi di data entry $\langle k, * \rangle$.

Attenzione: non si richiedono le definizioni, ma le differenze e in quali casi conviene usare un tipo o l'altro, quindi una rielaborazione di ciò che è stato spiegato.

B. Dato un B+-tree con $m=5$ contenente 125 chiavi, qual è il costo massimo di accesso ai dati nel caso in cui il B+-tree sia usato come struttura di memorizzazione primaria (ovvero, come indice clusterizzato fuso con l'area primaria)?

Soluzioni.

A. Si vedano gli appunti, slide, testo.

B. Sappiamo che $\log_m(N) \leq L \leq \log_m(N) + 1$, quindi, nel nostro caso: $3 \leq L \leq 4$. Il costo massimo quindi è 4, poiché non è necessario il costo aggiuntivo per accedere all'aria primaria contenente i record.

12. Esonero Teoria B+-Tree V2

A. Spiegare in modo succinto ma preciso le principali differenze tra i vari tipi di data entry $\langle k, * \rangle$.

Attenzione: non si richiedono le definizioni, ma le differenze e in quali casi conviene usare un tipo o l'altro, quindi una rielaborazione di ciò che è stato spiegato.

B. Dato un B+-tree con $m=5$ contenente 625 chiavi, qual è il costo massimo di accesso ai dati nel caso in cui il B+-tree sia usato come struttura di memorizzazione primaria (ovvero, come indice clusterizzato fuso con l'area primaria)?

Soluzione.

A. Si vedano gli appunti, slide e il testo.

B. Sappiamo che $\log_m(N) \leq L \leq \log_m(N) + 1$, quindi, nel nostro caso, dato che $5^4 = 625$, $4 \leq L \leq 5$. Il costo massimo quindi è 5, poiché non è necessario il costo aggiuntivo per accedere all'aria primaria contenente i record.

13. Esonero Teoria Log V1

Si consideri un file di log L con il seguente contenuto in seguito a un crash:

$\langle T1, START \rangle;$
 $\langle T2, START \rangle;$
 $\langle T1, BS(t1[A], 5), AS(t1[A], 10) \rangle;$
 $\langle T2, BS(t2[B], 3), AS(t2[B], 5) \rangle;$
 $\langle T2, ABORT \rangle$
 $\langle T3, START \rangle;$
 $\langle T1, COMMIT \rangle;$
 $\langle T3, BS(t3[C], 3), AS(t3[C], 5) \rangle;$
 $\langle T3, COMMIT \rangle$

crash!

Indicare il contenuto delle liste LA (lista delle transazioni da disfare) ed LC (lista delle transazioni da rifare) per il ripristino.

Soluzione.

Le transazioni T1 e T3 vanno in commit prima del crash. Vanno quindi rifatte e $LC = \{T1, T3\}$. La transazione T2 va in abort prima del crash, quindi non va disfatta. Di conseguenza $LA = \{\emptyset\}$.

14. Esonero Teoria Log V2

Si consideri un file di log L con il seguente contenuto in seguito a un crash:

$\langle T1, START \rangle;$
 $\langle T2, START \rangle;$
 $\langle T2, BS(t2[B], 3), AS(t2[B], 99) \rangle;$
 $\langle T3, START \rangle;$
 $\langle T2, ABORT \rangle$
 $\langle T1, BS(t1[A], 5), AS(t1[A], 60) \rangle;$
 $\langle T3, BS(t3[C], 10), AS(t3[C], 20) \rangle;$
 $\langle T3, COMMIT \rangle$
 $\langle T1, COMMIT \rangle;$

crash!

Indicare il contenuto delle liste LA (lista delle transazioni da disfare) ed LC (lista delle transazioni da rifare) per il ripristino.

Soluzione.

Le transazioni T1 e T3 vanno in commit prima del crash. Vanno quindi rifatte e $LC = \{T1, T3\}$. La transazione T2 va in abort prima del crash, quindi non va disfatta e $LA = \{\emptyset\}$.