

Cognome e nome: _____

Matricola: _____

Esame di SQL (rispondere su fogli separati rispetto alla parte di teoria)

Le seguenti relazioni definiscono una base di dati “Pizzeria” per gestire gli ordini di una pizzeria. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

ORDINE(Pizza, Tavolo, Data, Quantità, Note)

PIZZA(NomePizza, Prezzo, Tipo)

CONTIENE(Pizza, Ingrediente)

INGREDIENTE(NomeIngrediente, Vegetariano, SenzaGlutine)

Vincoli di integrità referenziale: ORDINE(Pizza) referencia PIZZA(NomePizza), CONTIENE(Pizza) referencia PIZZA(NomePizza), CONTIENE(Ingrediente) referencia INGREDIENTE(NomeIngrediente).

“Tipo” può assumere i valori "Normale" e “Gourmet”. “Vegetariano” e “SenzaGlutine” sono booleani.

I rimanenti attributi sono autoesplicativi. Gli attributi sono tutti NOT NULL.

Con riferimento alla base di dati "Pizzeria" esprimere in SQL le seguenti interrogazioni.

Domanda 1 (bassa complessità).

Trovare gli ordini (tavolo e data) composti da almeno due pizze con l’ananas o con le pere.

Soluzione 1.

```
SELECT DISTINCT Data, Tavolo
FROM ordine O JOIN pizza P ON (O.Pizza=P.NomePizza)
      JOIN contiene C ON (P.NomePizza=C.Pizza)
      JOIN ingrediente I ON (C.Ingrediente=I.NomeIngrediente)
WHERE O.Quantità>=2 AND (I.NomeIngrediente='Ananas' OR I.NomeIngrediente='Pera');
```

Domanda 2 (media complessità).

Mostrare le date in cui dei tavoli hanno speso più di 50 euro in pizze gourmet.

Soluzione 2.

```
SELECT DISTINCT O.Data
FROM ordine O JOIN pizza P ON (O.Pizza=P.NomePizza)
WHERE P.Tipo='Gourmet'
GROUP BY O.Data, O.Tavolo
HAVING SUM(P.Prezzo*O.Quantità)>50;
```

Domanda 3 (alta complessità).

Trovare il conto medio per l’anno 2018 degli ordini composti solo da pizze vegetariane (cioè pizze in cui tutti gli ingredienti sono vegetariani).

Soluzione 3.

```
WITH ContiVegetariani AS (
  SELECT O.Data, O.Tavolo, SUM(P.Prezzo*O.Quantità) Conto
  FROM ordine O JOIN pizza P ON (O.Pizza=P.NomePizza)
  WHERE (O.Data, O.Tavolo) NOT IN (
    SELECT O1.Data, O1.Tavolo
    FROM ordine O1 JOIN pizza P1 ON (O1.Pizza=P1.NomePizza)
      JOIN contiene C ON (P1.NomePizza=C.Pizza)
      JOIN ingrediente I ON (C.Ingrediente=I.NomeIngrediente)
    WHERE I.Vegetariano=False)
  GROUP BY O.Data, O.Tavolo)
SELECT AVG(C.Conto)
FROM ContiVegetariani C
WHERE C.Data>='01-01-2018' AND C.Data<'01-01-2019';
```

Esame di Teoria (rispondere su fogli separati rispetto alla parte di SQL)

Domanda 1.

Con riferimento alla base di dati “Pizzeria”:

A. Esprimere in Algebra Relazionale l'interrogazione

Elencare gli ingredienti che sono presenti in almeno due pizze.

B. Esprimere, nel calcolo dei predicati su tuple con dichiarazione di range, la seguente domanda:

Elencare le pizze (si richiede nome e prezzo) di tipo gourmet che hanno solo ingredienti senza glutine.

Soluzione 1.

A. Una possibile soluzione (in algebra relazionale) è la seguente:

$$\pi_{C1.Ingrediente} \left(\left(\rho_{C1 \leftarrow Contiene}(Contiene) \bowtie_{\theta_1} \rho_{C2 \leftarrow Contiene}(Contiene) \right) \right)$$

dove $\theta_1 := C1.Ingrediente = C2.Ingrediente \wedge C1.Pizza \neq C2.Pizza$

B. Una possibile soluzione è la seguente:

$$\{p.NomePizza, p.Prezzo \mid p(PIZZA) \mid p.Tipo='Gourmet' \wedge \forall c(CONTIENE)(c.Pizza=p.NomePizza \Rightarrow \exists i (INGREDIENTE)(c.Ingrediente=i.NomeIngrediente \wedge i.SenzaGlutine=True))\}$$

Domanda 2.

A. Definire il concetto di dipendenza funzionale.

Si considerino $R(A, B, C, D, E)$ e $F' = \{B \rightarrow D, D \rightarrow E\}$.

B. Le dipendenze funzionali in F' non giustificano A come chiave di R : aggiungere a F' delle dipendenze funzionali in modo che A sia chiave di R .

C. Considerando il nuovo insieme di dipendenze funzionali ottenuto nel punto precedente, dire se R è in 3FN e se non lo è decomporla in relazioni in 3FN esplicitando tutti i passaggi. Lo schema ottenuto è BCNF?

Soluzione 2.

A. Vedere libro di testo o slide.

B. Se aggiungiamo la d.f. $A \rightarrow BC$, la chiusura di A comprende tutti gli attributi di R quindi è una superchiave ed essendo composta da un solo attributo è minimale ed è una chiave.

C. $R(A, B, C, D, E)$ e $F = \{A \rightarrow BC, B \rightarrow D, D \rightarrow E\}$ non è 3NF perché, ad esempio, nella d.f. $B \rightarrow D$ B non è una superchiave, D non è un attributo primo e la d.f. è non banale.

Calcolo l'insieme di copertura minimale.

Trasformo le d.f. mettendo un solo attributo a destra: $F'' = \{A \rightarrow B, A \rightarrow C, B \rightarrow D, D \rightarrow E\}$.

Non ci sono attributi estranei perché ogni d.f. ha un solo attributo a sinistra.

Ogni d.f. ha a destra un attributo differente, quindi ogni d.f. è necessaria per avere quell'attributo nella chiusura e non ci sono d.f. ridondanti. F'' , quindi, è una copertura minimale.

Decompongo R in $R1(\underline{A}, B, C)$, $R2(\underline{B}, D)$, $R3(\underline{D}, E)$. Non ci sono relazioni sottoinsieme di altre e $R1$ comprende la chiave di R quindi ho terminato la normalizzazione in 3NF.

Lo schema ottenuto è BCNF perché ogni d.f. è rappresentata tramite la chiave primaria delle relazioni.

Domanda 3.

- A. Spiegare in modo succinto ma preciso le principali differenze tra B-tree e B+-tree e quali vantaggi apportano i B+-tree rispetto ai B-tree nella gestione degli indici.
- B. Disegnare un B+-tree con $m=4$ contenente come chiavi i numeri da 1 a 20. Non occorre simulare gli inserimenti (le chiavi potrebbero essere state inserite in un ordine qualunque) ma soltanto rappresentare un possibile B+-tree contenente tutte le chiavi.

Soluzione 3.

- A. Vedere libro di testo o slide.
- B. È valido qualsiasi B+-tree che rispetta le proprietà dei B-tree e dei B+-tree riportate sul libro di testo e sulle slide. In particolare, tutti i nodi (eccetto la radice) devono avere due o tre chiavi, ogni chiave k dei nodi intermedi deve essere presente nella foglia più a destra del sottoramo sinistro della chiave k stessa. Se un nodo ha j chiavi, deve avere esattamente $j+1$ figli. Ogni nodo foglia ha un puntatore al nodo foglia adiacente. Inoltre, devono essere rispettate le proprietà base dei B-tree. In particolare, le foglie devono essere tutte al medesimo livello.

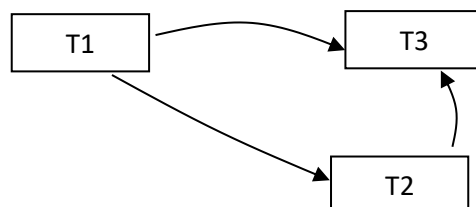
Domanda 4.

- A. Enunciare il criterio di serializzabilità.
- Considerare la seguente storia interfogliata $S = r1(x), r2(y), w2(x), r3(y), w3(x)$.
- B. La storia S è conflict-serializzabile? Giustificare la risposta.
- C. La storia S è view-serializzabile? Giustificare la risposta.
- D. (Caso 1) Nel caso la storia S sia conflict-serializzabile, è possibile aggiungere a S un'azione che la renda non conflict-serializzabile? Se sì, quale?
- (Caso 2) Nel caso la storia S non sia conflict-serializzabile, è possibile togliere da S un'azione che la renda conflict-serializzabile? Se sì, quale?
- N.B.: ovviamente, solo uno dei due casi dell'esercizio D è da svolgere.**

Soluzione 4.

- A. Vedere libro di testo o slide.

B e C. Il grafo dei conflitti è il seguente:



Poiché il grafo è aciclico, la storia è conflict-serializzabile e view-serializzabile.

- D.** Aggiungendo al termine della storia ad esempio l'azione $w1(x)$ si aggiungono al grafo dei conflitti gli archi $T2 \rightarrow T1$ e $T3 \rightarrow T1$ e si formano due cicli rendendo così la storia non conflict-serializzabile.

Cognome e nome: _____

Matricola: _____

Esame di SQL (rispondere su fogli separati rispetto alla parte di teoria)

Le seguenti relazioni definiscono una base di dati “NerdFlix” per gestire una piattaforma di streaming online di serie TV. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

SERIETV(NomeSerie, Produttore, Nazione, Genere)

EPISODIO(NomeSerie, NumStagione, NumEpisodio, TitoloEpisodio, Anno)

AUDIO(NomeSerie, NumStagione, NumEpisodio, Lingua, AdattatoreDialoghi*, Originale)

SOTTOTITOLI(NomeSerie, NumStagione, NumEpisodio, Lingua)

Vincoli di integrità referenziale:

EPISODIO(NomeSerie) riferenzia SERIETV(NomeSerie),

AUDIO(NomeSerie,NumStagione,NumEpisodio) riferenzia

EPISODIO(NomeSerie,NumStagione,NumEpisodio),

SOTTOTITOLI(NomeSerie,NumStagione,NumEpisodio) riferenzia

EPISODIO(NomeSerie,NumStagione,NumEpisodio).

“Genere” può assumere i valori “Animazione”, “Dramma”, “Fantascienza” e “Commedia”. “Originale” può assumere i valori True (se l’audio è quello della lingua originale) e False (se si tratta di un doppiaggio). I rimanenti attributi sono autoesplicativi. AdattatoreDialoghi può essere NULL.

Con riferimento alla base di dati “NerdFlix” esprimere in SQL le seguenti interrogazioni.

Domanda 1 (bassa complessità).

Trovare le serie TV giapponesi di genere “Animazione” con dialoghi adattati da “Gualtiero Cannarsi”.

Soluzione 1.

```
SELECT DISTINCT s.NomeSerie
FROM serietv s JOIN episodio e ON (s.NomeSerie=e.NomeSerie)
      JOIN audio a ON (a.NomeSerie=e.NomeSerie AND a.NumStagione=e.NumStagione
                      AND a.NumEpisodio=e.NumEpisodio)
WHERE s.Genere='Animazione' AND s.Nazione='Giappone'
      AND a.AdattatoreDialoghi='Gualtiero Cannarsi';
```

Domanda 2 (media complessità).

Mostrare le serie TV di cui sono state prodotte almeno tre stagioni con sottotitoli in romeno nel periodo 2015-2019.

Soluzione 2.

```
SELECT e.NomeSerie
FROM episodio e JOIN sottotitoli s ON (e.NomeSerie=s.NomeSerie
      AND e.NumStagione=s.NumStagione AND e.NumEpisodio=s.NumEpisodio)
WHERE s.Lingua='Romeno' AND e.Anno>=2015 AND e.Anno<=2019
GROUP BY e.NomeSerie
HAVING COUNT(DISTINCT e.NumStagione)>=3;
```


Esame di Teoria (rispondere su fogli separati rispetto alla parte di SQL)

Domanda 1 (8 punti).

Con riferimento alla base di dati “NerdFlix”:

A. Esprimere in Algebra Relazionale l'interrogazione

Elencare le serie TV doppiate^(*) in almeno due lingue^() (oltre all'originale).**

^(*) è sufficiente che esista il doppiaggio di un solo episodio della serie

^(**) il doppiaggio può riguardare anche episodi diversi della stessa serie

B. Esprimere, nel calcolo dei predicati su tuple con dichiarazione di range, la seguente domanda:

Elencare le serie TV drammatiche (nome e nazione) per cui, di ogni episodio, esiste solo l'audio in lingua originale.

Soluzione 1.

A. Una possibile soluzione (in algebra relazionale) è la seguente:

$$\pi_{A1.NomeSerie} \left(\sigma_{A1.Originale=False \wedge A2.Originale=False} \left(\rho_{A1 \leftarrow AUDIO}(audio) \bowtie_{\theta_1} \rho_{A2 \leftarrow AUDIO}(audio) \right) \right)$$

dove $\theta_1 := A1.NomeSerie = A2.NomeSerie$

B. Una possibile soluzione è la seguente:

$$\{s.NomeSerie, s.Nazione \mid s(SERIETV) \mid s.Genere='Dramma' \wedge \\ \neg \exists a(AUDIO)(a.NomeSerie=s.NomeSerie \wedge a.Originale=False)\}$$

Domanda 2 (9 punti).

A. Riportare la definizione di chiusura di un insieme di attributi.

B. Dire se i seguenti due insiemi di dipendenze funzionali sono equivalenti, dimostrandolo:

$$F1 = \{A \rightarrow CE, C \rightarrow AB, AE \rightarrow F\}, \quad F2 = \{AB \rightarrow CE, A \rightarrow B, C \rightarrow B, C \rightarrow AF\}$$

C. Data una relazione R(ABCDEF), dire se R è in 3NF rispetto all'insieme F2 della domanda B motivando la risposta. Se non lo è, scomporla in 3NF esplicitando tutti i passi.

Soluzione 2.

A. Vedere libro di testo o slide.

B. Applicando una qualsiasi delle definizioni di equivalenza si trova facilmente che F1 ed F2 sono equivalenti.

Il modo più semplice di procedere è di calcolare la copertura minimale di F1 ed F2 e calcolare la chiusura solo per le d.f. non sintatticamente incluse in entrambi gli insiemi.

C. Per prima cosa, bisogna individuare le chiavi di R (a partire da F2). Si trovano facilmente le uniche due chiavi $K1=\{AD\}$ e $K2=\{CD\}$. Nessuna delle dipendenze funzionali è quindi di tipo superchiave, attributi primi o banale.

Calcolo l'insieme di copertura minimale.

Trasformo le d.f. mettendo un solo attributo a destra: $F2''=\{AB \rightarrow C, AB \rightarrow E, A \rightarrow B, C \rightarrow B, C \rightarrow A, C \rightarrow F\}$.

In $AB \rightarrow C$ e $AB \rightarrow E$, B è attributo estraneo (a causa di $A \rightarrow B$); il nuovo insieme è quindi:

$$F2''=\{A \rightarrow C, A \rightarrow E, A \rightarrow B, C \rightarrow B, C \rightarrow A, C \rightarrow F\}.$$

$A \rightarrow B$ è ridondante per transitività, perché $A \rightarrow C$ e $C \rightarrow B$. L'insieme di copertura minimale è quindi:

$$F2''=\{A \rightarrow C, A \rightarrow E, C \rightarrow B, C \rightarrow A, C \rightarrow F\}.$$

Raggruppo le d.f. con lo stesso antecedente e ottengo:

$$F2'' = \{A \rightarrow CE, C \rightarrow ABF\}.$$

Posso quindi scomporre R nelle relazioni R1(ACE) ($C \rightarrow A$ nella restrizione) e R2(CABF) (con $A \rightarrow C$ nella restrizione) cui devo aggiungere una relazione con la chiave AD (o CD), quindi R3(AD).

Domanda 3 (8 punti).

A. Spiegare la differenza tra ottimizzazione logica e ottimizzazione fisica di un'interrogazione.

B. Data la seguente query sulla base di dati “NerdFlix”:

$$\sigma_{audio.Lingua='Italiano' \wedge serietv.NomeSerie='Stranger Things'}((audio \bowtie episodio) \bowtie serietv)$$

dove i join sono tutti naturali, effettuare l'ottimizzazione logica sapendo che

$$CARD(SERIETV) = 1000$$

$$CARD(EPISODIO) = 50.000$$

$$CARD(AUDIO) = 100.000$$

$$VAL(Lingua, AUDIO) = 10$$

Soluzione 3.

A. Vedere libro di testo o slide.

B. Decomponendo l'AND della selezione e trasferendo le selezioni verso le foglie (passi 1 e 2 dell'algoritmo di ottimizzazione logica) ottengo:

$$(\sigma_{audio.Lingua='Italiano'}(audio) \bowtie episodio) \bowtie (\sigma_{serietv.NomeSerie='Stranger Things'}(serietv))$$

Esaminando le varianti dell'albero di parsificazione (passo 7 dell'algoritmo di ottimizzazione logica) scelgo

la variante di costo minimo: dato che $|\sigma_{audio.Lingua='Italiano'}(audio)| = \frac{1}{10} \cdot 100.000 = 10.000$,

$|episodio| = 50.000$ e, poiché NomeSerie è chiave primaria di serietv,

$|\sigma_{serietv.NomeSerie='Stranger Things'}(serietv)| = 1$, posticipiamo il join con la relazione episodio:

$$(\sigma_{audio.Lingua='Italiano'}(audio) \bowtie (\sigma_{serietv.NomeSerie='Stranger Things'}(serietv))) \bowtie episodio$$

Domanda 4 (8 punti).

A. Riportare la definizione di protocollo 2PL (two-phase lock) stretto.

Considerare la seguente storia interfogliata $S = r1(x), r1(y), r2(y), r3(x), w1(x), r2(x), w2(y)$

B. La storia S è conflict-serializzabile? Giustificare la risposta.

C. La storia S è view-serializzabile? Giustificare la risposta

D. (Caso 1) Nel caso la storia S sia view-serializzabile, dire se è anche compatibile con il 2PL stretto, giustificando la risposta.

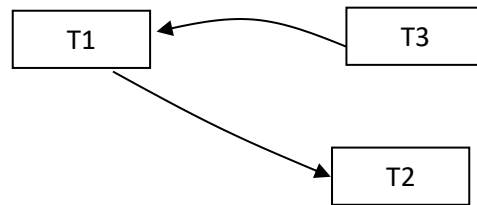
(Caso 2) Nel caso la storia S non sia view-serializzabile, è possibile togliere da S un'azione che la renda view-serializzabile? Se sì, quale?

N.B.: ovviamente, solo uno dei due casi dell'esercizio D è da svolgere.

Soluzione 4.

A. Vedere libro di testo o slide.

B e C. Il grafo dei conflitti è il seguente:



Poiché il grafo è aciclico, la storia è conflict-serializzabile e view-serializzabile.

D. (Caso 1) È compatibile con 2PL stretto perché, per esempio, è possibile aggiungere i lock ottenendo la storia seguente, compatibile con il 2PL stretto:

$S = LS1(x), r1(x), LS1(y), r1(y), LS2(y), r2(y), LS3(x), r3(x), UL3(x), LX1(x), w1(x), UL1(y), UL1(x), LS2(x), r2(x), LX2(y), w2(y), UL2(x), UL2(y)$

Si noti come i lock $LX1(x)$ e $LX2(y)$ sono semplici upgrade di lock shared precedentemente acquisiti dalle rispettive transazioni.

Cognome e nome: _____ Matricola: _____ Turno: _____

Esame di SQL

Rispondere su fogli separati rispetto alla parte di teoria.

Riportare sui fogli i seguenti dati: cognome, nome, matricola e turno di laboratorio.

Le seguenti relazioni definiscono una base di dati “PFWWrestling” per gestire le informazioni di una federazione di wrestling. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

WRESTLER(RingName, Nome, Cognome, Sesso, Peso, StatoProvenienza)

CINTURA(Campione, Titolo, NumeroDiSuccessi)

STABLE(NomeStable, Capitano*)

COMPOSIZIONESTABLE(NomeStable, NomeComponente)

Vincoli di integrità referenziale:

CINTURA(Campione) referencia WRESTLER(RingName),

STABLE(Capitano) referencia WRESTLER(RingName),

COMPOSIZIONESTABLE(NomeStable) referencia STABLE(NomeStable),

COMPOSIZIONESTABLE(NomeComponente) referencia WRESTLER(RingName),

STABLE(NomeStable, Capitano) referencia COMPOSIZIONESTABLE(NomeStable, NomeComponente).

“Sesso” può assumere i valori “Uomo” e “Donna”. Capitano può essere NULL. Titolo può assumere i valori “assoluto”, “intercontinentale” e “femminile”. Una stable è composta da almeno tre wrestler. Solo le donne possono avere un titolo femminile. I rimanenti attributi sono autoesplicativi.

Con riferimento alla base di dati "PFWWrestling" esprimere in SQL le seguenti interrogazioni.

Domanda 1 (bassa complessità).

Trovare gli stati di provenienza dei capitani di stable che hanno vinto almeno 3 volte il titolo assoluto.

Soluzione 1.

```
SELECT DISTINCT StatoProvenienza
FROM wrestler JOIN stable ON (Capitano = RingName)
      JOIN cintura ON (Capitano = Campione)
WHERE Titolo = 'assoluto' AND NumeroDiSuccessi >= 3;
```

Domanda 2 (media complessità).

Trovare ringname, nome e cognome dei wrestlers canadesi che fanno parte di almeno 5 stable i cui capitani hanno vinto il titolo intercontinentale.

Soluzione 2.

```
SELECT RingName, Nome, Cognome
FROM wrestler JOIN composizionestable c ON (RingName = NomeComponente)
      JOIN stable s ON (s.NomeStable = c.NomeStable)
      JOIN cintura ON (Capitano = Campione)
WHERE StatoProvenienza = 'Canada' AND Titolo = 'intercontinentale'
GROUP BY RingName, Nome, Cognome
HAVING COUNT(DISTINCT NomeStable)>=5;
```

Domanda 3 (alta complessità).

Tra le stable composte da soli uomini, trovare quelle che hanno il minimo numero di componenti.

Soluzione 3.

```
WITH StableSenzaDonne AS (  
    SELECT s.NomeStable, COUNT(s.NomeComponente) NumeroComponenti  
    FROM composizionestable s  
    WHERE NOT EXISTS (  
        SELECT *  
        FROM composizionestable c  
        JOIN wrestler w ON (w.RingName = c.NomeComponente)  
        WHERE c.NomeStable=s.NomeStable AND w.Sesso = 'Donna')  
    GROUP BY s.NomeStable)  
SELECT NomeStable  
FROM StableSenzaDonne  
WHERE NumeroComponenti = (SELECT MIN(NumeroComponenti)  
                           FROM StableSenzaDonne);
```

Esame di Teoria (rispondere su fogli separati rispetto alla parte di SQL)

Domanda 1 (9 punti).

Con riferimento alla base di dati “PFWWrestling”:

- A. Esprimere in algebra relazionale l'interrogazione
Elencare i wrestler uomini che hanno peso massimo.
- B. Esprimere, nel calcolo dei predicati su tuple con dichiarazione di range, la seguente domanda:
Elencare le stable composte da soli campioni e capitanate da wrestler canadesi.

Soluzione 1.

A. Una possibile soluzione è la seguente:

$$\pi_{W1.RingName} \left(\rho_{W1 \leftarrow WRESTLER} \left(\sigma_{sesso='Uomo'}(wrestler) \right) \right) \\ - \pi_{W1.RingName} \left(\rho_{W1 \leftarrow WRESTLER} \left(\sigma_{sesso='Uomo'}(wrestler) \right) \right) \bowtie_{W1.Peso < W2.Peso} \\ \rho_{W2 \leftarrow WRESTLER} \left(\sigma_{sesso='Uomo'}(wrestler) \right)$$

B. Una possibile soluzione è la seguente:

$$\{s.NomeStable \mid s(STABLE) \mid \exists w(WRESTLER)(s.Capitano=w.RingName \wedge w.StatoProvenienza='Canada' \wedge \\ \forall cs(COMPOSIZIONESTABLE)(cs.NomeStable=s.NomeStable \Rightarrow \\ \exists c(CINTURA)(cs.NomeComponente=c.Campione)))\}$$

Domanda 2 (8 punti).

- A. Riportare la definizione di BCNF.
- B. Data una relazione $R(A,B,C,D,E,F)$ e l'insieme di dipendenze funzionali $F = \{AB \rightarrow EF, A \rightarrow CD, C \rightarrow B, E \rightarrow F\}$, dire se F è in 3NF motivando la risposta e se non lo è decomporla in 3NF mostrando tutti i passaggi.
Dire se il risultato è in BCNF motivando la risposta.

Soluzione 2.

A. Vedere libro di testo o slide.

B. Per dire se R è in 3NF, trovo prima tutte le chiavi della relazione R . Dato che A non compare mai nei conseguenti, deve appartenere a ogni chiave. Possiamo vedere che $A+ = ABCDEF$, cioè tutti gli attributi di R , quindi A è superchiave. Dato che è una superchiave composta da un solo attributo, è già minimale, quindi è anche chiave. Non ci sono altre chiavi perché, come abbiamo osservato, dovrebbero contenere A , quindi non sarebbero superchiavi non minimali.

R non è in 3NF perché, ad es., $E \rightarrow F$ non è in 3NF perché non è banale, E non è superchiave e F non è primo.

Trasformo le d.f. mettendo un solo attributo a destra: $F' = \{AB \rightarrow E, AB \rightarrow F, A \rightarrow C, A \rightarrow D, C \rightarrow B, E \rightarrow F\}$.

Calcolo un insieme di copertura minimale.

Elimino gli attributi estranei. Dato che A è chiave, sicuramente in $AB \rightarrow E$ e in $AB \rightarrow F$ B è estraneo perché posso ricavare E e F con il solo attributo A . Le altre d.f. hanno un solo attributo a sinistra, quindi non hanno attributi estranei. Ottengo $F'' = \{A \rightarrow E, A \rightarrow F, A \rightarrow C, A \rightarrow D, C \rightarrow B, E \rightarrow F\}$.

Elimino le dipendenze ridondanti.

$A \rightarrow E$ è ridondante? Dato $F^* = \{A \rightarrow F, A \rightarrow C, A \rightarrow D, C \rightarrow B, E \rightarrow F\}$, controllo se $E \subseteq AF^*+$. Dato che E non compare nel conseguente di nessuna d.f. in F^* , E non sarà nella chiusura di A e quindi $A \rightarrow E$ non è ridondante. $A \rightarrow F$ è ridondante? Dato $F^* = \{A \rightarrow E, A \rightarrow C, A \rightarrow D, C \rightarrow B, E \rightarrow F\}$, controllo se $F \subseteq AF^*+$. Ricavo F da A con $A \rightarrow E$ e $E \rightarrow F$, quindi $A \rightarrow F$ è ridondante e la elimino.

$A \rightarrow C$ è ridondante? Dato $F^* = \{A \rightarrow E, A \rightarrow D, C \rightarrow B, E \rightarrow F\}$, controllo se $C \subseteq AF^*+$. Dato che C non compare nel conseguente di nessuna d.f. in F^* , sicuramente C non è contenuta nella chiusura di A, quindi $A \rightarrow C$ non è ridondante.

$A \rightarrow D$ è ridondante? Dato $F^* = \{A \rightarrow E, A \rightarrow C, C \rightarrow B, E \rightarrow F\}$, controllo se $D \subseteq AF^*+$. Dato che D non compare nel conseguente di nessuna d.f. in F^* , sicuramente D non è contenuto nella chiusura di A, quindi $A \rightarrow D$ non è ridondante.

$C \rightarrow B$ è ridondante? Dato $F^* = \{A \rightarrow E, A \rightarrow C, A \rightarrow D, E \rightarrow F\}$, controllo se $B \subseteq CF^*+$. Dato che C non compare nell'antecedente di nessuna d.f. in F^* , la chiusura contiene solo C stesso e non B, quindi $C \rightarrow B$ non è ridondante.

$E \rightarrow F$ è ridondante? Dato $F^* = \{A \rightarrow E, A \rightarrow C, A \rightarrow D, C \rightarrow B\}$, controllo se $F \subseteq EF^*+$. Dato che E non compare nell'antecedente di nessuna d.f. in F^* , la chiusura contiene solo E stesso e non F, quindi $E \rightarrow F$ non è ridondante. La copertura minimale è $\{A \rightarrow E, A \rightarrow C, A \rightarrow D, C \rightarrow B, E \rightarrow F\}$.

Decompongo in 3NF. Raggruppando per antecedente, ottengo: $R_1(A, C, D, E)$, $R_2(C, B)$, $R_3(E, F)$.

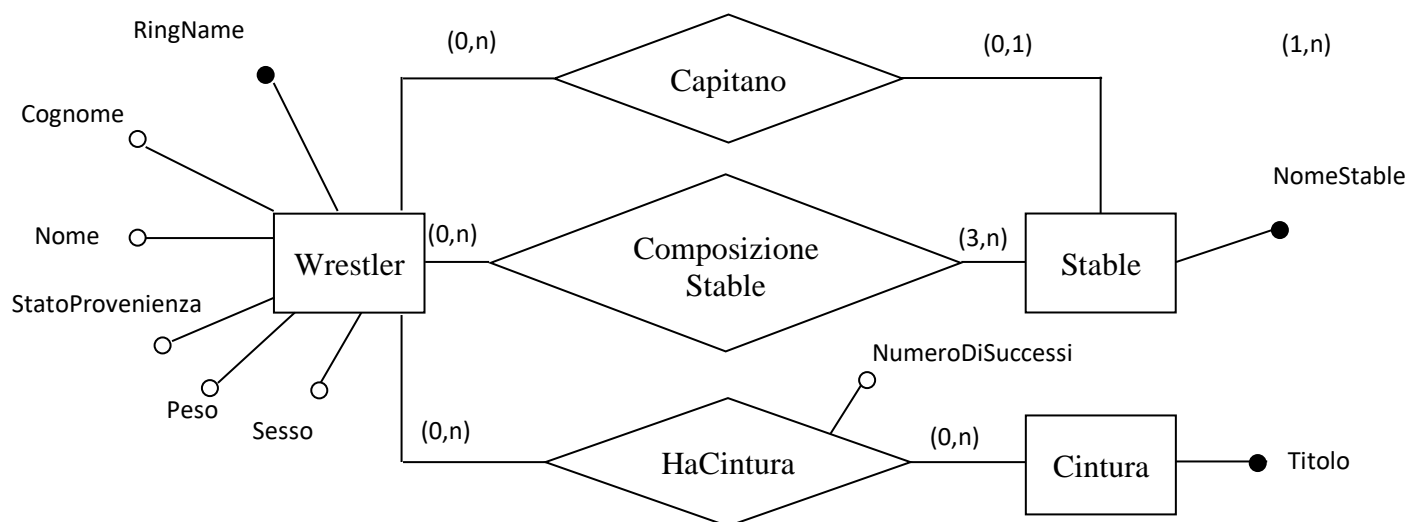
Nessuna relazione è sottoinsieme di un'altra. Inoltre R_1 contiene una chiave di R, quindi ho terminato la normalizzazione in 3NF.

Il risultato è in BCNF perché ogni d.f. ha come antecedente una chiave.

Domanda 3 (8 punti).

Progettare uno schema concettuale che rappresenti la base di dati "PWF Wrestling". Ricordarsi di riportare, sotto forma di regole aziendali, tutti i vincoli non esprimibili attraverso l'ER.

Soluzione 3.



Regole aziendali:

- $\text{Wrestler.Sesso} \in \{\text{"Uomo"}, \text{"Donna"}\}$
- $\text{Cintura.Titolo} \in \{\text{"assoluto"}, \text{"intercontinentale"}, \text{"femminile"}\}$
- Se un Wrestler partecipa all'associazione HaCintura con $\text{Cintura.Titolo} = \text{"femminile"}$ allora $\text{Wrestler.Sesso} = \text{"Donna"}$.
- Le occorrenze di Capitano sono un sottoinsieme delle occorrenze di ComposizioneStable (cioè un capitano di una stable deve fare parte della stable)

Domanda 4 (8 punti).

Con riferimento alla base di dati "PWF Wrestling", si considerino due indici IW e IS realizzati con B+-tree rispettivamente su $\text{wrestler(RingName)}$ e stable(Capitano) e i seguenti dati quantitativi:

$$\text{Nfoglie(IW)} = 25$$

$$\text{CARD(wrestler)} = 2000$$

$$\text{Npage(wrestler)} = 100$$

$$\text{Nfoglie(IS)} = 10$$

$$\text{CARD(stable)} = 6000$$

$$\text{Npage(stable)} = 15$$

$$\text{VAL(Capitano, stable)} = 500$$

Confrontare le stime dei costi della seguente interrogazione, già logicamente ottimizzata, eseguita sfruttando il join con nested loop con indice e con block-nested loop con $B=6$ (fornire sia le formule risolutive che i risultati numerici). Si tenga presente che un record di wrestler occupa circa 200 Byte.

$(\sigma_{\text{RingName}='John\ Cena'}(\text{wrestler})) \bowtie_{\text{RingName}=\text{Capitano}} \text{stable}$

Soluzione 4.

Il costo della selezione è dato da: $C(\sigma) = CI + CD$

$$CI = \lceil fp \cdot \text{Nfoglie(IW)} \rceil = \lceil 1/\text{CARD(wrestler)} \cdot \text{Nfoglie(IW)} \rceil = \lceil 1/2000 \cdot 25 \rceil = 1$$

$$CD = \min\{\text{Erec}, \text{Npage(wrestler)}\} = \min\{\lceil fp \cdot \text{CARD(wrestler)} \rceil, \text{Npage(wrestler)}\}$$

$$CD = \min\{\lceil 1/2000 \cdot 2000 \rceil, 100\} = \min\{1, 100\} = 1$$

$$C(\sigma) = CI + CD = 1 + 1 = 2 \text{ accessi}$$

Il costo del join (nested loop con indice) è dato da: $C(\bowtie) = \text{Npage}(\sigma) + \text{CARD}(\sigma) \cdot (CI + CD)$

$\text{CARD}(\sigma) = 1$ record perché selezioniamo usando la chiave primaria

$\text{Npage}(\sigma) = 1$ (una pagina occupa 4 KB / 8 KB, un record di wrestler occupa 200 B quindi è sufficiente una pagina per contenere il record).

Calcoliamo ora il costo del join con nested loop con indice:

$$CI = \lceil fp \cdot \text{Nfoglie(IS)} \rceil = \lceil 1/\text{VAL(Capitano, stable)} \cdot \text{Nfoglie(IS)} \rceil = \lceil 1/500 \cdot 10 \rceil = 1$$

$$CD = \min\{\text{Erec}, \text{Npage(stable)}\} = \min\{\lceil fp \cdot \text{CARD(stable)} \rceil, \text{Npage(stable)}\}$$

$$CD = \min\{\lceil 1/500 \cdot 6000 \rceil, 15\} = \min\{12, 15\} = 12$$

$$\text{Quindi: } C(\bowtie) = 1 + 1 \cdot (1 + 12) = 14$$

Il costo totale con nested loop con indice è quindi dato da: $C_{\text{tot}} = C(\sigma) + C(\bowtie) = 2 + 14 = 16$ accessi

Il costo del block-nested loop è dato da

$$C(\bowtie) = \text{Npage}(\sigma) + \lceil \text{Npage}(\sigma) / (B - 1) \rceil \cdot (\text{Npage}(\text{stable}))$$

$$C(\bowtie) = 1 + \lceil 1/5 \rceil \cdot 15 = 16$$

Il costo totale con block-nested loop è quindi $C_{\text{tot}} = C(\sigma) + C(\bowtie) = 2 + 16 = 18$ accessi

Il costo del join con nested loop con indice è inferiore al costo del join con nested loop con block-nested loop. L'ottimizzatore userà quindi il join con nested loop con indice.