

Basi di Dati – Corso B – Appello: 13 giugno 2018

Cognome, Nome _____

Matricola _____

Domanda 1

Le seguenti relazioni definiscono una base dati "Partite" per gestire partite di calcio tra squadre di città italiane. Gli attributi sottolineati sono le chiavi principali delle relazioni, l'attributo con * può avere valore nullo. Significato degli attributi non autoesplicativi: "CodF" = Codice fiscale di arbitro e giocatore, "PuntiSC" e "PuntiSO" indicano rispettivamente i punti realizzati dalla squadra che gioca in casa e quelli realizzati dalla squadra ospite. Una squadra vince se realizza più punti dell'altra oppure le due squadre pareggiano.

ARBITRO(CodF, Nome, DataNascita, CittàNascita, NazioneNascita, Categoria)

GIOCATORE(CodF, Nome, DataNascita, CittàNascita, NazioneNascita, GiocaIn*)

SQUADRA(NomeSquadra, Categoria, Città)

PARTITA(SquadraCasa, SquadraOspite, DataPartita, PuntiSC, PuntiSO, Arbitro)

Vincoli di integrità referenziale: "GiocaIn" referencia SQUADRA, "SquadraCasa" e "SquadraOspite" referenziano entrambi SQUADRA, "Arbitro" referencia ARBITRO.

Specificare la seguente interrogazione:

"Elencare le partite tra squadre di categoria differente arbitrate da un italiano"

A. in calcolo relazionale su tuple con dichiarazione di range

B. in algebra relazionale, evitando ridondanze

C. Rappresentare in notazione ad albero l'espressione in algebra relazionale fornita al punto precedente ed eseguirne l'ottimizzazione logica.

Soluzione 1

A. Calcolo relazionale

$$\{p.\text{SquadraCasa}, p.\text{SquadraOspite}, p.\text{DataPartita} \mid p(\text{PARTITA}) \mid \\ \exists s'(\text{SQUADRA}) \exists s''(\text{SQUADRA}) \exists a(\text{ARBITRO}) (s'.\text{NomeSquadra} = p.\text{SquadraCasa} \wedge \\ s''.\text{NomeSquadra} = p.\text{SquadraOspite} \wedge a.\text{CodF} = p.\text{Arbitro} \wedge s'.\text{Categoria} \neq s''.\text{Categoria})\}$$

B. Algebra relazionale:

$$\prod_{\text{SquadraCasa}, \text{SquadraOspite}, \text{DataPartita}} \sigma_{\text{NazioneNascita} = \text{'Italia'} \wedge \text{SC.Categoria} \neq \text{SO.Categoria}} \\ (\text{ARBITRO} \bowtie_{\Theta_3} (\rho_{\text{SC} \leftarrow \text{SQUADRA}}(\text{SQUADRA}) \bowtie_{\Theta_1} \text{PARTITA} \bowtie_{\Theta_2} \\ \rho_{\text{SO} \leftarrow \text{SQUADRA}}(\text{SQUADRA})))$$

Dove:

- $\Theta_1 = \text{SC.NomeSquadra} = \text{SquadraCasa}$
- $\Theta_2 = \text{SO.NomeSquadra} = \text{SquadraOspite}$
- $\Theta_3 = \text{CodF} = \text{Arbitro}$

C. Basta applicare l'algoritmo di ottimizzazione logica. Dopo aver applicato il passo 1 (decomposizione degli AND) è possibile trasferire le selezioni il più possibile verso le foglie (passo 2):

$$\prod_{\text{SquadraCasa}, \text{SquadraOspite}, \text{DataPartita}} (\sigma_{\text{NazioneNascita} = \text{'Italia'}} (\text{ARBITRO}) \bowtie_{\Theta_3} \sigma_{\text{SC.Categoria} \neq \text{SO.Categoria}} (\rho_{\text{SC} \leftarrow \text{SQUADRA}}(\text{SQUADRA}) \bowtie_{\Theta_1} \text{PARTITA} \bowtie_{\Theta_2} \\ \rho_{\text{SO} \leftarrow \text{SQUADRA}}(\text{SQUADRA})))$$

Bisogna però avanzare delle ipotesi per la soluzione del passo 7 dell'algoritmo (configurazione meno costosa dei join). Ad esempio, possiamo ipotizzare che: $\sigma_{\text{NazioneNascita} = \text{'Italia'}} (\text{ARBITRO}) \mid <$

$|SQUADRA| < |PARTITA|$. Di conseguenza, converrebbe anticipare \bowtie_{Θ_3} . L'espressione ottimizzata sarà quindi:

$$\prod_{SquadraCasa, SquadraOspite, DataPartita} (\sigma_{NazioneNascita='Italia'} (ARBITRO) \bowtie_{\Theta_3} (\sigma_{SC.Categoria \neq SO.Categoria} (\rho_{SC \leftarrow SQUADRA(SQUADRA)} \bowtie_{\Theta_1} PARTITA \bowtie_{\Theta_2} \rho_{SO \leftarrow SQUADRA(SQUADRA)})))$$

Domanda 2

Con riferimento alla base dati "Partite" specificare in SQL l'interrogazione:

“Elencare le squadre che non hanno mai giocato insieme ma hanno giocato con una terza stessa squadra almeno una volta e hanno perso entrambe contro tale terza squadra”.

Soluzione 2

```
SELECT s1.NomeSquadra, s2.NomeSquadra
FROM squadra s1 JOIN squadra s2 ON (s1.NomeSquadra < s2.NomeSquadra)
WHERE NOT EXISTS
  (SELECT * FROM partita p1
   WHERE (p1.SquadraCasa=s1.NomeSquadra AND p1.SquadraOspite=s2.NomeSquadra)
        OR (p1.SquadraCasa=s2.NomeSquadra AND p1.SquadraOspite=s1.NomeSquadra))
AND EXISTS
  (SELECT * FROM squadra s3
   WHERE s3.NomeSquadra<>s1.NomeSquadra AND s3.NomeSquadra<>s2.NomeSquadra
   AND EXISTS
     (SELECT * FROM partita p2
      WHERE (p2.SquadraCasa=s3.NomeSquadra AND
              p2.SquadraOspite=s1.NomeSquadra AND PuntiSC>PuntiSO) OR
            (p2.SquadraCasa=s1.NomeSquadra AND
              p2.SquadraOspite=s3.NomeSquadra AND PuntiSO>PuntiSC))
   AND EXISTS
     (SELECT * FROM partita p3
      WHERE (p3.SquadraCasa=s3.NomeSquadra AND
              P3.SquadraOspite=s3.NomeSquadra AND PuntiSC>PuntiSO) OR
            (p3.SquadraCasa=s3.NomeSquadra AND
              P3.SquadraOspite=s3.NomeSquadra AND PuntiSO>PuntiSC))));
```

Altre soluzioni sono possibili.

Domanda 3

Considerare la relazione:

ARBITRO(CodF, Nome, DataNascita, CittàNascita, NazioneNascita, Categoria)

sia data la dipendenza funzionale $f = (CittàNascita \twoheadrightarrow NazioneNascita)$

Tale dipendenza da sola non giustifica CodF (codice fiscale dell'arbitro) chiave della relazione.

Costruire un insieme di dipendenze funzionali minimale F che comprenda la dipendenza f data e tale che da F si possa concludere che CodF è chiave di ARBITRO. Motivare la minimalità di F.

Soluzione 3

Affinché CodF sia chiave, è necessario che $CodF^+ = \{CodF, Nome, DataNascita, CittàNascita, NazioneNascita, Categoria\}$. Devono quindi valere le d.f.

$f' = CodF \rightarrow Nome, DataNascita, CittàNascita, NazioneNascita, Categoria$

$f = CittàNascita \rightarrow NazioneNascita$

Per verificare la minimalità, costruiamo la forma canonica:

$F' = \{$
 $\underline{\text{CodF}} \rightarrow \text{Nome},$
 $\underline{\text{CodF}} \rightarrow \text{DataNascita},$
 $\underline{\text{CodF}} \rightarrow \text{CittàNascita},$
 $\underline{\text{CodF}} \rightarrow \text{NazioneNascita},$
 $\underline{\text{CodF}} \rightarrow \text{Categoria}$
 $\text{CittàNascita} \rightarrow \text{NazioneNascita}$
 $\}.$

Non ci sono attributi estranei, ma $\text{CodF} \rightarrow \text{NazioneNascita}$ è ridondante, in quanto deducibile per transitività da $\text{CodF} \rightarrow \text{CittàNascita}$ e $\text{CittàNascita} \rightarrow \text{NazioneNascita}$.

Pertanto l'insieme F che soddisfa la richiesta è

$F = \{$
 $\underline{\text{CodF}} \rightarrow \text{Nome},$
 $\underline{\text{CodF}} \rightarrow \text{DataNascita},$
 $\underline{\text{CodF}} \rightarrow \text{CittàNascita},$
 $\underline{\text{CodF}} \rightarrow \text{Categoria}$
 $\text{CittàNascita} \rightarrow \text{NazioneNascita}$
 $\}.$

Domanda 4

Sulla base delle dipendenze F della domanda 4 precedente, normalizzare in 3NF la relazione ARBITRO. Esplicitare i passi di normalizzazione eseguiti.

Soluzione 4

Dopo aver accorpato le d.f. di F con lo stesso antecedente, otteniamo le due relazioni:

$R1(\underline{\text{CodF}}, \text{Nome}, \text{DataNascita}, \text{CittàNascita}, \text{Categoria})$

$R2(\underline{\text{CittàNascita}}, \text{NazioneNascita})$

Domanda 5

A. Elencare e spiegare brevemente le proprietà ACID delle transazioni.

B. Definire la nozione di azioni in conflitto in una storia S .

C. Dire se la storia S che segue è view-serializzabile e perché.

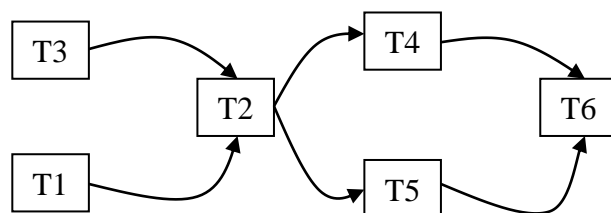
$r2(x), r3(x), r1(x), w2(x), r6(y), r5(x), r4(x), r5(y), r4(y), w6(y)$

Soluzione 5

A. Si vedano gli appunti/slide.

B. Si vedano gli appunti/slide

A. Il grafo dei conflitti è il seguente



- l'arco $T3 \rightarrow T2$ è dovuto a $r3(x)$ seguito da $w2(x)$
- l'arco $T1 \rightarrow T2$ è dovuto a $r1(x)$ seguito da $w2(x)$
- gli archi $T2 \rightarrow T5$ e $T2 \rightarrow T4$ sono dovuti a $w2(x)$ seguiti da $r5(x)$ ed $r4(x)$

- l'arco $T5 \rightarrow T6$ è dovuto a $r4(y)$ seguito da $w6(y)$
- l'arco $T4 \rightarrow T6$ è dovuto a $r4(y)$ seguito da $w6(y)$

Il grafo è aciclico, pertanto la storia S è view-serializzabile.

Domanda 6

- A. Descrivere il problema del **deadlock** avvalendosi del grafo di attesa.
- B. Presentare almeno una tecnica di **superamento** del deadlock.
- C. Presentare una tecnica di **prevenzione** del deadlock.

Soluzione 6

Si vedano gli appunti/slide.

Basi di Dati – Corso B – Appello: 6 luglio 2018

Cognome, Nome _____

Matricola _____

Domanda 1.

Si consideri la seguente base di dati (chiamata “Incarichi”) per la gestione degli incarichi da Primo Ministro della Repubblica Italiana:

PRIMOMINISTRO(Cognome, Nome, PartitoPolitico*, AnnoNascita)

LEGISLATURA(Numero, DataInizio, DataFine*)

INCARICO(CognomePM, NomePM, NumeroL, DataPreincarico, DataGiuramento*, Durata*)

La LEGISLATURA, della durata tipica di cinque anni, è caratterizzata da un numero, da una data di inizio e una data di fine legislatura.

Nella relazione INCARICO, CognomePM e NomePM costituiscono una chiave esterna su PRIMOMINISTRO, mentre NumeroL è in vincolo di integrità referenziale con LEGISLATURA. DataPreincarico è la data in cui il Primo Ministro riceve l’incarico (eventualmente esplorativo) da parte del Presidente della Repubblica. Se il preincarico (esplorativo) va a buon fine (cioè, il Primo Ministro individua una squadra di ministri e una maggioranza in parlamento), DataGiuramento contiene la data di effettiva formazione (corrispondente al giorno del giuramento) del governo presieduto dal Primo Ministro. Durata contiene invece la eventuale durata del mandato. Gli attributi asteriscati possono essere nulli. Quando PartitoPolitico è nullo, il Primo Ministro è un tecnico (ovvero, non appartenente ad alcun partito politico).

Con riferimento alla base dati “Incarichi” si richiede di rispondere alla seguente domanda in SQL:

Individuare la legislatura con il maggior numero di preincarichi a tecnici non andati a buon fine.

Soluzione 1

```
WITH num_incarichi AS (  
    SELECT NumeroL, COUNT(*) AS Num  
    FROM incarico JOIN primoministro ON (CognomePM=Cognome AND NomePM=Nome)  
    WHERE PartitoPolitico IS NULL AND DataGiuramento IS NULL  
    GROUP BY NumeroL)  
SELECT NumeroL  
FROM num_incarichi  
WHERE Num >= ALL (SELECT Num FROM num_incarichi);
```

Altre soluzioni sono possibili.

Domanda 2.

Con riferimento alla base dati “Incarichi” della Domanda 2, esprimere, sia in **algebra relazionale** che in **calcolo relazionale** su tuple con dichiarazione di range, l’interrogazione:

Indicare cognome, nome e anno di nascita del primo ministro che ha avuto l’incarico più lungo.

Soluzione 2

Calcolo relazionale

$$\{p.Cognome, p.Nome, p.AnnoNascita \mid p(PRIMOMINISTRO) \mid \\ \exists i(INCARICO)(i.CognomePM=p.Cognome \wedge i.NomePM=p.Nome \wedge \\ \neg \exists i'(INCARICO)(i'.Durata > i.Durata))\}$$

Algebra relazionale:

$$\prod_{\text{Cognome, Nome, AnnoNascita}} (\text{PRIMOMINISTRO}) \bowtie_{\Theta 2} (\sigma_{\text{Durata IS NOT NULL}} (\text{INCARICO}) - (\text{INCARICO} \bowtie_{\Theta 1} \rho_{I \leftarrow \text{INCARICO}} (\text{INCARICO})))$$

Dove:

- $\Theta 1$: $\text{INCARICO.Durata} < I.Durata$
- $\Theta 2$: $\text{Cognome} = \text{CognomePM} \wedge \text{Nome} = \text{NomePM}$

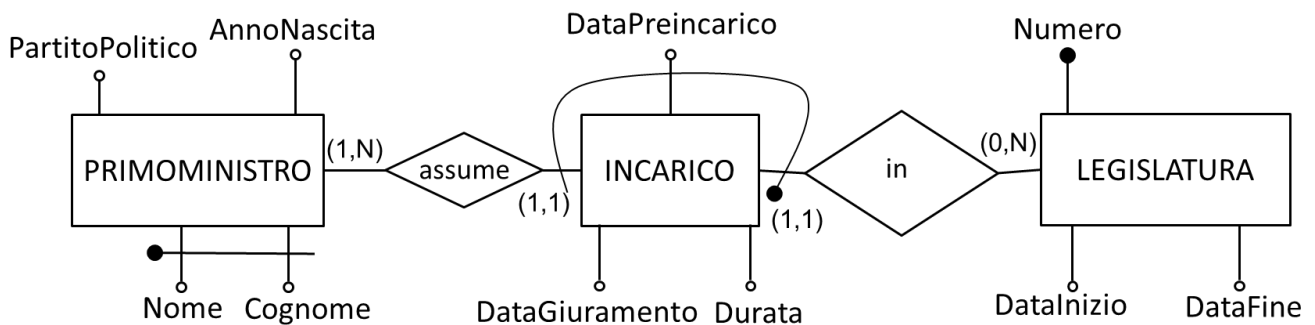
Domanda 3.

Progettare uno schema concettuale ER che specifichi la realtà considerata nella base dati "Incarichi" e insieme tenga conto che ciascun primo ministro può aver ricevuto più incarichi militando in partiti diversi. In tal modo si potrà avere anche l'informazione di quali partiti abbiano avuto un preincarico e, in caso, un incarico (è sufficiente presentare lo schema concettuale ER e le eventuali regole aziendali, **motivando le scelte**).

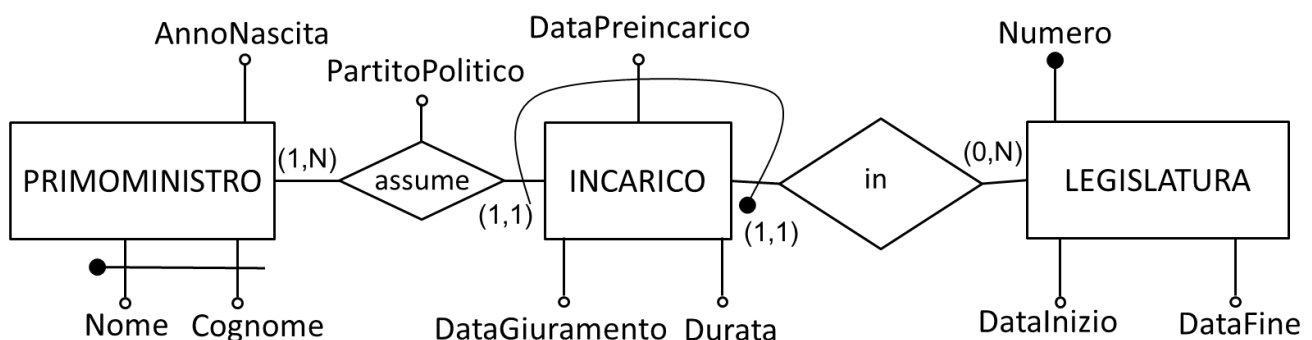
Non si chiede di considerare situazioni complesse quali cambiamento di partito e ritorno al medesimo o, addirittura di cambio partito tra preincarico e incarico. Più preincarichi o incarichi in una legislatura sono da considerare possibili.

Soluzione 3

Lo schema ER della domanda 1 è il seguente



Per soddisfare i requisiti, la soluzione più semplice è quella di “spostare” l’attributo PartitoPolitico sull’associazione “assume”:



Ma ovviamente, altre soluzioni più complesse sono possibili.

Regole aziendali:

R1: Se $\text{Legislatura.DataFine}$ IS NOT NULL, allora $\text{Legislatura.DataFine} > \text{Legislatura.DataInizio}$

R2: Se $\text{Incarico.DataGiuramento}$ IS NOT NULL, allora $\text{Incarico.DataGiuramento} > \text{Incarico.DataPreincarico}$

R3: Gli attributi PartitoPolitico, DataGiuramento, Durata e DataFine possono essere NULL

Domanda 4.

- A. Esprimere la condizione necessaria e sufficiente affinché una scomposizione sia senza perdita di informazione.
- B. Dire se la seguente relazione (con le sue dipendenze funzionali) è in 3FN e, se non lo è, effettuare la scomposizione in 3FN **esplicitando tutti i passaggi**:

$R(A,C,E,G,I,M,O,Q)$ con $F=\{CEG \rightarrow IM, CE \rightarrow IMQ, M \rightarrow GEO, CE \rightarrow O\}$

Soluzione 4

A. Si vedano gli appunti/slide.

B. Per prima cosa è necessario identificare la o le chiavi della relazione. L'attributo A non compare in nessuna dipendenza funzionale, quindi farà sicuramente parte della chiave. Degli altri attributi, I, O e Q compaiono sempre e solo a destra. Possiamo provare quindi, ad esempio, con CE:

$ACE^+ = \{ACEIMQGEO\}$ (usando nell'ordine $CE \rightarrow IMQ$ e $M \rightarrow GEO$).

I sottoinsiemi non sono chiavi ($AC^+ = \{AC\}$ e $AE^+ = \{AE\}$) quindi ACE^+ è chiave.

ATTENZIONE: anche ACM è chiave, infatti:

$ACM^+ = \{ACMGEOIQ\}$ (usando nell'ordine $M \rightarrow GEO$ e $CE \rightarrow IMQ$).

e il sottoinsieme stretto AM non è chiave, infatti:

$AM^+ = \{AMGEO\}$.

Di conseguenza, nessuna delle dipendenze funzionali è del tipo 1, 2 o 3, e la relazione non è in 3FN.

Per normalizzare, calcoliamo prima la copertura minimale dell'insieme F. Portiamo l'insieme in forma canonica:

$F' = \{$
 $CEG \rightarrow I,$
 $CEG \rightarrow M,$
 $CE \rightarrow I,$
 $CE \rightarrow M,$
 $CE \rightarrow Q,$
 $M \rightarrow G,$
 $M \rightarrow E,$
 $M \rightarrow O,$
 $CE \rightarrow O$
 $\}$

Identifichiamo gli attributi estranei. Nelle due dipendenze $CEG \rightarrow I$ e $CEG \rightarrow M$, G è estraneo, perché abbiamo $CE \rightarrow I$ e $CE \rightarrow M$. Non esistono altri attributi estranei, quindi:

$F' = \{$
 $CE \rightarrow I,$
 $CE \rightarrow M,$
 $CE \rightarrow Q,$
 $M \rightarrow G,$
 $M \rightarrow E,$
 $M \rightarrow O,$
 $CE \rightarrow O$
 $\}$

Eliminiamo le dipendenze ridondanti: $CE \rightarrow O$ è ridondante in quanto deducibile per transitività da $CE \rightarrow M$ ed $M \rightarrow O$. Non ci sono altre ridondanze. Quindi:

$F' = \{$
 $CE \rightarrow I,$
 $CE \rightarrow M,$
 $CE \rightarrow Q,$
 $M \rightarrow G,$
 $M \rightarrow E,$
 $M \rightarrow O,$
 $\}$

Accorpiamo le dipendenze con lo stesso antecedente:

$F' = \{$
 $CE \rightarrow IMQ,$
 $M \rightarrow GEO$
 $\}$

E scomponiamo la relazione R in:

$R1(\underline{CE}IMQ)$

$R2(\underline{M}GEO)$

Le chiavi ACE o ACM non sono contenute in nessuna delle relazioni, quindi va aggiunta una nuova relazione R3, per esempio:

$R3(\underline{ACE}).$

Domanda 5.

Con riferimento allo schema logico della Domada 1, considerare due indici secondari **IPM** ed **IINC** (entrambi con struttura B+Albero) definiti, rispettivamente, sull'attributo AnnoNascita di PRIMOMINISTRO e sulla coppia di attributi CognomePM e NomePM di INCARICO. Si considerino inoltre i seguenti dati quantitativi:

$N_{foglie}(IPM) = 9$

$CARD(PRIMOMINISTRO) = 60$

$N_{page}(PRIMOMINISTRO) = 4$

$MIN(AnnoNascita, PRIMOMINISTRO) = 1880$

$MAX(AnnoNascita, PRIMOMINISTRO) = 1970$

$N_{foglie}(IINC) = 5$

$CARD(INCARICO) = 100$

$N_{page}(INCARICO) = 4$

$VAL(CognomePM, INCARICO) = 10$

$VAL(NomePM, INCARICO) = 10$

Calcolare la stima del costo della seguente interrogazione (già logicamente ottimizzata) eseguita sfruttando tutti gli indici a disposizione (fornire sia le formule risolutive che i risultati numerici). Si tenga presente che una tupla di PRIMOMINISTRO occupa circa 250 Byte.

$(\sigma_{AnnoNascita > 1940}(PRIMOMINISTRO)) \bowtie_{Nome=NomePM \wedge Cognome=CognomePM} INCARICO$

Soluzione 5.

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

Il fattore di selettività della selezione è:

$$fp = (\text{MAX}(\text{AnnoNascita}, \text{PRIMOMINISTRO}) - 1940) / (\text{MAX}(\text{AnnoNascita}, \text{PRIMOMINISTRO}) - \text{MIN}(\text{AnnoNascita}, \text{PRIMOMINISTRO})) = (1970 - 1940) / (1970 - 1880) = 1/3.$$

$$CI = \lceil fp \cdot \text{Nfoglie}(\text{IPM}) \rceil = \lceil 1/3 \cdot 9 \rceil = 3$$

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

$$CD = \min\{\lceil 1/3 \cdot 60 \rceil, 4\} = \min\{20, 4\} = 4$$

$$C(\sigma) = CI + CD = 3 + 4 = 7$$

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

$$\text{CARD}(\sigma) = fp \cdot \text{CARD}(\text{PRIMOMINISTRO}) = 1/3 \cdot 60 = 20$$

$\text{Npage}(\sigma) = 1$ (una pagina occupa 4KB/8KB, una tupla di PRIMOMINISTRO occupa 250B quindi sono sufficienti due pagine da 4KB, o una pagina da 8KB, per contenere le 20 tuple).

Calcoliamo ora il costo del join, partendo dal fattore di selettività del predicato

“Nome=NomePM \wedge Cognome=CognomePM”:

$fp = fp1 \cdot fp2$, dove:

$$fp1 = 1/\text{VAL}(\text{CognomePM}, \text{INCARICO}) = 1/10$$

$$fp2 = 1/\text{VAL}(\text{NomePM}, \text{INCARICO}) = 1/10$$

$$\text{Quindi, } fp = 1/10 \cdot 1/10 = 1/100$$

$$CI = \lceil fp \cdot \text{Nfoglie}(\text{IINC}) \rceil = \lceil 1/100 \cdot 5 \rceil = 1$$

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

$$CD = \min\{\lceil 1/100 \cdot 100 \rceil, 4\} = \min\{1, 4\} = 1$$

$$\text{Quindi: } C(\bowtie) = 2 + 20 \cdot (1 + 1) = 42$$

Il costo totale è quindi dato da: $C_{\text{tot}} = C(\sigma) + C(\bowtie) = 7 + 42 = 49$

Domanda 6.

A. Dire cos'è e quali problemi risolve il protocollo **2PL (due fasi) stretto**.

B. Dire in cosa consiste la tecnica del **dump/restore** (ripresa a freddo) e quando si rende necessaria.

Soluzione 6.

Si vedano gli appunti/slide.

Basi di Dati – Corso B – Appello: 11 settembre 2018

Cognome, Nome _____

Matricola _____

Domanda 1.

Considerare la base dati “Cinguettii” contenente dati relativi ad un sito di microblogging con il seguente schema

UTENTE(IDUtente, Nome, Città)

SEGUACE(IDUtente1, IDUtente2, DataS)

CINGUETTIO(IDCinguettio, IDUtente, Testo, DataC, Ricinguettio)

Ogni tupla di SEGUACE rappresenta la seguente informazione: l’utente identificato da IDUtente1 inizia a seguire l’utente identificato da IDUtente2 nella data definita da DataS.

Ogni tupla di CINGUETTIO rappresenta un “micropost”, pubblicato nella data definita da DataC, dall’utente identificato da IDUtente. Se un micropost A cita un altro micropost B, l’attributo Ricinguettio contiene l’identificativo del micropost B, altrimenti è nullo.

Inoltre, valgono i seguenti vincoli di integrità referenziale: IDUtente1 e IDUtente2 in SEGUACE referenziano UTENTE. IDUtente in CINGUETTIO referencia UTENTE. Ricinguettio in CINGUETTIO referencia CINGUETTIO.

Produrre un’espressione in SQL che esprima la domanda:

“Elencare gli utenti (ID, nome e città) che hanno iniziato a seguire un altro utente dopo averne ricinguettato almeno un micropost”

Soluzione 1

```
SELECT *
FROM utente
WHERE IdUtente in
    (SELECT S.IdUtente1
     FROM seguace S JOIN cinguettio C1 ON (S.IdUtente1=C1.IdUtente)
     JOIN cinguettio C2 ON (C1.IdUtente2=C2.IdUtente AND
                          C1.Ricinguettio=C2.IdCinguettio)
     WHERE ( C1.DataC < S.DataS )
    );
```

Altre soluzioni sono possibili.

Domanda 2.

Facendo riferimento alla base dati “Cinguettii”:

A. Scrivere una espressione in algebra relazionale e in calcolo relazionale che esprima la seguente domanda:

Elencare gli utenti che seguono solo utenti che non hanno mai pubblicato (né ricinguettato) alcun post.

B. Dire se è possibile rispondere con una singola espressione in algebra relazionale (ma non è necessario fornire l’eventuale espressione algebrica) alla seguente domanda: **dire se esiste una catena di seguaci che collegano l’utente “@Macron” all’utente “@Tsipras” o viceversa**. Motivare la risposta.

Soluzione 2

A. Algebra relazionale:

$$\prod_{IDUtente1} (SEGUACE) - \prod_{IDUtente1} (SEGUACE \bowtie_{IDUtente2=IDUtente1} CINGUETTIO)$$

Calcolo relazionale:

$$\{u.* \mid u(UTENTE) \mid \exists s(SEGUACE)(s.IDUtente1=u.IDUtente) \wedge \\ \forall s'(SEGUACE)(s'.IDUtente1 = u.IDUtente) \Rightarrow \neg \exists c(CINGUETTIO)(c.IDUtente=s'.IDUtente2)\}$$

B. Non è possibile in quanto l'algebra relazionale tradizionale non ammette la ricorsione né la chiusura transitiva.

Domanda 3.

Data una relazione $R(A,B,C,D,E,F,G)$ e l'insieme delle dipendenze funzionali

$$F = \{AB \rightarrow EF, B \rightarrow G, A \rightarrow CDG, C \rightarrow B, E \rightarrow F\}$$

- A. trovare le chiavi della relazione R
- B. dare la definizione di insieme di copertura minimale
- C. calcolare una copertura minimale di F
- D. dire se R è in 3FN e se non lo è scomporla

Soluzione 3

- A. L'unica chiave della relazione R è A, infatti $A^+ = \{A|CDG|B|EF\}$.
- B. Si vedano gli appunti/slide.
- C. Forma canonica:

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

Attributi estranei:

l'unico attributo estraneo è B (per transitività da $A \rightarrow C$ e $C \rightarrow B$. Quindi:

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

Dipendenze ridondanti:

$A \rightarrow F$ è ridondante per transitività ($A \rightarrow E$ ed $E \rightarrow F$)

$A \rightarrow G$ è ridondante per transitività ($A \rightarrow C$, $C \rightarrow B$ e $B \rightarrow G$)

Quindi, la copertura minimale di F è:

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

- D. R non è in 3FN (ad esempio $B \rightarrow G$ non è né d.f. superchiave né d.f. con attributi primi).

Partiamo dal risultato dell'esercizio C, accorpando le d.f. con stesso antecedente:

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

E otteniamo le relazioni:

$R1(\underline{A}CDE)$, $R2(\underline{B}G)$, $R3(\underline{C}B)$ ed $R4(\underline{E}F)$.

Domanda 4.

- A.** A proposito di gestione della concorrenza descrivere il protocollo di lock a due fasi (2PL) e quello di lock a due fasi stretto.
- B.** La storia $S = r1(x), r1(y), r2(x), w3(z), w1(y), r2(y), w2(x)$ è compatibile con 2PL stretto? Motivare la risposta.

Soluzione 4

Per la prima parte della domanda, si rimanda alle slide o agli appunti.

La storia $S1$ è compatibile con il 2PL. Infatti, una possibile realizzazione, con i relativi lock e unlock, è la seguente:

$S1 = \text{LS1}(x), r1(x), \text{LS1}(y), r1(y), \text{LS2}(x), r2(x), \text{LX3}(z), w3(z), \text{UN3}(z), \text{LX1}(y), w1(y), \text{UN1}(x),$
 $\text{UN1}(y), \text{LS2}(y), r2(y), \text{LX2}(x), w2(x), \text{UN2}(y), \text{UN2}(x)$

Domanda 5.

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

```
<T1, START>;
<T2, START>;
<T1, BS(t1[A], 5), AS(t1[A], 10)>;
<T2, BS(t2[B], 3), AS(t2[B], 5)>;
<T3, START>;
<T2, COMMIT>;
-- checkpoint --;
<T3, BS(t3[C], 3), AS(t3[C], 5)>;
<T3, ABORT>
```

crash!

- A.** Quali politiche di gestione del buffer adotta il DBMS che ha generato il log?
- B.** Indicare il contenuto del record di checkpoint.
- C.** Descrivere l'algoritmo di ripristino corrispondente alle politiche del buffer descritte nella risposta alla domanda A e mostrarne l'esecuzione sul log L.

Soluzione 5

A. Nel log troviamo sia i before-state che gli after-state. Si può quindi concludere che il DBMS adotta le politiche di gestione del buffer steal/no flush.

B. Il record di checkpoint conterrà la lista delle transazioni non terminate (quindi, T1 e T3) con i relativi puntatori ai rispettivi $\langle Ti, \text{START} \rangle$. Contiene inoltre l'OK riguardante la buona riuscita delle operazioni previste dal checkpoint, quindi: $[(T1, p1); (T3, p3); \text{OK}]$

C. Per la descrizione dell'algoritmo si rimanda agli appunti/slide. Si noti che le liste LA e LC sono ricostruite a partire dal record di checkpoint.

La lista delle transazioni attive al momento del crash è $LA = \{T1\}$ (T3 è in abort). La lista delle transazioni in commit prima del crash è vuota, in quanto le modifiche effettuate dalla transazione T2 sono state rese persistenti dalla procedura di checkpoint.

Viene effettuato quindi l'UNDO di T1: viene ripristinato a '5' il valore dell'attributo A della tupla t1 (si effettua contestualmente anche un FORCE della pagina).

Basi di Dati – Corso B – Appello: 10 dicembre 2018

Cognome, Nome _____

Matricola _____

Domanda 1.

Si consideri la seguente base di dati (chiamata “Incarichi”) per la gestione degli incarichi da Primo Ministro della Repubblica Italiana:

POLITICO(Cognome, Nome, DataNascita)

ADESIONE(Cognome, Nome, NomePartito, DataAdesione)

PARTITO(NomePartito, Area*, DataFondazione)

PRIMOMINISTRO(Cognome, Nome, DataIncarico, Durata*)

Nella relazione PRIMOMINISTRO, *Cognome* e *Nome* costituiscono una chiave esterna su POLITICO. *DataIncarico* è la data in cui il politico avente *Cognome* e *Nome* riceve l’incarico (eventualmente esplorativo) da parte del Presidente della Repubblica. *Durata* contiene invece la eventuale durata (in giorni) del mandato ed è null se l’incarico non è andato a buon fine (cioè, se il politico non ha individuato una squadra di ministri e una maggioranza in parlamento).

In ADESIONE l’attributo *NomePartito* è una chiave esterna su PARTITO. *DataAdesione* indica invece la data in cui il politico *Cognome*, *Nome* ha aderito al partito politico indicato in *NomePartito*. Un politico **non può** appartenere a più partiti contemporaneamente.

In PARTITO l’attributo *Area* indica l’area di appartenenza del partito (destra, centro-destra, centro, centro-sinistra, sinistra). Se è null, il partito non fa riferimento a nessuna area politica tradizionale.

Si suppone che un Primo Ministro non possa cambiare partito politico durante il suo incarico.

Con riferimento alla base dati “Incarichi” si richiede di esprimere in SQL la seguente richiesta:

Individuare l’area politica tradizionale che ha avuto il maggior numero di politici incaricati che hanno effettivamente governato, cioè i cui incarichi sono andati a buon fine.

Soluzione 1.

Una possibile soluzione è la seguente:

```
with PrimoMinSuaArea as (  
    select distinct(p.Cognome, p.Nome, p.Area)  
    from PRIMOMINISTRO pm join POLITICO p using (Cognome, Nome)  
    join ADESIONE a using (Cognome, Nome)  
    join PARTITO pp using (NomePartito)  
    where (pm.Durata is not null)  
    and (pp.Area is not null)  
    and (pm.DataIncarico>=a.DataAdesione)  
    and NOT EXISTS (select *  
                     from ADESIONE a2  
                     where (a2.DataAdesione>a.DataAdesione)  
                           and (a2.DataAdesione<=pm.DataIncarico))  
),  
with AreaQuantipM as (  
    select Area, count(*) as NroIncSucc  
    from PrimoMinSuaArea
```

```

        group by Area
    ),
    select Area, NroIncSucc
    from AreaQuantipM
    where NroIncSucc = (select max(NroIncSucc)
                        from AreaQuantipM);

```

Domanda 2.

Con riferimento alla base dati “**Incarichi**” della Domanda 1, esprimere, in **algebra relazionale** e in **calcolo relazionale** su tuple con dichiarazione di range, l'interrogazione:

Indicare cognome, nome e data di nascita dei politici che hanno aderito a partiti di area diversa.

Soluzione 2.

Una possibile soluzione in algebra relazionale è la seguente:

$$\begin{aligned}
 & \text{POLITICO} \bowtie_{\Theta_1} \\
 & (\rho_{A1 \leftarrow \text{ADESIONE}} (\text{ADESIONE}) \bowtie_{\Theta_2} \rho_{PP1 \leftarrow \text{PARTITOPOLITICO}} (\text{PARTITO})) \\
 & \quad \bowtie_{\Theta_3} \\
 & (\rho_{A2 \leftarrow \text{ADESIONE}} (\text{ADESIONE}) \bowtie_{\Theta_4} \rho_{PP2 \leftarrow \text{PARTITOPOLITICO}} (\text{PARTITO}))
 \end{aligned}$$

Dove:

- Θ_1 : $\text{POLITICO.Cognome} = A1.Cognome \wedge \text{POLITICO.Nome} = A1.Nome$
- Θ_2 : $A1.NomePartito = PP1.NomePartito$
- Θ_3 : $(PP1.Area \neq PP2.Area) \vee (PP1.Area \text{ IS NULL} \wedge PP2.Area \text{ IS NOT NULL})$
 $\vee (PP1.Area \text{ IS NOT NULL} \wedge PP2.Area \text{ IS NULL}) \wedge$
 $(A1.Cognome = A2.Cognome \wedge A1.Nome = A2.Nome)$
- Θ_4 : $A2.PartitoPolitico = PP2.PartitoPolitico$

Una possibile soluzione in calcolo relazionale è la seguente:

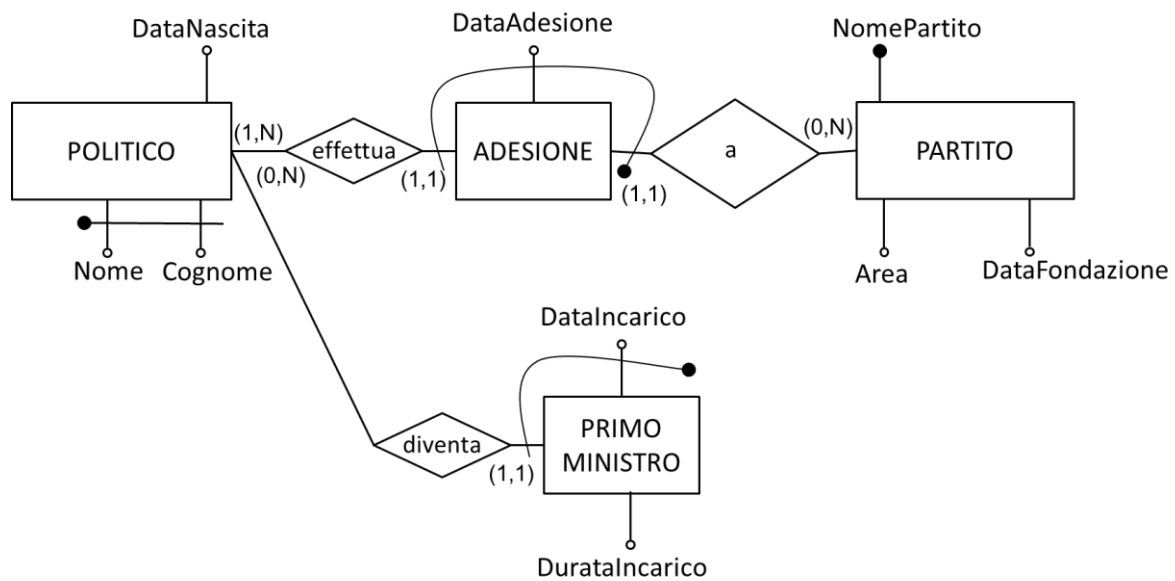
$$\begin{aligned}
 & \{ p.* \mid p(\text{POLITICO}) \mid \exists a(\text{ADESIONE}) \exists pp(\text{PARTITO}) \exists a'(\text{ADESIONE}) \exists pp'(\text{PARTITO}) \\
 & \quad (a.Cognome = p.Cognome \wedge a.Nome = p.Nome \wedge a.NomePartito = pp.NomePartito) \wedge \\
 & \quad (a'.Cognome = a.Cognome \wedge a'.Nome = a.Nome \wedge a'.NomePartito = pp'.NomePartito) \wedge \\
 & \quad (pp.Area \neq pp'.Area) \vee (pp.Area \text{ IS NULL} \wedge pp'.Area \text{ IS NOT NULL}) \\
 & \quad \vee (pp.Area \text{ IS NOT NULL} \wedge pp'.Area \text{ IS NULL}) \}
 \end{aligned}$$

Domanda 3.

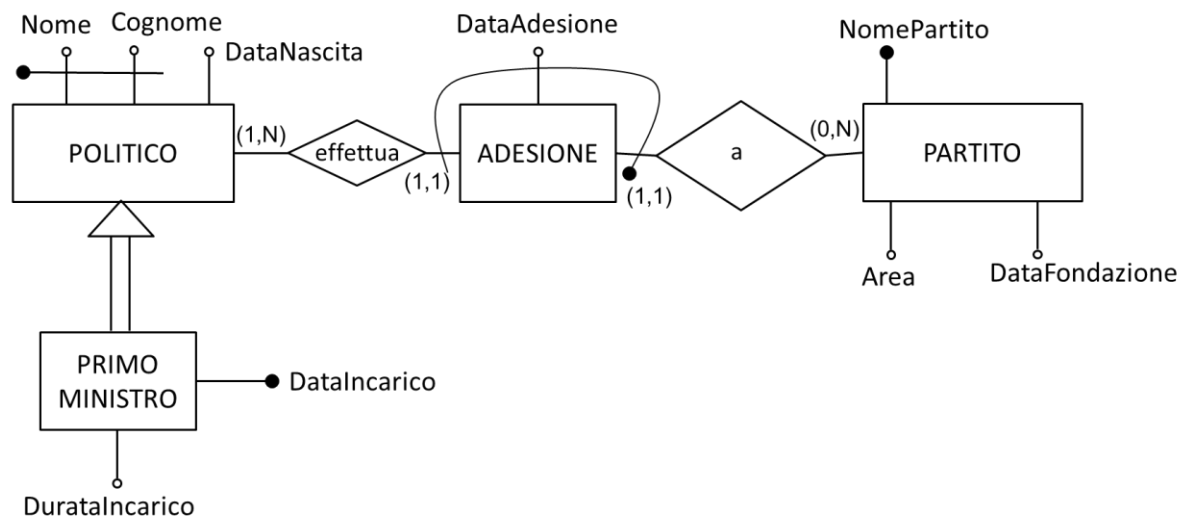
Progettare uno schema concettuale ER che specifichi la realtà considerata nella base dati "Incarichi" della Domanda 1. Elencare, sotto forma di regole aziendali, tutti i vincoli non esprimibili attraverso l'ER.

Soluzione 3.

Una possibile soluzione è la seguente:



oppure



con i seguenti vincoli e regole aziendali:

- V1. $PARTITO.Area \in \{destra, centro-destra, centro, centro-sinistra, sinistra\}$
- R1. Un politico, nel periodo in cui è in carica come primo ministro, non può cambiare partito.
- R2. Un politico non può aderire a più partiti contemporaneamente.
- R3. Se $PRIMOMINISTRO.Durata$ IS NULL, l'incarico non è andato a buon fine
- R4. Se $PARTITO.Area$ IS NULL, il partito non appartiene a nessun'area politica tradizionale.

Domanda 4.

Si consideri la relazione ERBARIO relativa alla classificazione in un erbario di fiori vari che crescono in Italia col loro periodo di fioritura. Chiave di ERBARIO è la coppia {Codice, Provincia} mentre F è l'insieme di dipendenze funzionali. Dire se ERBARIO è in 3FN e se non lo è scomporla.

ERBARIO(Codice, Provincia, NomeScientifico, DataClassif, Scopritore, PeriodoFiore, LinkScopritore)

con $F = \{ \text{Codice} \rightarrow \text{NomeScientifico}, \text{DataClassif}, \text{Scopritore}, \text{PeriodoFiore};$

$\text{Scopritore} \rightarrow \text{LinkScopritore};$

$\text{NomeScientifico} \rightarrow \text{Codice}, \text{DataClassif}, \text{Scopritore}, \text{PeriodoFiore} \}$

Soluzione 4.

La relazione ERBARIO non è in 3FN, in quanto nessuna delle dipendenze funzionali di F è in 3NF.

Per procedere alla sintesi in 3FN, bisogna prima calcolare la copertura minimale di F. A tal fine si deve prima porre F in forma canonica:

$$F = \begin{cases} F1: \text{Codice} \rightarrow \text{NomeScientifico}; \\ F2: \text{Codice} \rightarrow \text{DataClassif}; \\ F3: \text{Codice} \rightarrow \text{Scopritore}; \\ F4: \text{Codice} \rightarrow \text{PeriodoFiore}; \\ F5: \text{Scopritore} \rightarrow \text{LinkScopritore}; \\ F6: \text{NomeScientifico} \rightarrow \text{Codice}; \\ F7: \text{NomeScientifico} \rightarrow \text{DataClassif}; \\ F8: \text{NomeScientifico} \rightarrow \text{Scopritore}; \\ F9: \text{NomeScientifico} \rightarrow \text{PeriodoFiore} \end{cases}$$

Ovviamente non esistono attributi estranei. Esistono però dipendenze ridondanti. A seconda dell'attributo ponte considerato (Codice o NomeScientifico) si ottengono soluzioni diverse. Considerando Codice, ad esempio, F7, F8 ed F9 diventano ridondanti per transitività. L'insieme di copertura minimale diventa quindi:

$$F' = \begin{cases} F1: \text{Codice} \rightarrow \text{NomeScientifico}; \\ F2: \text{Codice} \rightarrow \text{DataClassif}; \\ F3: \text{Codice} \rightarrow \text{Scopritore}; \\ F4: \text{Codice} \rightarrow \text{PeriodoFiore}; \\ F5: \text{Scopritore} \rightarrow \text{LinkScopritore}; \\ F6: \text{NomeScientifico} \rightarrow \text{Codice}; \end{cases}$$

e accorpendo tutte le d.f. con lo stesso antecedente:

$$F' = \begin{cases} F1: \text{Codice} \rightarrow \text{NomeScientifico, DataClassif, Scopritore, PeriodoFiore}; \\ F2: \text{Scopritore} \rightarrow \text{LinkScopritore}; \\ F3: \text{NomeScientifico} \rightarrow \text{Codice}; \end{cases}$$

Si può quindi procedere alla scomposizione, ottenendo le seguenti relazioni in 3FN:

R1(Codice, NomeScientifico, DataClassif, Scopritore, PeriodoFiore)

R2(Scopritore, LinkScopritore)

R3(NomeScientifico, Codice)

cui bisogna aggiungere la relazione contenente la chiave (Provincia,Codice):

R4(Provincia, Codice)

Infine, è possibile accorpare R1 ed R3, in quanto R3 è contenuta in R1. La soluzione è quindi:

R1(Codice, NomeScientifico, DataClassif, Scopritore, PeriodoFiore)

con F1={ NomeScientifico \rightarrow Codice }

R2(Scopritore, LinkScopritore)

R4(Provincia, Codice)

Domanda 5.

- A. Esporre la differenza tra indici densi ed indici sparsi.
- B. Indicare almeno **due casi** in cui gli indici secondari si dimostrano inefficienti.

Soluzione 5.

Si vedano gli appunti/slide/testo.