

Basi di Dati, 17/12/2003, secondo compitino, compito A

1) Si consideri lo schema relazionale R(A,B,C,D,E) con le seguenti DF:

$A \rightarrow BC$, $CB \rightarrow A$, $CD \rightarrow A$, $D \rightarrow B$

- Si trovino le chiavi di R.
- Dire se lo schema è in 3FN o in FNBC.
- Si applichi allo schema l'algoritmo di sintesi, si specifichi se il risultato preserva dati e dipendenze, e si dica se lo schema è in FNBC.
- (Opzionale) Si applichi allo schema l'algoritmo di analisi, e si specifichi se il risultato preserva dati e dipendenze.

2) Si consideri lo schema relazionale:

CorsiDiLaurea(CorsoLaurea: string, Facoltà: string, TipoLaurea: string)

Studenti(Matricola: integer, SNome: string, CorsoLaurea*: string)

Iscrizioni(Matricola*: integer, SiglaC*: string)

Corsi(SiglaC: string, OraRicevimento: time, DocenteId*: integer)

Docenti(DocenteId: integer, DNome: string, Dipartimento: string)

Si scrivano le seguenti interrogazioni in SQL

- Trovare, senza duplicati, il DocenteId dei Corsi che non sono frequentati da nessuno studente
- Trovare la sigla dei corsi seguiti solo da studenti che appartengono al CorsoLaurea 'Informatica'
- Di ogni studente della Facoltà 'Scienze', trovare la matricola ed il numero di corsi che segue
- Trovare nome e DocenteId dei docenti che insegnano qualche corso seguito da più di 5 studenti
- (Opz.) Trovare la sigla dei corsi che sono frequentati da tutti gli studenti della facoltà di 'Scienze'

3) Si consideri la relazione Studenti(Matricola, Nome, AnnoNascita), ordinata sulla chiave primaria Matricola, e l'interrogazione

```
SELECT DISTINCT Matricola, count(*)
```

```
FROM Studenti
```

```
WHERE AnnoNascita = 1974
```

```
GROUP BY Matricola
```

```
ORDER BY Matricola;
```

a) Disegnare l'albero di sintassi astratta di un'espressione algebrica ("albero logico") per l'interrogazione

b/c): Si dica se il piano d'accesso a destra produce il risultato cercato. Se non va bene, lo si modifichi in due modi: (b) aggiungendo prima solo le parti mancanti (operatori e parametri) e (c) semplificando poi il piano eliminando operatori inutili:

```
Sort({Matricola } )
|
Distinct
|
Sort({Matricola} )
|
Project({Matricola})
|
GroupBy({Matricola})
|
TableScan( Studenti )
```

4) (Opzionale) Si consideri il seguente schema, che rappresenta informazioni relative agli spettacoli programmati per una stagione in un insieme di teatri:

Spettacoli(Compagnia, Regista, Titolo, Data, Teatro)

Si esprimano i seguenti vincoli come dipendenze funzionali, se possibile:

- Due spettacoli contemporanei hanno il regista diverso
- Non è possibile avere due spettacoli diversi nella stessa data e nello stesso teatro
- Se due spettacoli hanno il regista diverso anche le compagnie sono diverse

Basi di Dati, 17/12/2003, secondo compitino, compito B

1) Si consideri lo schema relazionale $R(A,B,C,D,E)$ con le seguenti DF:

$C \rightarrow DE, ED \rightarrow C, EA \rightarrow C, A \rightarrow D$

- Si trovino le chiavi di R .
 - Dire se lo schema è in 3FN o in FNBC.
 - Si applichi allo schema l'algoritmo di sintesi, si specifichi se il risultato preserva dati e dipendenze, e si dica se lo schema è in FNBC.
 - (Opzionale) Si applichi allo schema l'algoritmo di analisi, e si specifichi se il risultato preserva dati e dipendenze.
- 2) Si consideri lo schema relazionale:

CorsiDiLaurea(CorsoLaurea: string, Facoltà: string, TipoLaurea: string)

Studenti(Matricola: integer, SNome: string, CorsoLaurea*: string)

Iscrizioni(Matricola*: integer, SiglaC*: string)

Corsi(SiglaC: string, OraRicevimento:time, DocenteId*: integer)

Docenti(DocenteId: integer, DNome: string, Dipartimento: string)

Si scrivano le seguenti interrogazioni in SQL

- Trovare, senza duplicati, il nome degli studenti che non frequentano corsi
 - Di ogni corso del docente di nome Tizio, trovare la sigla e il numero degli studenti che lo seguono
 - Trovare il nome degli studenti che seguono solo corsi del docente con DocenteId '800'
 - Trovare la Facoltà dei corsi di laurea frequentati da qualche studente che segue meno di 3 corsi
 - (Opz.) Trovare i nomi degli studenti che frequentano tutti i corsi tenuti dal docente di nome 'Tizio'
- 3) Si consideri la relazione Studenti(Matricola, Nome, AnnoNascita), ordinata sulla chiave primaria Matricola, e l'interrogazione

```
SELECT DISTINCT Matricola, avg(AnnoNascita)
```

```
FROM Studenti
```

```
WHERE AnnoNascita > 1974
```

```
GROUP BY Matricola
```

```
ORDER BY Matricola;
```

a) Disegnare l'albero di sintassi astratta di un'espressione algebrica ("albero logico") per l'interrogazione

b/c): Si dica se il piano d'accesso a destra produce il risultato cercato. Se non va bene, lo si modifichi in due modi: (b) aggiungendo prima solo le parti mancanti (operatori e parametri) e (c) semplificando poi il piano eliminando operatori inutili:

```
Sort({Matricola} )
|
Distinct
|
Sort({Matricola} )
|
Project({Matricola})
|
GroupBy({Matricola})
|
TableScan( Studenti )
```

4) (Opzionale) Si consideri il seguente schema, che rappresenta informazioni relative agli spettacoli programmati per una stagione in un insieme di teatri:

Spettacoli(Compagnia, Regista, Titolo, Data, Teatro)

Si esprimano i seguenti vincoli come dipendenze funzionali, se possibile:

- Se due spettacoli sono di compagnie diverse anche il regista è diverso
- Due spettacoli contemporanei sono recitati da compagnie diverse
- Non è possibile avere due spettacoli diversi nella stessa data e nello stesso teatro

Basi di Dati, 17/12/2003, secondo compito, compito A, soluzioni, Versione 1.2

1) Si consideri lo schema relazionale $R(A,B,C,D,E)$ con le seguenti DF:

$A \rightarrow BC, CB \rightarrow A, CD \rightarrow A, D \rightarrow B$

a) Si trovino le chiavi di R .

DE devono essere in tutte le chiavi. DEA e DEC sono le due chiavi.

b) Dire se lo schema è in 3FN o in FNBC.

Non è né in 3FN né in FNBC

c) Si applichi allo schema l'algoritmo di sintesi, si specifichi se il risultato preserva dati e dipendenze, e si dica se lo schema è in FNBC.

Porto le dipendenze in forma canonica. Non ci sono attributi estranei, perché: $C^+ = C, B^+ = B, D^+ = DB$.

Elimino le ridondanze. $A \rightarrow B: A^+_{\{A \rightarrow C, CB \rightarrow A, CD \rightarrow A, D \rightarrow B\}} = AC$

$A \rightarrow C: A^+_{\{A \rightarrow B, CB \rightarrow A, CD \rightarrow A, D \rightarrow B\}} = AB$

$CB \rightarrow A: CB^+_{\{A \rightarrow BC, CD \rightarrow A, D \rightarrow B\}} = CB$

$CD \rightarrow A: CD^+_{\{A \rightarrow BC, CB \rightarrow A, D \rightarrow B\}} = CDBA: CD \rightarrow A$ è ridondante

$D \rightarrow B: D^+_{\{A \rightarrow BC, CB \rightarrow A\}} = D$

1) $R1(ABC) \{A \rightarrow BC\}, R2(BCA) \{BC \rightarrow A\}, R3(DB) \{D \rightarrow B\}$

2) Elimino $R2$ (e copio la dipendenza $BC \rightarrow A$ su $R1$): $R1(ABC) \{A \rightarrow BC, BC \rightarrow A\}, R3(DB) \{D \rightarrow B\}$

3) Aggiungo una chiave DEA : $R1(ABC) \{A \rightarrow BC, BC \rightarrow A\}, R3(DB) \{D \rightarrow B\}, R4(DEA) \{\};$ dati e dipendenze sono preservati, ed il risultato è in FNBC.

d) (Opzionale) Si applichi allo schema l'algoritmo di analisi, e si specifichi se il risultato preserva dati e dipendenze.

$R(ABCDE)$; parto da $A \rightarrow BC$:

$R1(ABC) \{A \rightarrow BC, CB \rightarrow A\} + R2(ADE) \{\};$ la dipendenza $D \rightarrow B$ va perduta (la proiezione di F su ABC e su ADE non produce altre dipendenze). I dati sono preservati e il risultato è in FNBC

Oppure: $R(ABCDE)$; parto da $CB \rightarrow A$:

$R1(ABC) \{A \rightarrow BC, CB \rightarrow A\} + R2(BCDE) \{D \rightarrow B\}$

$R1(ABC) \{A \rightarrow BC, CB \rightarrow A\} + R3(DB) \{D \rightarrow B\} + R4(DCE)$; nessuna dipendenza va perduta. I dati sono preservati e il risultato è in FNBC

Altre soluzioni sono possibili.

2) Si consideri lo schema relazionale:

CorsiDiLaurea(CorsoLaurea: string, Facoltà: string, TipoLaurea: string)

Studenti(Matricola: integer, SNome: string, CorsoLaurea: string)*

Iscrizioni(Matricola: integer, SiglaC*: string)*

Corsi(SiglaC: string, OraRicevimento: time, DocenteId: integer)*

Docenti(DocenteId: integer, DNome: string, Dipartimento: string)

Si scrivano le seguenti interrogazioni in SQL

- a) *Trovare, senza duplicati, il DocenteId dei Corsi che non sono frequentati da nessuno studente*

```
SELECT DISTINCT C.DocenteId
FROM Corsi C
WHERE NOT EXISTS (SELECT *
                  FROM Iscrizioni I
                  WHERE C.SiglaC = I.SiglaC);
```

```
SELECT DISTINCT C.DocenteId
FROM Corsi C
WHERE C.SiglaC NOT IN (SELECT I.SiglaC
                      FROM Iscrizioni I);
```

- b) *Trovare la sigla dei corsi seguiti solo da studenti che appartengono al CorsoLaurea 'Informatica'*

(Si osservi che questa soluzione riporta anche i corsi che non sono seguiti da nessuno studente)

```
SELECT C.SiglaC
FROM Corsi C
WHERE NOT EXISTS (SELECT *
                 FROM Iscrizioni I, Studenti S
                 WHERE C.SiglaC = I.SiglaC AND I.Matricola = S.Matricola
                 AND S.CorsoLaurea <> 'Informatica');
```

- c) *Di ogni studente della Facoltà 'Scienze', trovare la matricola ed il numero di corsi che segue*

(Questa soluzione riporta solo gli studenti che seguono almeno un corso)

```
SELECT S.Matricola, count(*)
FROM CorsiDiLaurea C, Studenti S, Iscrizioni I
WHERE C.Facoltà = 'Scienze' AND C.CorsoLaurea = S.CorsoLaurea
AND S.Matricola = I.Matricola
GROUP BY S.Matricola;
```

- d) *Trovare nome e DocenteId dei docenti che insegnano qualche corso seguito da più di 5 studenti*

```
SELECT DISTINCT D.DocenteId, D.DNome
FROM Docenti D, Corsi C, Iscrizioni I
WHERE D.DocenteId = C.DocenteId AND C.SiglaC = I.SiglaC
GROUP BY C.SiglaC, D.DocenteId, D.DNome
HAVING Count(*) > 5
```

```

SELECT D.DocenteId, D.DNome
FROM Docenti D
WHERE EXISTS (SELECT C.SiglaC
              FROM Corsi C, Iscrizioni I
              WHERE D.DocenteId = C.DocenteId AND C.SiglaC = I.SiglaC
              GROUP BY C.SiglaC
              HAVING Count(*) > 5 )

```

```

SELECT DISTINCT D.DocenteId, D.DNome
FROM Docenti D, Corsi C
WHERE D.DocenteId = C.DocenteId
      AND EXISTS (SELECT I.SiglaC
                 FROM Iscrizioni I
                 WHERE C.SiglaC = I.SiglaC
                 GROUP BY I.SiglaC
                 HAVING Count(*) > 5 )

```

```

SELECT DISTINCT D.DocenteId, D.DNome
FROM Docenti D, Corsi C
WHERE D.DocenteId = C.DocenteId AND
5 < (SELECT count(*)
     FROM Iscrizioni I
     WHERE C.SiglaC = I.SiglaC );

```

e) *Trovare la sigla dei corsi che sono frequentati da tutti gli studenti della facoltà di 'Scienze'*

```

SELECT SiglaC
FROM Corsi C
WHERE NOT EXISTS (SELECT *
                 FROM Studenti S, CorsiDiLaurea Cdl
                 WHERE S.CorsoLaurea = Cdl.CorsoLaurea
                      AND Cdl.Facoltà = 'Scienze'
                      AND NOT EXISTS (SELECT *
                                     FROM Iscrizioni I
                                     WHERE I.SiglaC = C.SiglaC AND I.Matricola = S.Matricola));

```

```

SELECT C.SiglaC
FROM Corsi C
WHERE NOT EXISTS (SELECT *
                  FROM Studenti S, CorsiDiLaurea Cdl
                  WHERE S.CorsoLaurea = Cdl.CorsoLaurea
                        AND Cdl.Facoltà = 'Scienze'
                        AND S.Matricola NOT IN (SELECT I.Matricola
                                              FROM Iscrizioni I
                                              WHERE I.SiglaC = C.SiglaC));

```

```

SELECT C.SiglaC
FROM Corsi C
WHERE NOT EXISTS (SELECT Matricola
                  FROM Studenti S, CorsiDiLaurea Cdl
                  WHERE S.CorsoLaurea = Cdl.CorsoLaurea
                        AND Cdl.Facoltà = 'Scienze'
                  EXCEPT
                  SELECT Matricola
                  FROM Iscrizioni I
                  WHERE I.SiglaC = C.SiglaC);

```

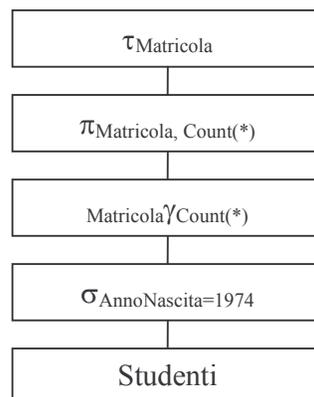
- 3) Si consideri la relazione Studenti(Matricola, Nome, AnnoNascita), ordinata sulla chiave primaria Matricola, e l'interrogazione

```

SELECT DISTINCT Matricola, count(*)
FROM Studenti
WHERE AnnoNascita = 1974
GROUP BY Matricola
ORDER BY Matricola;

```

Disegnare l'albero di sintassi astratta di un'espressione algebrica ("albero logico") per l'interrogazione



Si dica se il seguente piano d'accesso produce il risultato cercato. Se non va bene, lo si modifichi in due modi: (a) aggiungendo prima solo le parti mancanti (parametri e operatori) e (b) semplificando poi il piano eliminando operatori inutili:

```

Sort({Matricola} )
  |
Distinct
  |
Sort({Matricola} )
  |
Project({Matricola})
  |
GroupBy({Matricola})
  |
TableScan( Studenti )

```

Piano corretto

```

Sort({ Matricola } )
  |
Distinct
  |
Sort({ Matricola } )
  |
Project({ Matricola, count(*) })
  |
GroupBy({ Matricola}, {count(*) })
  |
Filter("AnnoNascita = 1974")
  |
TableScan( Studenti )

```

Piano semplificato togliendo operatori inutili (si tenga conto che la relazione è già ordinata su Matricola e che, essendo Matricola chiave, è inutile raggruppare per Matricola)

```

Project({ Matricola, 1})
  |
Filter("AnnoNascita = 1974")
  |
TableScan( Studenti )

```

4) (Opzionale) Si consideri il seguente schema, che rappresenta informazioni relative agli spettacoli programmati per una stagione in un insieme di teatri:

Spettacoli(Compagnia, Regista, Titolo, Data, Teatro)

Si esprimano i seguenti vincoli come dipendenze funzionali, se possibile:

- a) *Due spettacoli contemporanei hanno il regista diverso*
 Spettacolo \neq and Data $=$ \Rightarrow Regista \neq
 Regista $=$ and Data $=$ \Rightarrow Spettacolo $=$
 Regista, Data \rightarrow Compagnia, Titolo, Teatro
- b) *Non è possibile avere due spettacoli diversi nella stessa data e nello stesso teatro*
 Spettacolo \neq and Data $=$ and Teatro $=$ \Rightarrow False
 Data $=$ and Teatro $=$ \Rightarrow Spettacolo $=$
 Data, Teatro \rightarrow Compagnia, Regista, Titolo
- c) *Se due spettacoli hanno il regista diverso anche le compagnie sono diverse*
 Regista \neq \Rightarrow Compagnia \neq
 Compagnia $=$ \Rightarrow Regista $=$
 Compagnia \rightarrow Regista

Basi di Dati, 17/12/2003, secondo compito, compito B, soluzioni

1) Si consideri lo schema relazionale R(A,B,C,D,E) con le seguenti DF:

$C \rightarrow DE$, $ED \rightarrow C$, $EA \rightarrow C$, $A \rightarrow D$

a) *Si trovino le chiavi di R.*

AB devono essere in tutte le chiavi. ABC e ABE sono le due chiavi.

b) *Dire se lo schema è in 3FN o in FNBC.*

Non è né in 3FN né in FNBC

c) *Si applichi allo schema l'algoritmo di sintesi, si specifichi se il risultato preserva dati e dipendenze, e si dica se lo schema è in FNBC.*

Porto le dipendenze in forma canonica. Non ci sono attributi estranei, perché: $E^+ = E$, $D^+ = D$, $A^+ = AD$.

Elimino le ridondanze. $C \rightarrow D$: $C^+_{\{C \rightarrow E, ED \rightarrow C, EA \rightarrow C, A \rightarrow D\}} = CE$

$C \rightarrow E$: $C^+_{\{C \rightarrow D, ED \rightarrow C, EA \rightarrow C, A \rightarrow D\}} = CD$

$ED \rightarrow C$: $ED^+_{\{C \rightarrow DE, EA \rightarrow C, A \rightarrow D\}} = ED$

$EA \rightarrow C$: $CD^+_{\{C \rightarrow DE, ED \rightarrow C, A \rightarrow D\}} = EADC$: $EA \rightarrow C$ è ridondante

$A \rightarrow D$: $A^+_{\{C \rightarrow DE, ED \rightarrow C\}} = A$

1) R1(CDE) { $C \rightarrow DE$ }, R2{ $ED \rightarrow C$ }, R3(AD) { $A \rightarrow D$ }

2) Elimino R2 (e copio la dipendenza $DE \rightarrow C$ su R1): R1(CDE) { $C \rightarrow DE$, $ED \rightarrow C$ }, R3(AD) { $A \rightarrow D$ }

3) Aggiungo una chiave ABC: R1(CDE) { $C \rightarrow DE$, $ED \rightarrow C$ }, R3(AD) { $A \rightarrow D$ }, R4(ABC) {}; dati e dipendenze sono preservati, ed il risultato è in FNBC.

d) *(Opzionale) Si applichi allo schema l'algoritmo di analisi, e si specifichi se il risultato preserva dati e dipendenze.*

R(ABCDE)

$R(CDE) \{C \rightarrow DE, ED \rightarrow C\} + R2(ABC) \{\}$; la dipendenza $A \rightarrow D$ va perduta (la proiezione di F su DEC e su ABC non produce altre dipendenze). I dati sono preservati e il risultato è in FNBC

2) Si consideri lo schema relazionale:

CorsiDiLaurea(CorsoLaurea: integer, Facoltà: string, TipoLaurea: string)

Studenti(Matricola: integer, SNome: string, CorsoLaurea*: string)

Iscrizioni(Matricola*: integer, SiglaC*: string)

Corsi(SiglaC: string, OraRicevimento: time, DocenteId*: integer)

Docenti(DocenteId: integer, DNome: string, Dipartimento: string)

Si scrivano le seguenti interrogazioni in SQL

a) Trovare, senza duplicati, il nome degli studenti che non frequentano corsi

```
SELECT DISTINCT S.SNome
FROM Studenti S
WHERE NOT EXISTS (SELECT *
                  FROM Iscrizioni I
                  WHERE S.Matricola = I.Matricola);
```

```
SELECT DISTINCT S.SNome
FROM Studenti S
WHERE S.Matricola NOT IN (SELECT I.Matricola
                        FROM Iscrizioni I);
```

b) Di ogni corso del docente di nome Tizio, trovare la sigla e il numero degli studenti che lo seguono

(Si osservi che questa soluzione riporta solo i corsi seguiti da almeno uno studente)

```
SELECT C.SiglaC, count(*)
FROM Corsi C, Iscrizioni I, Docenti D
WHERE C.SiglaC = I.SiglaC
AND C.DocenteId = D.DocenteId and DNome = 'Tizio'
GROUP BY C.SiglaC;
```

c) Trovare il nome degli studenti che seguono solo corsi del docente con DocenteId '800'

(Si osservi che questa soluzione riporta anche gli studenti che non seguono alcun corso)

```
SELECT S.SNome
FROM Studenti S
WHERE NOT EXISTS (SELECT *
                  FROM Iscrizioni I, Corsi C
                  WHERE S.Matricola = I.Matricola AND I.SiglaC = C.SiglaC
                  AND C.DocenteId <> 800);
```

d) Trovare la Facoltà dei corsi di laurea frequentati da qualche studente che segue meno di 3 corsi

Queste soluzioni assumono che non esistano, o non interessino nella risposta, studenti non iscritti a nessun corso.

```
SELECT DISTINCT Cdl.Facoltà
FROM CorsiDiLaurea Cdl, Studenti S, Iscrizioni I
WHERE Cdl.CorsoLaurea = S.CorsoLaurea AND S.Matricola = I.Matricola
GROUP BY S.Matricola, Cdl.Facoltà
HAVING Count(*) < 3
```

```
SELECT DISTINCT Cdl.Facoltà
FROM CorsiDiLaurea Cdl
WHERE EXISTS (SELECT S.Matricola
              FROM Studenti S, Iscrizioni I
              WHERE Cdl.CorsoLaurea = S.CorsoLaurea
              AND S.Matricola = I.Matricola
              GROUP BY S.Matricola
              HAVING Count(*) < 3)
```

```
SELECT DISTINCT Cdl.Facoltà
FROM CorsiDiLaurea Cdl, Studenti S
WHERE Cdl.CorsoLaurea = S.CorsoLaurea
      AND EXISTS (SELECT I.Matricola
                  FROM Iscrizioni I
                  WHERE S.Matricola = I.Matricola
                  GROUP BY I.Matricola
                  HAVING Count(*) < 3)
```

```
SELECT DISTINCT Cdl.Facoltà
FROM CorsiDiLaurea Cdl, Studenti S
WHERE Cdl.CorsoLaurea = S.CorsoLaurea AND
3 > (SELECT count(*)
     FROM Iscrizioni I
     WHERE S.Matricola = I.Matricola);
```

Volendo trovare anche le facoltà frequentate da studenti che non sono iscritti ad alcun corso, occorre aggiungere una sottoselect, come mostrato sotto:

```

SELECT ...
UNION
SELECT DISTINCT Cdl.Facoltà
FROM CorsiDiLaurea Cdl, Studenti S
WHERE Cdl.CorsoLaurea = S.CorsoLaurea
AND NOT EXISTS (SELECT *
                 FROM Iscrizioni I
                 WHERE S.Matricola = I.Matricola)

```

e) *Trovare i nomi degli studenti che frequentano tutti i corsi tenuti dal docente di nome 'Tizio'*

Ammettendo che il nome del docente non sia chiave, l'interrogazione ammetteva due diverse interpretazioni. (Assumiamo comunque che esista un docente di nome 'Tizio' e che tenga almeno un corso.) Le due soluzioni sono equivalenti se il docente di nome 'Tizio' è unico.

a) *Trovare i nomi degli studenti che frequentano tutti i corsi il cui docente si chiama 'Tizio'*

```

SELECT SNome
FROM Studenti S
WHERE NOT EXISTS (SELECT *
                 FROM Corsi C, Docenti D
                 WHERE C.DocenteId = D.DocenteId
                    AND D.DNome = 'Tizio'
                    AND NOT EXISTS (SELECT *
                                    FROM Iscrizioni I
                                    WHERE I.SiglaC = C.SiglaC AND I.Matricola = S.Matricola));

```

sono possibili anche le due varianti presentate nella correzione del compito A.

b) *Trovare i nomi degli studenti che frequentano tutti i corsi di almeno un docente che si chiama 'Tizio'*

```

SELECT DISTINCT SNome
FROM Studenti S, Docenti D
WHERE D.DNome = 'Tizio'
AND NOT EXISTS (SELECT *
                FROM Corsi C
                WHERE C.DocenteId = D.DocenteId
                   AND NOT EXISTS (SELECT *
                                    FROM Iscrizioni I
                                    WHERE I.SiglaC = C.SiglaC AND I.Matricola = S.Matricola));

```

3) Si consideri la relazione Studenti(Matricola, Nome, AnnoNascita), ordinata sulla chiave primaria Matricola, e l'interrogazione

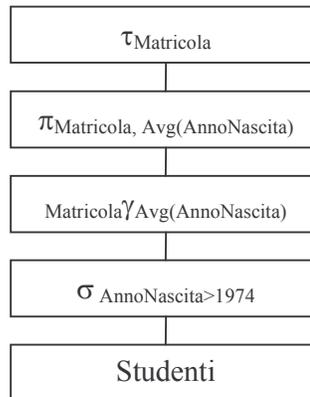
```

SELECT DISTINCT Matricola, avg(AnnoNascita)

```

FROM Studenti
 WHERE AnnoNascita > 1974
 GROUP BY Matricola
 ORDER BY Matricola;

Disegnare l'albero di sintassi astratta di un'espressione algebrica ("albero logico") per l'interrogazione



Si dica se il seguente piano d'accesso produce il risultato cercato. Se non va bene, lo si modifichi in due modi: (a) aggiungendo prima solo le parti mancanti (parametri e operatori) e (b) semplificando poi il piano eliminando operatori inutili:

```

    Sort({ Matricola } )
    |
    Distinct
    |
    Sort({ Matricola } )
    |
    Project({ Matricola })
    |
    GroupBy({ Matricola })
    |
    TableScan( Studenti )
  
```

Piano corretto

```

Sort({ Matricola } )
|
Distinct
|
Sort({ Matricola } )
|
Project({Matricola, avg(AnnoNascita)})
|
GroupBy({Matricola}, {avg(AnnoNascita)})
|
Filter("AnnoNascita > 1974")
|
TableScan( Studenti )

```

Piano semplificato togliendo operatori inutili (si tenga conto che la relazione è già ordinata su Matricola e che, essendo Matricola chiave, è inutile raggruppare per Matricola)

```

Project({Matricola, AnnoNascita})
|
Filter("AnnoNascita > 1974")
|
TableScan( Studenti )

```

- 4) (Opzionale) Si consideri il seguente schema, che rappresenta informazioni relative agli spettacoli programmati per una stagione in un insieme di teatri:

Spettacoli(Compagnia, Regista, Titolo, Data, Teatro)

Si esprimano i seguenti vincoli come dipendenze funzionali, se possibile:

- a) Se due spettacoli sono di compagnie diverse anche il regista è diverso

$Compagnia \neq \Rightarrow Regista \neq$

$Regista = \Rightarrow Compagnia =$

$Regista \rightarrow Compagnia$

- b) Due spettacoli contemporanei sono recitati da compagnie diverse

$Spettacolo \neq \text{ and } Data = \Rightarrow Compagnia$

$Compagnia = \text{ and } Data = \Rightarrow Spettacolo =$

$Compagnia, Data \rightarrow Regista, Titolo, Teatro$

- c) Non è possibile avere due spettacoli diversi nella stessa data e nello stesso teatro

$Spettacolo \neq \text{ and } Data = \text{ and } Teatro = \Rightarrow \text{False}$

$Data = \text{ and } Teatro = \Rightarrow Spettacolo =$

$Data, Teatro \rightarrow Compagnia, Regista, Titolo$