

## Compitino di basi di dati, 20/12/2001, compito A

1) Si consideri il seguente schema:

Attori(CodiceAtt, Nome, AnnoNascita);  
AttoriFilm(CodiceAtt\*, CodiceFilm\*)  
Film(CodiceFilm, Titolo, AnnoProduzione, Regista)  
Proiezioni(CodiceFilm\*, CodiceSala\*, Incasso, DataProiezione)  
Sale(CodiceSala, Posti, Nome, Città)



Si scrivano in SQL le seguenti interrogazioni.

- 1.1) Per ogni film in cui appaiono solo attori nati prima del 1970 restituire il titolo del film
  - 1.2) Per ogni film in cui appaiono solo attori nati prima del 1970 restituire il regista del film ed il numero di attori
  - 1.3) Scrivere le due interrogazioni seguenti:  
Restituire il nome ed il codice di tutte le sale di Pisa in cui ogni proiezione del film con codice 100 **che è avvenuta** il 10/12/2001 ha incassato almeno 300.000 Lire  
Restituire il nome ed il codice di tutte le sale di Pisa in cui ogni proiezione del film con codice 100 **è avvenuta** il 10/12/2001 **ed** ha incassato almeno 300.000 Lire
- 2) Per tenere traccia delle allocazioni delle aule di un polo didattico durante una certa settimana è necessario trattare i seguenti attributi: OraInizioLezione, OraFineLezione, GiornoSettimana, NomeCorso (es: Analisi), LetteraCorso (es: A, B, C, Unico), TitolareCorso, Aula, PianoDellAula (es.: Primo, Secondo), CorsoDiLaureaDelCorso. Lo stesso nome di corso può essere usato, con la stessa lettera, in diversi corsi di laurea. Specificate, per ciascuna delle seguenti affermazioni, la dipendenza funzionale che ne deriva:  
(2.1) Tutte le lezioni durano lo stesso tempo (2.2) Ogni corso ha un solo titolare (2.3) Non si possono tenere due corsi contemporaneamente nella stessa aula (2.4) Uno stesso docente non può essere in due aule contemporaneamente.
- 3) Si consideri il seguente insieme di dipendenze funzionali in forma canonica:  
 $R(ABCDE), AD \rightarrow B, CB \rightarrow A, DE \rightarrow A, A \rightarrow E$
- 3.1) Trovate tutte le chiavi
  - 3.2) Dite se lo schema è in 3FN o in BCNF
  - 3.3) Applicate l'algoritmo di sintesi
- 4.1) Si consideri lo schema dell'esercizio (1) e, per l'interrogazione seguente, si fornisca un piano di accesso efficiente, supponendo che sia presente un indice sugli attributi Film(CodiceFilm), Proiezioni(CodiceFilm\*)
- ```
select f.Titolo, count(*), sum(p.Incasso)
from Film f, Proiezioni p
where f.CodiceFilm = p.CodiceFilm
and f.Regista = 'John Waters'
group by f.CodiceFilm, f.Titolo
having sum(p.Incasso) > 100
```
- 4.2) Per l'interrogazione seguente, si forniscano due piani di accesso, uno che a vostro parere sarebbe particolarmente efficiente, ed uno che vi pare molto inefficiente, supponendo che sia presente un indice sugli attributi Attori(CodiceAtt), AttoriFilm(CodiceAtt), AttoriFilm(CodiceFilm\*), Film(CodiceFilm)
- ```
select f.Titolo, a.Nome
from Attori a, AttoriFilm af, Film f
where a.CodiceAtt=af.CodiceAtt and af.CodiceFilm = f.CodiceFilm
and a.AnnoNascita = 1970
```

## Compitino di basi di dati, 20/12/2001, compito B

1) Si consideri il seguente schema:

Attori(CodiceAtt, Nome, AnnoNascita);  
AttoriFilm(CodiceAtt\*, CodiceFilm\*)  
Film(CodiceFilm, Titolo, AnnoProduzione, Regista)  
Proiezioni(CodiceFilm\*, CodiceSala\*, Incasso, DataProiezione)  
Sale(CodiceSala, Posti, Nome, Città)



Si scrivano in SQL le seguenti interrogazioni.

- 1.1) Per ogni film proiettato solo in sale con più di 100 posti restituire il regista del film
- 1.2) Per ogni film proiettato solo in sale con più di 100 posti restituire il titolo ed il numero di attori
- 1.3) Scrivere le due interrogazioni seguenti:  
Restituire il codice ed il titolo di tutti i film di Fellini per i quali ogni proiezione nella sala con codice 100 **che è avvenuta** il 10/12/2001 ha incassato più di 800.000 Lire  
Restituire il codice ed il titolo di tutti i film di Fellini per i quali ogni proiezione nella sala con codice 100 **è avvenuta** il 10/12/2001 **ed** ha incassato più di 800.000 Lire
- 2) Per tenere traccia delle allocazioni delle aule di un polo didattico durante una certa settimana è necessario trattare i seguenti attributi: OraInizioLezione, OraFineLezione, GiornoSettimana, NomeCorso (es: Analisi), LetteraCorso (es: A, B, C, Unico), TitolareCorso, Aula, PianoDellAula (es.: Primo, Secondo), CorsoDiLaureaDelCorso. Lo stesso nome di corso può essere usato, con la stessa lettera, in diversi corsi di laurea. Specificate, per ciascuna delle seguenti affermazioni, la dipendenza funzionale che ne deriva:  
(2.1) Non si possono tenere due corsi contemporaneamente nella stessa aula (2.2) Uno stesso docente non può essere in due aule contemporaneamente (2.3) Ogni corso ha un solo titolare (2.4) Tutte le lezioni durano lo stesso tempo.
- 3) Si consideri il seguente insieme di dipendenze funzionali in forma canonica:

$R(ABCDE), EC \rightarrow B, AB \rightarrow E, CD \rightarrow E, E \rightarrow D$

- 3.1) Trovate tutte le chiavi
- 3.2) Dite se lo schema è in 3FN o in BCNF
- 3.3) Applicate l'algoritmo di sintesi
- 4.1) Si consideri lo schema dell'esercizio (1) e, per l'interrogazione seguente, si fornisca un piano di accesso efficiente, supponendo che sia presente un indice sugli attributi Sale(CodiceSala), Proiezioni(CodiceSala\*)

```
select s.Nome, count(*), sum(p.Incasso)
from Sale s, Proiezioni p
where s.CodiceSala = p.CodiceSala
and s.Città = 'Pisa'
group by s.CodiceSala, s.Nome
having sum(p.Incasso) > 100
```

- 4.2) Per l'interrogazione seguente, si forniscano due piani di accesso, uno che a vostro parere sarebbe particolarmente efficiente, ed uno che vi pare molto inefficiente, supponendo che sia presente un indice su Film(CodiceFilm), Proiezioni(CodiceFilm\*), Proiezioni(CodiceSala\*), Sale(CodiceSala)

```
select f.Titolo, s.Nome
from Film f, Proiezioni p, Sale s
where s.CodiceSala = p.CodiceSala and p.CodiceFilm = f.CodiceFilm
and s.Città = 'Pisa'
```

## Compitino di basi di dati, 20/12/2001, compito A, soluzione, V1.1

1) Si consideri il seguente schema:

Attori(CodiceAtt, Nome, AnnoNascita);  
AttoriFilm(CodiceAtt\*, CodiceFilm\*)  
Film(CodiceFilm, Titolo, AnnoProduzione, Regista)  
Proiezioni(CodiceFilm\*, CodiceSala\*, Incasso, DataProiezione)  
Sale(CodiceSala, Posti, Nome, Città)



Si scrivano in SQL le seguenti interrogazioni.

1.1) Per ogni film in cui appaiono solo attori nati prima del 1970 restituire il titolo del film

```
select f.Titolo
from Film f
where not exists ( select *
                  from AttoriFilm af, Attori a
                  where a.CodiceAtt=af.CodiceAtt and af.CodiceFilm = f.CodiceFilm
                  and a.AnnoNascita >= 1970)
```

oppure:

```
select f.Titolo
from Film f
where 1970 >all ( select a.AnnoNascita
                 from AttoriFilm af, Attori a
                 where a.CodiceAtt=af.CodiceAtt and af.CodiceFilm = f.CodiceFilm)
```

1.2) Per ogni film in cui appaiono solo attori nati prima del 1970 restituire il regista del film ed il numero di attori

```
select f.Regista, count(*) as NumeroAttori
from Attori a, AttoriFilm af, Film f
where a.CodiceAtt=af.CodiceAtt and af.CodiceFilm = f.CodiceFilm
group by f.CodiceFilm, f.Regista
having max(a.AnnoNascita) < 1970
```

1.3) Scrivere le due interrogazioni seguenti:

Restituire il nome ed il codice di tutte le sale di Pisa in cui ogni proiezione del film con codice 100 **che è avvenuta** il 10/12/2001 ha incassato almeno 300.000 Lire

Restituire il nome ed il codice di tutte le sale di Pisa in cui ogni proiezione del film con codice 100 **è avvenuta** il 10/12/2001 **ed** ha incassato almeno 300.000 Lire

```
select s.Nome, s.CodiceSala
from Sale s
where s.Citta = 'Pisa' and
      not exists ( select *
                  from Proiezioni p
                  where p.CodiceSala=s.CodiceSala
                  and p.CodiceFilm = 100
                  and p.DataProiezione = '10/12/2001' and not(p.Incasso >= 300000))
```

```
select s.Nome, s.CodiceSala
from Sale s
where s.Citta = 'Pisa' and
```

```

not exists ( select *
              from Proiezioni p, Film f
              where p.CodiceSala=s.CodiceSala
                 and p.CodiceFilm = 100
                 and Not(p.DataProiezione = '10/12/2001' and p.Incasso >= 300000))

```

- 2) Per tenere traccia delle allocazioni delle aule di un polo didattico durante una certa settimana è necessario trattare i seguenti attributi: OraInizioLezione, OraFineLezione, GiornoSettimana, NomeCorso (es: Analisi), LetteraCorso (es: A, B, C, Unico), TitolareCorso, Aula, PianoDellAula (es.: Primo, Secondo), CorsoDiLaureaDelCorso. Lo stesso nome di corso può essere usato, con la stessa lettera, in diversi corsi di laurea. Specificate, per ciascuna delle seguenti affermazioni, la dipendenza funzionale che ne deriva:  
 (2.1) Tutte le lezioni durano lo stesso tempo (2.2) Ogni corso ha un solo titolare (2.3) Non si possono tenere due corsi contemporaneamente nella stesa aula (2.4) Uno stesso docente non può essere in due aule contemporaneamente.

2.1 OraInizioLezione → OraFineLezione, OraFineLezione → OraInizioLezione

2.2 CorsoDiLaureaDelCorso, NomeCorso, LetteraCorso → TitolareCorso

2.3 GiornoSettimana, OraInizioLezione, Aula  
 → CorsoDiLaureaDelCorso, NomeCorso, LetteraCorso

2.4 GiornoSettimana, OraInizioLezione, TitolareCorso → Aula

- 3) Si consideri il seguente insieme di dipendenze funzionali in forma canonica:

R(ABCDE), AD → B, CB → A, DE → A, A → E

- 3.1) Trovate tutte le chiavi

CDA, CDB, CDE

- 3.2) Dite se lo schema è in 3FN o in BCNF

Lo schema è solo in 3FN

- 3.3) Applicate l'algoritmo di sintesi

Raggruppo gli attributi: R1(ADB), R2(CBA), R3(DEA), R4(AE)

Elimino la relazione in più: R1(ADB), R2(CBA), R3(DEA)

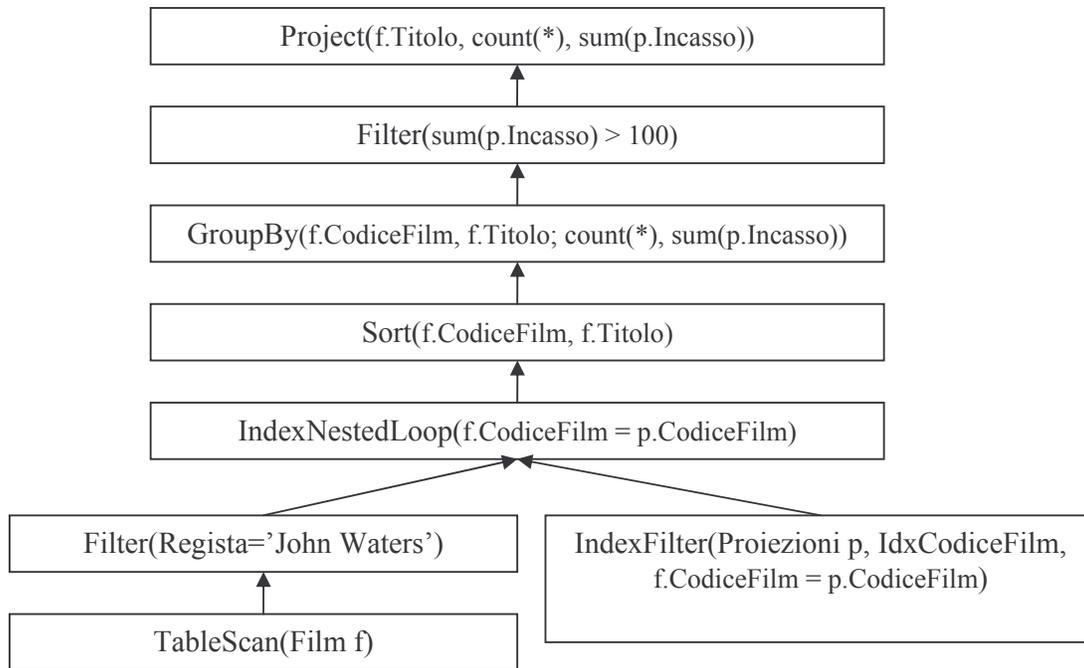
Aggiungo la chiave: R1(ADB), R2(CBA), R3(DEA), R4(CDA)

- 4.1) Si consideri lo schema dell'esercizio (1) e, per l'interrogazione seguente, si fornisca un piano di accesso efficiente, supponendo che sia presente un indice sugli attributi Film(CodiceFilm), Proiezioni(CodiceFilm\*)

```

select  f.Titolo, count(*), sum(p.Incasso)
from    Film f, Proiezioni p
where   f.CodiceFilm = p.CodiceFilm
        and f.Regista = 'John Waters'
group  by f.CodiceFilm, f.Titolo
having  sum(p.Incasso) > 100

```

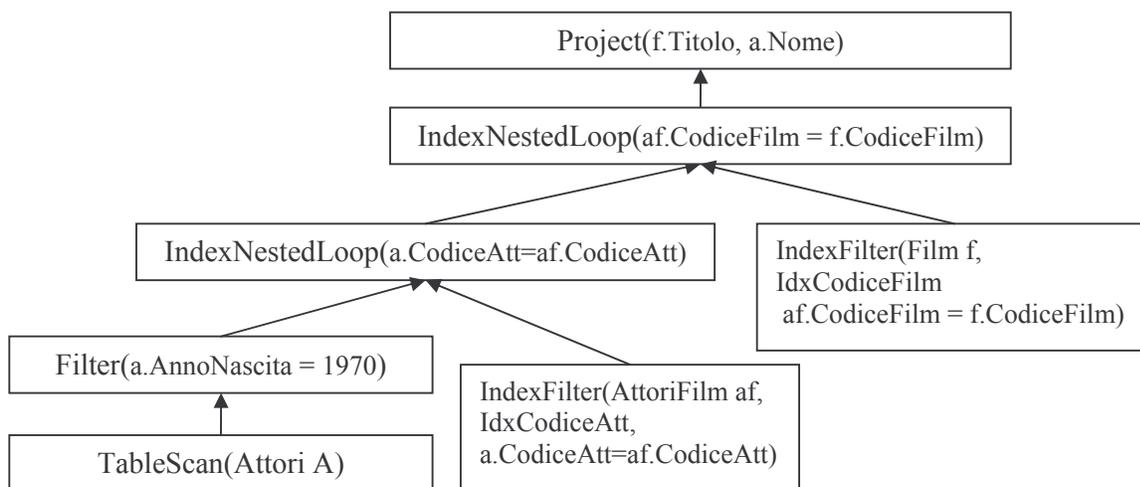


Per l'interrogazione seguente, si forniscano due piani di accesso, uno che a vostro parere sarebbe particolarmente efficiente, ed uno che vi pare molto inefficiente, supponendo che sia presente un indice sugli attributi Attori(CodiceAtt), AttoriFilm(CodiceAtt\*), AttoriFilm(CodiceFilm\*), Film(CodiceFilm)

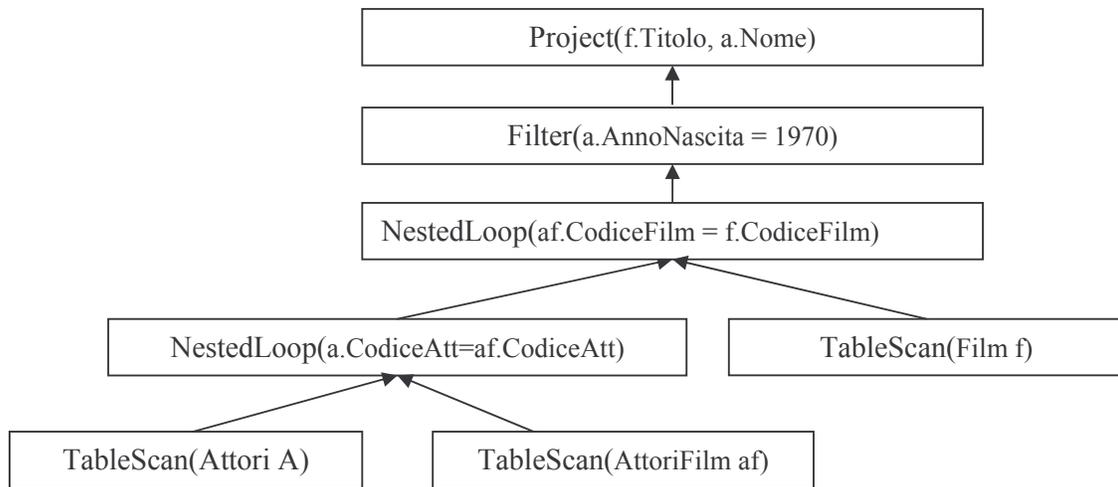
```

select  f.Titolo, a.Nome
from    Attori a, AttoriFilm af, Film f
where   a.CodiceAtt=af.CodiceAtt and af.CodiceFilm = f.CodiceFilm
        and a.AnnoNascita = 1970
  
```

*Piano efficiente:*



*Piano inefficiente*



## Compitino di basi di dati, 20/12/2001, compito B, soluzione, V1.1

1) Si consideri il seguente schema:

Attori(CodiceAtt, Nome, AnnoNascita);  
AttoriFilm(CodiceAtt\*, CodiceFilm\*)  
Film(CodiceFilm, Titolo, AnnoProduzione, Regista)  
Proiezioni(CodiceFilm\*, CodiceSala\*, Incasso, DataProiezione)  
Sale(CodiceSala, Posti, Nome, Città)



Si scrivano in SQL le seguenti interrogazioni.

1.1) Per ogni film proiettato solo in sale con più di 100 posti restituire il regista del film

```
select f.Regista
from Film f
where not exists ( select *
                  from Proiezioni p, Sale s
                  where f.CodiceFilm = p.CodiceFilm and p.CodiceSala=s.CodiceSala
                  and s.Posti <= 100)
```

oppure:

```
select f.Regista
from Film f
where 100 <all (select s.Posti
               from Proiezioni p, Sale s
               where f.CodiceFilm = p.CodiceFilm and p.CodiceSala=s.CodiceSala)
```

1.2) Per ogni film proiettato solo in sale con più di 100 posti restituire il titolo ed il numero di attori

```
select f.Titolo, count(*) as NumeroAttori
from Film f, AttoriFilm af
where f.CodiceFilm = af.CodiceFilm
and 100 <all ( select s.Posti
              from Proiezioni p, Sale s
              where f.CodiceFilm = p.CodiceFilm and p.CodiceSala=s.CodiceSala)
group by f.CodiceFilm, f.Titolo
```

1.3) Scrivere le due interrogazioni seguenti:

Restituire il codice ed il titolo di tutti i film di Fellini per i quali ogni proiezione nella sala con codice 100 **che è avvenuta** il 10/12/2001 ha incassato più di 800.000 Lire

Restituire il codice ed il titolo di tutti i film di Fellini per i quali ogni proiezione nella sala con codice 100 **è avvenuta** il 10/12/2001 **ed** ha incassato più di 800.000 Lire

```
select f.Titolo, f.CodiceFilm
from Film f
where f.Regista = 'Fellini' and
not exists ( select *
            from Proiezioni p
            where f.CodiceFilm = p.CodiceFilm
            and p.CodiceSala = 100
            and p.DataProiezione = '10/12/2001' and not(p.Incasso > 800000))
```

```

select f.Titolo, f.CodiceFilm
from Film f
where f.Regista = 'Fellini' and
not exists ( select *
from Proiezioni p
where f.CodiceFilm = p.CodiceFilm
and p.CodiceSala = 100
and not(p.DataProiezione = '10/12/2001' and p.Incasso > 800000))

```

- 2) Per tenere traccia delle allocazioni delle aule di un polo didattico durante una certa settimana è necessario trattare i seguenti attributi: OraInizioLezione, OraFineLezione, GiornoSettimana, NomeCorso (es: Analisi), LetteraCorso (es: A, B, C, Unico), TitolareCorso, Aula, PianoDellAula (es.: Primo, Secondo), CorsoDiLaureaDelCorso. Lo stesso nome di corso può essere usato, con la stessa lettera, in diversi corsi di laurea. Specificate, per ciascuna delle seguenti affermazioni, la dipendenza funzionale che ne deriva:  
(2.1) Non si possono tenere due corsi contemporaneamente nella stessa aula (2.2) Uno stesso docente non può essere in due aule contemporaneamente (2.3) Ogni corso ha un solo titolare (2.4) Tutte le lezioni durano lo stesso tempo.

- 2.1 GiornoSettimana, OraInizioLezione, Aula  
→ CorsoDiLaureaDelCorso, NomeCorso, LetteraCorso
- 2.2 GiornoSettimana, OraInizioLezione, TitolareCorso → Aula
- 2.3 CorsoDiLaureaDelCorso, NomeCorso, LetteraCorso → TitolareCorso
- 2.4 OraInizioLezione → OraFineLezione, OraFineLezione → OraInizioLezione

- 3) Si consideri il seguente insieme di dipendenze funzionali in forma canonica:

R(ABCDE), EC → B, AB → E, CD → E, E → D

- 3.1) Trovate tutte le chiavi

ACE, ACB, ACD

- 3.2) Dite se lo schema è in 3FN o in BCNF

Lo schema è solo in 3FN

- 3.3) Applicate l'algoritmo di sintesi

Raggruppo gli attributi: R1(ECB), R2(ABE), R3(CDE), R4(ED)

Elimino la relazione in più: R1(ECB), R2(ABE), R3(CDE)

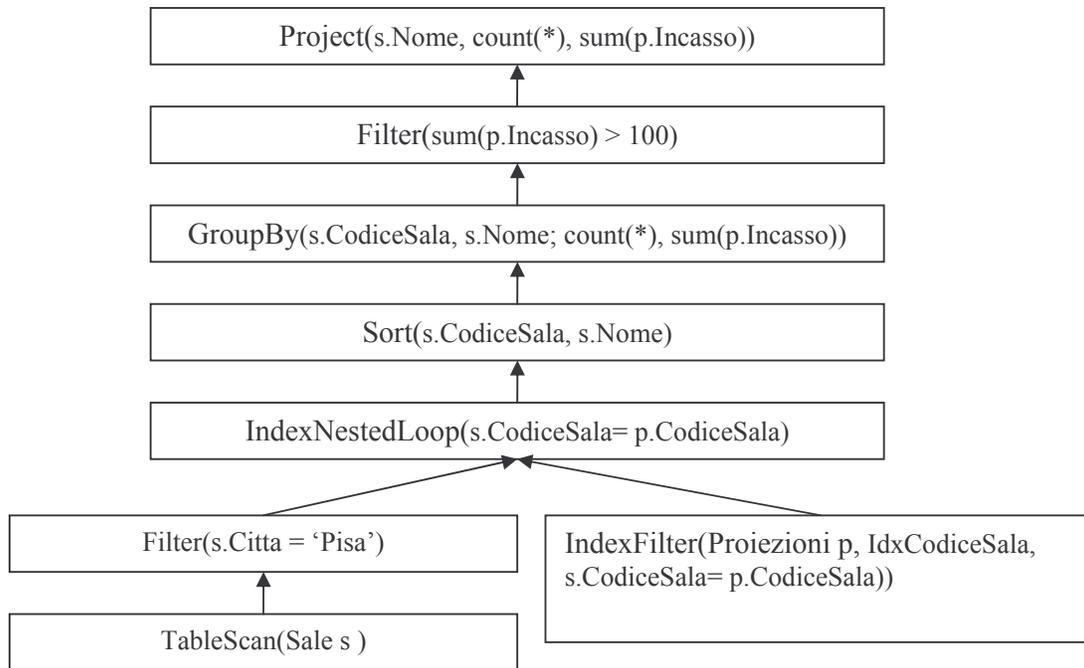
Aggiungo la chiave: R1(ECB), R2(ABE), R3(CDE), R4(ACE)

- 4.1) Si consideri lo schema dell'esercizio (1) e, per l'interrogazione seguente, si fornisca un piano di accesso efficiente, supponendo che sia presente un indice sugli attributi Sale(CodiceSala), Proiezioni(CodiceSala\*)

```

select s.Nome, count(*), sum(p.Incasso)
from Sale s, Proiezioni p
where s.CodiceSala = p.CodiceSala
and s.Città = 'Pisa'
group by s.CodiceSala, s.Nome
having sum(p.Incasso) > 100

```

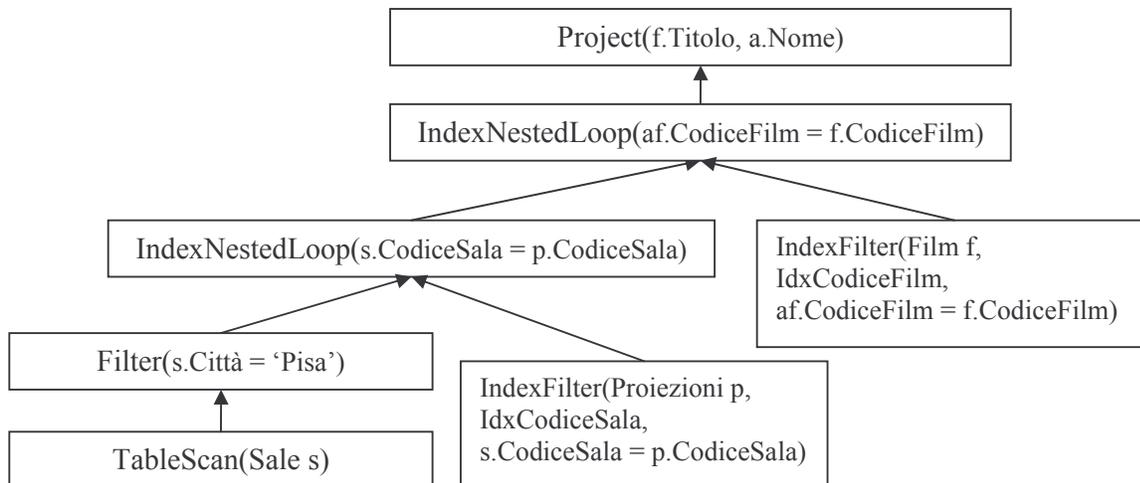


Per l'interrogazione seguente, si forniscano due piani di accesso, uno che a vostro parere sarebbe particolarmente efficiente, ed uno che vi pare molto inefficiente, supponendo che sia presente un indice su Film(CodiceFilm), Proiezioni(CodiceFilm\*), Proiezioni(CodiceSala\*), Sale(CodiceSala)

```

select f.Titolo, s.Nome
from Film f, Proiezioni p, Sale s
where s.CodiceSala = p.CodiceSala and p.CodiceFilm = f.CodiceFilm
and s.Città = 'Pisa'
  
```

*Piano efficiente:*



*Piano inefficiente*

