

ATTORE (CodAttore, Nome, AnnoNascita, Nazionalità)

RECITA (CodAttore, CodFilm)

FILM (CodFilm, Titolo, AnnoProduzione, Nazionalità, Regista, Genere, Durata)

PROIEZIONE (CodProiezione, CodFilm, CodSala, Incasso, DataProiezione)

SALA (CodSala, Posti, Nome, Città)

1. Il nome di tutte le sale di Pisa

ALGEBRA

```
SELECT s.NOME
FROM SALA AS s
WHERE s.CITTÀ = "PISA"
```

$$R_1 = \sigma_{CITTÀ = "PISA"}(SALA)$$

$$R_2 = \Pi_{NOME}(R_1)$$

2. Il titolo dei film di F. Fellini prodotti dopo il 1960.

```
SELECT f.TITOLO
FROM FILM AS f
WHERE f.REGISTA = "F.FELLINI" AND f.ANNOPRODUZIONE > 1960
```

3. Il titolo e la durata dei film di fantascienza giapponesi o francesi prodotti dopo il 1990

```
SELECT f.TITOLO, f.DURATA
FROM FILM AS f
WHERE f.GENERE = FANTASCIENZA AND ((f.NAZIONALITÀ = GIAPP) OR (f.NAZIONALITÀ = FRANCESE))
AND f.ANNOPRODUZIONE > 1990
```

4. Il titolo dei film di fantascienza giapponesi prodotti dopo il 1990 oppure francesi

```
SELECT f.TITOLO
FROM FILM AS f
WHERE f.GENERE = FANTASCIENZA AND ((f.NAZIONALITÀ = GIAPP) OR (f.NAZIONALITÀ = FRANCESE))
AND f.ANNOPRODUZIONE > 1990
```

5. Il titolo dei film dello stesso regista di "Casablanca"

```
SELECT f.TITOLO
FROM FILM AS f
WHERE f.REGISTA = (SELECT f2.REGISTA
                   FROM FILM AS f2
                   WHERE f2.NOME = "CASABLANCA")
```

6. Il titolo ed il genere dei film proiettati il giorno di Natale 2004

```
SELECT f.TITOLO, f.GENERE  
FROM FILM AS f JOIN PROIEZIONE AS p ON f.CODFILM = p.CODFILM  
WHERE p.DATAPROIEZIONE = "25/12/2004"
```

7. Il titolo ed il genere dei film proiettati a Napoli il giorno di Natale 2004

```
SELECT f.TITOLO, f.GENERE  
FROM (FILM AS f JOIN PROIEZIONE AS p ON f.CODFILM = p.CODFILM) JOIN  
SALA AS s ON p.CODSALA = s.CODSALA  
WHERE p.DATAPROIEZIONE = "25/12/2004" AND s.CITTÀ = "NAPOLI"
```

8. I nomi delle sale di Napoli in cui il giorno di Natale 2004 è stato proiettato un film con R.Williams

```
SELECT s.NOME  
FROM SALA AS s  
WHERE s.CITTÀ = 'NAPOLI' AND s.CODSALA IN  
(SELECT ss.CODSALA  
FROM (SALA AS ss JOIN PROIEZIONE AS p ON ss.CODSALA = p.CODSALA) JOIN RECITA AS r  
ON r.CODFILM = p.CODFILM) JOIN ATTORE AS a ON a.CODATTORE = r.CODATTORE  
WHERE a.NOME = 'R.WILLIAMS' AND p.DATAPROIEZIONE = '25-12-2004'
```

9. Il titolo dei film in cui recita M. Mastroianni oppure S.Loren

```
SELECT f.TITOLO  
FROM ((FILM AS f JOIN RECITA AS r ON f.CODFILM = r.CODFILM) JOIN ATTORE  
AS a ON r.CODATTORE = a.CODATTORE)  
WHERE a.NOME = "M. MASTROIANNI" OR a.NOME = "S. LOREN"
```

10. Il titolo dei film in cui recitano M. Mastroianni e S.Loren

```
SELECT f.TITOLO  
FROM ((FILM AS f JOIN RECITA AS r ON f.CODFILM = r.CODFILM) JOIN ATTORE  
AS a ON r.CODATTORE = a.CODATTORE)  
WHERE a.NOME = "M. MASTROIANNI" AND a.NOME = "S. LOREN"
```

11. Per ogni film in cui recita un attore francese, il titolo del film e il nome dell'attore

```
SELECT f.NOME, a.NOME  
FROM (FILM AS f JOIN RECITA AS r ON f.CODFILM = r.CODFILM) JOIN ATTORE AS  
a ON r.CODATTORE = a.CODATTORE  
WHERE a.NAZIONALITA = "FRANCESE"
```

12. Per ogni film che è stato proiettato a Pisa nel gennaio 2005, il titolo del film e il nome della sala.

```
SELECT f.NOME, s.NOME  
FROM (FILM AS f JOIN PROIEZIONE AS p ON f.CODFILM = p.CODFILM) JOIN SALA AS s  
ON p.CODSALA = s.CODSALA  
WHERE s.CITTÀ = "PISA" AND p.DATAPROIEZIONE = "*/01/2005"
```

13. Il numero di sale di Pisa con più di 60 posti

```
SELECT COUNT(*)  
FROM SALA AS s  
WHERE s.CITTÀ = "PISA" AND s.POSTI > 60
```

14. Il numero totale di posti nelle sale di Pisa

```
SELECT SUM(s.POSTI)  
FROM SALA AS s  
WHERE s.CITTÀ = "PISA"
```

15. Per ogni città, il numero di sale

```
SELECT s.CITTÀ, COUNT(*)  
FROM SALA AS s  
GROUP BY s.CITTÀ
```

16. Per ogni città, il numero di sale con più di 60 posti

```
SELECT s.CITTÀ, COUNT(*)  
FROM SALA AS s  
WHERE s.POSTI > 60  
GROUP BY s.CITTÀ
```

17. Per ogni regista, il numero di film diretti dopo il 1990

```
SELECT COUNT(*)
FROM FILM AS f
WHERE f.ANNOPRODUZIONE > 1990
GROUP BY f.REGISTA
```

18. Per ogni regista, l'incasso totale di tutte le proiezioni dei suoi film

```
SELECT SUM(p.INCASSO)
FROM (FILM AS f JOIN PROIEZIONE AS p ON f.CODFILM = p.CODFILM)
GROUP BY f.REGISTA
```

19. Per ogni film di S.Spielberg, il titolo del film, il numero totale di proiezioni a Pisa e l'incasso totale

```
SELECT f.NOME, COUNT(*), SUM(p.INCASSO)
FROM (FILM AS f JOIN PROIEZIONE AS p ON f.CODFILM = p.CODFILM) JOIN SALA AS s
ON s.CITTÀ = "PISA"
GROUP BY f.REGISTA = "S.SPIELBERG"
```

20. Per ogni regista e per ogni attore, il numero di film del regista con l'attore

```
SELECT COUNT(*)
FROM (FILM AS f JOIN RECITA AS r ON f.CODFILM = r.CODFILM) JOIN ATTORE AS a
ON r.CODATTORE = a.CODATTORE
GROUP BY f.REGISTA, a.NOME
```

21. Il regista ed il titolo dei film in cui recitano meno di 6 attori

```
SELECT f.REGISTA, f.TITOLO
FROM FILM AS f
WHERE (SELECT COUNT(*)
      FROM RECITA AS r JOIN ATTORE AS a ON a.CODATTORE = r.CODATTORE
      WHERE r.CODFILM = f.CODFILM) < 6;
```

$A_1 \leftarrow \rho_{\langle CODATTORE, NOME_ATTORE, ANNODIRETTA_ATTORE, NAZIONALITA_ATTORE \rangle} (ATTORE)$

$R_1 \leftarrow \text{FILM} * \text{RECITA} * A_1$

$R_2 \leftarrow \text{TITOLO}, \text{REGISTA} \nabla \text{COUNT(*)} (R_1)$

$R_3 \leftarrow \sigma_{\text{COUNT} < 6} (R_2)$

$R_4 \leftarrow \Pi_{\langle \text{REGISTA}, \text{TITOLO} \rangle} (R_3)$

22. Per ogni film prodotto dopo il 2000, il codice, il titolo e l'incasso totale di tutte le sue proiezioni

```
SELECT f.CODFILM, f.NOME, SUM(p.INCASSO)
FROM (FILM AS f JOIN PROIEZIONE AS p ON f.CODFILM = p.CODFILM)
WHERE f.ANNOPRODUZIONE > 2000
GROUP BY f.CODFILM, f.NOME
```

23. Il numero di attori dei film in cui appaiono solo attori nati prima del 1970

```
SELECT a.ANNONASCITA, COUNT(*)
FROM (FILM AS f JOIN RECITA AS r ON f.CODFILM = r.CODFILM) JOIN ATTORE AS a
ON r.CODATTORE = a.CODATTORE
WHERE a.ANNONASCITA < 1970
```

24. Per ogni film di fantascienza, il titolo e l'incasso totale di tutte le sue proiezioni

✓

```
SELECT f.NOME, SUM(p.INCASSO)
FROM (FILM AS f JOIN PROIEZIONI AS p ON f.CODFILM = p.CODFILM)
WHERE f.GENRE = "FANTASCIENZA"
```

25. Per ogni film di fantascienza il titolo e l'incasso totale di tutte le sue proiezioni successive al 1/1/01

✓

```
SELECT f.NOME, SUM(p.INCASSO)
FROM (FILM AS f JOIN PROIEZIONI AS p ON f.CODFILM = p.CODFILM AND
p.DATAPROIEZIONE > "1/1/01")
WHERE f.GENRE = "FANTASCIENZA"
```

26. Per ogni film di fantascienza che non è mai stato proiettato prima del 1/1/01 il titolo e l'incasso totale di tutte le sue proiezioni

```
SELECT f.NOME, SUM(p.INCASSO)
FROM (FILM AS f JOIN PROIEZIONI AS p ON f.CODFILM = p.CODFILM)
WHERE f.GENRE = "FANTASCIENZA" NOT EXISTS (SELECT *
FROM PROIEZIONE AS p1
WHERE p1.DATAPROIEZIONE < '01-01-2001' AND p1.CODFILM = f.CODFILM) GROUP BY f.TITOLO
```

27. Per ogni sala di Pisa, che nel mese di gennaio 2005 ha incassato più di 20000 €, il nome della sala e l'incasso totale (sempre del mese di gennaio 2005)

```
SELECT s.NOME , sum(p. INCASSO)
FROM SALA AS s JOIN PROIEZIONE AS p ON s.CODSALA = p.CODSALA
WHERE p.DATAPROIEZIONE BETWEEN '01-01-2005' AND '31-01-2005' AND s.CITTA = 'PISA'
GROUP BY s.NOME
HAVING sum > 20000 ;
```

28. I titoli dei film che non sono mai stati proiettati a Pisa

```
SELECT f.TITOLO
FROM FILM AS f
WHERE NOT EXISTS (SELECT *
                   FROM PROIEZIONE AS p JOIN SALA AS s ON p.CODSALA = s.CODSALA
                   WHERE s.CITTA = 'PISA' AND p.CODFILM = f.CODFILM )
```

29. I titoli dei film che sono stati proiettati solo a Pisa

```
SELECT f.TITOLO
FROM FILM AS f
WHERE NOT EXISTS (SELECT *
                   FROM PROIEZIONE AS p JOIN SALA AS s ON
                   p.CODSALA = s.CODSALA
                   WHERE s.CITTA <> 'PISA' AND p.CODFILM = f.CODFILM )
```

29. I titoli dei film dei quali non vi è mai stata una proiezione con incasso superiore a 500 €

30. I titoli dei film le cui proiezioni hanno sempre ottenuto un incasso superiore a 500 €

31. Il nome degli attori italiani che non hanno mai recitato in film di Fellini

```
SELECT a.NOME  
FROM ATTORE AS a  
WHERE a.NAZIONALITA = 'ITALIANO' AND NOT EXISTS (SELECT *  
        FROM RECITA AS r JOIN FILM AS f  
        ON r.CODFILM = f.CODFILM  
        WHERE f.REGISTA = 'F.FELLINI' AND  
              r.CODATTORE = a.CODATTORE)
```

32. Il titolo dei film di Fellini in cui non recitano attori italiani

```
SELECT f.TITOLO  
FROM FILM AS f  
WHERE f.REGISTA = 'F.FELLINI' AND NOT EXISTS (SELECT *  
        FROM ATTORE AS a JOIN RECITA AS r  
        ON a.CODATTORE = r.CODATTORE  
        WHERE a.NAZIONALITA = 'ITALIANO' AND  
              r.CODFILM = f.CODFILM )
```

33. Il titolo dei film senza attori

```
SELECT f.TITOLO  
FROM FILM AS f  
WHERE NOT EXISTS (SELECT *  
        FROM RECITA AS r  
        WHERE r.CODFILM = f.CODFILM )
```

34. Gli attori che prima del 1960 hanno recitato solo nei film di Fellini

- Un nodo ha un solo padre

CREATE ASSERTION PADRE_UNICO

CHECK (NOT EXIST (SELECT *))

FROM ARGO AS a1 JOIN ARGO AS a2 ON a1.GODALBERO = a2.GODALBERO
AND a1.NODOTARGET = a2.NODOTARGET AND a1.NODOSORGENTE <>
a2.NODOSORGENTE)

- Una radice non ha padri

CREATE ASSERTION NO_PADRE_RADICE

CHECK (NOT EXIST) SELECT *

FROM ALBERO AS A

WHERE A.CODRANCE IN (SELECT DR.NODTARGET
FROM ARGO AS DR)

))

- Tutti i nodi sono connessi

CREATION ASSERTIO NODI-CONNEXS

CHECK (NOT EXISTS) SELECT *

FROM NODO

WHERE CODNOB NOT IN (SELECT CODRADICE

FROM ALBERO

2

AND NOT EXISTS (SELECT *)

FROM ARCC

WHERE CODNODO = ARLO.NODOTARGET

1

1

Esercizio 01 Si scriva una interrogazione in algebra relazionale che fornisca se valutata il codice degli alberi in cui ogni nodo (tranne le foglie) ha esattamente due discendenti.

$R_1 \leftarrow \Pi_{\text{CODA}} (\text{COMPARCHI})$
 $R_2 \leftarrow \text{PADRE}, \text{CODA} f \text{ COUNT FIGLIO } (\text{COMPARCHI})$
 $R_3 \leftarrow \sigma_{\text{COUNT} \neq 2} (R_1)$
 $R_4 \leftarrow \Pi_{\text{CODA}} (R_3)$
 $R_5 \leftarrow R_1 - R_4$

Esercizio 02 Si scriva una interrogazione in SQL che fornisca coppie di codici di alberi tali che il secondo elemento della coppia è un sottoalbero del primo elemento della coppia (ogni nodo e ogni arco del sottoalbero deve essere nodo e arco, rispettivamente, del primo albero).

```
SELECT a1.CODA , a2.CODA
FROM ALBERO AS a1 JOIN ALBERO AS a2 ON a1.CODA <> a2.CODA
WHERE (NOT EXISTS (
    SELECT *
    FROM COMPNODO AS c
    WHERE c.CODA = a2.CODA AND c.CODN NOT IN
    (SELECT CODN
    FROM COMPNODI
    WHERE c.CODA = a1.CODA)
)
AND NOT EXISTS (
    SELECT *
    FROM COMPARCHI AS c
    WHERE c.CODA = a2.CODA AND (PADRE, FIGLIO) NOT IN
    (SELECT PADRE, FIGLIO
    FROM COMPARCHI
    WHERE CODA = a1.CODA)
))
))
```


- Quando viene rimosso un nodo da un grafo (cancellazione dalla tabella COMPNODI) devono essere rimossi tutti gli archi e tutti i nodi del sottoalbero radicato nel nodo (discendenti);

```
CREATE TRIGGER ELIMINA_ARCHI_NODI
AFTER DELETE ON COMPNODI
FOR EACH ROW
BEGIN
DELETE FROM COMPNODI
WHERE (OLD.CODA, OLD.CODN, COMPNODI.CODN) IN (SELECT CODA, PADRE, FIGLIO
                                                FROM COMPARCHI)
AND COMPNODI.CODA = OLD.CODA
DELETE FROM COMPARCHI
WHERE OLD.CODA = COMPARCHI.CODA AND (OLD.CODN = COMPARCHI.PADRE)
OR OLD.CODN = COMPARCHI.FIGLIO
END
```

Esercizio 01 (Punti 8) Si scriva una espressione in algebra relazionale che se valutata fornisce il nome delle classi che hanno mantenuti invariati (unchanged) tutti i loro metodi in tutte le versioni del progetto.

$R_i \leftarrow \pi_{\text{NONE}}(\text{CLASSE})$

$R_{(GOMNET, V_NET, PRO_NET, NOME_NET, PATH_NET, RIGA1_NET, RIGA2_NET, SEGNATURA_NET, CODC_NET, STATO_NET)} \leftarrow$
 $Tk \rightarrow (METODO)$

$R_3 \leftarrow \text{TI_NORE} \left(\text{STATO_MET} \Leftrightarrow \text{'UNCHANGED'} \text{ (CLASSE } \Delta_{(\text{CODE} = \text{CODE_MET})} \text{ MODO)} \right)$
 $R_4 \leftarrow R_1 - R_3$

Esercizio 02 (Punti 8) Si scriva la query SQL per l'Esercizio 01

```
SELECT C.NOME  
FROM CLASSE AS C  
WHERE NOT EXISTS( SELECT *  
                  FROM METODO AS m  
                  WHERE C.CODC = m.CODC AND m.STATUS <> 'UNCHANGED'  
                );
```

Esercizio 03 (Punti 8) Si implementino nel modo più adeguato i seguenti vicoli

1. Se per una classe esiste una istanza con *Versione* > 0, devono esistere istanze della classe per tutti i valori di versione compresi tra 0 e *Versione*.
 2. Se la segnalazione riguardante un metodo è stata assegnata a un autore, questi deve essere anche l'autore del metodo.
 3. La data di chiusura di una segnalazione se presente deve essere successiva alla data di apertura e la descrizione non deve essere nulla;
 4. Non ci può essere più di una segnalazione aperta (*DataChiusura* a NULL) per un metodo.

① CREATE ASSERTION CHECK_VERSION

```
    CHECK (NOT EXISTS (SELECT *
                        FROM CLASSE AS C
                       WHERE C.VERSIONE > 0
                     AND C.VERSIONE NOT IN (SELECT C.VERSIONE
                                              FROM CLASSE AS C2
                                             WHERE C2.VERSIONE BETWEEN 0
                                               AND C.VERSIONE
                                            )
                           )
        );
```

2)

```

CREATE ASSERTION CHECK_AUTORE_METODO_SIGNALAZIONE
CHECK ( NOT EXISTS ( SELECT *
                      FROM SIGNALAZIONE AS S JOIN ASSEGNAZIONE AS A
                      ON S.CODS = A.CODS
                     WHERE S.CODM NOT IN ( SELECT SC.CODM
                                              FROM SCRITTURA AS SC JOIN
                                              AUTORE AS AT ON
                                              SC.CODA = AT.CODA
                                             WHERE SC.CODA = A.CODA
                                         )
                    )
);

```

3) 4- SI POSSONO UTILIZZARE ANCHE SOLO LE ALTER TABLE PERCHÉ È INTRARELAZIONE.

```

CREATE ASSERTION CHECK_DATA_DESCRIZIONE
CHECK ( NOT EXISTS ( SELECT *
                      FROM SIGNALAZIONE AS S
                     WHERE S.DATACHIUSURA <= S.DATAAPERTURA AND
                           S.DESCRIZIONE IS NULL
                   )
);

```

4)

```

CREATE ASSERTION CHECK_UNICA_SIGNALAZIONE
CHECK ( NOT EXISTS ( SELECT *
                      FROM SIGNALAZIONE AS S1 FULL JOIN SIGNALAZIONE S2
                      ON S1.CODS <> S2.CODS AND *.DATACHIUSURA IS NULL
                     WHERE S1.CODM = S2.CODM
                   )
);

```

Esercizio 04 (Punti 8) Si scriva un trigger che viene attivato all'inserimento di una segnalazione per un metodo in SEGNALAZIONE. Il trigger aggiunge nella tabella ASSEGNAZIONE, l'assegnazione della segnalazione a tutti gli autori del metodo (un metodo è in genere scritto da più autori).

```
CREATE TRIGGER CHECK_SEGNALAZIONE
AFTER INSERT INTO SEGNALAZIONE
FOR EACH ROW
BEGIN
INSERT INTO ASSEGNAZIONE
SELECT NEW.CODS, S.CODA, NEW.DATAPERTURA, NEW.DATACHIUSURA, NEW.DESCRIZIO
FROM SCRITTURA AS S
WHERE NEW.CODM = S.CODM AND NEW.VERSIONE = S.VERSIONE
END
```

Esercizio 01 (Punti 9 D) Si scriva una interrogazione in algebra relazionale che, se valutata, fornisce l'identificativo di post commentati solo da amici dell'autore del post.

$R_1 \leftarrow \Pi_{ID_POST} (POST)$

$R_2 \leftarrow \rho_{COMMENTO_IDPOST, COMMENTO_ORDINE, COMMENTO_TESTO, COMMENTO_AUTORE, COMMENTO_DATA, COMMENTO_ORA} (COMMENTO)$

$R_3 \leftarrow POST \bowtie_{COMMENTO_IDPOST = COMMENTO_IDPOST} R_2$

$R_4 \leftarrow R_3 \bowtie_{AUTORE = IDUTENTE1 \text{ AND } COMMENTO_AUTORE = IDUTENTE2} AMICI$

$R_5 \leftarrow \Pi_{IDCOMMENTO, IDPOST, COMMENTO_ORDINE} (R_3)$

$R_6 \leftarrow \Pi_{IDCOMMENTO, IDPOST, COMMENTO_ORDINE} (R_4)$

$R_7 \leftarrow (R_5 - R_6)$

$R_8 \leftarrow \Pi_{IDCOMMENTO, IDPOST} (R_7)$

$R_9 \leftarrow (R_1 - R_8)$

Esercizio 02 (Punti 9 B) Si scriva una interrogazione SQL che fornisce l'identificativo di post e il tema del post ed il post è tale che il suo testo contiene tutte le parole associate al tema.

```
SELECT p.IDPOST, pr.TEMA
FROM POST AS p NJOIN PRESENZA AS pr
WHERE NOT EXISTS (SELECT *
                   FROM POST AS p1 NJOIN PRESENZA AS pr1 NJOIN KEY AS K
                   WHERE p.IDPOST = p1.IDPOST AND p1.TESTO NOT LIKE
                         '%' || K.PAROLA || '%'
)
};
```

Esercizio 03 (Punti 9 D) Si esprimano nel modo più opportuno i seguenti vincoli:

1. L'ordinamento dei commenti deve rispettare l'ordine di data e ora di pubblicazione (non può comparire prima nell'ordine ciò che è stato pubblicato dopo);
2. Se è presente una notifica del post ad un utente o l'utente è amico dell'autore del post o l'utente ha sottoscritto l'interesse al tema del post;
3. Un post viene notificato ad un utente solo una volta;
4. Le sottoscrizioni fanno riferimento a temi presenti e non cancellati.

1)

ALTER TABLE COMMENTO

ALTER COLUMN (DATA,ORA) ORDER BY (DATA DESC, ORA DESC)

2)

CREATE ASSERTION CHECK_AMICO_TEMA

CHECK (NOT EXISTS (SELECT *

FROM NOTIFICA AS m JOIN AMICI AS a

ON m.IDPOST = a.IDUTENTE1

WHERE m.IDUTENTE <> a.IDUTENTE2

) OR

(NOT EXISTS (SELECT *

FROM NOTIFICA AS m JOIN SOTTOSCRIZIONE AS s

ON m.IDUTENTE = s.IDUTENTE JOIN PRESENZA AS p

ON m.IDPOST = p.IDPOST

WHERE s.IDTEMA <> p.IDTEMA

)

));

3)

CREATE ASSERTION CHECK_UNICA_NOTIFICA

CHECK (NOT EXISTS (SELECT *

FROM NOTIFICA AS m1 JOIN NOTIFICA AS m2

ON m1.IDPOST = m2.IDPOST

WHERE m1.IDUTENTE = m2.IDUTENTE

)

);

4)

```

CREATE ASSERTION CHECK_TEMA
CHECK (NOT EXISTS (SELECT *
                    FROM SOTTOSCRIZIONE
                   WHERE TEMA NOT IN ( SELECT TEMA
                                         FROM KEY
                                       )
                  )
      );

```

Esercizio 04 (Punti 9 D) Si scriva una vista che per ogni utente ed ogni mese fornisca un riepilogo dell'attività dell'utente del tipo
(Utente,anno,mese,N.post,N.nocom, N.com_nonsottoscrittori). (N.post il numero di post dell'utente nel mese, N.com_nonomici il numero di commenti a post dell'utente nel mese fatti da utenti che non sono sottoscrittori del tema del post, N.nocom numero di post senza commenti)

```

CREATE VIEW NUM_POST AS
SELECT p.AUTORE , EXTRACT (YEARS FROM p.DATA) AS ANNO,
       EXTRACT (MONTH FROM p.DATA) AS MESE ,
       COUNT (p.IDPOST) AS N_POST
  FROM POST AS p
 GROUP BY p.AUTORE , ANNO , MESE ;
CREATE VIEW NUM_NOCOM AS
SELECT p.AUTORE AS UTENTE , EXTRACT (YEARS FROM p.DATA) AS ANNO ,
       EXTRACT (MONTH FROM p.DATA) AS MESE ,
       COUNT (p.IDPOST) AS N_NOCOM
  FROM POST AS p LEFT JOIN COMMENTO AS C ON p.IDPOST = C.IDPOST
 WHERE C.IDPOST IS NULL
 GROUP BY p.AUTORE , ANNO , MESE ;

```

```

CREATE VIEW NUM_COMNOSOTTOSCRITORI AS
SELECT p.AUTORE AS UTENTE , EXTRACT (YEARS FROM p.DATA) AS ANNO ,
       EXTRACT (MONTH FROM p.DATA) AS MESE ,
       COUNT (C.IDPOST) AS N_COMNOSOTTOSCRITORI
  FROM POST AS p JOIN COMMENTO AS C ON p.IDPOST = C.IDPOST LEFT JOIN
 PRESENZA AS PR ON PR.IDPOST = P.IDPOST
 WHERE PR.TEMA IS NOT NULL AND C.AUTORE NOT IN (

```

```
SELECT S.IDUTENTE  
FROM SOTTOSCRIZIONE AS S JOIN PRESENZA AS PR ON S.TEMA = PR.TEMA  
JOIN POST AS P2 ON PR.IDPOS = P2.IDPOST  
WHERE P2.IDPOST = P.IDPOST )  
GROUP BY UTENTE, ANNO, MESE ;
```

```
CREATE VIEW LASTVIEW  
SELECT *  
FROM NUM_POST AS M NATURAL LEFT JOIN NUM_NOSOTTOSCRITTI AS ANNO  
NATURAL LEFT JOIN NUM_NOCOM
```

ESERCITAZIONE ALBERI

Esercizio 01 Si scriva una interrogazione in algebra relazionale che fornisca se valutata il codice degli alberi in cui ogni nodo (tranne le foglie) ha esattamente due discendenti.

ALGEBRA

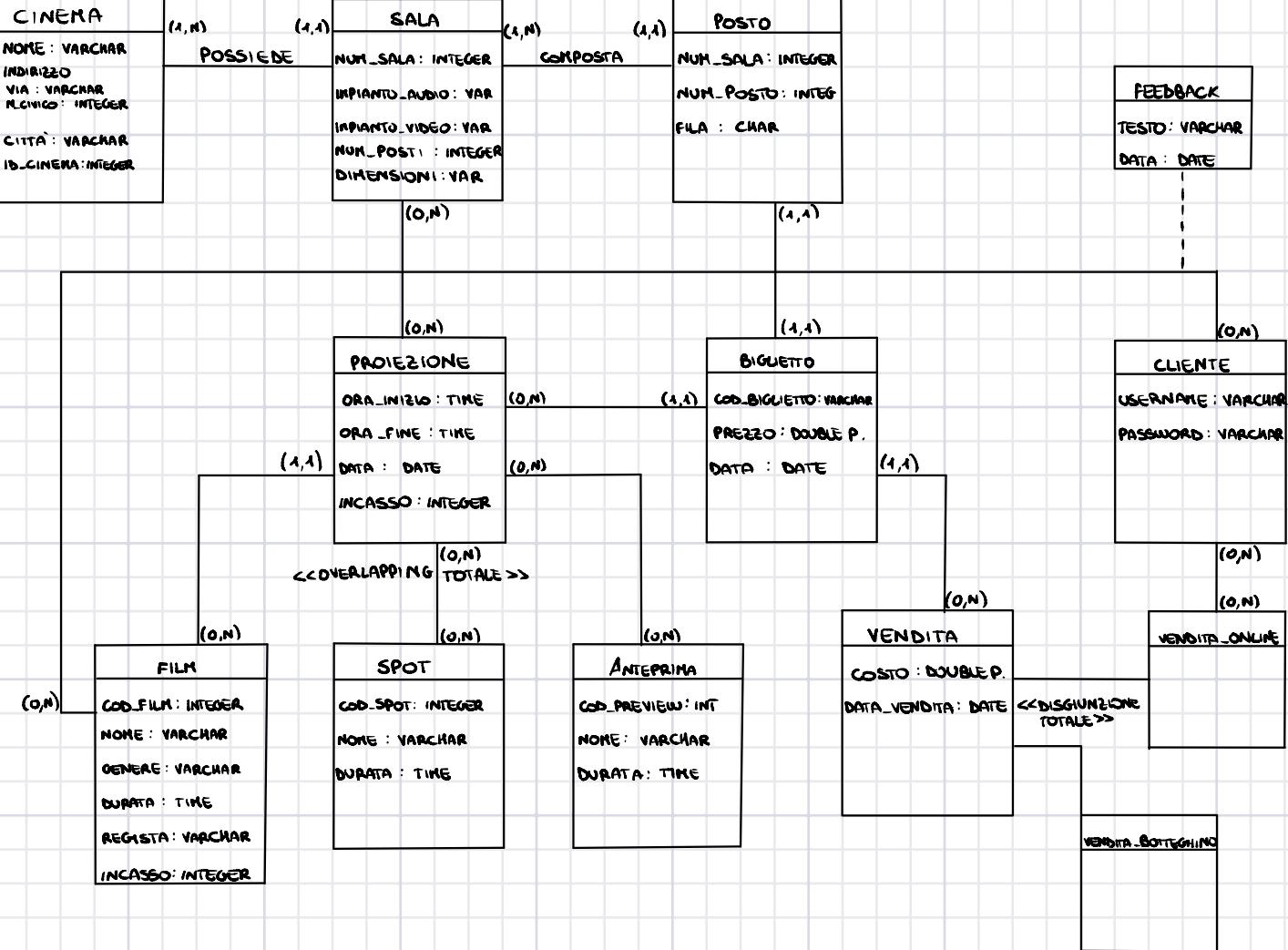
$R_1 \leftarrow \Pi_{\text{CODA}} (\text{ALBERI})$
 $R_2 \leftarrow \text{CODA, PADRE} f_{\text{COUNT FIGLI}} (\text{COMPARCHI})$
 $R_3 \leftarrow \Pi_{\text{FIGLIO} \neq 2} (R_2)$
 $R_4 \leftarrow \Pi_{\text{CODA}} (R_3)$
 $R_5 \leftarrow (R_1 - R_4)$

SQL

```
SELECT a.CODA
FROM ALBERO AS a
WHERE (NOT EXISTS (SELECT *
                   FROM COMPARCHI AS ca
                   WHERE ca.CODA = a.CODA
                   GROUP BY CODA, PADRE
                   HAVING COUNT(*) <> 2
                  )
      );
```

Esercizio 02 Si scriva una interrogazione in SQL che fornisca coppie di codici di alberi tali che il secondo elemento della coppia è un sottoalbero del primo elemento della coppia (ogni nodo e ogni arco del sottoalbero deve essere nodo e arco, rispettivamente, del primo albero).

```
SELECT a1.CODA, a2.CODA
FROM ALBERO AS a1 JOIN ALBERO AS a2 ON a1.CODA <> a2.CODA
WHERE (NOT EXISTS ( SELECT *
                     FROM NODI AS m NATURAL JOIN COMPNODI AS cm
                     WHERE cm.CODA = a2.CODA AND m.CODA NOT IN
                           (SELECT m1.CODA
                            FROM NODI AS m1 NATURAL JOIN COMPNODI AS cm
                            WHERE cm.CODA = a1.CODA
                           )
                    )
      AND
      (NOT EXISTS ( SELECT *
                     FROM COMPARCHI AS cq
                     WHERE cq.CODA = a2.CODA AND cq.CODA NOT IN
                           (SELECT cq1.CODA
                            FROM COMPARCHI AS cq1 WHERE cq1.CODA = a1.CODA)
                    )
      );
```



ESERCITAZIONE CINEMA IN ALGEBRA RELAZIONALE

1. Il nome di tutte le sale di Pisa

$$R_1 \leftarrow \Pi_{\langle \text{NOME_SALA} \rangle} (\sigma_{\langle \text{CITTÀ} = 'PISA' \rangle} (\text{SALA}))$$

2. Il titolo dei film di F. Fellini prodotti dopo il 1960.

$$\Pi_{\langle \text{TITOLO} \rangle} (\sigma_{\langle \text{REGISTA} = 'F.FELLINI' \text{ AND } \text{ANNOPRODUZIONE} > 1960 \rangle} (\text{FILM}))$$

3. Il titolo e la durata dei film di fantascienza giapponesi o francesi prodotti dopo il 1990

$$\Pi_{\langle \text{TITOLO}, \text{DURATA} \rangle} (\sigma_{\langle \text{GENERE} = 'FANTASCIENZA' \text{ AND } \text{NAZIONALITÀ} = 'GIAPPONESE' \text{ OR } \langle \text{NAZIONALITÀ} = 'FRANCESE' \text{ AND } \text{ANNOPRO} > 1990 \rangle} (\text{FILM}))$$

4. Il titolo dei film di fantascienza giapponesi prodotti dopo il 1990 oppure francesi

//

5. I titoli dei film dello stesso regista di "Casablanca"

$$R_1 \leftarrow \Pi_{\langle \text{REGISTA} \rangle} (\sigma_{\langle \text{TITOLO} = 'CASABLANCA' \rangle} (\text{FILM}))$$
$$R_2 \leftarrow \Pi_{\langle \text{TITOLO} \rangle} (\sigma_{\langle \text{REGISTA} = R_1.\text{REGISTA} \rangle} (\text{FILM}))$$

6. Il titolo ed il genere dei film proiettati il giorno di Natale 2004

$$\Pi_{\langle \text{TITOLO}, \text{GENERE} \rangle} (\sigma_{\langle \text{DATAPROIEZIONE} = '25-12-2004' \rangle} (\text{FILM} * \text{PROIEZIONE}))$$

7. Il titolo ed il genere dei film proiettati a Napoli il giorno di Natale 2004

$$R_1 \leftarrow \Pi_{\langle \text{TITOLO}, \text{GENERE} \rangle} (\sigma_{\langle \text{DATAPROIEZIONE} = '25-12-2004' \rangle} (\text{FILM} * \text{PROIEZIONE}))$$
$$R_2 \leftarrow \Pi_{\langle \text{TITOLO}, \text{GENERE} \rangle} (\sigma_{\langle \text{CITTÀ} = 'NAPOLI' \rangle} (R_1))$$

8. I nomi delle sale in cui il giorno di Natale 2004 è stato proiettato un film con R.Williams

$$\begin{aligned}R_1 &\leftarrow \Pi_{\text{CODFILM}} > (\sigma_{\text{CNAME} = 'R.WILLIAMS'} > (\text{RECITA} * \text{ATTORE})) \\R_2 &\leftarrow \Pi_{\text{CODFILM}, \text{CODSALA}} > (\sigma_{\text{DATAPROIEZIONE} = '25-12-2004'} > (R_1 * \text{PROIEZIONE})) \\R_3 &\leftarrow \Pi_{\text{CNAME}} > (\sigma_{\text{CITTA} = 'NAPOLI'} > (R_2 * \text{SALA}))\end{aligned}$$

9. Il titolo dei film in cui recita M. Mastroianni oppure S.Loren

$$\begin{aligned}R_1 &\leftarrow \Pi_{\text{CODFILM}} > (\sigma_{\text{CODA} = \text{CODA_R} \text{ AND } \text{NAME} = 'M.MASTROIANNI' \text{ OR } \text{NAME} = 'S.LOREN'} > (\text{RECITA} * \text{ATTORE})) \\R_2 &\leftarrow \Pi_{\text{TITOLO}} > (R_1 * \text{FILM})\end{aligned}$$

10. Il titolo dei film in cui recitano M. Mastroianni e S.Loren