



Corso Luigi Einaudi, 55 - Torino

Appunti universitari

Tesi di laurea

Cartoleria e cancelleria

Stampa file e fotocopie

Print on demand

Rilegature

NUMERO: 805

DATA: 31/01/2014

A P P U N T I

STUDENTE: Greco

MATERIA: Basi di Dati Esercizi

Prof. Baralis

Il presente lavoro nasce dall'impegno dell'autore ed è distribuito in accordo con il Centro Appunti.

Tutti i diritti sono riservati. È vietata qualsiasi riproduzione, copia totale o parziale, dei contenuti inseriti nel presente volume, ivi inclusa la memorizzazione, rielaborazione, diffusione o distribuzione dei contenuti stessi mediante qualunque supporto magnetico o cartaceo, piattaforma tecnologica o rete telematica, senza previa autorizzazione scritta dell'autore.

**ATTENZIONE: QUESTI APPUNTI SONO FATTI DA STUDENTIE NON SONO STATI VISIONATI DAL DOCENTE.
IL NOME DEL PROFESSORE, SERVE SOLO PER IDENTIFICARE IL CORSO.**

Algebra relazionale.

15/10/201

Esercizio 1.

Dato lo schema relazionale costituito dalle tabelle (le chiavi primarie sono sottolineate):

RIVISTA (CodR, NomeR, Editore)

ARTICOLO (CodA, Titolo, Argomento, CodR)

esprimere in algebra relazionale le seguenti interrogazioni:

a) Trovare il codice e il nome delle riviste^T che hanno pubblicato almeno un articolo^M di argomento 'motorismo', σ

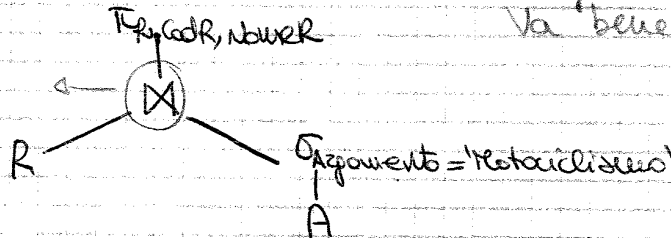
- selezione
- join (sceglie da R qll che hanno corrispondenza con argomento motoric)
- proiezione

Versione più efficiente.

Versioni equivalenti.
Se faccio prima join e poi selezione è uguale

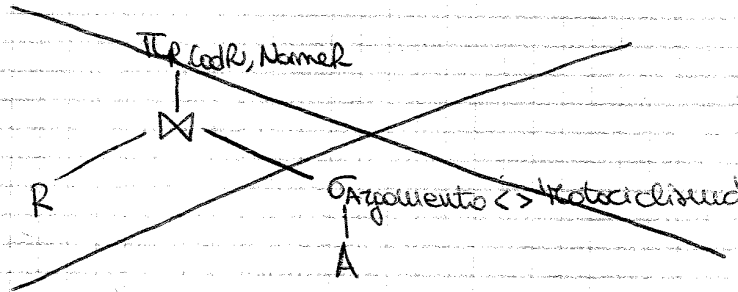
Va bene anche theta-jo

Non scrivere un'equivalente a p: $R.CodR = A.CodR$ (non necessario)



b) Trovare il codice e il nome delle riviste^T che non hanno mai pubblicato articoli di argomento 'motorismo', σ

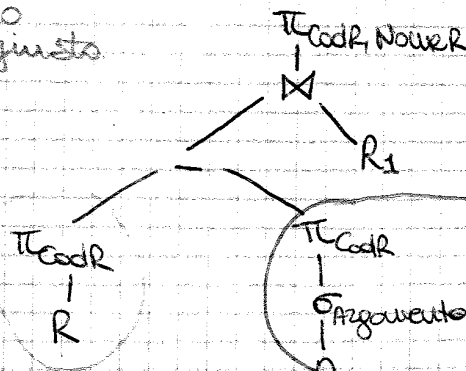
- selezione
- join
- proiezione



R	CodR	NomeR	Editore
	1		
	2		

A	CodA	Titolo	Argomento	CodR
	A1	Moto		1
	A2	Auto		1
	A3	Auto		2

Questo soluz da anche le riviste che pubblicano il prossimo mese, ecco qui, ed è giusto

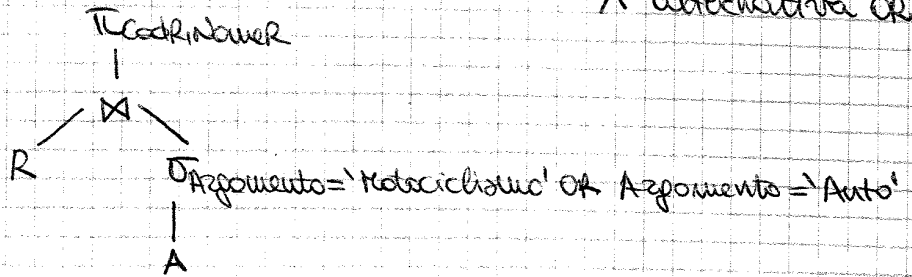


Insieme da escludere: CodR che hanno pubblicato almeno un articolo di

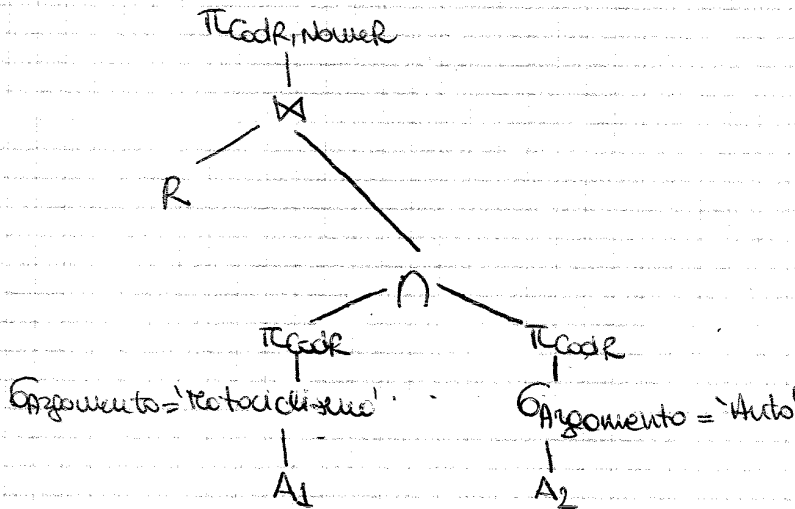
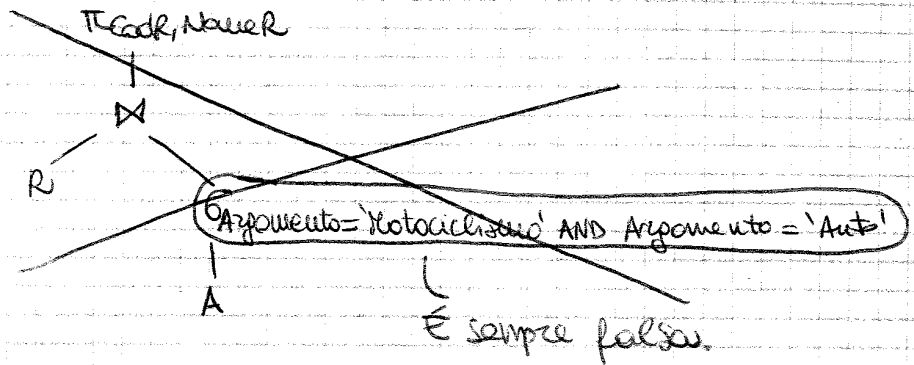
Attenzione: la differenza richiede la compatibilità di schema.

2) Trovare il codice e il nome delle riviste che pubblicano articoli di motoriciclismo oppure di auto.

^ alternativa OR

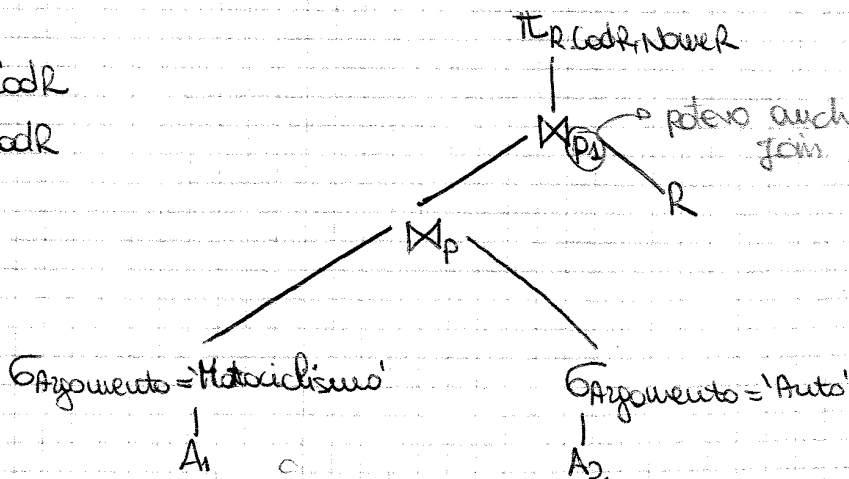


3) Trovare il codice e il nome delle riviste che pubblicano articoli sia di motoriciclismo sia di auto.



1: $R.CodR = A_1.CodR$

2: $A_1.CodR = A_2.CodR$



2) Trovare il codice e il nome delle riviste che hanno pubblicato solo articoli di motociclismo.

```

DISTINCT
SELECT R.CodR, NomeR
FROM RIVISTA R, ARTICOLO A
WHERE R.CodR NOT IN (SELECT CodR
                     FROM ARTICOLO A
                     WHERE Argomento <> 'Motociclismo')
AND R.CodR = A.CodR;
    
```

Riferisce: Trovare cod. e nome riviste che non hanno mai pubblicato articoli con argomento ≠ motociclismo, e che abbiano pubblicato almeno un articolo.

```

SELECT CodR, NomeR
FROM RIVISTA
WHERE CodR NOT IN (SELECT CodR
                  FROM ARTICOLO A
                  WHERE Argomento <> 'Motociclismo')
AND CodR IN (SELECT CodR
            FROM ARTICOLO);
    
```

3) Trovare il codice e il nome delle riviste che hanno pubblicato almeno 2 articoli di motociclismo.

```

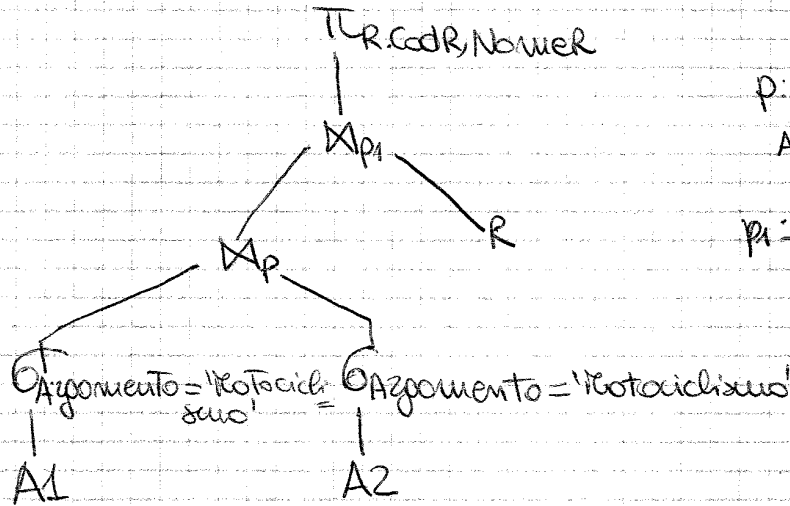
SELECT R.CodR, NomeR
FROM RIVISTA R, ARTICOLO A
WHERE R.CodR = A.CodR AND Argomento = 'Motociclismo'
GROUP BY R.CodR, NomeR
HAVING COUNT(*) ≥ 2; >1
    
```

```

SELECT CodR, NomeR
FROM RIVISTA
WHERE CodR IN (SELECT CodR
              FROM ARTICOLO
              WHERE Argomento = 'Motociclismo'
              GROUP BY CodR
              HAVING COUNT(*) > 1);
    
```

```
SELECT DISTINCT R.CodR, NomeR
FROM Rivista R, Articolo A1, Articolo A2
WHERE R.CodR=A1.CodR AND R.CodR=A2.CodR AND
      A1.Argomento='Motociclismo' AND A2.Argomento='Auto';
```

2) Trovare il codice e il nome delle riviste che hanno pubblicato almeno 2 articoli di motociclismo.



$p: A1.CodR=A2.CodR$
 $AND A1.CodA \neq A2.CodA$

$p_1: R.CodR=A1.CodR$

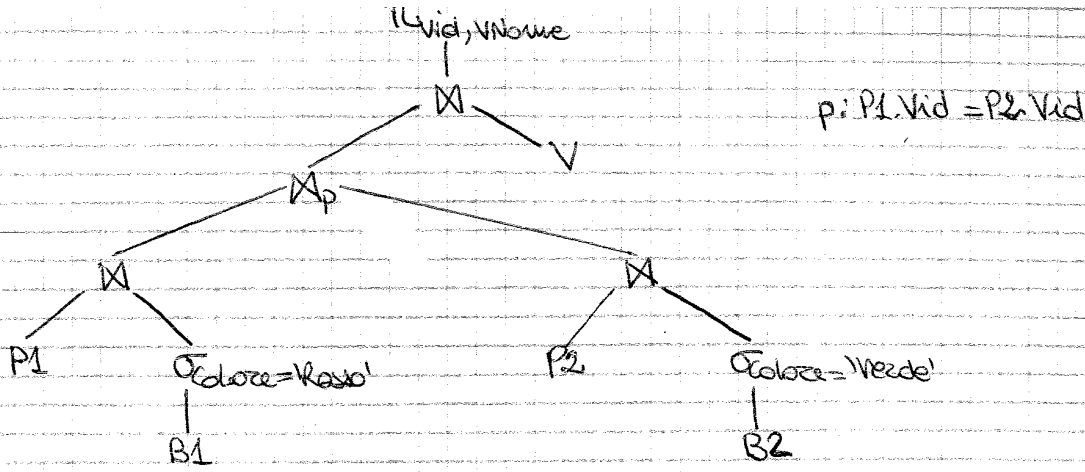
3) Trovare il codice e il nome delle riviste che hanno pubblicato un solo articolo di motociclismo (possono aver scritto quanti articoli desiderano relativamente ad altri argomenti).

```
SELECT R.CodR, NomeR
FROM Articolo A, Rivista R
WHERE R.CodR=A.CodR AND Argomento='Motociclismo'
GROUP BY R.CodR, NomeR
HAVING COUNT(*)=1;
```

L'algebra di questa richiesta è molto complicata.

Sobbiamo pensarla come 2 sottoproblemi e poi sottrarli tra loro.

- 1) CodR che hanno pubblicato almeno un articolo di motociclismo.
 - (meno)
- 2) CodR che hanno pubblicato più di un articolo di motociclismo
 (almeno due)



SELECT Vid, VNome

FROM Velisti

WHERE Vid IN (SELECT Vid
 ① FROM Prenotazioni P1, Barche B1
 WHERE P1.Bid = B1.Bid AND Colore = 'Rosso')

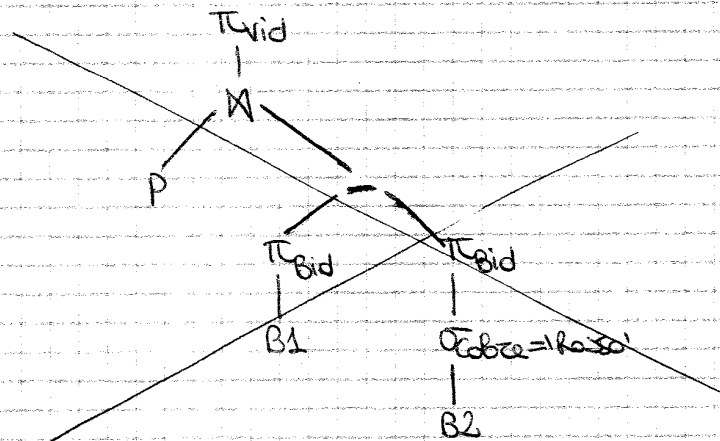
AND Vid IN (SELECT Vid
 ② FROM Prenotazioni P2, Barche B2
 WHERE P2.Bid = B2.Bid AND Colore = 'Verde');

SELECT DISTINCT V.Vid, VNome

FROM Velisti V, Prenotazioni P1, Prenotazioni P2, Barche B1, Barche B2

WHERE B1.Bid = P1.Bid AND P2.Bid = B2.Bid AND
 (P1.Vid = V.Vid AND P2.Vid = V.Vid) AND ^{oppure} P1.Vid = P2.Vid AND
 B1.Colore = 'Rosso' AND B2.Colore = 'Verde'; V.Vid = P2.Vid

3) Trovare i codici dei velisti che non hanno mai prenotato una barca rossa.



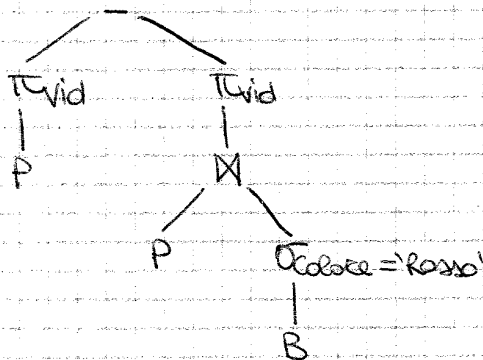
L'insieme da escludere non è un insieme di barche.

SELECT Vid, VNome

FROM Velisti

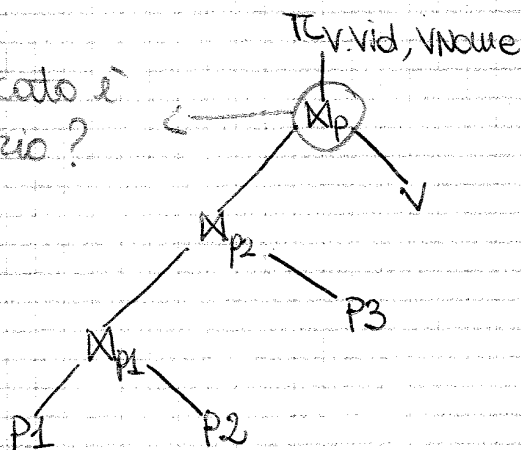
WHERE Vid NOT IN (SELECT Vid
 FROM Barche B, Prenotazioni P
 WHERE P.Bid=B.Bid AND Colore='Rosso');

Velisti che hanno prenotato solo barche non rosse



Trovare i codici e i nomi dei velisti che hanno prenotato almeno tre barche.

Il predicato è obbligatorio?



p1: P1.Vid = P2.Vid and
 P1.Bid <> P2.Bid

p2: P1.Vid = P3.Vid and
 P1.Bid <> P3.Bid and
 P2.Bid <> P3.Bid

p: V.Vid = P1.Vid

SELECT Vid, VNome

FROM Velisti V, Prenotazioni P

WHERE V.Vid = P.Vid

GROUP BY Vid, VNome

HAVING COUNT(*) >= 3;

SECCIZIO 01/09/2009.

LOCALE (CodL, Indirizzo, Città, MetaQuadrati)

INTESTATARIO (CodFiscale, Nome)

CONTRATTO-AFFITTO (CodC, CodL, MeseInizio, MeseFine, CostoMensile)

INTESTATARI-CONTRATTO (CodC, CodFiscale)

) Visualizzare le codice, l'indirizzo, e la città dei locali che sono stati affittati in periodi diversi con lo stesso importo mensile.

SELECT CodL, Indirizzo, Città

FROM locale

WHERE CodL IN (SELECT CodL

FROM Contratto-affitto

GROUP BY CodL, CostoMensile,

HAVING COUNT(*) > 1);

tra le partizioni (gruppi) che hanno lo stesso locale e lo stesso costo mensile.

oppure count(DISTINCT meseInizio)

Nella 2ª esecuzione di laboratorio c'è un esercizio simile.

B: La select non è vincolata nella group by (non è detto che tutti i campi della group by devono essere nella select), però quelli che sono in select devono poter essere messi in group by.

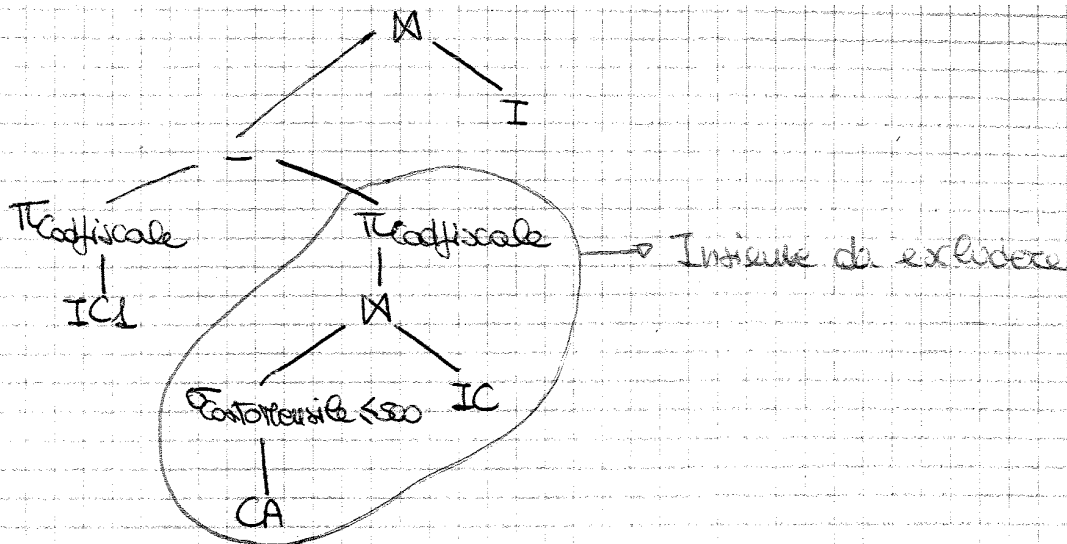
CONTRATTO-AFFITTO

CodC	CodL	MeseInizio	MeseFine	CostoMensile
C1	L1	Genn 2012	Mar 2012	400
C2	L1	Aprile 2012	Giug 2012	400
C3	L1	Luglio 2012	Sett. 2012	500
C4	L2	Genn 2012	Marzo 2012	400

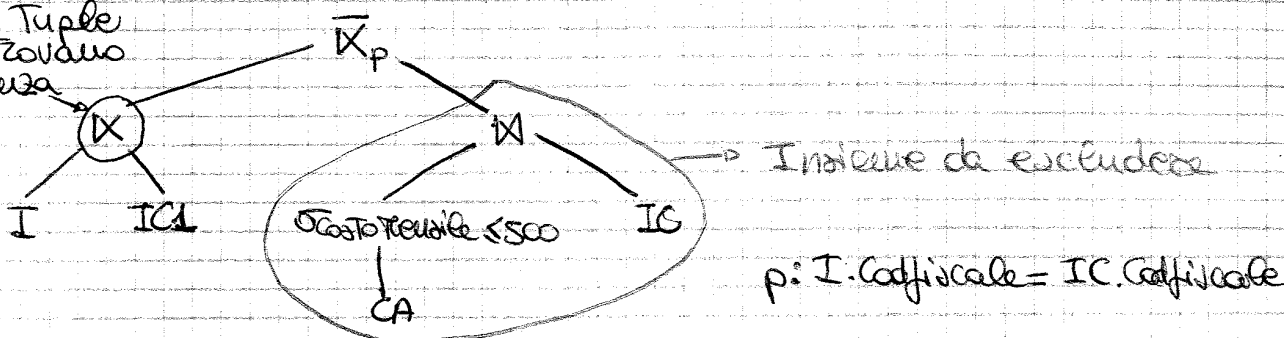
→ Gruppo 1

applicano gruppi con lo stesso CodL e lo stesso costoMensile. alla selezione esce solo il locale L1, gruppo 2 e 3 non vengono visualizzati

B: In SQL tutte le interrogazioni che si possono risolvere con aggregati si risolvono con aggregati e non con il join:



ende solo Tuple
I che trovano
corrispondenza
IC1.



INTESTATARIO

Codfiscale	Nome
xxx	Mario

CONTRATTO-AFFITTO

CodC	CodL	...	CostoRendite
C1	L1		400
C2	L2		600

INTESTATARI-CONTRATTO

CodC	Codfiscale
C1	xxx
C2	xxx

Se come insieme da escludere uso quello del CodC allora viene eliminato xxx e non va bene perché anzi come risultato l'insieme vuoto.

Visualizzare il codice, l'indirizzo, la città e i metri quadri di superficie del locale con la superficie maggiore di ogni città.

Non si può fare in algebra.

SELECT CodL, Città, Indirizzo, MetriQuadrati
FROM locale L

WHERE MetriQuadrati = (SELECT MAX(MetriQuadrati)
FROM locale L1
WHERE (L1.Città = L.Città));

Calcola il MAX per i locali della stessa città della riga corrente di L.

Condizioni di correlazione

```

SELECT NomeA, Nazione
FROM Atleta A, Pratica-Specialità S
WHERE A.CodA = S.CodA AND A.CodA NOT IN (
    SELECT CodA
    FROM Pratica-Specialità
    WHERE Specialità <> 'slalom gigante')
    
```

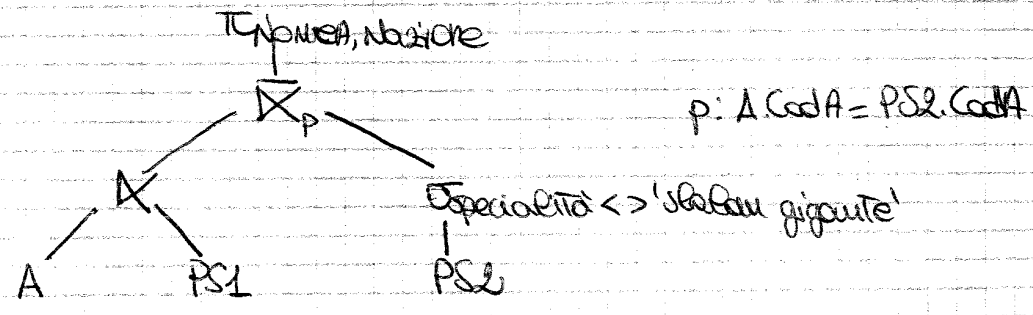
garantisce che l'atleta pratici almeno una specialità.

Insieme da escludere.

```

SELECT NomeA, Nazione
FROM Atleta
WHERE CodA NOT IN (
    SELECT CodA
    FROM Pratica-Specialità
    WHERE Specialità <> 'slalom gigante')
    AND CodA IN (
    SELECT CodA
    FROM Pratica-Specialità);
    
```

garantisce che l'atleta pratici almeno una specialità.



2) Per gli atleti che hanno gareggiato per almeno 2 anni, visualizzare il nome, la data di nascita, la nazione, il numero complessivo di gare effettuate e il numero di anni in cui hanno partecipato a gare.

```

SELECT NomeA, dataNascita, Nazione, COUNT(*), COUNT(DISTINCT AnnoEffettuazione)
FROM Atleta A, Partecipa-Gara PG
WHERE A.CodA = PG.CodA
GROUP BY A.CodA, NomeA, dataNascita, Nazione
HAVING COUNT(DISTINCT AnnoEffettuazione) >= 2;
    
```

min. anno # in cui hanno gareggiato

```
SELECT COUNT(*), COUNT(DISTINCT NomePersona)
FROM Questionario
WHERE CodQ IN (
    SELECT CodQ
    FROM Risposte-Questionario
    GROUP BY CodQ
    HAVING COUNT(*) >= 10)

```

questionari che contengono almeno 10 domande.

→ conta questionari

← conta domande

ATTENZIONE: è obbligato fare il join tra questionari e risposte-questionario.

;) Visualizzare il nome delle persone che hanno risposto esattamente ad almeno il 70% delle domande contenute in almeno uno dei loro questionari

Partiamo da un caso più semplice:

visualizzare il codice dei questionari in cui tutte le risposte sono esatte.

conteggio delle domande del questionario } devono essere uguali.
 conteggio delle risposte esatte. }

```
SELECT CodQ
FROM Domanda D, Risposte-Questionario RQ
WHERE D.CodD = RQ.CodD AND Risposta = RispostaEsatta
GROUP BY CodQ
HAVING COUNT(*) = (
    SELECT COUNT(*)
    FROM Risposte-Questionario RQ1
    WHERE RQ1.CodQ = R.CodQ);

```

Conteggio delle risposte esatte per ogni CodQ

condizione di correlazione.

conteggio delle domande presenti nel questionario correlante.

Se uso NOT IN per fare il 1° passo non va bene poi per fare il 2° passo.

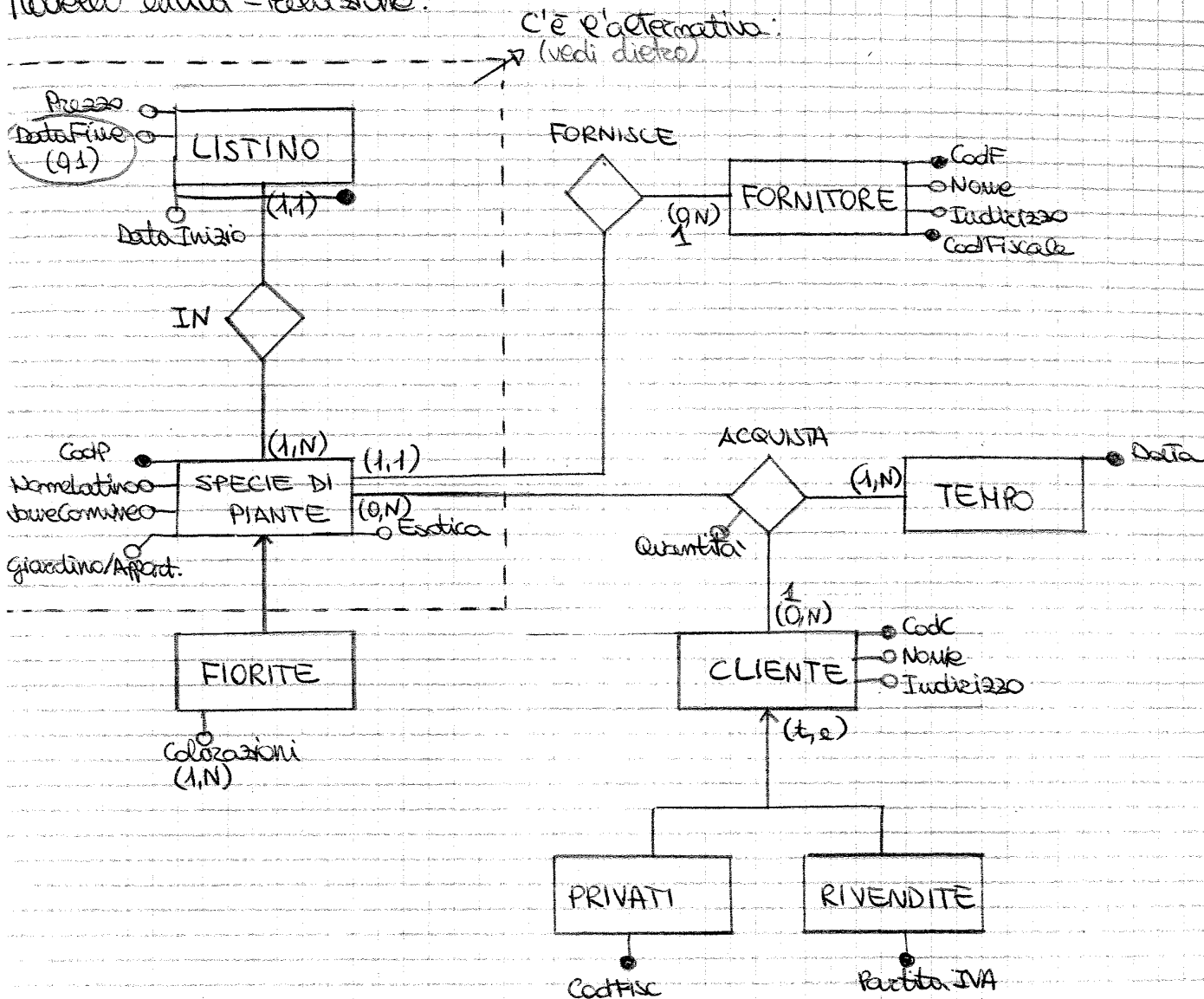
Interrogazione completa:

```
SELECT DISTINCT NomePersona
FROM Questionario Q, Domanda D, Risposte-Questionario RQ
WHERE Q.CodQ = RQ.CodQ AND D.CodD = RQ.CodD AND Risposta = RispostaEsatta
GROUP BY Q.CodQ, NomePersona
HAVING COUNT(*) >= 0,7 * (
    SELECT COUNT(*)
    FROM Risposte-Questionario RQ1
    WHERE RQ1.CodQ = R.CodQ);

```

1. Vendita all'ingrosso di piante.

Modello entità-relazione:



○ (0,1) vuol dire opzionale

o come è fatto questo modello non vengono distinti i diversi acquisti di uno stesso cliente nello stesso giorno, ma anzi la quantità totale acquistata dal cliente in quella data, non gli acquisti separati.

ha senso usare una gerarchia se le subentità hanno attributi, relazioni, quindi per le piante non serve la gerarchia.

nel caso di piante fiorite c'è l'informazione in più e quindi serve un concetto più ricco, cioè una gerarchia, posso decidere di fare 2 sottoinsiemi: fiorite e verdi (gerarchia Totale ed esclusiva) oppure un solo sottoinsieme fiorite (usiamo quest'ultimo). L'informazione su pianta verde è tutto ciò che non è fiorito.

per le colozioni ha senso usare un attributo multi-valore per le colozie oppure uso un'entità per le colozioni ma poiché il colore è meno importante della distinzione fiorite o verdi allora uso attributo multivalore.

Schema logico relazionale: (vanno utilizzate le entità semplificate)

CLIENTE (CodC, Nome, Indirizzo, TipoC, CodFisc*, PIVA*)

SPECIE-DI-PIANTE (CodP, NomeLatino, NomeComune, Giardino/Appart, Esatta, TipoPianta, CodF)

~~COLAZIONI (Collozazione)~~

FORNITORE (CodF, Nome, Indirizzo, CodFisc)

uso solo qst come chiave primaria perché è più semplice per le relazioni.

ISTINO (DataInizio, CodP, Prezzo, DataFine*)

~~EMO (Data)~~

ACQUISTA (CodC, CodP, Data, Quantità)

LA-COLORAZIONI (CodP, Collozazione)

Le tabelle che sono tutte chiave ma sono relazioni, non vanno eliminate, ma solo eliminare tempo e colorazioni.

"Colorazioni" non è obbligatorio eliminarla però in questo caso decidiamo di farlo.

Modi d'integrità referenziale:

SPECIE-DI-PIANTE (CodP) REFERENCES FORNITORE (CodF)

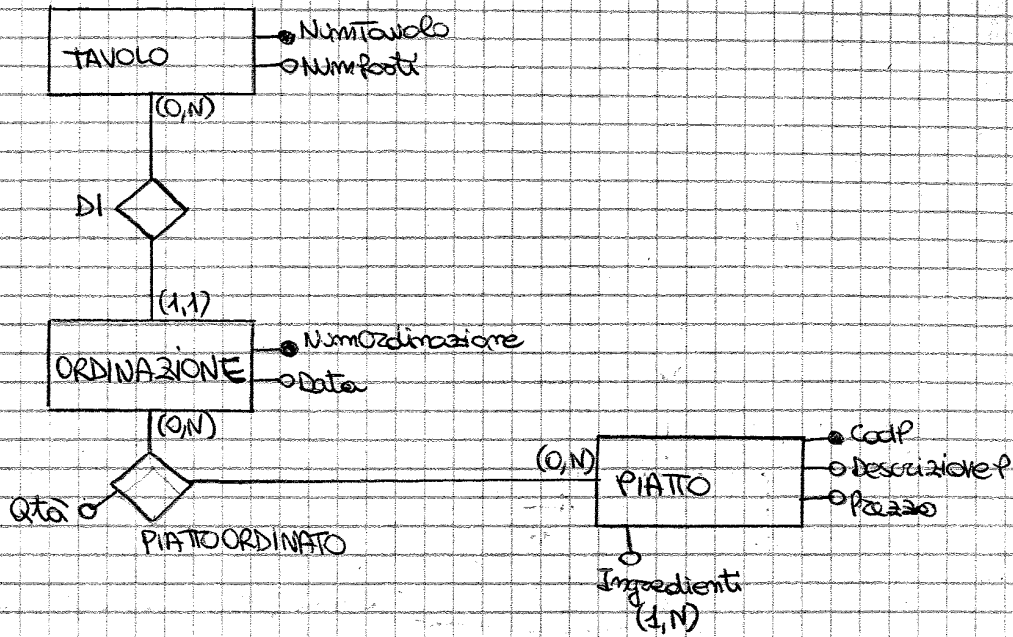
ISTINO (CodP) REFERENCES SPECIE-DI-PIANTE (CodP)

ACQUISTA (CodP) REFERENCES SPECIE-DI-PIANTE (CodP)

ACQUISTA (CodC) REFERENCES CLIENTE (CodC)

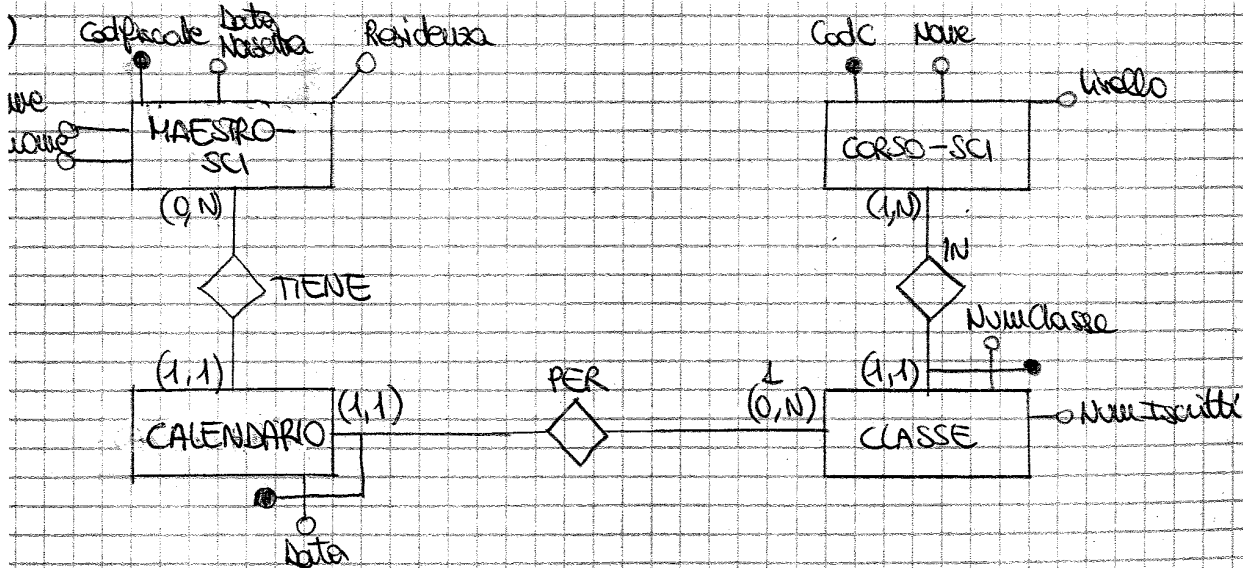
LA-COLORAZIONI (CodP) REFERENCES SPECIE-DI-PIANTE (CodP)

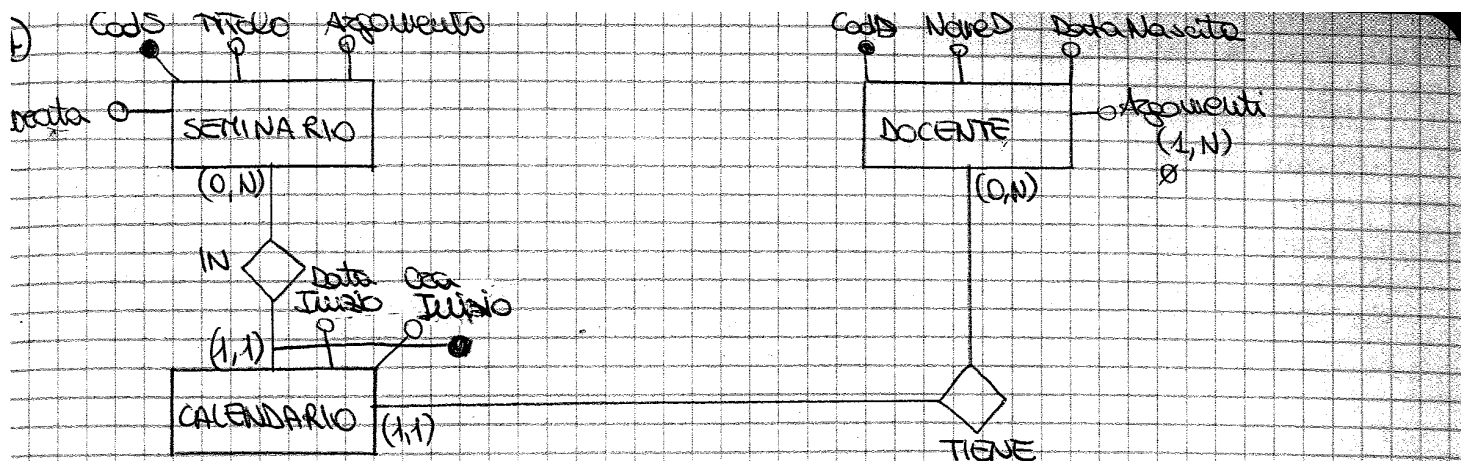
Schema concettuale associato:



```

SELECT P.CodP, DescrizioneP, sum(quantita'), sum(prezzo * quantita')
FROM Piatto P, Ordinazione O, Piatto-Ordinato PO
WHERE P.CodP = PO.CodP AND O.NumOrdinazione = PO.NumOrdinazione AND
Prezzo > 10
GROUP BY P.CodP, data, Descrizione
HAVING COUNT (DISTINCT NumTavolo) >= 10
    
```





insieme da escludere: docenti che hanno tenuto almeno un seminario su un argomento che non fa parte delle loro competenze

```

SELECT D.Cods, Nome D, Titolo S, Count(*)
FROM Docente D, Seminario S, Calendario C
WHERE (D.Cods = C.Cods and S.Cods = C.Cods)
and D.Cods NOT IN (
    SELECT Cods
    FROM Calendario C1, Seminario S1
    WHERE C1.Cods = S1.Cods and Argomento NOT IN
    (
        SELECT Argomento
        FROM Competenze CP
        WHERE CP.Cods = C1.Cods
    )
)
GROUP BY D.Cods, S.Cods, Nome D, Titolo S;
    
```

condizioni di join
insieme da escludere
condizione di correlazione
posso metterlo qui che non c'è che è da chiedere prima
idem titolo S.

Alternativa per l'insieme da escludere:

```

(SELECT Cods
FROM Calendario C1, Seminario S1
WHERE C1.Cods = S1.Cods and ((C1.Cods, S1.Argomento) NOT IN
(SELECT Cods, Argomento
FROM Competenze))
)
    
```

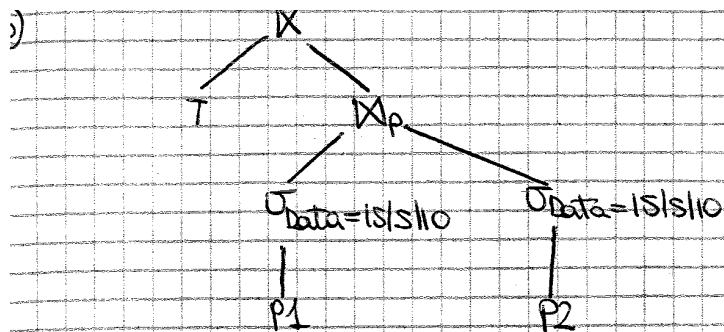
costruttore di tupla
insieme da escludere

Alternativa "non esiste nella tabella COMPETENZE una tupla con quel Cods e quell'argomento" *

```

(SELECT Cods
FROM Calendario C1, Seminario S1
WHERE C1.Cods = S1.Cods AND NOT EXISTS (
    SELECT *
    FROM Competenze CP
    WHERE CP.Cods = C1.Cods and
    CP.Argomento = S1.Argomento
)
)
    
```

insieme da escludere
condizioni di correlazione.
oppo su *



$P1.CodT = P2.CodT$ ← stesso terminata
 and $(P1.OraI = P2.OraI)$ ← stessa ora
 and $(P1.CodC <> P2.CodC$ or
 $P1.CodI <> P2.CodI)$ ← campo diverso

In SQL conteggio con COUNT

```

SELECT T.CodT, NomeT
FROM Terminata T, Promozione P
WHERE T.CodT = P.CodT and data = 15/5/10
GROUP BY (T.CodT, OraI), NomeT
HAVING (COUNT(*) > 1, ← stesso terminata e stessa ora
    
```

Per imporre che l'ora sia la stessa allora metto nel raggruppamento anche l'ora.

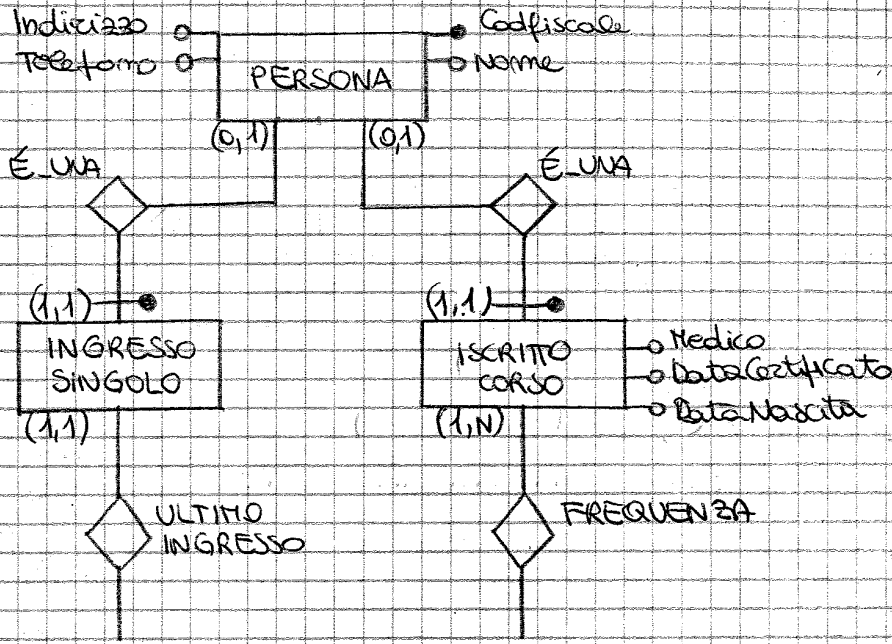
NomeT non posso metterlo nella select senza inserirlo nella group by. Posso aggiungerlo perché nella group by ho già la chiave primaria e non altera i gruppi.

conteggio promozioni (ok guardando la chiave primaria)

↳ alternativa:

$count(distinct CodI) > 1$ or $count(distinct CodC) > 1$

Gerarchia "PERSONA"



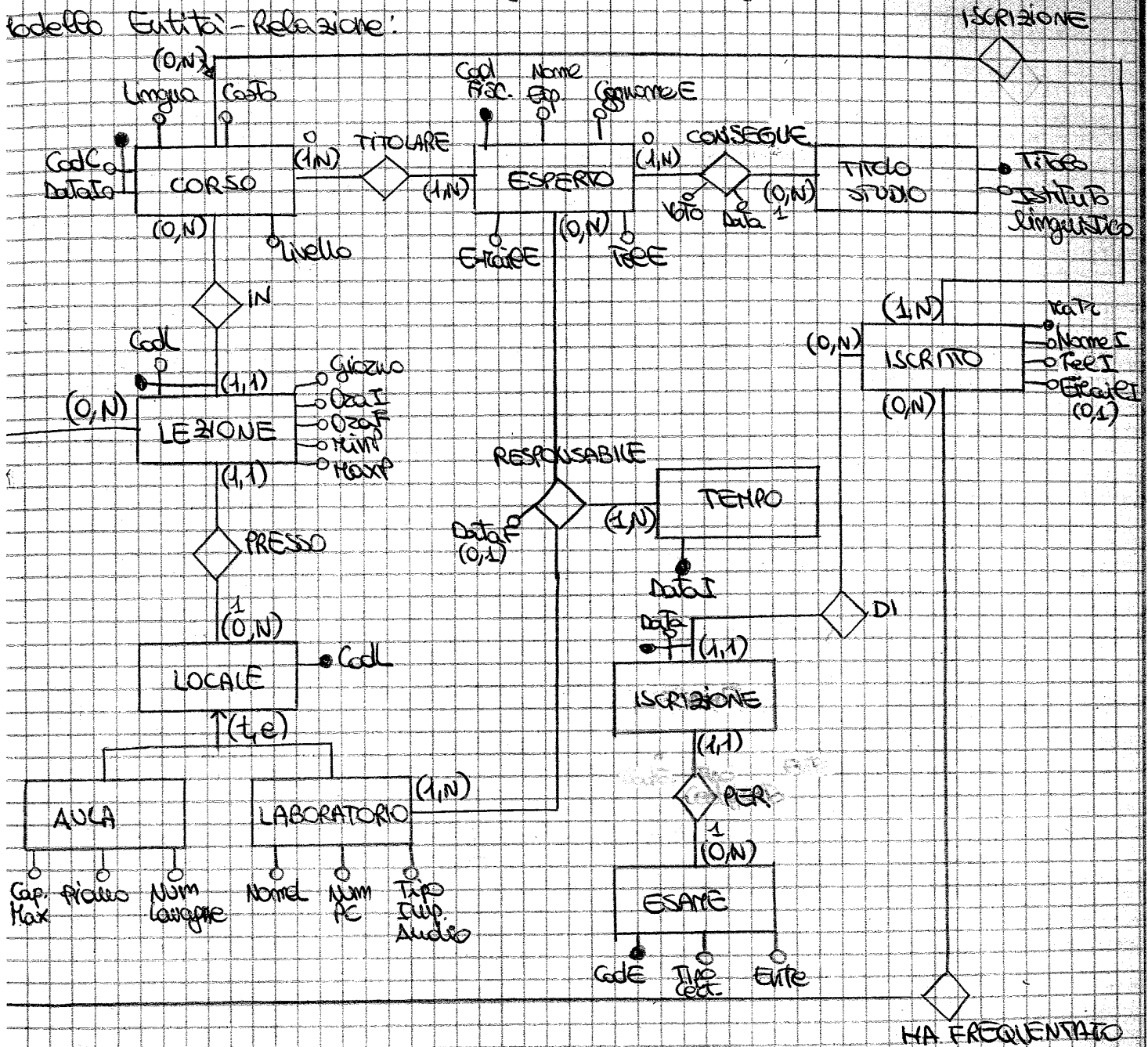
Alternative:

- accorpamento nell'entità padre
- accorpamento nelle entità figlie: l'identificatore ^{persona} CodFiscale rimane univoco perché è assegnato esternamente alle entità nel dominio del progetto.
- soluzione mista:
 - INGRESSO SINGOLO accorpata nel padre
 - ISCRITTO CORSO separata con relazione È UNA

Eliminazione attributo multivalore: "Qualifiche"

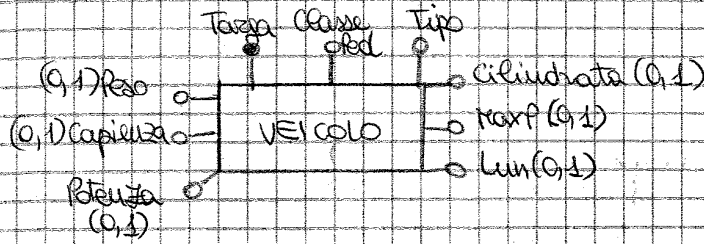


gestione di una scuola che eroga corsi di lingua.
 Modello Entità-Relazione:



- Esiste una relazione di dipendenza tra le relazioni ISCRIZIONE e HA FREQUENTATO che deve essere espressa con un vincolo d'integrità.
- "Non può essere presente in HA FREQUENTATO un legame per cui non esiste il corrispondente legame con corso nella relazione ISCRITTO".

Gerarchia su VEICOLO:



Schema logico-relazionale:

- AUTOSTRADA (CodA, NomeA, KmTot)
- CASELLO (CodC, CodA, NomeC, KmA, Stato)
- TELEPASS (CodT, Versione, Modello)
- UTENTE (CodFiscale, NomeU, CognomeU, Indirizzo, CAP, Città, NumCartaCredito*, NumContoCorrente*)
- PASSAGGI (CodT, CodA, CodC, Tipo, Importo*)
- VEICOLO (Targa, ClassePed, Tipo, MaxP*, Cilindrata*, Lun*, Reso*, Capienza*, Potenza*)
- UTENTE CON TELEPASS (CodT, CodFisc)
- VALIDITÀ VEICOLO (Targa, CodT)
- PASSAGGI AVVENUTI (CodT, CodA, CodC)
- PASSAGGI EFFETTUATI (Targa, CodT, CodA, CodC)
- TELEPASS (CodT, Versione, Modello, CodFisc, Targa)
- PASSAGGI (CodT, Trimestre, Tipo, Importo*, CodC, CodA, Targa)

Eliminando alcune tabelle che nascono dalle relazioni non scrivere cost e quindi Tabelle TELEPASS e PASSAGGI di prima.

Vincoli d'integrità referenziale:

- CASELLO (CodA) REFERENCES AUTOSTRADA (CodA)
- PASSAGGI (CodT) REFERENCES TELEPASS (CodT)
- PASSAGGI (CodA) REFERENCES AUTOSTRADA (CodA) } *
- PASSAGGI (CodC) REFERENCES CASELLI (CodC) }
- PASSAGGI (Targa) REFERENCES VEICOLI (Targa)
- TELEPASS (CodFisc) REFERENCES UTENTI (CodFisc)
- TELEPASS (Targa) REFERENCES VEICOLI (Targa)
- * PASSAGGI (CodA, CodC) REFERENCES CASELLI (CodA, CodC)
- VALIDITÀ VEICOLO (CodT) REFERENCES TELEPASS (CodT)
- VALIDITÀ VEICOLO (Targa) REFERENCES VEICOLO (Targa)

Quando si incorpora nel padre bisogna mettere dei vincoli che vadano a verificare che siano rispettati i precedenti attributi relativi ad ogni figlia.

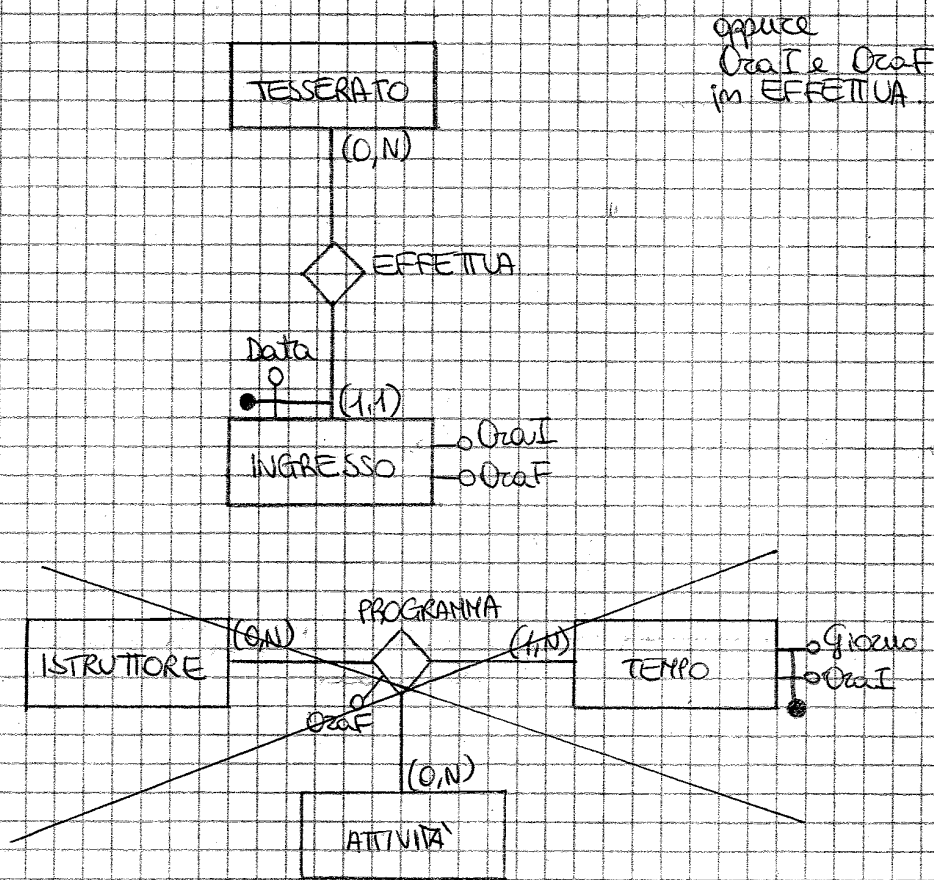
(0,N) è preferibile mettere il però va bene anche 1.

(0,N) perché dice possibili attrazzi

Introduco l'identificatore CodP per PACCHETTO perché non vi sono altri attributi adatti per l'identificazione.

Alternativa per identificatore di PROGRAMMA (Giorno, OraI, Sala) che però non consentiva di garantire il vincolo sugli istruttori (che non possono svolgere 2 lezioni contemporaneamente).

Modello equivalente per INGRESSO:



Questa soluzione permette che una coppia (istruttore, giorno+ora) sia associata a più attività diverse. ⇒ Non soddisfa il vincolo che un istruttore non possa svolgere 2 lezioni contemporaneamente.

DataFine in tempo2 e come identificatore DataI, DataFine non va bene perché si lascia troppa libertà perché così in tempo2 per ogni data inizio posso avere tante datafine diverse, e per stessa tessera posso avere stesso pacchetto.

Modello logico-relazionale:

ISTRUTTORE (Codfiscale, Nome, Cognome, DataNascita, TitoloStudio, Email, TEL*)

PROGRAMMA (Giorno, OraI, Codfiscale, OraF, Sala, CodA)

ATTIVITA' (CodA, TipoA)

ATTIVITA' MUSICALE (CodA, Nome, Livello)

~~ATTREZZI (Attrezzo)~~

PACCHETTO (CodP, Costo, OraI, OraF, TipoP, TipoI*, Durata*, Categoria*)

TESSERATO (NumTesserca, Nome, Cognome, DataNascita, DataInizio, DataFine, Categoria)

CATEGORIA (Categoria)

~~TEMPO1 (Data)~~

~~TEMPO2 (DataInizio)~~

ABILITATO (Codfiscale, CodA)

USO_ATTREZZI (CodA, Attrezzo)

CONTIENE (CodA, CodP)

ACQUISTO (CodP, NumTesserca, DataInizio, DataFine)

ACCESO (NumTesserca, Data, OraI, OraF)

La tabella CATEGORIA non deve essere eliminata perché l'entità corrispondente partecipa opzionalmente nelle relazioni PER e DI.

Link di integrità:

PROGRAMMA (Codfiscale) REFERENCES ISTRUTTORE (Codfiscale)

PROGRAMMA (CodA) REFERENCES ATTIVITA' (CodA)

ATTIVITA' MUSICALE (CodA) REFERENCES ATTIVITA' (CodA) V

PACCHETTO (Categoria) REFERENCES CATEGORIA (Categoria)

TESSERATO (Categoria) REFERENCES CATEGORIA (Categoria)

ABILITATO (Codfiscale) REFERENCES ISTRUTTORE (Codfiscale)

ABILITATO (CodA) REFERENCES ATTIVITA' (CodA)

USO_ATTREZZI (CodA) REFERENCES ATTIVITA' MUSICALE (CodA) V

CONTIENE (CodA) REFERENCES ATTIVITA' (CodA)

CONTIENE (CodP) REFERENCES PACCHETTO (CodP)

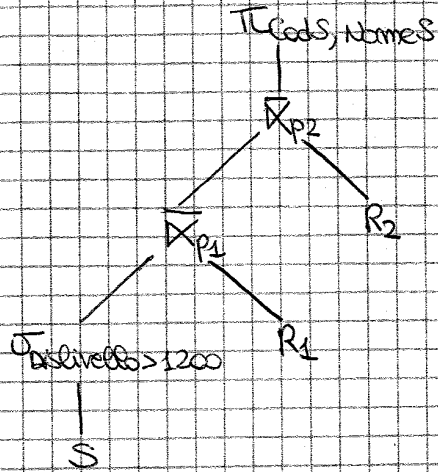
ACQUISTO (CodP) REFERENCES PACCHETTO (CodP)

ACQUISTO (NumTesserca) REFERENCES TESSERATO (NumTesserca)

ACCESO (NumTesserca) REFERENCES TESSERATO (NumTesserca)

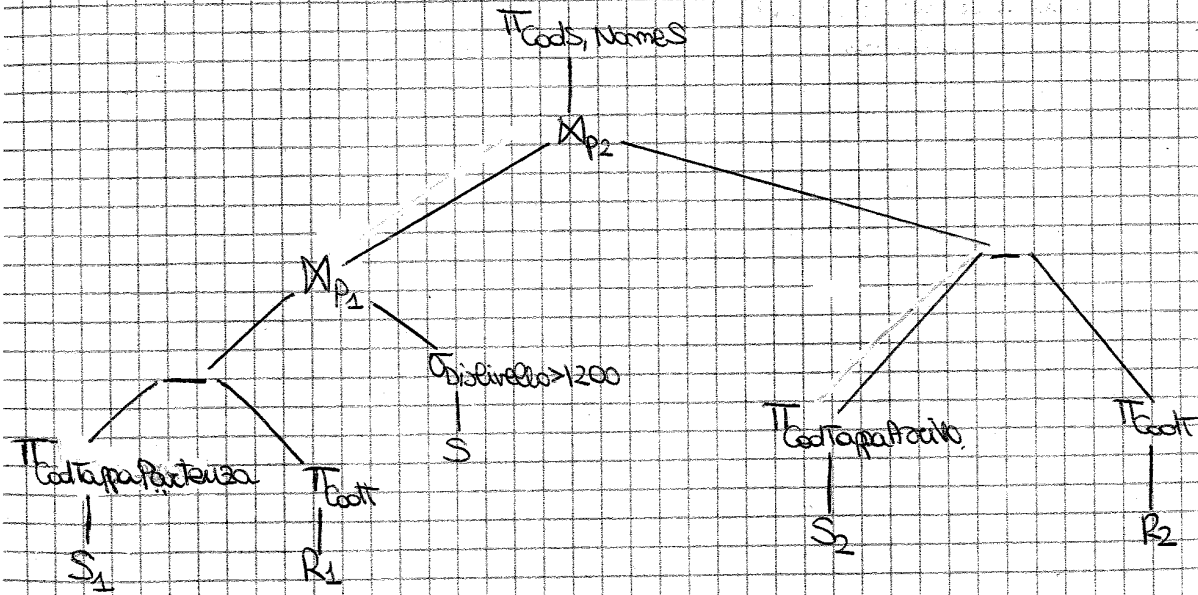
In algebra:

- Antigorlo (in questo caso è più semplice così)
- Differenza



$$P_1: S. \text{CodTappaPartenza} = R_1. \text{CodT}$$

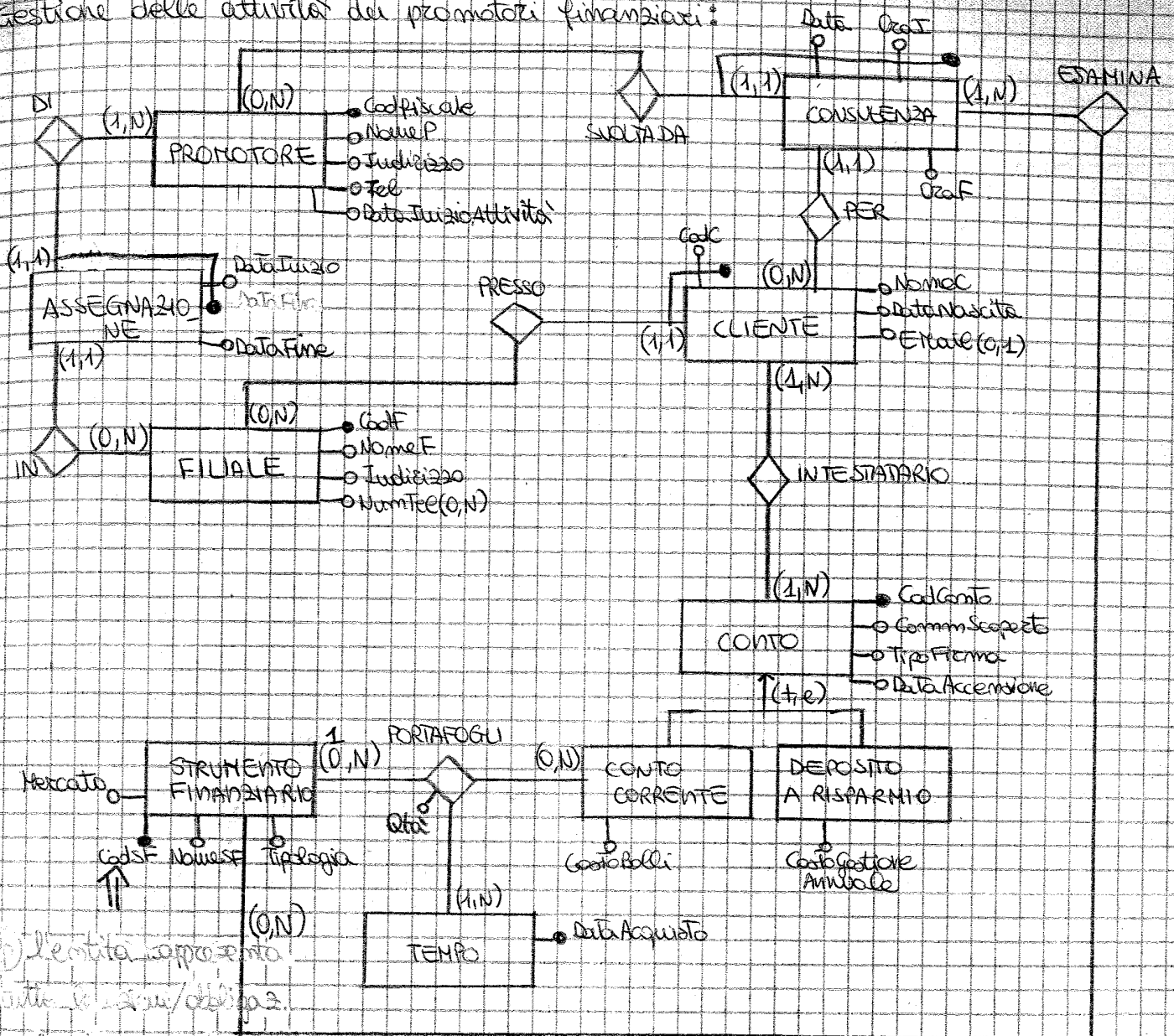
$$P_2: S. \text{CodTappaArrivo} = R_2. \text{CodT}$$



$$P_1: S_1. \text{CodTappaPartenza} = S. \text{CodTappaPartenza}$$

$$P_2: S_2. \text{CodTappaArrivo} = S. \text{CodTappaArrivo}$$

Gestione delle attività dei promotori finanziari:



(5) l'entità rappresenta tutte le azioni/obbligazioni
 tutte le azioni/obbligazioni
 con le stesse caratteristiche

(6) l'entità rappresenta tutte le azioni/obbligazioni con le stesse caratteristiche
 con una sola istanza (quindi non il numero di serie della singola azione/
 obbligazione).

SU ASSEGNAZIONE

(CodFiscale) REFERENCES PROIEZIONE (CodFiscale)

(CodF) REFERENCES FILIALE (CodF)

Tema d'esame 07 Febbraio 2011

↳ Sono date le seguenti relazioni (le chiavi primarie sono sottolineate):

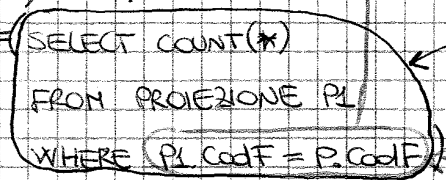
SALA-CINEMATOGRAFICA (CodSC, NomeSala, NomeCinema, NumPostiDisponibili)

FILM (CodF, Titolo, Regista)

PROIEZIONE (CodSC, Data, OraInizio, CodF, NumBigliettiVenduti)

```

c) SELECT S.CodSC, NomeSala
FROM SALA-CINEMATOGRAFICA S, PROIEZIONE P
WHERE S.CodSC = P.CodSC
GROUP BY S.CodSC, CodF, NomeSala
HAVING COUNT(*) =
    
```

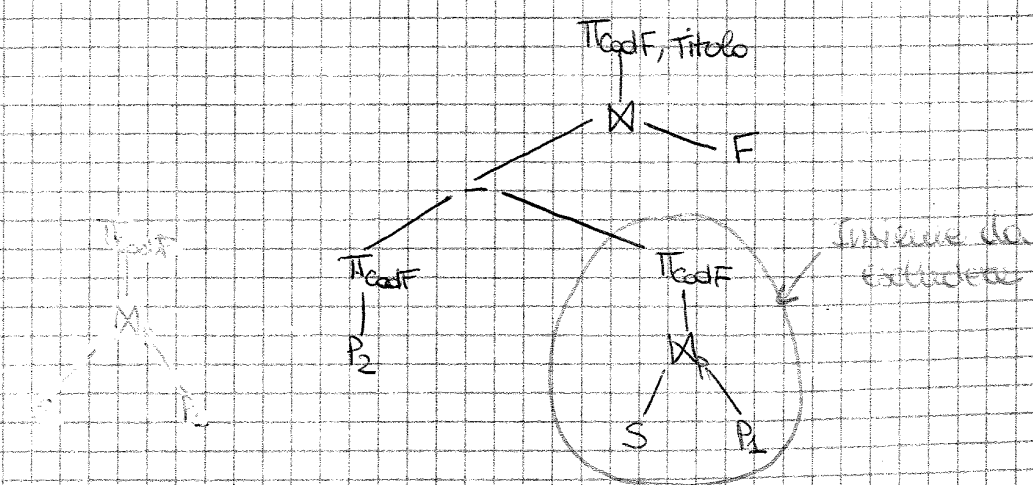


Numero Totale di proiezioni per un film
condizione di selezione

```

d) DISTINCT
SELECT F.CodF, Titolo, Regista
FROM FILM F, PROIEZIONE P
WHERE F.CodF = P.CodF AND Data >= 1/1/2011 AND Data <= 31/1/2011
GROUP BY F.CodF, Data, Titolo, Regista
HAVING COUNT (DISTINCT CodSC) > 10
    
```

↳ Insieme da escludere: film proiettati in sale cinematografiche con un numero di posti inferiore o = a 10 volte il numero dei biglietti venduti per la proiezione.



$$p: S.CodS = P1.CodS \text{ and } NumPostiDisponibili \leq 10 * NumBigliettiVenduti$$

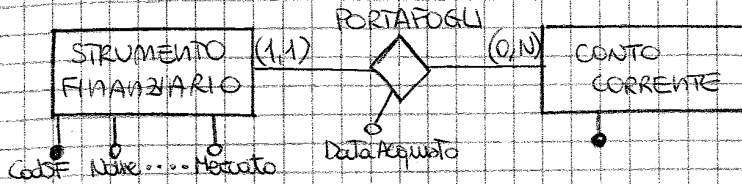
```
SELECT Categoria, M.CodM, NomeM
FROM MEDICO M, ACQUISTO A, FARMACO F
WHERE M.CodM = A.CodM AND F.CodF = A.CodF
GROUP BY M.CodM, Categoria, NomeM
HAVING SUM(Quantita) = (SELECT MAX(QTot)
FROM (SELECT CodM, Categoria, SUM(Quantita) AS QTot
FROM ACQUISTO A1, FARMACO F1
WHERE A1.CodF = F1.CodF
GROUP BY CodM, Categoria) AS T
WHERE T.Categoria = F.Categoria);
```

potrebbe essere omissa dalla soluzione

condizione di correlazione

- CodM può essere omissa dall'interrogazione, azione modificata (nella SELECT)

Alternativa per PORTAFOGLI:



⊕ Ogni CodSF rappresenta una singola azione/obbligazione

In questo caso

Nome SF → Categoria

e l'entità non è in forma normale di Boyce-Codd (BCNF)

CodSF rappresenta la singola azione o obbligazione.