NUMERO: 2139A- ANNO: 2017

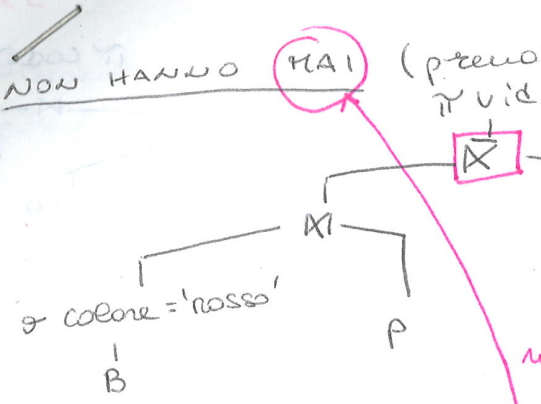
A P P U N T I

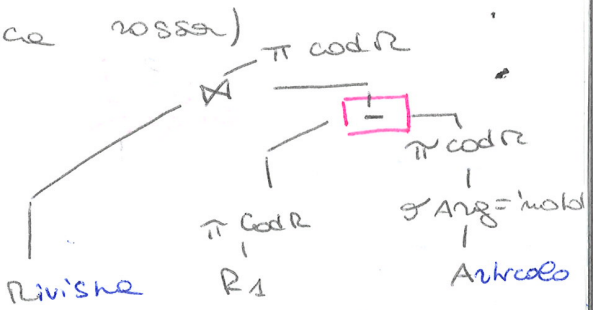
STUDENTE: Pinna Eleonora

MATERIA: Basi di dati - Prof. Farinetti

CONCETTI BASE ALGEBRA RELAZIONALE

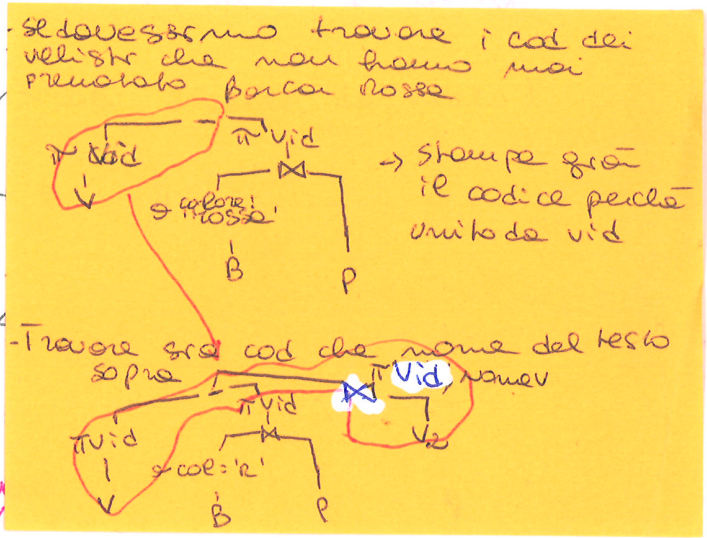
- NON HANNO MAI (prenotato una barca rossa)



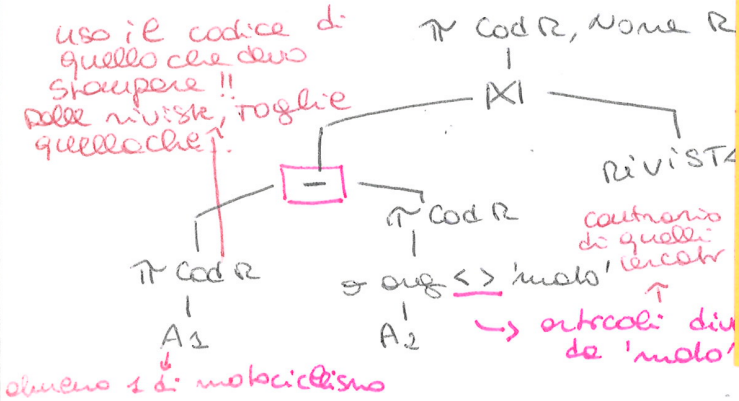


nota differenza della sottrazione

- HANNO PUBBLICATO SOLO (o

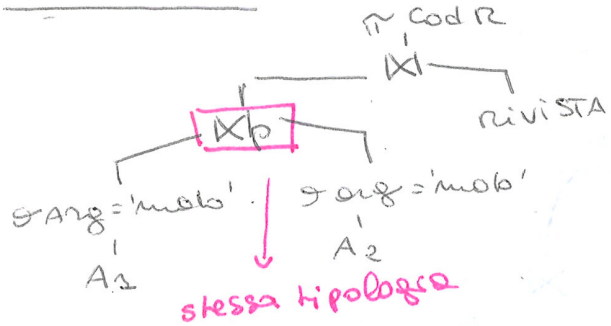


uso il codice di quello che devo stampare!! Dalle riviste, toglie quelloche?



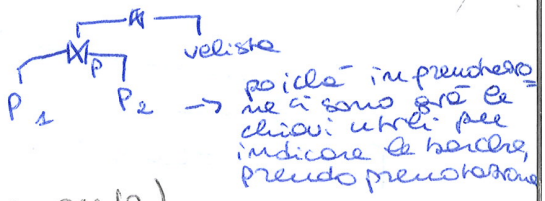
almeno 1 di motociclismo

- ALMENO 2 (articoli di moto) → almeno 3, faccio un doppio theta

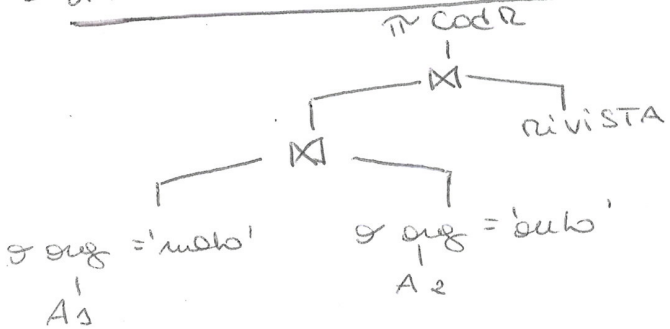


N.B. ragiona bene! Non creare ridondanze

→ Es: "almeno 2 barche ha prenotato"



- SIA DX UNO CHE DELL'ALTRO (sia di moto che auto)

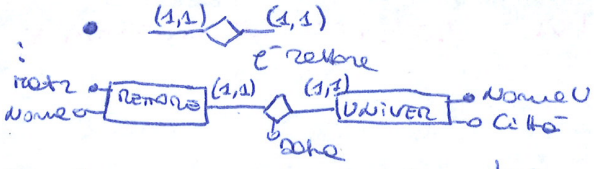


SQL.

SQL

- INIZIANO con la lettera A → COGNOME LIKE 'A%'
- ORDER BY nome ASC/DES
- ALMENO, PIÙ DI, MENO DI QUANTO, SUPERIORE → HAVING COUNT (uso solo quando devo contare qualcosa che non è già definita come numero nella del calcolo)
  ↓
  sempre con GROUP BY
- NON, NON HA MAI → NOT IN
- CON COGNOME DIA E con COGNOME PINNA → UNO NEL WHERE e UNO nell'IN
- STESSO DIPARTIMENTO, STESSO GG → alta probabilità di CONDIZIONE DI CORRELAZIONE
- ORE LEZIONE = 'oraFine - oraInizio)
- È SEMPRE STATA → faccio NOT IN con l'affermazione opposta per indicare che non è mai stata nel contrario
- PUNTEGGIO MAX ≠ PUNTEGGIO MAX POSSIBILE
  ↓
  somma dei punteggi possibili
- N.B. PER MAX → nella 3° SELECT, (DOVE HAVING COUNT E MAX(...) Devo inserire i nomi della group by, ovvero gli attributi che uso nella group by più esterna
- SE HO ALMENO 1, mettere having count non serve

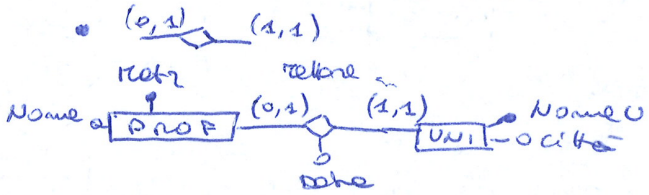
- RELAZIONE BINARIA UNO A UNO:



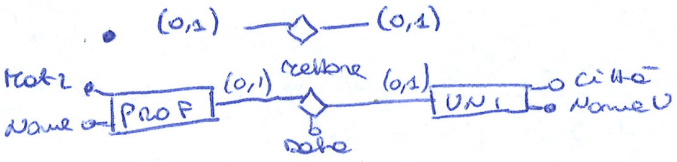
Attribuisco a una delle due entità la chiave p dell'altra messa come non primaria, e l'attribu della relazione.

A) RETTORE (matr, Nome, NomeU, Data)
UNIVER (NomeU, Città)

B) RETTORE (matr, Nome)
UNIVER (NomeU, Città, matr, Data)



PROF (matr, Nome)
UNI (NomeU, città, matr, Data)

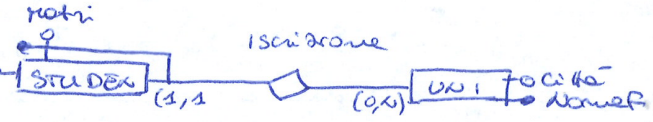


PROF (matr, Nome)
UNI (NomeU, città)

A) RETTORE (matr, NomeU, Data)
oppure
B) RETTORE (matr, NomeU, Data)

- oppure, scegliendo la coda che ha opzionalità più debole:
PROF (matr, Nome)
UNI (NomeU, Città, Matricola*, Data*)

- ENTITÀ CON IDENTIFICANTE ESTERNO:

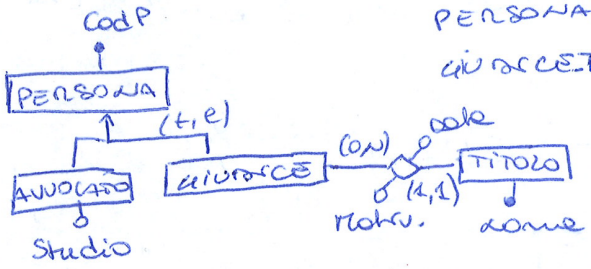


Attribuisco a quello con identificatore anche la chiave primaria dell'altro

STUDEN (matr, NomeF, Nome)
UNI (città, NomeF)

RUOLO CONSIDERATO NON ATTACCATO AL PADRE

(perché legato ad un'altra relazione

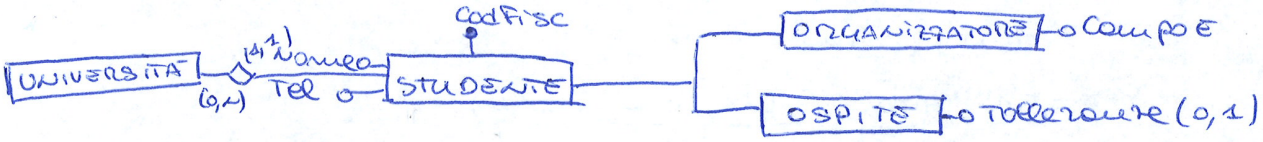


PERSONA(codP, tipo, Studio*)

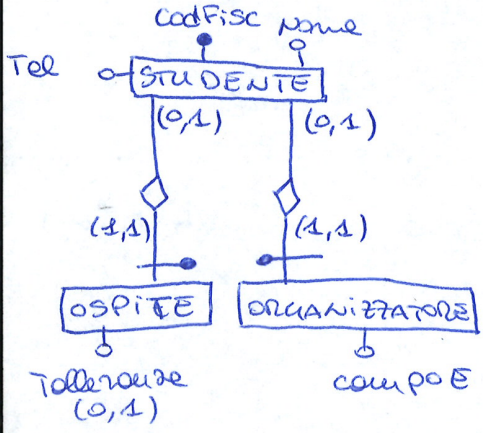
GIUDICETITOLO(codPGiudice, NomeTitolo, data, motivazione)

TITOLO(nome)

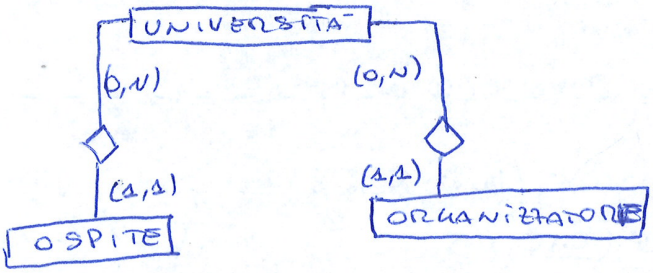
ACCORPAMENTO PADRE NELLE FIGLIE



1° METODO



2° METODO



- Non si può usare per le parziali o sovrapposte!

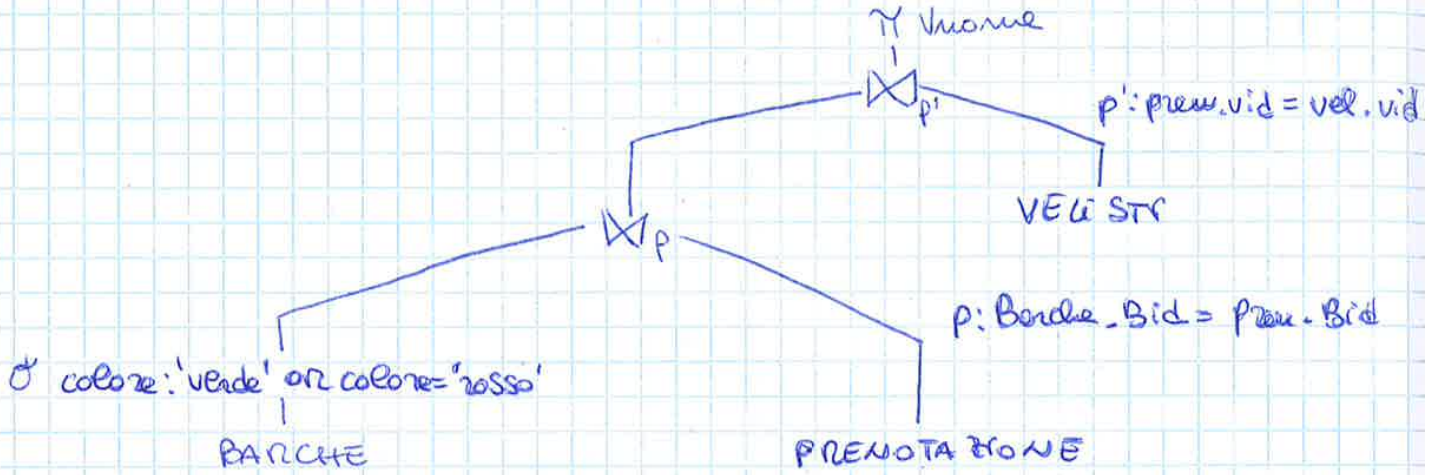
2) Sia dato lo schema relazionale costituito dalle tabelle (chiavi primarie sottolineate)

VELISTI (Vid, Vnome, Esperienza, DataNascita)

PRENOTAZIONI (Vid, Bid, Data) → Devo per forza inserire Data perché altrimenti potrei prenotare la stessa barca, con la stessa persona, una sola volta.

BARCHE (Bid, Bnome, Colore)

Trovare i nomi dei velisti che hanno prenotato almeno una barca rossa oppure una barca verde.



5) Sia dato lo schema relazionale costituito dalle tabelle (chiavi primarie sottolineate)

VELISTI (Vid, Vnome, Esperienza, DataNascita)

PRENOTAZIONI (Vid, Bid, Data)

BARCHE (Bid, Bnome, Colore)

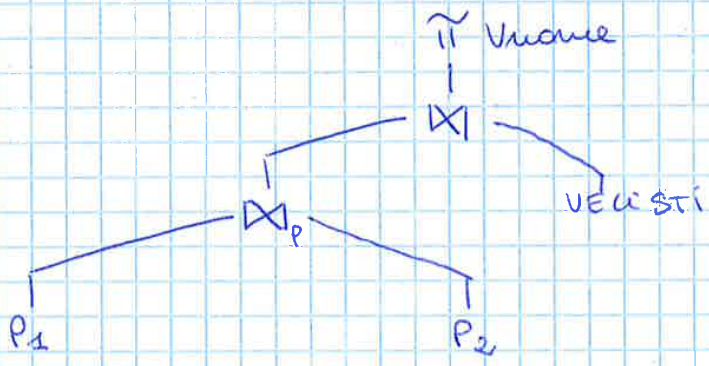
Trova i nomi dei velisti che hanno fatto almeno due prenotazioni.

Teoricamente:

PREN P_1			PREN P_2		
Vid	Bid	Data	Vid	Bid	Data

1° condizione: $(P_1.Vid = P_2.Vid)$ AND $(P_1.Data \neq P_2.Data)$ $(P_1.Bid \neq P_2.Bid)$

- basta che una di queste due sia vera
- uso OR

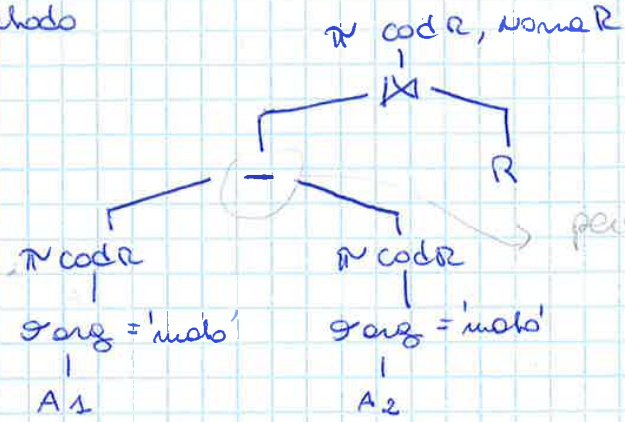


$P: (P_1.Vid = P_2.Vid)$ AND $(P_1.Data \neq P_2.Data)$ OR $(P_1.Bid \neq P_2.Bid)$

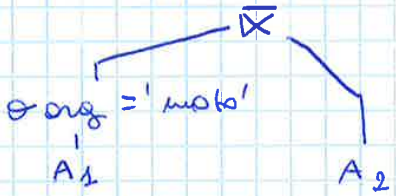
P_1, P_2 = due diverse prenotazioni

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

1° metodo



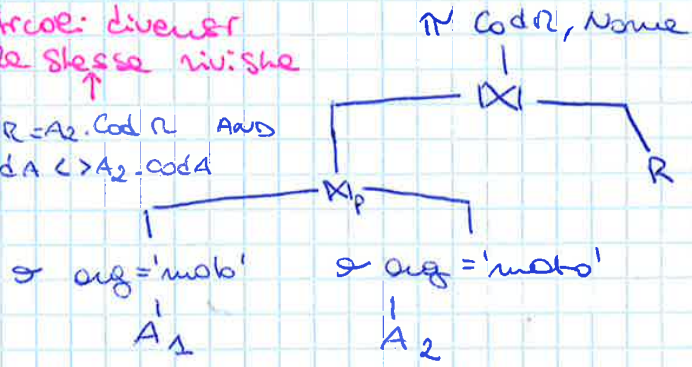
2° metodo:



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

articoli diversi nelle stesse riviste

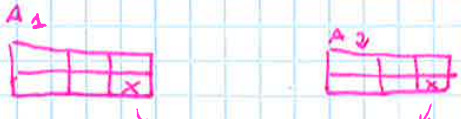
P: A1.CodR = A2.CodR AND A1.CodA <> A2.CodA



P: R.codR = A1.CodR

↓

Scrivere A1 o A2 è lo stesso perché li lo gro equoge:ar con Ap



cod A è diverso nelle riviste

N.B.: se avesse avuto "3 articoli: auto" avrei aggiunto anche A3

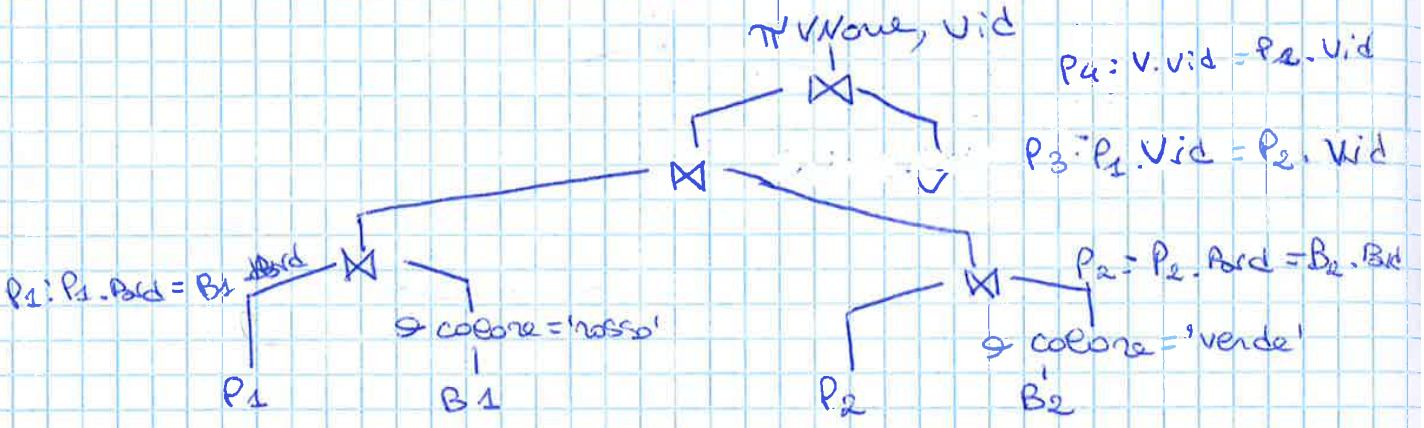
a) Dato lo schema relazionale costituito dalle tabelle

VELISTI (Vid, VNome, DataNascita, Esperienza)

PRENOTAZIONI (Vid, Bid, Data)

BARCHE (Bid, BNome, Colore)

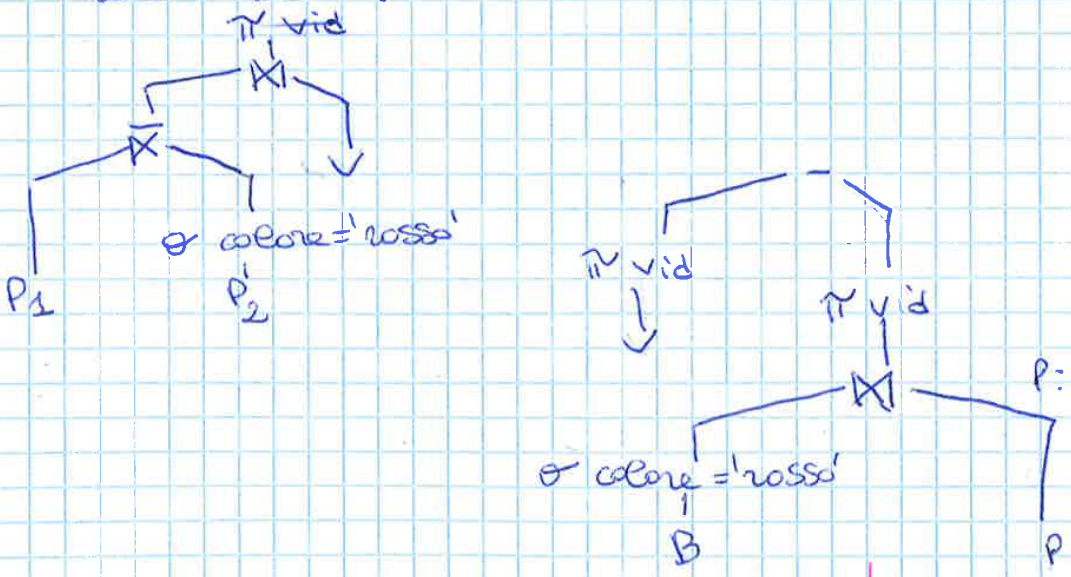
b) Trovare i nomi dei velisti che hanno prenotato una barca rossa e una verde.



devo farlo separato perché è come se avessi un AND, infatti voglio due prenotazioni di diverso tipo, ma le voglio entrambe.

Bid	Vid	Data	colore
x	y		rosso
x	z		verde

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



Codici che hanno prenotato almeno una barca rossa e ci sottraggo ai velisti

3) Dato lo schema relazionale costituito dalle tabelle

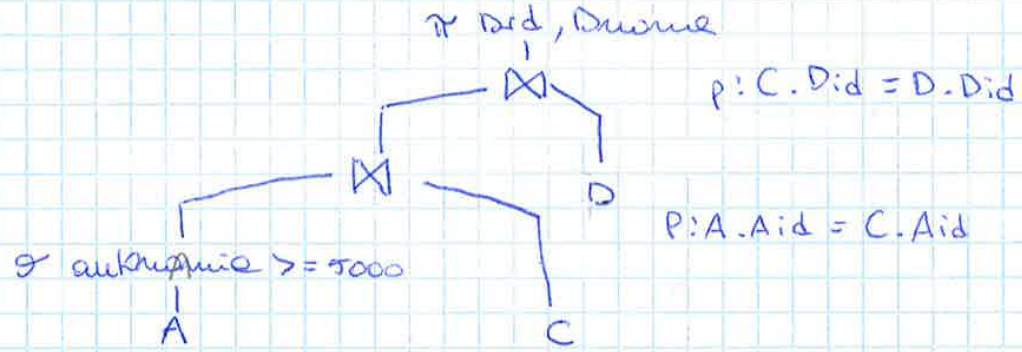
AEREO (Aid, ANome, Autonomia)

CERTIFICATO (Did, Aid)

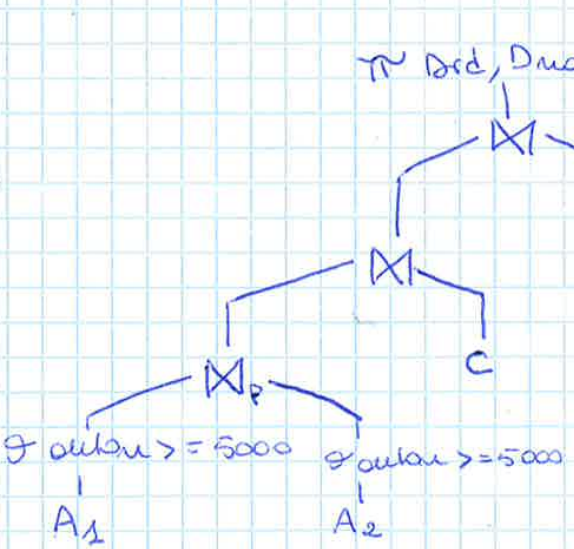
DIPENDENTE (Did, DNome, Stipendio)

esprimere in algebra relazionale le seguenti interrogazioni

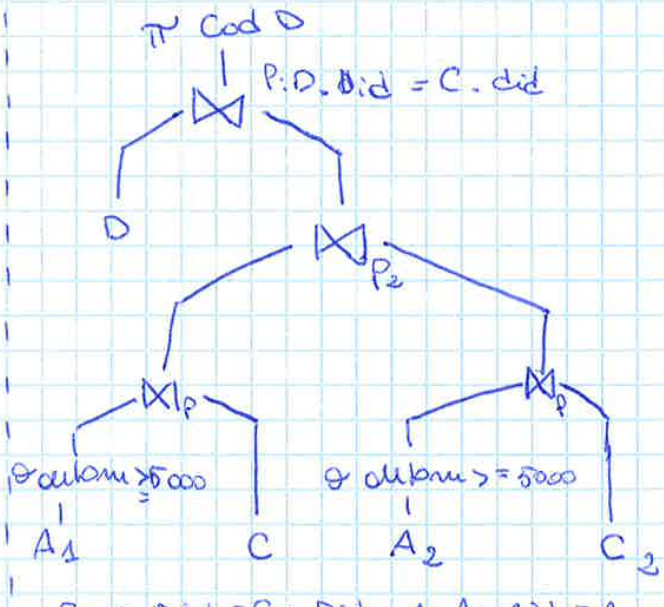
a) Trovare i codici e i nomi dei dipendenti abilitati al volo su un aereo in grado di coprire distanze superiori a 5000 km (autonomia ≥ 5000)



b) Trovare i codici e i nomi dei dipendenti abilitati al volo su almeno due aerei in grado di coprire distanze superiori a 5000 km.



Stesso dipendente ma diverso aereo.



P_2: C.Did = C_2.Did ∧ A_2.Aid = A_2.Aid

ES. 8-9)

Sono date le seguenti relazioni:

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

ATTORI (CodAttore, Nome, AnnoNascita, Nazionalità)

RECITA (CodAttore, CodFilm)

PROIEZIONI (CodProiezione, CodFilm, CodSala, incasso, DataProiezione)

SALE (CodSala, Posti, Nome, Città)

a) Trovare per ogni regista l'incasso totale di tutte le proiezioni dei suoi film.

```
SELECT   F.regista, SUM(P.incasso) AS incasso_tot
FROM     FILM AS F, PROIEZIONI AS P
WHERE    F.CodFilm = P.CodFilm
GROUP BY F.regista
```

→ perché ho fatto per ogni regista

regista	inc. tot

b) Per ogni film S. Spielberg, trova titolo, n° proiezioni a Torino e incasso totale.

```
SELECT   F.titolo, COUNT(*) AS num_proiezioni,
         SUM(incasso) AS incasso_tot
FROM     FILM AS F, PROIEZIONI AS P, SALE AS S
WHERE    F.CodFilm = P.CodFilm AND P.CodSala = S.CodSala
         AND S.Città = 'Torino' AND S.regista = 'S. Spielberg'
GROUP BY F.Film, F.Titolo
```

non inserire aggregati qui

→ condizioni di join per selezionare righe che ci interessano.

Ho dovuto mettere per forza se lo inserisco nella select per essere visualizzato. Non crea differenza in group by perché tanto ho già F.codFilm

HAVING: non inserire attributi

TEMA D'ESAME

AUTOMOBILE (Targa, Modello, Marca, Costo giornaliero)

CLIENTE (CodCliente, NomeCliente, NumPatente, DataNascita, Nazionalità).

CONTRATTO_AFFITTO_AUTO (Targa, DataInizio, DataFine, CodCliente, SedePartenza, SedeArrivo)

(5 punti)

Considerando i contratti d'affitto iniziati nel primo semestre del 2016, per ogni modello per cui sono stati stipulati almeno 20 contratti, visualizzare il modello, la marca, l'importo totale incassato (prodotto tra costo gg · periodo) la durata complessiva, media, max dei contratti d'affitto.

condizione di h va nel where

```
SELECT  A.Modello, A.Marca, SUM((DataFine - DataInizio) * Costo
        SUM(DataFine - DataInizio), AVG(DataFine - DataInizio
        MAX(DataFine - DataInizio)
FROM    Automobile AS A, CONTRATTO_AFFITTO_AUTO AS C
WHERE   A.Targa = C.Targa AND (DataInizio >= 1/1/2016)
        AND (DataFine <= 30/06/2016)
GROUP BY A.Modello, A.Marca
HAVING COUNT(*) >= 20
```

perché sono ini nel primo semes ma non per for finite.

N.B.: Almeno 20 contratti = non è di tupla perché non ho u voce che dice i n° dei contratti, quindi non lo devo mettere nel wh

Contenuto di impiegato, turno...

IMPIEGATO (CodImp, Cognome, Nome, CodDep, Stipendio)

DIPARTIMENTO (CodDep, Denominazione, Città)

TURNO (CodImp, Data, OraI, OraF)

c) Selezionare tutti i dati dei turni degli impiegati che hanno uno stipendio maggiore di 10'000 e che lavorano in dipartimenti di Roma.

```
SELECT   CodImp, Data, OraI, OraF
FROM     TURNO AS T, IMPIEGATO AS I, DIPARTIMENTO AS D
WHERE    T.CodImp = I.CodImp AND D.CodDep = I.CodDep
         AND Stipendio > 10'000 AND Città = 'Roma'
```

d) Trovare il valore max degli stipendi erogati dai dipartimenti di Milano.

```
SELECT   D.CodDep, D.Denominazione, MAX(Stipendio)
FROM     DIPARTIMENTO AS D, IMPIEGATO AS I
WHERE    D.CodDep = I.CodDep   AND   Città = 'Milano"
GROUP BY I.CodDep
```

e) Selezionare i codici dei dipartimenti che spendono più di 10 milioni in stipendi

```
SELECT   I.CodDep
FROM     IMPIEGATO I
WHERE
GROUP BY I.CodDep
HAVING   SUM(I.Stipendio) > 10'000'000
```

Avrei potuto anche inserire la tabella Dipartimento e fare il join tra le 2 in where, ma sarebbe stata un'informazione ridondante perché i dati necessari sono già tutti contenuti nella voce impiegato.

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 25 di 64

Dato lo schema relazionale costituito

RIVISTA (CodR, NomeR, Editore)

ARTICOLO (CodA, Titolo, Argomento, CodR)

b) Trovare il codice e il nome delle riviste che non hanno mai pubblicato articoli di argomento 'motociclismo'.

```
SELECT CodR, NomeR
FROM   RIVISTA R
WHERE  R.CodR NOT IN (SELECT CodA
                      FROM   ARTICOLO A
                      WHERE  Argomento = 'moto')
```

c) Trovare il nome delle riviste che hanno pubblicato solo articoli di argomento 'motociclismo'

```
SELECT NomeR
FROM   RIVISTA R, ARTICOLI A          (giusto?)
WHERE  R.CodR = A.CodR
   AND Argomento = 'moto'
```

Fede lui fatto con il not in e poi arg. ≠ moto

d) Trovare il nome e codice delle riviste che pubblicano articoli motociclismo oppure auto.

```
SELECT NomeR, CodR
FROM   RIVISTA R, ARTICOLO A
WHERE  R.CodR = A.CodR
   AND (Argomento = 'moto' OR argomento = 'auto')
```

↓

parentesi fondamentali, altrimenti l'AND ha la precedenza su OR

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

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

g) Trovare il codice e il nome delle riviste che hanno pubblicato un solo articolo di motociclismo.

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

TEMA D'ESAME

ATLETA (CodA, NomeAtleta, NazioneAtleta)

GARA_NUOTO (CodG, Data, Disciplina, LuogoGara, NazioneGara)

CLASSIFICA (CodG, CodA, PosizioneArrivo, Tempo)

Visualizza codice e nome atleta che hanno partecipato a tutte le gare della disciplina "100 m dorso" che sono state disputate in Italia, classificandosi sempre nelle prime 10 posizioni.

```
SELECT  CodA, NomeAtleta
FROM    ATLETA A, CLASSIFICA C
WHERE   A.CodA=C.CodA  AND NazioneGara = 'Italia'
        AND Disciplina = '100 m dorso'
        AND PosizioneArrivo <= 10
GROUP BY CodA, NomeAtleta
HAVING COUNT(*) = (SELECT  COUNT(*)
                   FROM    GARA_NUOTO
                   WHERE   Disciplina = '100 m dorso'
                           AND NazioneGara = 'Italia')
```

→ e↓ meglio perché a lei e stessi non vedono quelli interni!

→ conto tutte gare in Italia

↓

Io metto perché mi permette di avere TUTTE le gare con queste caratteristiche.
Se non lo mettessi non è avrei esattamente tutte

TEMA ESAME

1)

HOTEL (CodHotel, NomeHotel, Indirizzo, Città, Stelle)

RECENSIONI_HOTEL (CodRec, CodHotel, DataRec, VotoRec)

Selezionare nome e città degli hotel che hanno ricevuto almeno 1 recensione con voto > 8 e non hanno ricevuto recensioni nel 2013.

```
SELECT   NomeHotel, città
FROM     HOTEL H, RECENSIONI_HOTEL R
WHERE    H.CodHotel = R.CodHotel
         AND   VotoRec > 8
         AND   codrec NOT IN (SELECT DataRec
                              FROM R
                              WHERE Data > '31/12/2012'
                                    AND DATA < '01/01/2014')
```

2)

FILM (codfilm, Titolo, AnnoProduzione, Nazionalità, Regista, Genere)

PROIEZIONI (CodProiezioni, Codfilm, CodSala, Incasso, Data_proiezione)

SALE (CodSala, Posti, Nome, Città)

Trovare i titoli dei film che non sono mai stati proiettati a Torino.

```
SELECT   Titolo
FROM     Film F
WHERE    NOT EXIST (SELECT *
                    FROM SALE S, PROIEZIONI P
                    WHERE Città = 'TORINO'
                          AND P.codfilm = F.codfilm)
```

N.B. con il NOT EXIST non devo mettere davanti nessun predicato, nella select metto l'*. Ricorda condizione di CORRELAZIONE.

Esercizio risolvibile anche con il NOT IN.

condizione di correlazione

2) n° 2

CORSO (cod corso, NomeC, Anno, Semestre)

ORARIO-LEZIONI (CodCorso, Giorno Settimana, Ora inizio, Orafine, Aula)

Trovare codice corso, nome corso e n° tot di ore di lezione settimanali per i corsi del terzo anno per cui il numero complessivo di ore di lezione settimanali è superiore a 10 e le lezioni sono in più di tre giorni diversi della settimana.

```
SELECT  CodCorso, NomeC, SUM(orafine - ora inizio)
FROM    CORSO C, ORARIO_LEZIONI O
WHERE   C.CodCorso = O.CodCorso
        AND Anno = '3'
GROUP BY  CodCorso, NomeC
HAVING COUNT(DISTINCT GIORNISETTIMANA) > 3
       AND SUM(ORAFINE - ORAINIZIO) > 10
```

perché non mi stampi due doppi

RICORDA: indica che le ore settimanali sono più di 10.

3) n° 3

ALLOGGIO (codA, Indirizzo, Città, Superficie, CostoAffittoMensile)

CONTRATTO_AFFITTO (codC, DataInizio, Datafine, NomePersona, codA)

Trovare per le città in cui sono stati stipulati almeno 100 contratti, la città il costo mensile massimo degli affitti, il costo mensile medio, la durata massima dei contratti, la durata media e il numero di contratti stipulati totale.

```
SELECT  città, MAX(costo affitto mensile), AVG(costo affitto mensile),
        MAX(Datafine - Datainizio), AVG(Datafine - Datainizio)
        COUNT(*)]
FROM    ALLOGGIO A, CONTRATTO_AFFITTO C
WHERE   A.codA = C.codA
GROUP BY  codC, città
HAVING COUNT(*) >= 100
```

RICORDA! se nella select devo visualizzare il numero totale scrivo il count(*).

2) GARA (CodG, Luogo, Data, Disciplina)
ATLETA (CodA, Nome, Nazione, DataNascita)
PARTECIPAZIONE (CodG, CodA, PosizioneArrivo, Tempo)

a) Trovare il nome e la data di nascita degli atleti italiani che non hanno partecipato a nessuna gara di discesa libera.

```
SELECT   Nome, DataNascita
FROM     ATLETA A, G
WHERE    Nazione = 'Italia'
         AND CodA NOT IN ( SELECT   CodA
                           FROM     GARA G, PARTECIP. P
                           WHERE    G.CodG = P.CodG
                           AND      Disciplina = 'discesa')
```

b) Trovare le nazioni per cui concorrono almeno 5 atleti nati prima del 1980, ciascuno dei quali abbia partecipato ad almeno 10 gare di sci di fondo.

```
SELECT   Nazione
FROM     ATLETA A
WHERE    DataNascita < '1/1/1980'
         AND CodA IN (SELECT   CodA
                      FROM     GARA G, PARTECIP. P
                      WHERE    Disciplina = 'Fondo'
                               AND G.CodG = P.CodG
                      GROUP By CodG
                      HAVING COUNT (*) >= 10)
GROUP By CodA
HAVING COUNT(*) >= 5
```

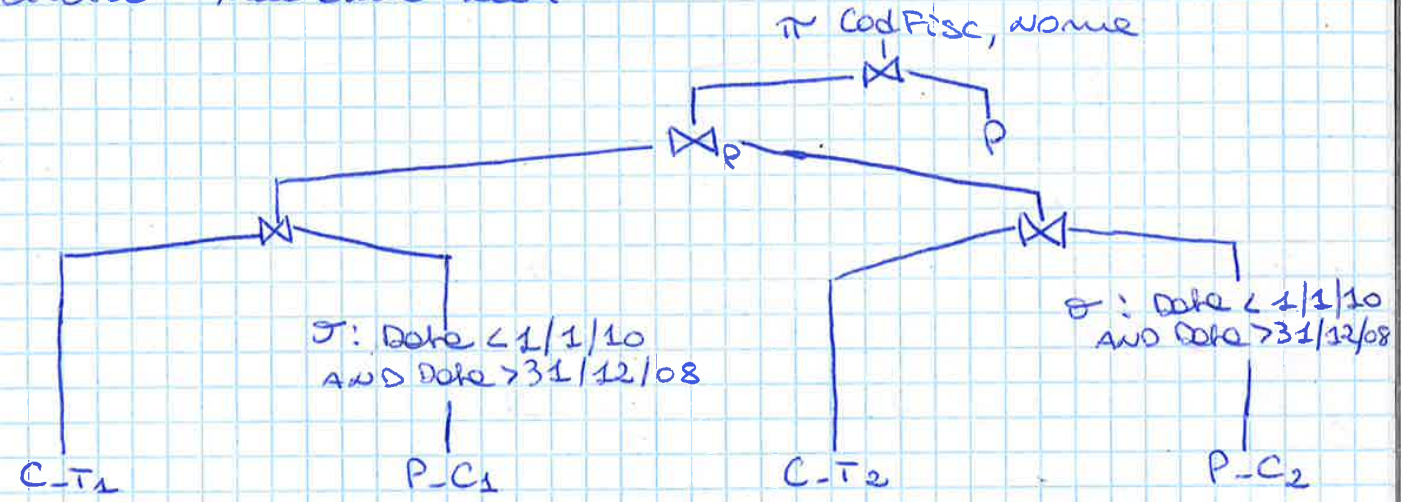
ESAME 5-2-10 parte A

PERSONA (CodFisc, Nome, DataNascita, Sesso)

CIRCOLO_TENNIS (CodC, NomeC, Indirizzo, Citta)

PRENOTAZIONE_CAMPO (CodC, Data, Ora, Campo, CodFisc)

a) Visualizzare il codice e il nome delle persone che hanno prenotato campi da tennis in almeno due città diverse nell'anno 2009



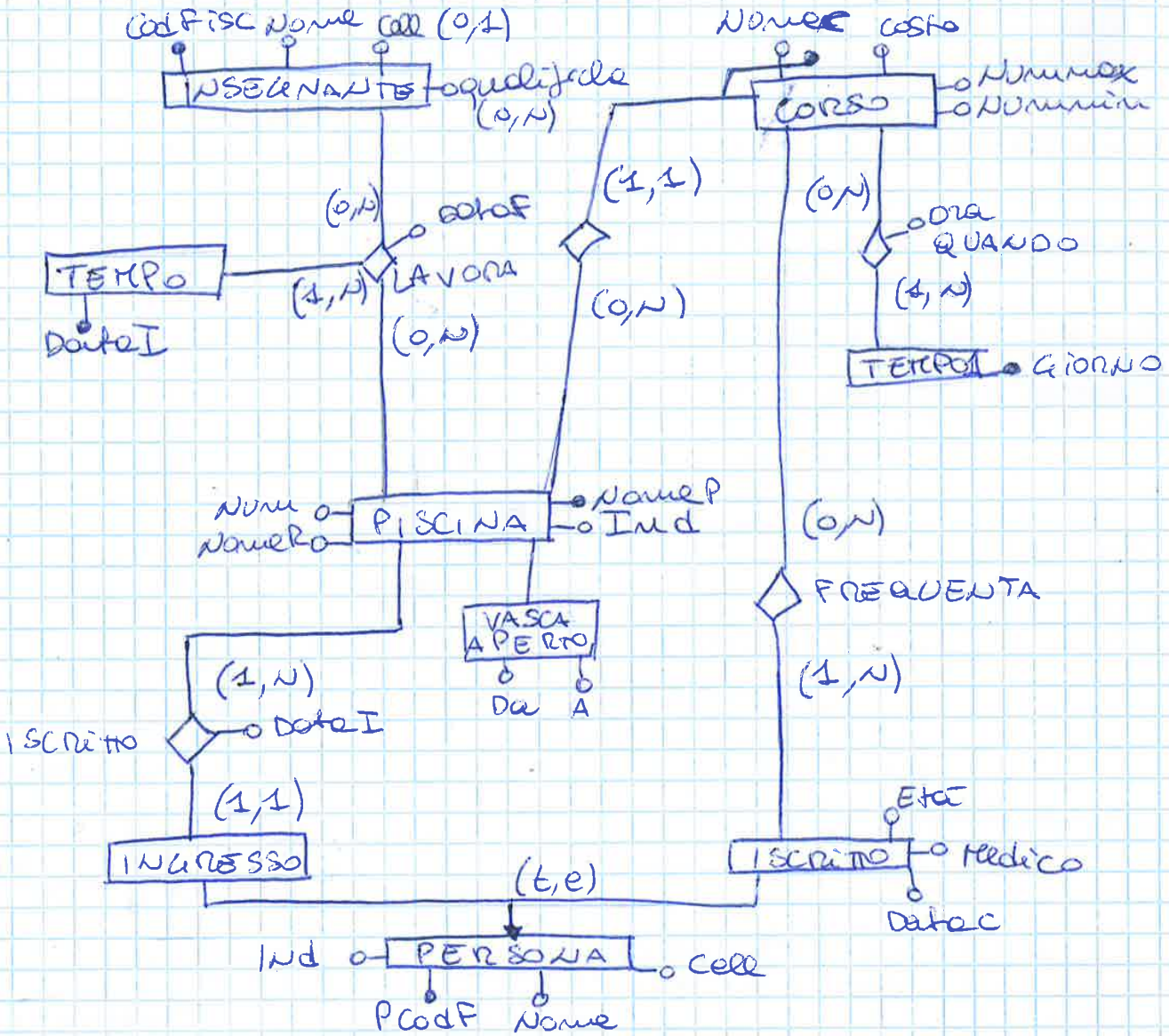
ρ: P_C1.CodFisc = P_C1.CodFisc AND C_T1.Citta <> C_T2.Citta

b) Visualizzare il codice, il nome e la città dei circoli in cui hanno prenotato solo persone nate dopo il 1970

```
SELECT   CodC, NomeC, Citta
FROM     CIRCOLO_TENNIS C_T
WHERE    CodC NOT IN (SELECT CodC
                      FROM PRENOTAZIONE_CAMPO PC, PERSONA P
                      WHERE PC.CodFisc = P.CodFisc
                      AND DATANASCITA 1970)
```

si può fare anche con "In" e " < di 1970 " (credo)

GESTIONE DELLE PISCINE COMUNALI DI TORINO



SCHEMA LOGICO - RELAZIONALE

INSEGNANTE (codFisc, Nome, ~~qualifiche~~, cell#)

TEMPO (DataI)

QUALIFICHE (CodFisc, Qualifica) ← Per elenco (0,N)

PISCINA (NomeP, Num, Ind, NomeR, Tipo, Da#, A#)

INGRESSO (PCodF, Ingresso, DataI) ← attributo a ingresso

PERSONA (pCodF, Nome, Ind, Cell)

ISCRITTO (PCodF, ~~Data~~Iscritto, Medico, età, DataC)

FREQUENTA (PCodF, Iscritto, NomeC, NomeP) → inserito per colon. (0,N) – (0,N)

CORSO (NomeC, NomeP, Costo, NumMax, NumMin)

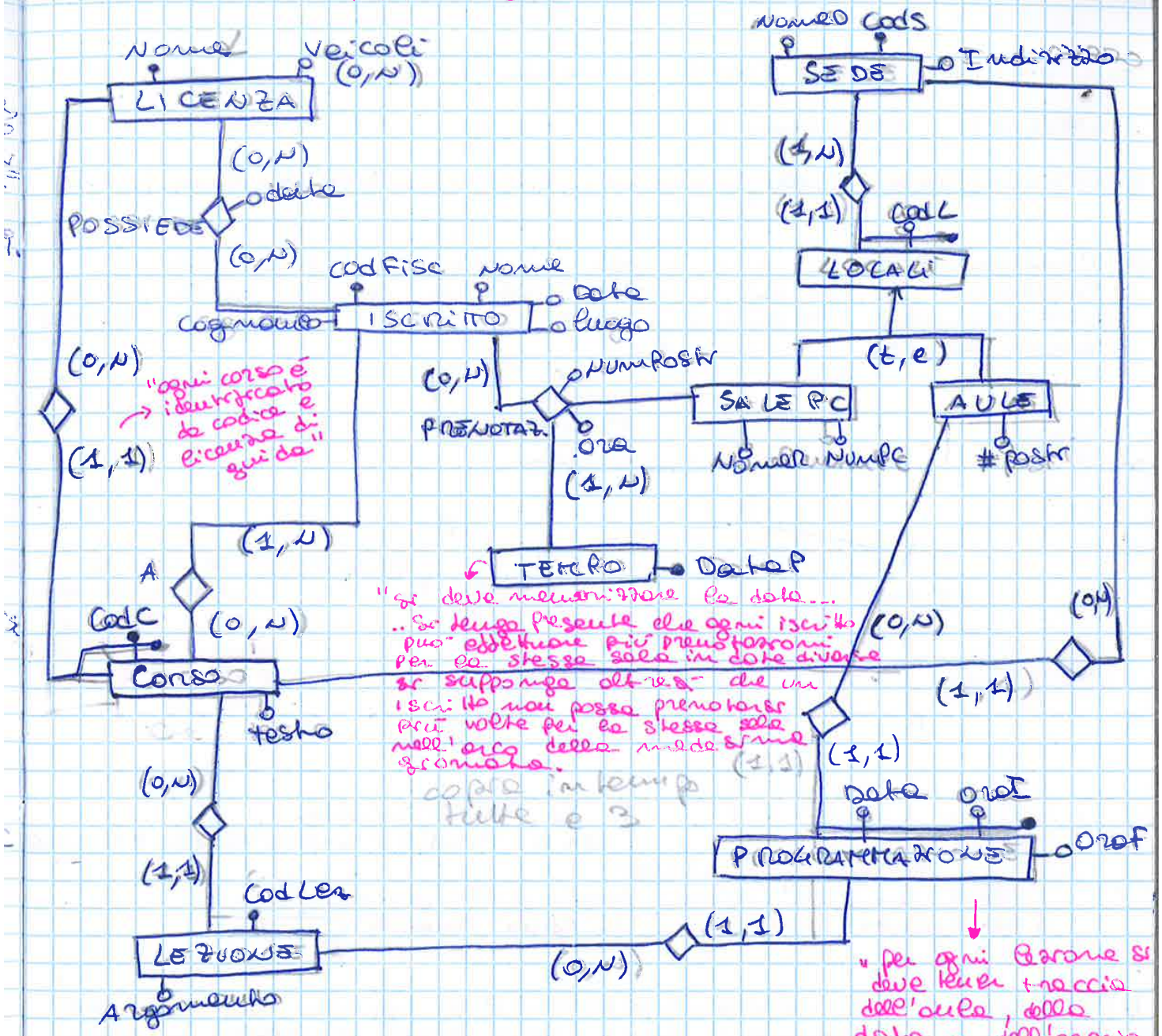
QUANDO (NomeC, NomeP, Giorno, Ora)

TEMPO1 (Giorno)

LAVORA (CodFisc, DataI, NomeP, DataF)

Mantengo le relazioni doppie

PARTE B



SCHEMA LOGICO-RELAZIONALE

LICENZA(NomeL)

VEICOLI(Nome, Veicolo)

ISCRITTO(CodFisc, Nome, Data, Luogo, Cognome)

A (CodFisc, CodC, Nome)

CORSO(CodC, NomeL, testo, CodS)

LEZIONE(codLez, Argomento, CodC, NomeL)

PROGRAMMAZIONE(Data, OraI, OraF, CodLez, CodS, CodL)

SEDE(CodS, NomeD, Indirizzo)

LOCALI(CodS, CodL, tipo, NomeR*, NumPC*, #PosM*)

PRENOTAZ(CodFisc, CodL, CodS, DataP, Ora, NumPosM)

TEMPO(DataP)

c) Visualizzare il codice e il nome delle sale in cui sono state tenute tutte le proiezioni di (almeno) un film.

```
SELECT   CodSC, NomeSala
FROM     SALACINEMATOGRAFICA S, FILM F, PROIEZIONE P
WHERE    S.CodSC = P.CodSC  AND  P.CodF = F.CodF
GROUP BY P.codF, CodSC, NomeSala
HAVING COUNT(*) = (SELECT  COUNT(*)
                   FROM    PROIEZIONE P1
                   WHERE   P1.CodF = P.CodF)
```

n° totale di proiezioni per un film

condizione di correlazione

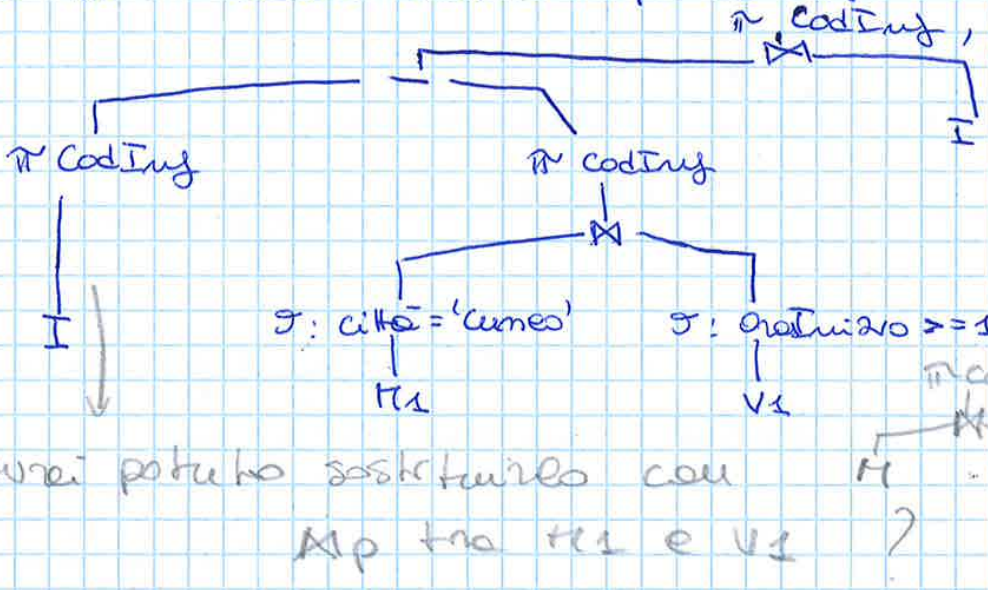
TEMA ESAME SVOLTO IN AULA

MEDICO (CodM, NomeM, Città)

INFORMATORE_FARMACEUTICO (codInf, NomeInf, AziendaFarm)

VISITA (CodInf, Data, OraInizio, DurataMinuti, CodM, TipoVisita)

a) Visualizzare codice e nome degli informatori farmaceutici che non hanno mai visitato medici di Cuneo con inizio della visita dopo le 16



avrei potuto sostituirlo con e fare

Mp tra M1 e V1 ?

b) Per i medici che, negli ultimi 999 di gennaio 2016 hanno ricevuto visite per la "prescrizione di farmaci di tipo pediatrico" per una durata complessiva inferiore a 90 minuti, visualizzare codice e nome del medico, numero tot, durata complessiva e media delle visite per la prescrizione di farmaci di tipo pediatrico.

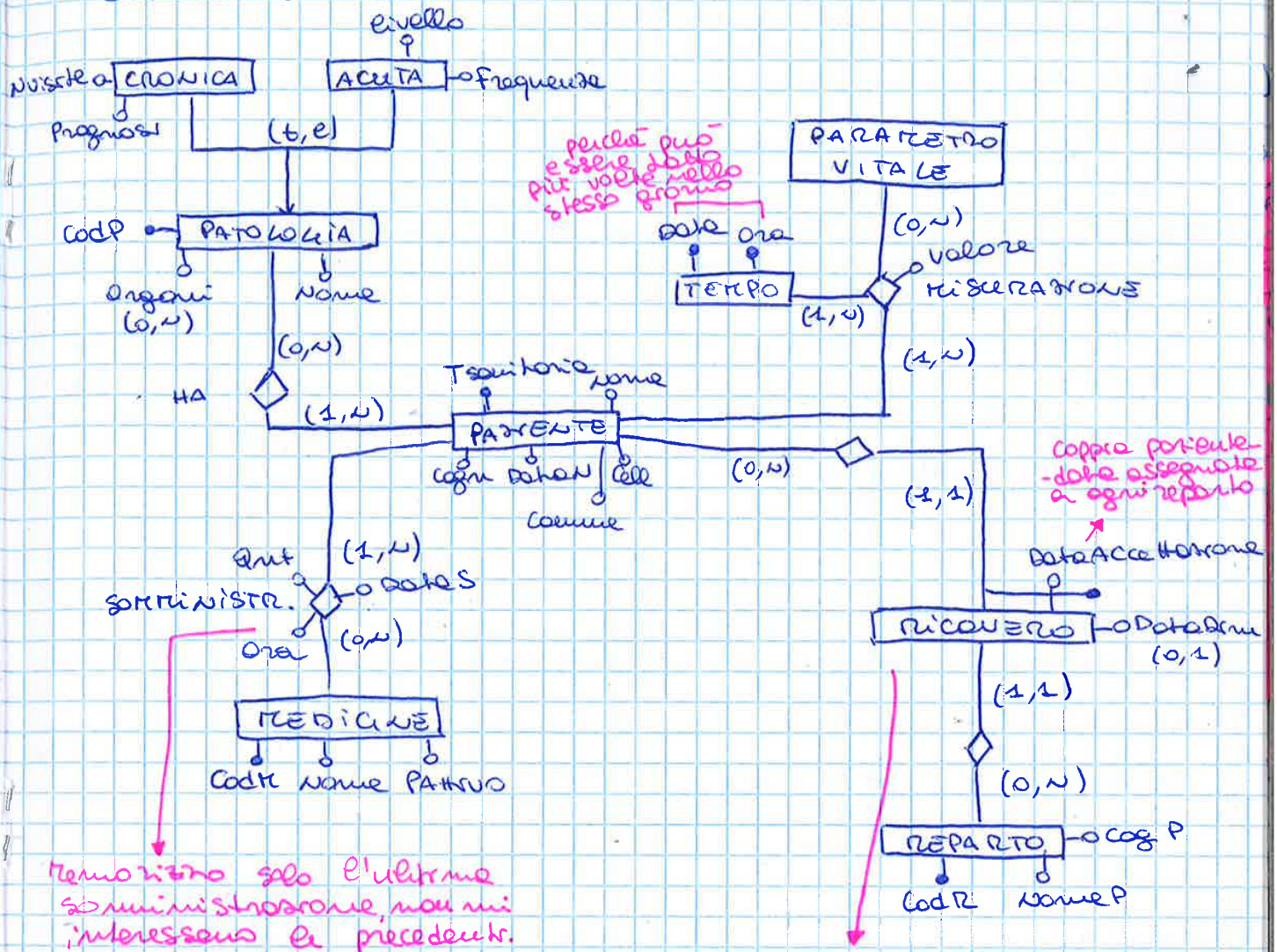
```
SELECT  CodM, NomeM, Count(*), SUM(DurataMinuti),
        AVG(DurataMinuti)
FROM    VISITA V, MEDICO M
WHERE   V.CodM, M.CodM  AND  DATA > 20/1/16  AND
        DATA < 31/1/16  AND  Tipo visita = 'Pediatrica'
GROUP BY CodM, NomeM
HAVING SUM(DurataMinuti) <= 90
```

può anche essere messo in IN

↓ se uguale se metto IN NOT IN con valori opposti

N.B. si intende durata complessiva di tutte le visite.

SCHEMA E-R



Memorizzo solo l'ultima somministrazione, non mi interessano le precedenti.

↓

per questo le scrivo nella relazione.

ENTITÀ STORICIZZATA!

- Ogni parente può andare in ogni reparto, ma in ricoveri diversi
- se avessi anche che in più ricoveri si può coprire in più reparti avrei usato la memoria

c) Per ogni regione, visualizzare codice, nome, quota e regione delle tappe che sono punto di arrivo del numero massimo di sentieri

```
SELECT  P.CodTappa, Regione, Quota, NomeT
FROM    PUNTO_TAPPA P, SENTIERO S        → inserito anche qui per condizione di connessione
WHERE   P.CodTappa = S.CodTappaArrivo
GROUP BY  P.CodTappa, Regione, Quota, NomeT        → per ogni regione
HAVING COUNT(*) = (SELECT MAX(tot sentieri)        (perché ho molte?)
                   FROM PUNTO_TAPPA P1,
                   (SELECT CodTappaArrivo, COUNT(*) AS tot_arrivi
                    FROM SENTIERO
                    GROUP BY CodTappaArrivo) AS T
                   WHERE P1.CodTappaArrivo = T.CodTappaArrivo,
                   AND P1.Regione = P.Regione)
```

Tabella temporanea per il calcolo del n° di sentieri che arrivano ad ogni tappa

Calcola il n° max di sentieri in arrivo in una tappa per una specifica regione

la regione è definita dalla condizione di connessione

nel where ci sono due join perché nel From ho la voce PUNTO_TAPPA e la select

Rivedere

ESERCIZI VARI

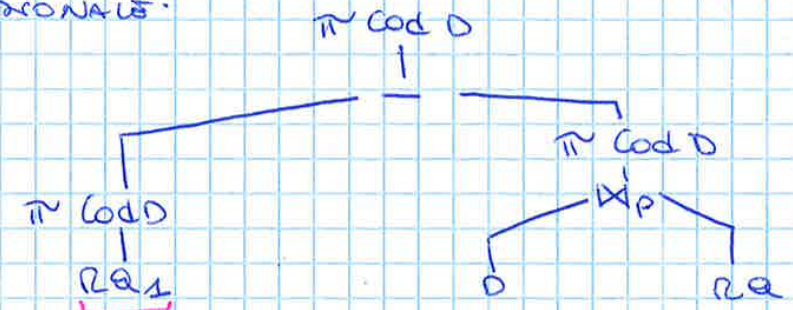
QUESTIONARIO (CodQ, NomePersona)

DOMANDA (CodD, Testo, RispostaEsatta)

RISP-QUESTIONARIO (CodQ, CodD, Risposta)

a) Visualizzare il codice delle domande a cui è stata data la risposta.
SEMPRE È SOLO! Se usano ex differenza è il NOT IN

ALGEBRA RELAZIONALE:



p: D.CodD = CQ.CodD AND RQ.Risposta <> D.Risp. esatta

USO RQ perché potrebbe succedere che, usando 'DOMANDA', alcune domande non siano ancora state risposte.

SQL:

```
SELECT DISTINCT CodD
FROM   RISP_QUESTIONARIO RQ
WHERE  CodD NOT IN ( SELECT D.CodD
                     FROM   DOMANDA, RQ2
                     WHERE  D.CodD = RQ.CodD
                     AND    Risposta <> RispEsatta)
```

c) Visualizzare il nome delle persone che hanno risposto esattamente ad almeno il 70% delle domande contenute in almeno 1 dei due questionari.

N.B: semplificazione:

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

```
SELECT  NomePersona
FROM    QUESTIONARIO Q, DOMANDA D
        RISPOSTA_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 R1
                         WHERE R1.CodQ = R.CodQ )
```

persone che hanno dato la risposta esatta

almeno il 70%

risp. contenute nel questionario considerato

c) Visualizzare il codice e il nome delle sale in cui sono state tenute tutte le proiezioni di (almeno) un film.

```
SELECT DISTINCT CodSC, NomeSala
FROM SALA_CINEMATOGRAFICA S, PROIEZIONE P
WHERE S.CodSC = P.CodSC
GROUP BY S.CodSC, P.CodFilm, NomeSala
HAVING COUNT(*) = (SELECT COUNT(*)
                   FROM PROIEZIONE P1
                   WHERE P1.CodFilm = P.CodFilm)
```

raggruppo anche per ogni sala

"sale in cui sono tenute tutte le proiezioni di almeno un film."

n° di tutte le proiezioni per un film, lo stesso per cui ho fatto la prima parte di SQL

c) Per ogni categoria, visualizzare il codice e il nome dei medici che hanno prescritto la quantità totale massima di farmaci nella categoria.

```
SELECT   CodM, NomeM, categoria
FROM     FARMACO F, MEDICO M, ACQUISTO A
WHERE    F.codF = A.codF AND M.codM = A.codM
GROUP BY CATEGORIA, codM, NOMEM
HAVING   SUM (Quantità) = (SELECT MAX(Farmaci)
                           FROM (SELECT categoria, SUM(Quantità) AS Farmaci
                                 FROM ACQUISTO A, FARMACO F
                                 GROUP BY codM, categoria) AST
                           WHERE T.categoria = F.categoria)
```

stessa categoria per la categoria per cui stiamo contando

cond. correlazione

l'acquisto di un farmaco è collegato con codM, per cui "Quantità" si riferisce ai farmaci prescritti da ogni medico

- Uso SUM perché potrebbe essere che QNT > 1

TEMI ESAME SQL

APPARTAMENTO (Coda, Superficie, Indirizzo, Città)

CONTRATTO_AFFITTO (Coda, DataInizio, DataFine, NomePersona, RataMensile)

a) Trovare il codice delle persone che hanno stipulato più di due contratti di affitto per lo stesso appartamento (in tempi diversi).

```
SELECT  NomePersona
FROM    CONTRATTO_AFFITTO  C
GROUP BY  Coda, NomePersona
HAVING COUNT (*) > 2;
```

b) Trovare il codice e l'indirizzo degli appartamenti di Torino in cui la rata mensile è sempre stata superiore a 500 € e per cui sono stati stipulati al più 5 contratti.

```
SELECT A.Coda, Indirizzo
FROM   APPARTAMENTO A, CONTRATTO_AFFITTO  C
WHERE  A.Coda = C.Coda
       AND Città = 'Torino'
       AND A.Coda  NOT IN (SELECT  Coda
                           FROM   CONTRATTO_AFFITTO  C2
                           WHERE  RataMensile <= 500)
GROUP BY  A.Coda, Indirizzo
HAVING COUNT (*) <= 5
```

b) Trovare il nome degli editori per cui almeno 10 pubblicazioni sono state vendute nel 2002 nelle librerie di Roma in più di 2000 copie.

```
SELECT  NomeEditore
FROM    EDITORE E, PUBBLICAZIONE P1
WHERE   P1.CodE = E.CodE
        AND CodP IN (SELECT CodP
                     FROM VENDITA V, LIBRERIA L
                     WHERE V.CodL = L.codL
                           AND Data >= '1/1/02'
                           AND Data <= '31/12/02'
                           AND L.Città = 'Roma'
                     GROUP By codP
                     HAVING SUM (copieVendute) > 2000)
GROUP By E.CodE, NomeEditore
HAVING COUNT (*) >= 10
```

N.B: Devo mettere in perché per il n° di copie vendute di volta e' HAVING SUM, infatti non so quante copie vengono vendute alla volta

copie vendute non in totale, ma ad acquisto !!

~ ~ ~ ~

Quiz (CodQuiz, Argomento, Punteggio)

STUDENTE (Matricola, Nome, Indirizzo, Città)

RISULTATO_TEST (Matricola, CodQuiz, RispCorretta)

a) Trovare il nome degli studenti che non hanno risposto correttamente a nessun quiz di matematica.

```
SELECT  Nome
FROM    STUDENTE S
WHERE   MATRICOLA  NOT IN ( SELECT MATRICOLA
                            FROM RISULTATO_TEST R
                                 QUIZ Q
                            WHERE RispostaCorretta = 'Si'
                                  AND Q.CodQuiz = R.CodQuiz
                                  AND ARGOMENTO = 'Matematica')
```