

# BASI DI DATI 1

Proff. S. Ceri, S. Comai, L. Tanca, G. Pozzi

Esame del 23/01/2020

Tempo: 2h 30m

Matricola o Codice Persona:	
Cognome e Nome:	
Firma:	

**Riconsegnare questo foglio e scrivere le soluzioni delle due parti su DUE FOGLI DISTINTI, entrambi intestati.**

## **Parte 1: LINGUAGGI DI INTERROGAZIONE** (su un foglio a parte rispetto alla Parte 2)

Lo schema sottostante registra le chiamate di soccorso sanitario in Lombardia, gli invii dei mezzi in risposta a tali chiamate e gli eventuali trasporti che i mezzi inviati effettuano verso gli ospedali. Gli attributi marcati con \* possono avere valore nullo. CODICEGRAVITÀ può avere i valori “Verde”, “Giallo” e “Rosso”. MEDICOABORDO è un flag booleano.

PAZIENTE ( IDPAZIENTE, COGNOME, NOME, DATA<sup>NASCITA</sup>\*, SESSO\* )

AMBULANZA ( TARGA, NUMEROSOCORRITORI, MEDICOABORDO, TELEFONO )

CHIAMATA ( IDCHIAMATA, DATA, ORA, CITTÀ, INDIRIZZO, CIVICO, DETTAGLI\* )

MEZZI\_INVIATI ( IDCHIAMATA, AMBULANZA, CODICEGRAVITÀ )

TRASPORTATI ( IDCHIAMATA, IDPAZIENTE, AMBULANZA, OSPEDALE, REPARTO\* )

### **A. SQL (13 punti)**

1. Specificare i comandi di creazione della tabella TRASPORTATI, definendo i vincoli di tupla e di dominio ritenuti opportuni ed esprimendo gli eventuali vincoli di integrità referenziale verso le altre tabelle (1 p.).
2. Per ogni città, trovare la data in cui è stato effettuato il maggior numero di chiamate di soccorso (3 p.)
3. Estrarre le chiamate alle quali è stata inviata più di un'ambulanza ma si sono concluse senza trasporti in ospedale (3 p.)
4. Trovare il cognome e il nome dei pazienti che sono stati trasportati in ospedale una sola volta nell'anno 2019, e quell'unica volta l'ambulanza non aveva il medico a bordo (4 p.).
5. Esprimere il vincolo che verifica che tutte le ambulanze inviate in codice rosso devono avere almeno tre soccorritori e un medico a bordo (2 p.)

### **B. Linguaggi formali (6 punti)**

Formulare in Algebra Relazionale, Calcolo Relazionale e Datalog l'interrogazione A.3. (2 + 2 + 2 p.)

## **Parte 2: PROGETTAZIONE E TEORIA** (su un foglio a parte rispetto alla Parte 1)

### C. Progetto Concettuale e Progetto Logico (13 punti)

Gli studenti iscritti ai corsi di studio della Laurea Magistrale della Scuola di Ingegneria Industriale e dell'Informazione sono caratterizzati da un codice persona e dai dati anagrafici. Al termine del proprio percorso preparano una tesi che può essere con o senza controrelatore (nel secondo caso è chiamata informalmente "tesina"). In entrambi i casi viene nominato un relatore tra i docenti di ruolo dell'ateneo; solo per la prima tipologia di tesi viene nominato anche un controrelatore scelto tra tutti i docenti, inclusi quelli di ruolo e a contratto. Dei docenti sono noti i dati anagrafici, il codice persona e la mail. Tutte le tesi hanno un titolo, un sommario, un insieme di parole chiave ed uno stato (inserita, approvata dal relatore, annullata). Le tesi possono essere presentate da uno o due studenti. Gli studenti possono conseguire più titoli di laurea all'interno della stessa Scuola (ad es. Informatica, Elettronica, Meccanica, ...), e quindi svolgere più tesi in appelli di laurea diversi.

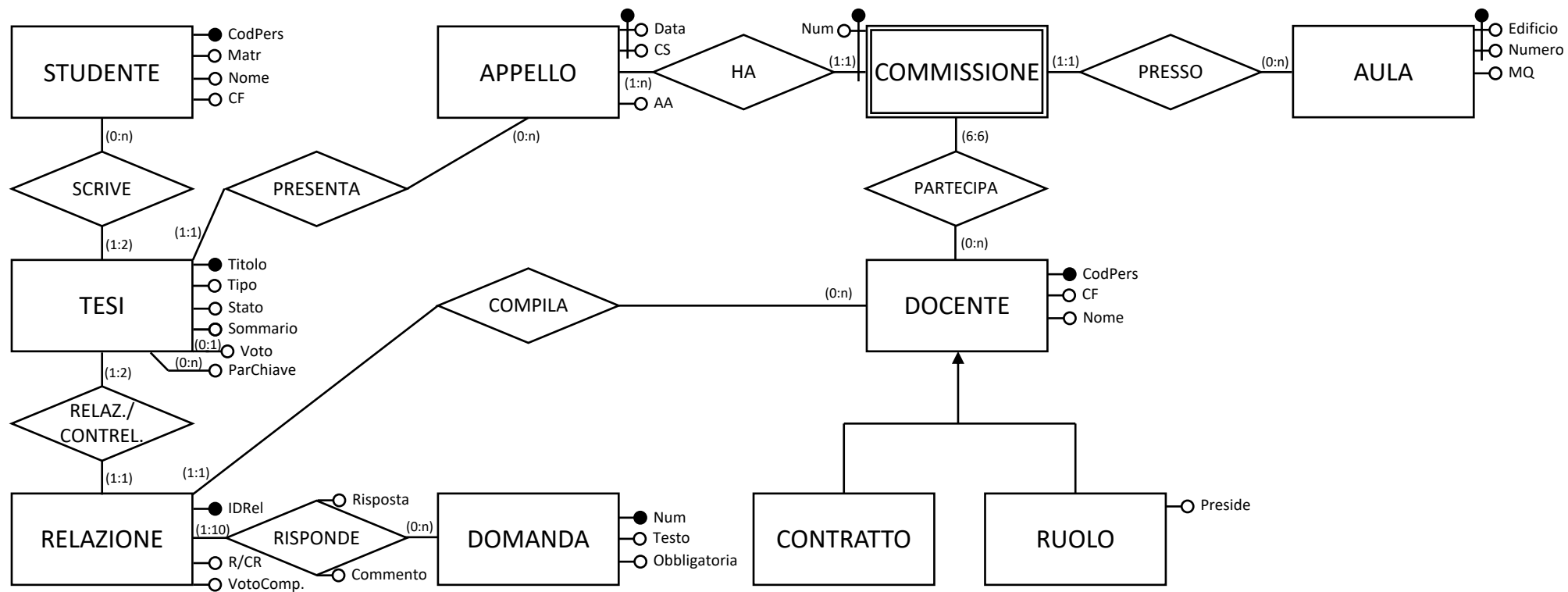
Per laurearsi, gli studenti devono iscriversi ad un appello di laurea, in cui discuteranno la tesi; l'appello è caratterizzato da un anno accademico, il corso di studi e da una data; l'iscrizione viene confermata o meno da parte della segreteria. Prima dell'esame di laurea, sia il relatore sia l'eventuale controrelatore compilano una relazione e assegnano un voto alla tesi. Ciascuna relazione è composta da un insieme di domande - di cui alcune obbligatorie - cui il docente deve rispondere con un commento testuale ed un voto compreso tra 1 e 10. Il sistema deve memorizzare, per ogni relazione, risposta e commento del docente in corrispondenza del numero di ogni domanda. Il voto finale della tesi viene deciso dall'intera commissione d'esame, tenendo conto delle relazioni preparate dal relatore e dall'eventuale controrelatore.

Gli esami di laurea della Scuola si svolgono in contemporanea in parecchie aule dell'ateneo, ciascuna con una propria commissione operativa caratterizzata da un numero e costituita da 6 professori e dal Preside della Scuola. Ogni tesi (e di conseguenza gli studenti che la presentano) viene assegnata ad una commissione. Le aule sono caratterizzate da un numero, edificio, nome e numero di posti disponibili.

- *Il progetto è valutato per completezza, correttezza, leggibilità, minimalità e autoesplicatività.*
- *Ricordare anche di specificare gli **identificatori** di tutte le entità e le **cardinalità** di tutte le associazioni, e di disporre il progetto concettuale e quello logico su due facciate affiancate, in modo da poterli osservare simultaneamente.*

### D. Teoria (1 punto)

Si confronti il potere espressivo dei linguaggi formali (Algebra Relazionale, Calcolo Relazionale e Datalog), indicando e motivando le relazioni di equivalenza e inclusione.



STUDENTE(CodPers, Matr, Nome, CF)

PAROLE\_CHIAVE(Tesi, Parola)

TESI(Titolo, Tipo, Stato, Sommario, Voto\*)

DOCENTE(CodPers, CF, Nome, Ruolo/Contratto, Preside)

APPELLO(CS, Data, AA)

COMMISSIONE(Num, CS, Data, Edificio, Numero)

AULA(Edificio, Numero, MQ)

SCRIVE(CPStud, Titolo)

RELAZIONE(IDRel, Relazione/Controrelazione, Tesi, CPDocente, VotoComplessivo)

DOMANDA(Num, Testo, Obbligatoria)

RISPONDE(IDRel, NumDomanda, Risposta, Commento)

PARTECIPA(CPDocente, NumCommissione, CS, Data)

**A.1) Specificare i comandi di creazione della tabella Trasportati, definendo i vincoli di tupla e di dominio ritenuti opportuni ed esprimendo gli eventuali vincoli di integrità referenziale verso le altre tabelle (1 p.).**

```
CREATE TABLE Trasportati (  
    IDChiamata INT(6),  
    IDPaziente INT(6) REFERENCES Paziente(IDPaziente)  
        ON UPDATE CASCADE  
        ON DELETE NO ACTION,  
    IDAmbulanza CHAR(7) NOT NULL,  
    Ospedale VARCHAR(20) NOT NULL,  
    Reparto VARCHAR(20),  
  
    PRIMARY KEY (IDChiamata, IDPaziente),  
    FOREIGN KEY (IDChiamata, Ambulanza) REFERENCES Mezzi_Inviati(IDChiamata, Ambulanza)  
        ON UPDATE CASCADE  
        ON DELETE NO ACTION  
)
```

**A.2) Per ogni città, trovare la data in cui è stato effettuato il maggior numero di chiamate di soccorso (3 p.)**

```
CREATE VIEW NumeroChiamate(Città, Data, NumChiamate) AS(  
    SELECT C.Città, C.Data, COUNT(*)  
    FROM C.Chiamate  
    GROUP BY C.Città, C.Data  
)  
  
SELECT N.Città, N.Data  
FROM NumeroChiamate N  
WHERE N.NumChiamate = (  
    SELECT MAX(N2.NumChiamate)  
    FROM NumeroChiamate N2  
    WHERE N2.Città = N.Città  
)
```

*oppure:*

```
CREATE VIEW NumeroChiamate(Città, Data, NumChiamate) AS(  
    SELECT C.Città, C.Data, COUNT(*)  
    FROM C.Chiamate  
    GROUP BY C.Città, C.Data  
)  
  
SELECT N.Città, N.Data  
FROM NumeroChiamate N  
WHERE (N.Città, N.NumChiamate) IN(  
    SELECT N2.Città, MAX(N2.NumChiamate)  
    FROM NumeroChiamate N2  
    GROUP BY N2.Città  
)
```

*oppure:*

```

SELECT C.Città, C.Data
FROM Chiamate C
GROUP BY N.Città, N.Data
HAVING COUNT(*) >= ALL(
    SELECT COUNT(*)
    FROM Chiamate C2
    WHERE C2.Città = C.Città
    GROUP BY C2.Data
)

```

**A.3) Estrarre le chiamate alle quali è stata inviata più di un'ambulanza ma si sono concluse senza trasporti in ospedale (3 p.)**

```

SELECT *
FROM Chiamate C
WHERE C.IDChiamata NOT IN(
    SELECT T.IDChiamata
    FROM Trasportati T
) AND C.IDChiamata IN(
    SELECT M.IDChiamata
    FROM Mezzi_Inviati M
    GROUP BY M.IDChiamata
    HAVING COUNT(*) > 1
)

```

*oppure:*

```

SELECT M.IDChiamata
FROM Mezzi_Inviati M
GROUP BY M.IDChiamata
HAVING COUNT(*) > 1

```

EXCEPT

```

SELECT T.IDChiamata
FROM Trasportati M

```

**A.4) Trovare il cognome e il nome dei pazienti che sono stati trasportati in ospedale una sola volta nell'anno 2019, e quell'unica volta l'ambulanza non aveva il medico a bordo (4 p.).**

```

SELECT Cognome, Nome
FROM Paziente P, Ambulanza A, Chiamata C, Trasportati T
WHERE P.IDPaziente = T.IDPaziente AND
      A.Targa = T.IDAmbulanza AND
      C.IDChiamata = T.IDChiamata AND
      C.Data.YEAR() = 2019 AND
      A.MedicoABordo = 0 AND
      1 = (
        SELECT COUNT(*)
        FROM Trasportati T2, Chiamata C2
        WHERE T2.IDChiamata = C2.IDChiamata
              C2.Data.YEAR() = 2019 AND
              T2.IDPaziente = P.IDPaziente
      )

```

**A.5) Esprimere il vincolo che verifica che tutte le ambulanze inviate in codice rosso devono avere almeno tre soccorritori e un medico a bordo (2 p.)**

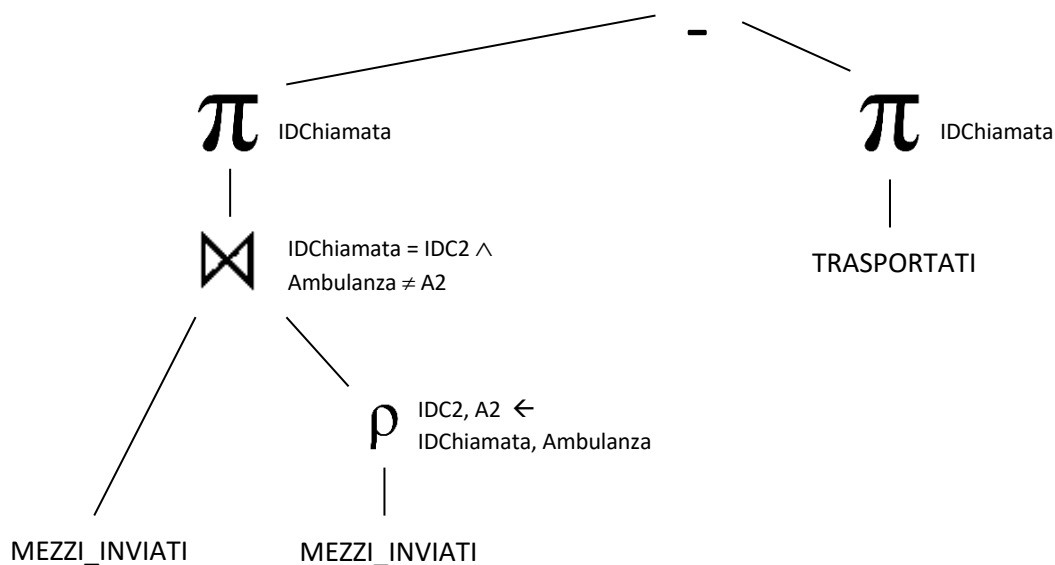
```

CREATE ASSERTION No3Rossi AS CHECK(
  NOT EXISTS(
    SELECT *
    FROM Mezzi_Inviati M, Ambulanza A
    WHERE M.Ambulanza = A.Targa AND
          M.CodiceGravità = 'Rosso' AND
          (
            A.NumeroSoccorritori < 3 OR
            A.MedicoABordo = 0
          )
  )
)

```

**B) Estrarre le chiamate alle quali è stata inviata più di un'ambulanza ma si sono concluse senza trasporti in ospedale (2 + 2 + 2 p.)**

Algebra relazionale



Calcolo relazionale

$$\{t \mid \exists t1 \in \text{MEZZI\_INVIATI}, \exists t2 \in \text{MEZZI\_INVIATI} \ (t[\text{IDChiamata}] = t1[\text{IDChiamata}] \wedge t1[\text{IDChiamata}] = t2[\text{IDChiamata}] \wedge t1[\text{Ambulanza}] \neq t2[\text{Ambulanza}] \wedge \neg (\exists t3 \in \text{TRASPORTATI} (t3[\text{IDChiamata}] = t1[\text{IDChiamata}]))\}$$

### Datalog

ALMENO\_2(IDC) :- MEZZI\_INVIATI(IDC, A1, \_), MEZZI\_INVIATI(IDC, A2, \_), A1 ≠ A2

RISULTATO(IDC) :- ALMENO\_2(IDC), ¬ TRASPORTATI(IDC, \_ , \_ , \_)

? – RISULTATO(x)