

Basi di dati I — 10 settembre 2019
Tempo a disposizione: un'ora e 30 minuti.

Possibili soluzioni

Cognome: _____ **Nome:** _____ **Matricola:** _____

Domanda 1 (20%)

Considerare la seguente base di dati, che fa riferimento alle pratiche gestite da una assicurazione sanitaria.

ASSISTITI		
Codice	Cognome	Nome
101	Rossi	Mario
102	Bruni	Gino
103	Verdi	Piero
104	Neri	Luigi

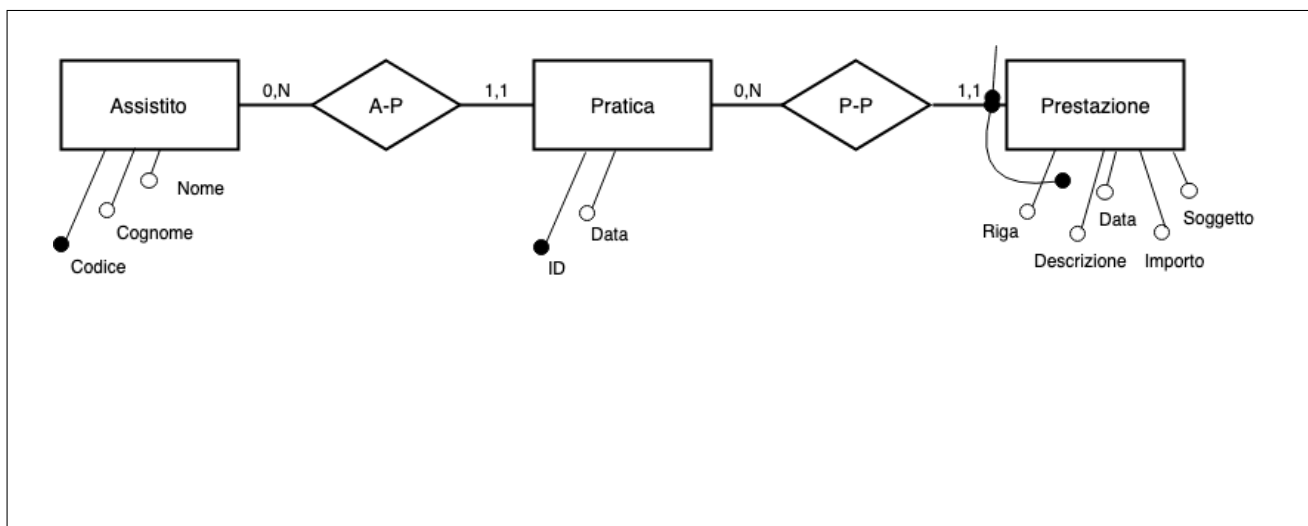
PRATICHE		
ID	Assistito	Data
1001	101	24/01/19
1002	102	30/03/19
1003	101	15/03/19

PRESTAZIONI					
Pratica	Riga	Descrizione	Soggetto	Data	Importo
1001	1	Vis. oculistica	Dott. Visus	15/01/2019	200,00
1001	2	Analisi sangue	Centro Salus	18/01/2019	105,00
1002	1	Visita Ortop.	Centro Salus	23/01/2019	135,00
1002	2	Radiografia	Centro Salus	22/01/2019	100,00
1002	3	Radiografia	Centro Salus	20/02/2019	100,00
1003	1	Vis. otorino	Dott. Bruni	21/02/2019	120,00

Come emerge dai dati:

- ogni pratica è relativa ad un assistito ed è composta da un certo numero di prestazioni
- ogni assistito ha codice, cognome e nome
- ogni prestazione (che costituisce una “riga” della relativa pratica) ha
 - una descrizione, che è una semplice stringa di testo
 - un soggetto (verso cui è stato effettuato il pagamento da parte dell’assistito) che può essere un medico o una azienda, ma assumiamo ancora che sia una semplice stringa di testo
 - una data e un importo

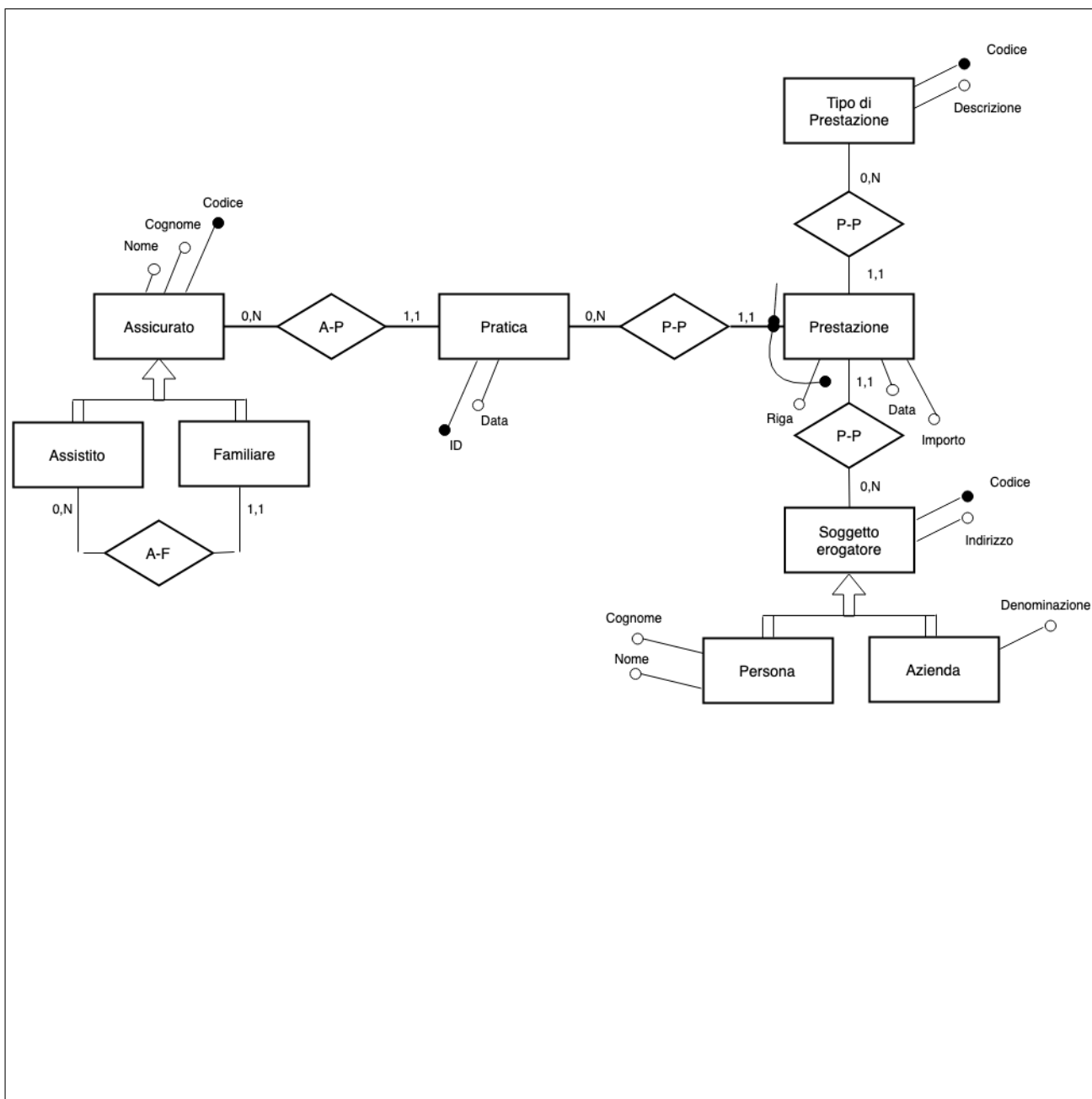
Definire uno schema Entity-Relationship che descriva questa applicazione e da cui lo schema fornito possa essere derivato nella fase di progettazione logica (**fare riferimento allo schema relazionale mostrato, senza aggiungere attributi**)



Domanda 2 (20%)

Modificare lo schema proposto in risposta alla domanda precedente, per tenere conto delle seguenti specifiche aggiuntive (**mostrare l'intero schema**):

- Ogni “soggetto” ha un codice e un indirizzo e può essere una persona fisica, con cognome e nome, oppure una azienda, con una denominazione.
- I possibili tipi di prestazioni sono predefiniti e hanno ciascuno un codice e una descrizione
- Per ogni assistito, c'è un insieme di familiari a cui possono riferirsi le pratiche. Ogni pratica (e quindi tutte le relative prestazioni) sono relative ad una stessa persona, assistito o familiare



Domanda 3 (30%) Considerare nuovamente la base di dati nella domanda 1, riportata qui sotto per comodità

ASSISTITI		
Codice	Cognome	Nome
101	Rossi	Mario
102	Bruni	Gino
103	Verdi	Piero
104	Neri	Luigi

PRATICHE		
ID	Assistito	Data
1001	101	24/01/19
1002	102	30/03/19
1003	101	15/03/19

PRESTAZIONI					
Pratica	Riga	Descrizione	Soggetto	Data	Importo
1001	1	Vis. oculistica	Dott. Visus	15/01/2019	200,00
1001	2	Analisi sangue	Centro Salus	18/01/2019	105,00
1002	1	Visita Ortop.	Centro Salus	23/01/2019	135,00
1002	2	Radiografia	Centro Salus	22/01/2019	100,00
1002	3	Radiografia	Centro Salus	20/02/2019	100,00
1003	1	Vis. otorino	Dott. Bruni	21/02/2019	120,00

1. Formulare in SQL l'interrogazione che calcola, per ogni assistito, l'importo totale delle prestazioni, mostrando codice, cognome, nome e importo totale

```
SELECT Codice, Cognome, Nome, SUM(Importo) AS ImpTotale
FROM Assistiti JOIN Pratiche ON (Codice=Assistito)
      JOIN Prestazioni ON (ID= Pratica)
GROUP BY Codice, Cognome, Nome
```

Volendo mostrare anche gli assistiti senza prestazioni (indicando per loro 0 come totale):

```
SELECT Codice, Cognome, Nome, SUM(Importo) AS ImpTotale
FROM Assistiti JOIN Pratiche ON (Codice=Assistito)
      JOIN Prestazioni ON (ID= Pratica)
GROUP BY Codice, Cognome, Nome
UNION
SELECT Codice, Cognome, Nome, 0 AS ImpTotale
FROM Assistiti A
WHERE NOT EXISTS (SELECT *
      FROM Assistiti A1 JOIN Pratiche ON (Codice=Assistito)
      JOIN Prestazioni ON (ID= Pratica)
      WHERE A1.Codice=A.Codice
)
```

2. Formulare in SQL l'interrogazione che trova il soggetto presso il quale è stato svolto il massimo numero di prestazioni; mostrare il soggetto e il numero delle prestazioni

```
CREATE VIEW ContaPrestazioni AS
SELECT Soggetto, COUNT(*) AS NumPrestazioni
FROM Prestazioni
GROUP BY Soggetto;

SELECT Soggetto, NumPrestazioni
FROM ContaPrestazioni
WHERE NumPrestazioni = (SELECT MAX(NumPrestazioni) From ContaPrestazioni)
```

3. Formulare in SQL l'interrogazione che trova gli assistiti che hanno un totale di importo delle richieste formulate superiore alla media di almeno il 50% (se la media delle richieste fosse 100 Euro, si dovrebbero trovare gli assistiti che hanno un totale di richieste superiore a 150 Euro)

```
CREATE VIEW TotalePerAssistito AS
  ( SELECT Codice, Cognome, Nome, SUM(Importo) AS ImpTotale
    FROM Assistiti JOIN Pratiche ON (Codice=Assistito)
                    JOIN Prestazioni ON (ID= Pratica)
  GROUP BY Codice, Cognome, Nome
  UNION
  SELECT Codice, Cognome, Nome, 0 AS ImpTotale
  FROM Assistiti A
  WHERE NOT EXISTS (SELECT *
                    FROM Assistiti A1 JOIN Pratiche ON (Codice=Assistito)
                    JOIN Prestazioni ON (ID= Pratica)
                    WHERE A1.Codice=A.Codice
  )
);

SELECT *
FROM TotalePerAssistito
WHERE ImpTotale >= 1.5*(SELECT AVG(ImpTotale) FROM TotalePerAssistito)
```

Domanda 4 (20%)

Considerare lo schema concettuale seguente:



Valutare la convenienza dell'introduzione di una relationship C-F fra Cliente e Filiale oppure di una relationship C-B fra Cliente e Banca oppure di entrambe. Tali relationship sono ridondanti in quanto derivabili dalla concatenazione delle relationship esistenti. Considerare un carico applicativo che includa come operazioni principali le seguenti:

1. inserimento di un nuovo rapporto fra cliente e agenzia (cioè di una occorrenza della relationship C-A), dati il codice del cliente e quello dell'agenzia, assumendo per semplicità che il cliente sia già presente e il rapporto sia con una agenzia di una filiale e di una banca con cui il cliente non ha ancora rapporti (cliente, agenzia, filiale e banca sono invece già nella base di dati), con frequenza $f_1 = 100$
2. ricerca delle filiali con cui un cliente (di cui è dato il codice) ha rapporti, con frequenza $f_2 = 1000$
3. ricerca delle banche con cui un cliente (di cui è dato il codice) ha rapporti, con frequenza $f_3 = 1000$

Supporre che, mediamente, un cliente abbia rapporti con $N = 5$ agenzie, di filiali diverse di banche diverse. Considerare i costi delle letture e scritture delle entità e delle relationship molti a molti (ignorando quelli delle relationship uno a molti) e considerare il costo delle scritture doppio di quello delle letture.

Soluzione

senza ridondanza $c_1 \times f_1 + c_2 \times f_2 + c_3 \times f_3 = 2 \times 100 + 2 \times N \times 1000 + 3 \times N \times 1000 = ca. 25.000$

- c_1 è pari a 2 perché si deve scrivere C-A (una sola occorrenza)
- c_2 è pari a $2 \times N = 2 \times 5$ perché si debbono leggere $N = 5$ occorrenze di C-A (dato il codice del cliente) e poi altrettante di Agenzia
- c_3 è pari a $3 \times N = 3 \times 5$ perché si debbono leggere $N = 5$ occorrenze di C-A (dato il codice del cliente) e poi altrettante di Agenzia e poi altrettante anche di Filiale

con ridondanza su C-F $c_1 \times f_1 + c_2 \times f_2 + c_3 \times f_3 = 5 \times 100 + N \times 1000 + 2 \times N \times 1000 = ca. 15.000$

- c_1 è pari a 5 perché si debbono scrivere C-A e C-F e si deve leggere Agenzia (un'occorrenza ciascuna)
- c_2 è pari a $N = 10$ perché è sufficiente leggere N occorrenze di C-F (dato il codice del cliente)
- c_3 è pari a $2 \times N = 2 \times 5$ perché si debbono leggere $N = 5$ occorrenze di C-F (dato il codice del cliente) e poi altrettante anche di Filiale

con ridondanza su C-B $c_1 \times f_1 + c_2 \times f_2 + c_3 \times f_3 = 5 \times 100 + 2 \times N \times 1000 + N \times 1000 = ca. 15.000$

- c_1 è pari a 6 perché si debbono scrivere C-A e C-B e si debbono leggere Agenzia e Filiale (un'occorrenza ciascuna)
- c_2 è pari a $2 \times N = 2 \times 5$ come nel caso senza ridondanza
- c_3 è pari a $N = 10$ perché è sufficiente leggere N occorrenze di C-B (dato il codice del cliente)

con ridondanza su C-F e C-B $c_1 \times f_1 + c_2 \times f_2 + c_3 \times f_3 = 8 \times 100 + N \times 1000 + N \times 1000 = ca. 10.800$

- c_1 è pari a 8 perché si debbono scrivere C-A, C-F e C-B e si debbono leggere Agenzia e Filiale (un'occorrenza ciascuna)
- c_2 è pari a $N = 10$ perché è sufficiente leggere N occorrenze di C-F (dato il codice del cliente)
- c_3 è pari a $N = 10$ perché è sufficiente leggere N occorrenze di C-B (dato il codice del cliente)

Risulta quindi conveniente introdurre tutte e due le ridondanze

Domanda 5 (10%)

Definire (con una opportuna notazione) su una relazione

RETRIBUZIONI(Matricola,Lordo,Imposte,StipNetto,Verifica)

i vincoli che impongano le seguenti condizioni

- Il valore di Verifica è “OK” se e solo se StipNetto è uguale alla differenza fra Lordo e Imposte.

```
CHECK (( (Verifica = 'OK') ) AND (StipNetto = Lordo - Imposte) ) OR  
        ( (Verifica <> 'OK') ) AND (StipNetto <> Lordo - Imposte) )
```

- Se il valore di Verifica è “OK”, allora StipNetto è uguale alla differenza fra Lordo e Imposte (si noti che in questo caso non si vuole invece imporre nessuna condizione se il valore di Verifica è diverso da “OK”).

```
CHECK ( (NOT (Verifica = 'OK') ) OR (StipNetto = Lordo - Imposte) )
```