

Centro Trapianti “Un Cuore per la Vita”

Abstract

“Un Cuore per la Vita” è un centro trapianti che ha deciso di rinnovare la propria base di dati. Il centro si occupa di gestire, per gli ospedali del territorio, tutte le fasi che riguardano il trapianto di un organo:

1. Registra i pazienti con i loro dati anagrafici e sanitari
2. Inserisce le richieste di organi in una lista d’attesa con priorità, in modo da garantire che gli organi disponibili vengano dati a chi più ne ha bisogno
3. Registra le anagrafiche dei potenziali donatori con i loro dati anagrafici e sanitari
4. Inserisce gli organi disponibili per la donazione in una lista, in modo da incrociarli con le richieste in lista d’attesa
5. Verifica la compatibilità dei gruppi sanguigni di ricevente e donatore
6. Gestisce le anagrafiche del personale medico
7. Segue la programmazione degli interventi chirurgici, in particolare organizza l’equipe medica che lo effettuerà
8. Calcola il costo dell’intervento e verifica l’eventuale copertura assicurativa del paziente ricevente
9. Tiene un archivio storico dei trapianti effettuati.

In questo modo, tramite il sistema informatico, il centro trapianti si assicura di far avere gli organi disponibili ai pazienti opportuni e ottimizza l’organizzazione di tutte le procedure.

Analisi dei requisiti

Nel database sono salvati tutti i **gruppi sanguigni** possibili. Per ciascun gruppo sanguigno sono salvate anche le **compatibilità**, ovvero verso quale gruppo sanguigno è possibile donare un organo.

Oltre ai gruppi sanguigni e alle relative compatibilità nel database sono salvati anche tutti gli **organi** che è possibile donare/ricevere. Per ciascun organo è noto anche il costo base per il relativo trapianto (che non può avere valori negativi).

Nella base di dati sono presenti tutte le **anagrafiche** delle figure coinvolte in ciascun trapianto. Di ciascuna anagrafica sono noti:

- Codice fiscale
- Cognome
- Nome
- Data di nascita

Autori del progetto:

Benetazzo Antonio – Matricola 2034528

Scantamburlo Alex – Matricola 2042326

Tra le varie anagrafiche sono presenti i **pazienti**. Ogni paziente può essere un **ricevente** o anche un **donatore**. Di ciascun paziente sono noti anche:

- Altezza (in cm)
- Peso (in kg)
- Gruppo sanguigno

Sempre tra le anagrafiche sono presenti anche i membri del **personale medico**. Sono presenti medici strutturati (ricevono un compenso per ogni intervento effettuato) e tirocinanti (non ricevono alcun compenso sugli interventi effettuati). Di ciascun membro del personale medico sono noti:

- Mansione
- Compenso (non può avere valori negativi)

Sia i pazienti che il personale medico possono avere un'**assicurazione sanitaria** che copre le spese mediche, in particolare in caso di un eventuale trapianto. Di ciascuna polizza assicurativa sono noti:

- Cliente
- Compagnia assicurativa
- Massimale (non può avere valori negativi)

Ogni paziente che si rende disponibile per la donazione di un organo viene registrato in una lista apposita dove viene salvato:

- Identificativo del paziente
- Organo che intende donare
- Data/ora dell'avvenuto inserimento in lista.

Anche ciascun paziente che attende di ricevere un organo viene inserito in una **lista di attesa**. Come per i donatori, anche in questo caso vengono salvati i dati di cui sopra. In aggiunta viene indicato anche un codice di priorità (valore compreso tra 0 e 10) sulla base della condizione clinica del paziente: ad una priorità maggiore corrisponde una precedenza rispetto a chi ha una priorità inferiore nell'assegnazione di un organo disponibile; a parità di priorità avrà precedenza chi risulta inserito in lista d'attesa da più tempo.

Una volta trovata una corrispondenza tra un paziente donatore e un paziente in lista d'attesa (sulla base della compatibilità dei gruppi sanguigni e delle priorità definite dalla lista d'attesa) viene programmato l'intervento per il trapianto. Viene registrato sulla lista dei **trapianti**, dove sono salvati trapianti programmati (ancora da effettuare) come anche trapianti già effettuati; dei vari trapianti sono noti:

- Ricevente
- Donatore (deve essere diverso dal ricevente)
- Organo da trapiantare
- Data dell'intervento

A ciascun trapianto vengono assegnati anche i membri del personale medico che vanno a formare l'**equipe chirurgica** per quel trapianto. In ogni equipe chirurgica non può essere presente un tirocinante senza la presenza di un medico strutturato con la stessa mansione che possa supervisionare il suo operato.

Il costo di ciascun trapianto che dovrà sostenere il paziente ricevente è calcolato partendo dal costo base per il trapianto del relativo organo a cui vanno sommati i compensi di ciascun medico che fa parte dell'equipe per quel trapianto. Se il paziente possiede una copertura assicurativa sarà questa a coprire il costo dell'intervento fino al raggiungimento del massimale previsto dalla polizza; qualora il costo dell'intervento fosse superiore al massimale il paziente dovrà farsi carico della quota eccedente.

Glossario dei termini

Termine	Descrizione
Trapianto	Intervento chirurgico volto alla sostituzione di un organo
Organo	Organo da trapiantare
Costo base	Costo di base del trapianto di un organo
Gruppo sanguigno (GS)	Gruppo sanguigno di un paziente di cui va controllata la compatibilità
Compatibilità	Possibilità da parte di un GS di donare ad un altro GS
Anagrafica	Dati anagrafici che identificano una persona
Paziente	Persona coinvolta nel trapianto, può essere donatore o ricevente
Ricevente	Persona coinvolta nel trapianto per la ricezione di un organo
Donatore	Persona coinvolta nel trapianto per la donazione di un organo
Personale medico	Persona che si occupa dell'intervento chirurgico
Mansione	Ruolo del medico (ad esempio infermiere, anestesista, chirurgo, ecc.)
Compenso	Compenso di ciascun medico per ciascun intervento
Medico strutturato	Medico che ha completato la formazione e può operare in autonomia
Tirocinante	Medico in fase di formazione, non può operare senza supervisione
Equipe chirurgica	Insieme di medici che si occupano di un trapianto
Assicurazione sanitaria	Assicurazione che copre (tutto o in parte) il costo del trapianto
Massimale	Cifra massima coperta dall'assicurazione
Lista d'attesa	Lista che gestisce le richieste di organi in base alle priorità assegnate
Priorità	Codice assegnato sulla base delle condizioni cliniche

Operazioni tipiche

Operazione	Frequenza
Inserimento di un paziente	1.000 al giorno
Inserimento di un medico	50 al giorno
Aggiunta di un paziente alla lista d'attesa	1.000 al giorno
Ricerca di organi disponibili	10.000 al giorno
Programmazione di un trapianto	500 al giorno
Formazione di un'equipe di medici per un trapianto	500 al giorno
Verifica dell'eventuale copertura assicurativa	500 al giorno
Calcolo del costo di un trapianto	500 al giorno
Ricerca di trapianti effettuati su di un paziente	100 al giorno

Progettazione concettuale

Lista entità

- Organi:
 - Id INT PRIMARY KEY
 - Nome VARCHAR(255) NOT NULL UNIQUE
 - Costo INT NOT NULL
- Gruppi sanguigni:
 - Gruppo VARCHAR(3) PRIMARY KEY
- Anagrafiche:
 - Codice fiscale CHAR(16) PRIMARY KEY
 - Cognome VARCHAR (255) NOT NULL
 - Nome VARCHAR(255) NOT NULL
 - Data di nascita DATE NOT NULL

L'entità Anagrafiche si specializza in due sottocategorie con generalizzazione totale:

- Pazienti:
 - Altezza (in cm) INT
 - Peso (in kg) INT
 - Gruppo sanguigno VARCHAR(3) NOT NULL

L'entità Pazienti si divide a sua volta in due sottocategorie con generalizzazione totale ma senza caratterizzazione

- Pazienti riceventi
- Pazienti donatori

Autori del progetto:

Benetazzo Antonio – Matricola 2034528

Scantamburlo Alex – Matricola 2042326

- Personale medico:
 - Mansione RUOLO (tipo ENUM) NOT NULL
 - Compenso INT NOT NULL
- Assicurazioni:
 - Cliente CHAR(16) PRIMARY KEY
 - Massimale INT NOT NULL
 - Compagnia assicurativa VARCHAR(255) NOT NULL
- Trapianti:
 - Id INT PRIMARY KEY
 - Ricevente CHAR(16) NOT NULL
 - Donatore CHAR(16) NOT NULL
 - Organo INT NOT NULL
 - Data DATE NOT NULL
- Lista organi:
 - Paziente CHAR(16) PRIMARY KEY
 - Organo INT PRIMARY KEY
 - Data inserimento TIMESTAMP NOT NULL

L'entità Lista organi si specializza in due sottocategorie con generalizzazione totale:

- Organi disponibili (senza caratterizzazione)
- Lista d'attesa:
 - Priorità INT NOT NULL

Tabella delle relazioni

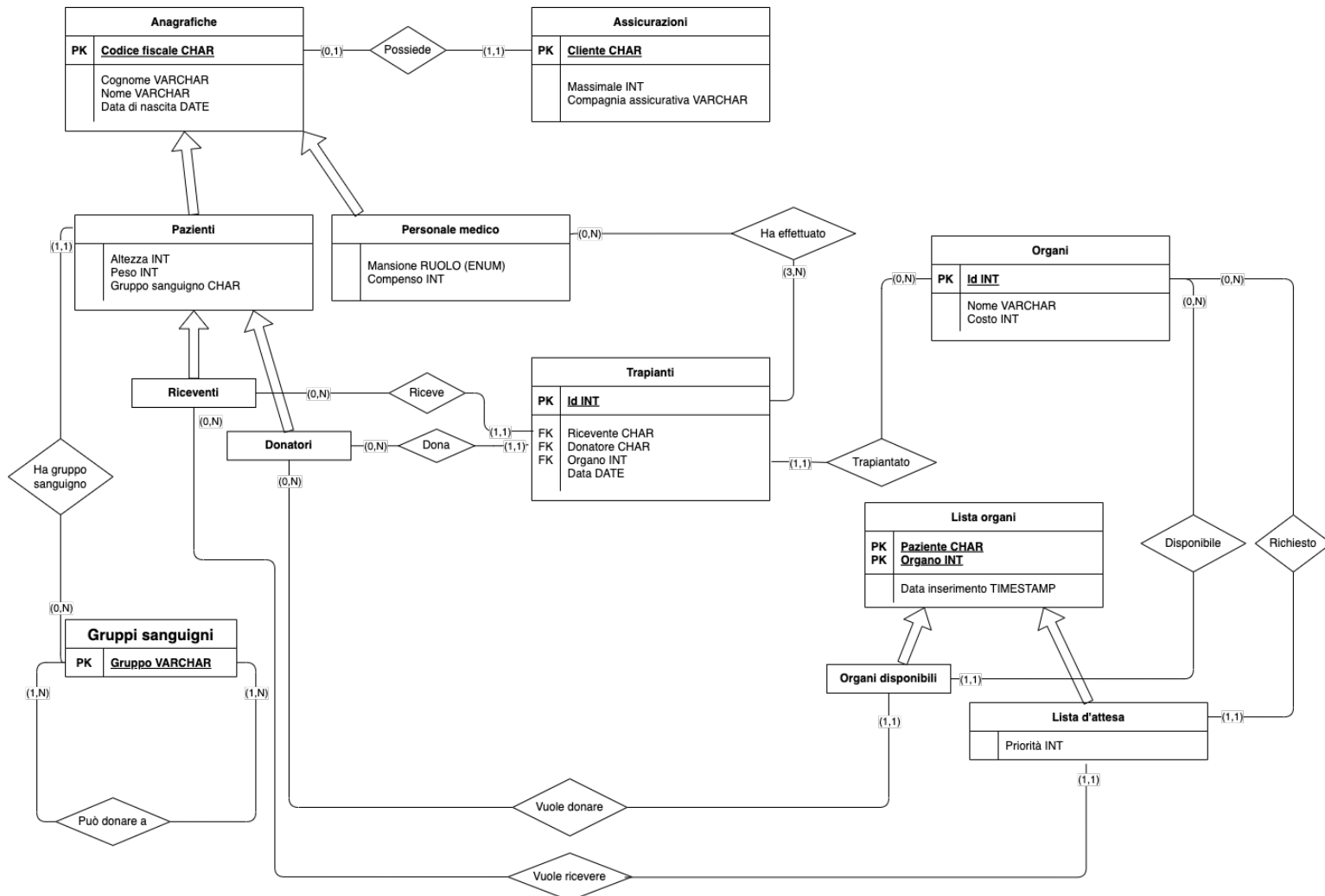
Relazione	Entità coinvolte	Descrizione
Ha gruppo sanguigno	Pazienti (1,1) Gruppi sanguigni (0,N)	Un paziente deve avere uno e un solo gruppo sanguigno, un gruppo sanguigno può essere presente in 0-N pazienti
Ha assicurazione	Anagrafiche (0,1) Assicurazioni (1,1)	Una persona può avere al massimo una polizza assicurativa (o anche nessuna), una polizza assicurativa copre una sola persona
Può donare a	Gruppi sanguigni (1,N) Gruppi sanguigni (1,N)	Un gruppo sanguigno può donare sempre allo stesso gruppo sanguigno ma potrebbe donare anche ad altri gruppi sanguigni
Vuole donare	Donatori (0,N) Organi disponibili (1,1)	Un donatore può essere disponibile a donare da 0 a N organi, un organo disponibile appartiene ad uno e un solo donatore
Disponibile	Organi (0,N) Organi disponibili (1,1)	Un organo può essere disponibile in quantità da 0 a N, una disponibilità fa riferimento ad uno ed un solo organo

Vuole ricevere	Riceventi (0,N) Lista d'attesa (1,1)	Un ricevente può essere in lista d'attesa per ricevere da 0 a N organi, un organo in lista d'attesa appartiene ad uno e un solo ricevente
Richiesto	Organi (0,N) Lista d'attesa (1,1)	Un organo può essere richiesto in quantità da 0 a N, una richiesta fa riferimento ad uno ed un solo organo
Riceve	Ricevente (0,N) Trapianti (1,1)	Un ricevente può aver ricevuto (o aver programmato) da 0 a N organi, un trapianto ha uno e un solo ricevente
Dona	Donatore (0,N) Trapianti (1,1)	Un donatore può aver donato (o aver programmato) da 0 a N organi, un trapianto ha uno e un solo donatore
Trapiantato	Organi (0,N) Trapianti (1,1)	Un organo può essere stato trapiantato da 0 a N volte, un trapianto coinvolge uno e un solo organo
Ha effettuato	Personale medico (0,N) Trapianti (3,N)	Un medico può aver effettuato da 0 a N trapianti, ogni trapianto può essere effettuato da uno o più medici

Vincoli non rappresentabili tramite schema E-R

- Ogni trapianto programmato o già effettuato deve avere donatore e ricevente con gruppi sanguigni compatibili
- Per ogni trapianto deve esserci almeno un chirurgo, almeno un anestesista e almeno un infermiere nell'equipe chirurgica
- Per ogni trapianto un medico non può comparire sia come membro dell'equipe che come paziente (ricevente o donatore)
- Per ogni trapianto non può essere presente nell'equipe un medico tirocinante senza un medico strutturato con la stessa mansione che supervisioni il suo operato

Schema concettuale



Progettazione logica

Ristrutturazione - Analisi delle ridondanze

Dall'analisi effettuata non risultano ridondanze sui dati.

Ristrutturazione - Eliminazione delle generalizzazioni

Anagrafiche \leftarrow Pazienti, Personale medico:

- Le entità Pazienti e Personale medico effettivamente sono due entità distinte, con caratterizzazioni peculiari. Per i pazienti è necessario indicare i dati sanitari, quali altezza, peso e gruppo sanguigno. Per il personale medico non sono necessari i dati sanitari ma sono necessari altri dati quali la mansione e il compenso. Si decide quindi di mantenere le entità separate.

Autori del progetto:

Benetazzo Antonio – Matricola 2034528

Scantamburlo Alex – Matricola 2042326

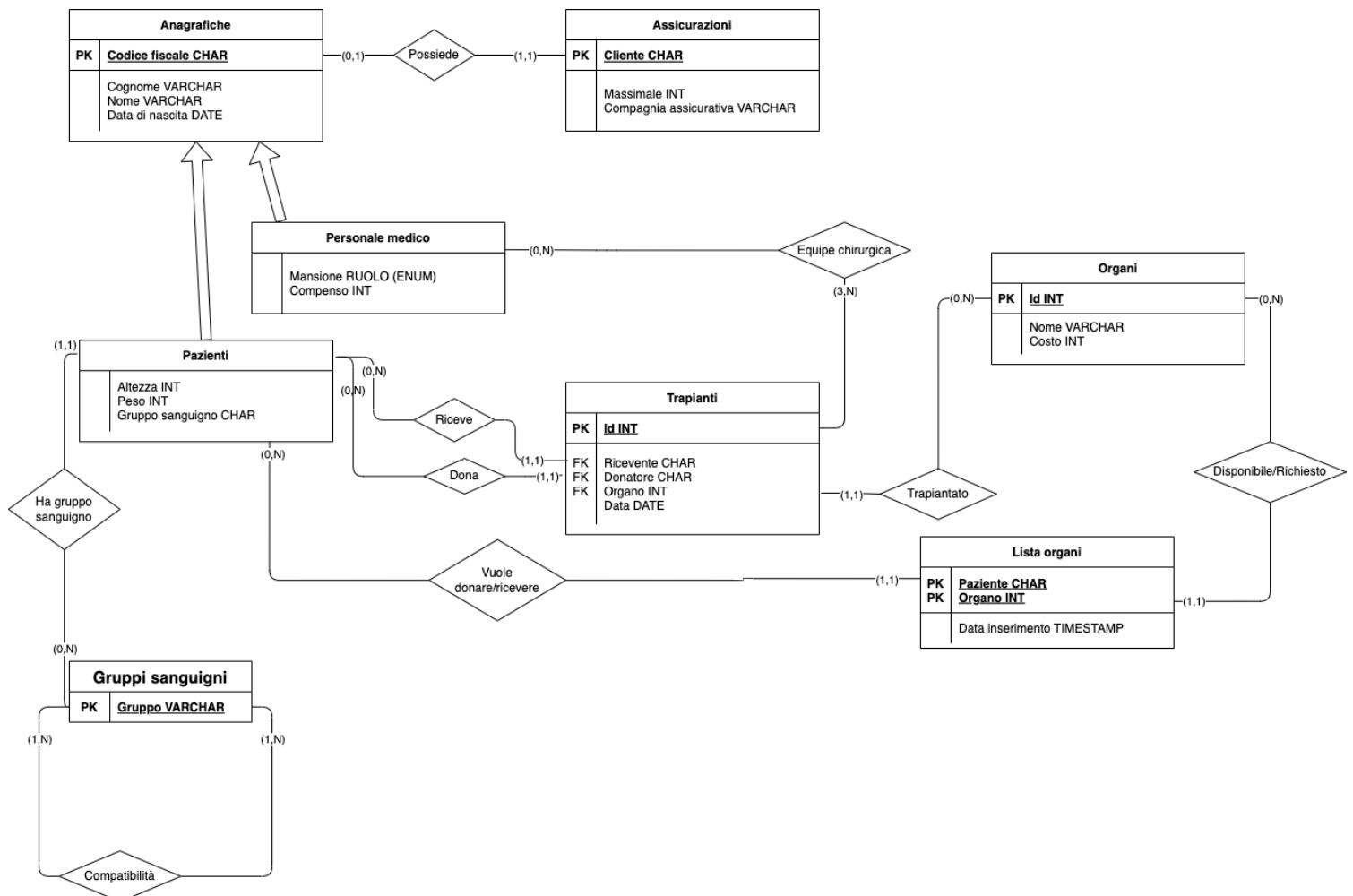
Pazienti \Leftarrow Riceventi, Donatori:

- Si è scelto di accorpare le entità figlie in quella genitrice, in quanto rappresentano lo stesso concetto, la distinzione sta nel come entrano in rapporto con le altre entità.

Lista organi \Leftarrow Organi disponibili, Lista d'attesa:

- Si è scelto di accorpare le entità figlie in quella genitrice, in quanto rappresentano lo stesso concetto. Viene aggiunto l'attributo priorità, indispensabile per il concetto di lista d'attesa; tale attributo potrà avere valore NULL per rappresentare il concetto di organo disponibile.

Diagramma ER ristrutturato



Schema relazionale e vincoli di integrità referenziale

Gli attributi con l'asterisco possono avere valori nulli, in seguito ad ogni relazione sono riportate le chiavi esterne e i relativi attributi coinvolti.

(A→B indica che B è chiave esterna di A)

Autori del progetto:

Benetazzo Antonio – Matricola 2034528

Scantamburlo Alex – Matricola 2042326

Organi (Id, Nome, Costo)

Gruppi_sanguigni (Gruppo)

Anagrafiche (Codice_fiscale, Cognome, Nome, Data_nascita)

Pazienti (Codice_fiscale→Anagrafiche.Codice_fiscale, Altezza_cm*, Peso_kg*,
Gruppo_sanguigno→Gruppi_sanguigni.Gruppo)

Personale_medico (Codice_fiscale→Anagrafiche.Codice_fiscale, Mansione, Compenso)

Compatibilità (Gruppo_donatore→Gruppi_sanguigni.Gruppo,
Gruppo_ricevente→Gruppi_sanguigni.Gruppo)

Assicurazioni (Cliente→Anagrafiche.Codice_fiscale, Massimale, Compagnia_assicurativa)

Trapianti (Id, Ricevente→Pazienti.Codice_fiscale, Donatore→Pazienti.Codice_fiscale,
Organo→Organi.Id)

Equipe_chirurgica (Trapianto→Trapianti.Id, Membro_equipe→Personale_medico.Codice_fiscale)

Lista_organ (Paziente→Pazienti.Codice_fiscale, Organo→Organi.Id, Data_ora, Priorità)

Query e indici

Query

Di seguito verranno presentate delle query significative, volte a mostrare alcune possibili interrogazioni sulla base di dati implementata.

- 1) Per ogni trapianto effettuato o programmato mostrare i codici fiscali di tutte le coppie di pazienti riceventi e donatori aventi gruppi sanguigni non compatibili.

```
SELECT errori.ricevente, riceventi.gruppo_sanguigno AS gruppo_ricevente, errori.donatore, donatori.gruppo_sanguigno AS gruppo_donatore
FROM (SELECT trapianti.ricevente, trapianti.donatore
FROM trapianti, pazienti AS riceventi, pazienti AS donatori
EXCEPT (SELECT donatori.donatore, riceventi.ricevente
FROM (SELECT pazienti.codice_fiscale AS donatore, pazienti.gruppo_sanguigno AS gruppo_donatore
FROM pazienti) AS donatori, compatibilita,
(SELECT pazienti.codice_fiscale AS ricevente, pazienti.gruppo_sanguigno AS gruppo_ricevente
FROM pazienti) AS riceventi
WHERE donatori.gruppo_donatore = compatibilita.gruppo_donatore
AND riceventi.gruppo_ricevente = compatibilita.gruppo_ricevente) AS errori, pazienti AS riceventi, pazienti AS donatori
WHERE errori.ricevente = riceventi.codice_fiscale
AND errori.donatore = donatori.codice_fiscale;
```

Questa query rappresenta un vincolo non esprimibile in SQL, non dovrebbe mai restituire alcun risultato.

```
Inserisci il valore: 0
ricevente | gruppo_ricevente | donatore | gruppo_donatore |
```

Autori del progetto:

Benetazzo Antonio – Matricola 2034528

Scantamburlo Alex – Matricola 2042326

- 2) Per ogni organo mostrare il costo medio aggiuntivo riferito allo staff medico necessario per effettuare il trapianto (media relativa ai trapianti passati o programmati).

```
SELECT organi.nome AS organo, AVG(equipe.costo_equipe) AS costo_medio_equipe
FROM (SELECT equipe_chirurgica.trapianto, SUM(personale_medico.compenso) AS costo_equipe
      FROM equipe_chirurgica, personale_medico
      WHERE personale_medico.codice_fiscale = equipe_chirurgica.membro_equipe
      GROUP BY (trapianto)) AS equipe, trapianti, organi
WHERE trapianti.id = equipe.trapianto
AND trapianti.organo = organi.id
GROUP BY organi.nome;
```

Inserisci il valore: 1

organo	costo_medio_equipe
Muscoli	4050
Midollo Osseo	1100
Fegato	4500
Pelle	5020
Cuore	1500
Polmoni	4200
Pancreas	4700
Occhi	6300
Cervello	5833.333333333333
Reni	3260

La query può essere ottimizzata attraverso l'utilizzo di un indice. Essendo una query molto utilizzata, il costo di tenere aggiornato l'indice e il ritardo negli inserimenti sarebbe giustificato dal risparmio sulle numerose interrogazioni effettuate.

```
CREATE INDEX idx_massimale ON assicurazioni (massimale);
```

- 3) Costo complessivo di un trapianto a carico del paziente, considerata quindi l'eventuale copertura assicurativa.

```
SELECT CASE
  WHEN costo_trapianto.massimale IS NULL
  THEN costo_trapianto.costo_trapianto
  WHEN costo_trapianto.costo_trapianto > costo_trapianto.massimale
  THEN costo_trapianto.costo_trapianto - costo_trapianto.massimale
  ELSE 0
END
AS costo_paziente
FROM (SELECT costo_trapianto.paziente, costo_trapianto.costo_trapianto, assicurazioni.massimale
      FROM (SELECT costo_staff.paziente, organi.costo + costo_staff.costo_equipe AS costo_trapianto
            FROM (SELECT compenso_equipe.paziente, compenso_equipe.organo, SUM(compenso) AS costo_equipe
                  FROM (SELECT equipe.paziente, equipe.membro_equipe, personale_medico.compenso, equipe.organo
                        FROM (SELECT trapianti.ricevente AS paziente, equipe_chirurgica.membro_equipe, trapianti.organo
                              FROM equipe_chirurgica
                              JOIN trapianti
                              ON equipe_chirurgica.trapianto = trapianti.id
                              WHERE trapianti.id = 2) -- ID trapianto parametrico
                        AS equipe, personale_medico
                        WHERE equipe.membro_equipe = personale_medico.codice_fiscale)
                        AS compenso_equipe
                        GROUP BY (compenso_equipe.paziente, compenso_equipe.organo)) AS costo_staff, organi
                  WHERE organi.id = costo_staff.organo) AS costo_trapianto
            LEFT JOIN assicurazioni
            ON costo_trapianto.paziente = assicurazioni.cliente) AS costo_trapianto;
```

Si tratta di una query parametrica, è quindi possibile, da programma, indicare come parametro l'id di un trapianto per ricevere il dato relativo al trapianto indicato.

Autori del progetto:

Benetazzo Antonio – Matricola 2034528

Scantamburlo Alex – Matricola 2042326

```
Inserisci il valore: 2

Comandi:
    s <I>          // per selezionare il record.
    p <I> <J>      // per stampare i record con indice compreso tra I e J.
Esempi:
    's 0' seleziona l'ultimo elemento.
    'p 0 36' stampa tutta la lista.
Input: s 12

costo_paziente      |
19800                |
```

- 4) Mostrare l'elenco dei trapianti effettuati o programmati da parte di un paziente in qualità di ricevente.

```
SELECT organi.nome AS organo, trapianti_paziente.data, anagrafiche.cognome AS cognome_donatore, anagrafiche.nome AS nome_donatore
FROM (SELECT trapianti.organo, trapianti.data, trapianti.donatore
      FROM trapianti
      WHERE ricevente = '01234567890') AS trapianti_paziente, organi, anagrafiche -- CF ricevente parametrico
WHERE organi.id = trapianti_paziente.organo
AND anagrafiche.codice_fiscale = trapianti_paziente.donatore;
```

Si tratta di una query parametrica, è quindi possibile, da programma, indicare come parametro il codice fiscale di un paziente per avere l'elenco dei trapianti di quel paziente.

```
Inserisci il valore: 3

Comandi:
    s <I>          // per selezionare il record.
    p <I> <J>      // per stampare i record con indice compreso tra I e J.
Esempi:
    's 0' seleziona l'ultimo elemento.
    'p 0 54' stampa tutta la lista.
Input: s 12

organo                | data                | cognome_donatore    | nome_donatore       |
Fegato                | 2024-03-07         | ugo                  | berlusconi           |
```

- 5) Mostrare l'elenco dei trapianti per i quali ha operato un'equipe chirurgica avente un tirocinante senza nessuno strutturato con la stessa mansione che potesse supervisionarlo e indicare la mansione che viola tale vincolo.

```
SELECT DISTINCT trapianti.id AS id_trapianto, personale_medico.mansione
FROM trapianti, personale_medico, equipe_chirurgica
WHERE trapianti.id = equipe_chirurgica.trapianto
AND equipe_chirurgica.membro_equipe = personale_medico.codice_fiscale
EXCEPT (SELECT DISTINCT trapianti_equipe1.id AS id_trapianto, trapianti_equipe1.mansione
FROM (SELECT trapianti.id, personale_medico.mansione, personale_medico.compenso
FROM trapianti, personale_medico, equipe_chirurgica
WHERE trapianti.id = equipe_chirurgica.trapianto
AND personale_medico.codice_fiscale = equipe_chirurgica.membro_equipe) AS trapianti_equipe1
JOIN (SELECT trapianti.id, personale_medico.mansione, personale_medico.compenso
FROM trapianti, personale_medico, equipe_chirurgica
WHERE trapianti.id = equipe_chirurgica.trapianto
AND personale_medico.codice_fiscale = equipe_chirurgica.membro_equipe) AS trapianti_equipe2
ON trapianti_equipe1.id = trapianti_equipe2.id
AND trapianti_equipe1.mansione = trapianti_equipe2.mansione
AND ((trapianti_equipe1.compenso = 0 AND trapianti_equipe2.compenso <> 0)
OR trapianti_equipe1.compenso <> 0)
ORDER BY id_trapianto)
ORDER BY id_trapianto;
```

Questa query rappresenta un vincolo non esprimibile in SQL, non dovrebbe mai restituire alcun risultato.

Inserisci il valore: 4

id_trapianto	mansione

- 6) Per ogni organo disponibile e per ogni paziente in lista d'attesa mostrare tutti i possibili abbinamenti ricevente-donatore con gruppo sanguigno compatibile e ordinare i risultati in base ad organo, priorità e data inserimento in lista d'attesa.

```
SELECT abbinamenti.ricevente, organi.nome AS organo, abbinamenti.priorita, abbinamenti.data_ora, abbinamenti.donatore
FROM (SELECT *
FROM (SELECT lista_organi.paziente AS ricevente, pazienti.gruppo_sanguigno AS gruppo_ricevente, lista_organi.organo AS organo_richiesto,
lista_organi.priorita, lista_organi.data_ora
FROM lista_organi, pazienti
WHERE priorita IS NOT NULL
AND lista_organi.paziente = pazienti.codice_fiscale
ORDER BY lista_organi.priorita DESC, lista_organi.data_ora ASC) AS richieste
JOIN compatibilita
ON richieste.gruppo_ricevente = compatibilita.gruppo_ricevente
JOIN (SELECT lista_organi.paziente AS donatore, pazienti.gruppo_sanguigno AS gruppo_donatore, lista_organi.organo AS organo_disponibile
FROM lista_organi, pazienti
WHERE priorita IS NULL
AND lista_organi.paziente = pazienti.codice_fiscale) AS disponibile
ON disponibile.gruppo_donatore = compatibilita.gruppo_donatore
WHERE richieste.organico_richiesto = disponibile.organico_disponibile) AS abbinamenti, organi
ORDER BY organo ASC, priorita DESC, data_ora ASC, ricevente ASC;
```

Inserisci il valore: 5

ricevente	gruppo_ricevente	organo	priorita	data_ora	donatore	gruppo_donatore
brlmc16r21g224z	0-	Midollo Osseo	1	2025-05-17 00:00:00	brlms80d16f241y	0-
grbfm77p21g224z	A+	Midollo Osseo	1	2025-07-16 00:00:00	brlms80d16f241y	0-
brmc123d11f241y	0-	Muscoli	3	2025-04-26 00:00:00	brlms80d16f241y	0-
brlbr60h21g224z	AB+	Occhi	7	2025-03-22 00:00:00	bncmrc12e24f241y	A+
brlbr60h21g224z	AB+	Occhi	7	2025-03-22 00:00:00	brlms80d16f241y	0-
vrdfm94b24g224z	AB-	Occhi	7	2025-06-05 00:00:00	brlms80d16f241y	0-
rnzfrn62118g224z	AB+	Occhi	4	2024-04-20 00:00:00	brlms80d16f241y	0-
rnzfrn62118g224z	AB+	Occhi	4	2024-04-20 00:00:00	bncmrc12e24f241y	A+
brnma20r05f241y	0+	Pelle	1	2025-01-27 00:00:00	brlms80d16f241y	0-
vrdfm94b24g224z	0-	Polmoni	4	2025-07-14 00:00:00	brlms80d16f241y	0-

La query può essere ottimizzata attraverso l'utilizzo di due indici. Essendo non solo una query molto utilizzata ma forse una delle interrogazioni più frequenti, il costo di tenere aggiornato l'indice e il ritardo negli inserimenti sarebbe enormemente giustificato dal risparmio sulle numerose interrogazioni effettuate. Per questi indici abbiamo preferito utilizzare le funzioni hash,

Autori del progetto:

Benetazzo Antonio – Matricola 2034528

Scantamburlo Alex – Matricola 2042326

in quanto vengono utilizzati operatori di uguaglianza, per i quali le funzioni hash hanno un effetto migliore.

```
CREATE INDEX idx_lista_attesa_pr ON lista_organismi USING hash (priorita);  
CREATE INDEX idx_lista_attesa_do ON lista_organismi USING hash (data_ora);
```

7) Dato un anno come parametro, mostrare tutti i pazienti che hanno subito (in qualità di ricevente) più di un trapianto durante quell'anno e indicare quanti.

```
SELECT anagrafiche.nome, anagrafiche.cognome, COUNT(trapianti.ricevente) AS n_trapianti  
FROM anagrafiche, trapianti  
WHERE anagrafiche.codice_fiscale = trapianti.ricevente  
AND EXTRACT(YEAR FROM trapianti.data) = 2024 -- Anno parametrico  
GROUP BY anagrafiche.nome, anagrafiche.cognome  
HAVING COUNT(trapianti.ricevente) > 1;
```

Si tratta di una query parametrica, è quindi possibile, da programma, indicare come parametro un anno a piacere per visualizzare i dati relativi all'anno indicato.

Inserisci il valore: 6

Comandi:

s <I> // per selezionare il record.

p <I> <J> // per stampare i record con indice compreso tra I e J.

Esempi:

's 0' seleziona l'ultimo elemento.

'p 0 4' stampa tutta la lista.

Input: s 4

nome	cognome	n_trapianti
bianco	lucia	2
verdi	massimiliano	2

Autori del progetto:

Benetazzo Antonio – Matricola 2034528

Scantamburlo Alex – Matricola 2042326

Codice C++

Descrizione dell'utilizzo del codice

Il codice C++ per l'esecuzione delle query consiste in un singolo file c++ standalone, che viene compilato con i comandi sotto riportati per avere un programma interattivo.

```
g++ -c codice.cpp -o codice.o -I /usr/include/postgresql -lpq  
g++ codice.o -o codice -I /usr/include/postgresql -lpq
```

Il programma è stato scritto, compilato, eseguito e testato sui computer del Paolotti, tramite connessione remota ssh (sistema operativo Linux Ubuntu). Per la compilazione è necessario aver installate le librerie c++ di Postgres, già presenti sulle nostre macchine. Per eseguire il codice è sufficiente avviare l'eseguibile 'codice'; sarà subito mostrato un menù di opzioni ordinate da -1 a 6. La prima (-1) consente di terminare il programma, mentre le opzioni da 0 a 6 sono le 7 query eseguibili nel database. Al fine del login con il Database viene definita una stringa alle righe da 35 a 39 del file **codice.cpp**, che va modificata in base alla configurazione del database importato nella macchina locale.

La struttura del programma è pigra: all'avvio del codice non viene eseguita nessuna connessione al database finché non viene selezionata una query da eseguire. Per selezionare una query basta inserire da tastiera l'indice subito a sinistra di essa.

Le query con indice 2, 3 e 6 richiedono ulteriori parametri:

2: chiede l'identificativo del trapianto del quale viene calcolato il costo a carico del paziente

3: chiede il codice fiscale di un paziente per visualizzare tutti i trapianti subiti (o programmati) da parte di quel paziente (in qualità di ricevente)

6: chiede l'anno e mostra i pazienti che hanno subito più di 1 trapianto in quell'anno

Documentazione del codice

Classi utilizzate:

```
class QSelect{  
public:  
    vector<string> headers;  
    vector<vector<string>> data;  
    QSelect();  
    friend ostream& operator<<(ostream&, const QSelect&);  
};
```

QSelect è una classe utilizzata per contenere i risultati delle query SELECT.

La classe possiede due attributi: headers (vettore di stringhe) e data (vettore di vettori di stringhe), ognuno contenente i nomi dei campi e i rispettivi valori.

Autori del progetto:

Benetazzo Antonio – Matricola 2034528

Scantamburlo Alex – Matricola 2042326

La classe possiede inoltre un overload del metodo di stampa su una ostream per semplificarne la stampa con cout.

Funzioni utilizzate:

```
PGconn * const get_conn()
```

Restituisce il puntatore della connessione. L'oggetto è statico, viene costruito una sola volta e viene restituito ad ogni chiamata di funzione.

```
QSelect select_from(const string& qstr, const vector<string>& args = {})
```

Restituisce un oggetto QSelect contenente i risultati della stringa qstr. Se la query richiede dei parametri è possibile passarli sotto forma di vettore di stringhe.

```
const vector<string>& get_record_from_qs(const QSelect& qs)
```

Funzione di astrazione per select_from nel caso siano necessari degli argomenti.

Dato un QSelect, questo corrisponde agli input possibili, la funzione attende che l'utente digiti uno dei seguenti comandi:

- p: dati <I> <J> vengono stampati tutti i parametri accettabili compresi tra I e J.
- s: dato <I> viene selezionato il parametro a indice I.

Il comando 'p' riporta la funzione in attesa dell'input, mentre 's' esce dal ciclo e ritorna il record selezionato.

Esempi:

- 'p 0 12' stampa tutti i record selezionabili con indici che vanno da 0 a 12 (compresi).
- 's 11' seleziona l'undicesimo record e lo ritorna al chiamante.

```
Comandi:
s <I>      // per selezionare il record.
p <I> <J>  // per stampare i record con indice compreso tra I e J.
Esempi:
's 0' seleziona l'ultimo elemento.
'p 0 4' stampa tutta la lista.
Input: p 0 4

Indice | date_part |
0 | 2029 |
1 | 2027 |
2 | 2024 |
3 | 2028 |
4 | 2026 |

Comandi:
s <I>      // per selezionare il record.
p <I> <J>  // per stampare i record con indice compreso tra I e J.
Esempi:
's 0' seleziona l'ultimo elemento.
'p 0 4' stampa tutta la lista.
Input: s 2

nome | cognome | n_trapianti |
```

Autori del progetto:

Benetazzo Antonio – Matricola 2034528

Scantamburlo Alex – Matricola 2042326