

# Progetto basi di dati

Guccione Riccardo IN0301143

May 2023

## 1 Introduction

Il progetto nasce dalla richiesta di un ente esterno di creare ed implementare un database relazionale per la gestione ed organizzazione dei casi clinici di un medico di medicina generale. Sono state condotte varie interviste al professionista interessato, atte a cercare di creare l'infrastruttura più adatta alle sue esigenze.

Viene inoltre richiesto lo sviluppo di un applicazione android con database locale (per motivi di sicurezza) che possa andare a sostituire il file pdf che veniva usato precedentemente per gestire i pazienti.

Nella sezione successiva sono illustrate ed organizzate le nozioni principali che sono emerse durante le suddette interviste.

## 2 Richieste

La prima informazione fornita è riguardante al file pdf della cartella clinica utilizzato precedentemente. Nel documento notiamo che sono indicate le informazioni principali del paziente:

- Nome.
- le allergie.
- Cognome.
- le patologie pregresse rilevanti.
- data di nascita.
- tutte le patologie precedenti con le date delle diagnosi.
- codice fiscale.
- le date e gli elenchi dei ricoveri.
- codice sanitario.
- gli esami strumentali e di laboratorio.
- eventuali esenzioni.
- un diario per l'organizzazione della terapia.
- le patologie in atto.
- la terapia in atto.

Il secondo set di informazioni riguardano alcune precisazioni ricavate dall'intervista, che è necessario aggiungere. Cioè:

- l'indirizzo di casa.
- aggiungere una sezione per l'esame specialistico.
- il genere.
- gli ospedali in cui sono avvenuti i ricoveri.
- le date di stop delle terapie precedenti.
- le informazioni sugli operatori sanitari.
- il dosaggio per le terapie.
- se un paziente è seguito da caregiver per una terapia.
- il costo degli esami.

### 3 Azioni al database

Infine si è parlato di quali richieste poter fare al database. Le richieste più immediate sono:

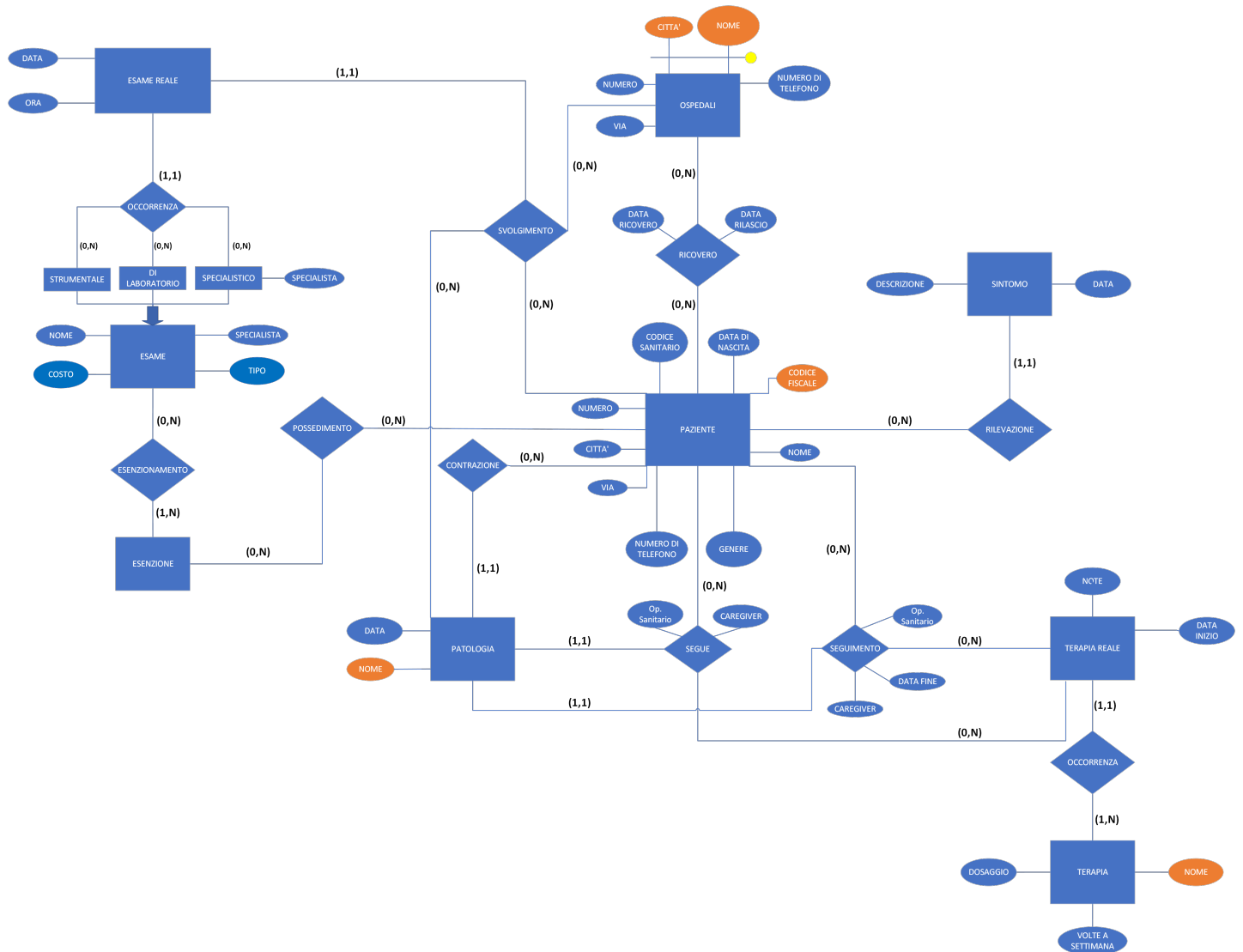
- avere la possibilità di visualizzare ogni attributo associato al paziente.
- l'eventuale ospedale di ricovero e dove si trova.
- i ricoveri passati.
- la visualizzazione delle patologie.
- un elenco degli esami passati e futuri.
- un elenco delle terapie passate ed attuali.
- bisogna poter vedere per quali esami un paziente possiede l'esenzione.
- un elenco dei pazienti con un set prefissato di patologie.

Mentre le richieste più complesse:

- Il professionista chiede di poter visualizzare i pazienti hanno contratto una certa patologia nell'ultimo mese. (1a volta al mese)
- il professionista chiede di poter visualizzare un elenco delle terapie più efficaci di un cert'anno, cioè quali hanno una durata media minore. (1a volta al l'anno)
- il professionista chiede di poter vedere se in un certo ospedale sono programmati più di 100 esami dei propri pazienti. (1a volta a settimana)
- il professionista chiede di poter stilare una lista dei pazienti seguiti da caregiver e da operatori sanitari per poter segnalare la sovrapposizione. (1a volta al mese)
- il professionista chiede di avere un recap annuale dei pazienti che hanno terminato una certa terapia durante un cert'anno. (1a volta l'anno)

## 4 Diagramma Entità-Relazioni

Sfruttando le informazioni ricavate dalle interviste è quindi possibile costruire un primo diagramma E-R



## 5 Analisi dei dati

In questa sezione viene illustrata l'analisi tabulare delle entità e relazioni.

### 5.1 Eliminazione attributi composti

Per prima cosa eliminiamo gli attributi composti; ne identifichiamo 2 con gli indirizzi del Paziente e dell'Ospedale. Per rimuoverli è sufficiente collegare le specificazioni direttamente con l'entità.

### 5.2 Definizione attributi primari

Definiamo ora gli identificatori:

- Per paziente: il Codice sanitario.
- Per il sintomo: aggiungiamo l'Id. Sintomo.
- Per l'Ospedale: La coppia Nome e Città.
- Per la Patologia: il Nome.
- Per l'Esame: il Nome.
- Per l'Esenzione: il Nome.
- Per la Terapia: il Nome.
- Per l'Esame reale: aggiungiamo l'Id Esame.
- Per la Terapia reale: aggiungiamo l'Id Terapia.

### 5.3 Tabelle di analisi

ENTITA'	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Paziente	Pazienti del professionista	Nome, Codice sanitario, Codice Fiscale Data di nascita, Numero di telefono, Esenzione, Città, Via, Numero	Codice sanitario
Sintomo	Sintomi di vario genere	Id. Sintomo, Descrizione sintomo, Data di rilevazione	Id. Sintomo
Ospedale	Ospedali della zona di lavoro	Nome, Città, Via, Numero, Numero di telefono	Nome e Città
Patologia	Patologie di vario genere	Nome, Data di rilevazione	Nome
Esame	Esami prescrivibili	Nome, Descrizione, Costo	Nome
Esenzione	Le varie esenzioni	Nome	Nome
Terapia	Terapie prescrivibili	Nome, Dosaggio, Volte a settimana	Nome
Esame reale	Esami programmati e passati	Data ed ora	Id. Esame
Terapia reale	Terapie programmate e passate	Id. terapia, Data di inizio, Note	Id. terapia

RELAZIONE	DESCRIZIONE	COMPOSIZIONE	ATTRIBUTI
Rilevazione	Rilevazione del sintomo di un paziente	Sintomo, Paziente	Nessuno
Ricovero	Ricovero di un paziente in un Ospedale	Paziente, Ospedale	Data ricovero, Data rilascio
Contrazione	La contrazione del paziente di una patologia	Paziente, Patologia	Nessuno
Esenzionamento	Se un esame ha una certa esenzione	Esame, Esenzione	Nessuno
Occorrenza	L'occorrenza di un esame nel tempo	Esame, Esame reale	Nessuno
Svolgimento	Lo svolgimento dell'esame di un paziente	Esame reale, Paziente, Ospedale, Patologia	Nessuno
Occorrenza	L'occorrenza di una terapia nel tempo	Terapia, Terapia reale	Nessuno
Seguimento	Lo svolgimento della terapia attuale	Terapia reale, Paziente, Patologia	Caregiver, Operatore Sanitario
Seguimento	Lo svolgimento della terapia nel passato, giunta al termine	Terapia reale, Paziente, Patologia	Caregiver, Operatore Sanitario, Data fine
Possedimento	La possessione di una o più esenzioni da parte di un paziente	Paziente, Esenzione	Nessuno

## 6 Vincoli non esprimibili

Illustriamo in breve i vincoli non esprimibili :

- I pazienti non possono avere più esami alla stessa ora dello stesso giorno
- i pazienti non possono essere ricoverati in più ospedali allo stesso tempo ma possono essere ricoverati in diversi ospedali nel tempo

## 7 Tabella dei volumi

Nel paragrafo viene data la tabella coi volumi approssimati delle entità e delle relazioni.

RELAZIONE	TIPO	VOLUME
Paziente	E	1500
Sintomo	E	15000
Ospedale	E	20
Patologia	E	500
Esame	E	3000
Esenzione	E	200
Terapia	E	15000
Esame reale	E	200
Terapia reale	E	10000
Rilevazione	R	150000
Ricovero	R	10000
Contrazione	R	15000
Esenzionamento	R	3000
Possedimento	R	1000
Occorrenza	R	30000
Svolgimento	R	30000
Occorrenza	R	15000
Seguimento	R	500
Seguimento	R	14500

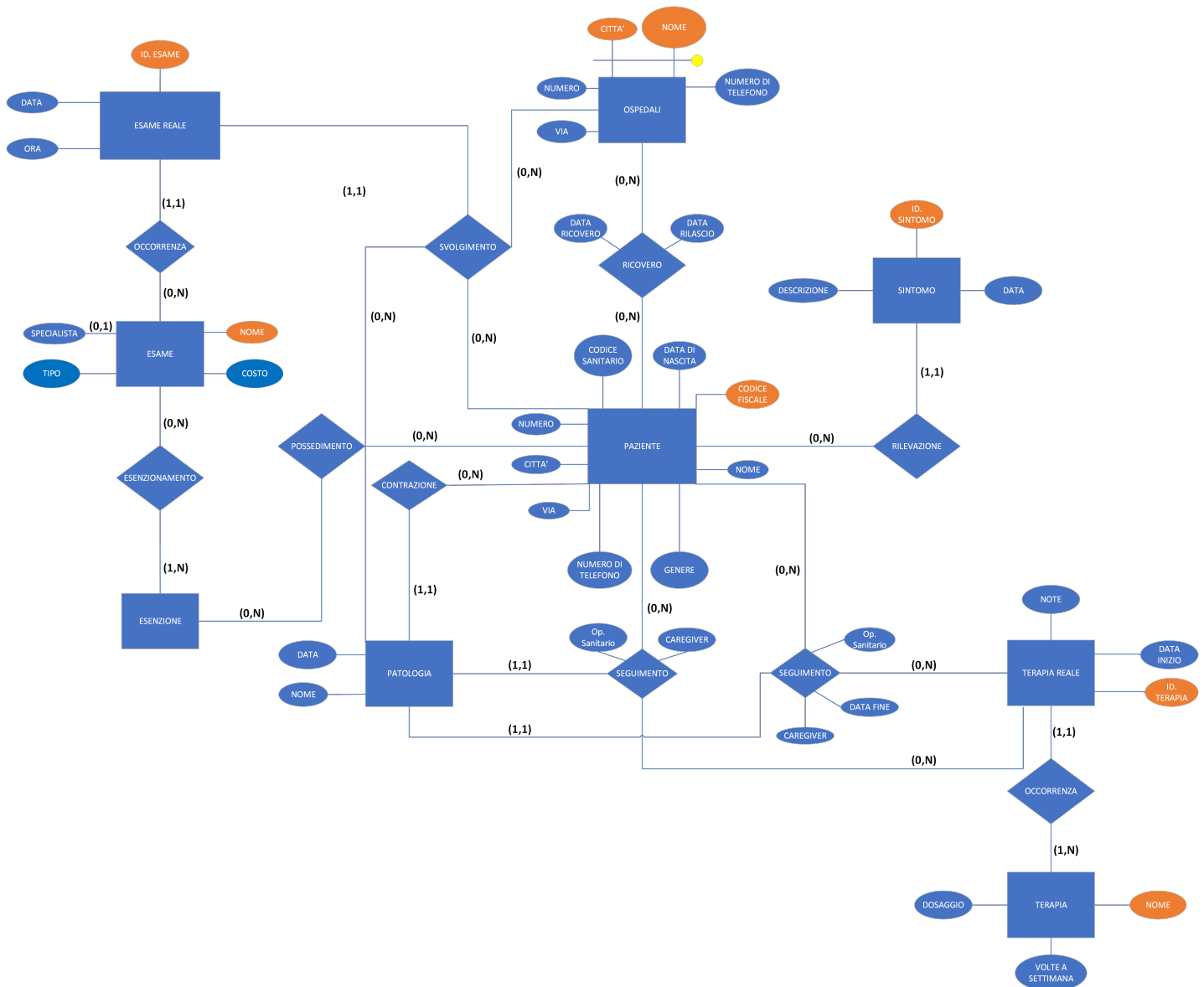
## 8 Analisi delle generalizzazioni e ridondanze e ristrutturazione schema E-R

### 8.1 Analisi generalizzazione e ridondanze

Analizziamo le generalizzazioni, si vede che ne è presente una associata all'entità padre Esami. Bisogna quindi eliminare la generalizzazione, si è scelto di aggregarla al padre, poichè, nonostante la generalizzazione sia totale e quindi c'è la possibilità che vengano a crearsi numerosi valori nulli (maggiore uso di memoria), l'accesso alle occorrenze dell'entità padre è logicamente lo stesso a quelle delle occorrenze delle entità figlie, ed inoltre verrebbero a crearsi diverse entità per l'occorrenza degli esami reali, per cui non è preferibile l'aggregazione ai figli.

Per quanto riguarda le ridondanze, non ne sono state rilevate; infatti tutti gli attributi sono unici e non ricavabili, ed i cicli non permettono di ricavare le singole relazioni, se una per volta venissero rimosse.

### 8.2 Ristrutturazione dello schema E-R



## 9 Schema logico

### 9.1 Costruzione delle tabelle

Voglio passare dallo schema E-R alle tabelle che andranno poi a formare il database relazionale, per fare ciò guardo le cardinalità nelle relazioni e costruisco di conseguenza il numero di tabelle adeguato.

- Per la relazione "Rilevazione", vedo che è uno a molti, per cui costruisco 2 tabelle, una per i Sintomi e una per i Pazienti; con chiave esterna Il codice Fiscale sulla tabella Sintomi.
- Per la relazione "Ricovero", vedo che la relazione è molti a molti, ma ho già la tabella Pazienti, è quindi necessario costruire 2 ulteriori tabelle una Ricoveri e una Ospedali; con chiavi esterne Codice Fiscale e nome ospedale sulla tabella Ricoveri .
- Per la relazione "Contrazione", anche qui la relazione è uno a molti, ho già la tabella Pazienti, quindi creo una tabella Patologie contratte; con chiave esterna Codice Fiscale paziente.
- Per la relazione "Occorrenza" dell'Esame, anche qui la relazione è uno a molti, quindi creo una tabella Descrizione Esami e una tabella Esami; con chiave esterna il Nome dell'Esame sulla tabella Esami.
- Per la relazione "Svolgimento", vedo che la relazione è uno a molti, quindi non devo creare ulteriori tabelle ma uso le tabelle Pazienti, Ospedali e Patologia; con chiave esterna rispettivamente Codice Fiscale, Nome Ospedale e Id. Patologia, sulla tabella Esami.
- Per la relazione "Occorrenza" della Terapia, anche qui la relazione è uno a molti, quindi creo una tabella Descrizione Terapie e una tabella Terapie; con chiave esterna il Nome della Terapia sulla tabella Terapie.
- Per la relazione "Segue" e la relazione "Seguiva", sono entrambe uno a molti ma hanno degli attributi, che posso aggiungere la tabella Terapie, in tal modo non devo creare nuove tabelle e posso usare le tabelle Pazienti, Patologie; con chiave esterna rispettivamente Codice Fiscale e Id. Patologia, sulla tabella Terapie.
- Per la relazione "Esenzionamento", vedo che è una relazione molti a molti, per cui creo la tabella Esenzioni; con chiavi esterne il Nome dell'Esame e l'identificativo dell'esenzione e la tabella Registro esenzioni.
- Per la relazione "Possedimento", vedo che è una relazione molti a molti, per cui sfruttando la tabella Registro esenzioni creo un'ulteriore tabella che chiamo Esenzioni pazienti; con chiavi esterne il paziente come Codice Fiscale e l'identificativo dell'esenzione.

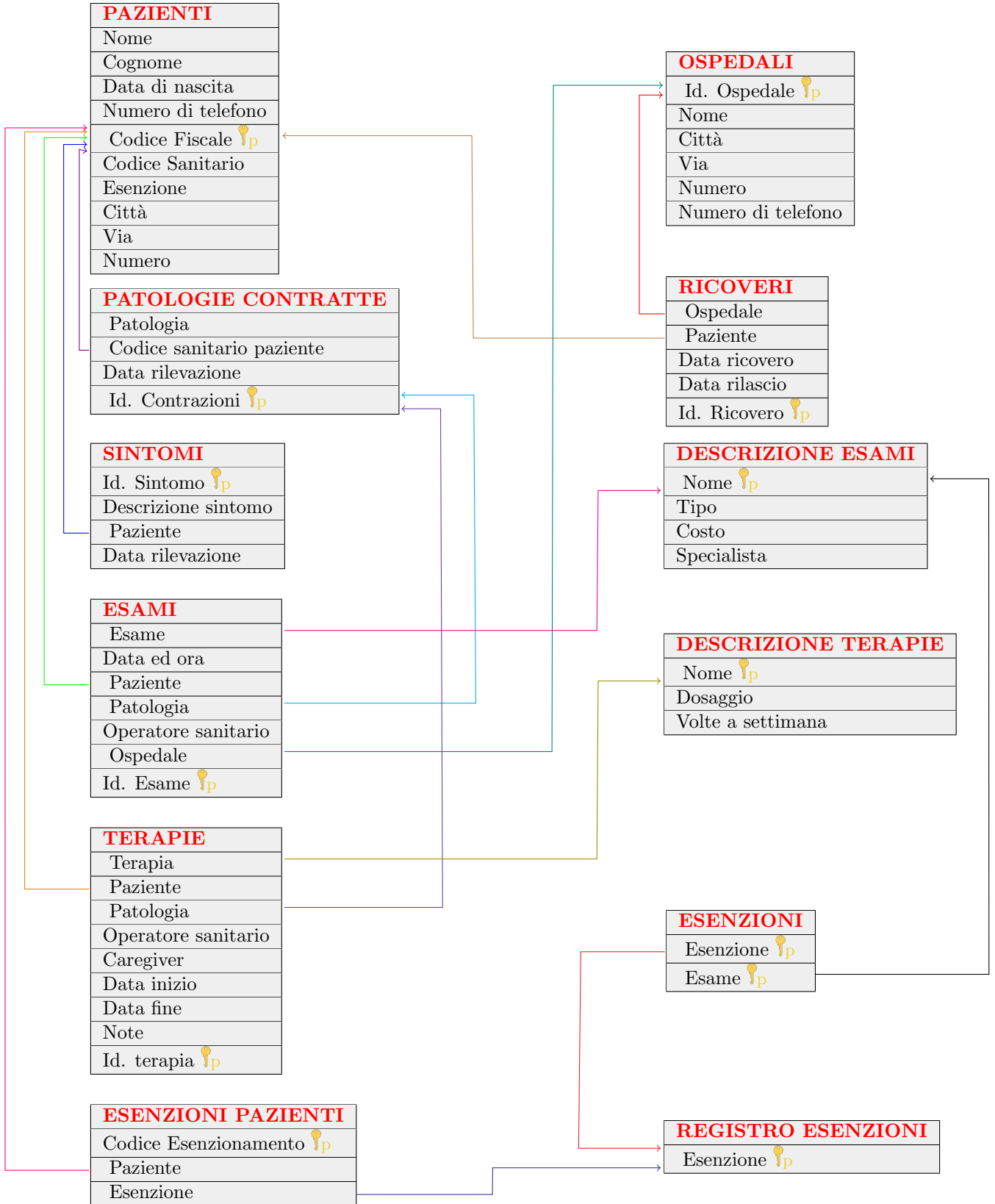
### 9.2 Scelta delle chiavi primarie

Per completare la progettazione del database relazionale scegliamo le chiavi primarie delle tabelle. Ricordo che è necessario che il valore non sia mai nullo e che non si ripeta.

- Per la tabella Paziente posso utilizzare il Codice Fiscale.
- Per la tabella Ospedali è meglio aggiungere un identificatore univoco, che chiamo Id. Ospedale, per evitare problematiche di utilizzo (più sedi o rare omonimie).
- Per la tabella Ricoveri ho bisogno di aggiungere una colonna per il codice identificativo che chiamo Id. ricovero.
- Per la tabella Patologie contratte uso la colonna Id. Patologia.
- Per la tabella Descrizione Esami posso utilizzare la colonna Nome.
- Per la tabella Esami uso la colonna Id. Esame.
- Per la tabella Descrizione Terapie posso usare la colonna Nome.
- Per la tabella Terapie uso la colonna Id. Terapia.
- Per la tabella Esenzioni la coppia Esenzione ed Esame è chiave primaria.
- Per la tabella Registro esenzioni la colonna Nome.
- Per la tabella Esenzioni pazienti creo la colonna Codice esenzionamento



### 9.3 Tabelle



## 10 Riduzione in forma Normale

### 10.1 1<sup>a</sup> forma Normale

La base di dati è in prima forma normale visto che ogni colonna è composta da valori atomici.

### 10.2 2<sup>a</sup> forma Normale

La base di dati è in seconda forma normale perchè è in prima forma normale e tutti gli attributi dipendono dall'interezza della chiave primaria.

### 10.3 3<sup>a</sup> forma Normale

La base di dati è in terza forma normale perchè è in seconda forma normale e tutti gli attributi dipendono solo dalla chiave primaria.

## 10.4 Sviluppo dell'applicazione android

Si è poi passati allo sviluppo dell'applicazione android richiesta; si è deciso di usare SQLITE per la creazione e gestione del database, poichè dovendo essere riempito di dati possibilmente sensibili, si è ritenuto che fosse opportuno limitarne l'uso in locale.

L'applicazione è in via di sviluppo ma disponibile alla repository con link seguente

[https://github.com/RickGuc/Progetto\\_schedario\\_app](https://github.com/RickGuc/Progetto_schedario_app)

Tuttavia è già disponibile una versione con delle funzioni principali e con la possibilità di vedere i risultati di alcune query predefinite.

L'apk in questione si trova invece nella repository del progetto, alla pagina:

[https://github.com/RickGuc/Progetto\\_basi\\_di\\_dati](https://github.com/RickGuc/Progetto_basi_di_dati)

In più è anche presente un database "fac-smile" per testare l'applicazione che è però necessario aggiungere alla directory del database dell'app per l'utilizzo nel proprio dispositivo.

Il database può inoltre essere aperto con l'ausilio dell'applicazione desktop sqlite browser, scaricabile al link

<https://sqlitebrowser.org/dl/>

In futuro è prevista la continuazione dello sviluppo per mezzo di flutter, ma sarà necessario comprendere il linguaggio e convertire il lavoro fatto in java ed xml in dart.

## 10.5 Lista Queries

### 10.5.1 creazione tabelle

```
CREATE TABLE TABELLA_PAZIENTE (NOME VARCHAR, COGNOME VARCHAR, CODICEFISCALE CHAR(16) PRIMARY KEY, CODICESANITARIO CHAR(16), DATA DATE, CITTA VARCHAR, VIA VARCHAR, NUMERO CHAR(4), GENERE VARCHAR(9));
```

```
CREATE TABLE TABELLA_OSPEDALI (ID INT PRIMARY KEY AUTOINCREMENT, NOME VARCHAR, CITTA VARCHAR, VIA VARCHAR, NUMERO CHAR(4), NUMERODITELEFONO VARCHAR);
```

```
CREATE TABLE TABELLA_PATOLOGIE_CONTRATTE (ID INT PRIMARY KEY AUTOINCREMENT, NOME VARCHAR, PAZIENTE CHAR(16), DATA_RILEVAZIONE DATE, FOREIGN KEY (PAZIENTE) REFERENCES TABELLA_PAZIENTE (CODICEFISCALE));
```

```
CREATE TABLE TABELLA_RICOVERI (ID INT PRIMARY KEY AUTOINCREMENT, OSPEDALE INT, PAZIENTE CHAR(16), DATA_RICOVERO DATE, DATA_RILASCIO DATE, FOREIGN KEY (PAZIENTE) REFERENCES TABELLA_PAZIENTE (CODICEFISCALE), FOREIGN KEY (OSPEDALE) REFERENCES TABELLA_OSPEDALI (ID));
```

```
CREATE TABLE TABELLA_SINTOMI (ID INT PRIMARY KEY AUTOINCREMENT, DESCRIZIONE_SINTOMO VARCHAR, DATA_RILEVAZIONE DATE, PAZIENTE CHAR(16), FOREIGN KEY (PAZIENTE) REFERENCES TABELLA_PAZIENTE (CODICEFISCALE));
```

```
CREATE TABLE TABELLA_DESCRIZIONE_ESAMI (NOME VARCHAR PRIMARY KEY, TIPO VARCHAR, COSTO SMALLINT, SPECIALISTA VARCHAR);
```

```
CREATE TABLE TABELLA_DESCRIZIONE_TERAPIE (NOME VARCHAR PRIMARY KEY, DOSAGGIO VARCHAR, VOLTE_A_SETTIMANA TINYINT);
```

```
CREATE TABLE TABELLA_ESAMI (ID INT PRIMARY KEY AUTOINCREMENT, ESAME VARCHAR, DATA DATE, ORA TIME, PAZIENTE CHAR(16), PATOLOGIA INT, OPERATORE_SANITARIO VARCHAR, OSPEDALE INT, FOREIGN KEY (ESAME) REFERENCES TABELLA_DESCRIZIONE_ESAMI (NOME), FOREIGN KEY (PAZIENTE) REFERENCES TABELLA_PAZIENTE (CODICEFISCALE), FOREIGN KEY (PATOLOGIA) REFERENCES TABELLA_PATOLOGIE_CONTRATTE (ID));
```

```
CREATE TABLE TABELLA_TERAPIE (ID INT PRIMARY KEY AUTOINCREMENT, TERAPIA VARCHAR, PAZIENTE CHAR(16), PATOLOGIA INT, OPERATORE_SANITARIO VARCHAR, CAREGIVER VARCHAR, DATA_INIZIO DATE, DATA_FINE DATE, NOTE VARCHAR, FOREIGN KEY (TERAPIA) REFERENCES TABELLA_DESCRIZIONE_TERAPIE (NOME), FOREIGN KEY (PAZIENTE) REFERENCES TABELLA_PAZIENTE (CODICEFISCALE), FOREIGN KEY (PATOLOGIA) REFERENCES TABELLA_PATOLOGIE_CONTRATTE (ID), FOREIGN KEY (OSPEDALE) REFERENCES TABELLA_OSPEDALI (ID));
```

```
CREATE TABLE TABELLA_ESENZIONI_PAZIENTI (CODICE INT PRIMARY KEY AUTOINCREMENT, PAZIENTE CHAR(16), ESENZIONE VARCHAR, FOREIGN KEY (PAZIENTE) REFERENCES TABELLA_PAZIENTE (CODICEFISCALE), FOREIGN KEY (ESENZIONE) REFERENCES TABELLA_REGISTRO_ESENZIONI (ESENZIONE));
```

```
CREATE TABLE TABELLA_ESENZIONI (ESENZIONE VARCHAR PRIMARY KEY, ESAME VARCHAR PRIMARY KEY, FOREIGN KEY (ESAME) REFERENCES TABELLA_DESCRIZIONE_ESAMI (NOME), FOREIGN KEY (ESENZIONE) REFERENCES TABELLA_REGISTRO_ESENZIONI (ESENZIONE));
```

```
CREATE TABLE TABELLA_REGISTRO_ESENZIONI (ESENZIONE VARCHAR PRIMARY KEY);
```

### 10.5.2 Altre Queries

```
SELECT * FROM TABELLA_PAZIENTE;
```

```
SELECT NOME FROM TABELLA_PAZIENTE (uguale per ogni attributo necessario nella tabella paziente);
```

```
SELECT p.NOME, p.COGNOME, o.NOME, r.DATA_RICOVERO, o.CITTA, o.VIA, o.NUMERO FROM  
TABELLA_PAZIENTE p INNER JOIN TABELLA_RICOVERI r ON p.CODICEFISCALE = r.PAZIENTE  
INNER JOIN TABELLA_OSPEDALI o ON o.ID = r.OSPEDALE WHERE r.DATA_RILASCIO IS NULL;
```

```
SELECT p.NOME, p.COGNOME, o.NOME, r.DATA_RICOVERO, r.DATA_RILASCIO FROM TABELLA_PAZIENTE  
p INNER JOIN TABELLA_RICOVERI r ON p.CODICEFISCALE = r.PAZIENTE INNER JOIN TABELLA_OSPEDALI  
o ON o.ID = r.OSPEDALE WHERE p.CODICEFISCALE = (SELECT CODICEFISCALE FROM TABELLA_PAZIENTE  
WHERE NOME = ? AND COGNOME = ?);
```

```
SELECT p.NOME, p.COGNOME, t.PATOLOGIA FROM TABELLA_PAZIENTE p INNER JOIN  
TABELLA_PATOLOGIE_CONTRATTE t ON p.CODICEFISCALE = t.PAZIENTE ORDER BY COG-  
NOME DESC;
```

```
SELECT p.NOME, p.COGNOME, r.NOME FROM TABELLA_PAZIENTE p INNER JOIN  
TABELLA_PATOLOGIE_CONTRATTE r ON p.CODICEFISCALE = r.PAZIENTE WHERE p.CODICEFISCALE  
= (SELECT CODICEFISCALE FROM TABELLA_PAZIENTE WHERE NOME = ? AND COGNOME  
= ?);
```

```
SELECT p.NOME, p.COGNOME, e.ESAME, e.DATA, e.ORA FROM TABELLA_PAZIENTE p INNER  
JOIN TABELLA_ESAMI e ON p.CODICEFISCALE = e.PAZIENTE;
```

```
SELECT p.NOME, p.COGNOME, t.TERAPIA, t.DATA_INIZIO, t.DATA_FINE FROM TABELLA_PAZIENTE  
p INNER JOIN TABELLA_TERAPIE t ON p.CODICEFISCALE = t.PAZIENTE;
```

```
SELECT p.NOME, p.COGNOME, e.ESENZIONE FROM TABELLA_PAZIENTE p INNER JOIN  
TABELLA_ESENZIONI_PAZIENTI e ON p.CODICEFISCALE = e.PAZIENTE ORDER BY COGNOME  
DESC;
```

```
SELECT p.NOME, p.COGNOME, e.ESAME FROM TABELLA_PAZIENTE p INNER JOIN  
TABELLA_ESENZIONI_PAZIENTI t ON p.CODICEFISCALE = t.PAZIENTE INNER JOIN  
TABELLA_REGISTRO_ESENZIONI r ON t.ESENZIONE = r.ESENZIONE INNER JOIN TABELLA_ESENZIONI  
e ON r.ESENZIONE = e.ESENZIONE;
```

```
SELECT * FROM TABELLA_PAZIENTE p WHERE EXISTS (SELECT 1 FROM TABELLA_DESCRIZIONE_PATOLOGIE  
a INNER JOIN TABELLA_PAZIENTE p ON p.CODICEFISCALE = a.PAZIENTE WHERE a.PATOLOGIA  
IN ('GASTRITE','MAL DI SCHIENA');
```

```
SELECT p.NOME, p.COGNOME, pc.NOME FROM TABELLA_PAZIENTE p INNER JOIN  
TABELLA_PATOLOGIE_CONTRATTE pc ON p.CODICEFISCALE = pc.PAZIENTE AND pc.DATA_RILEVAZIONE  
<= date('now', '-1 month') ORDER BY pc.NOME;
```

```
SELECT t.TERAPIA AS Terapia, AVG(julianday(t.DATA_FINE) - julianday(t.DATA_INIZIO)) AS Dif-  
ferenzaMedia FROM TABELLA_TERAPIE t WHERE strftime('%Y', t.DATA_FINE) = ? GROUP BY  
t.TERAPIA ORDER BY DifferenzaMedia DESC;
```

```
SELECT o.NOME AS Ospedale, COUNT(e.ID) AS NumeroEsami FROM TABELLA_OSPEDALI o IN-  
NER JOIN TABELLA_ESAMI e ON o.ID = e.OSPEDALE WHERE e.DATA < date('now') GROUP BY  
o.NOME HAVING COUNT(e.ID) > 100
```

```
SELECT p.NOME AS Paziente, t.CAREGIVER AS Caregiver, t.OPERATORE_SANITARIO AS Opera-  
toreSanitario FROM TABELLA_PAZIENTE p INNER JOIN TABELLA_TERAPIE t ON p.CODICEFISCALE  
= t.PAZIENTE WHERE t.CAREGIVER IS NOT NULL AND t.OPERATORE_SANITARIO IS NOT  
NULL
```

```
SELECT p.NOME, p.COGNOME, t.TERAPIA FROM TABELLA_PAZIENTE p INNER JOIN TABELLA_TERAPIE  
t ON p.CODICEFISCALE = t.PAZIENTE WHERE t.TERAPIA = ? AND strftime('%Y', t.DATA_FINE)  
= ?
```