



Laboratorio Interdisciplinare B

Documentazione database

Preparato da

Daniel Satriano	Mat. 745232
Claudio Menegotto	Mat. 745394
Cristian De Nicola	Mat. 744954
Francesco Cavallini	Mat. 746933

Sommario

INTRODUZIONE	3
Schema logico database.....	3
Schema E/R del database.....	4
Implementazione del database in SQL.....	5
Chiavi esterne	6

INTRODUZIONE

Questo database è stato creato usando PostgreSQL in combinazione con l'utilizzo di JDBC per la comunicazione con il server. Di seguito andiamo a mostrare lo schema logico del database, il diagramma ER e l'implementazione.

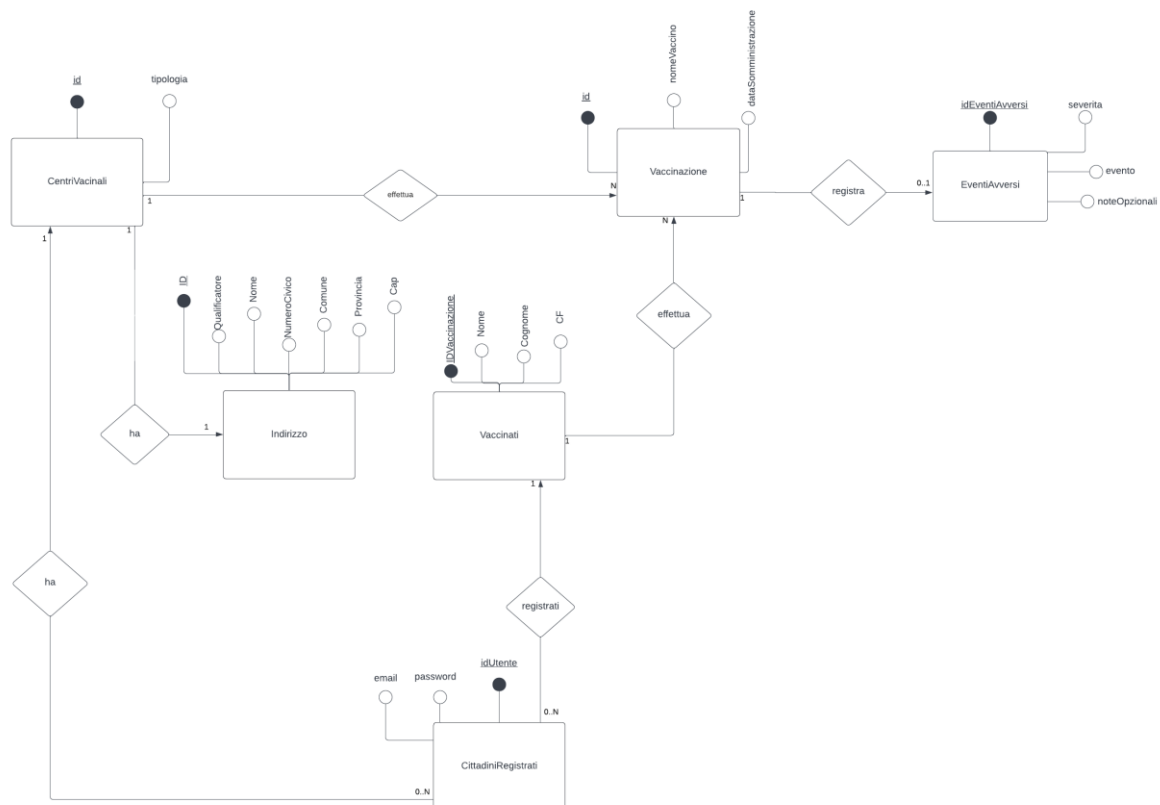
Schema logico database

- CentriVaccinali (***idCentriVaccinali***, Indirizzo, tipologia, nomeCentro)
- Vaccinati (nome, cognome, ***CF***)
- Vaccinazione (***idVaccinazioni***, nomeVaccino, Vaccinati, dataSomministrazione, centriVaccinali)
- Cittadini_Registrati (***idUtente***, Vaccinati, password, email, CentriVaccinali)
- Indirizzo (qualificatore, nome, numeroCivico, comune, provincia, cap, ***idIndirizzo***)
- Eventi avversi (***idEventiAvversi***, severità, evento, noteOpzionali, Vaccinazione)

In questo schema abbiamo messo in corsivo-grassetto le chiavi primarie della tabella e in corsivo sottolineato le chiavi esterne della tabella.

Le chiavi esterne presentano il nome della tabella a cui puntano; pertanto, non sono veritiere con il nome dato all'interno del database. Per informazioni più veritiere controllare il paragrafo dell'implementazione.

Schema E/R del database



Abbiamo deciso di rimuovere la tabella dinamica denominata “*Vaccinati_CentroVaccinale*” con una tabella Vaccinati che presente al suo interno tutte le informazioni degli utenti attualmente vaccinati.

Quest’ultima a sua volta si appoggia alla tabella Vaccinazione, la quale contiene una chiave esterna all’ID di Vaccinati e viene adoperata per il salvataggio di tutte le vaccinazioni effettuate da parte di un centro (o più centri) ad una persona.

Vaccinazione presenta anche una referenza al CentroVaccinale che ha effettuato la vaccinazione, quindi da Vaccinati (che ribadendo, presenta le informazioni di un utente vaccinato) è possibile risalire al centro vaccinale che ha effettuato la vaccinazione.

N.B: Abbiamo deciso di inserire una tabella Vaccinazione in quanto l’utente nel corso della sua vita può fare più di una vaccinazione; quindi, è necessario avere una tabella esterna che ne tenga traccia.

In questo caso la cardinalità Vaccinati - Vaccinazione è di 1 a N; quindi, un vaccinato può effettuare fino a N vaccinazioni.

Implementazione del database in SQL

Di seguito viene riportato il codice SQL della struttura del database [Figura 1]. Una versione copiabile sarà inserita a fine documento.

```
CREATE TABLE Indirizzo(  
    idIndirizzo SERIAL PRIMARY KEY,  
    qualificatore VARCHAR(6) NOT NULL,  
    nome VARCHAR(30) NOT NULL,  
    numeroCivico INT NOT NULL,  
    comune VARCHAR(40) NOT NULL,  
    provincia VARCHAR(5) NOT NULL,  
    cap INT NOT NULL  
);  
  
CREATE TABLE CentriVaccinali(  
    idCentriVaccinali SERIAL PRIMARY KEY,  
    tipologia VARCHAR(11) NOT NULL,  
    nomeCentro VARCHAR(30) NOT NULL,  
    indirizzo INT REFERENCES Indirizzo(idIndirizzo) ON UPDATE CASCADE ON DELETE SET NULL  
);  
  
CREATE TABLE Vaccinati(  
    nome VARCHAR(30) NOT NULL,  
    cognome VARCHAR(30) NOT NULL,  
    codiceFiscale VARCHAR(16) PRIMARY KEY  
);  
  
CREATE TABLE CittadiniRegistrati(  
    idUtente SERIAL PRIMARY KEY,  
    password VARCHAR NOT NULL,  
    email VARCHAR NOT NULL UNIQUE,  
    codiceFiscale VARCHAR(16) REFERENCES Vaccinati(codiceFiscale) ON UPDATE CASCADE ON DELETE CASCADE,  
    idCentriVaccinali INT REFERENCES CentriVaccinali(idCentriVaccinali) ON UPDATE CASCADE ON DELETE CASCADE  
);  
  
CREATE TABLE Vaccinazione(  
    idVaccinazioni SERIAL PRIMARY KEY,  
    nomeVaccino VARCHAR(11) NOT NULL,  
    dataSomministrazione TIMESTAMP not null default CURRENT_TIMESTAMP,  
    codiceFiscale VARCHAR(16) REFERENCES Vaccinati(codiceFiscale) ON UPDATE CASCADE ON DELETE CASCADE,  
    idCentriVaccinali INT REFERENCES CentriVaccinali(idCentriVaccinali) ON UPDATE CASCADE ON DELETE CASCADE  
);  
  
CREATE TABLE EventiAvversi(  
    idEventiAvversi SERIAL PRIMARY KEY,  
    severita VARCHAR(16) NOT NULL,  
    evento VARCHAR(29) NOT NULL,  
    noteOpzionali VARCHAR(255) DEFAULT '',  
    idVaccinazioni INT REFERENCES Vaccinazione(idVaccinazioni) ON UPDATE CASCADE ON DELETE CASCADE  
);
```

Figura 1

Chiavi esterne

CentriVaccinali presenta una chiave esterna denominata **indirizzo** a cui è impostato ON UPDATE cascade, eventuali modifiche alla chiave di Indirizzo verranno riportate anche alla chiave esterna di CentriVaccinali e ON DELETE set null, in quanto anche se l'indirizzo viene eliminato vogliamo comunque avere la referenza del centro per tutti gli eventuali vaccini che ha effettuato in passato.

CittadiniRegistrati presenta due chiavi esterne: **codiceFiscale** e **idCentriVaccinali**, da adesso in poi non andremo più a discutere la motivazione di on update CASCADE, in quanto sarà sempre la stessa per tutte le chiavi esterne, ma ci concentreremo su on delete.

codiceFiscale (chiave esterna che si riferisce alla tabella vaccinati ed usata per prelevarne le relative informazioni) presenta ON DELETE cascade, in quanto se viene eliminato un vaccinato dal database allora anche il record in CittadiniRegistrati dev'essere eliminato in quanto un cittadino può essere registrato solo e soltanto se prima è stato vaccinato. La registrazione di un cittadino permette allo stesso di creare una nuova segnalazione di eventi avversi.

idCentriVaccinali è la referenza a CentriVaccinali che serve per capire a quale centro il cittadino si è registrato, presenta ON DELETE cascade in quanto se un centro viene eliminato, allora non ha senso avere salvati cittadini registrati a quel centro. (N.B se un CittadinoRegistrato viene eliminato non si perdono i dati della sua vaccinazione).

Vaccinazione presenta due chiavi esterne: **codiceFiscale** e **idCentriVaccinali**. Le impostazioni di queste due chiavi esterne sono uguali a quelle di CittadiniRegistrati, guardare il [paragrafo sopra seguente](#) per più informazioni.

EventiAvversi presenta una chiave esterna denominata **idVaccinazioni**, questa chiave serve a tener traccia a quale vaccinazione l'evento avverso si riferisce, in quanto un Vaccinato può avere più di una vaccinazione, di conseguenza può avere più di un report di eventi avversi, ma soltanto uno per vaccinazione. Questo attributo presenta ON DELETE cascade, in quanto se viene eliminato il record della vaccinazione a cui questo evento si riferisce non ha senso tenere un evento che non rappresenta alcun tipo di vaccinazione.