

Manuale tecnico dell'applicazione Centri Vaccinali





INDICE

1. Introduzione	pag.3
2. Sviluppo Base di Dati	
2.1 RACCOLTA E ANALISI DEI REQUISITI	pag.4
2.2 STRUTTURA DELLE RELAZIONI	pag.5
2.3 PROGETTAZIONE CONCRETA	pag.5
2.4 QUERY UTILIZZATE	pag.6
3. Formato dei File e loro gestione	
3.1 JAVA FX	pag.8
3.2 GESTIONE CONNESSIONI REMOTE	pag.8



Introduzione

1. Strumenti e linguaggi utilizzati

Per sviluppare l'applicazione sono stati utilizzati i seguenti strumenti:

- JDK versione 13
- IDE per java: Eclipse con i seguenti plug-in:
 - SceneBuilder



Sviluppo Base di Dati

1. Raccolta e analisi dei requisiti:

I dati relativi ai centri vaccinali da memorizzare sono:

- Il nome del centro stesso (stringa di caratteri);
- Il tipo del centro (stringa di caratteri);
- L'indirizzo del centro (campo composto).

Il nome del centro, in quanto univoco, può essere utilizzato come chiave primaria di questa tabella.

Per memorizzare un vaccinato di un determinato centro vaccinale, gli attributi utili sono:

- Il nome del centro di appartenenza (stringhe di caratteri);
- Nome e cognome del vaccinato (stringa di caratteri);
- Codice fiscale del vaccinato (stringa di caratteri);
- Data della vaccinazione (tipo data);
- Vaccino somministrato (stringa di caratteri);
- L'id univoco di vaccinazione (id numerico su 16 bit, quindi un big int).

In questo insieme di tabelle, una per centro vaccinale registrato, possiamo usare l'id della vaccinazione, in quanto dato univoco, come chiave primaria, mentre il nome del centro può essere usato come chiave esterna per collegare queste tabelle alle tuple della tabella dei centri.

I cittadini registrati a loro volta necessitano dei seguenti dati per essere memorizzati:

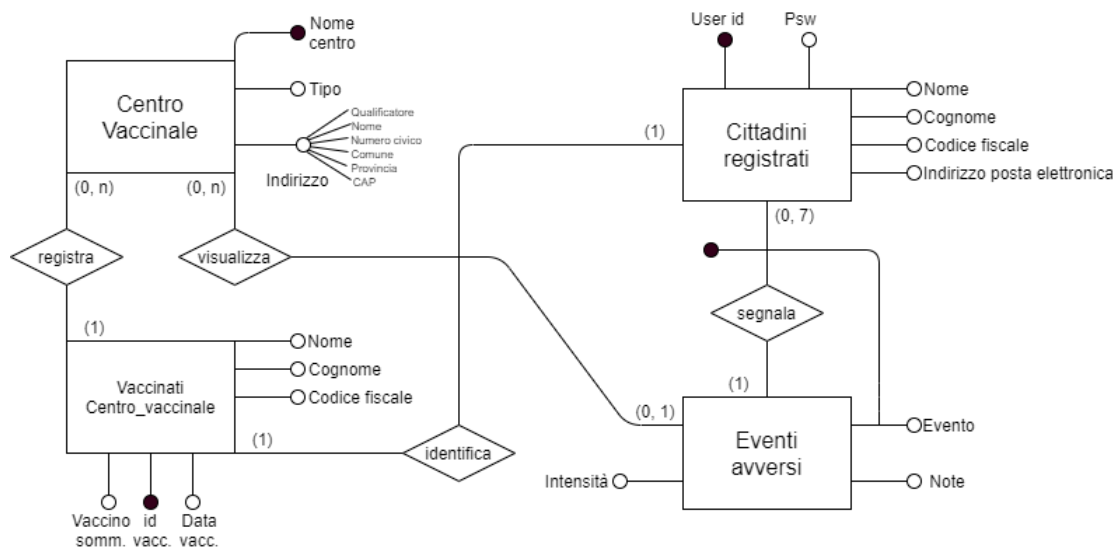
- Nome e cognome del cittadino (stringhe di caratteri);
- Il suo codice fiscale (stringa di caratteri);
- Il suo indirizzo di posta elettronica (stringa di caratteri);
- Uno user id da utilizzare come nickname (stringa di caratteri);
- La relativa password per loggare (stringa di caratteri);
- L'id vaccinazione (big int).

Mentre la chiave esterna per collegare i cittadini registrati alla rispettiva controparte come vaccinati nei centri vaccinali è l'id univoco della registrazione, la primary key non può che essere lo user id.

Infine per memorizzare gli eventi avversi servono:

- L'evento stesso (stringa di caratteri);
- L'intensità di quest'ultimo (intero, valore compreso tra 1 e 5);
- Le note opzionali (stringa di caratteri, massimo 256 caratteri);
- Il segnalatore (stringa di caratteri);
- Il centro dove è avvenuta la vaccinazione (stringa di caratteri).

Chiaramente quest'ultimo campo serve come chiave esterna per collegare questa tabella a quella dei centri vaccinali, mentre invece, dato che nessun altro campo è di per se unico e univoco, come chiave primaria non possiamo che utilizzare l'identità del segnalatore unita all'evento stesso (così facendo un utente non potrà eseguire più volte la segnalazione su uno stesso evento avverso).



2. Struttura delle relazioni:

La relazione tra Centro Vaccinale e Vaccinati Nome_centro è una relazione uno-a-molti, in quanto ogni centro vaccinale può registrare una quantità a piacere di individui vaccinati, mentre ogni vaccinato riceve la vaccinoprofilassi in un solo centro; quella tra Vaccinati Nome_centro e Cittadini Registrati è invece uno-a-uno, in quanto ogni vaccinato identifica un cittadino reale, il quale corrisponde ad un solo vaccinato; quella poi tra Cittadini Registrati ed Eventi Avversi è nuovamente uno-a-molti, in quanto un cittadino può non presentare alcuna reazione avversa alla somministrazione del vaccino come anche vi è la possibilità che la vaccinazione dia all'immunizzato tutte e sette gli eventi avversi segnalabili (mal di testa, febbre, dolori muscolari e articolari, linfadenopatia, tachicardia, crisi ipertensiva e un generico altro); infine la relazione presente fra Centro Vaccinale ed Eventi Avversi è di tipo uno a molti in quanto, in un determinato centro possono avvenire tra le nessuna e le svariate reazioni avverse, mentre un evento avverso, se si presenta, verrà segnalato solamente nell'hub vaccinale di riferimento del segnalatore.

3. Progettazione concreta:

-Creazione database:

```
CREATE DATABASE esame_lab_b;
```

-Creazione tabelle:

Centro Vaccinale:

```
CREATE TABLE centrivaccinali (  
    Nome_centro varchar(40) PRIMARY KEY,  
    Tipo_centro varchar(20) NOT NULL,  
    Qualificatore varchar(20) NOT NULL,  
    Nome_via varchar(40) NOT NULL,  
    Numero_civico varchar(10) NOT NULL,  
    Comune varchar (30) NOT NULL,  
    Provincia varchar (20) NOT NULL,  
    CAP int NOT NULL  
);
```

Vaccinati per centro vacc.:

```
CREATE TABLE vaccinati_Nome_Centro(  
    Nome_centro varchar(40) NOT NULL REFERENCES CentroVaccinale ON DELETE RESTRICT,  
    Nome varchar (20) NOT NULL,  
    Cognome varchar(20) NOT NULL,
```



```
CodiceFiscale varchar(16) UNIQUE NOT NULL,  
Vaccino_somministrato varchar(20) NOT NULL,  
id_vaccinazione bigint PRIMARY KEY,  
Data_vaccinazione date NOT NULL  
);
```

Cittadini registrati nel sistema in seguito al vaccino:

```
CREATE TABLE Cittadini(  
    UserID varchar(30) PRIMARY KEY,  
    Password varchar(20) UNIQUE NOT NULL,  
    Nome varchar (20) NOT NULL,  
    Cognome varchar(20) NOT NULL,  
    CodiceFiscale varchar(16) NOT NULL,  
    Mail varchar(30) NOT NULL,  
    ID_vaccinazione bigint UNIQUE NOT NULL  
);
```

Nota: la chiave esterna verso vaccinati_Nome_Centro è qui assente in quanto serve una tabella di cittadini a livello nazionale e non locale.

Effetti collaterali:

```
CREATE TABLE EventiAvversi(  
    Evento varchar(20) NOT NULL,  
    Intensita int NOT NULL,  
    Vaccino varchar (20) NOT NULL,  
    Nome_centro varchar(40) REFERENCES centrivaccinali,  
    Note varchar(256),  
    UserID varchar(30) REFERENCES Cittadini,  
    PRIMARY KEY (UserID, Evento)  
);
```

Vincolo: Intensita può assumere solo valori compresi tra 1 e 5 (estremi inclusi).

-Popolamento delle tabelle:

Centri vaccinali:

```
INSERT INTO centrivaccinali (Nome_centro, Tipo_centro, Qualificatore, Nome_via, Numero_civico,  
Comune, Provincia, CAP) VALUES ('?', '?', '?', '?', '?', '?', '?', '?');
```

Vaccinati per centro:

```
INSERT INTO vaccinati_Nome_Centro (Nome_centro, Nome, Cognome, CodiceFiscale,  
Vaccino_somministrato, id_vaccinazione, Data_vaccinazione) VALUES ('?', '?', '?', '?', '?', '?', '?');
```

Utenti registrati:

```
INSERT INTO cittadini (Nome, Cognome, CodiceFiscale, Mail, Userid, Password, ID_vaccinazione)  
VALUES ('?', '?', '?', '?', '?', '?', '?');
```

Eventi avversi:

```
INSERT INTO eventi_avversi (Evento, Intensita, Vaccino, Nome_centro, Note, UserID) VALUES ('?',  
'?', '?', '?', '?', '?');
```

4. Query:

Autenticazione:

```
SELECT Password  
FROM Cittadini  
WHERE userid = ?
```

Info centro vaccinale (ricerca per nome):

```
SELECT *
```



```
FROM centrivaccinali
WHERE Nome_centro ILIKE '%?%'
```

Nomi centri vaccinali:

```
SELECT *
FROM centrivaccinali
```

Info centro vaccinale (ricerca per comune e tipologia):

```
SELECT *
FROM centrivaccinali
WHERE Comune ILIKE '%?%' AND Tipo_centro = '%?%'
```

Ricerca ID:

```
SELECT id_vaccinazione
FROM vaccinati_Nome_Centro
WHERE id_vaccinazione = '?'
```

ID dell'utente:

```
SELECT id_vaccinazione
FROM cittadini
WHERE userid = '?'
```

Eventi Avversi:

```
SELECT COUNT (Evento) AS Casi, AVG (Intensita) AS IntensitaMedia
FROM Eventi_Avversi
WHERE Nome_centri = '?' AND Evento = '?'
```



Formato dei File e la loro gestione

1. JAVAFX

Per la gestione delle interfacce è stato scelto di utilizzare il più moderno e versatile JavaFx. Tuttavia, l'utilizzo di questa libreria di java prevede la creazione di file .fxml che contengono le informazioni necessarie alla creazione e alla corretta visualizzazione della Graphic User Interface. L'editing di questi file è stato delegato al software Scene Builder.

2. GESTIONE CONNESSIONI REMOTE

Tra i metodi noti per la gestione delle comunicazioni remote, è stato prediletto il Remote Method Invocation, al fine di eliminare i problemi di concorrenza che, in caso di accesso a dati condivisi si sarebbero potuti verificare. In questo modo i singoli Client possono facilmente inviare le proprie Query alle tabelle del Database, utilizzando in remoto i metodi del Server.

L'applicazione utilizza di default localhost e il Registro 1099 al fine di simulare una connessione remota. Il server accetta connessioni da più client contemporaneamente, ma solo un singolo Server può essere aperto in un dato momento.