

# Tatum Vaccini

## Documentazione database

Garegnani Federico	746789	VA
Rossi Giorgio	746571	VA
Canali Luca	744802	VA
Callegari Pietro	746568	VA
Invernizzi Daniele	746484	VA

3 febbraio 2023

## Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Schema logico</b>	<b>2</b>
<b>3</b>	<b>Schema ER</b>	<b>2</b>
<b>4</b>	<b>Implementazione SQL</b>	<b>3</b>
4.1	Chiavi esterne e integrità referenziale . . . . .	3
4.2	Creazione tabelle . . . . .	4

# 1 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.

## 2 Schema logico

- Centro\_vaccinale(Nome, tipologia)
- Cittadino(CF, Nome, Cognome, email, Username, Password)
- Vaccinazione(IDvacc, CF<sup>citt</sup>, Data, Vaccino)
- EventoAvverso(ID<sub>ea</sub>, ID<sup>vacc</sup>, Evento, Intensità)

In questo schema sono state evidenziate in ***corsivo grassetto*** le chiavi primarie della tabella e in *corsivo sottolineato* le chiavi esterne della tabella.

## 3 Schema ER

Riportiamo in Figura 1 lo schema ER del database e in Figura 2 lo stesso schema ristrutturato. Riportiamo infine in Figura 3 uno schema più ad alto livello che permette di farsi direttamente un'idea dell'organizzazione in tabelle.

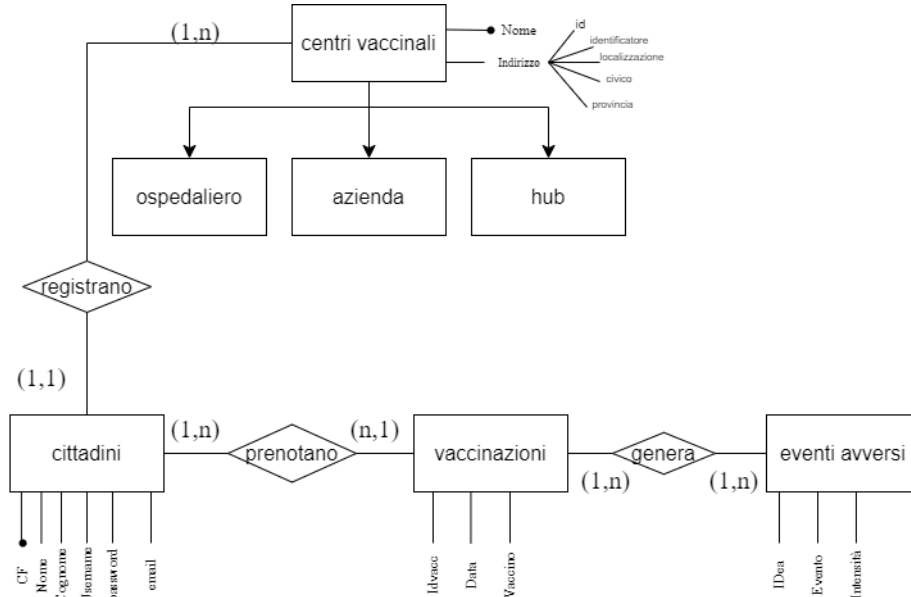


Figura 1: Schema ER della base di dati

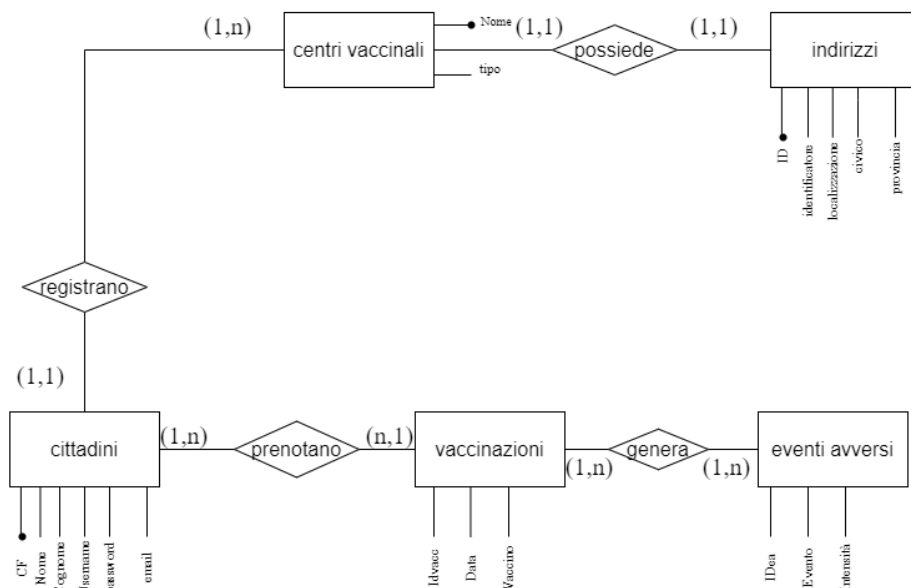


Figura 2: Schema ER ristrutturato della base di dati

## 4 Implementazione SQL

### 4.1 Chiavi esterne e integrità referenziale

**Indirizzo** Presenta una chiave esterna denominata `centro_vaccinale` alla quale è impostata `ON UPDATE CASCADE` e `ON DELETE CASCADE`, questo al fine di mantenere la base di dati in uno stato consistente; vogliamo infatti che alla cancellazione di un centro vaccinale venga eliminato anche il relativo indirizzo.

**Registrazione** Ha una chiave primaria costituita da due elementi: `(centro_vaccinale, CF)`, entrambi chiavi esterne. Il primo elemento fa riferimento al centro vaccinale in cui avviene la registrazione, mentre il secondo al codice fiscale del cittadino che si è registrato. Entrambi i campi sono ovviamente impostati con `ON UPDATE CASCADE` e `ON DELETE CASCADE` perché l'eliminazione di uno dei due valori comporterebbe l'impossibilità di effettuare una vaccinazione nel dominio reale e un'inconsistenza logica nel dominio astratto della base di dati. La relazione esistente tra la tabella `centro_vaccinale` e `registrazione` è del tipo *uno a molti*, per ogni centro vaccinale sono infatti permesse molte registrazioni; la relazione tra `cittadino` e `registrazione` è invece *uno a uno*, ogni cittadino è infatti registrato presso un solo centro vaccinale.

**Vaccinazione** Presenta la chiave esterna `CF_citt` avente come riferimento il codice fiscale nella tabella `cittadino`. Al campo `CF_citt` sono impostati `ON UPDATE CASCADE` e `ON DELETE CASCADE` per le stesse ragioni discusse con le tabelle precedenti. La relazione esistente tra `cittadino` e

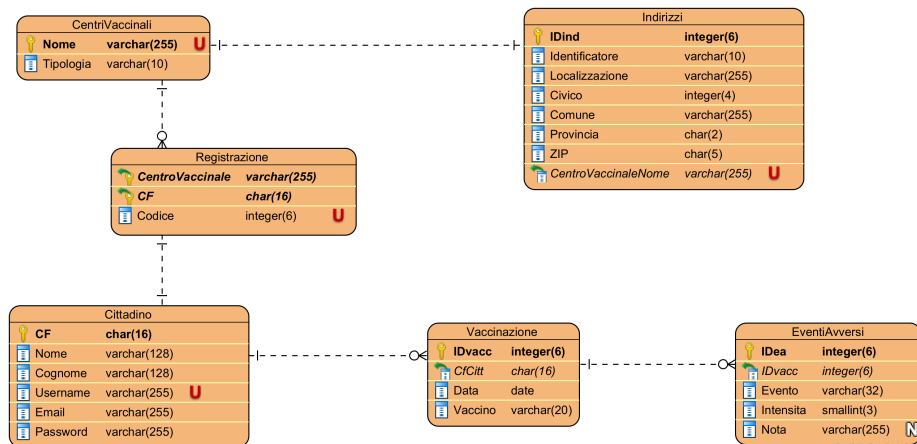


Figura 3: Schema delle tabelle del database

**vaccinazione** è del tipo *uno a molti*, è infatti previsto il caso in cui un cittadino venga sottoposto a diverse vaccinazioni.

**EventoAvverso** Presenta una chiave esterna verso la tabella **vaccinazione** nel campo **ID\_vaccino** a cui è associato il solito **ON UPDATE CASCADE** e **ON DELETE CASCADE**. Questa tabella tiene traccia di ogni evento avverso generatosi in seguito a una vaccinazione e registrato dal cittadino; per ogni vaccinazione possono essere registrati più eventi avversi, la relazione tra la tabella **vaccinazione** e la tabella **evento\_avverso** è quindi del tipo *uno a molti*.

## 4.2 Creazione tabelle

Riportiamo di seguito le istruzioni per la creazione delle tabelle all'interno del database.

```

CREATE TABLE centro_vaccinale (
    nome VARCHAR(255),
    tipologia VARCHAR(10),
    PRIMARY KEY (nome)
);
  
```

Listing 1: Codice per la creazione della tabella CentriVaccinali

```

CREATE TABLE Indirizzo (
    id_ind SERIAL PRIMARY KEY,
    identificatore VARCHAR(10),
    localizzazione VARCHAR(255),
    civico NUMERIC(4),
    comune VARCHAR(255),
    provincia CHAR(2),
    zip CHAR(5),
    centro_vaccinale VARCHAR(255),
    FOREIGN KEY (centro_vaccinale) REFERENCES centro_vaccinale(nome)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

```

Listing 2: Codice per la creazione della tabella Indirizzi

```

CREATE TABLE Registrazione (
    centro_vaccinale VARCHAR(255) REFERENCES centro_vaccinale
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CF CHAR(16) UNIQUE REFERENCES cittadino
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    codice SERIAL,
    PRIMARY KEY (centro_vaccinale,CF)
);

```

Listing 3: Codice per la creazione della tabella Registrazione

```

CREATE TABLE cittadino (
    CF CHAR(16),
    nome VARCHAR(128),
    cognome VARCHAR(128),
    email VARCHAR(255),
    password VARCHAR(255),
    username VARCHAR(255) UNIQUE,
    PRIMARY KEY (CF)
);

```

Listing 4: Codice per la creazione della tabella Cittadini

```

CREATE TABLE Vaccinazione (
    ID_vaccino SERIAL PRIMARY KEY,
    data DATE,
    vaccino VARCHAR(20),
    CF_citt CHAR(16),
    FOREIGN KEY (CF_citt) REFERENCES Cittadino
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

```

Listing 5: Codice per la creazione della tabella Vaccinazioni

```

CREATE TABLE EventoAvverso (
    ID_ea SERIAL PRIMARY KEY,
    evento VARCHAR(32),
    intensita SMALLINT,
    ID_vaccino NUMERIC(6),
    FOREIGN KEY (ID_vaccino) REFERENCES Vaccinazione(ID_vaccino)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

```

Listing 6: Codice per la creazione della tabella EventiAvversi