

2.5 Schema relazionale

CITTADINO(Cod, CF, Nome, Cognome, Et , Email*, Cellulare*, Indirizzo, Citt , PrecedentePositivit , Tipo)

CENTROVACCINALE(Cod, Citt , Indirizzo)

MEDICO(Cod, Tipo, CentroVaccinale, AbilitazioneSingolaDose)

LOTTO(Cod, TipoVaccino, DataScadenza, DataProduzione, NumeroDosi)

TIPOVACCINO(Cod, Nome, Et Min, Et Max, Efficacia, DosiRichieste, IntervalloSomministrazione*)

APPUNTAMENTO VACCINALE(Cod, Data, Ora, CentroVaccinale, Cittadino, Lotto)

POSSIEDE DOSI(CentroVaccinale, TipoVaccino, NumeroDosi)

ALLERGIA(Nome, Cittadino)

REPORT(AppuntamentoVaccinale, Medico, CentroVaccinale, Vaccino, Data)

Vincoli di integrit  referenziale:

ALLERGIA(Cittadino) referencia CITTADINO(Cod)

REPORT(AppuntamentoVaccinale) referencia APPUNTAMENTO VACCINALE(Cod)

REPORT(Medico) referencia MEDICO(Cod)

REPORT(CentroVaccinale) referencia CENTRO VACCINALE(Cod)

REPORT(Vaccino) referencia TIPO VACCINO(Cod)

APPUNTAMENTO VACCINALE(Cittadino) referencia CITTADINO(Cod)

APPUNTAMENTO VACCINALE(CentroVaccinale) referencia CENTRO VACCINALE(Cod)

APPUNTAMENTO VACCINALE(Lotto) referencia LOTTO(Cod)

MEDICO(CentroVaccinale) referencia CENTRO VACCINALE(Cod)

MEDICO(Cod) referencia CITTADINO(Cod)

POSSIEDE DOSI(CentroVaccinale) referencia CENTRO VACCINALE(Cod)

POSSIEDE DOSI(TipoVaccino) referencia TIPOVACCINO(Cod)

LOTTO(TipoVaccino) referencia TIPO VACCINO(Cod)

3 IMPLEMENTAZIONE

3.1 DDL di creazione database

-- Tipi

```
create type tipo_cittadino as ENUM('personale sanitario', 'personale scolastico',  
  'soggetto fragile', 'altro');  
create type tipo_medico as ENUM('altro medico', 'medico di base');  
create type nome_vaccino as ENUM('Covidin', 'Coronax', 'Flustop');
```

-- Sequenze

```
create sequence if not exists id_vaccino  
increment by 1 start 1;  
create sequence if not exists id_cittadino  
increment by 1 start 1;  
create sequence if not exists id_appuntamento_vaccinale  
increment by 1 start 1;  
create sequence if not exists id_centro  
increment by 1 start 1;
```

-- Tabelle

```
create table if not exists cittadino (  
  cod integer default nextval('id_cittadino') primary key,  
  cf varchar(16) not null check (length(cf)=16) unique,  
  nome varchar(32) not null,  
  cognome varchar(32) not null,  
  eta integer not null check (eta>=0),  
  email varchar(128) unique,  
  cellulare bigint unique check (cellulare>3200000000 and cellulare<3939999999),  
  indirizzo varchar(128) not null,  
  citta varchar(64) not null,  
  precedente_positivita boolean not null default FALSE,  
  tipo tipo_cittadino not null default 'altro',  
  check ((cellulare is not null and email is null) or (cellulare is null and email is not null))  
);
```

/* Nota sugli attributi città: Al fine di evitare la ripetizione di una stringa per decine di migliaia di volte, in un contesto reale sarebbe da valutare la creazione di una tabella città (con id, nome e opzionalmente altri parametri) a cui riferirsi tramite id

Discorsi analogo potrebbe essere fatto con l'attributo indirizzo, creando un'entità indirizzo collegata a città tramite associazione tuttavia in questo caso il risparmio in termini di spazio sarebbe minore in quanto VAL(indirizzo, cittadino) << VAL(città, cittadino)

Il discorso è analogo in tutti i casi in cui trattiamo le città o gli indirizzi

*/

```

create table if not exists centro_vaccinale (
  cod integer default nextval('id_centro') primary key,
  indirizzo varchar(128) not null,
  citta varchar(64) not null,
  unique (indirizzo, citta)
);

```

```

create table if not exists tipo_vaccino (
  cod integer default nextval('id_vaccino') primary key,
  nome varchar(32) unique,
  eta_min integer not null check (eta_min >= 0),
  eta_max integer not null check (eta_max > eta_min),
  efficacia real not null check (efficacia > 0 and efficacia <= 100),
  dosi_richieste integer not null check (dosi_richieste = 1 or dosi_richieste = 2),
  intervallo_somministrazione integer default null check (intervallo_somministrazione > 0
    or intervallo_somministrazione is null),
  check (
    case
      when dosi_richieste = 1 then intervallo_somministrazione is null
      else intervallo_somministrazione is not null
    end
  )
);

```

```

create table if not exists medico (
  cod integer primary key,
  tipo tipo_medico not null,
  centro integer not null,
  abilitazione_singola_dose boolean not null,
  foreign key (cod) references cittadino (cod) on delete cascade on update cascade,
  foreign key (centro) references centro_vaccinale(cod) on update cascade,
  check (
    case
      when abilitazione_singola_dose = TRUE then tipo = 'altro medico'
      else tipo = 'medico di base'
    end
  )
);

```

```

create table if not exists lotto (
  cod varchar(6) check (length(cod)=6) primary key,
  -- supponiamo che gli id siano alfanumerici e di lunghezza costante
  vaccino integer,
  num_dosi integer not null default 500 check (num_dosi>0),
  data_produzione date not null,
  data_scadenza date not null check (data_scadenza > data_produzione),
  unique (data_produzione, data_scadenza, vaccino),
  foreign key (vaccino) references tipo_vaccino (cod) on update cascade
);

```

```

create table if not exists appuntamento_vaccinale (
  cod integer default nextval('id_appuntamento_vaccinale') primary key,
  data_appuntamento date,
  ora time,
  centro integer,
  lotto varchar(6) not null,
  cittadino integer not null,
  /*
  si ipotizza che gli appuntamenti vaccinali siano creati in funzione del cittadino
  e non che vengano prima creati i vari appuntamenti e in un secondo momento associati ai cittadini
  */
  foreign key (centro) references centro_vaccinale (cod) on update cascade,
  foreign key (lotto) references lotto(cod) on update cascade,
  foreign key (cittadino) references cittadino (cod) on delete cascade on update cascade,
  unique (data_appuntamento, ora, centro)
);

```

```

create table if not exists possiede_dosi (
  centro integer,
  vaccino integer,
  num_dosi integer not null check (num_dosi>=0),
  -- da rendere consistente, ogni volta che viene somministrata una dose va sottratto 1
  primary key (centro, vaccino),
  foreign key (centro) references centro_vaccinale (cod) on update cascade,
  foreign key (vaccino) references tipo_vaccino (cod) on update cascade
);

```

```

create table if not exists allergia (
  nome varchar(128) not null,
  cittadino integer,
  foreign key (cittadino) references cittadino (cod) on delete cascade on update cascade
);

```

```
create table if not exists report (  
  appuntamento_vaccinale integer not null,  
  medico integer not null,  
  centro integer not null,  
  data_report date not null,  
  vaccino integer not null,  
  foreign key (centro) references centro_vaccinale (cod) on update cascade,  
  foreign key (vaccino) references tipo_vaccino (cod) on update cascade,  
  foreign key (medico) references medico (cod) on update cascade,  
  foreign key (appuntamento_vaccinale) references appuntamento_vaccinale (cod),  
  primary key (appuntamento_vaccinale)  
);
```

3.2 DML di popolamento delle tabelle

begin;

-- Cittadino

-- nota: essendo dati fittizi si sceglie di mettere come
suffisso di CF a11a123b (o 51 nel caso di donne)

-- e rendere coerente solo la prima parte

insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)

values ('dptfri11a11a123b', 'Fiero', 'Di Pietro', 10, null, 3290145027, 'Via del sole 7',
'Torino', FALSE, 'altro');

insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)

values ('brmpal01a11a123b', 'Palamede', 'Bramati', 20, 'p.bramati@email.com', null,
'Strada portorosso 12', 'Genova', TRUE, 'personale sanitario');

insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)

values ('ghmngl98a51a123b', 'Angela', 'Aghemio', 23, null, 3376295481,
'Corso san giovanni 138', 'Roma', FALSE, 'personale scolastico');

insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)

values ('smrlni60a11a123b', 'Lino', 'Sommariva', 61, null, 3292674805, 'Stradale torino',
'Milano', FALSE, 'soggetto fragile');

insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)

values ('tnorsl76a51a123b', 'Rossella', 'Tonani', 45, 'ross.tonani@gmail.com', null,
'Via tempesta 5', 'Milano', TRUE, 'altro');

insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)

values ('zpagbr91a11a123b', 'Gabriele', 'Zappa', 30, null, 3902837401, 'Via alighieri 1a',
'Napoli', FALSE, 'altro');

insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)

values ('dflflp95a11a123b', 'Filippo', 'De Filippi', 26, 'defilippi.fil@yahoo.com', null,
'Via de medici 90', 'Napoli', TRUE, 'altro');

insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)

values ('zmllei70a11a123b', 'Elio', 'Zemello', 51, 'zemee51@libero.it', null, 'Via arsenale 5',
'Milano', FALSE, 'personale sanitario');

insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)

values ('stfbrd86a11a123b', 'Berardo', 'Stefanelli', 35, null, 3290394850, 'Via garibaldi 10',

```

'Torino', TRUE, 'personale scolastico');
insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)
values ('sprsmn90a11a123b', 'Simone', 'Sparaventi', 31, 'ventisimone_31@email.net', null,
'Via sapienza 50', 'Roma', FALSE, 'altro');
insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)
values ('vsclrn40a11a123b', 'Lorenzo', 'Viscomi', 81, null, 3592531876, 'Via del sentiero',
'Milano', TRUE, 'soggetto fragile');
insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)
values ('mzania69a11a123b', 'Iana', 'Mazzuia', 52, 'maziana_2@gmail.com', null,
'Strada della repubblica', 'Torino', FALSE, 'altro');
insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)
values ('brcpio85a11a123b', 'Pippo', 'Burchi', 36, null, 3236987459, 'Corso caselle 70',
'Torino', TRUE, 'personale sanitario');
insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)
values ('flnlmr66a11a123b', 'Leonardo', 'Filano', 55, 'fil.leo66@gmail.co,', null,
'Corso cesare 177', 'Milano', FALSE, 'personale sanitario');
insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)
values ('dgzrti85a51a123b', 'Rita', 'De Guzzi', 36, null, 3596542587, 'Via dei mercanti 8',
'Genova', FALSE, 'altro');
insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)
values ('mntfda60a51a123b', 'Fadia', 'Menta', 61,
'f.menta@virgilio.it', null, 'Strada del colle 3', 'Roma', TRUE, 'personale sanitario');
insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)
values ('dgvant70a11a123b', 'Antonio', 'De Giovanni', 51, null, 3549631478,
'Corso romano 75', 'Napoli', FALSE, 'personale sanitario');
insert into cittadino (cf, nome, cognome, eta, email, cellulare, indirizzo, citta,
precedente_positivita, tipo)
values ('ctsmta89a11a123b', 'Matteo', 'Actis', 32,
'mattewactis@gmail.com', null, 'Via del mare 5', 'Genova', TRUE, 'personale sanitario');

```


-- Centro Vaccinale

```
insert into centro_vaccinale (indirizzo, citta)
values ('Via ritrovata 15', 'Torino');
insert into centro_vaccinale (indirizzo, citta)
values ('Via retta 87', 'Milano');
insert into centro_vaccinale (indirizzo, citta)
values ('Via Roma 1', 'Roma');
insert into centro_vaccinale (indirizzo, citta)
values ('Via della speranza 42', 'Napoli');
insert into centro_vaccinale (indirizzo, citta)
values ('Via degli artisti 7', 'Genova');
```

-- Medico

```
insert into medico (cod, tipo, centro, abilitazione_singola_dose)
values (8, 'medico di base', 2, FALSE);
insert into medico (cod, tipo, centro, abilitazione_singola_dose)
values (14, 'altro medico', 2, TRUE);
insert into medico (cod, tipo, centro, abilitazione_singola_dose)
values (13, 'medico di base', 1, FALSE);
insert into medico (cod, tipo, centro, abilitazione_singola_dose)
values (16, 'altro medico', 3, TRUE);
insert into medico (cod, tipo, centro, abilitazione_singola_dose)
values (17, 'altro medico', 4, TRUE);
insert into medico (cod, tipo, centro, abilitazione_singola_dose)
values (18, 'altro medico', 5, TRUE);
```

-- Vaccino

```
insert into tipo_vaccino (nome, eta_min, eta_max, efficacia, dosi_richieste,
    intervallo_somministrazione)
values ('Covidin', 12, 85, 90, 2, 50);
insert into tipo_vaccino (nome, eta_min, eta_max, efficacia, dosi_richieste,
    intervallo_somministrazione)
values ('Coronax', 45, 150, 99, 2, 40);
insert into tipo_vaccino (nome, eta_min, eta_max, efficacia, dosi_richieste)
values ('Flustop', 65, 100, 95, 1);
```

-- Lotto

```
insert into lotto
values ('cn0987', 1, 500, '2020-11-05', '2021-02-23');
insert into lotto (cod, vaccino, data_produzione, data_scadenza)
values ('1234cx', 2, '2021-01-21', '2021-04-17');
insert into lotto
values ('f4567p', 3, 500, '2021-03-13', '2021-06-30');
```

-- Appuntamento Vaccinale

```
insert into appuntamento_vaccinale (data_appuntamento, ora, centro, lotto, cittadino)
values ('2021-01-15', '09:00', 2, 'cn0987', 4);
insert into appuntamento_vaccinale (data_appuntamento, ora, centro, lotto, cittadino)
values ('2021-02-20', '18:15', 2, 'cn0987', 4);
insert into appuntamento_vaccinale (data_appuntamento, ora, centro, lotto, cittadino)
values ('2021-02-09', '15:30', 3, '1234cx', 3);
insert into appuntamento_vaccinale (data_appuntamento, ora, centro, lotto, cittadino)
values ('2021-07-1', '08:23', 2, 'cn0987', 14);
insert into appuntamento_vaccinale (data_appuntamento, ora, centro, lotto, cittadino)
values ('2021-03-03', '14:57', 5, 'cn0987', 15);
insert into appuntamento_vaccinale (data_appuntamento, ora, centro, lotto, cittadino)
values ('2021-05-01', '15:45', 5, 'cn0987', 15);
insert into appuntamento_vaccinale (data_appuntamento, ora, centro, lotto, cittadino)
values ('2021-04-30', '20:35', 1, 'cn0987', 1);
insert into appuntamento_vaccinale (data_appuntamento, ora, centro, lotto, cittadino)
values ('2021-01-7', '11:00', 1, 'f4567p', 9);
insert into appuntamento_vaccinale (data_appuntamento, ora, centro, lotto, cittadino)
values ('2021-06-15', '16:17', 4, 'cn0987', 6);
insert into appuntamento_vaccinale (data_appuntamento, ora, centro, lotto, cittadino)
values ('2021-07-15', '09:20', 4, 'cn0987', 6);
insert into appuntamento_vaccinale (data_appuntamento, ora, centro, lotto, cittadino)
values ('2021-03-03', '09:20', 2, '1234cx', 8);
```

-- Possiede Dosi

```
insert into possiede_dosi
values (1, 3, 75000);
insert into possiede_dosi
values (2, 1, 125000);
insert into possiede_dosi
values (3, 2, 9000);
insert into possiede_dosi
values (4, 1, 30000);
insert into possiede_dosi
values (5, 1, 500);
```

-- Allergia

```
insert into allergia
values ('lattosio', 1);
insert into allergia
values ('amoxicillina', 8);
insert into allergia
values ('niche', 14);
```

```
-- Report
insert into report (appuntamento_vaccinale, medico, centro, data_report, vaccino)
values (11, 14, 2, '2021-03-03', 2);
insert into report (appuntamento_vaccinale, medico, centro, data_report, vaccino)
values (6, 18, 5, '2021-05-01', 2);

commit;
```

3.3 Operazioni di cancellazione e modifica

/*

Tutte le operazioni svolte si riferiscono al database appena creato
e non sono da considerarsi sequenziali

*/

--select

-- Ogni sera viene stilato un rapporto che indica quante vaccinazioni sono state fatte nella
-- giornata in tutti i centri vaccinali suddivise per categoria di cittadino.

```
select c.tipo, count(*)
from appuntamento_vaccinale av join cittadino c on av.cittadino = c.cod
where av.data_appuntamento = CURRENT_DATE
group by c.tipo;
```

-- Ogni sera, ogni centro vaccinale fa l'inventario del numero di dosi ancora
-- disponibili per ogni vaccino

```
select *
from possiede_dosi
order by centro, vaccino;
```

-- Ogni fine settimana, viene stilato un report che indica quante
-- vaccinazioni sono state fatte per ogni vaccino per ognuna delle categorie di
-- cittadini e quante di queste abbiano causato allergie.

```
with num_tot_vaccino as(
  select c.tipo as tipo_cittadino, l.vaccino as tipo_vaccino, count(*) as num_vaccini
  from appuntamento_vaccinale av join lotto l on av.lotto = l.cod join cittadino c on av.cittadino = c.cod
  group by c.tipo, l.vaccino
), num_tot_report as(
  select c.tipo as tipo_cittadino, l.vaccino as tipo_vaccino, count(*) as num_report
  from report r join appuntamento_vaccinale av on r.appuntamento_vaccinale = av.cod
  join cittadino c on av.cittadino = c.cod join lotto l on av.lotto = l.cod
  where r.data_report between CURRENT_DATE - 7 and CURRENT_DATE
  group by c.tipo, l.vaccino
)
select v.tipo_cittadino, v.tipo_vaccino, num_vaccini, num_report
from num_tot_vaccino v, num_tot_report r
where v.tipo_cittadino = r.tipo_cittadino and v.tipo_vaccino = r.tipo_vaccino;
```

--edit

```
update allergia
set cittadino = 15
where cittadino = 1;
```

```
update centro_vaccinale
set cod=6
where cod=4;
```

```
update tipo_vaccino
set cod=4
where cod=2;
```

```
update lotto
set cod='0987cx'
where cod='1234cx';
```

```
update cittadino
set cod=1000
where cod=8;
```

```
update centro_vaccinale
set cod=1234
where cod=2;
```

```
update medico
set cod=12
where cod=2;
```

--delete--

```
delete from cittadino
where cod = 1;
```

```
delete from tipo_vaccino
where cod = 3; -- non viene portato a termine perché viola vincolo in lotto
```

```
delete from cittadino
where cod=17;
-- provoca la rimozione anche di una entry da medico
```

```
delete from centro_vaccinale
where cod=2; -- non viene portata a termine perché viola un vincolo in medico
```

```
delete from lotto  
where cod='cn0987';
```

```
delete from cittadino  
where cod=8;
```

-- queste operazioni non vengono portate a termine perché i valori sono referenziati in altre tabelle