

Esercizi

Un database per la memorizzazione dei dati relativi ad una elezione svoltasi in data 01/04/2015 è caratterizzato dalla seguente porzione di schema logico relazionale:¹

```
DISTRETTI(nome)
SEZIONI(distrettofk, numero, indirizzo)
PERSONE(codice_fiscale, cognome, nome, data_nascita, data_morte*)
TESSERE_ELETTORALI(numero, [sezione_distretto, sezione_numero]fk, personafk, data_emissione)
CANDIDATURE(personafk, distrettofk)
VOTI_ESPRESSI(tesserafk, ora_voto)
VOTI_SCRUTINATI([sezione_distretto, sezione_numero]fk, progr, candidatofk*)
```

Con riferimento allo schema relazionale suddetto, esprimere le seguenti interrogazioni in linguaggio SQL (ove non altrimenti specificato).

1. Scrivere l'istruzione DDL per la definizione della relazione TESSERE_ELETTORALI.
2. Aggiungere il vincolo di integrità che impedisce ad una persona di essere in possesso di più di una tessera elettorale.
3. Definire la vista relazionale “Elettorato attivo”, che associa ad ogni sezione le persone autorizzate a votare presso quella sezione nel corso di quelle elezioni. Sono autorizzati a votare in una determinata sezione le persone che, con riferimento alla data delle elezioni:
 - hanno compiuto i 18 anni di età e non sono (ancora) deceduti;
 - sono in possesso di una tessera elettorale della sezione;
 - la tessera elettorale è stata emessa prima della data delle elezioni.

Per ogni persona dell'elettorato attivo, indicare il numero di tessera elettorale ed i dati anagrafici.

4. Eliminare dal database tutti i voti espressi la cui tessera si riferisce ad una persona che non fa parte dell'elettorato attivo.
5. Definire una vista relazionale che associ ad ogni distretto il numero di *voti espressi* nelle sezioni di quel distretto entro le ore 12:00. Definire una seconda vista relazionale che calcoli per ogni distretto il numero delle persone componenti l'elettorato attivo. Usando le viste suddette, calcolare la *percentuale di affluenza* alle ore 12:00.
6. Usando l'algebra relazionale, esprimere il vincolo di integrità che impedisce di avere voti espressi al di fuori dell'intervallo temporale dalle ore 08:00 alle ore 22:30.

¹Un attributo annullabile è indicato con la notazione A*. I vincoli di chiave esterna sono indicati con la notazione A_{fk}; la relazione a cui fa riferimento il vincolo è ricavabile facilmente dal nome dell'attributo.

```

DISTRETTI(nome)
SEZIONI(distrettofk, numero, indirizzo)
PERSONE(codice_fiscale, cognome, nome, data_nascita, data_morte*)
TESSERE_ELETTORALI(numero, [sezione_distretto, sezione_numero]fk, personafk, data_emissione)
CANDIDATURE(personafk, distrettofk)
VOTI_ESPRESSI(tesserafk, ora_voto)
VOTI_SCRUTINATI([sezione_distretto, sezione_numero]fk, progr, candidato*fk)

```

Es: 1)

Scrivere l'istruzione DDL per la definizione della relazione tessere elettorali.

```

create table distretti (
    nome varchar(60) not null primary key
);

create table sezioni (
    distretto varchar(60) not null references distretti(nome),
    numero integer not null,
    indirizzo varchar(200) not null,
    primary key(distretto,numero)
);

create table persone(
    codice_fiscale char(16) not null primary key,
    cognome varchar(100) not null,
    nome varchar(100) not null,
    data_nascita date not null,
    data_morte date null, --opzionale
);

create table tessere_elettorali (
    numero integer not null primary key,
    sezione_distretto varchar(60) not null,
    sezione_numero integer not null,
    persona char(16) not null references persone (codice_fiscale),
    data_emissione date not null,
    foreign key (sezione_distretto, sezione_numero)
        references sezioni(distretto,numero)
);

create table candidature(
    persona char(16) not null references persone(codice_fiscale),
    distretto varchar(60) not null references distretti(nome),
    primary key (persona)
);

create table voti_espressi (
    tessera integer not null primary key references tessere_elettorali(numero),
    ora_voto time not null
);

```

```

DISTRETTI(nome)
SEZIONI(distrettofk, numero, indirizzo)
PERSONE(codice_fiscale, cognome, nome, data_nascita, data_morte*)
TESSERE_ELETTORALI(numero, [sezione_distretto, sezione_numero]fk, personafk, data_emissione)
CANDIDATURE(personafk, distrettofk)
VOTI_ESPRESSI(tesserafk, ora_voto)
VOTI_SCRUTINATI([sezione_distretto, sezione_numero]fk, progr, candidatofk)

```

```

create table voti_scrutinati (
    sezione_distretto varchar(60) not null,
    sezione_numero integer not null,
    progr integer not null,
    candidato char(16) null references candidature(persona),
    primary key (sezione_distretto, sezione_numero, progr),
    foreign key (sezione_distretto, sezione_numero)
        references sezioni(distretto, numero)
);

```

Es: 2)

Aggiungere il vincolo di integrità che impedisce ad una persona di essere in possesso di più di una tessera elettorale.

-- chiediamo che la persona compaia solo una volta nelle tessere elettorali (vincolo di unicità / vincolo di chiave).

```

alter table tessere_elettorali
    add constraint non_piu_di_una_tessera_a_testa
    unique (persona);

```

Es: 3)

Definire la vista relazionale “Elettorato attivo”, che associa ad ogni sezione le persone autorizzate a votare presso quella sezione nel corso di quelle elezioni. Sono autorizzati a votare in una determinata sezione le persone che, con riferimento alla data delle elezioni:

- hanno compiuto i 18 anni di età e non sono (ancora) deceduti;
- sono in possesso di una tessera elettorale della sezione;
- la tessera elettorale è stata emessa prima della data delle elezioni.

Per ogni persona dell’elettorato attivo, indicare il numero di tessera elettorale ed i dati anagrafici.

```

create or replace view elettorato_attivo (
(sezione_distretto, sezione_numero, codice_fiscale, numero_tessera, cognome, nome, data_nascita)
as
select t.sezione_distretto, t.sezione_numero, p.codice_fiscale, t.numero_tessera, p.cognome, p.nome,
p.data_nascita
from persone p, tessere_elettorali t
where p.codice_fiscale = t.persona
    and t.data_emissione < '2015-04-01'
    and (to_date('2015-04-01', 'YYYY-MM-DD') - p.data_nascita) / 365 >= 18
    and (p.data_morte > '2015-04-01' or p.data_morte is null)

```

Si poteva fare la join così:

```

from persone p join tessere_elettorali t
    on p.codice_fiscale = t.persona

```

...

```

DISTRETTI(nome)
SEZIONI(distrettojk, numero, indirizzo)
PERSONE(codice_fiscale, cognome, nome, data_nascita, data_morte*)
TESSERE_ELETTORALI(numero, [sezione_distretto, sezione_numero]jk, personajk, data_emissione)
CANDIDATURE(personajk, distrettojk)
VOTI_ESPRESSI(tesserajk, ora_voto)
VOTI_SCRUTINATI([sezione_distretto, sezione_numero]jk, progr, candidatojk*)

```

Es: 4)

Eliminare dal database tutti i voti espressi la cui tessera si riferisce ad una persona che non fa parte dell'elettorato attivo.

```

delete from voti_espressi
where tessera not in (
    select numero_tessa
    from elettorato_attivo
);

```

Es: 5)

Definire una vista relazionale che associ ad ogni distretto il numero di voti espressi nelle sezioni di quel distretto entro le ore 12:00. Definire una seconda vista relazionale che calcoli per ogni distretto il numero delle persone componenti l'elettorato attivo. Usando le viste suddette, calcolare la percentuale di affluenza alle ore 12:00.

```

create or replace view voti_espressi_entro_mezzogiorno
(distretto, numero_voti)
as
select te.sezione_distretto, count(*)
from voti_espressi ve, tessere_elettorali te
where ve.tessera = te.numero
    and ve.ora_voto <= '12:00:00'
group by te.sezione_distretto;

```

```

create or replace view numero_elettori_per_distretto
(distretto, numero_elettori)
as
select sezione_distretto, count(*)
from elettorato_attivo
group by sezione_distretto;

```

```

select ve.distretto,
    cast((ve.numero_voti * 100.0) / ne.numero_elettori as numeric(5,2)) as percentuale
from numero_elettori_per_distretto ne, voti_espressi_entro_mezzogiorno ve
where ve.distretto = ne.distretto

```

Es: 6)

Usando l'algebra relazionale, esprimere il vincolo di integrità che impedisce di avere voti espressi al di fuori dell'intervallo temporale dalle ore 08:00 alle ore 22:30.

Cercare ennuple che violano il vincolo e imporre che nel risultato finale non ci siano ennuple che violino il vincolo (quindi che sia insieme vuoto).

```

SEL_{ ora_voto not between '8:00':time and '22:30':time } (voti_espressi) = { }

```