

Esercizi

Un database per un insieme di musei pubblici italiani è caratterizzato dalla seguente porzione di schema logico relazionale:¹

```
MUSEI(nome, città)
ARTISTI(nome, anno_nascita, anno_morte*, nazionalità)
OPERE(codice, artistafk*, titolo, anno, museofk, tipo, altezza, larghezza, profondità*)
PRESTITI(operafk, museofk, data_inizio, data_fine)
RESTAURI(operafk, data_inizio, data_fine*)
```

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 PRESTITI includendo, oltre ai vincoli indicati nello schema, gli ovvi vincoli di ennupla.
2. Il tipo di un'opera può essere "dipinto", "scultura" o "installazione". Aggiungere alla tabella OPERE un vincolo di integrità che, oltre al requisito suddetto, imponga che la profondità sia valorizzata per tutte le opere di tipo diverso da "dipinto".
3. Eliminare dal database tutti i prestiti conclusi da almeno cinque anni.
4. Elencare gli artisti le cui opere non sono mai state oggetto di restauro.
5. Definire la vista relazionale OPERE_FRUIBILI, che associa ad ogni museo le opere che sono di proprietà del museo oppure sono attualmente concesse in prestito a questo museo da parte di altri musei e sono effettivamente fruibili per il pubblico, in quanto non sono oggetto di restauro e non sono state concesse in prestito ad altro museo. Nota: fare riferimento alla situazione valida in data odierna.
6. Durante tutto il mese di maggio 2016, il museo "Uffizi" di Firenze propone una mostra sull'artista Giotto. A tale scopo, si fa prestare (da tutti gli altri musei) le opere di Giotto di loro proprietà, per il periodo dal 15 aprile al 5 giugno 2016. Modificare la tabella PRESTITI per tenere traccia di questo evento. Nota: per semplicità, ignorare eventuali prestiti preesistenti di opere di Giotto che riguardano il periodo in questione.
7. Elencare gli artisti che, nell'insieme di tutti i musei, hanno almeno 10 opere, ordinandoli in base al numero di opere realizzate (a parità di opere, usare l'ordinamento in base al nome dell'artista). Per ogni artista, indicare il numero di musei che possiedono almeno un'opera dell'artista.
8. Usando l'algebra relazionale, codificare il vincolo di chiave primaria per la relazione MUSEI.

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.

```

MUSEI(nome, città)
ARTISTI(nome, anno_nascita, anno_morte*, nazionalità)
OPERE(codice, artista_fk, titolo, anno, museo_fk, tipo, altezza, larghezza, profondità*)
PRESTITI(opera_fk, museo_fk, data_inizio, data_fine)
RESTAURI(opera_fk, data_inizio, data_fine*)

```

Es: 1,2)

1: Scrivere l'istruzione DDL per la definizione della relazione prestiti includendo, oltre ai vincoli indicati nello schema, gli ovvi vincoli di ennupla.

2: Il tipo di un'opera può essere "dipinto", "scultura" o "installazione". Aggiungere alla tabella opere un vincolo di integrità che, oltre al requisito suddetto, imponga che la profondità sia valorizzata per tutte le opere di tipo diverso da "dipinto".

```

create table musei (
    nome varchar(100) not null primary key,
    citta varchar(100) not null,
);

```

```

create table artisti(
    nome varchar(100) not null,
    anno_nascita numeric(4) not null,
    anno_morte numeric(4) null, -- annullabile per quelli in vita
    nazionalita varchar(100) not null,
    primary key (nome)
);

```

-- per avere data nascita minore o uguale di data morte serve un vincolo di ennupla

```

alter table artisti
    add constraint no_zombie
    check (anno_nascita <= data_morte);

```

```

create table opere (
    codice integer not null primary key,
    artista varchar(100) null,
    titolo varchar(100) not null,
    anno numeric(4) not null,
    museo varchar(100) not null references musei(nome),
    tipo varchar(20) not null,
    altezza numeric(6,1) not null,
    larghezza numeric(6,1) not null,
    profondita numeric(6,1) null,
    check (altezza > 0 and larghezza > 0 and profondita > 0),
    check (tipo in ('dipinto','scultura','installazione') and (tipo='dipinto' or profondita is not null))
    foreign key (artista) references artisti(nome)
);

```

```

create table prestiti (
    opera integer not null primary key references opere(codice),
    museo varchar(100) not null references musei(nome),
    data_inizio date not null,
    data_fine date not null,
    check (data_inizio < data_fine)
);

```

MUSEI(nome, città)
 ARTISTI(nome, anno_nascita, anno_morte*, nazionalità)
 OPERE(codice, artista_{fk}, titolo, anno, museo_{fk}, tipo, altezza, larghezza, profondità*)
 PRESTITI(opera_{fk}, museo_{fk}, data_inizio, data_fine)
 RESTAURI(opera_{fk}, data_inizio, data_fine*)

```

create table restauri (
    opera integer not null primary key references opere(codice),
    data_inizio date not null,
    data_fine date null,
    check (data_inizio < data_fine)
);

```

Es: 3)

Trovare/Eliminare dal database tutti i prestiti conclusi da almeno cinque anni.

```

select *
from prestiti
where cast('today' as date) - data_fine >= 5*365;

delete
from prestiti
where cast('today' as date) - data_fine >= 5*365;

```

Es: 4)

Elencare gli artisti le cui opere non sono mai state oggetto di restauro.

```

select nome -- alcuni vogliono che sia scritto a.nome
from artisti a
where not exist (
    select *
    from opere o, restauri r
    where o.codice = r.opera
    and o.artista = a.nome
);

```

Es: 6)

Durante tutto il mese di maggio 2016, il museo “Uffizi” di Firenze propone una mostra sull’artista Giotto. A tale scopo, si fa prestare (da tutti gli altri musei) le opere di Giotto di loro proprietà, per il periodo dal 15 aprile al 5 giugno 2016. Modificare la tabella prestiti per tenere traccia di questo evento. Nota: per semplicità, ignorare eventuali prestiti preesistenti di opere di Giotto che riguardano il periodo in questione.

```

insert into prestiti (opera, museo, data_inizio, data_fine) -- insert particolare
select codice, 'Uffizi', '2010-04-15'::date, '2016-06-05'::date
from opere
where artista = 'Giotto' and museo != 'Uffizi';

```

-- Le chiavi esterne all'artista e al museo sono il nome stesso dell'artista e del museo quindi ho già dei dati senza fare join.

```

MUSEI(nome, città)
ARTISTI(nome, anno_nascita, anno_morte*, nazionalità)
OPERE(codice, artista_fk, titolo, anno, museo_fk, tipo, altezza, larghezza, profondità*)
PRESTITI(opera_fk, museo_fk, data_inizio, data_fine)
RESTAURI(opera_fk, data_inizio, data_fine*)

```

Es: 5)

Definire la vista relazionale opere fruibili, che associa ad ogni museo le opere che sono di proprietà del museo oppure sono attualmente concesse in prestito a questo museo da parte di altri musei e sono effettivamente fruibili per il pubblico, in quanto non sono oggetto di restauro e non sono state concesse in prestito ad altro museo. Nota: fare riferimento alla situazione valida in data odierna.

-- opere di proprietà del museo

```

create or replace view opere_fruibili (museo, opera) as (
    select museo, codice
    from opere
    where codice not in ( -- bisogna togliere quelle concesse dal museo ad altri
        select opera
        from opere_attualmente_in_prestito
    )
    and codice not in ( -- bisogna togliere quelle in restauro
        select opera
        from opere_attualmente_in_restauro
    )
)
union (
-- opere in prestito da altri musei
    select museo, opera
    from opere_attualmente_in_prestito
    where opera not in (
        select opera
        from opere_attualmente_in_restauro
    )
)

```

```

create or replace view opere_attualmente_in_prestito (opera, museo) as
    select opera, museo
    from prestiti
    where 'today'::date between data_inizio and (data_fine - 1) -- between include gli estremi

```

```

create or replace view opere_attualmente_in_restauro (opera) as
    select opera
    from restauri
    where 'today'::date >= data_inizio
        and ('today'::date < data_fine or 'today'::date is null)
-- i restauri hanno data fine annullabile e va gestita

```

Es: 7)

Elencare gli artisti che, nell'insieme di tutti i musei, hanno almeno 10 opere, ordinandoli in base al numero di opere realizzate (a parità di opere, usare l'ordinamento in base al nome dell'artista). Per ogni artista, indicare il numero di musei che possiedono almeno un'opera dell'artista.

```

select artista, count(*) as numero_opere, count(distinct museo) as numero_musei
from opere
group by artista
having count(*) >= 10
order by numero_opere desc, artista

```

```

MUSEI(nome, città)
ARTISTI(nome, anno_nascita, anno_morte*, nazionalità)
OPERE(codice, artistafk*, titolo, anno, museofk, tipo, altezza, larghezza, profondità*)
PRESTITI(operafk, museofk, data_inizio, data_fine)
RESTAURI(operafk, data_inizio, data_fine*)

```

Es: 8)

Usando l'algebra relazionale, codificare il vincolo di chiave primaria per la relazione musei.

$m1(nome1, città1) := proj_{\{nome, città\}}(musei)$

$m2(nome2, città2) := m1$

$m1 \text{ join_} \{nome1=nome2 \text{ and } città1 \neq città2\} m2 = \{\}$ -- insieme vuoto