

Basi di dati

Prova di autovalutazione 2020/2021

A. Dato il seguente schema logico e i seguenti vincoli, si scriva il codice SQL che crea le tabelle corrispondenti. Si ricordi di creare le tabelle nell'ordine richiesto dal linguaggio SQL.

autori(id,nome,cognome, dataNascita)
stesure(autore,idLibro,nazioneLibro,ordine)
libri(id,nazione,titolo,anno,editore, copertina,ISBN,disponibile,prezzo)

Vincoli integrità referenziale

stesure.autore -> autori.id

stesure.idLibro, stesure.nazioneLibro -> libri.id, libri.nazione

Altri vincoli e osservazioni

La colonna ordine di stesure contiene un numero intero positivo. Il valore di default è 1.

La colonna anno contiene un valore intero positivo successivo al 1950.

La colonna copertina di un libro deve poter contenere un'immagine di al più 5M.

La colonna ISBN contiene una sequenza lunga esattamente 13 caratteri. Il codice ISBN è unico per ciascun libro.

Il nome dell'editore è una stringa lunga fino a 50 caratteri. Non possono esistere due editori con lo stesso nome nella stessa nazione,

Il campo disponibile di un libro contiene i valori "1" o "0".

Il prezzo di un libro deve contenere valori positivi fino a 99,99.

A parte le chiavi, il nome, il cognome degli autori, il titolo e il codice ISBN dei libri, tutti gli altri campi possono essere nulli.

Sia dato il seguente schema relazionale

progetti(id,nomeProgetto,budget,sede)
incarichi(progetto,dipendente,dataIncarico)
dipendenti(id,nomeDipendente,dataNascita,stipendio)
genitori(genitore,figlio)

Vincoli integrità referenziale

incarichi.progetto -> progetti.id
incarichi.dipendente -> dipendenti.id
genitori.genitore->dipendenti.id
genitori.figlio->dipendenti.id

**B. Scrivere le seguenti interrogazioni in algebra relazionale con theta join
(non per gli studenti di servizi giuridici)**

1. Trovare il nome dei dipendenti che non hanno dipendenti con incarichi
2. Trovare il nome del dipendente per il quale ne esiste un altro nell'archivio nato nello stesso nella stessa data

**C. Scrivere le seguenti interrogazioni in calcolo su tuple
(non per gli studenti di servizi giuridici)**

1. Trovare il nome del dipendente più vecchio fra quelli nati dopo il '1/1/1960'
2. Trovare il nome dei dipendenti dipendenti a cui sono stati assegnati incarichi solo dopo il dopo il '1/1/2020'. Siete liberi di visualizzare o non visualizzare le anche i dipendenti senza progetti a seconda di come vi torni più comodo

D. Scrivere le seguenti interrogazioni sia in SQL

3. Trovare il nome e lo stipendio dei dipendenti nati prima del '1/1/1980' e con stipendio maggiore di 30.000 che abbiano almeno un incarico. Ordinarli per nome.
4. Trovare i progetti a cui partecipano dipendenti con stipendio minore di 50.000 restituendo nome del progetto e nome del dipendente. Includere anche i progetti senza dipendenti (con stipendio minore di 50.000).
5. Trovare il nome dei figli del dipendente 'paperino'. (Si assume che tali figli siano anch'essi dipendenti, presenti nell'archivio e che la relazione genitore-figlio sia memorizzata nella tabella 'genitori').
6. Trovare la somma dei budget dei progetti svolti in ogni sede restituendo la sede e la somma.
7. Ripetere l'interrogazione precedente considerando nel calcolo della somma solo i progetti con budget minore di 10.000 e restituendo solo le sedi con più di 3 progetti.
8. Usando gli operatori insiemistici, si trovino i dipendenti con stipendio maggiore di 100.000 senza incarichi
9. Usando le sottointerrogazioni si trovino i progetti i cui dipendenti hanno TUTTI ricevuto l'incarico dopo il '1/1/2020'. Restituire il nome del dipendente. Siete liberi di visualizzare o non visualizzare le anche i progetti senza dipendenti a seconda di come vi torni più comodo.
10. Si trovi il nome dei dipendenti più giovani
11. Si trovi il massimo del numero dei dipendenti che partecipano ad un progetto, restituendo solo il massimo (Non è necessario restituire informazioni sul progetto che raggiunge il massimo, Serve usare una vista).

Soluzioni

Esercizio A

```
create table autori(  
  id serial not null primary key,  
  nome varchar(30) not null,  
  cognome varchar(30) not null,  
  dataNascita date)
```

```
create table libri(  
  id integer not null,  
  nazione varchar(20) not null,  
  anno int check(anno is null or anno >= 1950),  
  editore varchar(50),  
  copertina blob(5M),  
  ISBN char(13) not null unique,  
  disponibile bit,  
  prezzo decimal(4,2),  
  primary key(id,nazione)  
)
```

```
create table stesure(  
  autore int not null references autori(id),  
  idLibro int not null,  
  nazioneLibro varchar(20) not null,  
  ordine int check(ordine is null or ordine > 0) default 1,  
  primary key-autore, idLibro, nazioneLibro),  
  foreign key(idLibro,nazioneLibro) references libri(id,nazione)  
)
```

B. Scrivere le seguenti interrogazioni in algebra relazionale con theta join

1. Trovare il nome dei dipendenti che non hanno dipendenti con incarichi

$$\text{DIPENDENTI_CON_INCARCHI} = \pi_{\text{id}, \text{nomeDipendente}} (\text{dipendenti} \bowtie_{\text{id}=\text{dipendente}} \text{incarichi})$$
$$\pi_{\text{nomeDipendente}} (\pi_{\text{id}, \text{nomeDipendente}} (\text{dipendenti}) - \text{DIPENDENTI_CON_INCARCHI})$$

2. Trovare il nome dei dipendenti per quali ne esistono altri nell'archivio nati nello stessa nella stessa data

DIPENDENTE2 =

$$\delta_{\text{id2}, \text{nomeDipendente2}, \text{dataNascita2}, \text{stipendio2} < - \text{id}, \text{nomeDipendente}, \text{dataNascita}, \text{stipendio}} (\text{DIPENDENTE})$$
$$\pi_{\text{nomeDipendente}} (\text{DIPENDENTE2} \bowtie_{\text{dataNascita2}=\text{dataNascita} \wedge \text{id2} \neq \text{id}} \text{dipendente})$$

C. Scrivere le seguenti interrogazioni in calcolo su tuple

1. Trovare il nome del dipendente più vecchio fra quelli nati dopo il '1/1/1960'

```
{d.nomeDipendente | d(dipendente) | d.dataNascita > '1/1/1960' ^  
  ¬∃ d2(dipendente) d2.dataNascita < d.dataNascita  
  ^ d2.dataNascita > '1/1/1960', }
```

2. Trovare il nome dei dipendenti dipendenti a cui sono stati assegnati incarichi solo dopo il dopo il '1/1/2020'. Siete liberi di visualizzare o non visualizzare le anche i dipendenti senza progetti a seconda di come vi torni più comodo

```
{d.nomeDipendente | d(dipendente) | ¬∃ i(incarico) i.dataIncarico > '1/1/2020' ^ i.dipendente = d.id}
```

Esercizio D

3. Trovare il nome e lo stipendio dei dipendenti nati prima del '1/1/1980' e con stipendio maggiore di 30.000 che abbiano almeno un incarico. Ordinarli per nome.

```
Select distinct nomeDipendente, stipendio  
from dipendenti, incarichi  
where dipendenti.id=incarichi.dipendente  
and dataNascita < '1/1/1980' and stipendio > 30000  
order by nomeDipendente
```

4. Trovare i progetti a cui partecipano dipendenti con stipendio minore di 50.000 restituendo nome del progetto e nome del dipendente. Includere anche i progetti senza dipendenti (con stipendio minore di 50.000).

```
select nomeDipendente, stipendio  
from progetti left join incarichi on progetti.id=incarichi.progetto  
left join dipendenti on dipendenti.id=incarichi.dipendente and stipendio < 50000
```

5. Trovare il nome dei figli del dipendente 'paperino'. (Si assume che tali figli siano anch'essi dipendenti, presenti nell'archivio e che la relazione genitore-figlio sia memorizzata nella tabella 'genitori').

```
select dipendenti2.nomeDipendente as "nome figlio"  
from dipendenti as dipendenti1, dipendenti as dipendenti2, genitori  
where dipendenti1.id=genitori.genitore and  
  genitori.figlio=dipendenti2.id and  
  dipendenti1.nomeDipendente='paperino'
```

6. Trovare la somma dei budget dei progetti svolti in ogni sede restituendo la sede e la somma.

```
select sede, sum(budget) as "somma"  
from progetti  
group by sede
```

7. Ripetere l'interrogazione precedente considerando nel calcolo della somma solo i progetti con budget minore di 10.000 e restituendo solo le sedi con più di 3 progetti.

```
select sede, sum(budget) as "somma"
from progetti
where budget < 10000
group by sede
having count(*) >3
```

8. Usando gli operatori insiemistici, si trovi l'id e il nome dei dipendenti con stipendio maggiore di 100.000 senza incarichi

```
select id, nomeDipendente
from dipendenti
where stipendio > 100000
except
select id, nomeDipendente
from dipendenti, incarichi
where stipendio > 100000
and incarichi.dipendente=dipendenti.id
```

9. Usando le sottointerrogazioni si trovino i progetti i cui dipendenti hanno TUTTI ricevuto l'incarico dopo il '1/1/2020'. Restituire il nome del dipendente. Siete liberi di visualizzare o non visualizzare le anche i progetti senza dipendenti a seconda di come vi torni più comodo.

```
select id, nomeProgetto
from progetti
where not exists (select *
                  from incarichi
                  where incarichi.progetto=progetti.id
                    and incarichi.dataIncarico <='1/1/2020' )

oppure

select id, nomeProgetto
from progetti
where (select dataIncarico
       from incarichi
       where incarichi.progetto=progetti.id ) > all '1/1/2020'
```

10. Si trovi il nome dei dipendenti più giovani

```
select dipendenti.nomeDipendente
from dipendenti
where dataNascita=(select max(dataNascita)
                  from dipendenti)
```

11. Si trovi il massimo del numero dei dipendenti che partecipano ad un progetto, restituendo solo il massimo (Non è necessario restituire informazioni sul progetto che raggiunge il massimo, Serve usare una vista).

```
create view numeroDipendentiPerProgetto(progetto,numero) as
select nomeProgetto, count(*)
from progetti, incarichi
where progetti.id=incarichi.progetto
group by id, nomeProgetto
```

```
select max(numero) AS "massimo"
from numeroDipendentiPerProgetto
```