

ITINERE_10_2021

Tema_A

BIBLIOTECHE(codice, nome, citta, indirizzo)

LIBRI(codice, titolo, edizione, anno, pagine)

AUTORI(codice, nome, cognome, anno_nascita, biografia)

AUTORI_LIBRI(libro_{fk}, autore_{fk})

COPIE_LIBRI(seriale, libro_{fk}, biblioteca_{fk}, collocazione)

PRESTITI(codice, data_inizio, data_fine_prevista, data_fine_effettiva*, copia_libro_{fk})

1. Scrivere l'istruzione DDL per la definizione della relazione autori_libri includendo, oltre ai vincoli indicati nello schema, il vincolo che impone che per ogni libro non vi possano essere più autori con la stessa posizione (ordine sequenza) nella sequenza degli autori.

```
CREATE TABLE autore_libri (  
    libro INTEGER NOT NULL,  
    autore INTEGER NOT NULL,  
    ordine_sequenza INTEGER NOT NULL,  
    UNIQUE(libro, autore, ordine_sequenza),  
    PRIMARY KEY (libro, autore),  
    FOREIGN KEY (libro) REFERENCES libri(codice),  
    FOREIGN KEY (autore) REFERENCES autori(codice)  
);
```

2. Definire la vista relazionale libri_con_prestiti_scaduti(codice_libro, titolo) che elenca i codici e i titoli dei libri per i quali esiste almeno un prestito in corso la cui data prevista di restituzione è precedente alla data odierna.

```
CREATE OR REPLACE VIEW libri_con_prestiti_scaduti AS  
SELECT DISTINCT l.codice AS codice_libro, l.titolo AS titolo  
FROM prestiti p, copie_libri cl, libri l  
WHERE p.data_fine_prevista < CURRENT_DATE  
    AND p.copia_libro = cl.seriale  
    AND cl.libro = l.codice
```

3. Modificare i prestiti in corso per le copie di libri della biblioteca di nome "Biblioteca Pavese" di Parma, spostando in avanti di 30 giorni la data fine prevista.

```
UPDATE prestiti  
SET data_fine_prevista = data_fine_prevista + 30  
WHERE copia_libro IN (  
    SELECT cl.seriale  
    FROM prestiti p, copie_libri cl, biblioteche b, libri l  
    WHERE p.copia_libro = cl.seriale  
        AND cl.biblioteca = b.codice  
        AND cl.libro = l.codice  
        AND b.nome = 'Biblioteca Pavese'  
        AND b.citta = 'Parma'  
        AND data_fine_prevista > current_date  
);
```

4. Per ogni città e per ogni autore, calcolare il numero di prestiti registrati, dall'inizio del 2015 alla fine del 2019, in una biblioteca di quella città e che hanno riguardato (una copia di) un libro di quell'autore.

```
SELECT b.città, a.cognome, COUNT(p.codice) AS prestiti_redistrati
FROM biblioteche b, autori a, prestiti p,
     copie_libri cl, libri l, autori_libri al
WHERE p.data_inizio ≥ '2015-01-01'
      AND p.data_inizio ≤ '2019-12-31'
      AND p.copia_libro = cl.seriale
      AND cl.libro = l.codice
      AND cl.biblioteca = b.codice
      AND al.libro = l.codice
      AND al.autore = a.codice
GROUP BY b.città, a.cognome
ORDER BY b.città, a.cognome;
```

5. Modificare lo schema della tabella Prestiti, aggiungendo il vincolo di integrità che impedisce di avere una data inizio superiore alla data fine prevista e alla data fine effettiva.

```
ALTER TABLE prestiti
ADD CONSTRAINT valid_date
CHECK (
    data_inizio < data_fine_prevista
    AND
    data_inizio < data_fine_effettiva
);
```

Tema_B

BIBLIOTECHE(codice, nome, città, indirizzo)

LIBRI(codice, titolo, edizione, anno, pagine)

AUTORI(codice, nome, cognome, anno_nascita, biografia)

AUTORI_LIBRI(libro_{fk}, autore_{fk})

COPIE_LIBRI(seriale, libro_{fk}, biblioteca_{fk}, collocazione)

PRESTITI(codice, data_inizio, data_fine_prevista, data_fine_effettiva*, copia_libro_{fk})

1. Scrivere l'istruzione DDL per la definizione della relazione copie libri includendo, oltre ai vincoli indicati nello schema, il vincolo che impone che ogni biblioteca non possa avere più copie dello stesso libro.

```
CREATE TABLE copie_libri (
    seriale SERIAL NOT NULL,
    libro INTEGER NOT NULL,
    biblioteca INTEGER NOT NULL,
    collocazione INTEGER NOT NULL,
    PRIMARY KEY (seriale),
    FOREIGN KEY (libro) REFERENCES libri(codice),
    FOREIGN KEY (biblioteca) REFERENCES biblioteche(codice),
    UNIQUE(libro, biblioteca)
);
```

2. Definire la vista relazionale autori ignorati(codice, cognome, nome) che elenca gli autori per i cui libri non sono stati registrati prestiti nel corso dell'anno 2021 (considerare la data d'inizio del prestito).

```

CREATE OR REPLACE VIEW autori_ignorati AS
SELECT a.codice, a.cognome, a.nome
FROM prestiti p, copie_libri cl, autori_libri al, autori a
WHERE p.copia_libro = cl.seriale
      AND cl.libro = al.libro
      AND al.autore = a.codice
      AND a.codice NOT IN (
          SELECT a.codice
          FROM prestiti p, copie_libri cl, autori_libri al, autori a
          WHERE p.data_inizio ≥ '2021-01-01'
                AND p.data_inizio ≤ '2021-12-31'
                AND p.copia_libro = cl.seriale
                AND cl.libro = al.libro
                AND al.autore = a.codice
      );

```

3. Eliminare i libri per i quali non sono presenti copie nelle biblioteche.

```

DELETE FROM libri
WHERE codice NOT IN (
    SELECT l.codice
    FROM copie_libri cl, libri l, biblioteche b
    WHERE cl.libro = l.codice
          AND cl.biblioteca = b.codice
);

```

4. Per ogni biblioteca e per ogni autore, calcolare il numero di copie di libri di quell'autore presenti nella biblioteca (nota: una copia si considera presente anche se è al momento in prestito).

```

SELECT b.nome AS biblioteca, a.cognome AS autore, COUNT(cl.seriale)
FROM copie_libri cl, autori_libri al, biblioteche b, autori a
WHERE cl.libro = al.libro
      AND cl.biblioteca = b.codice
      AND al.autore = a.codice
GROUP BY b.nome, a.cognome
ORDER BY b.nome, a.cognome

```

5. Modificare lo schema della tabella Prestiti, aggiungendo il vincolo di integrità che impedisce di avere una data inizio superiore alla data fine prevista e alla data fine effettiva.

```

ALTER TABLE prestiti
ADD CONSTRAINT valid_date
CHECK (
    data_inizio < data_fine_prevista
    AND
    data_inizio < data_fine_effettiva
);

```