

Soluzione Esercitazione 4

Esercizio 1

Trovare tutti i titoli dei libri acquisiti nel 2010 in ordine alfabetico

```
SELECT Titolo
FROM Copia AS C, DatiLibro AS D
WHERE C.ISBN = D.ISBN AND
      DataAcquisizione >= 1/1/2010 AND
      DataAcquisizione <= 31/12/2010
ORDER BY Titolo
```

oppure utilizzando l'operatore *between*:

```
SELECT Titolo
FROM Copia AS C, DatiLibro AS D
WHERE C.ISBN = D.ISBN AND
      DataAcquisizione BETWEEN 1/1/2010 AND 31/12/2010
ORDER BY Titolo
```

La clausola ORDER BY, che compare in coda all'interrogazione, ordina le righe del risultato. Dopo l'attributo si può specificare ASC o DESC per indicare ordine ascendente o discendente.

Esercizio 2

Trovare quanti prestiti sono stati effettuati.

```
SELECT COUNT(*) as NumeroTotalePrestiti
FROM Prestito
```

Esercizio 3

Trovare quante copie sono state acquisite nel 2018 e quanto sono state pagate complessivamente (in migliaia di euro).

Ci si aspetta un risultato di una riga con totale copie acquisite e costo complessivo

```
SELECT COUNT(*) AS TotCopie, SUM(Costo)/1000 AS TotEuroK
FROM Copia
WHERE DataAcquisizione BETWEEN 1/1/2018 AND 31/12/2018
```

Gli aggregati vengono calcolati su cosa rimane dopo aver filtrato la tabella Copia, secondo quanto previsto dal WHERE

Esercizio 4

Trovare quanti prestiti di libri di genere Giallo sono stati effettuati. Scrivere due query, una considerando i libri come copie fisiche una con i libri entità.

Libri come copie:

```
SELECT COUNT(*) AS TotPrestiti
FROM Prestito AS P, Copia AS C, DatiLibro AS D
WHERE Genere = "Giallo"
      AND P.Collocazione = C.Collocazione
      AND C.Isbn = D.Isbn
```

Utilizzando il join naturale:

```
SELECT COUNT(*) AS TotPrestiti
FROM Prestito NATURAL JOIN Copia NATURAL JOIN DatiLibro
WHERE Genere = "Giallo"
```

Per contare i libri come entità è necessario guardare all'ISBN: per averlo bisogna passare in ogni caso dalla tabella Copia. La query sarà quindi molto simile e deve utilizzato un DISTINCT all'interno del COUNT sull'attributo richiesto (ISBN).

```
SELECT COUNT(DISTINCT(D.Isbn)) AS TotPrestitiISBN
FROM Prestito AS P, Copia AS C, DatiLibro AS D
WHERE Genere = "Giallo"
AND P.Collocazione = C.Collocazione
AND C.Isbn = D.Isbn
```

Prendere l'ISBN da C o da D è indifferente.

Esercizio 5

Trovare quanti libri diversi sono stati acquisiti nel 2020

```
SELECT COUNT(DISTINCT(ISBN)) AS TotAcquisizioni
FROM Copia
WHERE DataAcquisizione BETWEEN 1/1/2020 AND 31/12/2020
```

Esercizio 6

Trovare il libro più costoso e con titolo che inizia per 'I' acquisito dalla biblioteca.

```
SELECT D1.Titolo, C1.Collocazione, C1.Costo
FROM Copia AS C1
NATURAL JOIN DatiLibro as D1
WHERE C1.Titolo LIKE "I%"
AND C1.Costo = (SELECT MAX(C2.Costo)
FROM Copia AS C2
NATURAL JOIN DatiLibro AS D2
WHERE C2.Titolo LIKE "I%")
```

Attenzione: in presenza di un raggruppamento, nella target list si possono inserire solo:

- funzioni aggregate
- attributi sui quali si sta raggruppando

ERRORI GRAVI:

```
SELECT Titolo, MAX(Costo)
FROM Copia NATURAL JOIN DatiLibro
WHERE Titolo LIKE "I%"
```

questa query restituisce una tabella di due colonne, la prima con TUTTI i libri i cui titoli iniziano per I la seconda con SEMPRE il costo massimo tra tutti quei libri.

Aggregato nel WHERE che non esprime condizione:

```
SELECT Titolo, Costo
FROM Copia NATURAL JOIN DatiLibro
WHERE Titolo LIKE "I%" AND MAX(Costo)
```

Esercizio 7

Trovare il primo libro acquisito dalla biblioteca

```
SELECT D.ISBN, D.Titolo, C.DataAcquisizione
FROM Copia C JOIN DatiLibro D ON C.Isbn = D.Isbn
WHERE C.DataAcquisizione = (SELECT MIN(C2.DataAcquisizione)
                             FROM Copia AS C2)
```

Esercizio 8

Trovare quanti prestiti ha fatto ogni utente. Aggiungere anche il Nome di questi utenti.

```
SELECT CodUtente, Nome, COUNT(*) AS NumPrestiti
FROM Prestito
JOIN Utente
ON Codice = CodUtente
GROUP BY CodUtente, Nome
```

ATTENZIONE: serve raggruppare SIA per nome (che è richiesto in target list) SIA per codice utente (che è necessario per non raggruppare utenti con nome uguale)

Esercizio 9

La query precedente conta i prestiti, non i «libri» presi in prestito. Quanti diversi libri (isbn) ha preso ogni utente? (se ne indichi anche il nome)

```
SELECT CodUtente, Nome, COUNT(DISTINCT Isbn)
FROM Prestito P
JOIN Utente ON Codice = CodUtente
JOIN Copia C ON C.Collocazione = P.Collocazione
GROUP BY CodUtente, Nome
```

Esercizio 10

Quando è stato pubblicato il libro giallo più vecchio di ciascuna casa editrice che abbia pubblicato almeno 100 gialli diversi?

```
SELECT MIN(AnnoPub), CasaEd
FROM DatiLibro
WHERE Genere = "Giallo"
GROUP BY CasaEd
HAVING COUNT(*) >= 100
```

ATTENZIONE: La clausola where viene valutata per ogni riga La clausola having viene valutata per ogni gruppo: "pubblicare almeno 100 gialli diversi" deve essere valutata considerando i gruppi, per cui la funzione aggregata deve comparire all'interno della clausola having.

Esercizio 11

Trovare i libri che hanno almeno due copie che sono state acquisite il 01/05/2022 e il 01/06/2022

```
SELECT DISTINCT(Isbn)
FROM Copia
WHERE DataAcquisizione = 01/05/2022
```

INTERSECT

```
SELECT DISTINCT(Isbn)
FROM Copia
WHERE DataAcquisizione = 01/06/2022
```

Non si possono utilizzare né l'AND (darebbe insieme vuoto) né l'OR (nel risultato vi sarebbero anche copie acquisite solo in una delle due date).

Esercizio 12

Elencare tutti i nomi degli autori dei libri e degli utenti della biblioteca

```
SELECT Nome
FROM Utente
```

UNION

```
SELECT PrimoAut AS Nome
FROM DatiLibro
```

Attenzione: usando l'UNION si effettua automaticamente un DISTINCT sul risultato finale. Per tenere tutte le righe, si userebbe UNION ALL

Esercizio 13

Elencare tutti i nomi degli autori dei libri che non sono anche nomi di utenti della biblioteca

```
SELECT PrimoAut AS Nome
FROM DatiLibro
```

EXCEPT

```
SELECT Nome
FROM Utente
```

Attenzione: mentre con UNION e INTERSECT l'ordine delle due query era irrilevante con l'except è importante (come per la differenza in algebra)

Esercizio 14

Trovare gli utenti che non hanno comunicato il numero di telefono quando si sono iscritti ma hanno inserito l'indirizzo di casa

```
SELECT Codice, Nome, Cognome
FROM Utente
WHERE Telefono IS NULL AND Indirizzo IS NOT NULL
```

Esercizio 15

Trovare il numero di telefono degli utenti che hanno preso in prestito almeno un libro nel 2013

```
SELECT Codice , Telefono
FROM Utente
WHERE Codice IN (SELECT CodUtente
                  FROM Prestito
                  WHERE DataPrestito BETWEEN 01-01-2013
                  AND 31-12-2013)
```

Oppure con l'ANY:

```
SELECT Codice , Telefono
FROM Utente
WHERE Codice = ANY(SELECT CodUtente
                   FROM Prestito
                   WHERE DataPrestito BETWEEN 01-01-2013
                   AND 31-12-2013)
```

Esercizio 16

Trovare utenti che hanno preso in prestito libri di tutti i generi

```
SELECT Codice , Nome , Cognome
FROM Utente U
WHERE NOT EXISTS(
    SELECT *
    FROM DatiLibro D
    WHERE NOT EXISTS(
        SELECT *
        FROM Prestito P JOIN Copia C ON P.Coll = C.Coll
        JOIN DatiLibro D1 ON C.Isbn = D1.Isbn
        WHERE CodUtente = U.Codice
        AND Genere = D.Genere))
```

Soluzione alternativa 2:

```
SELECT Codice , Nome , Cognome
FROM Utente U
WHERE NOT EXISTS (
    SELECT *
    FROM DatiLibro
    WHERE Genere NOT IN (
        SELECT Genere
        FROM Prestito P JOIN Copia C on P.Coll = C.Coll
        JOIN DatiLibro D1 on C.Isbn = D1.Isbn
        WHERE CodUtente = U.Codice ) )
```

Soluzione alternativa 3:

```
SELECT Codice , Nome , Cognome
FROM Utente U
WHERE NOT EXISTS (
```

```
SELECT *
FROM DatiLibro
WHERE (Genere, U.Codice) NOT IN (
    SELECT Genere, CodUtente
    FROM Prestito P JOIN Copia C on P.Coll = C.Coll
    JOIN DatiLibro D1 on C.Isbn = D1.Isbn ))
```

Soluzione alternativa 4: sfruttare i conteggi.

```
SELECT Codice, Nome, Cognome
FROM Utente
JOIN Prestito P on Codice = CodUtente
JOIN Copia C on P.Coll = C.Coll
JOIN DatiLibro D on C.Isbn = D.Isbn
GROUP BY Codice, Nome, Cognome
HAVING COUNT(DISTINCT Genere)=( SELECT COUNT(DISTINCT Genere)
                                FROM Datilibro )
```