

A) Sia dato il seguente schema di base di dati:

UTENTI(Codice_Utente, Nome, Cognome, Eta, Sesso, Indirizzo, Telefono, Categoria_Utenti)
CATEGORIE(Codice_Categoria, Nome)
GENERI(Codice_Genere, Nome)
LIBRI(Codice_Libro, Titolo, Autore, Editore, Scaffale, Cod_Genere, Num_Pagine, Anno_Pub)
PRESTITI(Cod_Utente, Data, Durata, Cod_Libro)

con i seguenti vincoli di integrità referenziale:

UTENTI.Categoria_Utenti → CATEGORIE.Codice_Categoria

LIBRI.Cod_Genere → GENERI.Codice_Genere

PRESTITI.Cod_Libro → LIBRI.Codice_Libro

PRESTITI.Cod_Utente → UTENTI.Codice_Utente

Scrivere ed eseguire le query SQL per trovare:

- a. Codice e titolo dei libri pubblicati dall'editore Mondadori.

```
SELECT Codice_Libro, Titolo
FROM Libri
WHERE Editore = 'Mondadori'
```

- b. Cognome e nome degli utenti appartenenti alla categoria 'Studenti' che hanno più di 50 anni

```
SELECT Codice_Libro, Titolo
FROM Utenti U JOIN Categorie C ON U.Cod_Categoria = C.Codice_Categoria
WHERE C.Nome='Studenti'
```

- c. Cognome e nome degli utenti cui sono stati prestati libri pubblicati nel 2016.

```
SELECT DISTINCT U.Cognome, U.Nome
FROM Utenti U, Prestiti P, Libri L
WHERE U.CodiceUtente=P.Cod_Utente
AND L.Codice_libro = P.Cod_Libro
AND L.Anno_Pub = 2016
```

- d. Titolo dei libri che appartengono al genere 'Saggio'.

```
SELECT Titolo
FROM Libri L JOIN Generi G ON L.Cod_Genere = G.Codice_Genere
WHERE G.Nome='Saggio'
```

- e. Nome e cognome dei utenti di nome Giuseppe che hanno ottenuto il prestito di libri che hanno più di 500 pagine.

```
SELECT DISTINCT U.Nome U.Cognome
FROM Utenti U, Prestiti P, Libri L
WHERE U.CodiceUtente=P.Cod_Utente
AND L.Codice_libro = P.Cod_Libro
AND U.Nome = 'Giuseppe'
AND L.Num_Pagine > 500
```

- f. L'elenco dei libri appartenenti al genere 'Romanzo' che hanno più pagine di almeno un libro di genere 'Storico'.

```
SELECT DISTINCT L.Titolo
FROM Libri L1 JOIN Generi G1 ON L1.Cod_Genere = G1.Codice_Genere,
     Libri L2 JOIN Generi G2 ON L2.Cod_Genere = G2.Codice_Genere
WHERE G1.Nome='Romanzo'
AND    G2.Nome = 'Storico'
AND    L1.Num_Pagine > L2. Num_Pagine
```

oppure

```
SELECT DISTINCT L1.Titolo
FROM Libri L1 JOIN Generi G1 ON L1.Cod_Genere = G1.Codice_Genere,
WHERE G1.Nome='Romanzo'
AND L1.Num_Pagine > ANY ( SELECT L2. Num_Pagine
                        FROM Libri L2 JOIN Generi G2 ON L2.Cod_Genere = G2.Codice_Genere
                        WHERE G2.Nome='Storico')
```

- g. Quante volte sono stati prestati libri che hanno come autore 'Manzoni'

```
SELECT COUNT(*)
FROM Prestiti P, Libri L
WHERE L.Codice_libro = P.Cod_Libro
AND L.Autore='Manzoni'
```

- h. Quante volte è stato prestato ciascun libro che ha come autore 'Manzoni'.

```
SELECT L.Titolo, COUNT(*)
FROM Prestiti P, Libri L
WHERE L.Codice_libro = P.Cod_Libro
AND L.Autore='Manzoni'
GROUP BY L.Titolo
```

- i. Per ciascun genere, il numero medio di pagine dei libri che gli appartengono.

```
SELECT G.Nome, AVG(NumPagine) AS NPagMedio
FROM Generi G, Libri L
WHERE L.Cod_Genere = G.Codice_Genere
GROUP BY G.Nome
```

- j. Per ogni età, il numero medio di pagine dei libri di genere 'Avventura' presi in prestito.

```
SELECT U.Eta, AVG(L.NumPagine)
FROM Utenti U, Prestiti P, Libri L, Generi G
WHERE U.CodiceUtente=P.Cod_Utente
AND L.Codice_libro = P.Cod_Libro
AND G.CodiceGenere = L.CodGenere
AND G.Nome='Avventura'
GROUP BY U.Eta
```

- k. Il numero di prestiti totali, anno per anno (usare la funzione YEAR(Data) per estrarre l'anno da una data).

```
SELECT YEAR(P.Data) AS Anno, COUNT(*) AS TotalePrestiti
FROM Prestiti P
GROUP BY YEAR(P.Data)
ORDER BY Anno
```

- l. Il numero di prestiti totali, anno per anno, per i diversi generi.

```
SELECT G.Nome, YEAR(P.Data), COUNT(*) AS TotalePrestiti
FROM Prestiti P, Libri L, Generi G
WHERE L.Codice_libro = P.Cod_Libro
AND G.CodiceGenere = L.CodGenere
GROUP BY G.Nome, YEAR(P.Data)
```

- m. I generi per i quali, nel 2005, si sono avuti più di 50 prestiti.

```
SELECT G.Nome, COUNT(*) AS TotalePrestiti
FROM Prestiti P, Libri L, Generi G
WHERE L.Codice_libro = P.Cod_Libro
AND G.CodiceGenere = L.CodGenere
GROUP BY G.Nome
HAVING TotalePrestiti > 50
```

- n. Elenco dei libri di Camilleri, in ordine decrescente di numero di prestiti.

```
SELECT L.Titolo, COUNT(*) AS TotalePrestiti
FROM Prestiti P, Libri L
WHERE L.Codice_libro = P.Cod_Libro
AND L.Autore = 'Camilleri'
GROUP BY L.Titolo
ORDER BY TotalePrestiti DESC
```

- o. In quale scaffale sono localizzati più libri.

```
SELECT L.Scaffale S1, COUNT(*) AS NLibri
FROM Libri L
GROUP BY S1
HAVING NLibri >= ALL (SELECT COUNT(*)
                      FROM Libri L2
                      GROUP BY L2.Scaffale)
```

- p. La categoria di utenti che nel 2015 ha preso in prestito meno libri.

```
SELECT C.Nome, YEAR(P.Data) AS AnnoPrestito, COUNT(*) AS TotalePrestiti
FROM Prestiti P, Utenti U, Categorie C
WHERE U.Codice_Utente = P.Cod_Utente
AND C.Codice_Categoria = U.Cod_Categoria
AND AnnoPrestito = 2015
GROUP BY C.Nome, YEAR(P.Data)
HAVING TotalePrestiti <= ALL (
    SELECT COUNT(*) AS TotalePrestitiGenere
    FROM Prestiti P1, Utenti U1
    WHERE U1.Codice_Utente = P1.Cod_Utente
    AND YEAR(P1.Data) = 2015
    GROUP BY U1.Cod_Categoria
)
```

- q. Per ogni genere, il libro che nel 2014 è risultato il più richiesto.

```
SELECT G.Nome, COUNT(*) AS TotalePrestiti
FROM Prestiti P, Libri L, Generi G
WHERE L.Codice_libro = P.Cod_Libro
AND G.CodiceGenere = L.CodGenere
GROUP BY G.Nome, L.Codice_Libro
HAVING TotalePrestiti >= ALL (
    SELECT COUNT(*)
    FROM Prestiti P1, Libri L1
    WHERE L1.Codice_libro = P1.Cod_Lib
    AND L1.Cod_Genere = L1.Codice_Genere
    GROUP BY L1.Codice_Libro
)
```

- r. L'utente di 35 anni che ha 'letto' di più (cioè che ha preso in prestito libri per il massimo numero totale di pagine).

```
SELECT U.Nome, SUM(L.NumPagine) AS PagineLette
FROM Prestiti P, Libri L, Utenti U
WHERE L.Codice_libro = P.Cod_Libro
AND U.Codice_Utente = P.Cod_Utente
AND U.Eta=35
GROUP BY U.Codice_Utente
HAVING PagineLette >= ALL (
    SELECT SUM(L1.NumPagine)
    FROM Prestiti P1, Libri L1, Utenti U1
    WHERE L1.Codice_libro = P1.Cod_Libro
    AND U1.Codice_Utente = P1.Cod_Utente
    AND U1.Eta=35
    GROUP BY U1.Codice_Utente
)
```

- s. Gli utenti che hanno preso in prestito più di 50 libri di genere 'Avventura' ma non hanno letto alcun libro di genere 'Storico'.

```
SELECT U.Nome
FROM Utenti
WHERE U.Codice_Utente IN (
    SELECT U.Codice_Utente
    FROM Prestiti P, Libri L, Generi G, Utenti U
    WHERE L.Codice_libro = P.Cod_Libro
    AND P.Cod_Utente = U.CodiceUtente
    AND G.CodiceGenere = L.CodGenere
    AND G.Nome='Avventura'
    GROUP BY U.CodiceUtente
    HAVING COUNT(*) > 50
)
AND NOT IN (
    SELECT U1.Codice_Utente
    FROM Prestiti P1, Libri L1, Generi G1, Utenti U1
    WHERE L1.Codice_libro = P1.Cod_Libro
    AND P1.Cod_Utente = U1.CodiceUtente
    AND G1.CodiceGenere = L1.CodGenere
    AND G1.Nome='Storico'
    GROUP BY U.CodiceUtente
)
```

- t. Supponendo che Codice_Libro sia diverso per due copie diverse di uno stesso testo, trovare tutti i “doppioni” nel catalogo.

```
SELECT L1.Codice_Libro, L2.Codice_Libro
FROM Libri L1, Libri L2
WHERE L1.Titolo = L2. Titolo
AND L1.Codice_Libro > L2.Codice_Libro
```

- u. Definire una vista Saggi che contenga le tuple contenute nel risultato della query d.

```
CREATE VIEW Saggi(Titolo) AS
(
    SELECT Titolo
    FROM Libri L JOIN Generi G ON L.Cod_Genere = G.Codice_Genere
    WHERE G.Nome='Saggio'
)
```

- v. Formulare le query q., r., definendo un’opportuna vista che poi semplifichi il codice dell’interrogazione. Osservare come cambia la leggibilità del codice rispetto alla query senza la vista.

- (q) CREATE VIEW GBestsellers(Genere, Titolo, Nprestiti) AS

```
(
    SELECT G1.Nome, L1.Titolo, COUNT(*)
    FROM Prestiti P1, Libri L1, Generi G1
    WHERE L1.Codice_Libro = P1.Cod_Libro
    AND L1.Cod_Genere = G1.Codice_Genere
    GROUP BY L1.Codice_Libro, G1.Nome
)
```

```
SELECT GB.Nome, GB.NPrestiti
FROM GBestellers GB
WHERE GB.NPrestiti= (
    SELECT MAX(Nprestiti)
    FROM GBestellers GB1
    WHERE GB1.Genere=GB.Genere
)
```

- (r) CREATE VIEW U35(Codice_Trentacinquenne, Nome, Cognome, PagineLette) AS

```
(
    SELECT U1.Codice_Utente, U1.Nome, U1.Cognome, SUM(L1.NumPagine)
    FROM Prestiti P1, Libri L1, Utenti U1
    WHERE L1.Codice_libro = P1.Cod_Libro
    AND U1.Codice_Utente = P1.Cod_Utente
    AND U1.Eta=35
    GROUP BY U1.Codice_Utente
)
```

```
SELECT *
FROM U35
WHERE U35.PagineLette = (
    SELECT MAX(PagineLette)
    FROM U35B
)
```

- w. Eliminare le tuple relative ai libri dell’editore Einaudi dalla tabella Libri.

```
DELETE FROM Libri
WHERE Libri.Editore='Einaudi'
```