

Soluzione esercizi interrogazioni SQL

ESERCIZIO 1

Schema

UTENTE (Codice, Nome, Cognome, Indirizzo, Telefono)

PRESTITO (Collocazione, CodUtente, DataPrestito, DataRest)

COPIA (Collocazione, ISBN, DataAcq)

DATILIBRO (ISBN, Titolo, AnnoPub, CasaEd, PrimoAut, Genere)

- A. Trovare i titoli dei libri di informatica prestati nel giugno '02 in ordine alfabetico (ogni titolo compaia una volta).

```
SELECT distinct D.Titolo
FROM Datilibro D, Copia C, Prestito P
WHERE D.Genere='Informatica' and C.isbn = D.isbn and
P.Collocazione = C.Collocazione and P.DataPrestito>='01/06/2002' and
P.DataPrestito<'01/07/2002'
ORDER BY D.Titolo asc
```

- B. Estrarre nome, cognome e codice degli utenti che non hanno mai preso in prestito libri di informatica.

```
CREATE VIEW UtentiDiInf (Codice) as
SELECT U.Codice
FROM ((Datilibro D join Copia C on C.isbn = D.isbn) join Prestito P on P.Collocazione =
C.Collocazione) join Utente U on U.Codice = P.CodUtente
WHERE D.Genere = 'Informatica'

SELECT U.Nome, U.Cognome, U.Codice
FROM Utente U
WHERE U.Codice not in (SELECT Codice FROM UtentiDiInf)
```

- C. Trovare il genere dei libri con il maggior numero di prestiti.

```
CREATE VIEW GeneriPrestati (Genere, NumeroPrestiti) as
SELECT D.Genere, count(*)
FROM (Datilibro D join Copia C on C.isbn = D.isbn) join Prestito P on P.Collocazione =
C.Collocazione
GROUP BY D.Genere
```

```
SELECT D.Genere
FROM GeneriPrestati D
WHERE D.NumeroPrestiti >= all (SELECT NumeroPrestiti FROM GeneriPrestati)
```

- D. Trovare gli utenti che hanno preso libri di tutti i generi.

```
CREATE VIEW PrestitiEff (codice, genere) as
SELECT distinct U.Codice, D.Genere
FROM ((Datilibro D join Copia C on C.isbn = D.isbn) join Prestito P on P.Collocazione =
C.Collocazione) join Utente U on
U.Codice=P.CodUtente
```

```
CREATE VIEW TotaleGeneri (numero) as
SELECT count(distinct Genere)
FROM Datilibro
```

```
SELECT Codice
FROM PrestitiEff
GROUP BY codice
HAVING count(*) = (SELECT numero FROM TotaleGeneri)
```

ESERCIZIO 2

Schema

CLIENTE(Id, Cognome, Nome, NumNoleggi, Tipologia)

FILM(Titolo, Regista, Genere, Durata)

CASSETTA(Id, Titolo)

DVD(Id, Titolo)

NOLEGGIO(IdCassODvd, CodCliente, DataPrestito, DataRestituzione)

- A. Estrarre in SQL tutti i film di cui sono stati noleggiati più volte i DVD che le videocassette.

```
CREATE VIEW Dvd (Titolo, Numero) AS
SELECT Titolo, COUNT(*)
FROM DVD JOIN Noleggio ON Id = IdCassODvd
GROUP BYTitolo
```

```
CREATE VIEW Cass (Titolo, Numero) AS
SELECT Titolo, COUNT(*)
FROM Cassetta JOIN Noleggio ON Id = IdCassODvd
GROUP BYTitolo
```

```
SELECT Titolo
FROM Dvd JOIN Cass ON DVD.Titolo = Cass.Titolo
WHERE Dvd.Numero > Cass.Numero
```

- B. Estrarre in SQL tutti i film le cui cassette sono state noleggate più di venti volte nel mese di Gennaio ma non presenti in formato DVD.
***Fare in modo che vengano stampati i titoli in ordine opposto a quello alfabetico e che vengano visualizzati solo quelli in posizione 2 e 3 della lista

```
SELECT Titolo
FROM Cassetta JOIN Noleggio ON Id = IdCassODvd
WHERE DataPrestito BETWEEN 1/1/2012 AND 31/1/2012 AND Titolo NOT IN (
SELECT Titolo FROM DVD )
GROUP BYTitolo
HAVING COUNT(*) > 20
ORDER BY TITOLO desc
LIMIT 2 OFFSET 1
```

- C. Determinare il cognome e il nome dei clienti che hanno noleggiato tutti i film (in formato cassetta o DVD).

```
CREATE VIEW Supporto(Id, Titolo) AS
SELECT Id, Titolo
FROM Cassetta
UNION
SELECT Id, Titolo
FROM DVD
```

```
SELECT Cognome, Nome
FROM Cliente AS C
WHERE NOT EXISTS
    ( SELECT *
      FROM Film AS F
      WHERE NOT EXISTS
          ( SELECT *
            FROM Noleggio JOIN Supporto ON IdCassODvd = Id
            WHERE Titolo = F.Titolo AND CodCliente = C.Id ) )
```

- D. Determinare il nome e cognome dei clienti che hanno noleggiato DVD solo di genere giallo.

```
SELECT Cognome, Nome
FROM Cliente AS C JOIN Noleggio AS N ON C.Id = CodCliente JOIN DVD AS D ON
IdCassODvd = D.Id
WHERE NOT EXISTS
    ( SELECT *
      FROM Noleggio AS N1 JOIN DVD AS D1 ON D1.Id =
      N1.IdCassODvd JOIN Film AS F1 ON D1.Titolo = F1.Titolo
      WHERE C.Id = N1.CodCliente AND Genere <> 'Giallo' )
```

ESERCIZIO 3

Schema

`prereq(id_corso, id_prereq)`

Trovare tutti i pre-requisiti del corso "ID0018"

```
WITH RECURSIVE c_prereq(id_corso, id_prereq) AS (  
    SELECT id_corso, id_prereq  
    FROM prereq  
    WHERE id_corso = "ID0018"  
    UNION  
    SELECT prereq.id_prereq, c_prereq.id_corso  
    FROM prereq P, c_prereq  
    WHERE P.id_corso= c_prereq.id_prereq AND P.id_corso =  
    "ID0018"  
)  
SELECT *  
FROM c_prereq;
```

ESERCIZIO 4

Schema

`SQUADRA(nome, media_punti)`

Costruire una tabella che mostri il ranking delle squadre
→ viene concesso l'utilizzo della sola funzione count()

```
SELECT nome, (1 +  
    (SELECT count(*)  
    FROM SQUADRA B  
    WHERE B.media_punti > A.media_punti)) AS rank  
FROM SQUADRA A  
ORDER BY rank
```

ESERCIZIO 5

Schema

AEROPORTO(Id, Città, Nazione, NumPiste)

VOLO(NumeroVolo, GiornoSett, IdAeropPartenza, IdAeropArrivo, Compagnia, OraPartenza, OraArrivo, CodAereo)

AEREO(Codice, Tipo, NumPasseggeri)

- A. Esprimere in SQL la query che estrae la coppia di città tra cui viene offerta la maggiore capacità complessiva di trasporto passeggeri.

```
CREATE VIEW Trasporto (Da, A, Passeggeri) AS
SELECT A1.Città, A2.Città, SUM(NumPasseggeri)
FROM Volo JOIN Aereo ON CodAereo = Codice JOIN Aeroporto AS A1
ON IdAeropPartenza = A1.Id JOIN Aeroporto AS A2 ON IdAeropArrivo
= A2.Id
GROUP BY A1.Città, A2.Città
```

```
CREATE VIEW Totali (CUno, CDue, Passeggeri) AS
SELECT T1.Da, T1.A, T1.Passeggeri+T2.Passeggeri
FROM Trasporto AS T1 JOIN Trasporto AS T2 ON T1.Da = T2.A AND
T2.Da = T1.A
WHERE T1.Da < T1.A
UNION
SELECT Da, A, Passeggeri
FROM Trasporto
WHERE Da NOT IN (
                SELECT A
                FROM Trasporto )
```

```
SELECT DISTINCT CUno, CDue
FROM Totali
WHERE Passeggeri = (
                    SELECT MAX(Passeggeri)
                    FROM Totali )
```

- B. Estrarre in SQL le compagnie che servono la città di Bergamo con più di settanta voli alla settimana, usando aerei di un solo tipo per tutti i voli.

```
CREATE VIEW Bg (Compagnia) AS
SELECT Compagnia
FROM Volo JOIN Aeroporto AS A1 ON IdAeropPartenza = A1.Id
JOIN Aeroporto AS A2 ON IdAeropArrivo = A2.Id
WHERE A1.Città = 'Bergamo' OR A2.Città = 'Bergamo'
GROUP BY Compagnia
HAVING COUNT(*) ≥ 70

SELECT Compagnia
FROM Volo AS V JOIN Aereo AS A ON CodAereo = Codice
JOIN Bg ON V.Compagnia = Bg.Compagnia
WHERE NOT EXISTS (
    SELECT *
    FROM Volo AS V1 JOIN Aereo AS A1 ON CodAereo = Codice
    WHERE A1.Tipo <> A.Tipo AND V1.Compagnia = V.Compagnia
)
```