

Linguaggio SQL: Esercizi Parte 2

- 1. Sia dato il seguente schema relazionale (le chiavi primarie sono sottolineate, gli attributi opzionali sono indicati con “*”)**

GARA (CodG, Luogo, Data, Disciplina)

ATLETA (CodA, Nome, Nazione, DataNascita)

PARTECIPAZIONE (CodG, CodA, PosizioneArrivo, Tempo)

- a) Trovare il nome e la data di nascita degli atleti italiani che non hanno partecipato a nessuna gara di discesa libera.

```
SELECT Nome, DataNascita
FROM ATLETA
WHERE Nazione='Italia'
AND CodA NOT IN
    (SELECT CodA
     FROM PARTECIPAZIONE P, GARA G
     WHERE Disciplina='discesa libera' AND P.CodG=G.CodG)
```

- b) Trovare le nazioni per cui concorrono almeno 5 atleti nati prima del 1980, ciascuno dei quali abbia partecipato ad almeno 10 gare di sci di fondo.

```
SELECT Nazione
FROM ATLETA
WHERE DataNascita < 1/1/1980 AND CodA IN
    (SELECT CodA
     FROM GARA G, PARTECIPAZIONE P
     WHERE Disciplina = 'Fondo' AND G.CodG=P.CodG
     GROUP BY CodA
     HAVING COUNT(*) >=10)
GROUP BY Nazione
HAVING COUNT(*) >=5
```

2. Sia dato il seguente schema relazionale (le chiavi primarie sono sottolineate, gli attributi opzionali sono indicati con “*”)

EDITORE (CodE, NomeEditore, Indirizzo, Città)
 PUBBLICAZIONE (CodP, Titolo, NomeAutore, CodE)
 LIBRERIA (CodL, NomeLibreria, Indirizzo, Città)
 VENDITA (CodP, CodL, Data, CopieVendute)

- a) Trovare il nome delle librerie in cui non è stata venduta nessuna pubblicazione di editori con sede a Torino.

```
SELECT NomeLibreria
FROM LIBRERIA L
WHERE CodL NOT IN
    (SELECT CodL
     FROM VENDITA V, PUBBLICAZIONE P, EDITORE E
     WHERE Citta='Torino' AND P.CodE=E.CodE AND V.CodP=P.CodP)
```

- b) Trovare il nome degli editori per cui almeno 10 pubblicazioni sono state vendute nel 2002 nelle librerie di Roma in più di 2.000 copie

```
SELECT NomeEditore
FROM EDITORE E, PUBBLICAZIONE P
WHERE E.CodE = P.CodE
AND CodP IN
    (SELECT CodP
     FROM VENDITA V, LIBRERIA L
     WHERE Citta='Roma' AND Data>=1/1/2002 AND Data<=31/12/2002
     AND V.CodL = L.CodL
     GROUP BY CodP
     HAVING SUM(CopieVendute)>2000)
GROUP BY E.CodE, NomeEditore
HAVING COUNT(*)>=10;
```

3. Sia dato il seguente schema relazionale (le chiavi primarie sono sottolineate, gli attributi opzionali sono indicati con “*”)

QUIZ (CodQuiz, Argomento, Punteggio)
 STUDENTE (Matricola, Nome, Indirizzo, Città)
 RISULTATO_TEST (Matricola, CodQuiz, RispostaCorretta)

- a) Trovare il nome degli studenti che non hanno risposto correttamente a nessun quiz di matematica.

```
SELECT Nome
FROM STUDENTE
WHERE Matricola NOT IN
    (SELECT Matricola
     FROM RISULTATO_TEST R, QUIZ Q
     WHERE RispostaCorretta='si' AND Argomento='matematica'
     AND R.CodQuiz = Q.CodQuiz)
```

- b) Trovare il nome degli studenti di Torino che hanno conseguito il punteggio massimo possibile nei quiz di matematica.

```
SELECT Nome
FROM STUDENTE S, RISULTATO_TEST R, QUIZ Q
WHERE Citta='Torino' AND RispostaCorretta='si' AND Argomento='matematica'
AND S.Matricola = R.Matricola AND Q.CodQuiz = R.CodQuiz
GROUP BY S.Matricola, Nome
HAVING SUM(Punteggio) =
    (SELECT SUM(Punteggio)
     FROM QUIZ
     WHERE Argomento='matematica')
```

4. Sia dato il seguente schema relazionale (le chiavi primarie sono sottolineate, gli attributi opzionali sono indicati con “*”)

AEREI (Matr, Modello, NumPosti)

ORARIO (Sigla, ParteDa, Destinaz, OraPart, OraArr)

VOLI (Sigla, Matr, Data, PostiPren)

- a) Trovare la sigla e l'ora di partenza dei voli in partenza da Milano per Napoli il 1 ottobre 1993, che dispongono ancora di posti liberi e la cui durata (differenza tra l'ora di arrivo e l'ora di partenza) è inferiore alla durata media dei voli da Milano a Napoli.

```
SELECT O.Sigla, OraPart
FROM ORARIO O, VOLO V, AEREO A
WHERE O.ParteDa='Milano' AND O.Destinaz='Napoli' AND V.Data=1/10/2010
AND O.Sigla = V.Sigla AND V.Matr = A.Matr AND V.PostiPren < A.NumPosti
AND (O.OraArr-OraPart) < (SELECT AVG(OraArr-OraPart)
                             FROM ORARIO O2
                             WHERE O2.ParteDa='Milano' AND O2.Destinaz='Napoli');
```

5. Sia dato il seguente schema relazionale (le chiavi primarie sono sottolineate, gli attributi opzionali sono indicati con “*”)

MECCANICO (MatrM, NomeM)

SA-RIPARARE (MatrM, TipoGuasto)

EFFETTUA-RIPARAZIONE (CodR, MatrM, Targa, Data, Durata, TipoGuasto)

- a) Trovare il nome dei meccanici che hanno effettuato almeno una riparazione di un guasto che non sapevano riparare.

```
SELECT NomeM
FROM MECCANICO M, EFFETTUA-RIPARAZIONE ER
WHERE M.MatrM = ER.MatrM
AND (MatrM, TipoGuasto) NOT IN (SELECT (MatrM, TipoGuasto)
                                  FROM SA-RIPARARE);
```

Soluzione alternativa

```
SELECT NomeM  
FROM MECCANICO M, EFFETTUA-RIPARAZIONE ER  
WHERE M.MatrM = ER.MatrM  
AND NOT EXISTS (SELECT *  
                  FROM SA-RIPARARE SR  
                  WHERE SR.MatrM = ER.MatrM AND SR.TipoGuasto = ER.TipoGuasto);
```

- b) Per le autovetture per cui sono state necessarie riparazioni effettuate da almeno 3 meccanici diversi nello stesso giorno, visualizzare la targa dell'autovettura, la data delle riparazioni e i tipi di guasto che si sono verificati, ordinando il risultato in ordine crescente di targa e decrescente di data.

```
SELECT Targa, Data, TipoGuasto  
FROM EFFETTUA-RIPARAZIONE  
WHERE (Targa, Data) IN (SELECT (Targa, Data)  
                        FROM EFFETTUA-RIPARAZIONE  
                        GROUP BY Targa, Data  
                        HAVING COUNT (DISTINCT MatrM)>=3)  
ORDER BY Targa ASC, Data DESC;
```

6. Sia dato il seguente schema relazionale (le chiavi primarie sono sottolineate, gli attributi opzionali sono indicati con “*”)

SALA_RIUNIONI (CodS, NumeroMaxPosti, Proiettore)
 PRENOTAZIONE_SALA (CodS, Data, OraInizio, OraFine, CodDip)
 DIPENDENTE (CodDip, Nome, Cognome, DataNascita, Città)

- a) Visualizzare il codice e il numero massimo di posti delle sale dotate di proiettore che sono state prenotate almeno 15 volte per riunioni che iniziano prima delle ore 15:00, ma non sono mai state prenotate per riunioni che cominciano dopo le ore 20:00.

```
SELECT S.CodS, NumeroMaxPosti
FROM PRENOTAZIONE_SALA PS, SALA S
WHERE PS.CodS = S.CodS
AND Proiettore='si' AND OraInizio<15:00
AND S.CodS NOT IN (Select CodS
                    FROM PRENOTAZIONE_SALA
                    WHERE OraInizio>20:00)
GROUP BY S.CodS, NumeroMaxPosti
HAVING COUNT(*)>=15;
```

- b) Visualizzare per ogni sala il codice della sala, il numero massimo di posti e il numero di prenotazioni considerando solo l'ultima data in cui la sala è stata prenotata.

```
SELECT S.CodS, NumeroMaxPosti, COUNT(*)
FROM PRENOTAZIONE_SALA PS1, SALA S
WHERE PS1.Cods=S.CodS
AND DATA = (SELECT MAX(Data)
            FROM PRENOTAZIONE_SALA PS2
            WHERE PS2.CodS=PS1.CodS)
GROUP BY S.CodS, NumeroMaxPosti;
```

Soluzione alternativa

```
SELECT S.CodS, NumeroMaxPosti, COUNT(*)  
FROM PRENOTAZIONE_SALA PS1, SALA S, (SELECT PS2.CodS, MAX(Data) AS MaxData  
                                         FROM PRENOTAZIONE_SALA PS2  
                                         GROUP BY PS2.CodS) AS UltimaPrenotazione  
WHERE PS1.CodS=S.CodS  
AND PS1.CodS = UltimaPrenotazione.CodS AND PS1.Data = UltimaPrenotazione.MaxData  
GROUP BY S.CodS, NumeroMaxPosti
```

7. Sia dato il seguente schema relazionale (le chiavi primarie sono sottolineate, gli attributi opzionali sono indicati con “*”)

CLIENTE (Cod-Cli, Nome)

CONTO (Cod-Conto, saldo, agenzia, stato)

CONTO-CLIENTE (Cod-Conto, Cod-Cli)

- a) Trovare tutte le agenzie che hanno almeno un cliente titolare da solo (senza cointestatari) di un unico conto corrente (cliente a cui non è intestato nessun altro conto corrente).

```
SELECT DISTINCT C.Agenzia
FROM CONTO C, CONTO-CLIENTE CL
WHERE C.Cod-Conto=CL.Cod-Conto
AND NOT EXISTS (SELECT *
    FROM CONTO-CLIENTE CL2
    WHERE CL2.Cod-Conto=CL.Cod-Conto
    AND CL2.Cod-Cli<>CL.Cod-Cli)
AND NOT EXISTS (SELECT *
    FROM CONTO-CLIENTE CL3
    WHERE CL3.Cod-Cli=CL.Cod-Cli
    AND CL3.Cod-Conto<>CL.Cod-Conto);
```

Oppure

```
SELECT DISTINCT C.Agenzia
FROM CONTO C, CONTO-CLIENTE CL
WHERE C.Cod-Conto=CL.Cod-Conto
AND C.Cod-Conto IN (SELECT CL2.Cod-Conto
    FROM CONTO-CLIENTE CL2
    GROUP BY CL2.Cod-Conto
    HAVING COUNT (*) = 1)
AND Cod-Cli IN (SELECT Cod-Cli
    FROM CONTO-CLIENTE CL3
    GROUP BY CL3.Cod-Cli
    HAVING COUNT (*) = 1)
```

8. Sia dato il seguente schema relazionale (le chiavi primarie sono sottolineate, gli attributi opzionali sono indicati con “*”)

CONTRIBUENTE (CodFiscale, Nome, Via, Città)
 DICHIARAZIONE (CodDichiarazione, Tipo, Reddito)
 PRESENTA (CodFiscale, CodDichiarazione, Data)

- b) Visualizzare codice, nome e media dei redditi dichiarati dal 1990 in poi per i contribuenti tali che il massimo reddito da loro dichiarato dal 1990 in poi sia superiore alla media dei redditi calcolata su tutte le dichiarazioni nel database.

```
SELECT C.CodFiscale, C.Nome, AVG(Reddito)
FROM CONTRIBUENTE C, DICHIARAZIONE D, PRESENTA P
WHERE C.CodFiscale=P.CodFiscale AND D.CodDichiarazione=P.CodDichiarazione
AND P.Data>1/1/2010
GROUP BY C.CodFiscale, C.Nome
HAVING MAX(D.Reddito) > (SELECT AVG(Reddito)
                             FROM DICHIARAZIONE)
```

9. Sia dato il seguente schema relazionale (le chiavi primarie sono sottolineate, gli attributi opzionali sono indicati con “*”)

PERSONA (Nome, Sesso, Età)
 GENITORE (Nome-Gen, Nome-Figlio)

- a) Trovare il nome di tutte le persone con età inferiore a 10 anni che sono figli unici.

```
SELECT DISTINCT Nome
FROM PERSONA P, GENITORE G
WHERE P.Nome =G.Nome-Figlio
AND P.Età<10
AND P.Nome-Gen NOT IN (SELECT G1.Nome-Gen
                           FROM GENITORE G1
                           GROUP BY G1.Nome-Gen
                           HAVING COUNT (*) > 1);
```