Politecnico di Torino

SQL: Esercizi Parte 2

SQL: Esercizi Parte 2

Esercizio #1a,1b

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.

(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.

Esercizio #1a

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 P.CodG=G.CodG
                 AND G.Disciplina='discesa libera');
```

Esercizio #1b

GARA(CodG, Luogo, Data, Disciplina)

ATLETA(CodA, Nome, Nazione, DataNascita)

PARTECIPAZIONE(CodG, CodA,PosizioneArrivo, Tempo)

(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 PARTECIPAZIONE P,GARA G
                WHERE P.CodG=G.CodG AND Disciplina='fondo'
               GROUP BY CodA
               HAVING COUNT(*)>=10)
GROUP BY Nazione
HAVING COUNT(*)>=5;
```

Esercizio #2a,#2b

EDITORE(CodE, NomeEditore, Indirizzo, Citta)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Citta)

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.

(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.

Esercizio #2a

EDITORE(CodE, NomeEditore, Indirizzo, Citta)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Citta)

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 V.CodP=P.CodP AND P.CodE=E.CodE AND Citta='Torino');
```

Esercizio #2b

EDITORE(CodE, NomeEditore, Indirizzo, Citta)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Citta)

VENDITA(CodP, CodL, Data,CopieVendute)

(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 P1
WHERE P1.CodE=E.CodE
AND CodP IN
            (SELECT CodP FROM VENDITA V, LIBRERIA L
            WHERE V.CodL=L.CodL
            AND Data>='1/1/2002' AND Data<='31/12/2002’ AND L.Citta='Roma'
            GROUP BY CodP
            HAVING SUM(CopieVendute)>2000)
GROUP BY E.CodE, NomeEditore
HAVING COUNT(*)>=10;
```

Esercizio #3a,#3b

QUIZ(CodQuiz, Argomento, Punteggio)

STUDENTE(Matricola, Nome, Indirizzo, Citta)

RISULTATO TEST(Matricola, CodQuiz,RispostaCorretta)

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

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

Esercizio #3a

QUIZ(CodQuiz, Argomento, Punteggio)

STUDENTE(Matricola, Nome, Indirizzo, Citta)

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

Esercizio #3b

QUIZ(CodQuiz, Argomento, Punteggio)

STUDENTE(Matricola, Nome, Indirizzo, Citta)

RISULTATO TEST(Matricola, CodQuiz,RispostaCorretta)

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

Esercizio #4a

AEREI (Matr, Modello, NumPosti)

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

VOLI (Sigla, Matr, Data, PostiPren)

Trovare la sigla e l'ora di partenza dei voli in partenza da Milano per Napoli il 1 ottobre 2010, che dispongono ancora di posti liberi 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.

Esercizio #4a

AEREI (Matr, Modello, NumPosti)

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

VOLI (Sigla, Matr, Data, PostiPren)

Trovare la sigla e l'ora di partenza dei voli in partenza **da Milano per Napoli** il **1 ottobre 2010**, che **dispongono** ancora **di posti liberi** 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
WHERE O.ParteDa='Milano' AND O.Destinaz='Napoli’
AND O.Sigla  = V.Sigla
AND V.Data='1/10/2010’
AND V.PostiPren < A.NumPosti
AND (O.OraArr-OraPart)  < (SELECT AVG(OraArr-OraPart)
                           FROM ORARIO O2
                           WHERE O2.ParteDa='Milano' AND
                           O2.Destinaz='Napoli');
```

Esercizio #5a

MECCANICO(MatrM, NomeM)

SA-RIPARARE(MatrM, TipoGuasto)

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

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 (ER.MatrM, TipoGuasto) NOT IN (SELECT (SR.MatrM, TipoGuasto)
                                   FROM SA-RIPARARE SR);
```

Esercizio #5b

MECCANICO(MatrM, NomeM)

SA-RIPARARE(MatrM, TipoGuasto)

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

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;
```

Esercizio #6a

SALA RIUNIONI(CodS, NumeroMaxPosti, Proiettore)

PRENOTAZIONE_SALA(CodS, Data, OraInizio, OraFine, CodDip)

DIPENDENTE(CodDip, Nome, Cognome, DataNascita, Città)

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 PS1, SALA S
WHERE PS.Cods=S.CodS
AND Proiettore='si'
AND OraInizio<'15:00'
AND S.Cods NOT IN  (Select CodS
                    FROM PRENOTAZIONE_SALA PS2
                    AND OraInizio>'20:00')
GROUP BY S.CodS, NumeroMaxPosti
HAVING COUNT(*)>=15;
```

Esercizio #6b

SALA RIUNIONI(CodS, NumeroMaxPosti, Proiettore)

PRENOTAZIONE_SALA(CodS, Data, OraInizio, OraFine, CodDip)

DIPENDENTE(CodDip, Nome, Cognome, DataNascita, Città)

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

Esercizio #6b

SALA RIUNIONI(CodS, NumeroMaxPosti, Proiettore)

PRENOTAZIONE_SALA(CodS, Data, OraInizio, OraFine, CodDip)

DIPENDENTE(CodDip, Nome, Cognome, DataNascita, Città)

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,
                        (SELECT PS2.CodS, MAX(Data) AS MaxData
                         FROM PRENOTAZIONE_SALA PS2
                         GROUP BY PS2.Cods) AS LAST
WHERE PS.Cods=S.CodS
AND PS.CodS  = LAST.CodS
AND DATA = LAST.MaxData
GROUP BY S.CodS, NumeroMaxPosti;
```

Esercizio #7a

GUIDA (CodGuida, Nome, Cognome, Nazionalità)

TIPO-VISITA (CodTipoVisita, Monumento, Durata, Città)

GRUPPO (CodGR, NumeroPartecipanti, Lingua)

VISITA-GUIDATA-EFFETTUATA (CodGR, Data, OraI, CodTipoVisita, CodGuida)

Tra i monumenti per cui sono state effettuate almeno 10 visite guidate, visualizzare il monumento che è stato visitato complessivamente dal maggior numero di persone.

Esercizio #7a

Vedere soluzioni Esercizio #1a del blocco Esercizi SQL Parte 3

Esercizio #8a

RAGAZZO(CodFiscale, Nome, Cognome, DataNascita, CittàResidenza)

ATTIVITA'(CodAttività, NomeA, Descrizione, Categoria)

CAMPO-ESTIVO(CodCampo, NomeCampo, Città)

ISCRIZIONE-PER-ATTIVITA'-IN-CAMPO-ESTIVO(CodFiscale, CodAttività, CodCampo, DataIscrizione)

Visualizzare il nome e cognome del ragazzo che ha partecipato al maggior numero di campi estivi per l'attività della categoria «Tennis».

Esercizio #8a

Vedere soluzioni Esercizio #2b del blocco Esercizi SQL Parte 3

Esercizio #9a

CLIENTE(Cod-Cli,nome)

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

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

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).

Esercizio #9a

CLIENTE(Cod-Cli,nome)

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

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

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);
```

Esercizio #9a

CLIENTE(Cod-Cli,nome)

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

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

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 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)
```

Esercizio #9a – Soluzioni erreate

CLIENTE(Cod-Cli,nome)

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

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

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).

Selezionando i clienti che hanno un unico conto e poi conterebbe che questi conti hanno solo 1 intestatario. Tuttavia NON elimino i conti dove c'è un cointestatario con più conti.

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

Troveremmo i clienti che hanno UN SOLO CONTO tra i conti aventi UN UNICO INTESTATARIO.

```
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)
GROUP BY CL.Cod-Cli, C.Agenzia
HAVING COUNT(*) = 1
```

Cod-Cli	Cod-Conto
Co1	C1
Co1	C2
Co2	C2
Co3	C3

Esercizio #10b

CONTRIBUENTE(CodFiscale, Nome, Via, Citta)

DICHIARAZIONE(CodDichiarazione, Tipo, Reddito)

PRESENTA(CodFiscale, CodDichiarazione, Data)

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.

Esercizio #10b

CONTRIBUENTE(CodFiscale, Nome, Via, Citta)

DICHIARAZIONE(CodDichiarazione, Tipo, Reddito)

PRESENTA(CodFiscale, CodDichiarazione, Data)

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);
```

Esercizio #10b - correlazione

CONTRIBUENTE(CodFiscale, Nome, Via, Citta)

DICHIARAZIONE(CodDichiarazione, Tipo, Reddito)

PRESENTA(CodFiscale, CodDichiarazione, Data)

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'
AND (SELECT MAX(Reddito)
      FROM PRESENTA P1, DICHIARAZIONE D1
      WHERE P1.CodDichiarazione=D1.CodDichiarazione
      AND P1.CodFiscale=C.CodFiscale        Correlazione
      AND D1.Data>'1/1/2010')
                              > (SELECT AVG(REddito)
                                 FROM DICHIARAZIONE)
GROUP BY C.CodFiscale, C.Nome;
```

Esercizio #11a

PERSONA(Nome, Sesso, Età)

GENITORE(Nome-Gen, Nome-Figlio)

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
                    );
```