

Fondamenti di basi di dati
Docente: Alessandro Fiori
Titolo argomento:
Esercizi SQL

Esercizio 1

Dato lo schema relazionale costituito dalle tabelle (le chiavi primarie sono sottolineate):

RIVISTA (CodR, NomeR, Editore)
ARTICOLO (CodA, Titolo, Argomento, CodR)

esprimere in SQL le seguenti interrogazioni:

- (a) Trovare il codice e il nome delle riviste che hanno pubblicato almeno un articolo di argomento motociclismo.

```
SELECT R.CodR, NomeR
FROM RIVISTA R, ARTICOLO A
WHERE R.CodR=A.CodR
AND Argomento='motociclismo';
```

```
SELECT CodR, NomeR
FROM RIVISTA
WHERE CodR IN
    (SELECT CodR
     FROM ARTICOLO
     WHERE Argomento='motociclismo');
```

- (b) Trovare il codice e il nome delle riviste che non hanno mai pubblicato articoli di argomento motociclismo.

```
SELECT CodR, NomeR
FROM RIVISTA
WHERE CodR NOT IN
    (SELECT CodR
     FROM ARTICOLO
     WHERE Argomento='motociclismo');
```

- (c) Trovare il codice e il nome delle riviste che hanno pubblicato solo articoli di motociclismo.

```
SELECT R.CodR, NomeR
FROM RIVISTA R, ARTICOLO A
WHERE R.CodR = A.CodR
AND R.CodR NOT IN
    (SELECT CodR
     FROM ARTICOLO
     WHERE Argomento <> 'motociclismo');
```

- (d) Trovare il codice e il nome delle riviste che pubblicano articoli di motociclismo oppure di auto.

```
SELECT R.CodR, NomeR
FROM RIVISTA R,ARTICOLO A
WHERE R.CodR=A.CodR
AND (Argomento='motociclismo' OR Argomento='auto');
```

- (e) Trovare il codice e il nome delle riviste che pubblicano articoli sia di motociclismo sia di auto.

```

SELECT R.CodR, NomeR
FROM RIVISTA R,ARTICOLO A
WHERE R.CodR=A.CodR
AND Argomento='motociclismo'
AND R.CodR IN
    (SELECT CodR
     FROM ARTICOLO
     WHERE Argomento='auto');

```

```

SELECT CodR, NomeR
FROM RIVISTA
WHERE CodR IN
    (SELECT CodR
     FROM ARTICOLO
     WHERE Argomento='motociclismo')
AND CodR IN
    (SELECT CodR
     FROM ARTICOLO
     WHERE Argomento='auto');

```

- (f) Trovare il codice e il nome delle riviste che hanno pubblicato almeno 2 articoli di motociclismo.

```

SELECT R.CodR, NomeR
FROM RIVISTA R,ARTICOLO A
WHERE R.CodR=A.CodR
AND Argomento='motociclismo'
GROUP BY R.CodR, NomeR
HAVING COUNT(*) >= 2;

```

- (g) Trovare il codice e il nome delle riviste che hanno pubblicato un solo articolo di motociclismo. (Possono aver scritto quanti articoli desiderano relativamente ad altri argomenti).

```

SELECT R.CodR, NomeR
FROM RIVISTA R,ARTICOLO A
WHERE R.CodR=A.CodR
AND Argomento='motociclismo'
GROUP BY R.CodR, NomeR
HAVING COUNT(*) = 1;

```

Esercizio 2

Dato lo schema relazionale costituito dalle tabelle (le chiavi primarie sono sottolineate):

```

VELISTI(Vid, VNome, Esperienza, DataNascita)
PRENOTAZIONI(Vid, Bid, Data)
BARCHE(Bid, BNome, Colore)

```

esprimere in SQL le seguenti interrogazioni:

- (a) Trovare i codici e i nomi dei velisti che hanno prenotato una barca rossa oppure una barca verde.

```

SELECT V.Vid, VNome
FROM VELISTI V, PRENOTAZIONI P, BARCHE B
WHERE V.Vid = P.Vid AND B.Bid = P.Bid

```

AND (Colore='Rosso' OR Colore='Verde');

```
SELECT Vid, VNome
FROM VELISTI V
WHERE Vid IN
    (SELECT Vid
     FROM PRENOTAZIONI P, BARCHE B
     WHERE B.Bid = P.Bid
     AND (Colore='Rosso' OR Colore='Verde');
```

- (b) Trovare i codici e i nomi dei velisti che hanno prenotato una barca rossa e una verde.

```
SELECT V.Vid, VNome
FROM VELISTI V, PRENOTAZIONI P, BARCHE B
WHERE V.Vid = P.Vid AND B.Bid = P.Bid
AND Colore='Rosso'
AND V.Vid IN
    (SELECT Vid
     FROM PRENOTAZIONI P, BARCHE B
     WHERE B.Bid = P.Bid
     AND Colore='Verde');
```

```
SELECT Vid, VNome
FROM VELISTI V
WHERE Vid IN
    (SELECT Vid
     FROM PRENOTAZIONI P, BARCHE B
     WHERE B.Bid = P.Bid
     AND Colore='Rosso')
AND Vid IN
    (SELECT Vid
     FROM PRENOTAZIONI P, BARCHE B
     WHERE B.Bid = P.Bid
     AND Colore='Verde');
```

- (c) Trovare i codici dei velisti che non hanno mai prenotato una barca rossa.

```
SELECT Vid
FROM VELISTI V
WHERE Vid NOT IN
    (SELECT Vid
     FROM PRENOTAZIONI P, BARCHE B
     WHERE B.Bid = P.Bid
     AND Colore='Rosso');
```

- (d) Trovare i codici e i nomi dei velisti che non hanno mai prenotato una barca rossa.

```
SELECT Vid, VNome
FROM VELISTI V
WHERE Vid NOT IN
    (SELECT Vid
     FROM PRENOTAZIONI P, BARCHE B
     WHERE B.Bid = P.Bid
     AND Colore='Rosso');
```

- (e) Trovare i codici e i nomi dei velisti che hanno prenotato almeno due barche diverse.

```
SELECT V.Vid, VNome
FROM VELISTI V, PRENOTAZIONI P
WHERE V.Vid = P.Vid
GROUP BY V.Vid, VNome
HAVING COUNT(DISTINCT Bid) >= 2;
```

```
SELECT Vid, VNome
FROM VELISTI V
WHERE Vid IN (SELECT Vid
              FROM PRENOTAZIONI
              GROUP BY Vid
              HAVING COUNT(DISTINCT Bid) >= 2);
```

- (f) Trovare i codici e i nomi dei velisti che hanno prenotato almeno tre barche diverse.

```
SELECT V.Vid, VNome
FROM VELISTI V, PRENOTAZIONI P
WHERE V.Vid = P.Vid
GROUP BY V.Vid, VNome
HAVING COUNT(DISTINCT Bid) >= 3;
```

```
SELECT Vid, VNome
FROM VELISTI V
WHERE Vid IN (SELECT Vid
              FROM PRENOTAZIONI
              GROUP BY Vid
              HAVING COUNT(DISTINCT Bid) >= 3);
```

- (g) Trovare i codici e i nomi dei velisti che hanno effettuato almeno tre prenotazioni.

```
SELECT V.Vid, VNome
FROM VELISTI V, PRENOTAZIONI P
WHERE V.Vid = P.Vid
GROUP BY V.Vid, VNome
HAVING COUNT(*) >= 3;
```

```
SELECT Vid, VNome
FROM VELISTI V
WHERE Vid IN (SELECT Vid
              FROM PRENOTAZIONI
              GROUP BY Vid
              HAVING COUNT(*) >= 3);
```

Esercizio 3

Dato lo schema relazionale costituito dalle tabelle (le chiavi primarie sono sottolineate):

AEREO(Aid, ANome, Autonomia)
CERTIFICATO(Did, Aid)
DIPENDENTE(Did, DNome, Stipendio)

esprimere in SQL le seguenti interrogazioni:

- (a) Trovare i codici e i nomi dei dipendenti abilitati al volo su un aereo in grado di coprire distanze superiori a 5000 Km (Autonomia \geq 5000).

```
SELECT D.Did, DNome
FROM DIPENDENTE D, CERTIFICATO C, AEREO A
WHERE D.Did = C.Did AND C.Aid = A.Aid
AND Autonomia > 5000;
```

```
SELECT Did, DNome
FROM DIPENDENTE
WHERE Did IN
    (SELECT Did
     FROM CERTIFICATO C, AEREO A
     WHERE C.Aid = A.Aid
     AND Autonomia > 5000);
```

- (b) Trovare i codici e i nomi dei dipendenti abilitati al volo su almeno due aerei in grado di coprire distanze superiori a 5000 Km (Autonomia \geq 5000) e che siano abilitati al volo su qualche Boeing.

```
SELECT D.Did, DNome
FROM DIPENDENTE D, CERTIFICATO C, AEREO A
WHERE D.Did = C.Did AND C.Aid = A.Aid
AND Autonomia > 5000
AND Did IN
    (SELECT Did
     FROM CERTIFICATO C, AEREO A
     WHERE C.Aid = A.Aid
     AND ANome = 'Boeing')
GROUP BY D.Did, DNome
HAVING COUNT(*) >= 2;
```

```
SELECT Did, DNome
FROM DIPENDENTE
WHERE Did IN
    (SELECT Did
     FROM CERTIFICATO C, AEREO A
     WHERE C.Aid = A.Aid
     AND Autonomia > 5000
     GROUP BY Did
     HAVING COUNT(*) >= 2)
AND Did IN
    (SELECT Did
     FROM CERTIFICATO C, AEREO A
     WHERE C.Aid = A.Aid
     AND ANome = 'Boeing');
```

Esercizio 4

Dato lo schema relazionale costituito dalle tabelle (le chiavi primarie sono sottolineate):

CORSO (CodCorso, NomeC, Anno, Semestre)
ORARIO-LEZIONI (CodCorso, GiornoSettimana, OraInizio, OraFine, Aula)

esprimere in SQL le seguenti interrogazioni:

- (a) Trovare le aule in cui non si tengono mai lezioni di corsi del primo anno.

```
SELECT DISTINCT Aula
FROM ORARIO-LEZIONI
WHERE Aula NOT IN
  (SELECT Aula
   FROM ORARIO_LEZIONI OL, CORSO C
   WHERE OL.CodCorso=C.CodCorso
   AND C.Anno=1);
```

- (b) Trovare codice corso, nome corso e numero totale di ore di lezione settimanali per i corsi del terzo anno per cui il numero complessivo di ore di lezione settimanali è superiore a 10 e le lezioni sono in più di tre giorni diversi della settimana.

```
SELECT C.CodCorso, C.NomeC, SUM(OraFine-OraInizio)
FROM CORSO C, ORARIO-LEZIONI OL
WHERE C.CodCorso=OL.CodCorso
AND C.Anno = 3
GROUP BY C.Corso, C.NomeC
HAVING SUM(OraFine-OraInizio)>10 AND COUNT(DISTINCT GiornoSettimana)>3;
```

Esercizio 5

Dato lo schema relazionale costituito dalle tabelle (le chiavi primarie sono sottolineate):

ALLOGGIO (CodA, Indirizzo, Città, Superficie, CostoAffittoMensile)
CONTRATTO-AFFITTO (CodC, DataInizio, DataFine*, NomePersona, CodA)

esprimere in SQL le seguenti interrogazioni:

- (a) Trovare, per le città in cui sono stati stipulati almeno 100 contratti, la città, il costo mensile massimo degli affitti, il costo mensile medio degli affitti, la durata massima dei contratti, la durata media dei contratti e il numero totale di contratti stipulati.

```
SELECT Città, MAX(CostoAffittoMensile), AVG(CostoAffittoMensile),
MAX(DataFine-DataInizio), AVG(DataFine-DataInizio), COUNT(*)
FROM ALLOGGIO A, CONTRATTO-AFFITTO C
WHERE A.CodA=C.CodA
GROUP BY Città
HAVING COUNT(*)>=100;
```

- (b) Trovare il nome delle persone che non hanno mai affittato alloggi con superficie superiore a 80 metri quadri.

```

SELECT DISTINCT NomePersona
FROM CONTRATTO-AFFITTO
WHERE NomePersona NOT IN
    (SELECT NomePErsona
     FROM CONTRATTO-AFFITTO C, ALLOGGIO A
     WHERE C.CodA=A.CodA
     AND Superficie>80);

```

N.B. Superficie espressa in metri quadri. Per i contratti in corso, DataFine è NULL.

Esercizio 5

Dato lo schema relazionale costituito dalle tabelle (le chiavi primarie sono sottolineate):

```

ORCHESTRA(CodO, NomeO, NomrDirettore, numElementi)
CONCERTI(CodC, Data, CodO, CodS, PrezzoBiglietto)
SALE(CodS, NomeS, Citt`a, Capienza)

```

Esprimere in SQL le seguenti interrogazioni:

- (a) Trovare il codice e il nome delle orchestre con più di 30 elementi che hanno tenuto concerti sia a Torino, sia a Milano e non hanno mai tenuto concerti a Bologna.

```

SELECT O.CodO, O.NomeO
FROM ORCHESTRA O
WHERE O.NumElementi>30
AND O.CodO IN
    (SELECT C1.CodO FROM CONCERTI C1, SALE S1
     WHERE C1.CodS=S1.CodS
     AND S1.Citta='Torino')
AND O.CodO IN
    (SELECT C2.CodO FROM CONCERTI C2, SALE S2
     WHERE C2.CodS=S2.CodS
     AND S2.Citta='Milano')
AND O.CodO NOT IN
    (SELECT C3.CodO FROM CONCERTI C3, SALE S3
     WHERE C3.CodS=S3.CodS
     AND S3.Citta='Bologna');

```

Esercizio 6

Dato lo schema relazionale costituito dalle tabelle (le chiavi primarie sono sottolineate):

```

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

```

Esprimere in SQL le seguenti interrogazioni:

- (a) Visualizzare codice, nome e media dei redditi dichiarati dal 2010 in poi per i contribuenti tali che il massimo reddito da loro dichiarato dal 2010 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);
```