

Esercizi di SQL

È riportato di seguito un insieme di esercizi risolti in SQL. Per ogni esercizio una o più soluzioni equivalenti sono presentate.

Esercizio 1.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
PERSONE(CodFisc, Nome, Cognome, Indirizzo, Città)
MULTE(IdMulta, CodFisc, DataMult
```

Interrogazione

Selezionare il nome e il cognome delle persone per cui il numero di multe ricevute nel 2005 è superiore al numero di multe ricevute nel 2004 dalla stessa persona.

Soluzione

```
SELECT Nome, Cognome
FROM PERSONE P, MULTE M
WHERE P.CodFisc=M.CodFisc
      AND DataMult>='1/1/2005'
      AND DataMult<='31/12/2005'
GROUP BY P.CodFisc, Nome, Cognome
HAVING COUNT(*)>(SELECT COUNT(*) FROM MULTE M2
                  WHERE M2.CodFisc=P.CodFisc
                     AND DataMult>='1/1/2004'
                     AND DataMult<='31/12/2004');
```

Esercizio 2.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
SALA_RIUNIONE(CodiceSala, CodiceSede, NomeSala)
SEDE(CodiceSede, Nome, Indirizzo, Città, Stato)
```

Operazione.

Eliminare dalla base di dati tutte le sedi tedesche e le relative sale riunione.

Soluzione

```
DELETE FROM SALA_RIUNIONE
WHERE CodiceSede IN (SELECT CodiceSede
                    FROM SEDE
                    WHERE Stato='Germania');

DELETE FROM SEDE
WHERE Stato='Germania';
```

Esercizio 3.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

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

Interrogazione

Selezionare il codice e il nome delle riviste che hanno pubblicato almeno 1 articolo.

Soluzione

```
SELECT DISTINCT RIVISTA.CodR, NomeR
FROM RIVISTA, ARTICOLO
WHERE RIVISTA.CodR=ARTICOLO.CodR;
```

Esercizio 4.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
DIPENDENTE(Matr, NomeD)
LINGUE-CONOSCIUTE(Matr, Lingua)
PROGETTO(CodP, NomeP)
LAVORA-IN(Matr, CodP, DataInizio, DataFine, Mansione)
```

Interrogazione

Per i dipendenti che hanno lavorato complessivamente per più di 6 mesi nello stesso progetto, selezionare matricola, nome e numero totale di progetti diversi in cui hanno lavorato.

Soluzione

```
SELECT D.Matr, NomeD, COUNT(DISTINCT CodP)
FROM DIPENDENTE D, LAVORA-IN LI
WHERE LI.Matr=D.Matr
AND D.Matr IN
  (SELECT Matr FROM LAVORA-IN
   GROUP BY Matr, CodP
   HAVING SUM(MONTHS(DataFine)-MONTHS(DataInizio))>6)
GROUP BY D.Matr, NomeD;
```

MONTHS rappresenta una semplificazione dell'applicazione delle funzioni che permettono di estrarre il mese di una data.

Esercizio 5.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

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

Interrogazione

Selezionare il nome degli editori per cui almeno 10 pubblicazioni sono state vendute nel 2002 nelle librerie di Roma in più di 2.000 copie.

Soluzione

```
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 6.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

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

Interrogazione

Selezionare il codice delle riviste che hanno pubblicato almeno 1 articolo di motociclismo (Argomento='motociclismo').

Soluzione

```
SELECT DISTINCT CodR FROM ARTICOLO
WHERE Argomento='motociclismo';
```

Esercizio 7.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

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

Interrogazione

Selezionare il numero di articoli pubblicati sulla rivista D100 (CodR='D100').

Soluzione

```
SELECT COUNT(*) FROM ARTICOLO
WHERE CodR='D100';
```

Esercizio 8.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
DIPENDENTE(Matricola, Nome, Cognome, Filiale)
LINGUE_CONOSCIUTE(Matricola, Lingua)
```

Interrogazione.

Selezionare matricola, nome e cognome dei dipendenti che conoscono sia il francese sia l'inglese.

Soluzione

```
SELECT Matricola, Nome, Cognome
FROM DIPENDENTE D, LINGUE_CONOSCIUTE L
WHERE D.Matricola=L.Matricola
      AND Lingua='Francese'
      AND Matricola IN (SELECT Matricola
                        FROM LINGUE_CONOSCIUTE
                        WHERE Lingua='Inglese');
```

Esercizio 9.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
FORNITORE(PartitaIVA, RagioneSociale, Stato)
CLIENTE(CodiceFiscale, Nome, Cognome, Stato)
```

Interrogazione.

Selezionare gli stati associati ad almeno un cliente oppure ad almeno un fornitore.

Soluzione

```
SELECT Stato
FROM FORNITORE
UNION
SELECT Stato
FROM CLIENTE;
```

Esercizio 10.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
GARA(CodG, Luogo, Data, Disciplina)
ATLETA(CodA, Nome, Nazione, DataNascita)
PARTECIPAZIONE(CodG, CodA, PosizioneArrivo, Tempo)
```

Interrogazione

Selezionare il codice e il nome degli atleti italiani che sono giunti primi in almeno una gara di lancio del peso.

Soluzione

```
SELECT DISTINCT A.CodA, Nome FROM ATLETA A, PARTECIPAZIONE P, GARA G
WHERE P.CodA=A.CodA
      AND P.CodG=G.CodG
      AND Disciplina='lancio del peso'
      AND Nazione='Italia'
      AND PosizioneArrivo=1;
```

Esercizio 11.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
DOCENTE(CodDocente, Nome, Cognome, DataNascita)
CORSI(CodCorso, NomeCorso, Sede, CodDocente)
LEZIONI_SVOLTE(CodCorso, Data, OraInizio, Aula, Durata)
```

Operazione di gestione da eseguire.

Scrivere, nell'ordine appropriato, i comandi SQL necessari per eliminare le tabelle CORSI e LEZIONI_SVOLTE.

Soluzione

```
DROP TABLE LEZIONI_SVOLTE;
```

```
DROP TABLE CORSI;
```

Esercizio 12.

È dato lo schema relazionale costituito dalla seguente tabella (le chiavi primarie sono sottolineate)

```
DOCENTE(CodDocente, Nome, Cognome, AnnoNascita)
```

Operazione di gestione da eseguire.

Scrivere il comando SQL necessario per aggiungere un vincolo sulla colonna AnnoNascita della tabella DOCENTE tale da imporre che AnnoNascita assuma valori successivi al 1899.

Soluzione

```
ALTER TABLE DOCENTE
ADD CONSTRAINT VincoloAnnoDiNascita CHECK(AnnoNascita>1899);
```

Esercizio 13.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
PROFESSORI(Matricola, Nome, Cognome, Dipartimento)
PRESIDI(Matricola, Facoltà, DataNomina)
VICE_PRESIDI(Matricola, Facoltà, DataNomina)
```

Interrogazione.

Selezionare la matricola dei professori che non sono presidi.

Soluzione

```
SELECT Matricola
FROM PROFESSORI
WHERE Matricola NOT IN (SELECT Matricola
                        FROM PRESIDI);
```

Oppure

```
SELECT Matricola
FROM PROFESSORI
EXCEPT
SELECT Matricola
FROM PRESIDI;
```

Esercizio 14.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
SALA_RIUNIONE(CodiceSala, CodiceSede, NomeSala, Dimensione, NumeroPosti)
SEDE(CodiceSede, Nome, Indirizzo, Città, Stato)
```

Operazione.

Ridurre a 50 il numero massimo di posti per le sale più piccole di 100mq che hanno attualmente un numero di posti superiore a 50.

Soluzione

```
UPDATE SALA_RIUNIONE SET NumeroPosti=50
WHERE Dimensione<100
AND NumeroPosti>50;
```

Esercizio 15.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

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

Interrogazione

Selezionare il codice, il nome e il numero di articoli pubblicati su ogni rivista.

Soluzione

```
SELECT RIVISTA.CodR, NomeR, COUNT(*)
FROM RIVISTA, ARTICOLO
WHERE RIVISTA.CodR=ARTICOLO.CodR
GROUP BY RIVISTA.CodR, NomeR;
```

Esercizio 16.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

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

Interrogazione

Selezionare il codice delle riviste che hanno pubblicato almeno due articoli di motociclismo.

Soluzione

```
SELECT CodR FROM ARTICOLO
WHERE Argomento='motociclismo'
GROUP BY CodR
HAVING COUNT(*)>1;
```

Esercizio 17.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
CORSO (CodCorso, NomeC, Anno, Semestre)
ORARIO_LEZIONI (CodCorso, GiornoSettimana, OraInizio, OraFine, Aula)
```

Interrogazione

Selezionare 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.

Soluzione

```
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 18.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
ALLOGGIO (CodA, Indirizzo, Città, Superficie, CostoAffittoMensile)
CONTRATTO-AFFITTO (CodC, DataInizio, DataFine, NomePersona, CodA)
N.B. Superficie espressa in metri quadri. Per i contratti in corso, DataFine è NULL.
```

Interrogazione

Selezionare, 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.

Soluzione

```
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;
```

Esercizio 19.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
SALA_RIUNIONE(CodiceSala, CodiceSede, NomeSala, NumeroPosti)
SEDE(CodiceSede, Nome, Indirizzo, Città, Stato)
```

Operazione.

Impostare a 100 il numero di posti della sala numero 1 (CodiceSala=1) collocata presso la sede 15 (CodiceSede=15).

Soluzione

```
UPDATE SALA_RIUNIONE SET NumeroPosti=100
WHERE CodiceSala=1 AND CodiceSede=15;
```

Esercizio 20.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

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

Interrogazione

Selezionare gli editori che non hanno mai pubblicato articoli di motociclismo.

Soluzione

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

Esercizio 21.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
GRANPREMIO(NumGP, Anno, Stato)
PILOTA(CodP, Cognome, Nome, Nazione)
PARTECIPAZIONE(NumGP, Anno, CodP)
```

Operazione di gestione da eseguire.

Scrivere il comando SQL necessario per aggiungere un vincolo tale da imporre che l'attributo NumGP della tabella GRANPREMIO possa assumere esclusivamente valori maggiori o uguali a 1.

Soluzione

```
ALTER TABLE GRANPREMIO
ADD CONSTRAINT VincoloNumGP CHECK(NumGP>=1);
```


Esercizio 22.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

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

Interrogazione

Selezionare 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.

Soluzione

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

Esercizio 23.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

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

Interrogazione

Selezionare il codice e il nome delle riviste che hanno pubblicato almeno 10 articoli di automobilismo e almeno 25 articoli di motociclismo.

Soluzione

```
SELECT CodR, NomeR FROM RIVISTA
WHERE CodR IN
  (SELECT CodR ARTICOLO
   WHERE Argomento='automobilismo'
   GROUP BY CodR
   HAVING COUNT(*)>=10)
  AND CodR IN
  (SELECT CodR ARTICOLO
   WHERE Argomento='motociclismo'
   GROUP BY CodR
   HAVING COUNT(*)>=25);
```

Esercizio 24.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
RIUNIONE(CodR, Descrizione, DataRiunione)
DIPENDENTE(CodD, Nome, Cognome, DataNascita, Città)
PARTECIPA_RIUNIONE(CodD, CodR)
```

Interrogazione

Visualizzare il codice dei dipendenti che hanno partecipato solamente alle riunioni alle quali ha partecipato il dipendente D100 (CodD='D100').

Soluzione

```
SELECT CodD
FROM PARTECIPA_RIUNIONE
WHERE CodR IN
    (SELECT CodR FROM PARTECIPA_RIUNIONE
     WHERE CodD='D100')
GROUP BY CodD
HAVING COUNT(*) = (SELECT COUNT(*) FROM PARTECIPA_RIUNIONE
                   WHERE CodD='D100');
```

Oppure

```
SELECT CodD
FROM DIPENDENTE D
WHERE CodD NOT IN
    (SELECT CodD FROM PARTECIPA_RIUNIONE
     WHERE CodR NOT IN
        (SELECT CodR FROM PARTECIPA_RIUNIONE
         WHERE CodD='D100'));
```

Oppure

```
SELECT CodD
FROM DIPENDENTE D
WHERE NOT EXISTS
    (SELECT * FROM PARTECIPA_RIUNIONE PR
     WHERE PR.CodD=D.CodD
      AND PR.CodR NOT IN
        (SELECT CodR FROM PARTECIPA_RIUNIONE
         WHERE CodD='D100'));
```

Esercizio 25.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
SALA_RIUNIONE(CodiceSala, CodiceSede, NomeSala, NumeroPosti)
SEDE(CodiceSede, Nome, Indirizzo, Città, Stato)
PRENOTAZIONE(CodiceSala, CodiceSede, Data, OraInizio, Durata)
```

Operazione.

Eliminare dalla base di dati le sale riunioni con meno di 10 posti e tutte le prenotazioni relative a tali sale.

Soluzione

```
DELETE FROM PRENOTAZIONE
WHERE (CodiceSala, CodiceSede) IN
      (SELECT CodiceSala, CodiceSede
       FROM SALA_RIUNIONE
       WHERE NumeroPosti<10);
```

```
DELETE FROM SALA_RIUNIONE
WHERE NumeroPosti<10);
```

Esercizio 26.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

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

Interrogazione

Selezionare il nome degli studenti di Torino che hanno conseguito il punteggio massimo possibile nei quiz di matematica.

Soluzione

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

Esercizio 27.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
RIUNIONE(CodR, Descrizione, DataRiunione)
DIPENDENTE(CodD, Nome, Cognome, DataNascita, Città)
PARTECIPA_RIUNIONE(CodD, CodR)
```

Interrogazione

Visualizzare il codice dei dipendenti che hanno partecipato a tutte le riunioni che si sono svolte a gennaio 2006.

Soluzione

```
SELECT CodD
FROM PARTECIPA_RIUNIONE PR, RIUNIONE R
WHERE PR.CodR=R.CodR AND DataRiunione>='01/01/2006'
AND DataRiunione<='31/01/2006'
GROUP BY CodD
HAVING COUNT(*)=(SELECT COUNT(*)
FROM RIUNIONE
WHERE DataRiunione>='01/01/2006'
AND DataRiunione<='31/01/2006');
```

Esercizio 28.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
GRANPREMIO(NumGP, Anno, Stato)
PILOTA(CodP, Cognome, Nome, Nazione)
PARTECIPAZIONE(NumGP, Anno, CodP)
```

Operazione di gestione da eseguire.

Scrivere i comandi SQL necessari per creare tutte le tabelle riportate nello schema relazionale. Selezionare per ogni attributo il tipo di dato ritenuto più opportuno e indicare i vincoli di integrità ritenuti necessari.

Soluzione

```
CREATE TABLE GRANPREMIO
( NumGP  SMALLINT,
  Anno   SMALLINT,
  STATO  VARCHAR(15) NOT NULL,
  PRIMARY KEY(NumGP, Anno)
);

CREATE TABLE PILOTA
( CodP    SMALLINT,
  Cognome  VARCHAR(20) NOT NULL,
  Nome     VARCHAR(20) NOT NULL,
  Nazione  VARCHAR(15) NOT NULL,
  PRIMARY KEY(CodP)
);

CREATE TABLE PARTECIPAZIONE
```

```
( NumGP  SMALLINT,  
  Anno   SMALLINT,  
  CodP   SMALLINT,  
  PRIMARY KEY(NumGP, Anno, CodP),  
  FOREIGN KEY (NumGP, Anno) REFERENCES GRANPREMIO(NumGP, Anno)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  FOREIGN KEY CodP REFERENCES PILOTA(CodP)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);
```

Esercizio 29.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

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

Interrogazione

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 nella base di dati.

Soluzione

```
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>'1/1/1990'  
GROUP BY C.CodFiscale, C.Nome  
HAVING MAX(D.Reddito)>(SELECT AVG(Reddito) FROM DICHIARAZIONE);
```

Esercizio 30.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

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

Interrogazione

Selezionare il codice e il nome di tutte le riviste presenti nella base di dati.

Soluzione

```
SELECT CodR, NomeR FROM RIVISTA;
```

Esercizio 31.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
SALA_RIUNIONE(CodiceSala, CodiceSede, NomeSala)
SEDE(CodiceSede, Nome, Indirizzo)
```

Operazione.

Inserire nella base di dati la sala 5 con nome: 'Sala conferenze', CodiceSede: 1.

Soluzione

```
INSERT INTO SALA_RIUNIONI(CodiceSala, NomeSala, CodiceSede)
VALUES(5, 'Sala conferenze', 1);
```

Esercizio 32.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
GRANPREMIO(NumGP, Anno, Stato, Città)
PILOTA(CodP, Nome, Nazionalità)
PARTECIPA(NumGP, Anno, CodP)
```

Interrogazione

Selezionare gli anni in cui si sono tenuti gran premi in almeno 15 stati diversi e meno di 2 gran premi in Italia.

Soluzione

```
SELECT Anno FROM GRANPREMIO
WHERE Anno NOT IN
  (SELECT Anno FROM GRANPREMIO
   WHERE Stato='Italia'
   GROUP BY Anno
   HAVING COUNT(*)>=2)
GROUP BY Anno
HAVING COUNT(DISTINCT Stato)>=15;
```

Esercizio 33.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
PERSONA(CodF, Cognome, DataNascita)
CAMPO(CodCampo, Coperto)
PRENOTAZIONI(CodCampo, Data, OraInizio, OraFine, CodFisc)
```

L'attributo Coperto della relazione CAMPO può assumere valore 'si' e 'no'.

Interrogazione

Selezionare il codice del campo e il numero di prenotazioni fatte nell'anno 1981 per tutti i campi che sono stati prenotati da almeno 50 persone diverse nel corso dell'anno 1980.

Soluzione

```
SELECT CodCampo, COUNT(*) FROM PRENOTAZIONI
WHERE Data>='1/1/1981'
      AND Data<='31/12/1981'
      AND CodCampo IN
      (SELECT CodCampo FROM PRENOTAZIONI
       WHERE Data>='1/1/1980'
        AND Data<='31/12/1980'
       GROUP BY CodCampo
       HAVING COUNT(DISTINCT CodFisc)>=50)
GROUP BY CodCampo;
```

Esercizio 34.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
DIPENDENTE(Matricola, Nome, Cognome, Filiale)
LINGUE_CONOSCIUTE(Matricola, Lingua)
```

Interrogazione.

Selezionare matricola, nome e cognome dei dipendenti che non conoscono l'inglese.

Soluzione

```
SELECT Matricola, Nome, Cognome
FROM DIPENDENTE
WHERE Matricola NOT IN (SELECT Matricola
                        FROM LINGUE_CONOSCIUTE
                        WHERE Lingua='Inglese');
```

Esercizio 35.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
ALBERGO(CodA, Nome, Categoria)
SERVIZIO(CodS, NomeServizio)
SERVIZI_OFFERTI(CodA, CodS)
```

Operazione di gestione da eseguire.

Scrivere i comandi SQL necessari per creare tutte le tabelle riportate nello schema relazionale. Selezionare per ogni attributo il tipo di dato ritenuto più opportuno e indicare i vincoli di integrità ritenuti necessari.

Soluzione

```
CREATE TABLE ALBERGO
( CodA      SMALLINT,
  Nome      VARCHAR(20) NOT NULL,
  Categoria SMALLINT,
  PRIMARY KEY(CodA)
);

CREATE TABLE SERVIZIO
( CodS      SMALLINT,
  NomeServizio VARCHAR(20) NOT NULL,
  PRIMARY KEY(CodS)
);

CREATE TABLE SERVIZI_OFFERTI
( CodA      SMALLINT,
  CodS      SMALLINT,
  PRIMARY KEY(CodA, CodS),
  FOREIGN KEY CodA REFERENCES ALBERGO(CodA)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY CodS REFERENCES SERVIZIO(CodS)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```


Esercizio 36.

È dato lo schema relazionale costituito dalle seguenti tabelle (le chiavi primarie sono sottolineate)

```
MEDICO(Matr, Nome)
MEDICINALE(CodM, NomeM)
PAZIENTE(CodP, NomeP, DataNascita)
PRESCRIZIONE(Matr, CodM, CodP, Data)
```

Interrogazione

Selezionare codice e nome dei pazienti a cui non sono mai stati prescritti medicinali oppure sono stati prescritti medicinali solo dopo il compimento di 40 anni.

Soluzione

```
SELECT CodP, NomeP FROM PAZIENTE
WHERE CodP NOT IN
  (SELECT P.CodP FROM PAZIENTE P, PRESCRIZIONE PR
   WHERE PR.CodP=P.CodP
    AND YEARS(PR.Data)-YEARS(P.DataNascita)<40);
```

YEARS rappresenta una semplificazione dell'applicazione delle funzioni che permettono di estrarre l'anno di una data.