

## TABELLA PERSONE

```
CREATE TABLE IF NOT EXISTS Persone (  
  Id char(2) PRIMARY KEY,  
  Nome varchar(10),  
  Reddito int(11),  
  Eta tinyint(4),  
  Sesso enum('M','F')  
) ENGINE=InnoDB;
```

## insert into PERSONE

```
INSERT INTO `Persone` (`Id`, `Nome`, `Reddito`, `Eta`,  
`Sesso`) VALUES  
( 'A1', 'Aldo', 25, 15, 'M'), ('A2', 'Andrea', 27, 21, 'M'), ('A3',  
'Antonino', 44, 40, 'M'), ('A4', 'Amelia', 79, 28, 'F'), ('A5', 'Anna',  
50, 29, 'F'), ('A6', 'AnnaMaria', 41, 30, 'F'), ('A7', 'Aldo', 25, 20,  
'M'), ('B1', 'Beatrice', 79, 30, 'F'), ('F1', 'Filippo', 26, 30, 'M'), ('F2',  
'Franco', 60, 20, 'M'), ('L1', 'Leonardo', 79, 30, 'M'), ('L2', 'Luigi',  
50, 40, 'M'), ('L3', 'Luisa', 75, 87, 'F'), ('M1', 'Marco', 11, 10, 'M'),  
( 'M2', 'Michele', 79, 30, 'M'), ('M3', 'Maria', 55, 42, 'F'), ('O1',  
'Olga', 30, 41, 'F'), ('S1', 'Sergio', 85, 35, 'M');
```

## TABELLA GENITORE

```
CREATE TABLE IF NOT EXISTS `Genitori` (  
  Figlio char(2),  
  Genitore char(2),  
  FOREIGN KEY (Genitore) REFERENCES Persone (Id)  
    ON DELETE CASCADE,  
  FOREIGN KEY (Figlio) REFERENCES Persone (Id)  
    ON DELETE CASCADE  
) ENGINE=InnoDB;
```

## insert into GENITORE

```
INSERT INTO `Genitori` (`Figlio`, `Genitore`) VALUES  
('A7', 'A3'), ('M1', 'A3'), ('A6', 'A4'), ('F1', 'A5'), ('O1', 'A5'), ('A7',  
'A6'), ('M1', 'A6'), ('A3', 'B1'), ('A1', 'F2'), ('A2', 'F2'), ('A3', 'L1'),  
('F1', 'L2'), ('O1', 'L2'), ('L2', 'L3'), ('M3', 'L3'), ('A6', 'M2'), ('A1',  
'M3'), ('A2', 'M3'), ('F2', 'S1');
```

## 1. Trovare id, nome e reddito delle donne con meno di 35 anni

```
SELECT p.Id, p.Nome, p.Reddito  
FROM Persone p  
WHERE (p.Eta < 35) AND (p.Sesso='F');
```

## 2. Trovare id, nome e reddito dei genitori di Antonino (Id='A3')

```
SELECT p.Id, p.Nome, p.Reddito  
FROM Persone p, Genitori g  
WHERE g.Figlio='A3' AND p.Id = g.Genitore
```

### 3. Restituire per ogni persona, il nome della persona e quello dei nonni

```
SELECT non.Nome AS Nonno, nip.Nome AS Nipote
FROM   Persone non, Genitori g1, Genitori g2, Persone nip
WHERE  non.Id=g1.Genitore AND g1.Figlio=g2.Genitore AND g2.
Figlio=nip.Id
ORDER BY non.Nome;
```

### 4. Trovare il reddito minimo, medio e massimo di donne e uomini

```
SELECT  p.Sesso, min(p.Reddito) AS Minimo,
        avg(p.Reddito) as Medio,
        max(p.Reddito) as Massimo
FROM    Persone p
GROUP BY p.Sesso;
```

## 5. Trovare il reddito medio dei padri per età

```
SELECT p.Eta, avg(p.Reddito) AS RedditoMedio
FROM   Persone p
WHERE  p.Sesso='M' AND EXISTS
      (SELECT *
       FROM   Genitori g
       WHERE  g.Genitore=p.Id)
GROUP BY p.Eta;
```

## 6. Fornire le coppie di fratelli (due persone sono fratelli se hanno un genitore in comune), indicandone i nomi

```
SELECT DISTINCT p1.Nome, p2.Nome
FROM   Persone p1, Genitori g1, Genitori g2, Persone p2
WHERE  g1.Figlio=p1.Id AND g2.Figlio=p2.Id AND g1.Genitore=g2.Genitore AND p1.
Id<p2.Id
ORDER BY p1.Nome;
```

## 7. Nomi di tutti i fratelli delle persone di nome Aldo

```
SELECT DISTINCT p.Id, p.Nome  
FROM   Persone a JOIN Genitori g1 ON (a.Id = g1.Figlio),  
       Persone p JOIN Genitori g2 ON (p.Id = g2.Figlio)  
WHERE  (g1.Genitore = g2.Genitore) AND (a.Nome='Aldo' ) AND (a.Id <> p.Id);
```

## 8. Nomi delle persone che sono genitori di almeno 2 figli

```
SELECT p.Nome  
FROM   Persone p JOIN Genitori g ON (p.Id=g.Genitore)  
GROUP BY p.Id, p.Nome  
HAVING COUNT(*) >1;
```



## 9. Mostrare per ciascun figlio i rispettivi genitori (Padre, Madre, Figlio), indicando, per ciascuno, il nome

```
SELECT m.Nome AS Madre, p.Nome AS Padre, f.Nome AS Figlio
FROM   Persone f, Genitori gm, Persone m, Genitori gp, Persone p
WHERE  f.Id=gm.Figlio AND m.Id=gm.Genitore AND m.Sesso='F' AND f.Id=gp.Figlio
AND p.Id=gp.Genitore AND p.Sesso='M';
```

## 10. Trovare l'elenco dei genitori i cui figli guadagnano tutti più di 20000 euro. Mostrare identificatore e nome, ordinando la lista per nome

```
SELECT DISTINCT g.Id, g.Nome
FROM Persone g JOIN Genitori gg1 ON (g.Id=gg1.Genitore)
WHERE NOT EXISTS (SELECT f.Id
                  FROM Persone f, Genitori gg
                  WHERE f.Id = gg.Figlio AND gg.Genitore=g.Id AND f.Reddito <= 20)
ORDER BY g.Nome;
```

**11. Definire una vista EtaMediaFigli che restituisce per ogni persona (id, nome) l'età media dei suoi figli. Visualizzare il contenuto della vista**

```
DROP VIEW IF EXISTS EtaMediaFigli;
```

```
CREATE VIEW EtaMediaFigli AS  
  SELECT g.Id, g.Nome, AVG(f.Eta) AS Media  
  FROM   Persone g, Genitori gg, Persone f  
  WHERE  g.Id=gg.Genitore AND gg.Figlio=f.Id  
  GROUP BY g.Id, g.nome;
```

```
SELECT * FROM EtaMediaFigli;
```

**12. Utilizzare la vista EtaMediaFigli per trovare tra tutti i genitori quelli i cui figli hanno la massima eta media**

```
SELECT Nome  
FROM EtaMediaFigli  
WHERE Media >=ALL (SELECT Media FROM EtaMediaFigli);
```

**13. Cancellare le Persone con reddito maggiore di 20000 euro. Mostrare le tabelle Persone e Genitori risultanti.**

```
DELETE FROM Persone  
WHERE Reddito>50;
```

```
SELECT * FROM Persone;
```

```
SELECT * FROM Genitori;
```

**14. Aumentare del 20 % tutti i redditi inferiori a 30000 euro.  
Mostrare la tabella Persone risultante.**

```
UPDATE Persone  
SET   Reddito=Reddito*1.2  
WHERE Reddito<30;  
  
SELECT * FROM Persone;
```