

Ripuliamo, eliminando le tabelle qualora esistessero già`

SET FOREIGN_KEY_CHECKS=0; (potrebbe essere utile per eliminare il controllo sulle foreign key)

DROP TABLE IF EXISTS File;

DROP TABLE IF EXISTS Directory;

DROP TABLE IF EXISTS GruppiUtenti;

DROP TABLE IF EXISTS Utenti;

DROP TABLE IF EXISTS Gruppi;

```
CREATE TABLE Utenti (  
    Login      CHAR(8) PRIMARY KEY,  
    NomeCognome VARCHAR(20)  
) ENGINE=InnoDB;
```

```
CREATE TABLE Gruppi (  
    Nome        CHAR(8) PRIMARY KEY,  
    DataCreaz   DATE  
) ENGINE=InnoDB;
```

```
CREATE TABLE GruppiUtenti (  
    Gruppo      CHAR(8),  
    Utente       CHAR(8),  
    PRIMARY KEY (Utente,Gruppo),  
    FOREIGN KEY (Utente) REFERENCES Utenti(Login)  
        ON DELETE CASCADE,  
    FOREIGN KEY (Gruppo) REFERENCES Gruppi(Nome)  
) ENGINE=InnoDB;
```

```
CREATE TABLE File (  
    Id      INT      PRIMARY KEY,  
    Nome    CHAR(20),  
    Utente  CHAR(8),  
    Gruppo  CHAR(8),  
    Padre   INT,  
    FOREIGN KEY (Utente) REFERENCES Utenti(Login)  
                ON DELETE CASCADE,  
    FOREIGN KEY (Gruppo) REFERENCES Gruppi(Nome),  
    FOREIGN KEY (Padre) REFERENCES Directory(Id)  
                ON DELETE CASCADE  
) ENGINE=InnoDB;
```

NOTA:

SET FOREIGN_KEY_CHECKS=0;

SET FOREIGN_KEY_CHECKS=1;

```
CREATE TABLE Directory (  
    Id    INT        PRIMARY KEY,  
    FOREIGN KEY (Id) REFERENCES File(Id)  
        ON DELETE CASCADE  
) ENGINE=InnoDB;
```

```
-- LOAD DATA LOCAL INFILE 'Utenti.txt' INTO TABLE Utenti;  
-- LOAD DATA LOCAL INFILE 'File.txt' INTO TABLE File;  
-- LOAD DATA LOCAL INFILE 'Gruppi.txt' INTO TABLE Gruppi;  
-- LOAD DATA LOCAL INFILE 'GruppiUtenti.txt' INTO TABLE GruppiUtenti;  
-- LOAD DATA LOCAL INFILE 'Directory.txt' INTO TABLE Directory;  
  
SET FOREIGN_KEY_CHECKS=1;
```

1. Trovare il nome e Id della directory radice

```
SELECT Nome, Id
```

```
FROM File
```

```
WHERE Padre=Id;
```

2. Elencare le directory vuote (nome e Id)

```
SELECT f.Nome, f.Id
```

```
FROM File f NATURAL JOIN Directory d
```

```
WHERE f.Id NOT IN (SELECT f1.Padre  
FROM File f1);
```

3. Creare una vista FileDir che restituisce la tabella dei file che sono Directory. Selezionare tutti gli elementi della vista.

```
DROP VIEW IF EXISTS FileDir;
```

```
CREATE VIEW FileDir AS  
  SELECT *  
  FROM File NATURAL JOIN Directory;
```

```
SELECT * FROM FileDir;
```

4. Utilizzare la vista FileDir per risolvere la query 2

```
SELECT d.Nome, d.Id  
FROM FileDir d  
WHERE d.Id NOT IN (SELECT f.Padre  
                  FROM File f);
```

5. Creare la vista FileOrd che restituisce la tabella dei file ordinari (non directory). Selezionare tutti gli elementi della vista.

```
DROP VIEW IF EXISTS FileOrd;
```

```
CREATE VIEW FileOrd AS  
  SELECT *  
  FROM File  
  WHERE Id NOT IN (SELECT * FROM Directory);
```

```
SELECT * FROM FileOrd;
```

6. Elencare gli utenti (login) ed il numero di file posseduti

```
SELECT u.Login, COUNT(f.Id) AS NumFile  
FROM Utenti u LEFT OUTER JOIN File f ON (u.Login = f.Utente)  
GROUP BY u.Login;
```


7. Elencare i gruppi i cui utenti posseggono, complessivamente, il massimo numero di file ordinari

```
SELECT      g.Nome, COUNT(*)
FROM        Gruppi g, GruppiUtenti gu, FileOrd f
WHERE       g.Nome=gu.Gruppo AND gu.Utente=f.Utente
GROUP BY    g.Nome
HAVING      COUNT(*) >=ALL
            (SELECT COUNT(*)
             FROM   GruppiUtenti gu1, FileOrd f1
             WHERE  gu1.Utente=f1.Utente
             GROUP BY gu1.Gruppo);
```

8. Elencare gli utenti che non appartengono a nessun gruppo o a tutti i gruppi

```
SELECT *  
FROM  Utenti u  
WHERE u.Login NOT IN (SELECT gu.Utente FROM GruppiUtenti gu)  
OR  
NOT EXISTS (SELECT *  
            FROM  Gruppi g1  
            WHERE g1.Nome NOT IN (SELECT gu2.Gruppo  
                                   FROM  GruppiUtenti gu2  
                                   WHERE u.Login=gu2.Utente));
```

9. Aggiungere l'utente root al gruppo 'none' e riprovare la query precedente

```
INSERT INTO GruppiUtenti VALUES ('none','root');
```

10. Cancellare le directory vuote e mostrare le directory rimanenti

```
DROP TABLE IF EXISTS Padri;  
CREATE TEMPORARY TABLE Padri AS  
  SELECT f.Padre  
  FROM   File f;
```

```
DELETE FROM File  
WHERE Id IN (SELECT * FROM Directory)  
  AND Id NOT IN (SELECT * FROM Padri);
```

```
/* Mostra le directory rimanenti */  
SELECT *  
FROM   FileDir;
```

11. Cancellare l'utente 'rossi' e verificare che siano stati cancellati i suoi file mostrando la tabella 'File'

```
DELETE FROM Utenti  
WHERE Login='rossi';
```

```
SELECT * FROM FILE;
```