

Cognome:

Nome:

Matricola:

## Basi di dati – Corso di Laurea in Informatica Umanistica

*Esercitazione 1: 25 novembre 2015*

*Parte 2 – SQL (raggruppamento, select annidate, viste)*

Si consideri il seguente schema di base di dati degli utenti di un social network tipo Facebook, i link di amicizia fra gli utenti, i “post” da loro prodotti ed i “like” che gli utenti assegnano ai “post”:

### TABLE Utenti

```
{Codice integer PRIMARY KEY,  
Cognome char(30) NOT NULL,  
Nome char(20) NOT NULL,  
Sesso char(1) NOT NULL,  
Nazionalità char(20) NOT NULL,  
AnnoNascita integer NOT NULL,  
Città char(20) NOT NULL}
```

### TABLE Like

```
{Post integer REFERENCES Post(CodPost),  
Utente integer REFERENCES Utenti(Codice) NOT NULL,  
DataLike Date NOT NULL,  
PRIMARY KEY(Post, Utente) }
```

### TABLE Amici

```
{Utente1 integer REFERENCES Utenti (Codice) NOT NULL,  
Utente2 integer REFERENCES Utenti (Codice) NOT NULL,  
DataInizio Date NOT NULL,  
PRIMARY KEY(Utente1, Utente2) }
```

### TABLE Post

```
{CodPost integer PRIMARY KEY,  
Testo char(256) NOT NULL,  
DataPost Date NOT NULL,  
Autore integer REFERENCES Utenti(Codice)}
```

Si risolvano le seguenti interrogazioni in SQL:

1. Elenco che riporta il numero di utenti di ogni nazionalità.

```
SELECT nazionalita, count(*)  
FROM UTENTI  
GROUP BY nazionalita
```

2. Elencare il codice dei post e il numero di Like (rinominato con Numero) che hanno ricevuto, includendo solo quelli che hanno ricevuto almeno cento Like.

```
SELECT Post, count(*) AS Numero  
FROM Like  
GROUP BY post  
HAVING count(*) >= 100
```

3. Visualizzare la classifica del numero di post scritti da ciascun utente (codice utente, nome e cognome).

```
SELECT Post.autore, Utenti.nome, Utenti.cognome,  
count(Post.autore) As numeropost  
FROM Post JOIN Utenti ON post.codPost = utenti.codice  
GROUP BY Post.autore, Utenti.nome, Utenti.cognome  
Order by numeropost, Post.cognome, Post.nome
```

4. Elencare per ogni autore (codice autore) il numero di post scritti, includendo solo quelli il cui testo inizia con la lettera A e che hanno postato meno di cento Post.

Ovvero:

Elencare per ogni autore (codice autore) il numero di post scritti. Elencare solo quelli che hanno postato meno di cento Post il cui testo inizia con la lettera A.

```
SELECT autore, count(*) as NumeroPost
FROM Post
WHERE testo LIKE 'A%'
GROUP BY autore
HAVING count(*) < 100
```

5. Elencare gli utenti (nome, cognome e anno di nascita) che hanno postato almeno un post e non hanno dato like (usare le select annidate).

```
SELECT nome, cognome, annoNascita
FROM utenti
WHERE codice = ANY [IN] (select autore from Post)
AND codice NOT IN [<> ALL] (select utente from Like)
```

oppure

```
SELECT nome, cognome, annoNascita
FROM utenti
WHERE codice = ANY [IN] (select autore
                        from Post
                        where autore NOT IN [<> ALL] (select utente
                                                    from Like))
```

6. Scrivere la sintassi della vista *vistaNumLike (codicePost, numeroLike)* che contiene per ogni (codice) post il numero di Like ricevuti.

```
CREATE VIEW vistaNumLike (codicePost, numeroLike)
SELECT post, count(*)
FROM Like
GROUP BY post
```

7. Utilizzare la vista *vistaNumLike (codicePost, numeroLike)* per ottenere il codice degli utenti (autore) per cui **almeno uno** dei suoi post ha ricevuto più di 100 like (usare le select annidate).

```
SELECT autore
FROM post
WHERE codPost IN [=ANY] (SELECT codicePost
                        FROM vistaNumLike
                        WHERE numeroLike > 100)
```

8. Utilizzare la vista *vistaNumLike (codicePost, numeroLike)* per ottenere il codice degli utenti (autore) per cui **tutti** i suoi post hanno ricevuto meno di 100 like (usare le select annidate).

```
SELECT autore
FROM post
WHERE codPost <>ALL (SELECT codPost
                   FROM vistaNumLike
                   WHERE numeroLike > 100)
```

9. Elencare il nome o i nomi degli utenti che hanno postato più post. Suggerimento: usare una vista che contiene per ogni codice utente il numero dei post scritti.

```
CREATE VIEW vistaPost (autore, numPost)
SELECT autore, count(*)
FROM post
GROUP BY autore;
```

```
SELECT nome, cognome, codice
FROM utenti JOIN vistaPost ON utenti.codice = vistaPost.autore
WHERE numPost = (SELECT max(numPost) FROM vistaPost)
```

oppure

```
SELECT nome, cognome, codice
FROM utenti
WHERE codice IN (
    SELECT autore
    FROM vistaPost
    WHERE numPost = (SELECT max(numPost)
                    FROM vistaPost) )
```

10. Elencare il codice, il nome, il cognome e il numero di post degli utenti che hanno scritto più post del numero medio di post (ovvero quelli scritti da tutti gli utenti).

```
CREATE VIEW vistaPost (autore, numPost)
SELECT autore, count(*)
FROM post
GROUP BY autore;
```

```
SELECT nome, cognome, codice
FROM utenti JOIN vistaPost ON codice = autore
WHERE numPost > (SELECT avg(numPost) from vistaPost)
```

oppure

```
CREATE VIEW vistaPost (autore, nome, cognome, numPost)
SELECT autore, count(*)
FROM utenti JOIN vistaPost ON codice = autore
GROUP BY autore;
```

```
SELECT nome, cognome, codice
FROM vistaPost
```