

*Laurea in Ingegneria Informatica – SAPIENZA Università di Roma*

*Insegnamento di Basi di Dati*

Esercitazione:

Interrogazioni SQL

Domenico Fabio Savo

# Sommario

---

1. Creazione del database utilizzato per l'esercitazione
2. Interrogazioni SQL

# Creazione del database

---

Creiamo un database denominato “esempio” utilizzando il “client mysql”.

Le istruzioni da utilizzare sono:

```
mysql> CREATE DATABASE esempio;  
Query OK, 1 row effected (0.06 sec)
```

# Creazione delle tabelle (1 / 2)

---

Vogliamo creare le seguenti tabelle:

- **persone(nome, reddito, eta, sesso)**
  - nome è una stringa di 20 caratteri (chiave primaria)
  - reddito è un intero di 10 cifre
  - eta è un intero di 3 cifre
  - sesso è un carattere
- **genitori(figlio,genitore)**
  - figlio (stringa di 20 caratteri, chiave esterna su PERSONE)
  - genitore (stringa di 20 caratteri, chiave esterna su PERSONE)
  - chiave primaria formata da “figlio” e “genitore”

# Creazione delle tabelle (2/2)

---

## - Creazione tabella Persone:

```
mysql> CREATE TABLE Persone(  
        Nome          CHARACTER(20) PRIMARY KEY,  
        Reddito       NUMERIC(10),  
        Eta           NUMERIC(3),  
        Sesso         CHARACTER,  
        );
```

## - Creazione tabella Genitori:

```
mysql> CREATE TABLE Genitori(  
        Figlio        CHARACTER(20) REFERENCES PERSONE (Nome),  
        Genitore      CHARACTER(20) REFERENCES PERSONE (Nome),  
        PRIMARY KEY (Figlio,Genitore)  
        );
```

# Inserimento dei dati nelle tabelle

---

Per inserire dei dati in una tabella si utilizza l'istruzione:

```
INSERT INTO nome_tabella [(nome_attributo1,nome_attributo2,...)]  
VALUES (valore1,valore2,...);
```

## **Attenzione:**

- ▶ L'ordinamento degli attributi (se presente) e dei valori è significativo.
- ▶ Le due liste di attributi e di valori devono avere lo stesso numero di elementi.
- ▶ Se la lista di attributi è omessa, si fa riferimento a tutti gli attributi della relazione secondo l'ordine con cui sono stati definiti.
- ▶ Se la lista di attributi non contiene tutti gli attributi della relazione, per gli altri viene inserito un valore nullo (che deve essere permesso) o un valore di default.

# ES: Inserimento dati

---

Inseriamo alcune tuple nella tabella

▶ **Persone(Nome, Reddito, Eta, Sesso)**

```
mysql> INSERT INTO PERSONE (Nome, Reddito, Eta, Sesso)
VALUES ('Aldo', 25, 15, 'M');
mysql> INSERT INTO PERSONE (Nome, Reddito, Eta, Sesso)
VALUES ('Andrea', 27, 21, 'M');
mysql> INSERT INTO PERSONE (Nome, Reddito, Eta, Sesso)
VALUES ('Luisa', 75, 87, 'F');
mysql> INSERT INTO PERSONE (Nome, Reddito, Eta, Sesso)
VALUES ('Maria', 55, 42, 'F');
```

# Interrogare un database

---

Per effettuare un'interrogazione in SQL si utilizza l'istruzione **SELECT**

```
SELECT nome_attributo,...,nome_attributo  
FROM nome_tabella, ...,nome_tabella  
[WHERE condizione]
```

Le tre parti sono solitamente chiamate:

- ▶ target list
- ▶ clausola from
- ▶ clausola where



# Le ridenominazioni

---

SQL permette di specificare un “alias” degli attributi (nella target list usando il comando **AS**) e delle tabelle (nella clausola FROM).

La ridenominazione è usata per:

1. Ottenere segnature più esplicative nei risultati;
2. Evitare ambiguità;

```
mysql> SELECT p1.nome as omonimoSenior,  
             p2.nome as omonimoJunior  
FROM persone p1, persone p2  
WHERE p1.nome = p2.nome and p1.eta > p2.eta;
```

# Esercitazione – prima parte

---

Date le tabelle:

<b>persone</b>	nome	reddito	eta	sex
<b>genitori</b>	figlio	genitore		

Effettuare le seguenti interrogazioni:

1. Trovare il nome delle persone con più di 30 anni;
2. Trovare nome e sesso delle persone con più di 30 e reddito superiore a 40;
3. Trovare nome e reddito dei genitori di 'Aldo';
4. Trovare la relazione che mostra per ogni persona i nonni;
5. Trovare il reddito medio dei padri per età;
6. Trovare la relazione che mostra le coppie di fratelli (due persone sono fratelli se hanno un genitore in comune).

# Soluzione (1)

---

Trovare il nome e sesso delle persone con più di 30 anni

<b>persone</b>	<b>nome</b>	<b>reddito</b>	<b>eta</b>	<b>sexso</b>
----------------	-------------	----------------	------------	--------------

```
mysql>      SELECT nome
            FROM persone
            WHERE eta > 30;

+-----+
| Nome  |
+-----+
| Olga  |
| Luisa |
| Maria |
+-----+
3 rows in set (0.09 sec)
```

## Soluzione (2)

Trovare nome e sesso delle persone con più di 30 anni e reddito superiore a 40

<b>persone</b>	<b>nome</b>	<b>reddito</b>	<b>eta</b>	<b>sesso</b>
----------------	-------------	----------------	------------	--------------

```
mysql> SELECT nome, sesso
        FROM persone
        WHERE reddito > 40 AND eta > 30 ;
```

```
+-----+-----+
| Nome   | Sesso |
+-----+-----+
| Luisa  | F     |
| Maria  | F     |
+-----+-----+
2 rows in set (0.01 sec)
```

## Soluzione (3)

---

Trovare nome e reddito dei genitori di 'Aldo';

<b>persone</b>	nome	reddito	eta	sezzo
<b>genitori</b>	figlio	genitore		

```
mysql> SELECT genitore, reddito
FROM persone, genitori
WHERE figlio= 'Aldo' AND genitore=nome;
```

## Soluzione (4)

---

Trovare la relazione che mostra per ogni persona i nonni;

<b>persone</b>	<b>nome</b>	<b>reddito</b>	<b>eta</b>	<b>sexso</b>
<b>genitori</b>	<b>figlio</b>	<b>genitore</b>		

```
mysql> SELECT P.Figlio AS Nipote,  
          N.Genitore AS Nonno  
FROM GENITORI P, GENITORI N  
WHERE P.Genitore=N.Figlio;
```

## Soluzione (5)

---

Trovare il reddito medio dei padri per età;

<b>persone</b>	nome	reddito	eta	sezzo
<b>genitori</b>	figlio	genitore		

```
mysql> SELECT eta, AVG(reddito) AS AVG_REDDITO
FROM Persone, Genitori
WHERE nome = genitore AND sesso = 'M'
GROUP BY eta;
```

# Soluzione (6)

---

Trovare la relazione che mostra le coppie di fratelli  
(due persone sono fratelli se hanno un genitore in comune).

<b>persone</b>	<b>nome</b>	<b>reddito</b>	<b>eta</b>	<b>sex</b>
<b>genitori</b>	<b>figlio</b>	<b>genitore</b>		

```
mysql> SELECT distinct G1.Figlio AS Nome1,  
                        G2.Figlio AS Nome2  
FROM GENITORI G1, GENITORI G2  
WHERE G1.Genitore = G2.Genitore and  
      G1.Figlio <> G2.Figlio;
```



# Esercitazione – seconda parte

---

Date le tabelle:

<b>persone</b>	nome	reddito	eta	sesto
<b>genitori</b>	figlio	genitore		

Effettuare le seguenti interrogazioni:

7. Trovare le persone che sono genitori di almeno 2 figli;
8. Trovare la relazione che mostra per ciascun figlio i rispettivi genitori (Padre, Madre, Figlio);
9. Trovare l'elenco ordinato dei genitori i cui figli guadagnano tutti più di 20 milioni.

# Soluzione (7)

---

Trovare le persone che sono genitori di almeno 2 figli;

<b>persone</b>	nome	reddito	eta	sex
<b>genitori</b>	figlio	genitore		

```
mysql> SELECT genitore
FROM Genitori
GROUP BY genitore
HAVING COUNT(figlio) >= 2;
```

# Soluzione (8)

---

Trovare la relazione che mostra per ciascun figlio i rispettivi genitori (Padre, Madre, Figlio);

<b>persone</b>	nome	reddito	eta	sex
<b>genitori</b>	figlio	genitore		

```
mysql> SELECT GP.Genitore AS Padre,  
             GM.Genitore AS Madre, GP.Figlio  
FROM genitori GP, genitori GM,  
     persone PP, persone PM  
WHERE GP.Figlio=GM.Figlio AND  
      GP.Genitore=PP.Nome AND PP.Sesso='M'  
AND  
      GM.Genitore=PM.Nome AND PM.Sesso='F';
```

## Soluzione (9)

---

Trovare l'elenco ordinato dei genitori i cui figli guadagnano tutti più di 20 milioni.

<b>persone</b>	<b>nome</b>	<b>reddito</b>	<b>eta</b>	<b>sesto</b>
<b>genitori</b>	<b>figlio</b>	<b>genitore</b>		

```
mysql> SELECT DISTINCT genitore
FROM Genitori
WHERE genitore NOT IN (SELECT genitore
                        FROM Genitori, Persone
                        WHERE nome = figlio AND
                        reddito <= 20)
ORDER BY genitore;
```