



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO DI MATEMATICA

LAUREA TRIENNALE IN INFORMATICA

BASI DI DATI - LABORATORIO 2

Query avanzate e operatori aggregati

Massimiliano de Leoni

deleoni@math.unipd.it

Alessandro Padella

alessandro.padella@phd.unipd.it

Samuel Cognolato

samuel.cognolato@studenti.unipd.it

Indice

1	Creazione della Base di Dati	4
2	Esercizi	5
3	Soluzioni	6
3.1	Interrogazioni	6

1 Creazione della Base di Dati

Si vuole creare una base di dati che memorizzi una collezione di persone e le informazioni relative alle relazioni genitori-figli esistenti. Occorrerà pertanto creare le seguenti tabelle:

- **Persone:**
 - Id stringa di 2 caratteri, chiave primaria;
 - Nome stringa di 20 caratteri, non può essere nullo;
 - Reddito numero intero in migliaia di euro, 0 per default;
 - Eta numero intero < 200;
 - Sesso 'M' oppure 'F'.
- **Genitori:**
 - Figlio stringa di 2 caratteri, chiave esterna sulla tabella **Persone**;
 - Genitore stringa di 2 caratteri, chiave esterna sulla tabella **Persone**.

Nella definizione delle chiavi esterne è tenuto conto che quando si cancella una persona devono essere eliminati i corrispondenti rapporti di parentela nella tabella **Genitori**. La creazione delle tabelle e la loro popolazione sono definite nel file **persone.sql**, il quale deve essere importato ed eseguito come nel laboratorio 1.

Qui di seguito, alcune interrogazioni utili allo svolgimento degli esercizi a seguire:

- Per cancellare determinate tuple di una tabella:

```
01 |      DELETE FROM NomeTabella WHERE Col1=Val1, Col2=Val2, ...;
```

- Per cancellare un'intera tabella:

```
01 |      DROP TABLE NomeTabella [CASCADE | RESTRICT];
```

- **CASCADE** elimina anche le viste e le tabelle che fanno riferimento alla tabella selezionata;
 - **RESTRICT** fallisce se ci sono righe all'interno della tabella.

- Per aggiungere un valore di default ad una colonna:

```
01 |      ALTER TABLE NomeTabella
02 |      ALTER NomeColonna SET DEFAULT Valore;
```

- Per aggiungere un vincolo sul valore di una colonna:

```
01 |      ALTER TABLE NomeTabella
02 |      ADD CHECK (Col1<Val1 AND Col2=Val2 OR...);
```

- Per creare un nuovo tipo di dato specificando la lista di valori che può assumere:

```
01 |      CREATE TYPE TipoDato AS ENUM('val1', 'val2', ...);
```

2 Esercizi

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

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

Esercizio 3 Restituire per ogni persona, il nome della persona e quello dei nonni (singolo nonno per riga).

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

Esercizio 5 Trovare il reddito medio dei padri per età.

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

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

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

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

Esercizio 10 Modificare la query precedente in modo da riportare tutte le persone, con campo NULL per padre e/o madre quando questi siano assenti.

Esercizio 11 Trovare l'elenco dei genitori i cui figli guadagnano tutti più di 20000 euro. Mostrare identificatore e nome, ordinando la lista per nome.

Esercizio 12 Definire una vista **EtaMediaFigli** che restituisce per ogni genitore (id, nome) l'età media dei suoi figli. Visualizzare il contenuto della vista.

Esercizio 13 Utilizzare la vista **EtaMediaFigli** per trovare tra tutti i genitori i cui figli hanno la massima età media.

Esercizio 14 Cancellare le Persone con reddito maggiore di 20000 euro. Mostrare le tabelle **Persone** e **Genitori** risultanti.

Esercizio 15 Aumentare del 20% tutti i redditi inferiori a 30000 euro. Mostrare la tabella **Persone** risultante.

Esercizio 16 Mostrare l'elenco delle persone che hanno più figli.

Esercizio 17 Mostrare l'elenco dei nonni paterni e dei loro nipoti.

3 Soluzioni

3.1 Interrogazioni

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

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

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

```
01 | SELECT p.Id, p.Nome, p.Reddito
02 | FROM Persone p JOIN Genitori g ON (p.Id=g.Genitore)
03 | WHERE g.Figlio='A3';
```

Esercizio 3 Restituire per ogni persona, il nome della persona e quello dei nonni (singolo nonno per riga).

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

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

```
01 | SELECT p.Sesso, MIN(p.Reddito) AS Minimo, AVG(p.Reddito) as Medio, MAX(
    p.Reddito) as Massimo
02 | FROM Persone p
03 | GROUP BY p.Sesso;
```

Esercizio 5 Trovare il reddito medio dei padri per età.

```
01 | SELECT p.eta, AVG(p.reddito)
02 | FROM persone p JOIN genitori g ON (p.id=g.genitore)
03 | WHERE p.sesso = 'M'
04 | GROUP BY p.eta;
```

In alternativa:

```
01 | SELECT p.Eta, AVG(p.Reddito) AS RedditoMedio
02 | FROM Persone p
03 | WHERE p.Sesso='M' AND EXISTS (SELECT *
04 |     FROM Genitori g
05 |     WHERE g.Genitore=p.Id)
06 | GROUP BY p.Eta;
```

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

```

01 | SELECT DISTINCT p1.Nome, p2.Nome
02 | FROM Persone p1, Genitori g1, Genitori g2, Persone p2
03 | WHERE g1.Figlio=p1.Id AND g2.Figlio=p2.Id AND g1.Genitore=g2.Genitore
    AND p1.Id<p2.Id
04 | ORDER BY p1.Nome;

```

In alternativa:

```

01 | SELECT DISTINCT p1.nome, p2.nome
02 | FROM persone p1 JOIN genitori g ON (g.figlio=p1.id) JOIN genitori g2 ON
    (g2.genitore=g.genitore) JOIN persone p2 ON (g2.figlio=p2.id)
03 | WHERE p1.id < p2.id;

```

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

```

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

```

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

```

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

```

In alternativa:

```

01 | SELECT p.Nome
02 | FROM Persone p
03 | WHERE p.Id IN (SELECT g.Genitore
04 |                FROM Genitori g
05 |                GROUP BY g.Genitore
06 |                HAVING COUNT(*) >1);

```

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

```

01 | SELECT m.Nome AS Madre, p.Nome AS Padre, f.Nome AS Figlio
02 | FROM Persone f, Genitori gm, Persone m, Genitori gp, Persone p
03 | 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';

```

In alternativa:

```

01 | SELECT f1.nome as figlio, m.nome as madre, p.nome as padre
02 | FROM persone f1 join genitori g1 on (f1.id=g1.figlio) JOIN persone m ON
    (m.id=g1.genitore),
03 | persone f2 JOIN genitori g2 on (f2.id=g2.figlio) JOIN persone p ON (p.
    id=g2.genitore)
04 | WHERE m.sesso='F' AND p.sesso='M' AND f1.id=f2.id;

```

Esercizio 10 Modificare la query precedente in modo da riportare tutte le persone, con campo NULL per padre o madre quando questi siano assenti.

```

01 | CREATE VIEW PersoneMadre AS
02 |     SELECT m.Id AS MadreId, m.Nome AS MadreNome, f.Id AS FiglioId, f.
03 |         Nome AS FiglioNome
04 |     FROM Persone f LEFT JOIN (Genitori g JOIN Persone m ON (m.Id=g.
05 |         Genitore)) ON (f.Id=g.Figlio AND m.Sesso='F');
06 |
07 | CREATE VIEW PersonePadre AS
08 |     SELECT p.Id AS PadreId, p.Nome AS PadreNome, f.Id AS FiglioId, f.
09 |         Nome AS FiglioNome
10 |     FROM Persone f LEFT JOIN (Genitori g JOIN Persone p ON (p.Id=g.
11 |         Genitore)) ON (f.Id=g.Figlio AND p.Sesso='M');
12 |
13 | SELECT pm.MadreNome, pp.PadreNome, pm.FiglioNome
14 | FROM PersoneMadre pm, PersonePadre pp
15 | WHERE pm.FiglioId=pp.FiglioId;

```

Esercizio 11 Trovare l'elenco dei genitori i cui figli guadagnano tutti più di 20000 euro. Mostrare identificatore e nome, ordinando la lista per nome.

```

01 | SELECT DISTINCT g.Id, g.Nome
02 | FROM Persone g JOIN Genitori gg1 ON (g.Id=gg1.Genitore)
03 | WHERE NOT EXISTS (SELECT f.Id
04 |     FROM Persone f, Genitori gg
05 |     WHERE f.Id = gg.Figlio AND gg.Genitore=g.Id AND f.Reddito <= 20)
06 | ORDER BY g.Nome;

```

In alternativa:

```

01 | SELECT p.id, p.nome
02 | FROM persone p JOIN genitori g ON (g.genitore=p.id) JOIN persone f ON (
03 |     g.figlio=f.id)
04 | GROUP BY p.id
05 | HAVING MIN(f.reddito) > 20
06 | ORDER BY p.nome;

```

Esercizio 12 Definire una vista EtaMediaFigli che restituisce per ogni persona (id, nome) l'età media dei suoi figli. Visualizzare il contenuto della vista.

```

01 | DROP VIEW IF EXISTS EtaMediaFigli;
02 |
03 | CREATE VIEW EtaMediaFigli AS
04 |     SELECT g.Id, g.Nome, AVG(f.Eta) AS Media
05 |     FROM Persone g, Genitori gg, Persone f
06 |     WHERE g.Id=gg.Genitore AND gg.Figlio=f.Id
07 |     GROUP BY g.Id, g.Nome;
08 |
09 | SELECT * FROM EtaMediaFigli;

```

Esercizio 13 Utilizzare la vista EtaMediaFigli per trovare tra tutti i genitori i cui figli hanno la massima età media.


```
01 | SELECT Nome
02 | FROM EtaMediaFigli
03 | WHERE Media >=ALL (SELECT Media FROM EtaMediaFigli);
```

In alternativa:

```
01 | SELECT Nome
02 | FROM EtaMediaFigli
03 | WHERE Media = (SELECT MAX(Media) FROM EtaMediaFigli);
```

Esercizio 14 Cancellare le **Persone** con reddito maggiore di 20000 euro. Mostrare le tabelle **Persone** e **Genitori** risultanti.

```
01 | DELETE FROM Persone
02 | WHERE Reddito>20;
03 |
04 | SELECT * FROM Persone;
05 |
06 | SELECT * FROM Genitori;
```

N.B: Dopo l'esecuzione di questa query è necessario popolare nuovamente la tabella **Persone**.

Esercizio 15 Aumentare del 20% tutti i redditi inferiori a 30000 euro. Mostrare la tabella **Persone** risultante.

```
01 | UPDATE Persone
02 | SET Reddito=Reddito*1.2
03 | WHERE Reddito<30;
04 |
05 | SELECT * FROM Persone;
```

N.B: Dopo l'esecuzione di questa query è necessario popolare nuovamente la tabella **Persone**.

Esercizio 16 Mostrare l'elenco delle persone che hanno più figli.

```
01 | DROP VIEW IF EXISTS GenitoriFigli;
02 |
03 | CREATE VIEW GenitoriFigli AS
04 | SELECT genitore, COUNT(Figlio) as numeroFigli
05 | FROM Genitori
06 | GROUP BY genitore;
07 |
08 | SELECT Id, nome
09 | FROM GenitoriFigli JOIN Persone on Id=Genitore
10 | WHERE numeroFigli=(SELECT MAX(numeroFigli)
11 | FROM GenitoriFigli)
12 | GROUP BY Id, nome;
```

Esercizio 17 Mostrare l'elenco dei nonni paterni e dei loro nipoti.

```
01 | DROP VIEW IF EXISTS Nonni;
02 |
03 | CREATE VIEW Nonni AS
```

```
04 | SELECT Persone.nome as nomeNonno, f.genitore as nonno, f.figlio as
    | genitore
05 | FROM (Genitori g JOIN Genitori f on g.genitore=f.figlio) JOIN Persone
    | on Persone.Id=f.Genitore;
06 |
07 | SELECT distinct nomeNonno, f.Nome as nomeNipote
08 | FROM ((Persone g JOIN Genitori on g.Id=Genitori.Genitore) JOIN Nonni on
    | Genitori.Genitore=Nonni.Genitore) JOIN Persone f on f.Id=Genitori.
    | Figlio
09 | WHERE g.Sesso='M';
```