



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

massimiliano.deleoni@unipd.it

Matteo Zavatteri

matteo.zavatteri@unipd.it

Alessandro Padella

alessandro.padella@unipd.it

Matteo Gioele Collu

matteogioele.collu@phd.unipd.it

Indice

1	Creazione della Base di Dati	4
2	Esercizi	6
3	Soluzioni	7
3.1	Interrogazioni	7

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 massimo 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', ...);
```

- Per modificare i valori di una colonna, data una certa condizione:

```
01 | UPDATE NomeTabella  
02 | SET NomeColonna = NuovoValore  
03 | WHERE CondizioneSuColonna;
```

2 Esercizi

Esercizio 1 Trovare il reddito minimo, medio e massimo di donne e uomini.

Esercizio 2 Trovare il reddito medio dei genitori di sesso maschile per età.

Esercizio 3 Fornire le coppie di fratelli (due persone sono fratelli se hanno un genitore in comune), indicandone i nomi. Ordinare poi i nomi in ordine alfabetico.

Esercizio 4 Nomi di tutti i fratelli delle persone di nome Aldo.

Esercizio 5 Nomi delle persone che sono genitori di almeno 2 figli.

Esercizio 6 Mostrare per ciascun figlio entrambi i genitori, indicando, per ciascuno, il nome.

Esercizio 7 Modificare la query precedente in modo da riportare tutte le persone, con campo NULL per padre e/o madre quando questi siano assenti.

Esercizio 8 Trovare l'elenco dei genitori i cui figli guadagnano tutti più di 20000 euro. Mostrare identificatore e nome, ordinando la lista per nome.

Esercizio 9 Definire una vista `EtaMediaFigli` che restituisce per ogni genitore (id, nome) l'età media dei suoi figli. Visualizzare il contenuto della vista.

Esercizio 10 Utilizzare la vista `EtaMediaFigli` per trovare tutti i genitori che hanno l'età media dei figli uguale al massimo di quelle di tutti i genitori nella vista.

Esercizio 11 Aumentare del 20% tutti i redditi inferiori a 30000 euro. Mostrare la tabella `Persone` risultante.

Esercizio 12 Mostrare l'elenco delle persone che hanno più figli.

Esercizio 13 Mostrare l'elenco dei nonni relativi ai genitori di sesso maschile e dei loro nipoti.

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

3 Soluzioni

3.1 Interrogazioni

Esercizio 1 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 2 Trovare il reddito medio dei genitori di sesso maschile per età.

```
01 | SELECT p.eta, AVG(p.reddito) AS RedditoMedio
02 | FROM persone p
03 | WHERE p.sesso = 'M' AND p.id IN (SELECT g.Genitore FROM GENITORI)
04 | GROUP BY p.eta;
```

In alternativa, ma molto meno efficiente:¹

```
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 3 Fornire le coppie di fratelli (due persone sono fratelli se hanno un genitore in comune), indicandone i nomi. Ordinare poi i nomi in ordine alfabetico.

```
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 4 Nomi di tutti i fratelli delle persone di nome Aldo.

```
01 | SELECT DISTINCT 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);
```

¹Quando si usano le query annidate in cui la query interna si riferisce alle tabelle della query esterna, il sistema di gestione delle basi di dati non è in grado di ottimizzare.

In alternativa, sfruttando l'esercizio precedente per creare una vista:

```
01 | CREATE VIEW fratelli as
02 | SELECT DISTINCT p1.nome as nome, p2.nome as nomeFratello
03 | 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)
04 | WHERE p1.id <> p2.id;
05 |
06 | Select f.nomeFratello
07 | From fratelli as f
08 | Where nome='Aldo';
```

Esercizio 5 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 6 Mostrare per ciascun figlio entrambi i genitori, indicando, per ciascuno, il nome.

```
01 | SELECT p1.nome AS genitore1, p2.nome AS genitore2, p3.nome AS figlio
02 | FROM persone p1, persone p2, persone p3, genitori g1, genitori g2
03 | WHERE g1.genitore=p1.id AND g2.genitore=p2.id AND g1.figlio=p3.id AND
    g2.figlio=p3.id AND g1.genitore < g2.genitore
04 | ORDER BY figlio;
```

Esercizio 7 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.
    Nome AS FiglioNome
03 | FROM Persone f LEFT JOIN (Genitori g JOIN Persone m ON (m.Id=g.
    Genitore)) ON (f.Id=g.Figlio AND m.Sesso='F');
04 |
05 | CREATE VIEW PersonePadre AS
06 | SELECT p.Id AS PadreId, p.Nome AS PadreNome, f.Id AS FiglioId, f.
    Nome AS FiglioNome
07 | FROM Persone f LEFT JOIN (Genitori g JOIN Persone p ON (p.Id=g.
    Genitore)) ON (f.Id=g.Figlio AND p.Sesso='M');
08 |
09 | SELECT pm.MadreNome, pp.PadreNome, pm.FiglioNome
10 | FROM PersoneMadre pm, PersonePadre pp
11 | WHERE pm.FiglioId=pp.FiglioId;
```


Esercizio 8 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 p.Id, p.Nome
02 | FROM Persone p JOIN Genitori gg1 ON (p.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=p.Id AND f.Reddito <= 20)
06 | ORDER BY p.Nome;
```

Esercizio 9 Definire una vista `EtaMediaFigli` che restituisce per ogni genitore (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 10 Utilizzare la vista `EtaMediaFigli` per trovare tutti i genitori che hanno l'età media dei figli uguale al massimo di quelle di tutti i genitori nella vista.

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

Esercizio 11 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;
```

Esercizio 12 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 13 Mostrare l'elenco dei nonni relativi ai genitori di sesso maschile 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';
```

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;
```

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