



UNIVERSITÀ DEGLI STUDI DI TRENTO

Dipartimento di Ingegneria e Scienza dell'Informazione

Corso di Laurea in
Informatica

ESERCIZI DI

DATABASE
Prof.re Yannis Velegarakis

Autore
Emanuele Nardi

Revisore

Anno accademico 2017/2018

Introduzione

Questo è un estratto dal capitolo 4.3 "Interrogazioni SQL" del libro "Basi di Dati" edito da McGrawHill IV ed. versione italiana.

Lo scopo principale di questi appunti è quello di svolgere esercizio in preparazione all'esame di Database tenuto all'Università degli Studi di Trento.

Queste note non sono complete, e la loro lettura non permette, da sola, di superare l'esame. La versione più recente di queste note si trova all'indirizzo:

<https://github.com/emanuelenardi/latex-sql>

Materiale

Per tutto il resto consulta la cartella Google Drive del corso di Informatica [↗](#).

Segnalazione di errori

Se hai trovato un errore ti prego di inviarmi un'e-mail [↗](#) allegando un esempio che possa riprodurre l'errore.

Riguardo l'autore

Emanuele Nardi è uno studente di informatica all'Università degli Studi di Trento, Rappresentante degli studenti e co-fondatore di Speck&Tech, la comunità tecnologica di Trento.

Ha prodotto diverse dispense per il corso di studi di informatica, fra le quali:

- programmazione funzionale
- programmazione ad oggetti
- laboratorio di algoritmi
- laboratorio di database
- laboratorio di web
- Linguaggi formali e compilatori
- Android

Puoi trovare altri contatti dell'ateneo sulla pagina del DISI [↗](#) .

Tabelle

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giovanni	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Gialli	Direzione	7	73	Genova
Paola	Rosati	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Table 1: IMPIEGATO

CodFiscale	Nome	Cognome	Città
x	Mario	Rossi	Verona
x	Carlo	Bianchi	Roma
x	Giovanni	Rossi	Verona
x	Pietro	Rossi	Milano

Table 2: PERSONA

Nome	Indirizzo	Città
Amministrazione	Via Tito Livio, 27	Milano
Produzione	P.le Lavater, 3	Torino
Distribuzione	Via Sagra, 9	Roma
Direzione	Via Tito Livio, 27	Milano
Ricerca	Via Venosa, 6	Milano

Table 3: DIPARTIMENTO

Interrogazioni semplici

1. Estrarre lo stipendio degli impiegati di cognome "Rossi".

```
SELECT Stipendio AS Salario
FROM Impiegato
WHERE Cognome = 'Rossi'
```

La query produce la seguente tabella:

Salario
45
80

2. Estrarre tutte le informazioni relative agli impiegati di cognome "Rossi".

```
SELECT *
FROM Impiegato
WHERE Cognome = 'Rossi'
```

La query produce la seguente tabella:

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Rossi	Direzione	14	80	Milano

3. Estrarre lo stipendio mensile dell'impiegato che ha cognome "Bianchi".

```
SELECT Stipendio/12 AS StipendioMensile
FROM Impiegato
WHERE Cognome = 'Bianchi'
```

La query produce la seguente tabella:

StipendioMensile
3,00

4. Estrarre il nome degli impiegati e le città in cui lavorano.

```
SELECT Impiegato.Nome, Impiegato.Cognome, Dipartimento.Città
FROM Impiegato, Dipartimento
WHERE Impiegato.Dipart = Dipartimento.Nome
```

La query produce la seguente tabella:

Impiegato.Nome	Impiegato.Cognome	Dipartimento.Città
Mario	Rossi	Milano
...
Marco	Franco	Torino

È necessario specificare il nome della tabella quando le tabelle presenti nella clausola **FROM** posseggono più attributi con lo stesso nome. Qualora non vi sia possibilità di ambiguità è possibile specificare l'attributo senza dichiarare la tabella di appartenenza.

5. L'interpretazione precedente può essere espressa facendo uso degli *alias* per la tabella allo scopo di abbreviare i riferimenti a esse.

```
SELECT I.Nome, Cognome, D.Citta
FROM   Impiegato AS I, Dipartimento AS D
WHERE  Dipart = D.Nome -- non presentano ambiguita'
```

Utilizzando gli alias è possibile fare accesso più volte alla stessa tabella, come avviene nel calcolo relazionale. Tutte le volte che si introduce un alias per una tabella si dichiara in effetti una variabile che rappresenta le righe della tabella di cui è alias.

Quando una tabella una sola volta in un interrogazione, non c'è differenza tra l'interpretare l'alias come uno pseudonimo o come una nuova variabile.

Quando una tabella compare invece più volte, è necessario considerare l'alias come una nuova variabile.

6. Estrarre il nome e il cognome degli impiegati che lavorano nell'ufficio 20 del dipartimento Amministrazione.

```
SELECT Nome, Cognome
FROM   Impiegato
WHERE  Ufficio = 20
AND    Dipart = 'Amministrazione'
```

La query produce la seguente tabella:

Nome	Cognome
Giovanni	Verdi

7. Estrarre i nomi e i cognomi degli impiegati che lavorano nel dipartimento Amministrazione o nel dipartimento Produzione.

```
SELECT Nome, Cognome
FROM   Impiegato
WHERE  Dipart = 'Amministrazione'
AND    Dipart = 'Produzione'
```

La query produce la seguente tabella:

Nome	Cognome
Mario	Rossi
Carlo	Bianchi
Giovanni	Verdi
Paola	Rosati
Marco	Franco

8. Estrarre i nomi propri degli impiegati di cognome "Rossi" che lavorano nei dipartimenti Amministrazione e Produzione.

```
SELECT Nome
FROM Impiegato
WHERE Cognome = 'Rossi'
AND (Dipart = 'Amministrazione' OR Dipart = 'Produzione')
```

La query produce la seguente tabella:

Nome

Mario

9. Estrarre gli impiegati che hanno un cognome che ha una "o" in seconda posizione e finisce per "i".

```
SELECT *
FROM Impiegato
WHERE Cognome LIKE '_o%i'
```

Nome	Cognome	Dipart	Ufficio	Stipendio	Citta
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Rossi	Direzione	14	80	Milano
Paola	Rosati	Amministrazione	75	40	Venezia

10. Estrarre le città delle persone il cui cognome è "Rossi" presentando eventualmente più volte lo stesso valore di città.

```
SELECT Citta
FROM Persona
WHERE Cognome = 'Rossi'
```

Citta

Verona

Verona

Milano

11. Estrarre le città delle persone con cognome "Rossi", facendo comparire ogni città al più una volta.

```
SELECT DISTINCT Citta
FROM Persona
WHERE Cognome = 'Rossi'
```

Citta

Verona

Milano

17. Estrarre tutti gli impiegati che hanno lo stesso cognome (ma diverso nome) di impiegati del dipartimento Produzione.

```
SELECT I1.Cognome, I1.Nome
FROM    Impiegato I1, Impiegato I2
WHERE   I1.Cognome = I2.Cognome
         AND I1.Nome <> I2.Cognome
         AND I2.Dipart = 'Produzione'
```

Operatori Aggregati

20. Estrarre il no. di impiegati del dipartimento Produzione.

```
SELECT count(*)
FROM   Impiegato
WHERE  Dipart = 'Produzione'
```

Il numero di impiegati corrisponderà al numero di tuple della relazione **IMPIEGATO** che possiedono "Produzione" come valore dell'attributo **Dipart**.

Operatori aggregati:

- (a) count
- (b) sum
- (c) max
- (d) min
- (e) avg

21. Estrarre il no. di diversi valori dell'attributo Stipendio fra tutte le righe di impiegato.

```
SELECT count(DISTINCT Stipendio)
FROM   Impiegato
```

22. Estrarre il no. di righe che possiedono un valore non nulla per l'attributo Nome.

```
SELECT count(ALL Nome)
FROM   Impiegato
```

23. Estrarre la somma degli stipendi del dipartimento Amministrazione.

```
SELECT sum(Stipendio)
FROM   Impiegato
WHERE  Dipart = 'Amministrazione'
```

24. Estrarre gli stipendi minimo, massimo e medio fra quelli di tutti gli impiegati.

```
SELECT min(Stipendio), max(Stipendio), avg(Stipendio)
FROM   Impiegato
```

25. Estrarre il massimo stipendio tra quelli degli impiegati che lavorano in un dipartimento con sede a Milano.

```
SELECT max(Stipendio)
FROM   Impiegato, Dipartimento
WHERE  Dipartimento.Citta = 'Milano'
      AND Dipart = D.Nome -- corrispondenza fra tabelle
```

Interrogazioni con raggruppamento

La sintassi SQL non ammette che nella stessa clausola **SELECT** compaiano funzioni aggregate espressioni al livello di riga, a meno che non si faccia uso della **GROUP BY**.

27. Estrarre la somma degli stipendi di tutti gli impiegati dello stesso dipartimento

```
SELECT  Dipart, sum(Stipendio)
FROM    Impiegato
GROUP BY Dipart
```

Una applicazione può aver bisogno di considerare solo i sottoinsiemi che soddisfano certe condizioni, questo ci porta alla prossima interrogazione.

31. Estrarre i dipartimenti che spendono più di 10 in Stipendi

```
SELECT  Dipart, sum(Stipendio)
FROM    Impiegato
GROUP BY Dipart
HAVING  sum(Stipendio) > 100
```

La clausola dettata dall'HAVING è *indipendente* da quella del SELECT, quindi potrei restituire anche un'altra proprietà anche se non avrebbe molto senso.

32. Estrarre i dipartimenti per cui la media degli stipendi degli impiegati che lavorano nell'ufficio 20 è superiore a 25.

```
SELECT  Dipart
FROM    Impiegato
WHERE    Ufficio = '20'
GROUP BY Dipart
HAVING  avg(Stipendio) > 25
```

Interrogazioni di tipo insiemistico

- union
- intersect
- except (chiamato anche minus)

Tutte le interrogazioni scritte con questi due costrutti possono essere riscritte tramite interrogazioni nidificate.

33. Estrarre i nomi e i cognomi dagli impiegati.

```
SELECT  Nome
FROM    Impiegato
UNION
SELECT  Cognome
FROM    Impiegato
```

Nome
Mario
Carlo
Giovanni
Franco
Lorenzo
Paola
Marco
Rossi
Bianchi
Verdi
Neri
Gialli
Rosato

34. Estrarre i nomi e i cognomi di tutti gli impiegati, eccetto quelli appartenenti al dipartimento Amministrazione, **mantenendo i duplicati**.

```
SELECT Nome
FROM Impiegato
UNION ALL
SELECT Cognome
FROM Impiegato
WHERE Dipart <> 'Amministrazione'
```

35. Estrarre i cognomi di impiegati che sono anche nomi.

```
SELECT Nome
FROM Impiegato
INTERSECT
SELECT Cognome
FROM Impiegato
```

Nome
Franco

36. Estrarre i nomi degli impiegati che non sono cognomi di qualche impiegato.

```
SELECT Nome
FROM Impiegato
EXCEPT (minus)
SELECT Cognome
FROM Impiegato
```

Nome
Mario
Carlo
Giovanni
Lorenzo
Paola
Marco

Interrogazioni nidificate

37. Estrarre gli impiegati che lavorano in dipartimenti situati a Firenze.

```
SELECT *
FROM Impiegato
WHERE Dipart = any (SELECT Nome
                     FROM Dipartimento
                     WHERE Citta = 'Firenze')
```

L'interrogazione seleziona le righe di **Impiegato** per cui il valore dell'attributo **Dipart** è uguale ad almeno uno dei valori dell'attributo **Nome** delle righe di **Dipartimento**.

38. Consideriamo un'interrogazione che permetta di trovare li impiegati che hanno lo stesso nome di un impiegato del Dipartimento Produzione ...

```
SELECT I1.Nome
FROM Impiegato I1, Impiegato I2
WHERE I1.Nome = I2.Nome
      AND I2.Dipart = 'Produzione'
```

39. ... con operazioni insiemistiche

```
SELECT Nome
FROM Impiegato
WHERE Nome = any (SELECT Nome
                  FROM Impiegato
                  WHERE Dipart = 'Produzione')
```

40. Estrarre i nomi dei dipartimenti in cui non lavorano persone di cognome "Rossi" ...

```
SELECT Nome
FROM Dipartimento
EXCEPT
SELECT Dipart
FROM Impiegato
WHERE Cognome = 'Rossi'
```

41. ... con operazioni insiemistiche

```
SELECT Nome
FROM Dipartimento
WHERE Nome <> all (SELECT Dipart
                  FROM Impiegato)
```

```
WHERE Cognome = 'Rossi')
```

Simboli di appartenenza/non appartenenza rispetto ad un insieme:

- = **any** \Leftrightarrow **in**
- <> **all** \Leftrightarrow **not in**

Interrogazioni che fanno uso degli operatori max e min possono essere rappresentate senza l'uso degli operatori stessi.

42. Estrarre il dipartimento dell'impiegato che guadagna lo stipendio massimo usando l'operatore max ...

```
SELECT Dipart
FROM   Impiegato
WHERE  Stipendio = any (SELECT max(Stipendio)
                        FROM   Impiegato)
```

43. ... usando solo un'interrogazione nidificata

```
SELECT Dipart
FROM   Impiegato
WHERE  Stipendio >= all (SELECT Stipendio
                        FROM   Impiegato)
```

44. Estrarre le persone che **hanno** degli omonimi (stesso nome e cognome ma CF diverso) ...

```
SELECT *
FROM   Persona P
WHERE  EXISTS (SELECT *
               FROM   Persona P1
               WHERE  P.Nome = P1.Nome
                   AND P.Cognome = P1.Cognome
                   AND P.CF <> P1.CF)
```

Per ogni chiamata nidificata devi controllare il valore della tabella superiore.

45. ... senza query nidificata

```
SELECT P.*
FROM   Persona P, Persona P1
WHERE  P1.Nome = P.Nome
    AND P1.Cognome = P.Cognome
    And P1.CF <> P.CF
```

46. Estrarre le persone che **non hanno** degli omonimi.

```
SELECT *
FROM   Persona P
WHERE  NOT EXISTS (SELECT *
                  FROM   Persona P1
                  WHERE  P.Nome = P1.Nome
                      AND P.Cognome = P1.Cognome
                      AND P.CF <> P1.CF)
```

Conclusione

Abbiamo raggiunto la fine del nostro viaggio, mio caro amico. Spero che tu abbia imparato qualcosa. Ma prima che te ne vada, considera di fare una donazione all'autore di questa dispensa che ha pensato di usare il proprio tempo per migliorare la vita degli studenti di questo dipartimento.

Sei anche tu uno studente squattrinato fuori sede? Ecco cosa puoi fare:

- *Segnalare* errori contenuti del testo e/o negli esercizi;
- *Correggere* gli errori contenuti del testo e/o negli esercizi;
- Condividere questa dispensa con un compagno di corso.

Ci vediamo!

Emanuele