

Sistemi Informativi

Alessandro Lori

C.d.L. Ing. Gestionale
Università di Pisa

17 Marzo 2008
Lab 03



UNIVERSITÀ DI PISA

Outline

- 1 Query SQL
 - con op. insiemistici
 - annidate
 - su valori aggregati

- 2 Trigger
 - introduzione
 - esempio

SQL

Structured Query Language:

- sviluppato originariamente da IBM e diventato standard *de facto*;
- standard ANSI/ISO in continua evoluzione (SQL:2006);
- SQL:1999 ha una collezione di funzioni chiamate *CoreSQL* ed una serie di pacchetti.

SQL (II)

Alcune componenti dello standard:

- Data Definition Language (DDL);
- Data Manipulation Language (DML);
- Trigger e VI avanzati;

Formulazione di una query

```
SELECT [DISTINCT] lista-select  
FROM lista-from  
WHERE qualificazione
```

- La **lista-from** contiene nomi di tabelle, eventualmente seguite da *variabili di range*;
- la **lista-select** specifica nomi di colonne delle tabelle nominate nella lista-from;
- la **qualificazione** nella clausola `WHERE` è una combinazione booleana di condizioni nella forma *espressione* `op` *espressione*. Un'*espressione* è un nome di colonna, una costante o un'espressione aritmetica o di stringa.

Valutazione di una query

Strategia di valutazione concettuale:

- 1 Prodotto cartesiano delle tabelle in **lista-from**;
- 2 cancellazione righe che non soddisfano le condizioni in **qualificazione**;
- 3 cancellazione colonne non in **lista-select**;
- 4 se specificato `DISTINCT`, cancellazione righe duplicate.

query di base

Impiegato(matricola, nome, cognome, dipartimento, ufficio, stipendio)

Trovare nome e cognome degli impiegati che percepiscono uno stipendio maggiore di 30 mila euro:

```
SELECT I.nome, I.cognome  
FROM Impiegato I  
WHERE I.stipendio > 30
```

inner join

Impiegato(matricola, nome, cognome, dipartimento, ufficio, stipendio)
Sedi(dipartimento, indirizzo, citta)

Trovare nome e cognome degli impiegati che lavorano a Pisa:

```
SELECT I.nome, I.cognome  
FROM Impiegato I, Sedi S  
WHERE I.dipartimento = S.dipartimento AND S.citta = "Pisa"
```

inner join (II)

Impiegato(matricola, nome, cognome, dipartimento, ufficio, stipendio)
Supervisione(capo, impiegato)

Trovare nome, cognome e stipendio dei capi degli impiegati che guadagnano più di 40 mila euro:

```
SELECT DISTINCT C.nome, C.cognome, C.stipendio  
FROM Impiegato C, Supervisione S, Impiegato I  
WHERE C.matricola = S.capo AND  
      S.impiegato = I.matricola AND  
      I.stipendio > 40
```

inner join (III)

Impiegato(matricola, nome, cognome, dipartimento, ufficio, stipendio)
Sedi(dipartimento, indirizzo, citta)
Supervisione(capo, impiegato)

Trovare nome, cognome e stipendio dei capi che lavorano a Firenze ed i cui impiegati guadagnano tra i 30 ed i 40 mila euro (estremi inclusi):

```
SELECT DISTINCT C.nome, C.cognome, C.stipendio  
FROM Impiegato C, Sedi SE, Supervisione S, Impiegato I  
WHERE C.matricola = S.capo AND  
      C.dipartimento = SE.dipartimento AND  
      S.impiegato = I.matricola AND  
      SE.citta = "Firenze" AND  
      I.stipendio > 30 AND I.stipendio < 40
```

outer join

Impiegato(matricola, nome, cognome, dipartimento, ufficio, livello)
Sedi(dipartimento, indirizzo, citta)

Trovare nome, cognome e città di lavoro degli impiegati di livello maggiore di 5, visualizzando comunque gli impiegati che non sono per il momento associati a nessun dipartimento:

```
SELECT I.nome, I.cognome  
FROM Impiegato I NATURAL LEFT OUTER JOIN Sede S  
WHERE I.livello > 5
```

UNION

Impiegato(matricola, nome, cognome, dipartimento, ufficio, livello)
Supervisione(capo, impiegato)

Trovare nome e cognome dei capi che supervisionano l'impiegato di matricola 929402 o l'impiegato di matricola 239582:

```
SELECT C.nome, C.cognome  
FROM Impiegato C, Supervisione S  
WHERE C.matricola = S.capo AND  
      S.impiegato = 929402
```

UNION

```
SELECT C.nome, C.cognome  
FROM Impiegato C, Supervisione S  
WHERE C.matricola = S.capo AND  
      S.impiegato = 239582
```

INTERSECT

Impiegato(matricola, nome, cognome, dipartimento, ufficio, livello)
Supervisione(capo, impiegato)

Trovare nome e cognome dei capi che supervisionano l'impiegato di matricola 929402 e l'impiegato di matricola 239582:

```
SELECT C.nome, C.cognome  
FROM Impiegato C, Supervisione S  
WHERE C.matricola = S.capo AND  
      S.impiegato = 929402
```

INTERSECT

```
SELECT C.nome, C.cognome  
FROM Impiegato C, Supervisione S  
WHERE C.matricola = S.capo AND  
      S.impiegato = 239582
```

EXCEPT

Impiegato(matricola, nome, cognome, dipartimento, ufficio, livello)
Supervisione(capo, impiegato)

Trovare nome e cognome dei capi che supervisionano l'impiegato di matricola 929402, ma non l'impiegato di matricola 239582:

```
SELECT C.nome, C.cognome  
FROM Impiegato C, Supervisione S  
WHERE C.matricola = S.capo AND  
       S.impiegato = 929402
```

EXCEPT

```
SELECT C.nome, C.cognome  
FROM Impiegato C, Supervisione S  
WHERE C.matricola = S.capo AND  
       S.impiegato = 239582
```

Query annidate

Impiegato(matricola, nome, cognome, dipartimento, ufficio, livello)
Supervisione(capo, impiegato)

Trovare nome e cognome dei capi che supervisionano l'impiegato di matricola 239582:

```
SELECT C.nome, C.cognome
FROM Impiegato C
WHERE C.matricola IN
      (SELECT S.capo
       FROM Supervisione S
       WHERE S.impiegato = 239582)
```

Query annidate (II)

Impiegato(matricola, nome, cognome, dipartimento, ufficio, livello)
Supervisione(capo, impiegato)

Trovare nome e cognome dei capi che supervisionano impiegati di nome "Mario":

```
SELECT C.nome, C.cognome
FROM Impiegato C
WHERE C.matricola IN
    (SELECT DISTINCT S.capo
     FROM Supervisione S
     WHERE S.impiegato IN
         (SELECT I.matricola
          FROM Impiegato I
          WHERE I.Nome = "Mario"))
```

Confronto tra insiemi

Impiegato(matricola, nome, cognome, dipartimento, ufficio, livello)

Trovare nome e cognome degli impiegati il cui livello è maggiore del livello di qualunque altro impiegato di nome "Antonio":

```
SELECT I1.nome, I1.cognome
FROM Impiegato I1
WHERE I1.livello > ANY
  (SELECT I2.livello
   FROM Impiegato I2
   WHERE I2.Nome = "Antonio")
```

Operatori di aggregazione

Impiegato(matricola, nome, cognome, eta, ufficio, livello)

Trovare l'età media degli impiegati il cui livello è maggiore di 5:

```
SELECT AVG(l.eta)
FROM Impiegato l
WHERE l.livello > 5
```

Clausola GROUP BY

Impiegato(matricola, nome, cognome, eta, dipartimento, livello)
Sedi(dipartimento, indirizzo, citta)

Trovare l'età media per ogni livello degli impiegati con sede a Pisa:

```
SELECT I.livello, AVG(I.eta) AS media_eta  
FROM Impiegato I NATURAL INNER JOIN Dipartimento D  
WHERE D.citta = "Pisa"  
GROUP BY I.livello
```

Clausola GROUP BY (II)

Impiegato(matricola, nome, cognome, eta, dipartimento, livello)
Sedi(dipartimento, indirizzo, citta)

Trovare l'età media per ogni livello con almeno 2 impiegati con sede a Pisa:

```
SELECT I.livello, AVG(I.eta) AS media_eta  
FROM Impiegato I NATURAL INNER JOIN Dipartimento D  
WHERE D.citta = "Pisa"  
GROUP BY I.livello  
HAVING COUNT(*) > 1
```

Trigger

Trigger: procedura eseguita dal DBMS in risposta ad un evento in basi di dati *attive*.

Esempi d'utilizzo:

- campo contatore con autoincremento;
- invio automatico di ordini in caso un magazzino scenda sotto una certa giacenza;
- statistiche o log;
- tabelle di backup attivate in caso di aggiornamento o cancellazione di tabelle da un database in maniera trasparente all'utente.

Trigger (II)

Struttura di un trigger:

- **evento**: cambiamento nella base di dati che *attiva* il trigger;
- **condizione**: test o query eseguita all'attivazione del trigger;
- **azione**: procedura eseguita all'attivazione del trigger se la condizione è soddisfatta.

Trigger (III)

- Può essere attivato da un inserimento, cancellazione o da un aggiornamento, indipendentemente dall'utente e dall'applicazione che ha causato l'evento;
- se la condizione è una query è *vera* se l'insieme di risposta è *non vuoto*;
- l'azione può esaminare la risposta all'interrogazione, accedere a vecchi e nuovi valori delle tuple modificate, modificare la base di dati o chiamare procedure nel linguaggio ospite;
- le procedure possono essere definite usando il linguaggio preferito (oltre a PL/pgSQL).

Trigger (IV)

L'azione del trigger può essere eseguita:

- **prima** (BEFORE) o **dopo** (AFTER) il comando attivante;
- **a livello di riga** (FOR EACH ROW) o **a livello di istruzione** (FOR EACH STATEMENT).

es. trigger

```
CREATE FUNCTION archivia_impiegati() RETURNS trigger AS $archivia_impiegati$
BEGIN
    -- Aggiorna la tabella di backup
    INSERT INTO ex_impiegati SELECT OLD.nome, OLD.cognome, OLD.matricola;
    RETURN OLD;
END
$archivia_impiegati$ LANGUAGE plpgsql;

CREATE TRIGGER archivia_impiegati
AFTER DELETE ON impiegati
FOR EACH ROW EXECUTE PROCEDURE archivia_impiegati();
```