



Basi di Dati

Esercitazione in Lab n.5

Prof. Mauro Conti

Dipartimento di Matematica - Università degli studi di Padova

conti@math.unipd.it - <http://www.math.unipd.it/~conti>

Giuseppe Cascavilla, PhD student

sites.google.com/site/gcascavilla



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- **Funzione**
- **Trigger**

Definire una funzione che dato un Id di file, restituisce il corrispondente pathname (es. per 121, che corrisponde a 'tmp0.txt' dovrebbe restituire '/Radice/tmp/tmp0.txt').

Possibile intestazione della funzione

```
CREATE FUNCTION givePath (FId INT) RETURNS  
CHAR(255).
```

Info per realizzare la funzione dovreste usare:

DECLARE: statement per dichiarare variabili locali all'interno della funzione

SET: assegna un valore a una variabile definita in **DECLARE** statement

REPEAT: crea un ciclo fino a quando *search_condition* diventa **true**

sintassi:

REPEAT

`statement_list`

UNTIL `search_condition`

END REPEAT [`end_label`]

Manuale: <https://dev.mysql.com/doc/refman/5.0/en/repeat.html>

Definire un trigger che assicuri la validità del seguente vincolo semantico: non ci possono essere più di 256 utenti.

Possibile intestazione del trigger

```
CREATE TRIGGER MaxUsers BEFORE INSERT ON Utenti  
FOR EACH ROW.
```

Info per realizzare il trigger dovreste usare:

DECLARE: dichiara variabile

INTO: salva risultato di una `select` in una variabile

LIMIT 1: limita la visualizzazione del risultato a una sola riga

**CREATE TRIGGER [nomeTrigg] [Before/After] INSERT ON
[Tab] FOR EACH STATEMENT**

dal **manuale** MySQL: <https://dev.mysql.com/doc/refman/5.1/en/faqs-triggers.html>

A.5.3: Does MySQL 5.1 have statement-level or row-level triggers?

In MySQL 5.1, all triggers are **FOR EACH ROW**—that is, the trigger is activated for each row that is inserted, updated, or deleted. MySQL 5.1 does not support triggers using **FOR EACH STATEMENT**.

per creare il Trigger, potrebbe esservi utile il link al manuale MySQL:
<https://dev.mysql.com/doc/refman/5.1/en/create-trigger.html>

```
DROP FUNCTION IF EXISTS givePath;
```

```
DELIMITER |
```

```
CREATE FUNCTION givePath (FId INT)  
RETURNS CHAR(255)  
BEGIN  
  DECLARE mypath VARCHAR(255)  
  DEFAULT "";  
  DECLARE cur_name VARCHAR(20);  
  DECLARE cur_id, prev_id INT;
```

```
  SET cur_id = FId;
```

```
  REPEAT  
    SET prev_id=cur_id;
```

```
    SELECT .....  
    FROM .....  
    WHERE .....
```

```
    SET mypath=CONCAT('/', cur_name,  
    mypath);
```

```
    SELECT .....  
    FROM .....  
    WHERE .....
```

```
  UNTIL cur_id = prev_id  
  END REPEAT;
```

```
    /* Nota: se l'Id passato non corrisponde  
    ad un file 'cur_id'
```

```
    /* diventa NULL e si esce subito */
```

```
    RETURN mypath;  
  END|
```

```
DELIMITER ;
```

DELIMITER |

```
CREATE TRIGGER maxUsers  
BEFORE INSERT ON Utenti  
FOR EACH ROW  
BEGIN  
DECLARE numUser INT;
```

```
SELECT ..... FROM ..... INTO numUser;
```

```
IF [condizione] THEN  
INSERT INTO ..... SELECT * FROM ..... LIMIT 1;  
END IF;
```

```
END|
```

```
DELIMITER ;
```