

Quarta Esercitazione di Basi di Dati

Giorgia Nadizar

6 maggio 2022

Parte 1

Riepilogo

Tabelle nel DB

studenti(matricola, nome, cognome, cf)

professori(matricola, nome, cognome, cf, settore)

corsi(codice, nome, cfu, professore)

esami(corso, studente, data, voto, lode)

Tabelle nel DB

studenti(matricola, nome, cognome, cf)

professori(matricola, nome, cognome, cf, settore)

corsi(codice, nome, cfu, professore)

esami(corso, studente, data, voto, lode)

Abbiamo già il DB pronto dalle esercitazioni precedenti, ma ricordiamoci di *usarlo*:

```
USE uni_db;
```

Parte 2

Esercizi in aula

User Defined Function 1

Scrivere una user defined function che restituisca il corso di laurea di uno studente

User Defined Function 1

Scrivere una user defined function che restituisca il corso di laurea di uno studente

```
DELIMITER $$

CREATE FUNCTION cdl(matricola char(9))
RETURNS CHAR(4) DETERMINISTIC
BEGIN
    RETURN SUBSTRING(matricola, 1, 4);
END $$

DELIMITER ;
```

User Defined Function 2

Scrivere una user defined function che restituisca la media ponderata di uno studente

User Defined Function 2

```
DELIMITER $$

CREATE FUNCTION media_ponderata(matricola char(9)
)
RETURNS float DETERMINISTIC
BEGIN
    DECLARE mp float;
    SELECT SUM(c.cfu * e.voto) / SUM(c.cfu)
    INTO mp
    FROM esami e INNER JOIN corsi s
    ON e.corso = s.codice
    WHERE e.studente = matricola;
    RETURN (mp);
END $$

DELIMITER ;
```

User Defined Function 3

Scrivere una user defined function che restituisca il rank di uno studente nel suo corso di laurea in base alla sua media ponderata

User Defined Function 3

```
DELIMITER $$

CREATE FUNCTION rank_cdl(matricola char(9))
RETURNS INT DETERMINISTIC
BEGIN
    DECLARE r INT;
    SELECT COUNT(*)
    INTO r
    FROM studenti s
    WHERE cdl(s.matricola) = cdl(matricola) AND
    media_ponderata(s.matricola) >= media_ponderata
        (matricola);
    RETURN (r);
END $$

DELIMITER ;
```

Trigger 1

Supponiamo si possano rifare gli esami, scrivere un trigger che verifichi che il nuovo voto non sia più basso del precedente

Trigger 1

```
DELIMITER $$

CREATE TRIGGER trg_no_peggioramento
BEFORE UPDATE ON esami
FOR EACH ROW BEGIN
    IF NEW.voto < OLD.voto OR
       NEW.lode < OLD.lode THEN
        SIGNAL sqlstate "45000" SET message_text
            = "Peggioramento_voto_non_ammesso!";
    END IF;
END $$

DELIMITER ;
```

Trigger 2

Scrivere un trigger per tenere traccia delle assunzioni (supponendo che l'inserimento di un docente nel DB coincida con la sua data di assunzione)

Trigger 2

Scrivere un trigger per tenere traccia delle assunzioni (supponendo che l'inserimento di un docente nel DB coincida con la sua data di assunzione)

- *Hint:* prima dobbiamo creare una tabella per le assunzioni

Trigger 2

Scrivere un trigger per tenere traccia delle assunzioni (supponendo che l'inserimento di un docente nel DB coincida con la sua data di assunzione)

- *Hint:* prima dobbiamo creare una tabella per le assunzioni

```
CREATE TABLE assunzioni(  
    matricola INT(4) PRIMARY KEY,  
    data_assunzione DATE  
);
```


Trigger 2

```
DELIMITER $$

CREATE TRIGGER trg_data_assunzione
AFTER INSERT ON professori
FOR EACH ROW BEGIN
    INSERT INTO assunzioni VALUES (matricola,
        CURDATE());
END $$

DELIMITER ;
```

Trigger 3

Scrivere un trigger che, nel momento in cui viene inserito un corso scoperto (cioè senza professore), lo assegna ad un prof. che non ha corsi (non importa a quale)

Trigger 3

```
CREATE TRIGGER trg_corso_scoperto
BEFORE INSERT ON corsi
FOR EACH ROW BEGIN
    IF NEW.professore IS NULL THEN
        SELECT matricola INTO @profe
        FROM professori
        WHERE matricola NOT IN (
            SELECT DISTINCT professore
            FROM corsi
            WHERE professore IS NOT NULL
        )
        LIMIT 1;
        SET NEW.professore = @profe;
    END IF;
END $$
```

Riassunto

User Defined Functions

- 1 Scrivere una user defined function che restituisca il corso di laurea di uno studente
- 2 Scrivere una user defined function che restituisca la media ponderata di uno studente
- 3 Scrivere una user defined function che restituisca il rank di uno studente nel suo corso di laurea in base alla sua media ponderata

Trigger

- 1 Supponiamo si possano rifare gli esami, scrivere un trigger che verifichi che il nuovo voto non sia più basso del precedente
- 2 Scrivere un trigger per tenere traccia delle assunzioni (data di inserimento di un docente nel DB = data di assunzione)
- 3 Scrivere un trigger che, nel momento in cui viene inserito un corso scoperto (cioè senza professore), lo assegna ad un prof. che non ha corsi (non importa a quale)