

Seconda Esercitazione di Basi di Dati

Giorgia Nadizar

1 aprile 2022

Parte 1

Riassunto dell'esercitazione precedente

Database dell'Università (completo)

Studenti

<u>Matricola</u>	Nome	Cognome	Codice Fiscale
------------------	------	---------	----------------

- Nome, Cognome e Codice Fiscale non devono essere NULL
- Codice Fiscale dev'essere UNIQUE

Professori

<u>Matricola</u>	Nome	Cognome	Codice Fiscale	Settore
------------------	------	---------	----------------	---------

- Nome, Cognome, Codice Fiscale e Settore non devono essere NULL
- Codice Fiscale dev'essere UNIQUE

Corsi

<u>Codice</u>	Nome	CFU	Professore [Matricola]
---------------	------	-----	------------------------

- Nome e CFU non devono essere NULL

Esami

<u>Corso</u> [Codice]	<u>Studente</u> [Matricola]	Data	Voto	Lode
-----------------------	-----------------------------	------	------	------

- Voto dev'essere tra 18 e 30
- Lode non può essere TRUE se il voto non è 30

Creazione DB

```
--Creiamo il DB  
CREATE DATABASE uni_db;
```

```
--Controlliamo di averlo creato  
SHOW DATABASES;
```

```
--I prossimi comandi li eseguiamo sul DB giusto  
USE uni_db;
```

Creazione tabelle: Studenti

Studenti

<u>Matricola</u>	Nome	Cognome	Codice Fiscale
------------------	------	---------	----------------

- Nome, Cognome e Codice Fiscale non devono essere NULL
- Codice Fiscale dev'essere UNIQUE

```
CREATE TABLE studenti(  
    matricola CHAR(9) PRIMARY KEY,  
    nome VARCHAR(45) NOT NULL,  
    cognome VARCHAR(45) NOT NULL,  
    cf CHAR(16) NOT NULL UNIQUE  
);
```

Creazione tabelle: Professori

Professori

<u>Matricola</u>	Nome	Cognome	Codice Fiscale	Settore
------------------	------	---------	----------------	---------

- Nome, Cognome, Codice Fiscale e Settore non devono essere NULL
- Codice Fiscale dev'essere UNIQUE

```
CREATE TABLE professori(  
    matricola INT(4) PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(45) NOT NULL,  
    cognome VARCHAR(45) NOT NULL,  
    cf CHAR(16) NOT NULL UNIQUE,  
    settore VARCHAR(12) NOT NULL  
);
```

Creazione tabelle: Corsi

Corsi

<u>Codice</u>	Nome	CFU	Professore [Matricola]
---------------	------	-----	------------------------

- Nome e CFU non devono essere NULL

```
CREATE TABLE corsi(  
    codice CHAR(5) PRIMARY KEY,  
    nome VARCHAR(45) NOT NULL,  
    cfu TINYINT NOT NULL,  
    professore INT(4),  
    FOREIGN KEY (professore)  
        REFERENCES professori(matricola)  
        ON DELETE SET NULL  
);
```

Creazione tabelle: Esami

Esami

<u>Corso</u> [Codice]	<u>Studente</u> [Matricola]	Data	Voto	Lode
-----------------------	-----------------------------	------	------	------

- Voto dev'essere tra 18 e 30
- Lode non può essere TRUE se il voto non è 30

```
CREATE TABLE esami(  
    corso CHAR(5), studente CHAR(9), data DATE,  
    voto TINYINT NOT NULL,  
    lode BOOL DEFAULT FALSE,  
    FOREIGN KEY (corso) REFERENCES corsi(codice),  
    FOREIGN KEY (studente) REFERENCES  
        studenti(matricola) ON DELETE CASCADE,  
    CHECK (voto BETWEEN 18 AND 30),  
    CHECK ((voto<=30 AND lode=FALSE) OR (voto=30  
        AND lode=TRUE))  
);
```


Creazione tabelle: controlliamo ci sia tutto

```
SHOW TABLES ;
```

Inserimento dati

File `uni_db.sql`

Elimina il DB se lo avete già creato, così da poter scrivere tutti le stesse identiche query e ottenere gli stessi risultati, lo crea come visto insieme, e lo popola.

Esecuzione file SQL da MySQL Workbench:

- 1 Modo artigianale: aprire il file con un editor di testo o blocco note, selezionare tutto, incollare su MySQL Workbench
- 2 Modo fatto bene: File > Open SQLScript ...

In entrambi i casi, selezioniamo la porzione di script che vogliamo eseguire (nel nostro caso tutto), e usiamo il pulsante col fulmine.

Parte 2

Query sul DB

Riassunto delle query già svolte

- 1 Elencare tutte le ragazze iscritte a ingegneria.
 - ▶ *Hint 1:* la matricola di ingegneria inizia con IN
 - ▶ *Hint 2:* nel codice fiscale delle donne la data di nascita viene modificata aggiungendo 40 al giorno di nascita (es. 27 diventa 67)
 - ▶ *Disclaimer:* se in un DB ci troviamo a fare cose molto complicate per eseguire una query che sarebbe semplice con una piccola modifica dello schema, forse è il caso di modificarlo. Qui avevamo aggiunto la colonna Genere.
 - ▶ LIKE, BETWEEN, IN, SUBSTRING
- 2 Quanti studenti hanno preso una lode negli esami del prof. De Lorenzo?
 - ▶ *Hint:* se uno studente prende la lode in più di un esame con il prof. De Lorenzo quante volte devo contarlo?
 - ▶ DISTINCT, COUNT, INNER JOIN
- 3 Quali studenti hanno preso più di una lode con il prof. De Lorenzo?
 - ▶ DISTINCT, GROUP BY, HAVING, INNER JOIN

Query 4

Quali studenti non hanno mai preso una lode?

Query 4

Quali studenti non hanno mai preso una lode?

- *Hint:* quelli per cui *non esiste* un esame con la lode

Query 4

Quali studenti non hanno mai preso una lode?

- *Hint*: quelli per cui *non esiste* un esame con la lode

```
SELECT *  
FROM studenti s  
WHERE NOT EXISTS (  
    SELECT *  
    FROM esami e  
    WHERE e.lode=TRUE AND  
    e.studente=s.matricola  
);
```

Query 4 (alternativa)

- *Hint: quelli che non appartengono all'insieme dei "lodati"*

Query 4 (alternativa)

- *Hint: quelli che non appartengono all'insieme dei "lodati"*

```
SELECT *  
FROM studenti s  
WHERE s.matricola NOT IN(  
    SELECT DISTINCT studente  
    FROM esami e  
    WHERE e.lode=TRUE  
);
```

Query 5

Quali docenti svolgono un monte ore annuo minore di 120 ore?

Query 5

Quali docenti svolgono un monte ore annuo minore di 120 ore?

- *Hint:* ogni CFU corrisponde a 8 ore di lezione

Query 5

Quali docenti svolgono un monte ore annuo minore di 120 ore?

- *Hint:* ogni CFU corrisponde a 8 ore di lezione

```
SELECT p.nome, p.cognome, SUM(8*c.cfu) as
monte_ore
FROM professori p
INNER JOIN corsi c
ON p.matricola = c.professore
GROUP BY c.professore
HAVING monte_ore < 120;
```

Query 6

Qual è la media ponderata di ogni studente?

Query 6

Qual è la media ponderata di ogni studente?

- *Hint:* la media ponderata si calcola moltiplicando il voto dell'esame per il suo peso in cfu, sommando, e poi dividendo per il numero di cfu

Query 6

Qual è la media ponderata di ogni studente?

- *Hint:* la media ponderata si calcola moltiplicando il voto dell'esame per il suo peso in cfu, sommando, e poi dividendo per il numero di cfu

```
SELECT s.matricola, s.nome, s.cognome,
SUM(e.voto*c.cfu)/SUM(c.cfu) as media
FROM studenti s
INNER JOIN esami e
ON s.matricola = e.studente
    INNER JOIN corsi c
    ON e.corso = c.codice
GROUP BY e.studente;
```

Prepared Statement

Creare uno statement che mostri tutti gli studenti di un corso di laurea passato come parametro

Prepared Statement

Creare uno statement che mostri tutti gli studenti di un corso di laurea passato come parametro

- *Hint:* il corso di laurea corrisponde ai primi 4 caratteri della matricola

Prepared Statement

Creare uno statement che mostri tutti gli studenti di un corso di laurea passato come parametro

- *Hint:* il corso di laurea corrisponde ai primi 4 caratteri della matricola

```
PREPARE studenti_cdl FROM  
"SELECT_*  
FROM_students  
WHERE_matricola LIKE_CONCAT(?, '%')";
```

Prepared Statement

Creare uno statement che mostri tutti gli studenti di un corso di laurea passato come parametro

- *Hint:* il corso di laurea corrisponde ai primi 4 caratteri della matricola

```
PREPARE studenti_cdl FROM  
"SELECT_*  
FROM_*studenti  
WHERE_*matricola_*LIKE_*CONCAT(?, '%')";
```

Usiamo lo statement per trovare gli ingegneri informatici triennali:

Prepared Statement

Creare uno statement che mostri tutti gli studenti di un corso di laurea passato come parametro

- *Hint:* il corso di laurea corrisponde ai primi 4 caratteri della matricola

```
PREPARE studenti_cdl FROM  
"SELECT_*  
FROM_*studenti  
WHERE_*matricola_*LIKE_*CONCAT(?, '%')";
```

Usiamo lo statement per trovare gli ingegneri informatici triennali:

```
SET @cdl = "IN05";  
EXECUTE studenti_cdl USING @cdl;
```

Riassunto

Query

- 4 Quali studenti non hanno mai preso una lode?
- 5 Quali docenti svolgono un monte ore annuo minore di 120 ore?
- 6 Qual è la media ponderata di ogni studente?

Prepared Statement

- 1 Creare uno statement che mostri tutti gli studenti di un corso di laurea passato come parametro
- 2 Usare lo statement per trovare gli ingegneri informatici triennali