Prima Esercitazione di Basi di Dati

Lorenzo Bonin
Università degli Studi di Trieste

24 marzo 2023

Contatti e materiale

- Lorenzo Bonin, dottorando in *Applied Data Science and Artificial Intelligence* (ADSAI)
- Email: lorenzo.bonin@phd.units.it
- Materiale: `https://github.com/lorenzobonin/DB_Exercises`

Overview

1. Creazione e popolamento del DB

2. Query sul DB

Database dell'Università

Studenti

Matricola	Nome	Cognome	Codice Fiscale

Professori

Matricola	Nome	Cognome	Codice Fiscale	Settore

Corsi

Codice	Nome	CFU	Professore

Esami

Corso	Studente	Data	Voto	Lode

Database dell'Università - chiavi...

Studenti

Matricola	Nome	Cognome	Codice Fiscale

Professori

Matricola	Nome	Cognome	Codice Fiscale	Settore

Corsi

Codice	Nome	CFU	Professore [Matricola]

Esami

Corso [Codice]	Studente [Matricola]	Data	Voto	Lode

Database dell'Università - ...e vincoli

Studenti

Matricola	Nome	Cognome	Codice Fiscale

- Nome, Cognome e Codice Fiscale non devono essere `NULL`
- Codice fiscale deve essere `UNIQUE`

Professori

Matricola	Nome	Cognome	Codice Fiscale	Settore

Corsi

Codice	Nome	CFU	Professore [Matricola]

Esami

Corso [Codice]	Studente [Matricola]	Data	Voto	Lode

Database dell'Università - ...e vincoli

Studenti

Matricola	Nome	Cognome	Codice Fiscale

Professori

Matricola	Nome	Cognome	Codice Fiscale	Settore

- Nome, Cognome e Codice Fiscale e Settore non devono essere `NULL`
- Codice fiscale deve essere `UNIQUE`

Corsi

Codice	Nome	CFU	Professore [Matricola]

Esami

Corso [Codice]	Studente [Matricola]	Data	Voto	Lode

Database dell'Università - ...e vincoli

Studenti

Matricola	Nome	Cognome	Codice Fiscale

Professori

Matricola	Nome	Cognome	Codice Fiscale	Settore

Corsi

Codice	Nome	CFU	Professore [Matricola]

- Nome e CFU non devono essere `NULL`

Esami

Corso [Codice]	Studente [Matricola]	Data	Voto	Lode

Database dell'Università - ...e vincoli

Studenti

Matricola	Nome	Cognome	Codice Fiscale

Professori

Matricola	Nome	Cognome	Codice Fiscale	Settore

Corsi

Codice	Nome	CFU	Professore [Matricola]

Esami

Corso [Codice]	Studente [Matricola]	Data	Voto	Lode

- Voto deve essere compreso tra 18 e 30
- Lode non può essere `TRUE` se Voto non è 30

Creazione del DB

Creazione del DB

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

Creazione del DB

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

```
--Controlliamo che sia stato creato correttamente
SHOW DATABASES;
```

Creazione del DB

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

```
--Controlliamo che sia stato creato correttamente
SHOW DATABASES;
```

```
--I prossimi comandi faranno riferimento al DB corretto
USE uni_db;
```

Creazione tabelle: Studenti

Studenti

Matricola	Nome	Cognome	Codice Fiscale

- Nome, Cognome e Codice Fiscale non devono essere `NULL`
- Codice fiscale deve essere `UNIQUE`

Creazione tabelle: Studenti

Studenti

Matricola	Nome	Cognome	Codice Fiscale

- Nome, Cognome e Codice Fiscale non devono essere `NULL`
- Codice fiscale deve 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

Matricola	Nome	Cognome	Codice Fiscale	Settore

- Nome, Cognome e Codice Fiscale e Settore non devono essere `NULL`
- Codice fiscale deve essere `UNIQUE`

Creazione tabelle: Professori

Professori

Matricola	Nome	Cognome	Codice Fiscale	Settore

- Nome, Cognome e Codice Fiscale e Settore non devono essere `NULL`
- Codice fiscale deve 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

Codice	Nome	CFU	Professore [Matricola]

- Nome e CFU non devono essere `NULL`

Creazione tabelle: Corsi

Corsi

Codice	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

Corso [Codice]	Studente [Matricola]	Data	Voto	Lode

- Voto deve essere compreso tra 18 e 30
- Lode non può essere `TRUE` se Voto non è 30

Creazione tabelle: Esami

Esami

Corso [Codice]	Studente [Matricola]	Data	Voto	Lode

- Voto deve essere compreso tra 18 e 30
- Lode non può essere `TRUE` se 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: check finale

```
--Controlliamo che ci siano tutte le tabelle
SHOW TABLES;
```

Creazione tabelle: check finale

```
--Controlliamo che ci siano tutte le tabelle
SHOW TABLES;
```

Inserimento dati

N.B:

Quando alcuni vincoli possono essere violati in fase di inserimento dei dati, è opportuno disattivarli con:

```
SET nomeVincolo = 0
```

ricordandosi poi di riattivarli con:

```
SET nomeVincolo = 1
```

```
--Disattivare i vincoli di integrita' referenziale
SET foreign_key_checks = 0

--Attivare i vincoli di integrita' referenziale
SET foreign_key_checks = 1
```

Inserimento dati - Studenti

Matricola	**Nome**	**Cognome**	**Codice Fiscale**
IN0500550	Lorenzo	Bonin	BNNLNZ98E25L424R
SM3500487	Irene	Ferfoglia	FRFRNI98L50L424C

Inserimento dati - Studenti

Matricola	Nome	Cognome	Codice Fiscale
IN0500550	Lorenzo	Bonin	BNNLNZ98E25L424R
SM3500487	Irene	Ferfoglia	FRFRNI98L50L424C

```
INSERT INTO studenti
    VALUES ("IN0500550", "Lorenzo", "Bonin",
    "BNNLNZ98E25L424R"),
    ("SM3500487", "Irene", "Ferfoglia",
    "FRFRNI98L50L424C");
```

Inserimento dati - Professori

Matricola	Nome	Cognome	Codice Fiscale	Settore
0001	Andrea	De Lorenzo	DLRNDR84B11G642N	ING-INF/05
0002	Eric	Medvet	MDVRCE79C02L424U	ING-INF/05

Inserimento dati - Professori

Matricola	Nome	Cognome	Codice Fiscale	Settore
0001	Andrea	De Lorenzo	DLRNDR84B11G642N	ING-INF/05
0002	Eric	Medvet	MDVRCE79C02L424U	ING-INF/05

```
INSERT INTO professori (nome, cognome, cf, settore)
    VALUES ("Andrea", "De Lorenzo",
    "DLRNDR84B11G642N", "ING-INF/05"),
    ("Eric", "Medvet", "MDVRCE79C02L424U",
    "ING-INF/05");
```

Inserimento dati - Corsi

Codice	**Nome**	**CFU**	**Professore**
079IN	Basi di Dati	9	001
587SM	Advanced Programming	12	

Inserimento dati - Corsi

Codice	Nome	CFU	Professore
079IN	Basi di Dati	9	001
587SM	Advanced Programming	12	

```
INSERT INTO corsi
    VALUES ("079IN", "Basi di Dati", 9, 0001);
INSERT INTO corsi(codice, nome, cfu)
    VALUES ("587SM", "Advanced Programming", 12);
```

Inserimento dati - Esami

Corso	Studente	Data	Voto	Lode
079IN	IN0500550	10.06.2020	30	TRUE
587SM	SM3500487	24.06.2021	30	

Inserimento dati - Esami

Corso	Studente	Data	Voto	Lode
079IN	IN0500550	10.06.2020	30	TRUE
587SM	SM3500487	24.06.2021	30	

```
INSERT INTO esami
    VALUES("079IN", "IN0500550", "2020-06-10",
30, TRUE);
INSERT INTO esami(corso, studente, data, voto)
    VALUES ("587SM", "SM3500487", "2021-06-24",
    30);
```

Inserimento dati - Test sui vincoli

- Una matricola doppia tra i professori?

```
INSERT INTO professori
    VALUES ("0001", "Luca", "Manzoni",
    "MNZLCU84A12E462M", "INF/01");
```

Inserimento dati - Test sui vincoli

- Una matricola doppia tra i professori?

```
INSERT INTO professori
    VALUES ("0001", "Luca", "Manzoni",
    "MNZLCU84A12E462M", "INF/01");
```

- Un esame di un corso inesistente?

```
INSERT INTO esami
    VALUES ("111AA", "IN0500550", "2019-06-03",
    25, NULL);
```

Inserimento dati - Test sui vincoli

- Una matricola doppia tra i professori?

```
INSERT INTO professori
    VALUES ("0001", "Luca", "Manzoni",
    "MNZLCU84A12E462M", "INF/01");
```

- Un esame di un corso inesistente?

```
INSERT INTO esami
    VALUES ("111AA", "IN0500550", "2019-06-03",
    25, NULL);
```

- Un 18 e lode?

```
INSERT INTO esami
    VALUES ("587SM", "IN0500550", "2019-06-03",
    18, TRUE);
```

Inserimento dati - Popolamento massiccio

Creiamo un DB più interessante...

File uni_db.sql

- Elimina il DB se esiste già
- Ne crea uno nuovo secondo quanto visto nelle slides precedenti
- Lo popola con altri dati

Esecuzione di un file SQL da MySQL Workbench

- File $\rightarrow$ Open SQLScript $\rightarrow$ ...
- Selezionare la porzione di codice da eseguire (in questo caso tutto)
- Selezionare il pulsante con il fulmine

Per chi non ha MySQL e MySQL Workbench...

Esiste questo editor online: https://www.jdoodle.com/execute-sql-online/. Svantaggi:

- potrebbe non supportare alcune operazioni
- è volatile, ovvero non ricorda quanto eseguito in precedenza (è necessario creare, popolare ed interrogare in un'unica esecuzione!)

Morale: può andare bene come soluzione temporanea.

Query 1

Task

Elencare tutte le ragazze iscritte ad ingegneria.

Query 1

Task

Elencare tutte le ragazze iscritte ad ingegneria.

- *Hint 1*: la matricola degli iscritti ad ingegneria inizia con "IN"

Query 1

Task

Elencare tutte le ragazze iscritte ad ingegneria.

- *Hint 1*: la matricola degli iscritti ad ingegneria inizia con "IN"
- *Hint 2*: nel codice fiscale delle donne la data di nascita viene modificata aggiungendo 40 al giorno di nascita (es: 16 diventa 56)

Query 1

Task

Elencare tutte le ragazze iscritte ad ingegneria.

- *Hint 1*: la matricola degli iscritti ad ingegneria inizia con "IN"
- *Hint 2*: nel codice fiscale delle donne la data di nascita viene modificata aggiungendo 40 al giorno di nascita (es: 16 diventa 56)

```
SELECT *
FROM studenti
WHERE matricola LIKE "IN%" AND
(cf LIKE "_________4%" OR cf LIKE "_________5%"
OR cf LIKE "_________6%" OR cf LIKE "_________7%");
```

Query 1 - soluzione alternativa

Usare la funzione `SUBSTRING(str, pos, len)`

Query 1 - soluzione alternativa

Usare la funzione `SUBSTRING(str, pos, len)`

```
SELECT *
FROM studenti
WHERE matricola LIKE "IN%" AND
SUBSTRING(cf, 10, 1) IN ("4","5","6","7");
```

Query 1 - soluzione alternativa

Usare la funzione `SUBSTRING(str, pos, len)`

```
SELECT *
FROM studenti
WHERE matricola LIKE "IN%" AND
SUBSTRING(cf, 10, 1) IN ("4","5","6","7");
```

Oppure...

```
SELECT *
FROM studenti
WHERE matricola LIKE "IN%" AND
SUBSTRING(cf, 10, 1) BETWEEN "4" AND "7";
```

Query 1 - un po' troppo fantasiosa?

Se in un DB ci troviamo a scrivere queries troppo complicate per accedere ad informazioni di interesse che sarebbero facilmente accessibili con una semplice modifica, forse è il caso di modificarlo...

Query 1 - un po' troppo fantasiosa?

Se in un DB ci troviamo a scrivere queries troppo complicate per accedere ad informazioni di interesse che sarebbero facilmente accessibili con una semplice modifica, forse è il caso di modificarlo...

$\rightarrow$ Aggiungiamo una colonna per il genere e popoliamola opportunamente!

Hint: andrà fatto un update di righe senza specificare alcuna chiave nella clausola `WHERE`... Va disabilitata temporaneamente la *safe mode*.

Query 1 - update della tabella studenti

```
ALTER TABLE studenti
ADD COLUMN genere CHAR(1) NOT NULL ;
```

Query 1 - update della tabella studenti

```
ALTER TABLE studenti
ADD COLUMN genere CHAR(1) NOT NULL ;
```

```
SET SQL_SAFE_UPDATES=0;
UPDATE studenti SET genere="M";
UPDATE studenti SET genere="F"
WHERE SUBSTRING (cf, 10, 1) BETWEEN "4" AND "7";
SET SQL_SAFE_UPDATES=1;
```

Query 1 - update della tabella studenti

```
ALTER TABLE studenti
ADD COLUMN genere CHAR(1) NOT NULL ;
```

```
SET SQL_SAFE_UPDATES=0;
UPDATE studenti SET genere="M";
UPDATE studenti SET genere="F"
WHERE SUBSTRING (cf, 10, 1) BETWEEN "4" AND "7";
SET SQL_SAFE_UPDATES=1;
```

```
ALTER TABLE studenti
ADD CHECK (genere IN ("M", "F"));
```

Query 1 - sul DB modificato

Task

Elencare tutte le ragazze iscritte ad ingegneria.

Query 1 - sul DB modificato

Task

Elencare tutte le ragazze iscritte ad ingegneria.

```
SELECT *
FROM studenti
WHERE matricola LIKE "IN%" AND genere="F";
```

Mooooolto meno fantasiosa!

Query 2

Task

Quanti studenti hanno preso una lode negli esami del prof. De Lorenzo?

Query 2

Task

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?

Query 2

Task

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?

```
SELECT COUNT (DISTINCT e.studente) as n_lodati
FROM esami e INNER JOIN corsi c
ON e.corso = c.codice
    INNER JOIN professori p
    ON c.professore = p.matricola
WHERE e.lode = TRUE AND p.cognome ="De Lorenzo";
```

Query 3

Task

Quali studenti hanno preso più di una lode con il prof. De Lorenzo?

Query 3

Task

Quali studenti hanno preso più di una lode con il prof. De Lorenzo?

```
SELECT e.studente , COUNT (e.lode) as n_lodi
FROM esami e INNER JOIN corsi c
ON e.corso = c.codice
    INNER JOIN professori p
    ON c.professore = p.matricola
WHERE e.lode = TRUE AND p.cognome ="De Lorenzo"
GROUP BY e.studente
HAVING n_lodi >=2;
```