Prima Esercitazione di Basi di Dati

Giorgia Nadizar

24 marzo 2022

Prima di iniziare...

- Giorgia Nadizar, tutor di DB (e dottoranda in AI)
- `giorgia.nadizar@gmail.com` oppure `giorgia.nadizar@phd.units.it`
- `https://github.com/giorgia-nadizar/DbExercises`
- `https://b.socrative.com/login/student/` $\rightarrow$ DBES1

Parte 1

Creazione e popolamento 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

Studenti

Matricola	Nome	Cognome	Codice Fiscale

Professori

Matricola	Nome	Cognome	Codice Fiscale	Settore

Corsi

Codice	Nome	CFU	Professore

Esami

Corso	Studente	Data	Voto	Lode

Non manca niente?

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

Non manca niente? **Chiavi** e **vincoli**!

Database dell'Università (completo)

Studenti

Matricola	Nome	Cognome	Codice Fiscale

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

Professori

Matricola	Nome	Cognome	Codice Fiscale	Settore

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

Corsi

Codice	Nome	CFU	Professore [Matricola]

- Nome e CFU non devono essere `NULL`

Esami

Corso [Codice]	Studente [Matricola]	Data	Voto	Lode

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

Creazione DB

Creazione DB

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

Creazione DB

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

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

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

Matricola	Nome	Cognome	Codice Fiscale

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

Creazione tabelle: Studenti

Studenti

Matricola	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

Matricola	Nome	Cognome	Codice Fiscale	Settore

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

Creazione tabelle: Professori

Professori

Matricola	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

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

Creazione tabelle: Esami

Esami

Corso [Codice]	Studente [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

Nel caso in cui alcuni vincoli siano complessi da rispettare in fase di inserimento, ricordiamoci che possiamo disattivarli temporaneamente con

```
SET nomeVincolo = 0
```

ricordandosi poi di riattivarli con

```
SET nomeVincolo = 1
```

Inserimento dati

Nel caso in cui alcuni vincoli siano complessi da rispettare in fase di inserimento, ricordiamoci che possiamo disattivarli temporaneamente con

```
SET nomeVincolo = 0
```

ricordandosi poi di riattivarli con

```
SET nomeVincolo = 1
```

Questo vale per i vincoli interrelazionali, per cui `nomeVincolo` diventa `foreign_key_checks`, o per altri tipi di vincoli.

Inserimento dati

Nel caso in cui alcuni vincoli siano complessi da rispettare in fase di inserimento, ricordiamoci che possiamo disattivarli temporaneamente con

```
SET nomeVincolo = 0
```

ricordandosi poi di riattivarli con

```
SET nomeVincolo = 1
```

Questo vale per i vincoli interrelazionali, per cui `nomeVincolo` diventa `foreign_key_checks`, o per altri tipi di vincoli.

Ovviamente in questo caso i vincoli devono avere un nome per poterli "chiamare". Un qualsiasi vincolo, per esempio un `CHECK` diventa:

```
CONSTRAINT nomeVincolo CHECK ...
```

Inserimento dati: `studenti`

Matricola	**Nome**	**Cognome**	**Codice Fiscale**
IN0500308	Giorgia	Nadizar	NZDGRG97M67L424K
SM3211162	Leonardo	Bianchi	BNCLRD99A12L424Y

Inserimento dati: studenti

Matricola	**Nome**	**Cognome**	**Codice Fiscale**
IN0500308	Giorgia	Nadizar	NZDGRG97M67L424K
SM3211162	Leonardo	Bianchi	BNCLRD99A12L424Y

```
INSERT INTO studenti
    VALUES("IN0500308", "Giorgia", "Nadizar",
    "NZDGRG97M67L424K"),
    ("SM3211162", "Leonardo", "Bianchi",
    "BNCLRD99A12L424Y");
```

Inserimento dati: `professori`

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

Inserimento dati: professori

Matr.	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	0001
511SM	Analisi Matematica 3	12	

Inserimento dati: corsi

Codice	**Nome**	**CFU**	**Professore**
079IN	Basi di Dati	9	0001
511SM	Analisi Matematica 3	12	

```
INSERT INTO corsi
    VALUES("079IN", "Basi di Dati", 9, 0001);

INSERT INTO corsi(codice,nome,cfu)
    VALUES("511SM", "Analisi Matematica 3", 12);
```

Inserimento dati: `esami`

Corso	**Studente**	**Data**	**Voto**	**Lode**
079IN	IN0500308	03.06.2019	30	TRUE
079IN	SM3211162	23.06.2019	28	

Inserimento dati: esami

Corso	**Studente**	**Data**	**Voto**	**Lode**
079IN	IN0500308	03.06.2019	30	TRUE
079IN	SM3211162	23.06.2019	28	

```
INSERT INTO esami
    VALUES("079IN", "IN0500308", "2019-06-03",
        30, TRUE);

INSERT INTO esami(corso,studente,data,voto)
    VALUES("079IN", "SM3211162", "2019-06-23",
        28);
```

Inserimento dati: testiamo i vincoli

- Una matricola doppia?

Inserimento dati: testiamo i vincoli

- Una matricola doppia?

```
INSERT INTO studenti
VALUES("IN0500308", "Mario", "Rossi",
"NZDGRG97M67L424K");
```

Inserimento dati: testiamo i vincoli

- Una matricola doppia?

```
INSERT INTO studenti
VALUES("IN0500308", "Mario", "Rossi",
"NZDGRG97M67L424K");
```

- Un esame di un corso che non esiste?

Inserimento dati: testiamo i vincoli

- Una matricola doppia?

```
INSERT INTO studenti
VALUES("IN0500308", "Mario", "Rossi",
"NZDGRG97M67L424K");
```

- Un esame di un corso che non esiste?

```
INSERT INTO esami
VALUES("111AA", "IN0500308", "2019-06-03",
25, NULL);
```

Inserimento dati: testiamo i vincoli

- Una matricola doppia?

```
INSERT INTO studenti
VALUES("IN0500308", "Mario", "Rossi",
"NZDGRG97M67L424K");
```

- Un esame di un corso che non esiste?

```
INSERT INTO esami
VALUES("111AA", "IN0500308", "2019-06-03",
25, NULL);
```

- Un 18 e lode?

Inserimento dati: testiamo i vincoli

- Una matricola doppia?

```
INSERT INTO studenti
VALUES("IN0500308", "Mario", "Rossi",
"NZDGRG97M67L424K");
```

- Un esame di un corso che non esiste?

```
INSERT INTO esami
VALUES("111AA", "IN0500308", "2019-06-03",
25, NULL);
```

- Un 18 e lode?

```
INSERT INTO esami
VALUES("511SM", "IN0500308", "2019-06-03",
18, TRUE);
```

Inserimento dati: ora facciamo sul serio

Un database con 2 righe in ciascuna tabella è poco interessante: è il caso di popolarlo seriamente.

Inserimento dati: ora facciamo sul serio

Un database con 2 righe in ciascuna tabella è poco interessante: è il caso di popolarlo seriamente.

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

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

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 ...

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.

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.

Per chi non ha MySQL (e MySQL Workbench), potete provare con questo editor online: `https://www.jdoodle.com/execute-sql-online/`.
Nota 1: alcune operazioni potrebbero non essere supportate.
Nota 2: non è un vero DB, ma è volatile, cioè dimentica tutto quello che ha eseguito in precedenza. Ergo dovete creare, popolare ed interrogare il DB in un'unica esecuzione (può andare bene per semplici esercizi come soluzione temporanea).

Parte 2

Query sul DB

Query 1

Elencare tutte le ragazze iscritte a ingegneria.

Query 1

Elencare tutte le ragazze iscritte a ingegneria.

- *Hint 1:* la matricola di ingegneria inizia con IN

Query 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)

Query 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)

```
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 (alternativa)

Usiamo la funzione `SUBSTRING(str, pos, len)`.

Query 1 (alternativa)

Usiamo 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 (alternativa)

Usiamo la funzione `SUBSTRING(str, pos, len)`.

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

Oppure ancora

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

Query 1: precisazione

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.

Query 1: precisazione

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 sarebbe opportuno avere una colonna con il genere: aggiungiamola e popoliamola opportunamente.

Query 1: precisazione

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 sarebbe opportuno avere una colonna con il genere: aggiungiamola e popoliamola opportunamente.

Hint: qui faremo un update di righe senza specificare una chiave nella clausola `WHERE`, ricordiamoci di disabilitare temporaneamente la safe mode.

Query 1: update della tabella

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

Query 1: update della tabella

```
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

```
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 aggiornato)

Elencare tutte le ragazze iscritte a ingegneria.

Query 1 (sul DB aggiornato)

Elencare tutte le ragazze iscritte a ingegneria.

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

Query 1 (sul DB aggiornato)

Elencare tutte le ragazze iscritte a ingegneria.

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

Mooolto più semplice ed elegante!

Query 2

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

Query 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?

Query 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?

```
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

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

Query 3

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;
```