



**UNIVERSITÀ DEGLI STUDI  
DI TRIESTE**



ARTIFICIAL INTELLIGENCE  
& DATA ANALYTICS



**dipartimento  
di ingegneria  
e architettura**

# Prima esercitazione di Basi di Dati

---

Università degli studi di Trieste

Giovanni Pinna

12 aprile 2024

# Contatti



GIOVANNI.PINNA@phd.units.it



Edificio: C3, Ufficio: C3.232  
(contattatemi prima via mail)



[https://github.com/giovannipinna96/DB\\_2024\\_Exercises](https://github.com/giovannipinna96/DB_2024_Exercises)



SCAN ME

# Sommario

1. Creazione e popolamento del DB
2. Query sul DB

# Database (molto semplice) 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 (molto semplice) 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

Mancano **CHIAVI** e **VINCOLI** !



# Database (molto semplice) dell'Università

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

- La Matricola sarà la chiave primaria.
- Nome, Cognome e Codice Fiscale non devono essere NULL.
- Codice Fiscale deve essere anche UNIQUE.

# Database (molto semplice) dell'Università

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

- La matricola sarà la chiave primaria. Vogliamo che sia un numero incrementale.
- Nome, Cognome e Codice Fiscale e Settore non devono essere NULL.
- Codice Fiscale deve essere anche UNIQUE.

# Database (molto semplice) dell'Università

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

- Il codice sarà la chiave primaria della tabella.
- Nome e CFU non devono essere NULL.
- La colonna Professore della tabella corsi farà riferimento alla chiave matricola della tabella Professori.

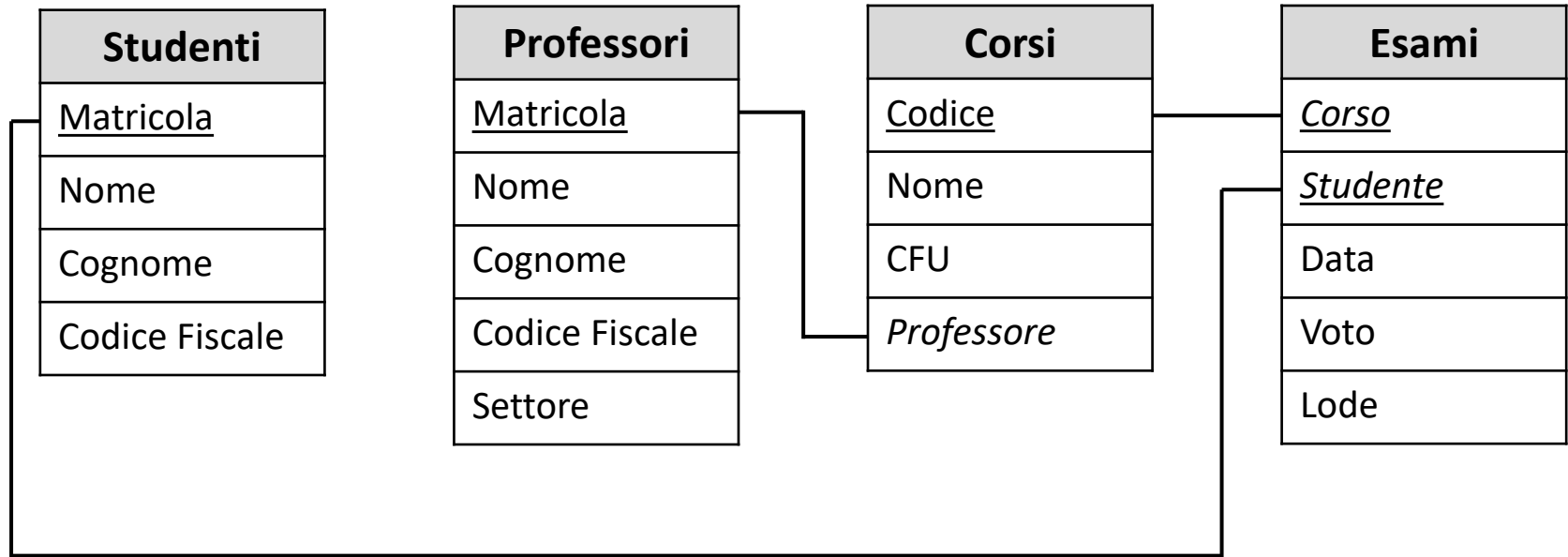


# Database (molto semplice) dell'Università

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

- Il Voto deve essere tra 18 e 30
- La Lode non può essere uguale a True se il Voto non è uguale a 30
- La colonna corso fa riferimento alla chiave Codice della tabella Corsi
- La colonna Studente fa riferimento alla chiave Matricola della tabella Studenti

# Database (molto semplice) dell'Università



# Per scrivere codice SQL

Per chi non ha MySQL o MySQL Workbench...

Esistono questi editor online:

- <https://onecompiler.com/studio>
- <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.

# Creazione del DB

```
1  --Creare il DB
2  CREATE DATABASE uni_db;
```

```
1  --Controllare di averlo creato
2  SHOW DATABASES;
```

```
1  --Selezionare il database
2  USE uni_db;
```

# Creazione tabella Studenti

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

- Nome, Cognome e Codice Fiscale non devono essere NULL.
- Codice Fiscale deve essere anche UNIQUE

```
1  --Creare la tabella studente
2  CREATE TABLE
3      studenti (
4          matricola CHAR(9) PRIMARY KEY,
5          nome VARCHAR(45) NOT NULL,
6          cognome VARCHAR(45) NOT NULL,
7          cf CHAR(16) NOT NULL UNIQUE
8      );
```

# Creazione tabella Professori

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

- Nome, Cognome e Codice Fiscale e Settore non devono essere NULL.
- Codice Fiscale deve essere anche UNIQUE
- Vogliamo che la matricola sia auto incrementale

# Creazione tabella Professori

```
1  --Creare la tabella professore
2  CREATE TABLE
3      professori (
4          matricola
5          nome
6          cognome
7          cf
8          settore
9      );
```

# Creazione tabella Professori

```
1  --Creare la tabella professore
2  CREATE TABLE
3      professori (
4          matricola INT(4) PRIMARY KEY AUTO_INCREMENT,
5          nome
6          cognome
7          cf
8          settore
9      );
```



# Creazione tabella Professori

```
1  --Creare la tabella professore
2  CREATE TABLE
3      professori (
4          matricola INT(4) PRIMARY KEY AUTO_INCREMENT,
5          nome VARCHAR(45) NOT NULL,
6          cognome VARCHAR(45) NOT NULL,
7          cf
8          settore
9      );
```

# Creazione tabella Professori

```
1  --Creare la tabella professore
2  CREATE TABLE
3      professori (
4          matricola INT(4) PRIMARY KEY AUTO_INCREMENT,
5          nome VARCHAR(45) NOT NULL,
6          cognome VARCHAR(45) NOT NULL,
7          cf CHAR(16) NOT NULL UNIQUE,
8          settore
9      );
```

# Creazione tabella Professori

```
1  --Creare la tabella professore
2  CREATE TABLE
3      professori (
4          matricola INT(4) PRIMARY KEY AUTO_INCREMENT,
5          nome VARCHAR(45) NOT NULL,
6          cognome VARCHAR(45) NOT NULL,
7          cf CHAR(16) NOT NULL UNIQUE,
8          settore VARCHAR(12) NOT NULL
9      );
```

# Creazione tabella Corsi

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

- Nome e CFU non devono essere NULL
- La colonna Professore della tabella corsi farà riferimento alla chiave matricola della tabella Professori

```
1  --Creare la tabella corso
2  CREATE TABLE
3      corsi (
4          codice CHAR(5) PRIMARY KEY,
5          nome VARCHAR(45) NOT NULL,
6          cfu TINYINT NOT NULL,
7          professore INT(4),
8          FOREIGN KEY (professore)
9              REFERENCES professori (matricola) ON DELETE SET NULL
10     );
```

# Creazione tabella Esami

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

- Il Voto deve essere tra 18 e 30
- La Lode non può essere uguale a True se il Voto non è uguale a 30

# Creazione tabella Esami

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

- Il Voto deve essere tra 18 e 30
- La Lode non può essere uguale a True se il Voto non è uguale a 30
- Data sarà di tipo DATE
- Voto sarà di tipo TINYINT
- Lode sarà un BOOL valore di DEFAULT sarà FALSE
- Per i controlli usate i CHECK

# Creazione tabella Esami

Esami
<u>Corso</u> [Codice]
<u>Studiante</u> [Matricola]
Data
Voto
Lode

```
1  --Creare la tabella esame
2  CREATE TABLE
3      esami (
4          corso CHAR(5),
5          studente CHAR(9),
6          data DATE,
7          voto TINYINT NOT NULL,
8          lode BOOL DEFAULT FALSE,
9          FOREIGN KEY (corso) REFERENCES corsi (codice),
10         FOREIGN KEY (studente) REFERENCES studenti (matricola)
11             ON DELETE CASCADE,
12         CHECK (voto BETWEEN 18 AND 30),
13         CHECK ((voto <= 30 AND lode = FALSE)
14             OR (voto = 30 AND lode = TRUE))
15     );
```

# Creazione di tabelle

Controlliamo di aver creato tutte le tabelle.

```
1  --Mostrare le tabelle create  
2  SHOW TABLES;
```



# Inserimento dati

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

```
1  --disattivare un vincolo  
2  SET nomeVincolovoto = 0
```

```
1  --riattivare un vincolo  
2  SET nomeVincolo = 1
```

# Inserimento dati

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:

```
1  --per un vincolo check
2  CONSTRAINT nomeVincolo CHECK ...
```

# Inserimento dati Studenti

<u>Matricola</u>	Nome	Cognome	Codice Fiscale
DR1400034	Giovanni	Pinna	PNNGNN96L17123X
SM3211162	Leonardo	Bianchi	BNCLRD99A12L424Y

# Inserimento dati Studenti

<u>Matricola</u>	Nome	Cognome	Codice Fiscale
DR1400034	Giovanni	Pinna	PNNGNN96L17123X
SM3211162	Leonardo	Bianchi	BNCLRD99A12L424Y

```
1  --Inserire degli studenti
2  INSERT INTO
3      studenti
4  VALUES
5      ("DR1400034", "Giovanni", "Pinna", "PNNGNN96L17123X"),
6      ("SM3211162", "Leonardo", "Bianchi", "BNCLRD99A12L424Y");
```

# Inserimento dati Professori

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

# Inserimento dati Professori

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

```
1  --Inserire dei professori
2  INSERT INTO
3      professori (nome, cognome, cf, settore)
4  VALUES
5      ("Andrea", "De Lorenzo", "DLRNDR84B11G642N", "ING-INF/05"),
6      ("Eric", "Medvet", "MDVRCE79C02L424U", "ING-INF/05");
```

# Inserimento dati Corsi

<u>Codice</u>	Nome	CFU	Professore
079IN	Basi di dati	9	0001
511SM	Natural language processing	6	

# Inserimento dati Corsi

<u>Codice</u>	Nome	CFU	Professore
079IN	Basi di dati	9	0001
511SM	Natural language processing	6	

```
1  --Inseriamo dei corsi
2  INSERT INTO
3      corsi
4  VALUES
5      ("079IN", "Basi di Dati", 9, 0001);
6
7  INSERT INTO
8      corsi (codice, nome, cfu)
9  VALUES
10     ("511SM", "Natural language processing", 6);
```



# Inserimento dati Esami

<u>Corso</u>	<u>Studente</u>	Data	Voto	Lode
079IN	DR1400034	02.02.2023	30	TRUE
079IN	SM3211162	13.07.2023	28	FALSE

# Inserimento dati Esami

<u>Corso</u>	<u>Studente</u>	Data	Voto	Lode
079IN	DR1400034	02.02.2023	30	TRUE
079IN	SM3211162	13.07.2023	28	FALSE

```
1  --Inseriamo degli esami
2  INSERT INTO
3      esami
4  VALUES
5      ("079IN", "DR1400034", "2023-06-03", 30, TRUE);
6
7  INSERT INTO
8      esami (corso, studente, data, voto)
9  VALUES
10     ("079IN", "SM3211162", "2021-06-23", 28);
```

# Inserimento dati Testiamo i vincoli

Proviamo ad inserire un codice fiscale doppia

# Inserimento dati Testiamo i vincoli

Proviamo ad inserire un codice fiscale doppia

```
1  --Testiamo i vincoli
2  --Codice fiscale già esistente
3  INSERT INTO
4      studenti
5  VALUES
6      ("IN0500308", "Mario", "Rossi", "PNNGNN96L17123X");
```

# Inserimento dati Testiamo i vincoli

Proviamo ad inserire un esame di un corso che non esiste

# Inserimento dati Testiamo i vincoli

Proviamo ad inserire un esame di un corso che non esiste

```
1  --Testiamo i vincoli
2  --Un esame di un corso che non esiste
3  INSERT INTO
4      esami
5  VALUES
6      ("111AA", "IN0500308", "2019-06-03", 25, NULL);
```

# Inserimento dati Testiamo i vincoli

Proviamo ad inserire un 18 con la lode

# Inserimento dati Testiamo i vincoli

Proviamo ad inserire un 18 con la lode

```
1  --Testiamo i vincoli
2  --Un 18 e lode
3  INSERT INTO
4      esami
5  VALUES
6      ("511SM", "IN0500308", "2019-06-03", 18, TRUE);
```



# Inserimento dati massiccio

Adesso popoliamo il database seriamente! (*file: popolamento.sql*)

Ci sono due metodi:

1. aprire il file con un editor di testo o blocco note, selezionare tutto, incollare su MySQL Workbench (POCO ELEGANTE)
2. File -> Open SQLScript -> ...

# Inserimento dati massiccio

Per chi non ha MySQL o MySQL Workbench...

Esistono questi editor online:

- <https://onecompiler.com/studio>
- <https://www.jdoodle.com/execute-sql-online/>
- ...

# Query 1

Elencare tutte gli studenti iscritti ad ingegneria

# Query 1

Elencare tutte gli studenti iscritti ad ingegneria

*Suggerimento:* la matricola degli iscritti a ingegneria inizia con «IN»

# Query 1

Elencare tutte gli studenti iscritti ad ingegneria

*Suggerimento:* la matricola degli iscritti a ingegneria inizia con «IN»

```
1  --Query 1: elencare tutti gli studenti iscritti ad ingegneria
2  SELECT *
3  FROM studenti
4  WHERE matricola LIKE "IN%";
```

## Query 2

Elencare tutte le studentesse iscritte ad ingegneria

*Suggerimento 1:* la matricola degli iscritti a ingegneria inizia con «IN»

## Query 2

Elencare tutte le studentesse iscritte ad ingegneria

*Suggerimento 1:* la matricola degli iscritti a ingegneria inizia con «IN»

*Suggerimento 2:* nel codice fiscale delle donne la data di nascita viene modificata aggiungendo 40 al giorno di nascita (es.  $16 + 40 = 56$ )

## Query 2

Elencare tutte le studentesse iscritte ad ingegneria

*Suggerimento 1:* la matricola degli iscritti a ingegneria inizia con «IN»

*Suggerimento 2:* nel codice fiscale delle donne la data di nascita viene modificata aggiungendo 40 al giorno di nascita (es. 16 + 40 = 56)

*Suggerimento 3:* come è fatto un codice fiscale

- sei caratteri alfabetici per il cognome e nome;
- due caratteri numerici per l'anno di nascita;
- un carattere alfabetico per il mese di nascita;
- **due caratteri numerici per il giorno di nascita ed il sesso;**
- quattro caratteri, di cui uno alfabetico e tre numerici per il luogo di nascita;
- Il sedicesimo carattere, ha funzione di controllo.



## Query 2

Elencare tutte le studentesse iscritte ad ingegneria

*Suggerimento 1:* la matricola degli iscritti a ingegneria inizia con «IN»

*Suggerimento 2:* nel codice fiscale delle donne la data di nascita viene modificata aggiungendo 40 al giorno di nascita (es. 16 + 40 = 56)

```
1  -- Query 2: elencare tutte le ragazze iscritte a ingegneria.
2  SELECT *
3  FROM studenti
4  WHERE matricola LIKE "IN%"
5         AND (
6             cf LIKE "_____4%"
7             OR cf LIKE "_____5%"
8             OR cf LIKE "_____6%"
9             OR cf LIKE "_____7%"
10         );
```

## Query 2: alternative

```
1  --Query 2 alternativa 1
2  SELECT *
3  FROM studenti
4  WHERE matricola LIKE "IN%"
5         AND SUBSTRING(cf, 10, 1) IN ("4", "5", "6", "7");
```

```
1  --Query 2 alternativa 2
2  SELECT *
3  FROM studenti
4  WHERE matricola LIKE "IN%"
5         AND SUBSTRING(cf, 10, 1) BETWEEN "4" AND "7";
```

## Query 2 modifica della tabella

Se in un DB è necessario scrivere delle query molto complicate per accedere a informazioni di interesse che sarebbero facilmente accessibili con una semplice modifica alla tabella, è il caso di modificare la tabella in questione.

➔ Aggiungiamo una colonna per il genere e popoliamola

## Query 2 modifica della tabella

Se in un DB è necessario scrivere delle query molto complicate per accedere a informazioni di interesse che sarebbero facilmente accessibili con una semplice modifica alla tabella, è il caso di modificare la tabella in questione.

➔ Aggiungiamo una colonna per il genere e popoliamola

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

## Query 2 modifica della tabella

```
1  --Update della tabella
2  ALTER TABLE studenti
3  ADD COLUMN genere CHAR(1) NOT NULL;
```

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

## Query 2 modifica della tabella

```
1  --Update tabella aggiunta vincolo  
2  ALTER TABLE studenti ADD CHECK (genere IN ("M", "F"));
```

Rifacciamo sul nuovo database la Query 2:

➔ Elencare tutte le studentesse iscritte ad ingegneria

## Query 2 dopo la modifica della tabella

```
1  --Query 2: dopo la modifica del DB
2  SELECT *
3  FROM studenti
4  WHERE matricola LIKE "IN%" AND genere = "F";
```

## Query 3

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



## Query 3

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

*Suggerimento:* Se uno studente prende la lode in più di un esame con il professor. De Lorenzo quante volte lo devo contare?

## Query 3

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

*Suggerimento:* Se uno studente prende la lode in più di un esame con il professor. De Lorenzo quante volte lo devo contare?

**Va contato una sola volta**

# Query 3

```
1  -- Query 3: quanti studenti hanno preso una lode negli esami del prof. De Lorenzo?
2  SELECT COUNT(DISTINCT e.studente) as n_lodati
3  FROM esami e
4       INNER JOIN corsi c ON e.corso = c.codice
5       INNER JOIN professori p ON c.professore = p.matricola
6  WHERE e.lode = TRUE
7       AND p.cognome = "De Lorenzo";
```

## Query 4

Quali studenti hanno preso **più di una** lode (e quante lodi) negli esami del professor. De Lorenzo?

# Query 4

```
1  -- Query 4: quali studenti hanno preso più di una lode con il prof. De Lorenzo?
2  SELECT e.studente, COUNT(e.lode) as n_lodi
3  FROM esami e
4       INNER JOIN corsi c ON e.corso = c.codice
5       INNER JOIN professori p ON c.professore = p.matricola
6  WHERE e.lode = TRUE AND p.cognome = "De Lorenzo"
7  GROUP BY e.studente
8  HAVING n_lodi >= 2;
```

## Query 5

Calcolare la media ponderata per lo studente IN0500273.

$$\textit{Media ponderata} = \frac{\sum_{i=1}^{\#esami} voto_i * CFU_i}{\sum_{i=1}^{\#esami} CFU_i}$$

# Query 5

```
1  --Query 5: media ponderata per uno specifico studente
2  SELECT
3      e.studente,
4      SUM(e.voto * c.cfu) / SUM(c.cfu) AS media_ponderata
5  FROM esami e
6      INNER JOIN corsi c ON e.corso = c.codice
7  WHERE e.studente = 'IN0500273'
8  GROUP BY e.studente;
```

# Ulteriori query per esercitarvi

- Query 6: trovare lo studente più performante in ogni corso
- Query 7: calcolare la media dei voti per ogni corso
- Query 8: calcolare il totale di cfu ottenuti da ogni studente
- Query 9: lista degli studenti che hanno fatto un esame in ogni settore





---

# Grazie !

---