

# Inserimento e modifica dei dati in MySQL

---

LABORATORIO DI BASI DI DATI  
A.A. 2019/2020

Dott. Marco Savi

Contenuti riadattati a partire da slide gentilmente concesse  
dai **Dott. Paolo Napoletano** e **Claudio Venturini**

# Riepilogo: SQL DDL, DML, DCL

## DDL – Data Definition Language

- Definizione e modifica dello schema del DB (db, tabelle, colonne, viste, ...)
- Operazioni CREATE, ALTER, DROP

```
mysql> create table studente (matricola int, nome varchar(100));  
mysql> drop table esame;
```

## DML – Data Manipulation Language

- Interrogazione e modifica dei dati
- Operazioni **CRUD**: Create, Read, Update, Delete

```
mysql> select * from studente;  
mysql> update studente set name = "Mario";
```

## DCL – Data Control Language

- Controllo del DBMS e dei database

```
mysql> use univ;  
mysql> show databases;
```

# MySQL DML – Inserimento

- × Inserimento di un record in una tabella

> INSERT INTO studente

```
SET matricola = "972061", cognome = "Rossi", nome = "Mario";
```

Assegnamento esplicito dei valori delle colonne



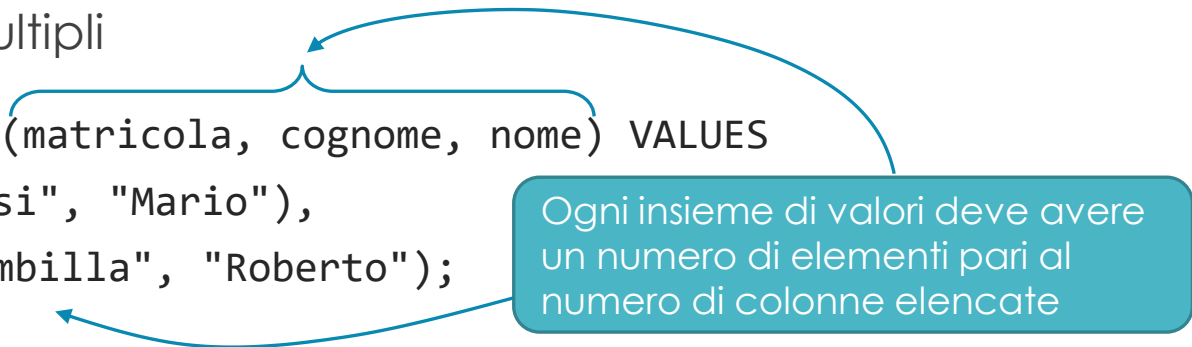
- × Inserimento di record multipli

> INSERT INTO studente (matricola, cognome, nome) VALUES

```
("972061", "Rossi", "Mario"),
```

```
("092682", "Brambilla", "Roberto");
```

Ogni insieme di valori deve avere un numero di elementi pari al numero di colonne elencate



- × Sintassi completa: <https://dev.mysql.com/doc/refman/5.7/en/insert.html>

# MySQL DML – Modifica/aggiornamento

- × Aggiornamento di uno o più record di una tabella
  - Sintassi simile ad una SELECT per selezionare il set di record da aggiornare

```
> UPDATE studente
    SET cognome = "Verdi", codice_fiscale = "VRDPLA92E03F205L"
    WHERE matricola = "073448"
    LIMIT 1;
```

Modifiche da apportare

Condizione di selezione dei record da aggiornare

Limita il numero di risultati da modificare (in questo esempio specifico è superfluo se matricola è chiave primaria)

- × Sintassi completa:  
<https://dev.mysql.com/doc/refman/5.7/en/update.html>

# MySQL DML – Modifica/aggiornamento

---

- × L'aggiornamento può coinvolgere dati di più tabelle
  - Le tabelle coinvolte possono essere selezionate con join implicito o esplicito, o con una subquery
  
- × Un esempio (decisamente irrealistico, ma è importante il concetto...)

*Assegnare allo studente con codice fiscale VRDPLA92E03F205L il voto massimo nell'esame con codice ES1234, e aggiungere un punto al suo voto di laurea*

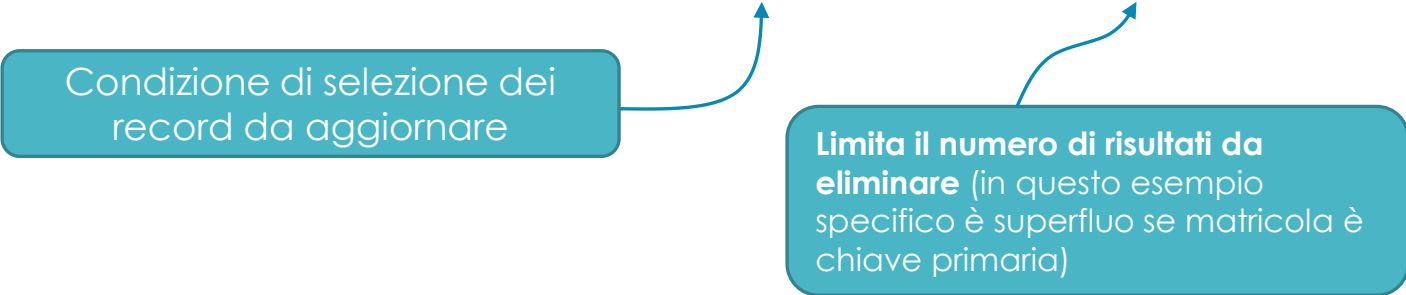
```
> UPDATE esame AS e
      INNER JOIN studente AS s ON e.studente = s.matricola
      SET s.voto_laurea = s.voto_laurea + 1, e.voto = 30
      WHERE s.codice_fiscale = "VRDPLA92E03F205L" AND e.codice = "ES1234";
```

# MySQL DML – Eliminazione

- × Eliminazione di uno o più record di una tabella
  - Sintassi simile ad una SELECT per selezionare il set di record da eliminare

> DELETE FROM studente WHERE matricola = "073448" LIMIT 1;

Condizione di selezione dei record da aggiornare



**Limita il numero di risultati da eliminare** (in questo esempio specifico è superfluo se matricola è chiave primaria)

- × Sintassi completa: <https://dev.mysql.com/doc/refman/5.7/en/delete.html>

# MySQL DML – Eliminazione

- × Anche l'eliminazione può coinvolgere dati di più tabelle
  - Sia in join (implicito o esplicito), sia in subquery
- × *Eliminare gli esami sostenuti dagli studenti che si sono laureati*

```
> DELETE e FROM esame AS e
      INNER JOIN studente AS s
            ON e.studente = s.matricola
      WHERE s.voto_laurea IS NOT NULL;
```

In caso di join bisogna specificare anche **da quali delle tabelle coinvolte effettuare l'eliminazione**

*... oppure con un'interrogazione nidificata ...*

```
> DELETE FROM esame AS e WHERE e.studente IN (
      SELECT s.matricola FROM studente AS s
      WHERE s.voto_laurea IS NOT NULL
    );
```

# MySQL DML – Safe mode

- × MySQL per impostazione di default cerca di impedire modifiche massive accidentali
  - Impedisce DELETE e UPDATE se la condizione nel WHERE non coinvolge la chiave primaria

```
> DELETE FROM studente  
    WHERE nome = "Paolo" AND cognome = "Rossi";
```

La chiave primaria è **matricola** → errore!

- × Il safe mode è bypassabile in due modi:
  - Includendo la clausola LIMIT 1

```
DELETE FROM studente WHERE nome = "Paolo" AND cognome = "Rossi" LIMIT 1;
```

- Disabilitandolo per la sessione in corso:

```
SET SQL_SAFE_UPDATES = 0;
```



# Importazione da CSV

- × MySQL consente di importare data set da file CSV
  - Un file CSV rappresenta dati tabellari
  - Un record per riga, e il valore di ogni colonna è separato da una virgola (o da un altro carattere separatore)

id,first\_name,last\_name,birth\_date,address,city,country

Linea di **intestazione**  
(nomi delle colonne)

Record

1,Kaitlin,Holman,1980-07-04,5279 Parturient Road,Gonars,France  
2,Graham,Williams,1978-06-02,843-1422 Fermentum Road,Enna,Togo  
3,Quail,Burns,1973-04-06,"184-1266 Sem, Rd.",Broxburn,Canada

...

Racchiuso tra **doppi apici**  
poiché contiene il separatore

- × Per importare da file CSV si usa lo statement `LOAD DATA INFILE`
  - <https://dev.mysql.com/doc/refman/5.7/en/load-data.html>

Tabella in cui  
importare i dati

```
> LOAD DATA INFILE 'file.csv'  
INTO TABLE table_name;
```

Path del file CSV  
da importare

# Importazione da CSV – Opzioni

- × Il formato CSV è molto "variabile"
  - Diversi separatori, linee in eccesso, diversi caratteri di fine riga, diverse codifiche...
- × Potrebbe essere necessario escludere delle righe o colonne, o elaborare i valori prima di inserirli
- × Alcune tuple potrebbero già esistere

**Se esiste già** una tuple con la stessa chiave primaria:

- REPLACE: la sovrascrive
- IGNORE: non la importa

Specifica se il file si trova in locale

```
> LOAD DATA [LOCAL] INFILE 'file.csv' [REPLACE | IGNORE]
```

```
INTO TABLE table_name
```

```
CHARACTER SET charset_name
```

Codifica del file (utf8, latin1, ...)

```
FIELDS TERMINATED ','
```

Separatore dei valori

Delimitatore di stringa e carattere di escape

```
[OPTIONALLY] ENCLOSED BY '"' ESCAPED BY '\\'
```

Separatori di linea

```
LINES STARTING BY '' TERMINATED BY '\\n'
```

Numero di linee da ignorare

```
IGNORE n LINES
```

Espressioni per assegnare valori calcolati

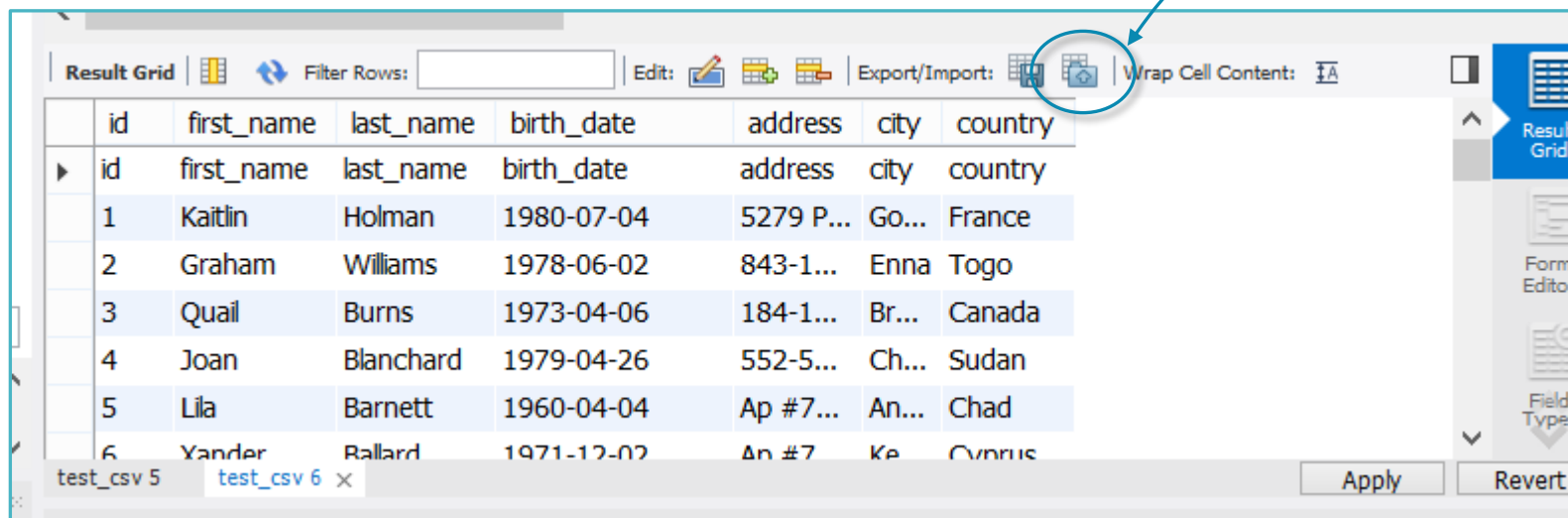
```
(col1, col2, ..., colN) }  
{ [SET col1 = expr1, col2 = expr2, ...];
```

Assegna le colonne del file alle colonne della tabella (un elemento per ogni campo del file)

# Importazione da CSV con Workbench

- ✗ MySQL Workbench consente di importare dati da file CSV, ma non consente di definire alcuna opzione!
  - Non possono esistere righe o colonne in eccesso
  - I valori possono essere separati unicamente da virgole
  - I caratteri separatori di linea devono essere quelli utilizzati dal sistema operativo in uso
  - Non devono esistere duplicati
  - Non è possibile applicare espressioni sui valori da inserire

Per importare un file CSV...



# Esportazione di dataset

- × MySQL consente anche di esportare data set in un file testuale
  - Si aggiunge la clausola INTO OUTFILE alla SELECT che seleziona i dati da esportare
  - Se non vengono specificate opzioni viene generato un file in cui i valori sono separati da caratteri di tabulazione (\t)

```
> SELECT ... INTO OUTFILE 'file.txt' FROM ... WHERE ...;
```

Path del file in cui esportare i dati

- Su Windows, la cartella in cui si possono scrivere/leggere files è `C:/ProgramData/MySQL/MySQL Server 8.0/Uploads`: path differenti, di default, non sono permessi
- × È possibile controllare il formato di output
  - Sintassi completa: <https://dev.mysql.com/doc/refman/5.7/en/select-into.html>

```
> SELECT ...
```

```
INTO OUTFILE 'file.csv'
```

```
CHARACTER SET utf8
```

```
FIELDS TERMINATED BY ',' [OPTIONALLY] ENCLOSED BY ''''
```

```
LINES TERMINATED BY '\n'
```

```
FROM ... WHERE ...;
```

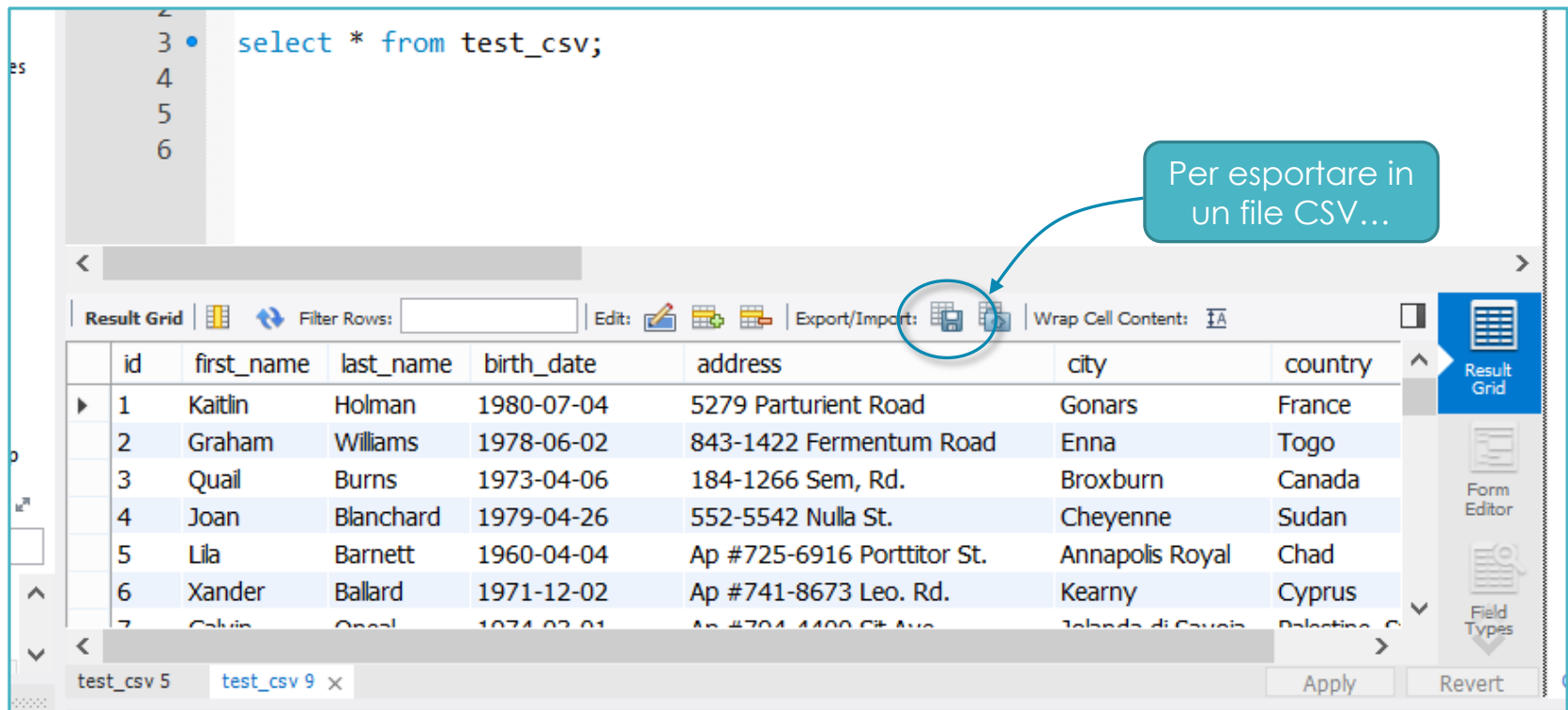
Codifica del file esportato

Delimitatore di stringa e carattere di escape

Separatore di linea

# Esportazione di dataset con Workbench

- × MySQL Workbench consente di esportare un dataset in CSV (o altri formati)
  - È possibile esportare il risultato dell'esecuzione di qualsiasi interrogazione
  - Come per l'importazione di dati da file CSV, Workbench non consente di specificare opzioni



Per esportare in un file CSV...

```
3 • select * from test_csv;
```

	id	first_name	last_name	birth_date	address	city	country
▶	1	Kaitlin	Holman	1980-07-04	5279 Parturient Road	Gonars	France
	2	Graham	Williams	1978-06-02	843-1422 Fermentum Road	Enna	Togo
	3	Quail	Burns	1973-04-06	184-1266 Sem, Rd.	Broxburn	Canada
	4	Joan	Blanchard	1979-04-26	552-5542 Nulla St.	Cheyenne	Sudan
	5	Lila	Barnett	1960-04-04	Ap #725-6916 Porttitor St.	Annapolis Royal	Chad
	6	Xander	Ballard	1971-12-02	Ap #741-8673 Leo. Rd.	Kearny	Cyprus
	7	Colin	Ones	1974-03-01	Ap #704-4400 Cit Ave	Johada di Couais	Palastina

# Esercizi

---

# Esercizio 10

---

## × **Costruzione del database**

1. Utilizzare il database progettato nell'Esercizio 7, rinominato `es10_scuola.sql` e reso disponibile per mezzo della piattaforma di eLearning: **Scuola Secondaria**
2. Eseguire lo script sull'installazione di MySQL locale

## × **Popolamento del database:** utilizzando lo statement `INSERT`, inserire nel database i seguenti dati:

1. Almeno 5 classi di 3 diversi anni, di cui almeno una classe del quinto anno
2. Almeno 5 studenti, di cui almeno 1 straniero, e 2 che siano fratello e sorella. Gli studenti devono risiedere in almeno 2 comuni diversi.
3. Almeno 3 materie, con almeno 1 argomento ognuna
4. Almeno 1 propedeuticità tra argomenti
5. Almeno 3 insegnanti, di cui almeno 1 insegna anche in un'altra scuola
6. Almeno 5 valutazioni relative ad almeno 3 studenti differenti
7. Creare tutte le associazioni necessarie per assegnare gli studenti alle relative classi, e per associare gli insegnanti alle materie e alle classi in cui insegnano

## × Suggerimento: per il codice fiscale potete usare un generatore random di stringhe (di dimensione 16 caratteri) come ad esempio <https://www.random.org/strings/>

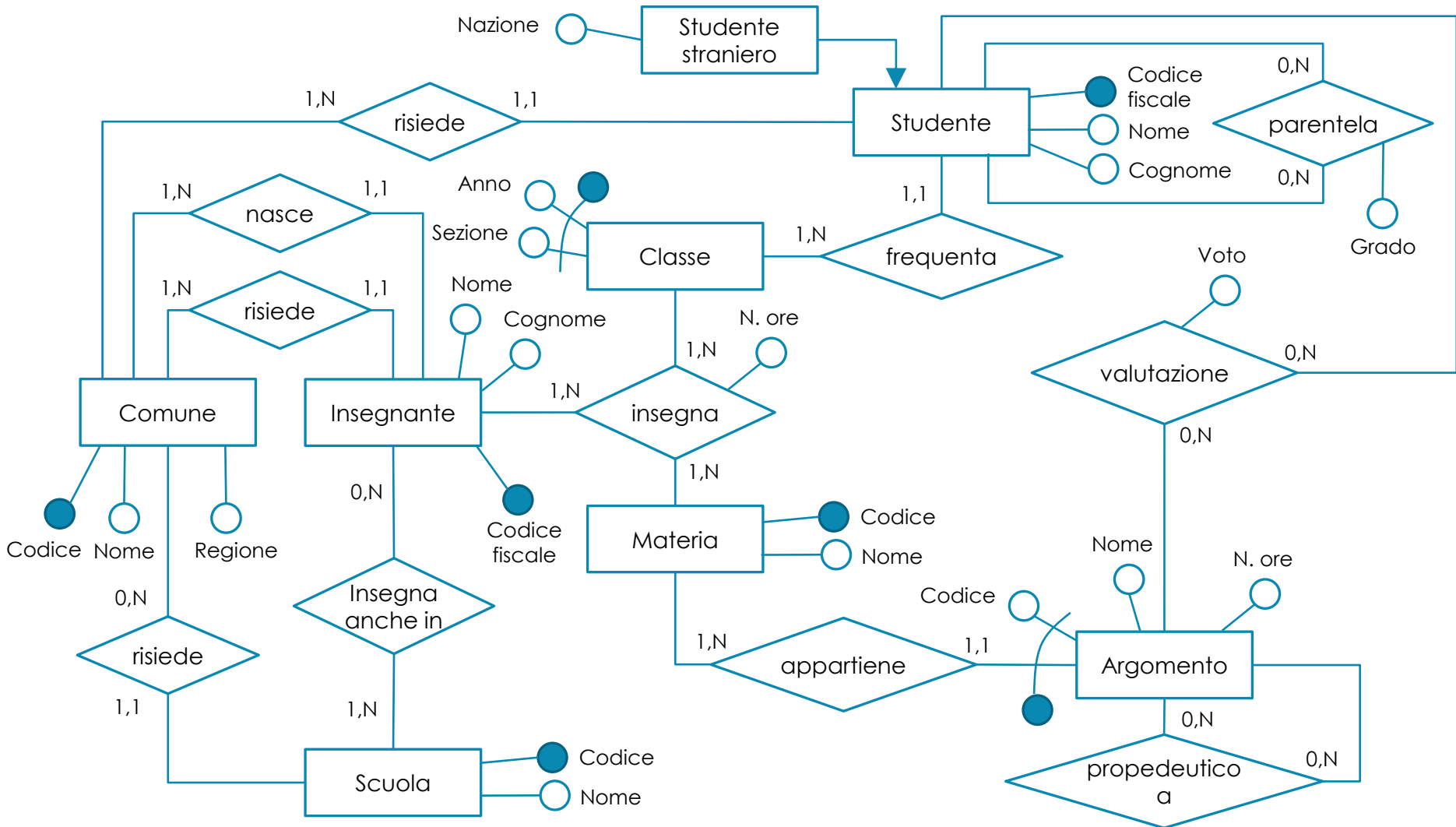
# Esercizio 10

---

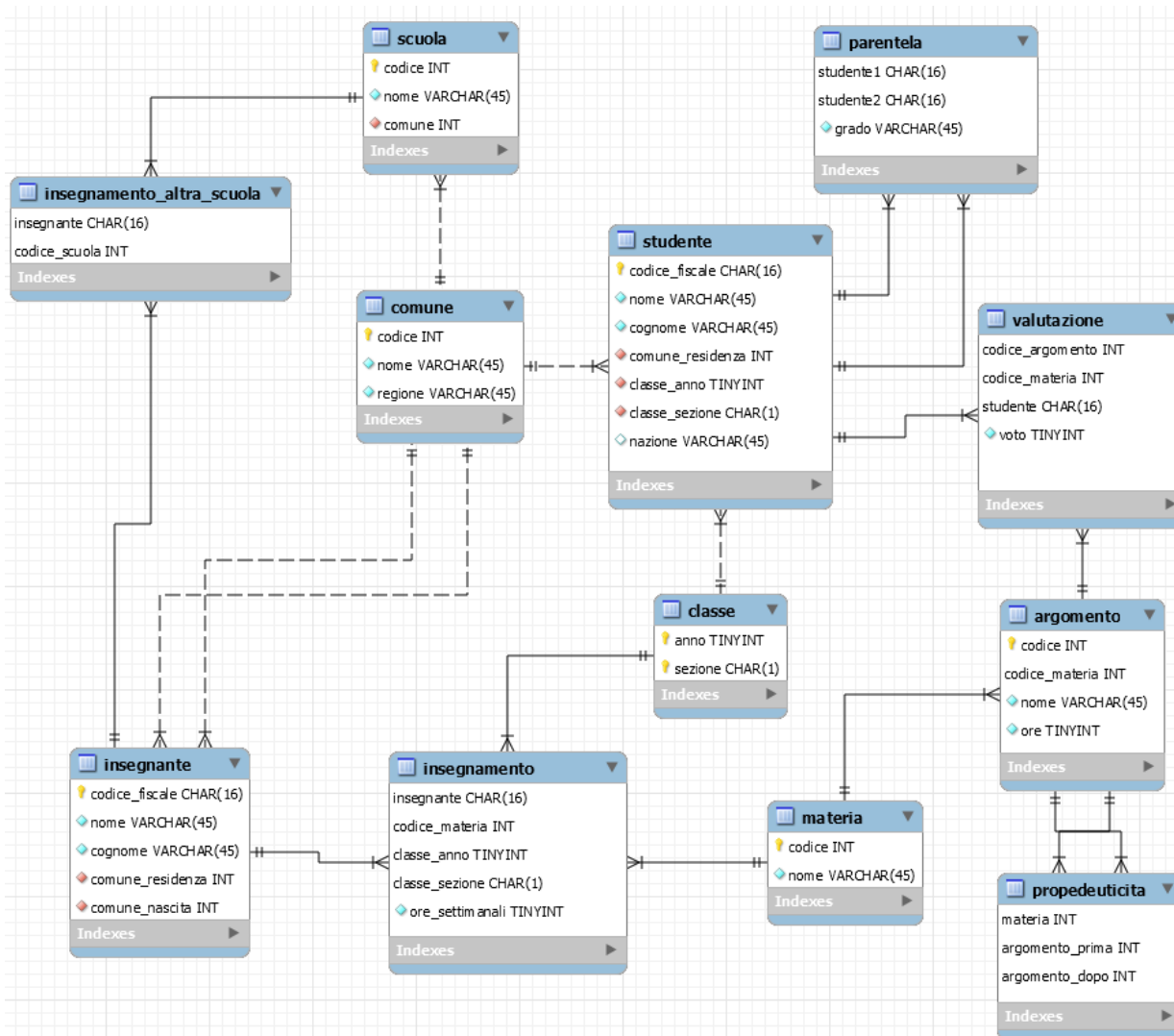
- × **Aggiornamento dei dati:** utilizzando gli statement UPDATE e DELETE, effettuare le seguenti modifiche:
  1. Modificare il cognome di uno degli studenti (selezionandolo per codice fiscale)
  2. Cambiare il comune di residenza di uno degli insegnanti (selezionandolo per nome e cognome)
  3. Raddoppiare il numero di ore di lezione insegnate da un docente per una certa materia in una specifica classe
  4. Eliminare una delle propedeuticità
  5. Modificare il voto di una valutazione di uno studente per uno specifico argomento
  6. Modificare il nome di una delle scuole in cui uno specifico docente insegna (selezionando la scuola sulla base del nome e del cognome del docente che vi insegna) [disattivare il Safe Mode]
  7. Simulare il passaggio all'anno scolastico successivo [disattivare il Safe Mode]:
    - Eliminare tutti gli studenti giunti alla classe 5° (attenzione ai vincoli di integrità referenziale!)
    - Far avanzare di un anno tutti gli altri studenti (con un'unica query)



# Esercizio 10 – Modello ER



# Esercizio 10 – Modellazione in Workbench



# Esercizio 11

---

- × Utilizzare il database popolato nell'Esercizio 10 e il file `es11_studenti.csv` reso disponibile sulla piattaforma di eLearning
  1. Per mezzo della GUI di MySQL Workbench, importare il file CSV nella tabella corrispondente (studenti)
  2. Utilizzando lo statement `SELECT ... INTO OUTFILE` esportare in un file CSV (con i valori separati da virgole) un data set contenente le valutazioni medie per ogni materia insegnata (con codice e nome della materia)
  3. [Opzionale] Scrivere, usando il Data Manipulation Language, uno script per eseguire la stessa operazione richiesta nel punto 1 (non è necessario eseguirlo)