Create ERD and SQLite file

- [ ] In editor.ponyorm.com create two entities: `Lecturer` and `Course`
- [ ]

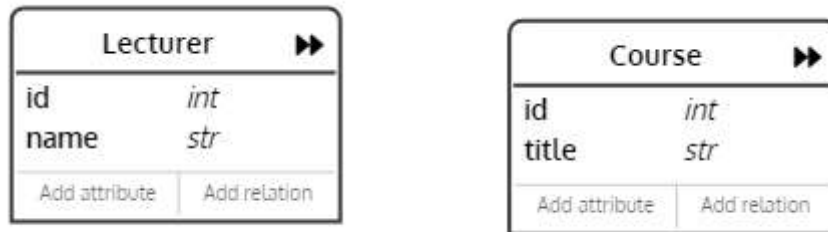 Add attributes `Lecturer.name` and `Course.title`

Figure 1: Entities Lecturer and Course

- [ ] Add a relationship between the entities:
    - A lecturer may be related to **many** courses
    - Each course may be related to **exactly one** lecturer

- You see that the ORM creates **foreign keys** in each entity: `Lecturer.courses` and `Courses.lecturer`.
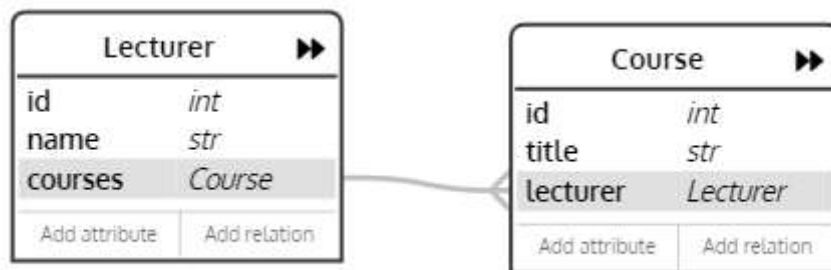

Figure 2: Foreign key attributes courses and lecturer

- [ ]

  You can open a third column with the menu button at the top of each table to see key properties:
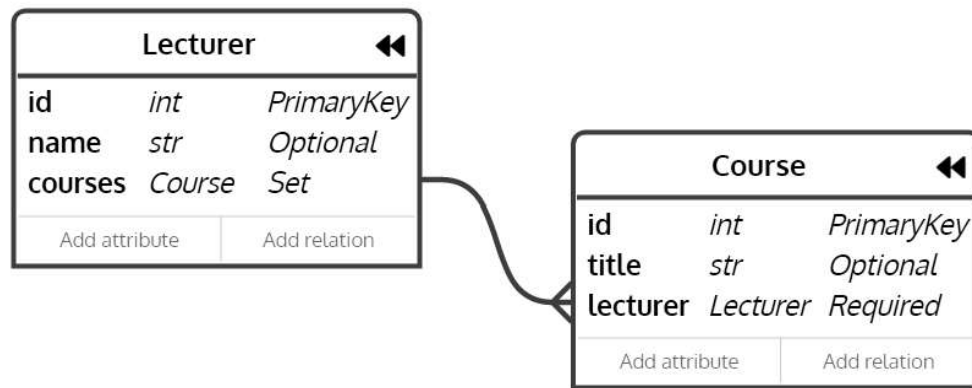
Figure 3: Foreign key attributes courses and lecturer

- [ ] Copy the SQLite code from Ponyorm and save it in a file `students.sql`

Create database

- [ ]

  Take the SQLite code from Ponyorm and create an SQLite database `students.db`

  ```
  #    sqlite3 students.db < students.sql
  ```

- [ ]

  Check for tables.

  ```
  .tables
  ```

  ```
  Course    Lecturer
  ```

- [ ]

  Check with `.schema` what the required data types are, or look at the Ponyorm ERD.

  ```
  .schema
  ```

  ```
  CREATE TABLE IF NOT EXISTS "Lecturer" (
    "id" INTEGER PRIMARY KEY AUTOINCREMENT,
    "name" TEXT NOT NULL
  );
  CREATE TABLE sqlite_sequence(name,seq);
  CREATE TABLE IF NOT EXISTS "Course" (
    "id" INTEGER PRIMARY KEY AUTOINCREMENT,
    "title" TEXT NOT NULL,
  ```

```
    "lecturer" INTEGER NOT NULL REFERENCES "Lecturer" ("id") ON DELETE CASCADE
);
CREATE INDEX "idx_course__lecturer" ON "Course" ("lecturer");
```

- When the table is deleted, CASCADE means that if the record in the parent table (Lecturer) is deleted, the record in the child table (Course) will also automatically be deleted - e.g. if a lecturer is removed - then his/her course links are also removed ([doc](#))

- It is instructive to look at the different ways in which RDBMS implement the PK constraints. E.g. for PostgreSQL (the RDBMS used in DataCamp), it looks like this:

```
ALTER TABLE "course"
ADD CONSTRAINT "fk_course__lecturer"
FOREIGN KEY ("lecturer") REFERENCES "lecturer" ("id")
ON DELETE CASCADE
```

- The FK course.lecturer is mapped onto lecturer.id as in SQLite
- SQLite keeps tracks of relationships with a separate index table

INSERT data into the Lecturer table

- [ ]

  Note that you cannot keep running these INSERT commands. After the first time a PK is used, it is protected as unique.

  ```
  INSERT INTO Lecturer VALUES (1, "Sonnier");
  INSERT INTO Lecturer VALUES (2, "Birkenkrahe");
  INSERT INTO Lecturer VALUES (3, "Jones");
  ```

- [ ]

  Check table content.

  ```
  SELECT * FROM Lecturer;
  ```

  ```
  id  name
  --  -----------
  1   Sonnier
  2   Birkenkrahe
  3   Jones
  ```

  ```
  id  name
  --  -----------
  1   Sonnier
  2   Birkenkrahe
  3   Jones
  ```

- [ ]

  The table Course requires three columns, one is the responsible lecturer. Show the schema only for Course.

```
.schema Course
```

```
CREATE TABLE IF NOT EXISTS "Course" (
  "id" INTEGER PRIMARY KEY AUTOINCREMENT,
  "title" TEXT NOT NULL,
  "lecturer" INTEGER NOT NULL REFERENCES "Lecturer" ("id") ON DELETE CASCADE
);
CREATE INDEX "idx_course__lecturer" ON "Course" ("lecturer");
```

- [ ]

Insert course information.

```
INSERT INTO Course VALUES (1, "Databases", 2);
INSERT INTO Course VALUES (2, "Modeling",  2);
INSERT INTO Course VALUES (3, "Logic", 1);
INSERT INTO Course VALUES (4, "Algorithms", 1);
INSERT INTO Course VALUES (5, "Operating_Systems", 2);
INSERT INTO Course VALUES (6, "Python", 3);
INSERT INTO Course VALUES (7, "Data_structures", 1);
INSERT INTO Course VALUES (8, "Data_science", 2);
```

- [ ]

Check content.

```
SELECT * FROM Course;
```

```
id  title             lecturer
--  ----------------  --------
1   Databases         2
2   Modeling          2
3   Logic             1
4   Algorithms        1
5   Operating_Systems 2
6   Python            3
7   Data_structures   1
8   Data_science      2
```

JOIN TABLES

- [ ]

Which courses do I teach?

```
SELECT Course.title AS title, Lecturer.name as name
FROM Course JOIN Lecturer ON Course.lecturer = Lecturer.id
WHERE name = "Birkenkrahe";
```

```
title               name
----------------    -----------
Databases           Birkenkrahe
Modeling            Birkenkrahe
Operating_Systems   Birkenkrahe
Data_science        Birkenkrahe
```

Created: 2022-04-12 Tue 22:48