studadmin database

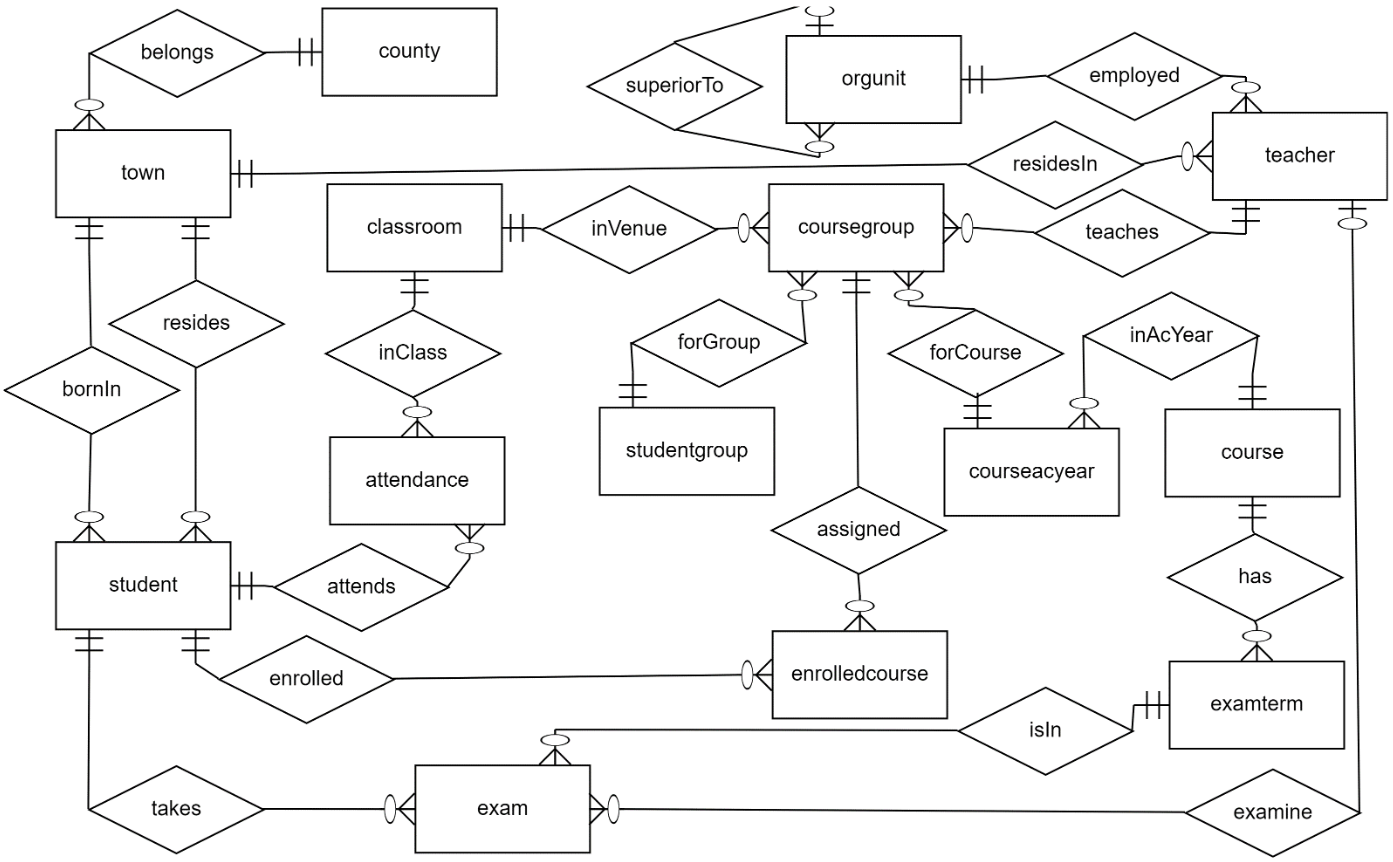
The studAdmin database thematically covers a narrow area from the domain of studying at a higher education institution. Students who enroll in certain courses, participate in courses activities divided into teaching groups and take exams from enrolled subjects are monitored. Records are kept of the classrooms in which each teaching group listens to lectures of a certain academic year, as well as of the teacher who performs those lectures.
For the exam in which the student takes the enrolled course, the final grade, the date of awarding the grade and the teacher who awarded the grade are recorded. Students who fail to pass the course, i.e. do not receive a passing grade, enroll in the course again in the next academic year.

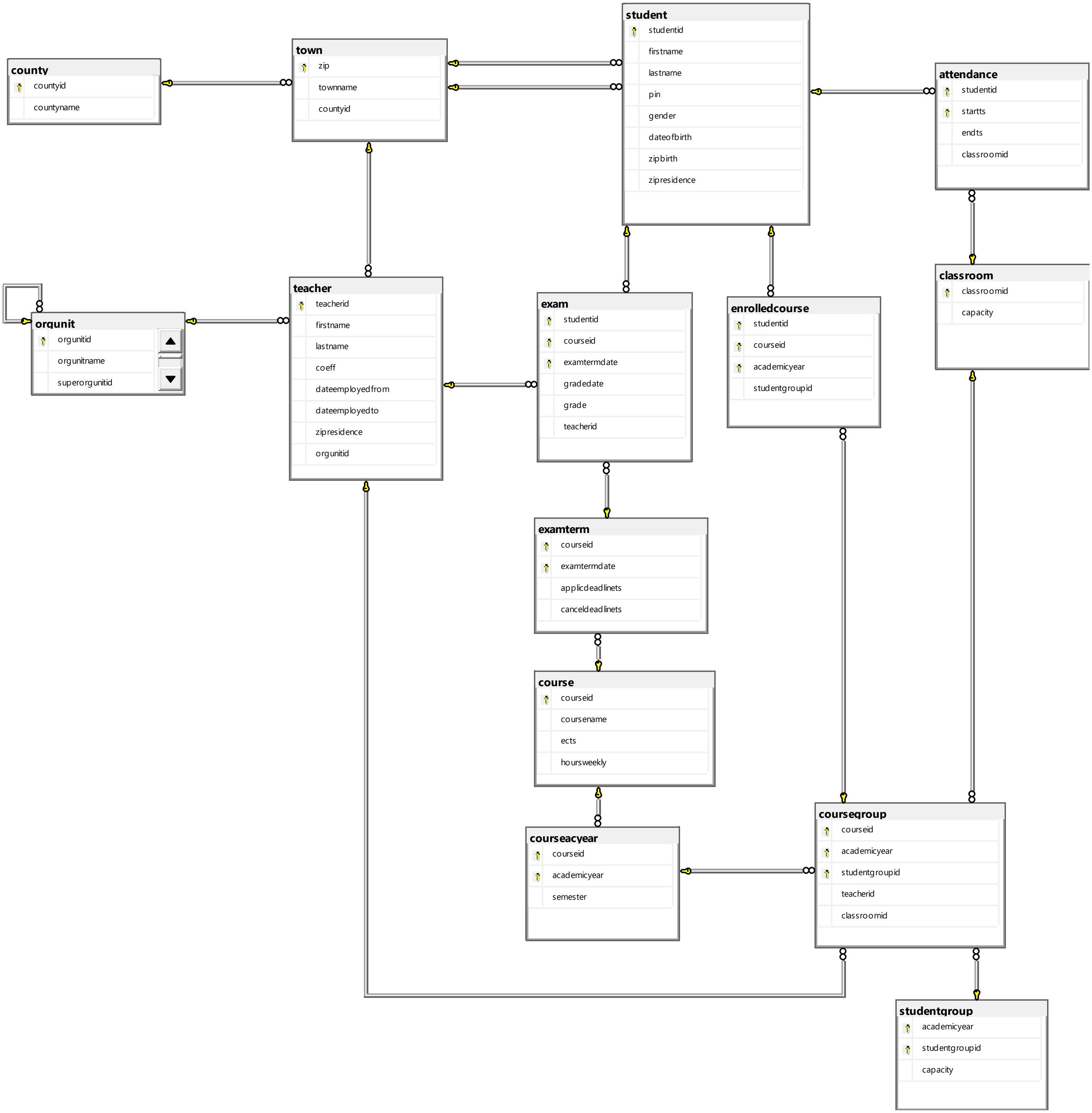
A list of tables and their meaning:

Table	Description	Identifier
county	County catalogue.	countryid
town	Town catalogue. Each town can belong to a county.	zip
classroom	Classrooms used for lectures. Data about each classroom contains its capacity – the number of students that fit into that classroom.	classroomid
orgunit	A catalogue of organizational units such as faculties, departments, laboratories etc. Each organizational unit can have a superior unit, but it doesn't have to.	orgunitid
course	Courses that are thought at the university. Besides a name, a course is also described by a number of ECTS points that a student earns by passing a course, as well as a total weekly number of lecture hours.	courseid
studentgroup	A catalogue of groups of students at the university for a certain academic year. Academic year is represented by a numerical value of a calendar year in which the academic year starts. I.e. for academic year 2020/2021, a numeric value stored in the table is 2020. Groups with the same id can appear in different academic years, but do not necessarily have to. Each group is also described by its capacity – a maximum number of students that can be assigned to it.	academicyear, studentgroupid
student	Students that are studying at the university (enrolling in courses). A student is uniquely identified by its **studentid** number. Each student is also described by his **pin** number, first and last name, gender, date of birth and zip code for his birth town and residence. If student's **pin** number is known, it has to be unique.	studentid
teacher	Persons that teach classes at the university. Each teacher is described by his first and last name, residence zip code, organizational unit that employs him, salary coefficient and the dates when the teacher's employment starts and ends. Employment start date is noted for all teachers, while end date is noted only for those teachers that stopped working at the university.	teacherid
courseacyear	A list of courses taught in the academic year, and the ordinal number of the semester in the academic year (1 for winter semester courses, 2 for summer semester courses). In one academic year, the course cannot be taught in both semesters.	courseid, academicyear

coursegroup	A catalogue of student groups for each course and academic year. Different courses have a different set of groups. Each course group needs also to be in the catalogue of student groups for the same academic year. Each course group is described by a classroom in which the lectures are given and the teacher giving lectures to that course group. One course group can have only one teacher in the same academic year.	courseid, academicyear, studentgroupid
enrolledcourse	Courses that students enrolled in. Each enrollment is described by an academic year in which the student enrolled in a course and by a course group in which the student was taking lectures. A student may enroll in the same subject more than once (only if the student did not pass the course in the previous enrollments) but not in the same academic year.	studentid, courseid, academicyear
examterm	Examination term held for each course at the university. For the examination term, the date of the term, the final date and time for registration at that term, and the final date and time for deregistration at that term are recorded	examtermdate, courseid
exam	Records of the exams at the university. The student’s **studentid** number, **coursed** of the examined course, the date of the exam and the date of awarding the grade, the grade the student received at the exam and the teacher who held the exam and awarded the grade are recorded. A grade can have one of the values from the set {1, 2, 3, 4, 5, NULL}. Grades 2, 3, 4 and 5 means that the student passed the course, grade 1 means that the student failed the course while value NULL means that the grade is yet unknown (i.e. the student is still taking lectures). A student can enroll in a course repeatedly (as long as he failed previous enrollments), but only once in the same academic year. Only one enrollment for the same course can have a passing grade.	courseid, studenid, examtermdate
attendance	Records of student entries in classroom using a smart card by checking in and out on a smart card reader. Studentid, classroomid as well as date and time of entry and exit from the classroom are recorded. Date and time are recorded to the level of seconds. NULL value of the date and time of exit from the classroom is recorded when students forget to check out when leaving the classroom.	studentid, dateTimeStart

ER model for studadmin database:





county
countyid
countyname

town
zip
townname
countyid

student
studentid
firstname
lastname
pin
gender
dateofbirth
zipbirth
zipresidence

attendance
studentid
startts
endts
classroomid

orgunit
orgunitid
orgunitname
superorgunitid

teacher
teacherid
firstname
lastname
coeff
dateemployedfrom
dateemployedto
zipresidence
orgunitid

exam
studentid
courseid
examtermdate
gradedate
grade
teacherid

enrolledcourse
studentid
courseid
academicyear
studentgroupid

classroom
classroomid
capacity

examterm
courseid
examtermdate
applicdeadlinets
canceldeadlinets

course
courseid
coursename
ects
hoursweekly

courseacyear
courseid
academicyear
semester

coursegroup
courseid
academicyear
studentgroupid
teacherid
classroomid

studentgroup
academicyear
studentgroupid
capacity

```
CREATE DATABASE studadmin;

CREATE TABLE county (
  countyid SMALLINT NOT NULL,
  countyname VARCHAR(40) NOT NULL,
  PRIMARY KEY (countyid)
);

CREATE TABLE town (
  zip INTEGER NOT NULL,
  townname VARCHAR(40) NOT NULL,
  countyid SMALLINT,
  PRIMARY KEY (zip),
  FOREIGN KEY (countyid) REFERENCES county(countyid)
);

CREATE TABLE classroom (
  classroomid VARCHAR(5) NOT NULL,
  capacity INTEGER DEFAULT 40,
  PRIMARY KEY (classroomid)
);

CREATE TABLE orgunit (
  orgunitid INTEGER NOT NULL,
  orgunitname VARCHAR(120) NOT NULL,
  superorgunitid INTEGER,
  PRIMARY KEY (orgunitid),
  FOREIGN KEY (superorgunitid) REFERENCES orgunit(orgunitid)
);

CREATE TABLE course (
  courseid INTEGER NOT NULL,
  coursename VARCHAR(60) NOT NULL,
  ects DECIMAL(4,1),
  hoursweekly SMALLINT,
  PRIMARY KEY (courseid)
);

CREATE TABLE studentgroup (
  academicyear SMALLINT NOT NULL,
  studentgroupid CHAR(10) NOT NULL,
  capacity SMALLINT,
  PRIMARY KEY (academicyear, studentgroupid)
);

CREATE TABLE student (
  studentid VARCHAR(10) NOT NULL,
  firstname VARCHAR(25) NOT NULL,
  lastname VARCHAR(25) NOT NULL,
  pin VARCHAR(11),
  gender VARCHAR(1),
  dateofbirth DATE,
  zipbirth INTEGER,
  zipresidence INTEGER,
  PRIMARY KEY (studentid),
  FOREIGN KEY (zipbirth) REFERENCES town(zip),
  FOREIGN KEY (zipresidence) REFERENCES town(zip),
  UNIQUE (pin)
);
```

```
CREATE TABLE teacher (
  teacherid INTEGER NOT NULL,
  firstname VARCHAR(25) NOT NULL,
  lastname VARCHAR(25) NOT NULL,
  coeff DECIMAL(3,2) NOT NULL,
  dateemployedfrom DATE NOT NULL,
  dateemployedto DATE,
  zipresidence INTEGER,
  orgunitid INTEGER NOT NULL,
  PRIMARY KEY (teacherid),
  FOREIGN KEY (zipresidence) REFERENCES town(zip),
  FOREIGN KEY (orgunitid) REFERENCES orgunit(orgunitid)
);

CREATE TABLE courseacyear (
  courseid INTEGER NOT NULL,
  academicyear SMALLINT NOT NULL,
  semester SMALLINT NOT NULL,
  PRIMARY KEY (courseid, academicyear),
  FOREIGN KEY (courseid) REFERENCES course(courseid)
);

CREATE TABLE coursegroup (
  courseid INTEGER NOT NULL,
  academicyear SMALLINT NOT NULL,
  studentgroupid VARCHAR(10) NOT NULL,
  teacherid INTEGER NOT NULL,
  classroomid VARCHAR(5),
  PRIMARY KEY (courseid, academicyear, studentgroupid),
  FOREIGN KEY (courseid, academicyear) REFERENCES courseacyear(courseid, academicyear),
  FOREIGN KEY (academicyear, studentgroupid)
                  REFERENCES studentgroup(academicyear, studentgroupid),
  FOREIGN KEY (teacherid) REFERENCES teacher(teacherid),
  FOREIGN KEY (classroomid) REFERENCES classroom(classroomid)
);

CREATE TABLE enrolledcourse (
  studentid VARCHAR(10) NOT NULL,
  courseid INTEGER NOT NULL,
  academicyear SMALLINT NOT NULL,
  studentgroupid VARCHAR(10) NOT NULL,
  PRIMARY KEY (studentid, courseid, academicyear),
  FOREIGN KEY (courseid, academicyear, studentgroupid) R
            REFERENCES coursegroup(courseid, academicyear, studentgroupid),
  FOREIGN KEY (studentid) REFERENCES student(studentid)
);

CREATE TABLE attendance (
  studentid VARCHAR(10) NOT NULL,
  startts TIMESTAMP(0) NOT NULL,
  endts TIMESTAMP(0),
  classroomid VARCHAR(5) NOT NULL,
  PRIMARY KEY (startts, studentid),
  FOREIGN KEY (studentid) REFERENCES student(studentid),
  FOREIGN KEY (classroomid) REFERENCES classroom(classroomid)
);
```

```
CREATE TABLE examterm (
  courseid INTEGER NOT NULL,
  examtermdate DATE NOT NULL,
  applicdeadlinets TIMESTAMP(0) NOT NULL,
  canceldeadlinets TIMESTAMP(0) NOT NULL,
  PRIMARY KEY (examtermdate, courseid),
  FOREIGN KEY (courseid) REFERENCES course(courseid)
);

CREATE TABLE exam (
  studentid VARCHAR(10) NOT NULL,
  courseid INTEGER NOT NULL,
  examtermdate DATE NOT NULL,
  gradedate DATE,
  grade SMALLINT,
  teacherid INTEGER,
  PRIMARY KEY (courseid, studentid, examtermdate),
  FOREIGN KEY (studentid) REFERENCES student(studentid),
  FOREIGN KEY (examtermdate, courseid) REFERENCES examterm(examtermdate, courseid),
  FOREIGN KEY (teacherid) REFERENCES teacher(teacherid)
);
```