

Lecture 3 Notes - Wednesday 09/08/2016

Reading Quiz:

Question 1: Ans = C

Question 2: Ans = B

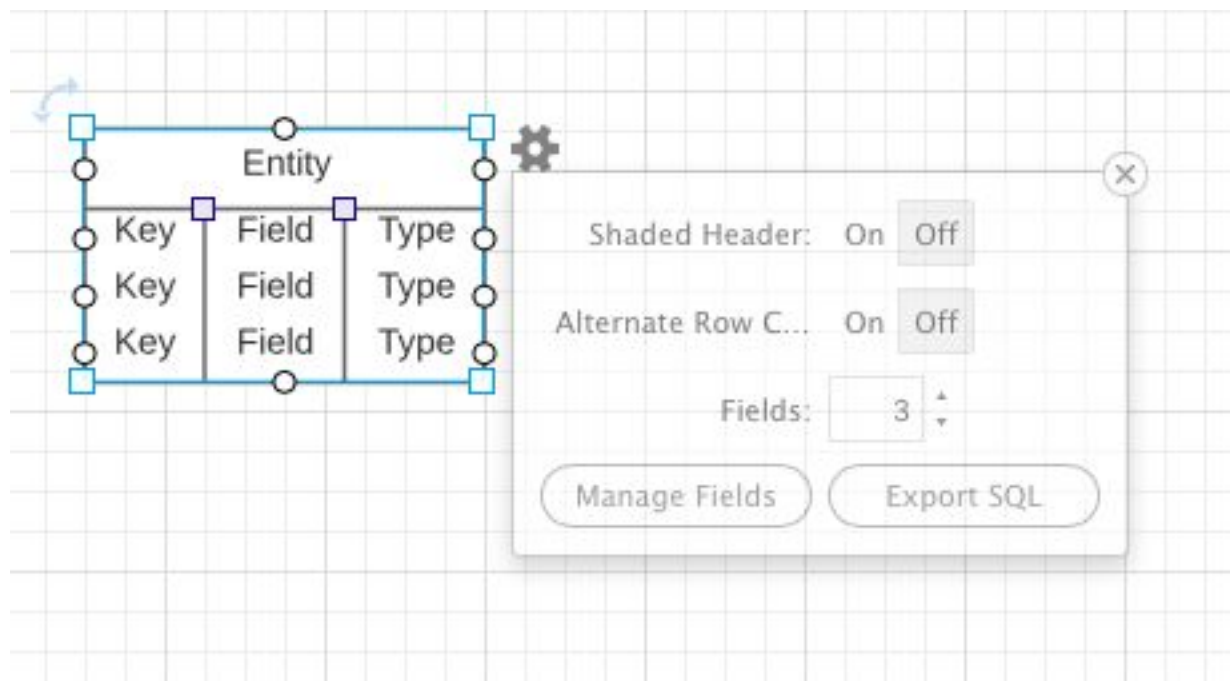
Question 3: Ans = B

Question 4: Ans = A

Question 5: Ans = B

Continued class enrollment exercise from last week: reviewed conceptual diagram (see below) and derived logical diagram (see below)

Conceptual models basically show you which entity classes relate to each other but not how, whereas *logical models* give you more details, e.g., key constraints -- logical models are like coming up with all of the information you need for create table statements, but they're still in picture form instead of SQL code. You should use the following representation (3 column chart) for each entity class in a *logical* model:



We use junction tables for many-to-many relationships, because you can't have multi-dimensional data types (like lists) in SQL. In other words, for row a in table A, we can't have attribute x referencing more than one row in table B -- so we need to use junction tables

when a row in table A wants to point to more than one row in table B and a row in table B wants to point to more than one row in table A.

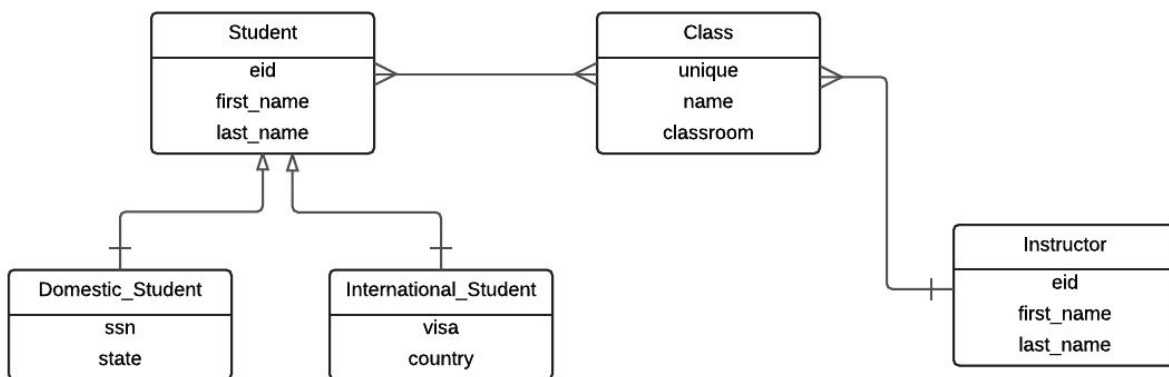
We usually use composite keys for junction tables in order to safeguard against duplicates. Otherwise, using the example from lecture, we could have many rows that all say “abr876” (under eid column) is in the course “CS327E” (under course ID column). Using the composite key (eid, course_id) keeps this from happening because every combination of eid and course must be unique.

Note that removing foreign key constraints can cause “orphan child records,” as an element in the *Child* table (one with the foreign key referring to another table’s primary key) may no longer have an element in the *Parent* table (one with the primary key being referred to) to point to, if the child’s foreign key is not required to be a primary key in the parent table. This can cause issues when you’re trying to perform joins on tables.

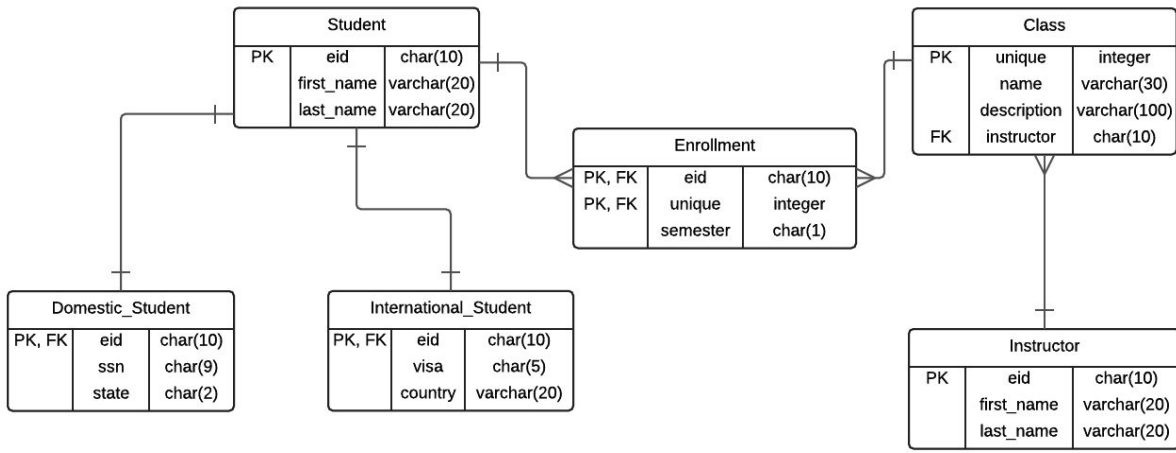
When you’re using LucidChart, keep in mind that it has these limitations:

- Can’t show check restraints
- Isn’t able to show composite keys well

Conceptual ERD - UT Class Enrollment



Logical ERD - UT Class Enrollment



Below are the create table statements for the logical model:

```
drop database if exists utexas;
create database utexas;
use utexas;
```

```
CREATE TABLE Student (
  eid char(10) primary key,
  first_name varchar(20) not null,
  last_name varchar(20) not null
);
```

```
CREATE TABLE Domestic_Student (
  eid char(10) primary key,
  ssn char(9) not null,
  state char(2) not null,
  foreign key (eid) references Student (eid)
);
```

```
CREATE TABLE International_Student (
  eid char(10) primary key,
  visa char(5) not null,
  country char(30) not null,
  foreign key (eid) references Student(eid)
);
```

```
CREATE TABLE Instructor (
  eid char(10) primary key,
  first_name varchar(20) not null,
  last_name varchar(20) not null
);
```

```
CREATE TABLE Class (
  unique_number integer primary key,
  name varchar(30) not null,
  description varchar(100),
  instructor char(10),
  foreign key (instructor) references Instructor(eid)
);
```

```
CREATE TABLE Enrollment (
  student_eid char(10),
  unique_number integer,
  primary key (student_eid, unique_number),
  foreign key (student_eid) references Student(eid),
  foreign key (unique_number) references Class(unique_number)
);
```