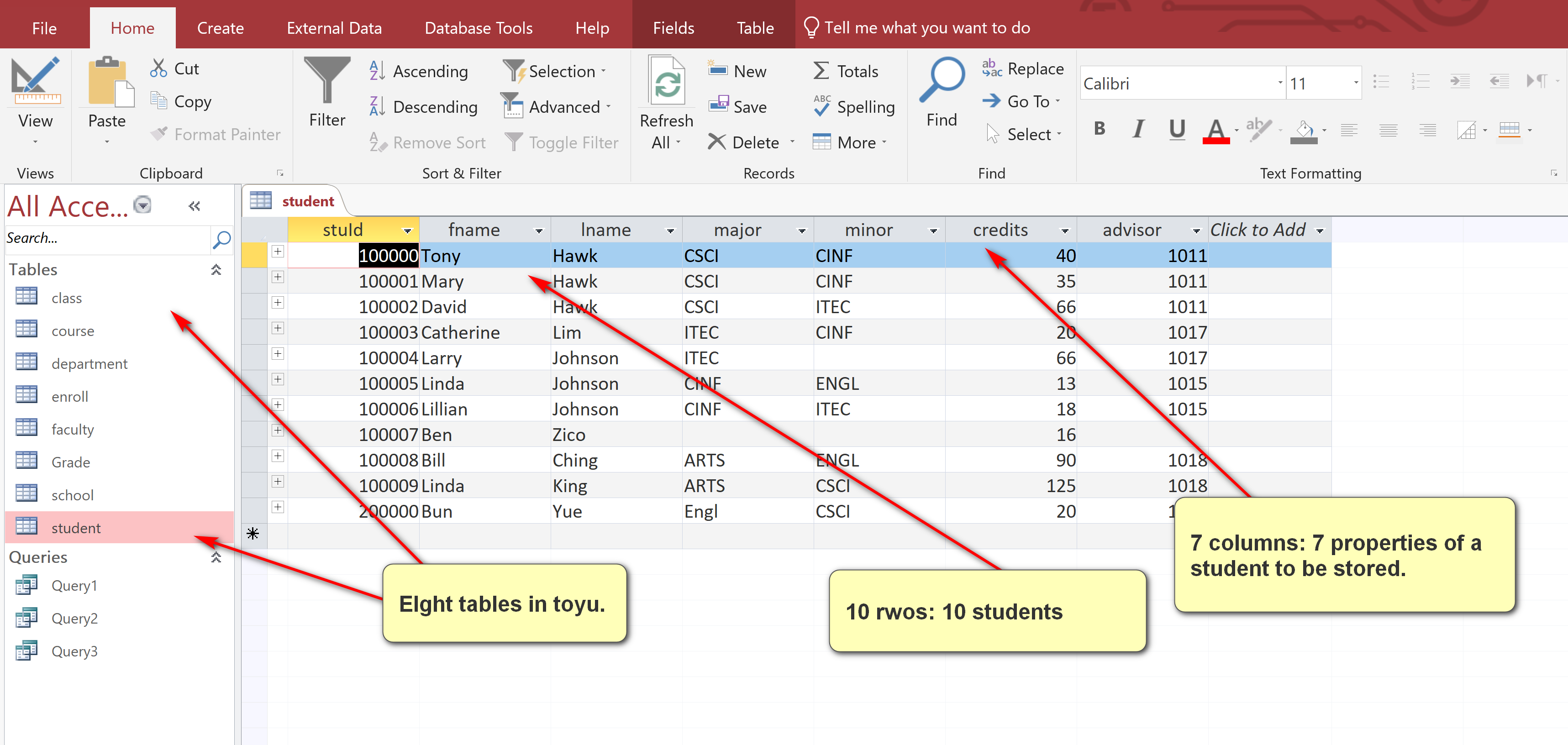
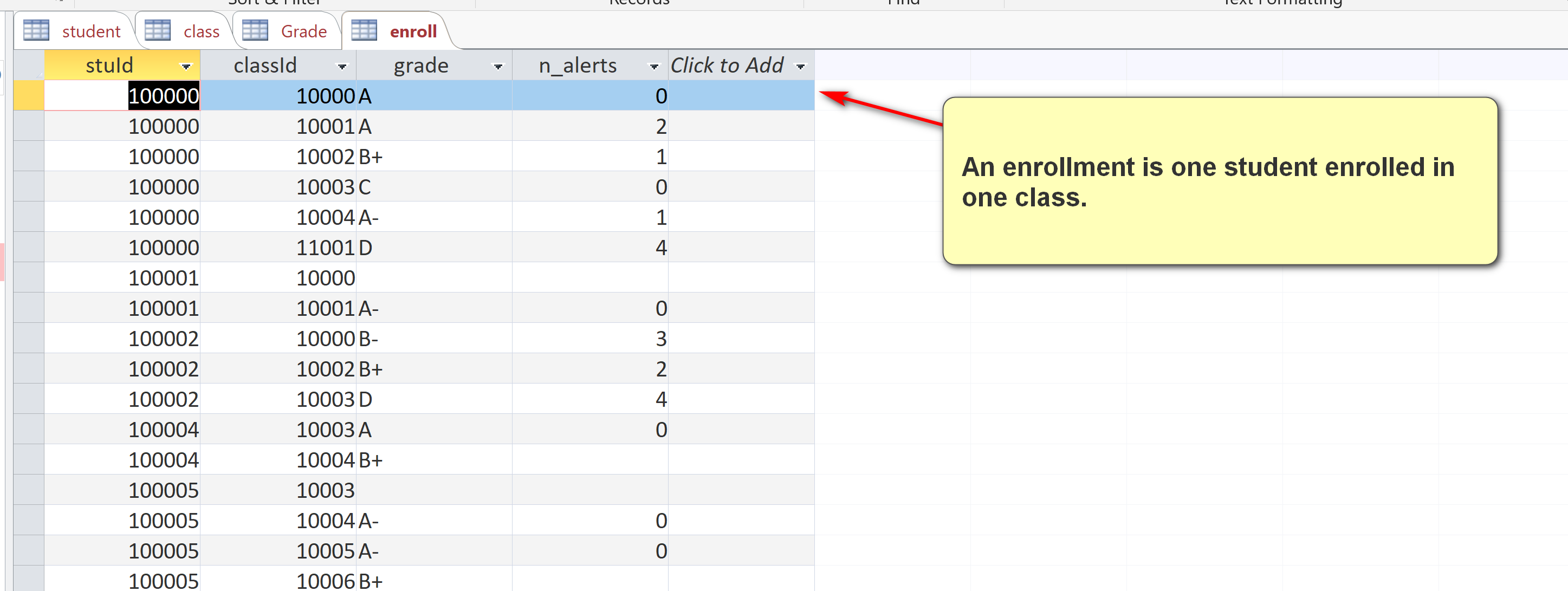
**CSCI 4333 Section 1  
8/31/2020**

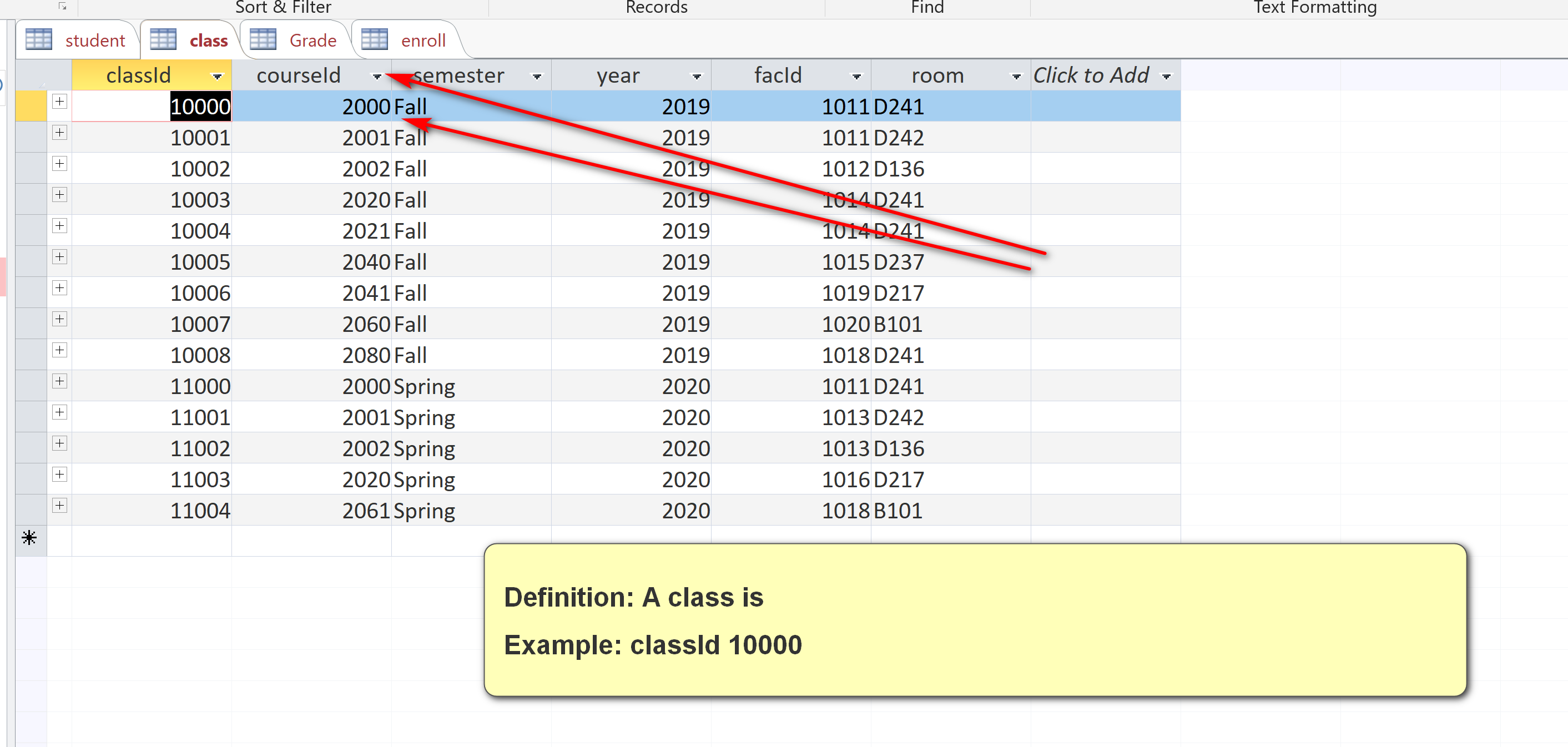
Self-Annotation

MS Access: relational DB: a collection of tables: rows (abstracts an entity) of column values (properties)

.accdb



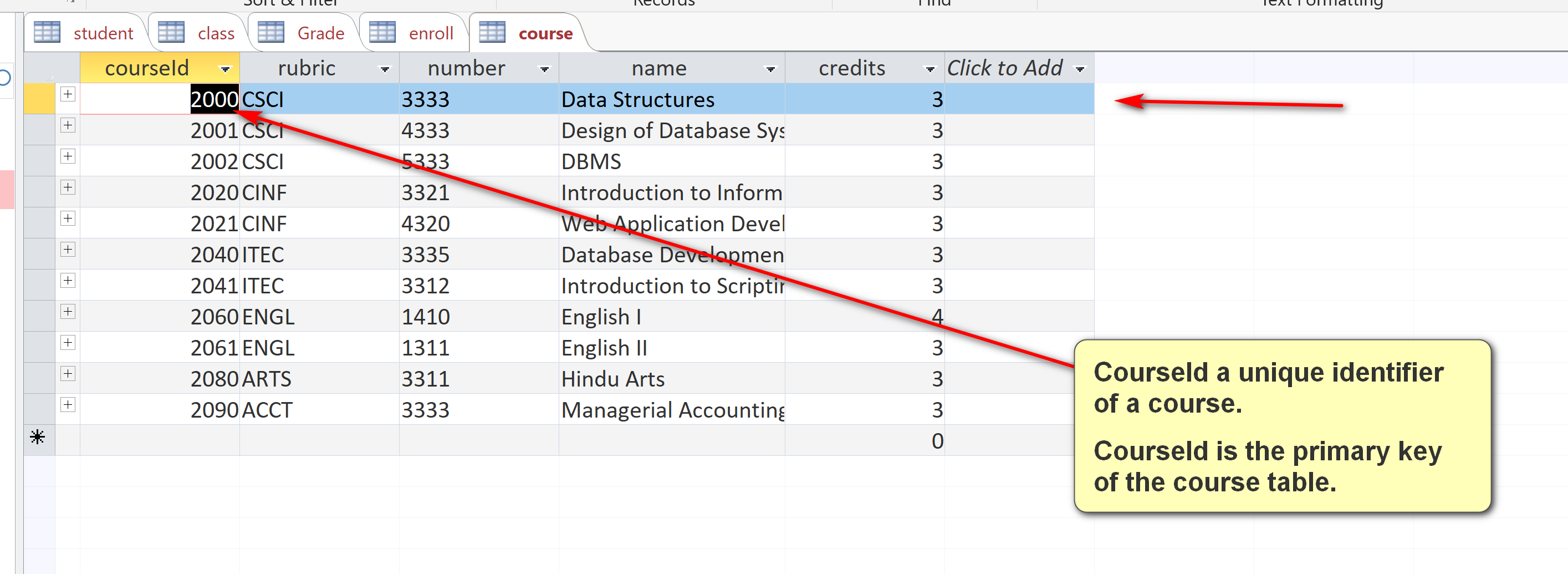


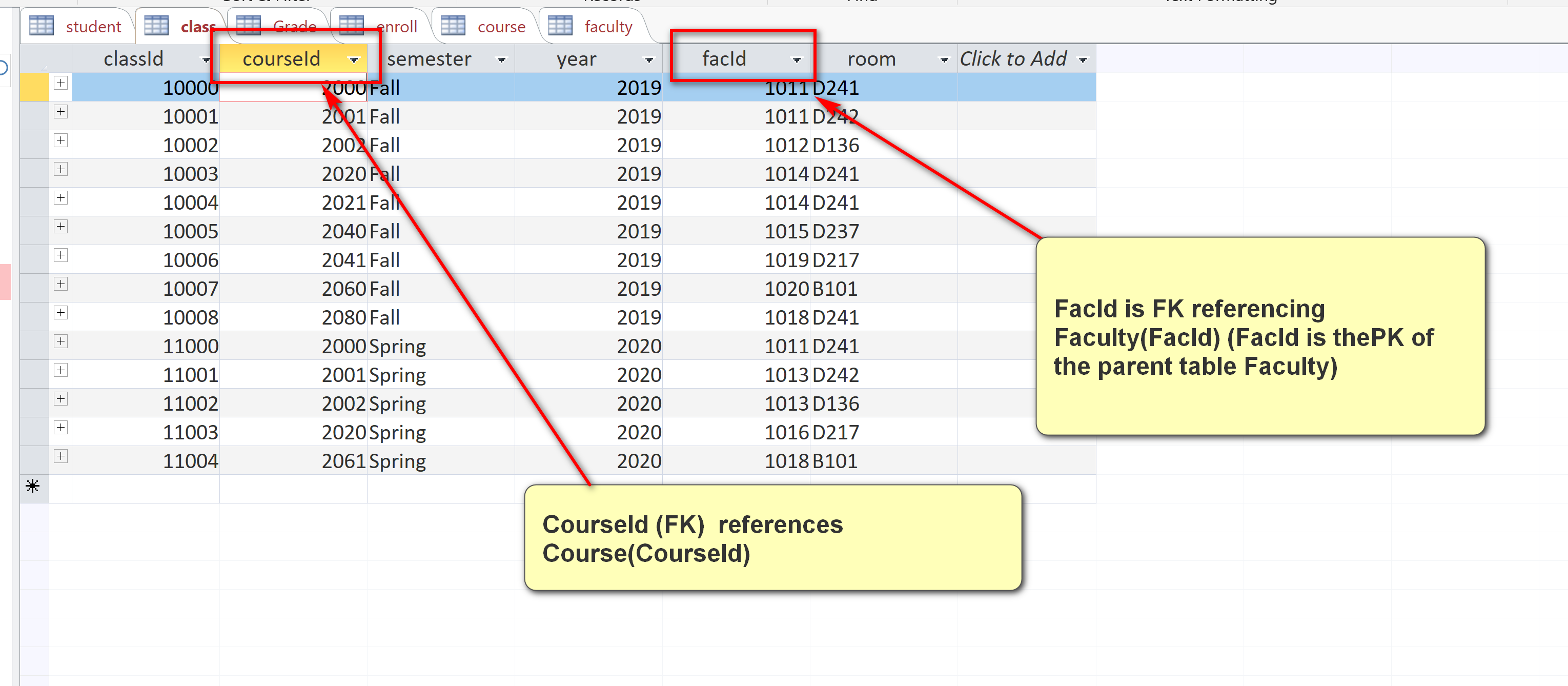


Def: a class is an offering of a course in a particular semester by a specific instructor,

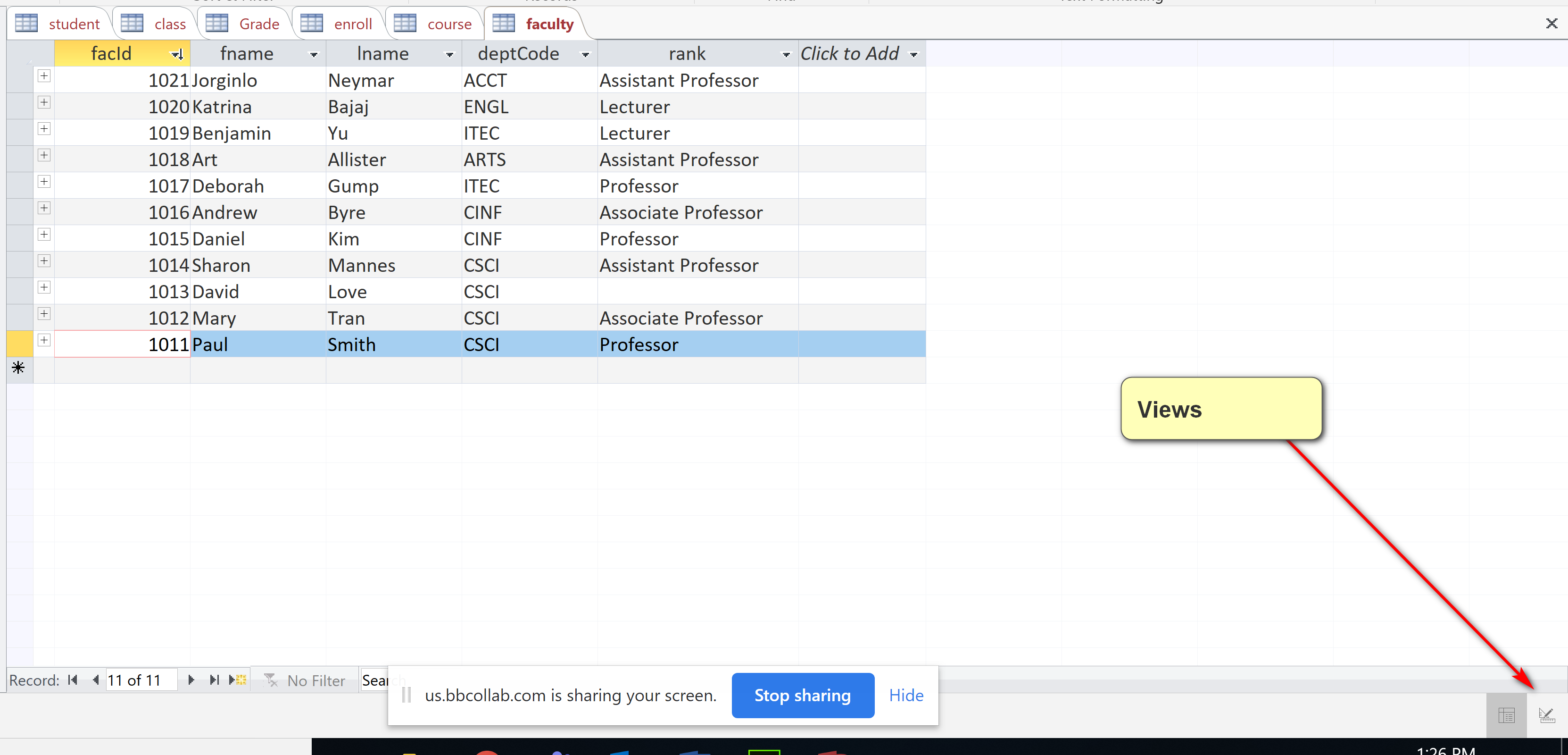
Class: Class(CourseId) references Course(CourseId) (e.g. 2000)

CourseId is a foreign key (FK), e.g. 2000, in the table class, references CourseId, the PK in table course (parent table).



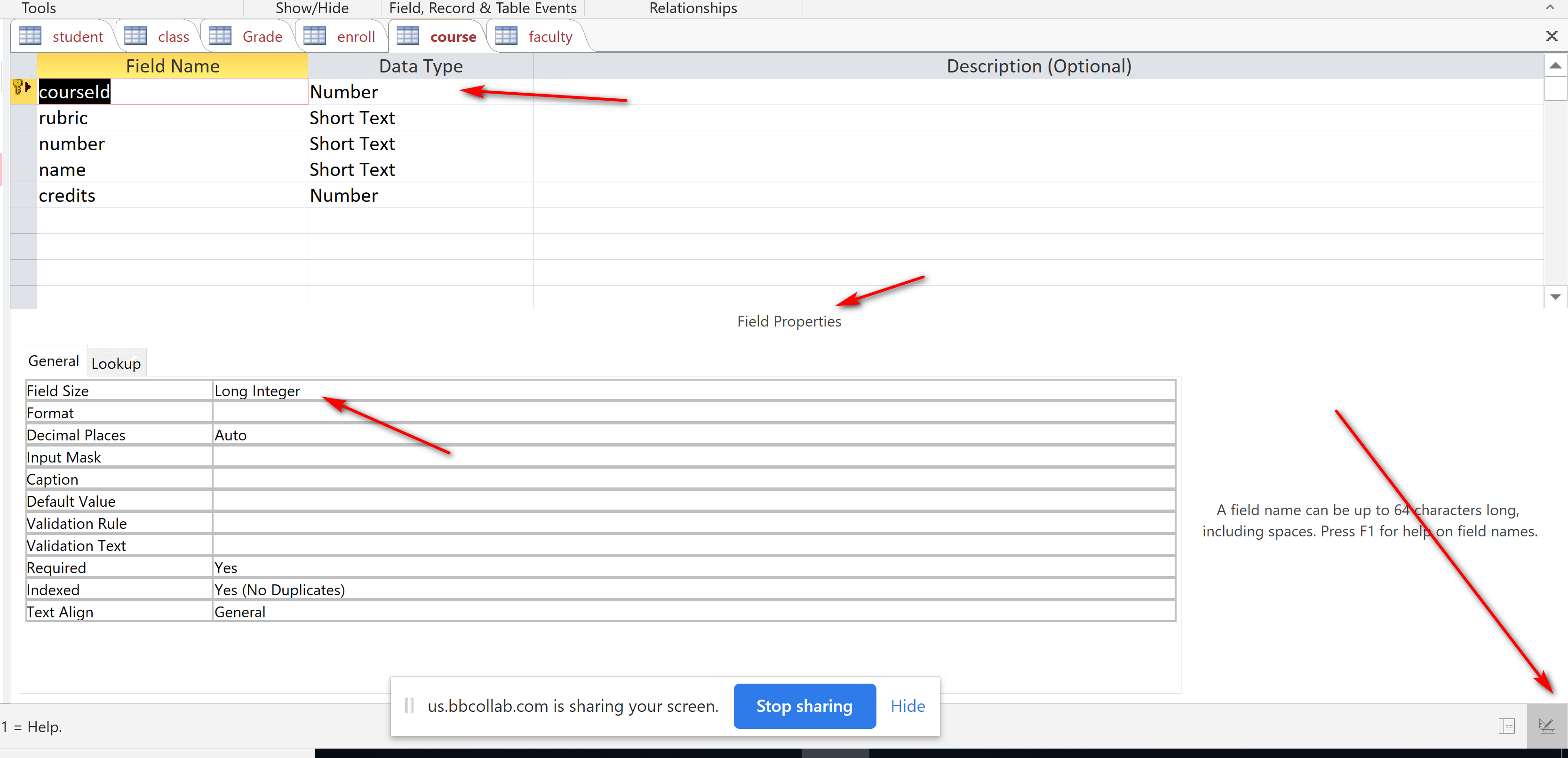


Views: most software provides multiple views.



FacId is a PK of faculty

FacId is not a PK of the table class.



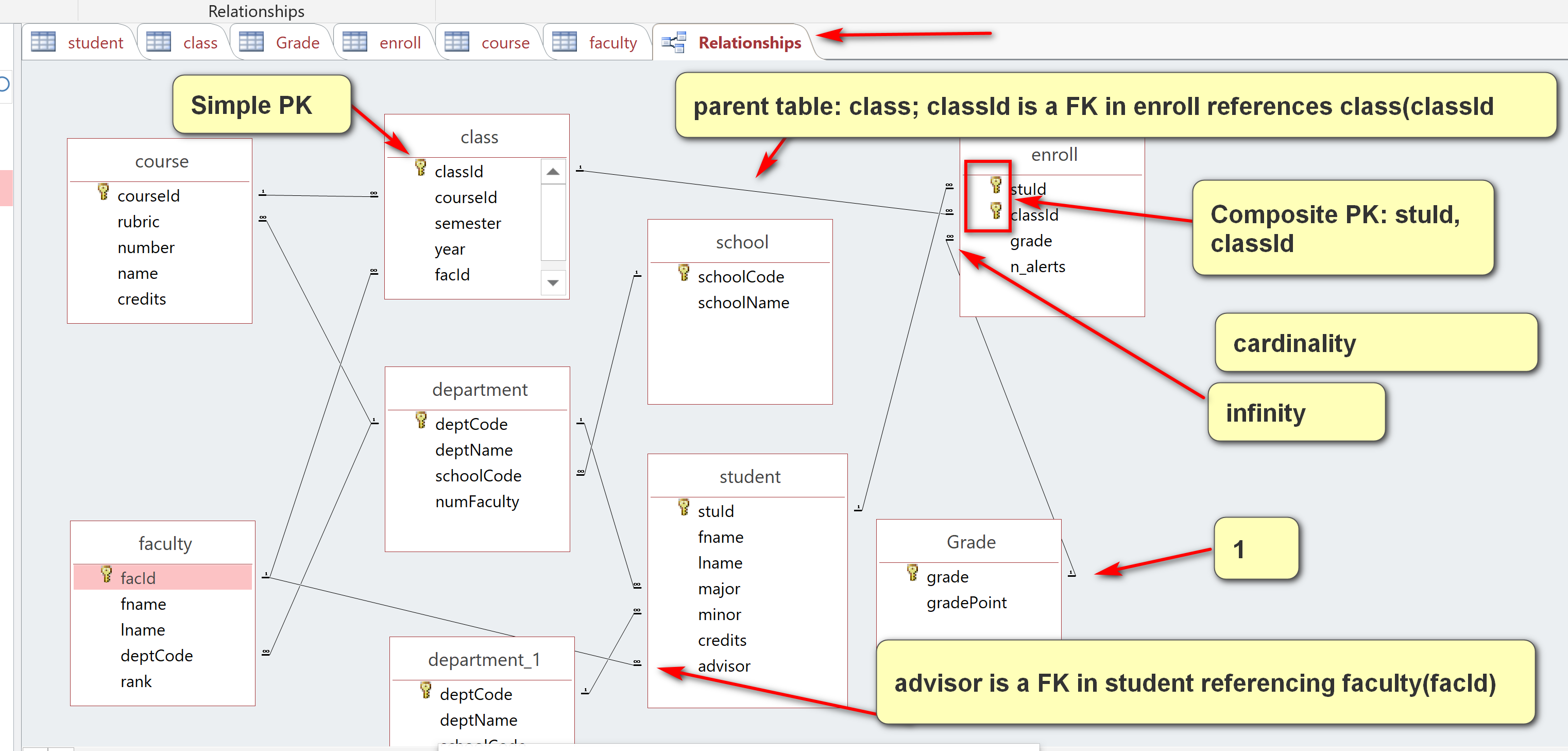
A table always have one and only one PK.

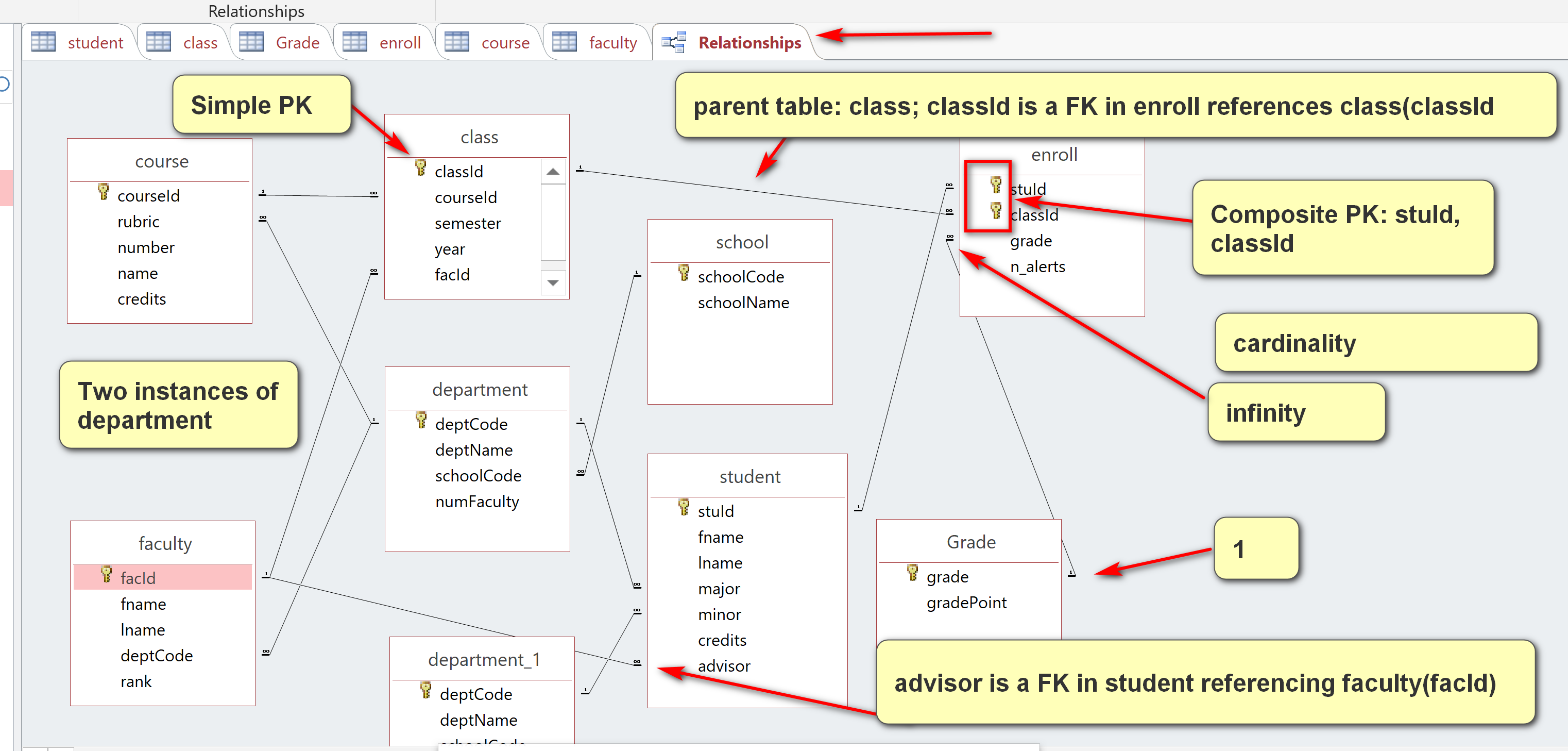
A table may have more than on unique identifiers (e.g. SSN, stuId): Candidate Keys (CK).

What is PK of the table Enroll? {StuId, ClassId}: composite key (more than one columns)

Enroll: AN enrollment of a student in a class (an offering of a course)

| **enroll** | | | |
| --- | --- | --- | --- |
| **stuId** | **classId** | **grade** | **n\_alerts** |
| 100000 | 10000 | A | 0 |
| 100000 | 10000 | B- | NO good: not allowed in RDB |
| 100000 | 10001 | A | 2 |
| 100000 | 10002 | B+ | 1 |
| 100000 | 10003 | C | 0 |
| 100000 | 10004 | A- | 1 |
| 100000 | 11001 | D | 4 |
| 100001 | 10000 |  |  |
| 100001 | 10001 | A- | 0 |
| 100002 | 10000 | B- | 3 |
| 100002 | 10002 | B+ | 2 |
| 100002 | 10003 | D | 4 |
| 100004 | 10003 | A | 0 |
| 100004 | 10004 | B+ |  |
| 100005 | 10003 |  |  |
| 100005 | 10004 | A- | 0 |
| 100005 | 10005 | A- | 0 |
| 100005 | 10006 | B+ |  |
| 100006 | 10004 | C+ |  |
| 100006 | 10005 | A |  |
| 100007 | 10007 | F | 4 |
| 100007 | 10008 | A- | 0 |
| 100008 | 10007 | C- | 0 |



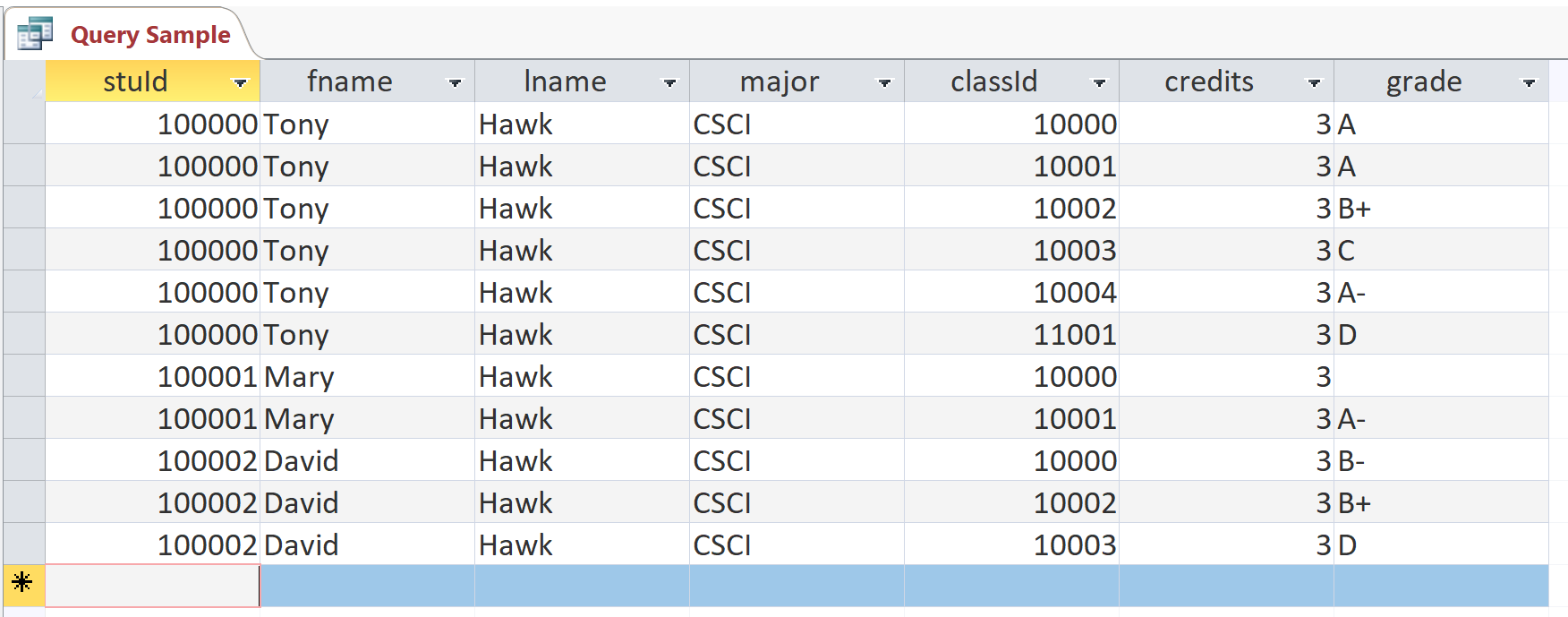


Q0. The suggested solution is a screenshot of the MS Access Query Design that produces the expected result tables.

Show the student id, fname, lname, class enrolled, credit of class and grade of class of every ‘CSCI’ major.

Analysis:

Expected result:



[1] Output:

Student: stuId, fname, lname, major

Enroll: classId, grade

Course: credits

[2] Source tables:

1. Student
2. Enroll
3. Course
4. Class: link between enroll and course (JOIN)

[3] Conditions:

1. Linking (JOIN): FK = PK of parent.
2. Problem: of every ‘CSCI’ major: major = ‘CSCI’

