

## Relationships in Access

We store data in the computer so that it can be retrieved when, how and where required. Think of the student information system at RCC. You as a student can look at your own information through web advisor anywhere in the world, whenever you need it. Other people at RCC, instructors and administrators, require different data from the student information system, and web advisor provides them the information they are authorized to access. The quality of the output depends on the quality of the input (GIGO – Garbage In, Garbage Out). We have to catch errors in the data as early on as possible; we are not going to be able to prevent errors. They will eventually be caught by the user. Example, if I give you a grade for this class as ‘Q’, if the error is not caught by the system, you will eventually come back to Admissions or to me to get it corrected. One of the methods for catching errors is Referential Integrity.

We store data in files. In Access, the extension of the file is accdb (access database). Files are made up of related records (information belonging to one occurrence of the entity that you are storing information about – examples: students, advisors, classes, books etc). All the data in that one record belongs to the specific student, it is related to that student. Each piece of data is called a field (examples: id number, name, address, phone number etc). So, files are made up of related records, records are made up of related fields, and eventually, fields are made up of bytes and bits (but we in this class are not interested in bytes and bits). So, what is a database? A database is a set of related files. Read on.

Why can we not have all data in one file? It would be too unwieldy. In a relational system, we create a file (table) for each entity. An entity can be a person, item, event etc. We then create a relationship between various related files. Relationships can be of three types: 1 to 1, 1 to many (1:M), and many to many (M:N). The 1 to 1 relationship has very limited applications; the 1 to many is the most common, and the many to many needs to be split into multiple 1 to many for implementation. An example of 1:M is: a counselor advises many students. An example of M:N is: students take many classes and classes are taken by many students. How to convert this to 1:M is a topic discussed in CSC/CIS2.

Why can we not keep information about students and their counselors in one file? Because it will result in duplication of data (redundancy), and data integrity problems if the duplicated data does not match. There are two entities here: students and counselors. Let us see the problem created by storing the information in one file, then look at how to store in two files and keep them related.

Student Id	Student Name	Major	GPA	Student Phone	Counselor Id	Counselor Name	Counselor Phone	Office Location
11	Lopez	CIS	3.2	555-1234	123	Plato	555-2000	BE300
22	Smith	PHY	3.3	555-1235	124	Vinci	555-3000	BE300
33	Doe	CHE	3.4	555-1236	125	Plato	555-4000	BE300
44	Jones	ENG	3.5	555-1237	126	Aristotle	555-5000	BE300

This is one file, with four records and nine fields. If we want to know Smith’s counselor’s phone number, we can easily find it: it is 555-3000. The problem with storing data about two entities (students and counselors) in one file is that all counselor information needs to be repeated. If a counselor is advising 40 students, his/her name and other information will have to be entered 40 times which is a waste of effort. Plus, the repeated data might not match: what is Plato’s phone number? If users find one error in the data, they might start to wonder about the reliability of the system and stop using the system. Let us split this into two files, one for each entity, and then relate the two files using Primary Keys and Foreign Keys.

PK (student file)					FK	PK (counselor file)			
Student Id	Student Name	Major	GPA	Student Phone	CId	Counselor Id	Counselor Name	Counselor Phone	Office Location
11	Lopez	CIS	3.2	555-1234	125	123	Plato	555-2000	BE300
22	Smith	PHY	3.3	555-1235	126	124	Vinci	555-3000	BE300
33	Doe	CHE	3.4	555-1236	123	125	Angelo	555-4000	BE300
44	Jones	ENG	3.5	555-1237	123	126	Aristotle	555-5000	BE300

The relationship here is one counselor advises many students. The PK of the one file (advisor) is stored as a FK in the many file. The values in the FK and the PK are the same. Counselor ID in the student file cannot be related to Counselor Name in the Counselor file.

To have a relationship, you need at least two tables in Access. Relationships can be created under Database Tools. Add both tables in the Relationships window, drag the PK of the one table to the FK of the many table and a link will be established. Double click the line and enforce Referential Integrity. What that does is prevent a student being assigned a non existent counselor number (like 999).

There are many other ways to validate data which we will learn about in this course.