B. Access Case: University Course Registrations

Purpose: To implement what you learned from the database tutorial without detailed instructions.

Please note:

- You will be creating a **NEW** database and no longer use the database that you created for the Access Tutorial.

- The instructions for this case are based on the assumption that you have worked through and understood the Access Tutorial: Employees & Compensation Case. As you work through the tables and queries for this Case, refer back to the various concepts that were introduced in your first Access Assignment.

Access Case (University Course Registrations):

B.1. Starting a New Database (yourLastNameRegistrations)

1. Refer to Guide #1– Starting and Saving Access Files for assistance on creating your second Access project file and saving on your OneDrive Cloud Storage area.

2. Name this new database ***yourLastNameRegistrations*** as in ***AppleRegistrations***. Access will add the file type of .accdb to your file name.

3. **NOTE**: Access will ask you for a file name and a location to save the file before you add any data to the database. Again check that you have selected your OneDrive area to store the file for current and future use.

B.2. Tables in the Registrations Database

Overview: This database is intended to help Small University manage course registrations. When complete it will consist of the following five tables

- Courses
- Students
- Instructors
- Sections
- Registrations

When complete your tables and relationships will look like Figure B-1. Read through the instructions to assist you to build these five tables and the relationships between them.

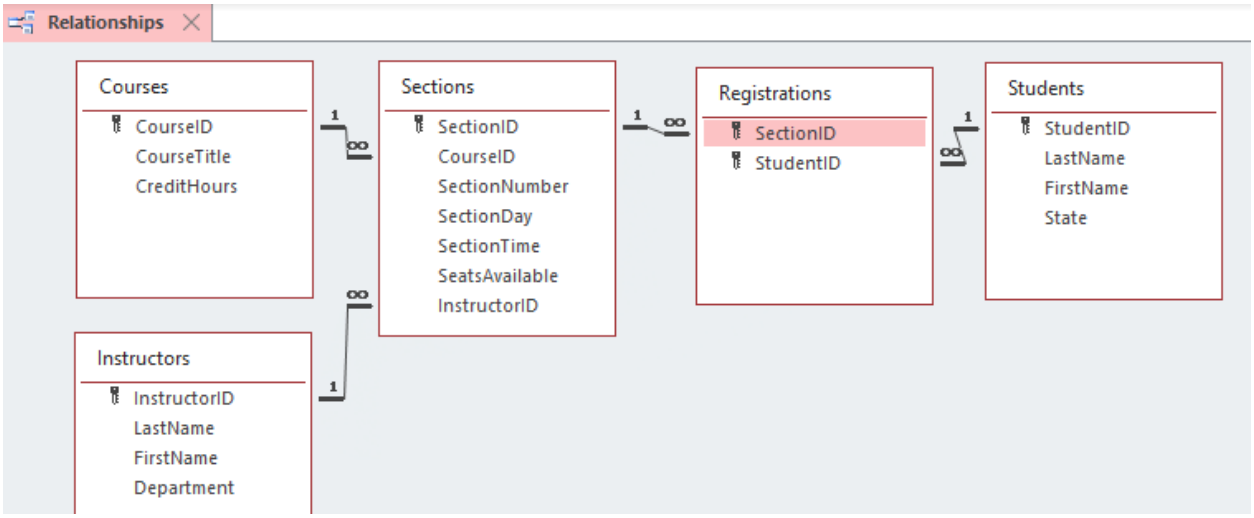


Figure B-1: Completed Registrations Database Structure

Important: When creating the five tables, please be sure to name the fields (columns) EXACTLY as indicated in the following instructions (also seen in Figure B-1). For example, note that the field names do not have spaces in them! In addition, do NOT ADD DATA until you have built all tables (steps 4-8), and established the relationships between the tables (step 9).

B.2.1 Tables and Fields in the Registrations Database

4. Create a table named **Courses** with the following three fields:

- **CourseID**
- **CourseTitle**
- **CreditHours**

    - For each field, choose an appropriate data type and length (for text fields) and set the primary key to a field that is unique (will not be the same entry in more than one row).
    - To assist you in the data type and length refer to Figure B-3 which displays the data that you will enter later into the Courses table.
    - Recall that most fields that will not be used for mathematical calculations or reflect currency will be short text or perhaps datetime.
    - For short text fields you will generally assign the size of the largest element if it is the primary key. If it is a descriptive field (CourseTitle) you may want add 20 or so additional characters and then round up.
    - For this first table, we recommend that you make CourseID short text, length 8; CourseTitle short text, length 70; and CreditHours a number, Long Integer.
    - Don't forget to indicate which field is the primary key
    - *If you need assistance on building this first table, refer back to Access Tutorial on Employees, Section A.2.*

Hint: for all of the tables in this database, we recommend that the ID column (CourseID, StudentID, InstructorID, SectionID) all be declared as Short Text. Why? We will never do math with these columns and also you might need to put a character, such as SectionID = 01A, and if you declared the field type as an integer it would not accept a character such as A.

5. Create a table named **Students** with the following four fields

- **StudentID**
- **LastName**
- **FirstName**
- **State**

    - For each field, choose an appropriate data type and length (for text fields) and set the primary key to a field that is unique. (See Figure B-4 for sample data). Hint all will be Short Text for this table.
    - Use the same field length for FirstName and LastName that you used in your Access Tutorial Project
    - Student ID and State field length should be set to the longest entry for each of those respective columns

6. Create a table named **Instructors** with the following four fields:

- **InstructorID**
- **LastName**
- **FirstName**
- **Department**

    - For each field, choose an appropriate data type and length (for text fields) and set the primary key to a field that is unique. Refer to Figure B-5 for assistance for length.
    - Use the field length for FirstName and LastName from your Access Tutorial Project

- InstructorID and Department field length should be set to the longest entry for each of those respective columns

7. Create a table named **Sections** (Figure B-6) with the following six fields:

- **SectionID**
- **CourseID**
- **SectionNumber**
- **SectionDay**
- **SectionTime**
- **SeatsAvailable**
- **InstructorID**

  - For each field, choose an appropriate data type and length (for text fields), and set the primary key to a field that is unique.
  - Hint: Some of the fields in the table Sections are foreign keys. This means that they correspond with fields in other tables that are primary keys, and therefore you will establish links between these tables. When building the fields in this table ensure these common fields have the same field types and sizes as the previously built table.
  - Hint: SectionTime should be datetime and then select the appropriate format to match those in the sample data for this table

8. Create a table named **Registrations** (Figure B-7) with the following two fields:

- **SectionID**
- **StudentID**

  - Hint: This table is an association table that links two other tables that form a many-to-many relationship. The links are established with the use of foreign key fields, and just like for the table Sections, the foreign keys need to be of the same data types (and lengths for text fields) as the primary key fields in the corresponding tables. Also note that the primary key in the table Registrations must be a combination of two fields, StudentID and SectionID; can you explain why? (See Access Tutorial, A.2.2 for how to create a compound key made up of more than one field).

B.2.2 Building Relationships Between the Tables

9. Before proceeding to entering the data, **build relationships** between the tables. Even though not strictly required, building relationships now is strongly recommended because it helps to reduce errors when entering data and will also be very helpful when setting up the queries later on.

   Please, review the Access Tutorial (A.2.5) if you need help with setting up relationships. After you are done, each table has to have at least one relationship, i.e., line connecting it to another table. Remember, relationships exist between the pairs of primary keys and corresponding foreign keys. You should select Enforce Referential Integrity for all relationships. Why? Compare your results with Figure B-2, and don't forget to save when prompted.

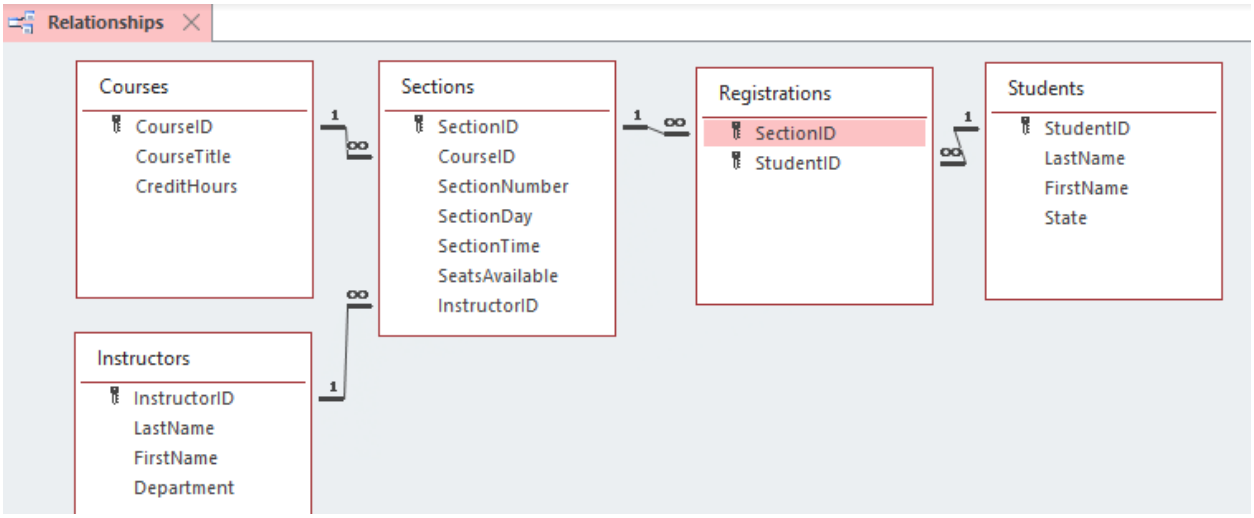


Figure B-2: Completed Relationships between tables in the Registrations database

B.2.4 Entering Data into your Tables

10. Now that you have built the tables and relationships, please proceed to **enter the data**. Data for each table is depicted in Figures B-3 through B-8 that follow. Remember, since you have enforced referential integrity, the order of entering data into specific tables makes a difference. Records must exist in the primary tables before a foreign key value can be used in other tables, thus enter in the order as shown in the following Figures.

Sections | Courses

CourseID	CourseTitle	CreditHours
ACG 201	Financial Accounting	3
ACG 201L	Accounting Lab	1
MIS 213	Intro to Management Information Systems	3
MIS 216	Intro to Business Development Programming	1
PED 101	Beginning Yoga	2

Figure B-3: Table Courses (3 fields, 5 records)

Query1 | Students

StudentID	LastName	FirstName	State
2011	Controller	Kelly	SC
2020	Meyer	Maya	NC
2021	Diaz	Cameron	NC
2055	Parker	Paul	NC
2222	Apar	Santiago	NC
2300	Jann	Mariah	VA
2301	Salinas	Dana	NC
2302	Merenda	Emma	NC
3030	Short	Izzy	TX
3333	Johnson	Gianni	DC

Figure B-4: Table Students (4 fields, 10 records)

Instructors | Sections

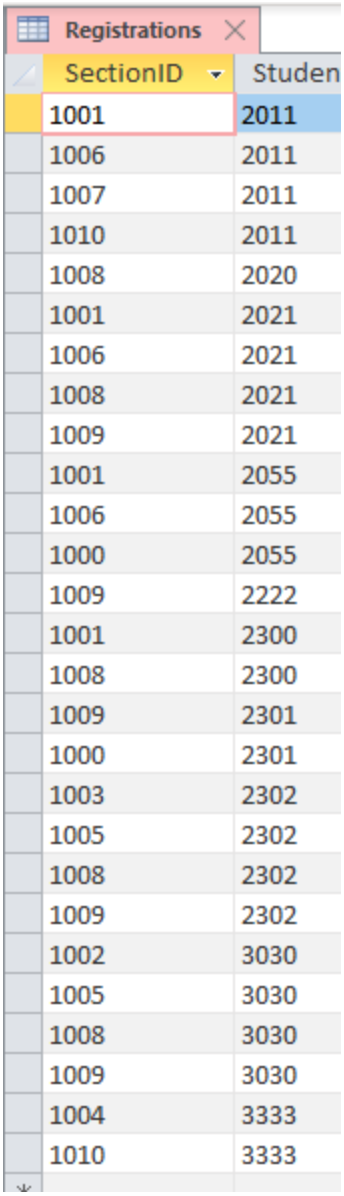
InstructorID	LastName	FirstName	Department
110	Motts	David	Accounting
111	Sawyer	Rebecca	Accounting
130	Walker	Carl	Sports
131	Smith	Kit	Sports
140	Gebauer	Judith	Info Systems
141	Matthews	Kevin	Info Systems
160	Clark	Ulku	Info Systems

Figure B-5: Table Instructors (4 fields, 7 records)

Sections

SectionID	CourseID	SectionNum	SectionDay	SectionTime	SeatsAvailat	InstructorID
1000	MIS 216	001	F	11:00 AM	15	141
1001	MIS 213	001	TR	3:00 PM	5	140
1002	MIS 213	002	MWF	1:00 PM	10	141
1003	MIS 213	003	TR	1:30 PM	5	160
1004	ACG 201	001	MW	9:00 AM	3	110
1005	ACG 201	002	MWF	10:00 AM	4	111
1006	ACG 201	003	TR	8:00 AM	10	110
1007	ACG 201L	001	Online		10	111
1008	ACG 201L	002	Online		5	111
1009	PED 101	001	TR	9:00 AM	5	130
1010	PED 101	002	TR	10:00 AM	8	131

Figure B-6: Table Sections (6 fields, 11 records)



Registrations

SectionID	Studen
1001	2011
1006	2011
1007	2011
1010	2011
1008	2020
1001	2021
1006	2021
1008	2021
1009	2021
1001	2055
1006	2055
1000	2055
1009	2222
1001	2300
1008	2300
1009	2301
1000	2301
1003	2302
1005	2302
1008	2302
1009	2302
1002	3030
1005	3030
1008	3030
1009	3030
1004	3333
1010	3333

Figure B-7: Table Registrations (2 fields, 27 records)

Comment: your data may appear in a different order than appears in Figure B-7. Do not worry as we did not indicate a specific order. To be concerned that you have 27 records and 2 fields.

B.3.0 Creating Forms

B.3.1 Instructors and their Sections

11. Create a form with a sub-form that shows the information on the ***instructors*** in the main form and the ***sections*** that each instructor teaches in the sub-form. See the Database Tutorial (Section A.5.1) for help. If you have built your relationships correctly, the sub-form will show up automatically when you create the main form.

    Compare your result with Figure B-8 and save this form as **InstructorsSections**.

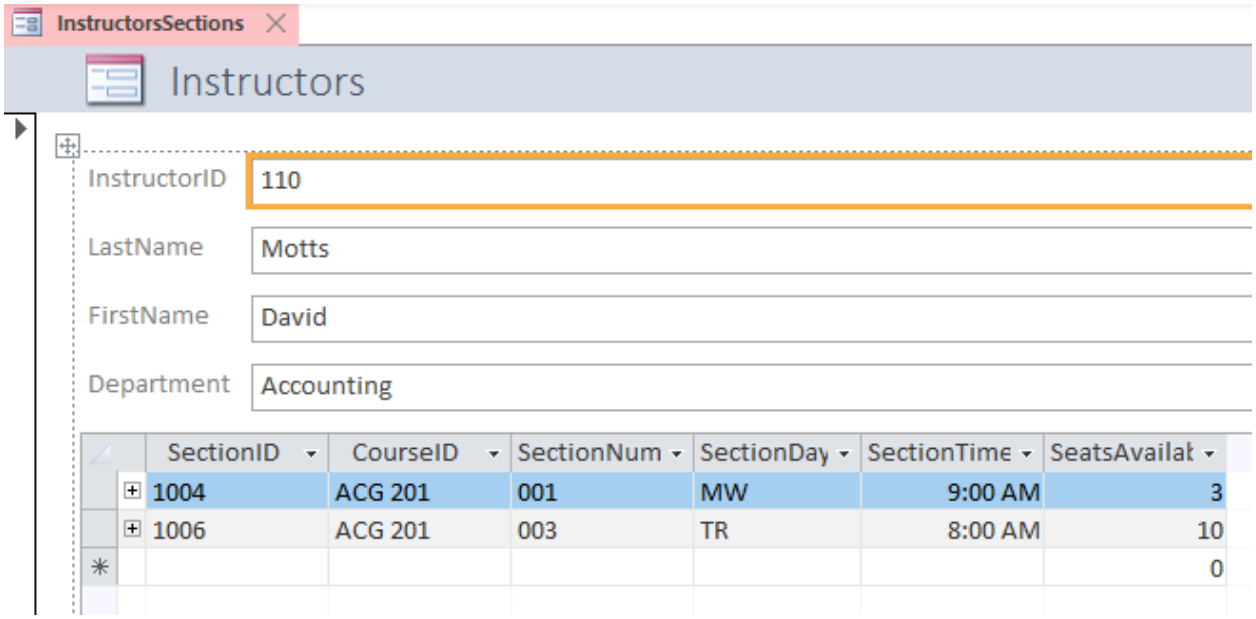


Figure B-8: Form InstructorsSections

B.3.2 Sections and their Registrations

12. Create a form with a sub-form that shows information on the ***sections*** in the main form and ***registrations*** for each section in the sub-form.

    Compare your result with Figure B-9 and save this form as **SectionsRegistrations**. Scroll through the sections (bottom arrows) and you will see all sections and all students enrolled.

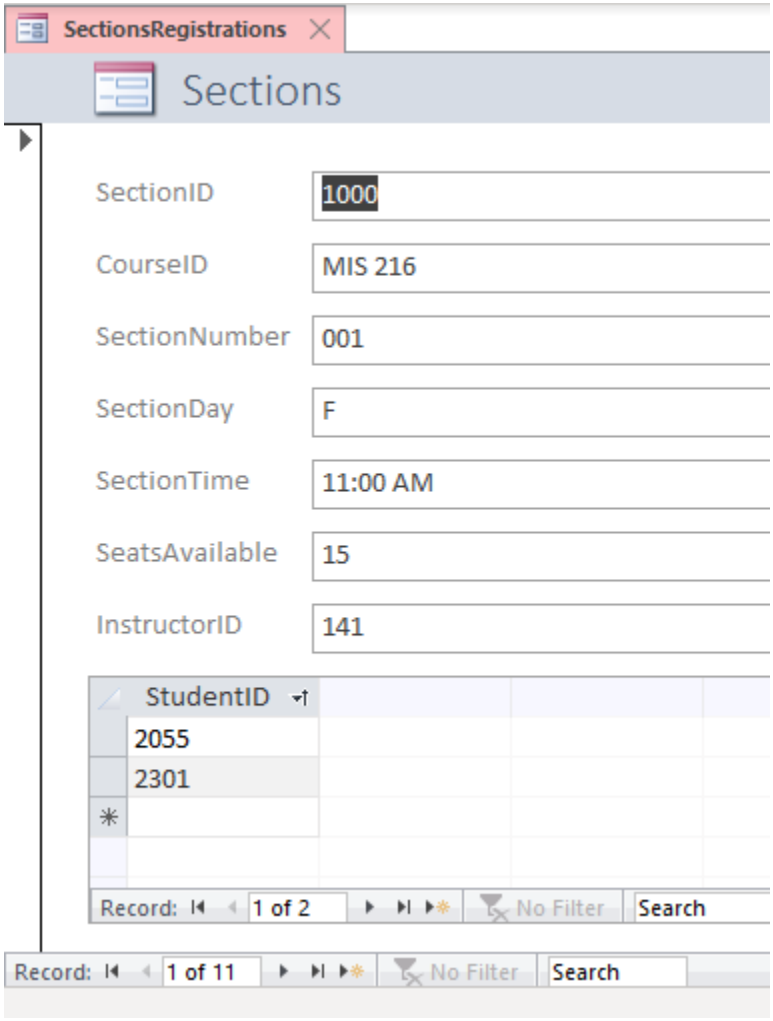


Figure B-9: Form SectionsRegistrations

13. **Bonus question (Bonus-B-1)** *(answer only if instructed by your course instructor)*:
    a. What is the main purpose of the forms and sub-forms?

B.4.0 Queries

***Common Query Error**: If you have more records than shown in the result figures, make sure you look at the query design. If you have any tables that have no links (relationships) to the other tables in the query you might need an association table to link the tables together. In this database it usually implies you are missing the association table named: Registrations. If needed, add this table to your design.*

B.4.1 Select Merenda's Registrations

MerendaRegistrations: What courses has Merenda registered to take?

14. Create a query that displays CourseID, SectionDay, SectionTime and the instructor's LastName of all courses for which the student with the last name of Merenda is registered. Sort on CourseID and rename the column of the instructor's LastName to read Instructor.

    HINTS: In your query design (upper part of the query design window in Access) you need to add four tables, even though you don't actually select fields from all four tables. Can you explain why? In addition you will have a column you will not SHOW. See A.3.2 if you need help with the criteria

    Compare your results with Figure B-10 and save your query as **MerendaRegistrations**.

MerendaRegistrations

CourseID	SectionDay	SectionTime	Instructor
ACG 201	MWF	10:00 AM	Sawyer
ACG 201L	Online		Sawyer
MIS 213	TR	1:30 PM	Clark
PED 101	TR	9:00 AM	Walker

Figure B-10 Results of Query MerendaRegistrations (show 4 fields, 4 records)

B.4.2 MIS213Registrations

MIS213Registrations: What students are registered for the course MIS 213?

15. Create a query that displays all students that are registered for the course with ID MIS 213. Include CourseID, SectionNumber, as well as FirstName and LastName for instructors and for students.

16. Rename the Instructor and Student Columns as shown in Figure B11. Compare your results with Figure B-11 and save your query as MIS213Registrations. Refer back to Figure A-46 for help with renaming columns.

MIS213Registrations

CourseID	SectionN	InstructorFN	InstructorLN	StudentFN	StudentLN
MIS 213	001	Judith	Gebauer	Kelly	Controller
MIS 213	001	Judith	Gebauer	Cameron	Diaz
MIS 213	001	Judith	Gebauer	Paul	Parker
MIS 213	001	Judith	Gebauer	Mariah	Jann
MIS 213	002	Kevin	Matthews	Izzy	Short
MIS 213	003	Ulku	Clark	Emma	Merenda

Figure B-11 Results of Query MIS213Registrations (show 6 fields, 6 records)

B.4.3 Calculate In State Tuition

CalcInStateTuition: What is the tuition for students from North Carolina?

17. Create a query that calculates the tuition for all students that are from North Carolina as $100 per credit hour for each registered course. Include students' FirstName, LastName, CourseID, CreditHours, and Tuition as a calculated field. Sort by Student Last Name.

    Compare your results with Figure B-11; don't forget to format the calculated field Tuition as currency. Save your query as **CalcInStateTuition**. See A.4.4 for help with the calculated column.

CalcInStateTuition

FirstName	LastName	CourseID	CreditHours	Tuition
Santiago	Apar	PED 101	2	$200.00
Cameron	Diaz	PED 101	2	$200.00
Cameron	Diaz	ACG 201L	1	$100.00
Cameron	Diaz	ACG 201	3	$300.00
Cameron	Diaz	MIS 213	3	$300.00
Emma	Merenda	PED 101	2	$200.00
Emma	Merenda	ACG 201L	1	$100.00
Emma	Merenda	ACG 201	3	$300.00
Emma	Merenda	MIS 213	3	$300.00
Maya	Meyer	ACG 201L	1	$100.00
Paul	Parker	MIS 216	1	$100.00
Paul	Parker	ACG 201	3	$300.00
Paul	Parker	MIS 213	3	$300.00
Dana	Salinas	MIS 216	1	$100.00
Dana	Salinas	PED 101	2	$200.00

Figure B-12: Results of Query CalcInStateTuition (show 5 fields, 15 records)

B.4.4 Calculate Out of State Tuition

CalcOutOfStateTuition: What is the tuition for students from outside of North Carolina?

18. Create a query that calculates the tuition for all students that are from outside of North Carolina as $200 per credit hour for each registered course. Include students' FirstName, LastName, CourseID, CreditHours, and Tuition as a calculated field. Sort on LastName.

    HINT: as this query is very close in design to the previous query, you can COPY the prior query and make the few changes needed. Close the InState Query, then in the left column, right click and copy the prior query and rename the copy as CalcOutOfStateTuition. Finally make the changes to the calculated column as well as the criteria and you are done.

    Compare your results with Figure B-13; don't forget to format the calculated field Tuition as currency.

19. Save your query as CalcOutOfStateTuition.

CalcOutOfStateTuition

FirstName	LastName	CourseID	CreditHours	Tuition
Kelly	Controller	PED 101	2	$400.00
Kelly	Controller	ACG 201L	1	$200.00
Kelly	Controller	ACG 201	3	$600.00
Kelly	Controller	MIS 213	3	$600.00
Mariah	Jann	ACG 201L	1	$200.00
Mariah	Jann	MIS 213	3	$600.00
Gianni	Johnson	PED 101	2	$400.00
Gianni	Johnson	ACG 201	3	$600.00
Izzy	Short	PED 101	2	$400.00
Izzy	Short	ACG 201L	1	$200.00
Izzy	Short	ACG 201	3	$600.00
Izzy	Short	MIS 213	3	$600.00

Figure B-13: Results of query CalcOutOfStateTuition (show 5 fields, 12 records)

B.4.5 Number of Students by Instructor

NumberStudents: How many students are registered with each instructor?

20. Create a query that determines how many students are registered with each instructor for all of their courses and sections. Include InstructorID, FirstName, LastName and Total Number of Students. Make sure to name the fourth column TotalStudents.

    Hint: See A.4.6 for help to determine the Total Students Column

    Compare your results with Figure B-14, and save your query as **NumberStudents**.

NumberStudents

InstructorID	FirstName	LastName	TotalStudents
160	Ulku	Clark	1
140	Judith	Gebauer	4
141	Kevin	Matthews	3
110	David	Motts	4
111	Rebecca	Sawyer	8
131	Kit	Smith	2
130	Carl	Walker	5

Figure B–14: Results of Query NumberStudents (show 4 fields, 7 records)

B.4.6 Seats Remaining in each Section

SeatsRemaining: How many seats are remaining in each section?

21. Create a query that determines how many seats are remaining in each of the sections that are offered in the term. Include CourseID, SectionNumber, SeatsAvailable, Registrations, and SeatsRemaining as columns. Please note that this is a **totals query that also contains a calculated field** (=expression). In order to get this to work, you need to change the Total option for the calculated field SeatsRemaining to be "Expression" from the dropdown selection.

    Hints: The easiest way to build this is to not try to build the entire query on your first pass.
    - We recommend that you first build a query to display the first three columns (you will only need one table at this point) shown in Figure B-15.
    - If that is correct add in another table to your design to show the registrations by course (you will need to Group and Count), also rename the column heading to Enrolled.
    - Add a calculated column to subtract enrolled seats from seats available (don't forget you will need []'s) and also change the Total row to Expression.
    - Finally sort by CourseID
    - Modify the headings as shown

    Compare your results with Figure B-15, and make sure that all of your columns are named exactly as shown in the Figure. Save your query as SeatsRemaining.

SeatsRemaining

CourseID	SectionNum	SeatsAvailable	Enrolled	SeatsRemaining
ACG 201	001	3	1	2
ACG 201	002	4	2	2
ACG 201	003	10	3	7
ACG 201L	001	10	1	9
ACG 201L	002	5	5	0
MIS 213	001	5	4	1
MIS 213	002	10	1	9
MIS 213	003	5	1	4
MIS 216	001	15	2	13
PED 101	001	5	5	0
PED 101	002	8	2	6

Figure B–15: Results of Query SeatsRemaining (show 5 fields, 11 records)

B.5. Reports for the Registrations Database

B.5.1 List all courses offered

Report CourseList: List all courses that are offered by this university

22. Generate a report that shows all of the courses offered at this university. Format it to look like Figure B-16 (including the change in Title) and save it as **CourseList**. Hint: this is an easy Report, the key is to click once on Courses in your Table list in the left column and then Create Report. Don't forget to change the column headings to be more readable.

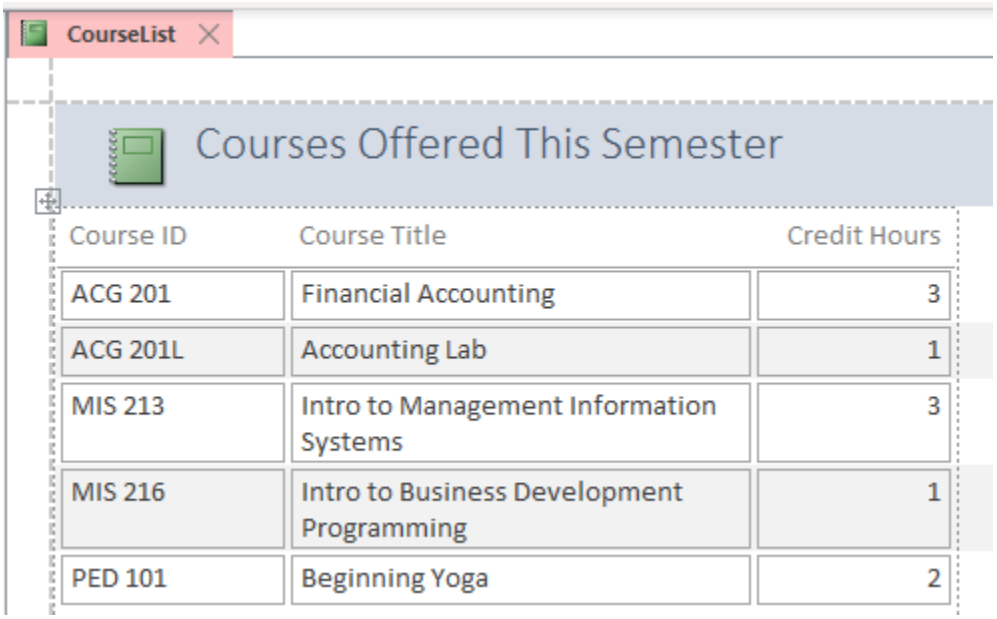


Figure B-16: Report CourseList

B.5.2 In State Tuition Totals

Report: InStateTuition: Summarize the tuition for all North Carolina Students

23. Use the Query CalcInStateTuition to generate a report that summarizes the tuition owed by all students from North Carolina.

    HINTS: the easiest way to build this report is to use the Report Wizard as described in the Access Tutorial, Section: A.6.3. This will provide an easy to group (by Last Name) and then sort by Course ID.

24. Modify the report headings to make them more user friendly and name the report **InStateTuition**. In addition, add your name to the heading of the report.

25. Optional formatting (based on instructor), remove the alternating gray background, revise the Total background to be Theme 1.

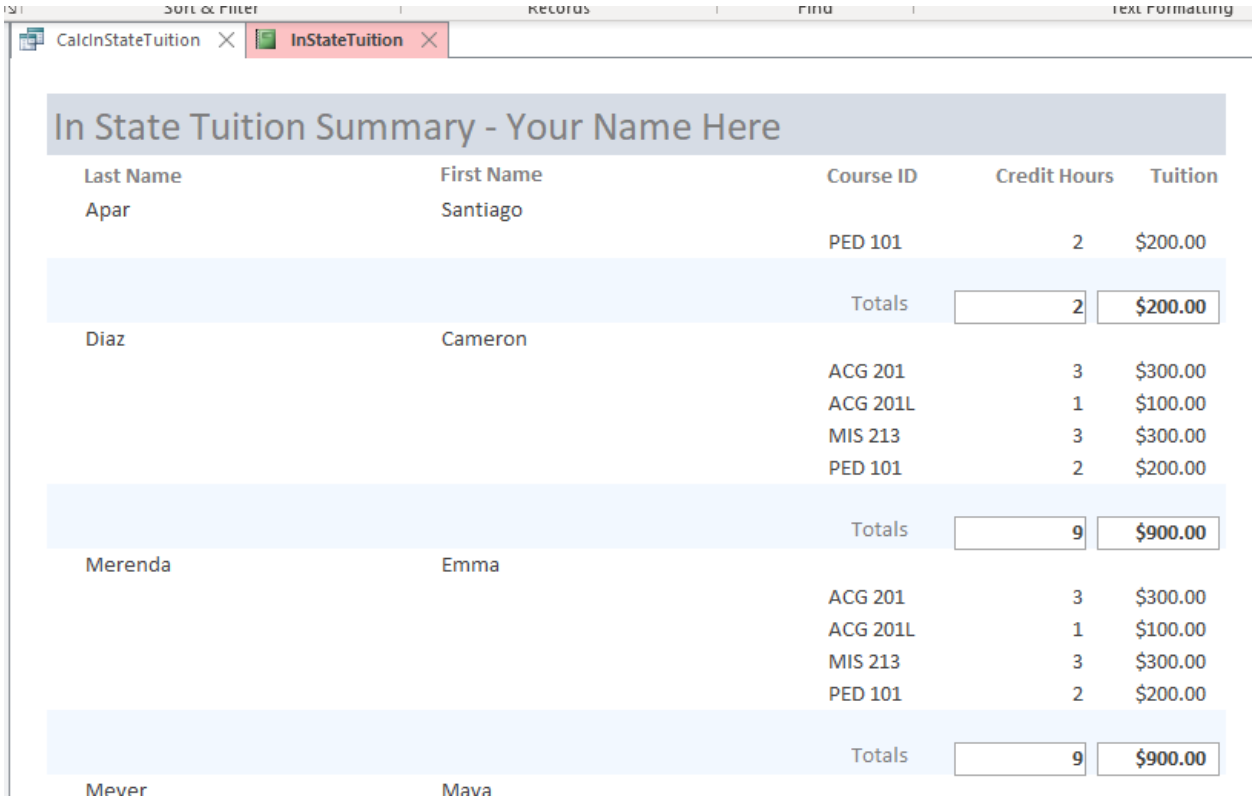


Figure B-17: Report InStateTuition (figure presents the top of the report only)

B.5.3 Student Schedules

Report :StudentSchedulesReport. Display the schedules for all students registered for the current semester.

26. For this report, you need to create a query first. Name this new query **StudentSchedules**

    For the query, display LastName, FirstName, CourseID, CourseTitle, SectionDay, SectionTime, the LastName of the Instructor and CreditHours for all registered students. Sort ascending by students' LastName, FirstName, and by CourseID. Save the query as **StudentSchedules**.

    Again, use the Report Wizard to assist you in building the report shown in Figure B-18. Name the report **StudentSchedulesReport**.

27. Formatting: add your name to the header row. Bold the Column headings and make them more readable.

28. *Optional Formatting based on instructor. Eliminate the alternating gray backgrounds (See A.6.4 if you need help) as well as provide a gray background for all the student names.*

Courses By Student Your Last Name

Last Name	First Name	CourseID	Course Title	Days	Time	Instructor	Hours
Apar	Santiago						
		PED 101	Beginning Yoga	TR	9:00 AM	Walker	2
							2
Controller	Kelly						
		ACG 201	Financial Accounting	TR	8:00 AM	Motts	3
		ACG 201L	Accounting Lab	Online		Sawyer	1
		MIS 213	Intro to Management Information S	TR	3:00 PM	Gebauer	3
		PED 101	Beginning Yoga	TR	10:00 AM	Smith	2
							9

Figure B-18: Report StudentSchedulesReport (top portion only and shown with extra formatting)

B.6 Uploading your file to Entropy for Grading

29. Ensure you have closed your file before trying to upload your database

30. Go to Entropy and upload the file.

31. If you need help, refer back to Access Project A, page A-50.