

Case Problem 4

Data File needed for this Case Problem: **AppTrail.accdb**

Hike Appalachia Molly and Bailey Johnson own Hike Appalachia, a business in which they guide clients on hikes in the Blue Ridge Mountains of North Carolina. They advertise in local and regional outdoor magazines and field requests from people all around the region. Molly and Bailey have created an Access database for their business and now want you to create several queries and modify the table design. To do so, you'll complete the following steps:

1. Open the **AppTrail** database located in the Access2 > Case4 folder provided with your Data Files.
2. Modify the first record in the tblHiker table datasheet by changing the Hiker First Name and Hiker Last Name column values to your first and last names, and then close the table.
3. Create a query to find all records in the tblHiker table in which the HikerLast field value starts with the letter S, sorted in ascending order by HikerLast. Display all fields except the HikerID field in the query recordset. Save the query as **qryHikerLastNameS**, run the query, and then close it.
4. Create a query to find all records in the tblTour table where the tour type is not hiking. Name this query **qryNonHikingTours**. Display all fields, sorting by TourName. Save and run the query, and then close it.
5. Create a query to find all records in the tblHiker table in which the State field value is NC, SC, or GA and display all fields from the tblHiker table in the query recordset, sorted by HikerLast in ascending order. Save the query as **qrySelectedStates**, run the query, and then close it.
6. Create a query to select all records from the tblTour table with a price per person of \$100 or less, where the TourType is Hiking. Display all fields in the query recordset, sorted by PricePerPerson in descending order. Save the query as **qryInexpensiveHikingTours**, run the query, and then close it.
7. Create a parameter query to select the tblTour table records for a TourType field phrase value that the user specifies. If the user doesn't enter a phrase field value, select all records from the table. Display all fields from the tblTour table in the query recordset, and sort in ascending order by TourName. Save the query as **qryTourParameter**. Run the query and enter no value as the phrase field value, and then run the query again and enter **Van** as the TourType phrase field value. Close the query.
8. Create a query that contains all records from the tblHiker table and all matching records from the tblReservation table. Display all fields from the tblHiker table and all fields except HikerID from the tblReservation table. Save the query as **qryHikersAndReservations**, run the query, and then close it.
9. **Explore** Create a crosstab query based on the qryHikersAndReservations query. Use the TourID field values for the row headings, the HikerID field values for the column headings, and the sum of the People field as the summarized value, and include row sums. Save the query as **qryReservationsCrosstab**, resize the columns in the query recordset to their best fit, and then save and close the query.
10. **Explore** Create a find duplicates query based on the qryHikersAndReservations query. Select HikerID as the field that might contain duplicates, and select the fields HikerFirst, HikerLast, TourID, TourDate, and People in the table as additional fields in the query recordset. Save the query as **qryMultipleReservations**, run the query, and then close it.
11. Create a find unmatched query that finds all records in the tblHiker table for which there is no matching record in the tblReservation table. Display the HikerFirst, HikerLast, City, State, and Phone fields from the tblHiker table in the query recordset. Save the query as **qryHikersWithoutReservations**, run the query, and then close it.