CONCEPTS REVIEW

Label each of the elements of the Select Query window shown in Figure 3-19.

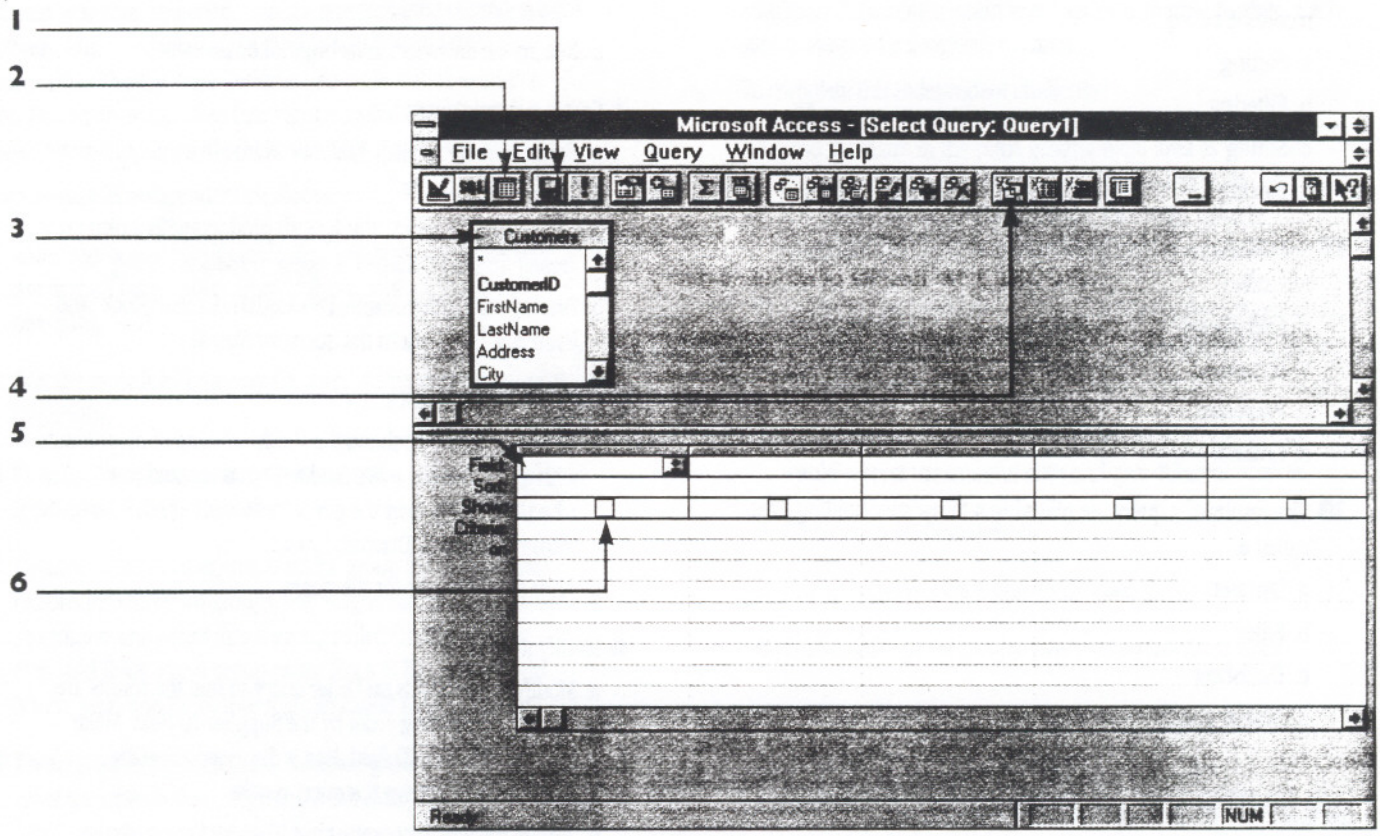

FIGURE 3-19

Match each button with its correct description.

7 
8 
9 
10 
11 
12 

a. Open Database
b. Print
c. Show All Records
d. Edit Filter/Sort
e. Datasheet View
f. Save

Select the best answer from the list of choices.

13 You can move to the first blank record by pressing
a. [Ctrl][*]
b. [Home][End]
c. [Ctrl][=]
d. [Home]

14 The button that sorts a table from A to Z is
a.
b.
c.
d.

15 Which button searches for text?

a. 

b. 

c. 

d. 

16 The method of organizing records to display only qualifying records is called

a. Sorting

b. Filtering

c. Sifting

d. Flexing

17 Specifications that define which records will be displayed are called

a. Extractions

b. Gauges

c. Properties

d. Criteria

18 The results of a query are placed in a temporary holding area called a

a. Dynaset

b. Filter

c. Datasheet

d. Design query

APPLICATIONS REVIEW

1 Find records.

a. Start Access and insert your Student Disk in the disk drive.

b. Open the database BICYCLE.MDB from your Student Disk.

c. Use the Find button to locate all records in all fields that match any part of a field and contain the text "bar."

d. How many occurrences are there?

2 Sort a table.

a. Sort the Products table in ascending order using the Product ID field containing the five-digit numbers. Print the datasheet.

b. Return the datasheet to its original order by clicking the Show All Records button.

c. Create and print a list of products in descending order by Units In Stock.

d. Return the datasheet to its original order.

3 Filter a table.

a. Create a filter that shows all products on order. A product on order has a value larger than 0 in the Units On Order field. (*Hint:* Set the criteria in the filter grid to ">0" for Units On Order.)

b. Print the datasheet, showing only the Product ID field data containing the five-digit numbers and the product name for the items on order.

c. Return the datasheet to its original order.

4 Create a simple query.

a. Create a query for Step 3a above. Name this query "Products on Order."

b. Create another query that lists all products with a unit price greater than $50. Name this query "Products costing >$50."

c. Display the ProductName, ProductID1, UnitsInStock, and UnitPrice field data in the query for Step 4b.

5 Create a complex query.

a. Create a query that shows the Product Name of all products with a DiscoStatus = Yes field AND that are sold by the pair (the Units field). Sort the query in ascending order by units. Name this query "Discoed, by Unit."

b. Print the datasheet for the query.

6 Modify a query.

a. Modify the Products on Order query so that the results are sorted in ascending order by the Supplier ID field. Make sure the Supplier ID field data is displayed. Save the modified query. Print the query results.

b. Modify the Products costing >$50 query to include the following fields: Product ID containing the five-digit number, Reorder Level, and Reorder Amount. Save the modified query. Print the query results.

c. Modify the Discoed, by Unit query so that the results are sorted in descending order by units, and so that the data for the Unit Price field is displayed. Save the modified query. Print the query results.

d. Close the file and exit Access.

INDEPENDENT CHALLENGE 1

The Melodies Music Store has hired you as the customer service manager. The store attracts a variety of customers with different tastes in music. To provide your customers with lists of the types of music available, you need to create queries in the store's music database, which is contained in the file MELODIES.MDB on your Student Disk. The database includes one table, called Available titles. This table divides data into eight musical classifications: Classic, Alternative, Rock, World Music, Jazz, Blues, Pop, and Metal.

To complete this independent challenge:

1. Using the Available titles table, find out how many records are in the Classic group. Should this classification be its own subgroup?
2. Sort the records by Category ID, then by Product Name. Print the results.
3. Create a filter that examines records with a Serial Number lower than 400000. Print this list.
4. Because musical categories overlap, group the eight classifications into three subgroups. You might, for example, include Alternative and Metal into a group called Grunge; Classic, Pop, and Rock into a group called Rock N Roll; and World Music, Jazz, and Blues into a group called Easy Listening.
5. Create a query for each of the three subgroups that finds all of the titles in that subgroup. Each query must use an "Or" specification in the query grid. Name each query for its classification.
6. Query the Available titles table using each query created in Step 5, and print the results of each query.
7. Modify one of the subgroup queries to include a musical classification already included in another subgroup. For example, you could include Pop in Easy Listening as well as Rock N Roll.
8. Print the results of the modified query.
9. Submit all printouts.

INDEPENDENT CHALLENGE 2

You work in the US Census Office for your city. Using the database STATS-1.MDB from your Student Disk, create several queries that examine the data in the Statistical Data table. The records in the Statistical Data table contain marriage information by state. Each state is assigned a geographical area.

To complete this independent challenge:

1. Find the states in the same geographical area as your state. For example, if your state is Utah, other states in the Mountain Region are New Mexico, Colorado, Nevada, Montana, Arizona, Idaho, and Wyoming.
2. Create a query that selects records in your geographical area and sorts them in ascending order by state. Display the State and Marriages fields.
3. On paper, write down at least three additional queries that would extract meaningful data and give different outputs.
4. Create each of these queries.
5. Print a sample of each query's dynaset.
6. Submit each of the samples with the handwritten work.