FIT 100

Lab Activity 14: More Microsoft Access and Relational Databases

Creating Views with SQL

Winter 2002

Introduction:
Today's lab looks at an additional component of the Data Services layer: formulating queries that select, project and join up data from different tables.

Access uses a GUI called the Query-By-Example so users can build queries visually. In the background, Access translates the visually represented query into SQL (Structured Query Language), the standardized language used to communicate with a database. This SQL View can also be used to build queries using straight SQL syntax.

> **Data Services**
> The main unit of data services (data storage) is a *Table.* Data storage allows for persistent data (the data stays even when the power goes off). A table is a collection of data about a specific topic.
>
> Tables organize that data in columns (called attributes) and rows (called records). One of the attributes (columns) of the table is almost always a number data type that is used as a key. A key is used to uniquely identify the data in a particular row. Key values must be unique and not null. A primary key is just the attribute(s) that have been designated as the main key for the table.
>
> **Data Services also includes the storage engine and the query engine which deal with the saving, accessing and manipulation of the data. A *Query* is used to access and manipulate the data. Queries are also called views because they are a user's "view" of the data in the database.**

Objectives:

1. To write out and then implement proper SQL to see a particular result set.
2. To become familiar with graphical query method of QBE (Query-By-Example) that is used in Access.
3. To create queries that view and/or manipulate data from one or more tables.
4. To create forms that will allow a user to add and retrieve data from tables and display it in a pleasing manner.

The important points to remember when creating a query are:

- Always include any tables that hold the data you want to see
- Always include any tables that are in the path to the data you want to see.
- When using a join operation, always indicate where the tables are related: Match the keys.
- When generating a query on more than one table, always use table name qualifiers to distinctly identify the correct table
  Example:

  tblAdvisor.LName and **tblStudent.LName** refer to 2 distinct attributes.

  If you simply use **LName** in a query on those 2 tables, Access will not know which attribute you are referring to.

1. **Writing Out SQL**
   Using what you have learned about Structured Query Language, write out the proper SQL syntax in the boxes below to return the requested data.

   Example: Show all advisor information from the Advisor table.

   ```
   SELECT AdvisorID, LName, FName, Department
   FROM tblAdvisor;
   ```

   OR, use the * as a wild card to show all columns:

   ```
   SELECT *
   FROM tblAdvisor;
   ```

   A) Show all students from the Student table.

   B) Show all students in alphabetical order by last name.

C) Show only the Advisor Names from a particular Department
[insert name of one of the Departments in the Advisor table].

D) Show all Advisors and the Students they advise. (Requires a Join)

E) Show the last name and first name of *[insert Major Name]* **majors***.* (Requires a Join)

F) Show all advisors working with *[insert Major Name]* **Majors.** (Requires a Join)

G) Show each student and their advisor, the advisor's department and their major. (Requires a Join)

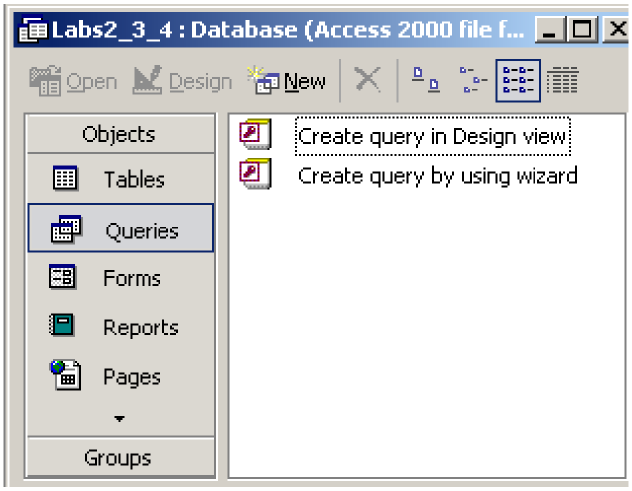
2. **Move from paper to database**
   A. Download the database you have been working on since Lab 12 from your Dante account to the desktop and open it.

   B. Add 2 additional records in each table using the forms your produced last lab.

Now that we have some seed data in our database, we can request the display of data from different tables.

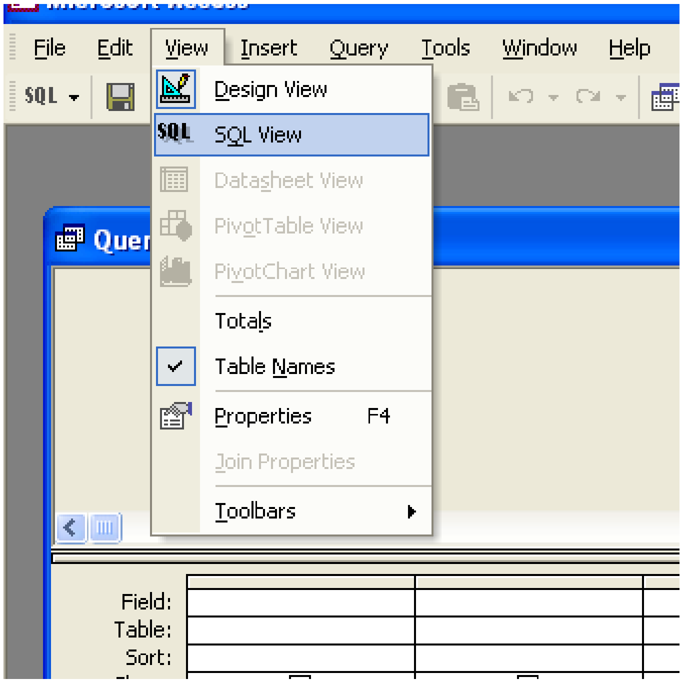
You can enter queries using the QBE or with straight SQL in Access. Spend some time using the SQL View to enter the queries A, B and C from step 1 above. Then use the QBE to generate the more difficult D, E, F and G queries.

3. **Query Construction in SQL View:**
   A. To get to the SQL View, go to the Queries window and double click on the **Create query in Design View** option.

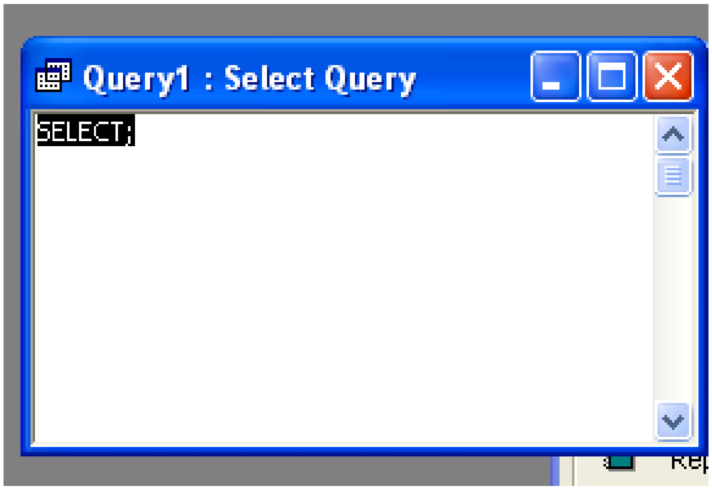

Labs2_3_4 : Database (Access 2000 file f...
Open
Design
New
Objects
Tables
Queries
Forms
Reports
Pages
Groups
Create query in Design view
Create query by using wizard

   B. When the Show Table window comes up, close it without adding any of the tables you have created.

   C. From the top menu, go to **View>SQL View** or click on the SQL icon on the tool bar:



SQL View:

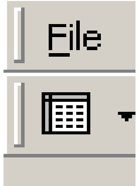


```
SELECT;
```

D. Enter the SQL statement from 3A into the window.
Example:

SELECT *
FROM tblAdvisor;

E. When the SQL Clauses are in place, click on the Datasheet view icon in the upper left hand or go to **View>DataSheet View** to display your query results



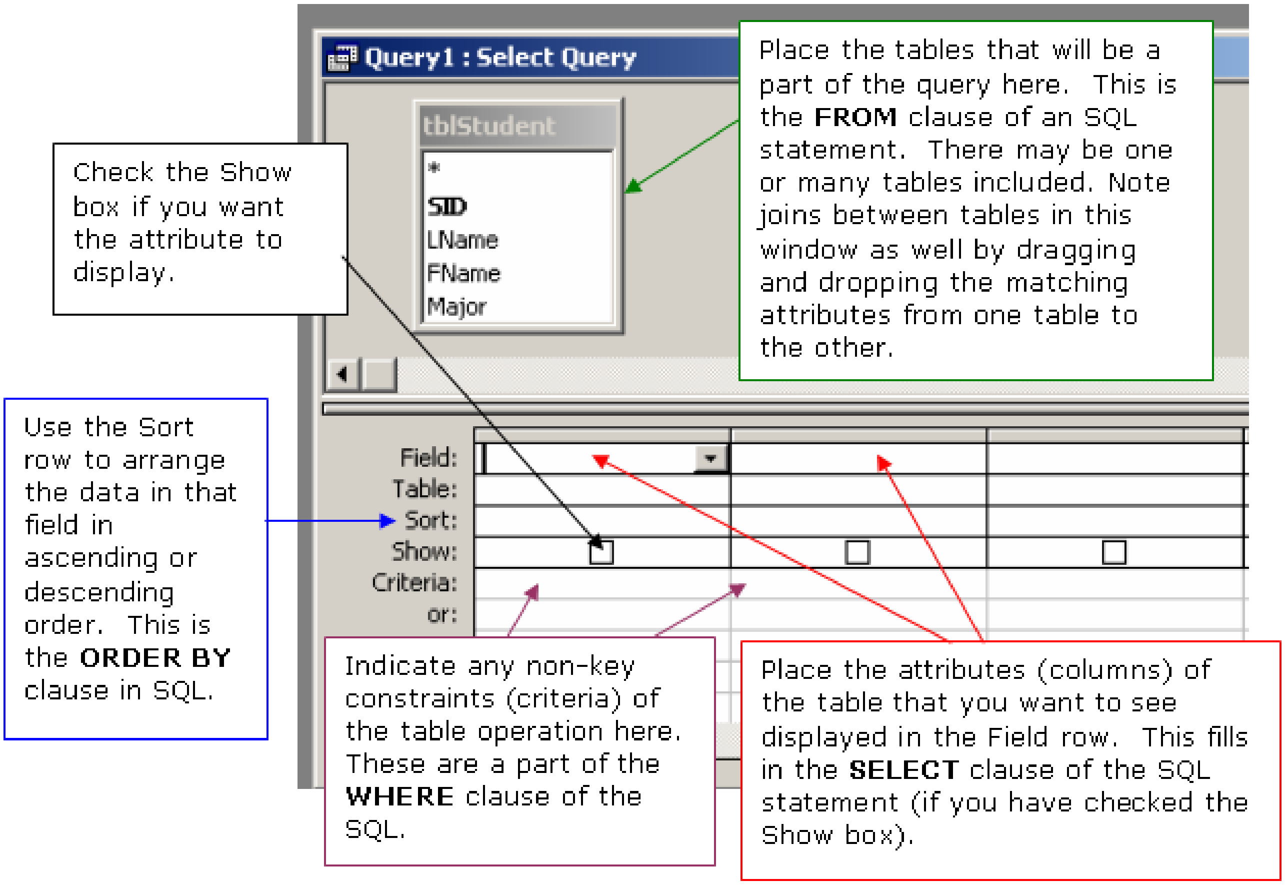
For the first view, number of rows displayed should be the same number as the records in your student table since you asked for all students.

F. Save the query for review later. Repeat the process for 3B and 3C.

What goes where in the QBE:

QBE, or Query By Example, is a graphical query language. This tool allows a user to follow a "drag and drop" method of constructing a query against a data store. You can consider the QBE as a translation tool that takes user input in graphical locations and generates SQL.

By breaking down the interface, a user can see how this tool is able to construct SQL with the input provided.



Place the tables that will be a part of the query here. This is the **FROM** clause of an SQL statement. There may be one or many tables included. Note joins between tables in this window as well by dragging and dropping the matching attributes from one table to the other.

Check the Show box if you want the attribute to display.

Use the Sort row to arrange the data in that field in ascending or descending order. This is the **ORDER BY** clause in SQL.

Indicate any non-key constraints (criteria) of the table operation here. These are a part of the **WHERE** clause of the SQL.

Place the attributes (columns) of the table that you want to see displayed in the Field row. This fills in the **SELECT** clause of the SQL statement (if you have checked the Show box).

4. Use the QBE:

A. Go to the Queries window and double click on the **Create Query in Design View** option.

B. For practice, repeat 3A using the QBE. Add the table needed for the first view from the Show Table window and then close it.

C. Drag all the fields you wish to see displayed down into the Select columns, or choose them from the Field combo box.

D. View the results in Form View

E. Go back to the **Design View** of the query. Can you figure out how to sort the records and display them alphabetically by last name?

You shouldn't have to double check to see if the query is correct, but if you aren't sure of the join, look in your tables to see what information you should get.

Practice (use the QBE):

Determine what information is necessary to show or what has been requested. Not all attributes used in a query need to be shown to the user.

5. Show me all students (just their first and last names) and the name of their Major.

6. Show all students (first name and last name), their major and their advisor. Sort by major.

7. Show all students with an ART major **OR** a CSE Major. (Use two Majors you actually entered if you don't have ART or CSE)

8. Show who advises student number 4 (StudentID).

9. Show all Students who have an Advisor from the certain department AND have a particular Major. Only show the student names.