Criterion E: Product development

**List of Various Techniques Used in Database Development**

Tools in Microsoft Access have been used to develop this database which employed these complex techniques:

1. Complex Query Development for Data Retrieval from Tables
2. Embedded Macros for Database Navigation, Print Commands and Record Saving Functions.
3. Authentic Data Representation on Forms and Reports through Linking Tables and Queries as Sources.

**Step by Step Product Development**

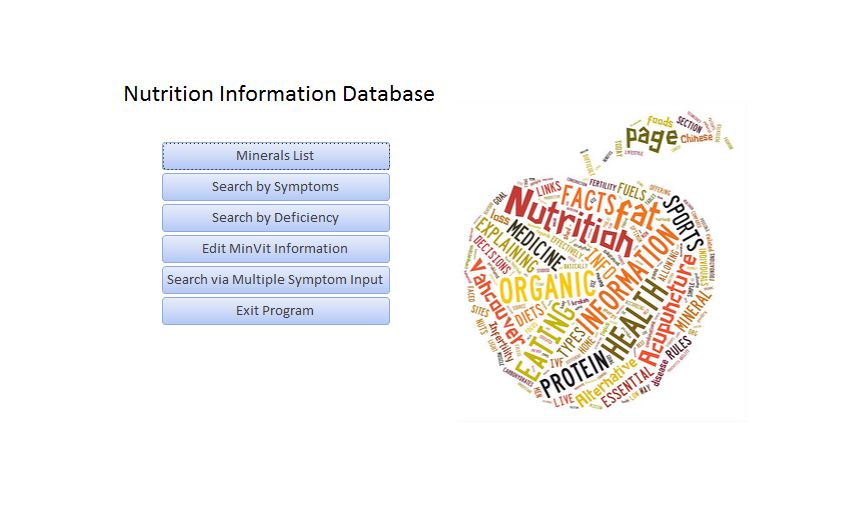
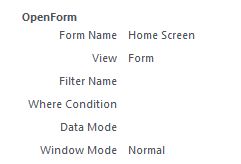


Figure 1-Home Screen form

The home screen opens due to AutoExec macro, which is set to run when the database opens. it performs the following function:



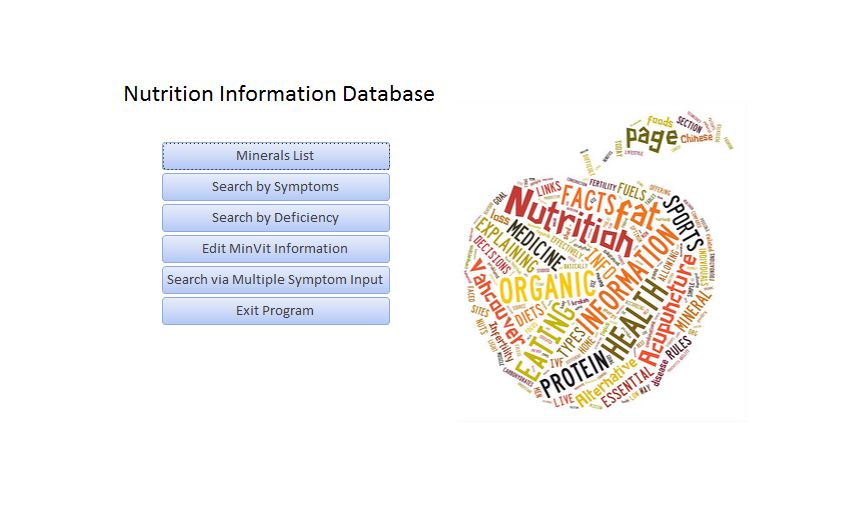


Figure 2- Functions on the home screen are listed as buttons

Complex Query Development for Data Retrieval from Tables in the Database

All functions except one require queries in retrieving data for any search. There are two types of queries; one prompts the user to input information from a single field and then outputs the record(s) that contain that information. The other prompts the user to provide multiple inputs one by one, and then outputs all records that have all the values that the user input.

The construction of both the queries is quite similar, with nearly the same steps followed to create the queries. However, they are constructed from different tables and have different criteria for parameter value e.g. a query that outputs the complete information about the deficiency of a mineral/vitamin is constructed as follows:

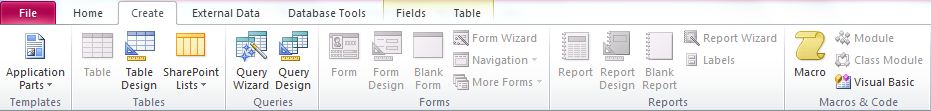


Figure 3-First we go on create and click on Query design where the program opens a prompt for us to choose a table or query from

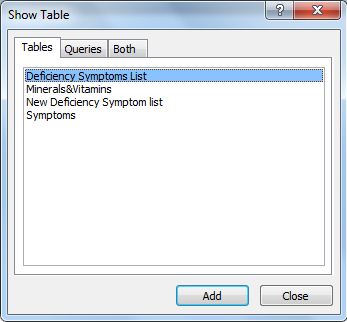
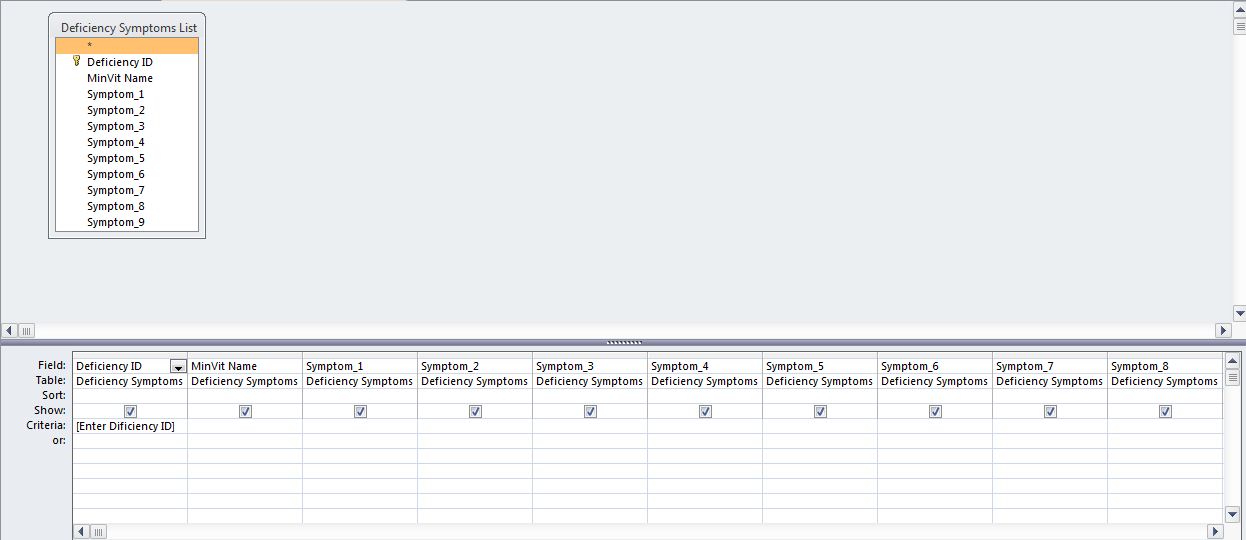


Figure 4-. We will choose the ‘Deficiency Symptoms List’ table

Once this table is selected and the button ‘Add’ is clicked, all fields from the table are added and the following is display is set:



Now to set the user prompt, in the criteria of ‘Deficiency ID’ a prompt instruction is set. Once the numerical ID is input by the user, the program outputs the values for all other fields of the records that have that ‘Deficiency ID’. The function is constructed like:

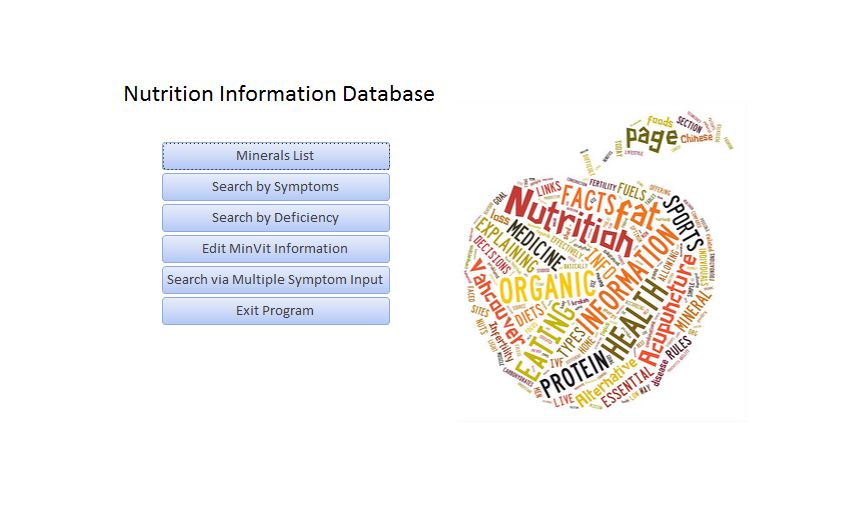


Figure 5-User clicks on 'search by Deficiency' button

User is directed to the ‘List of Deficiencies’ report.

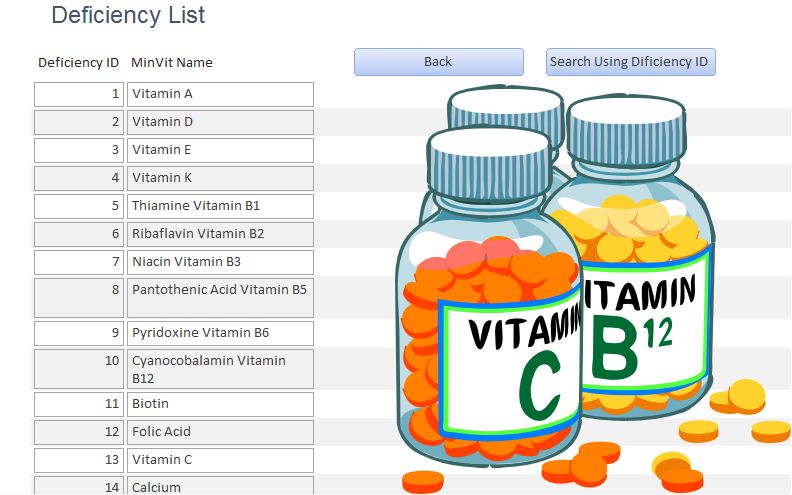
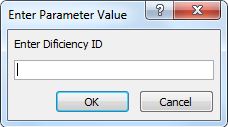


Figure 6-User would click the 'search by deficiency ID' buttonU

Then query runs and the user is prompted:

 When the user enters a Deficiency ID, the query matches the input with the IDs and retrieves the information of that deficiency in form of a report

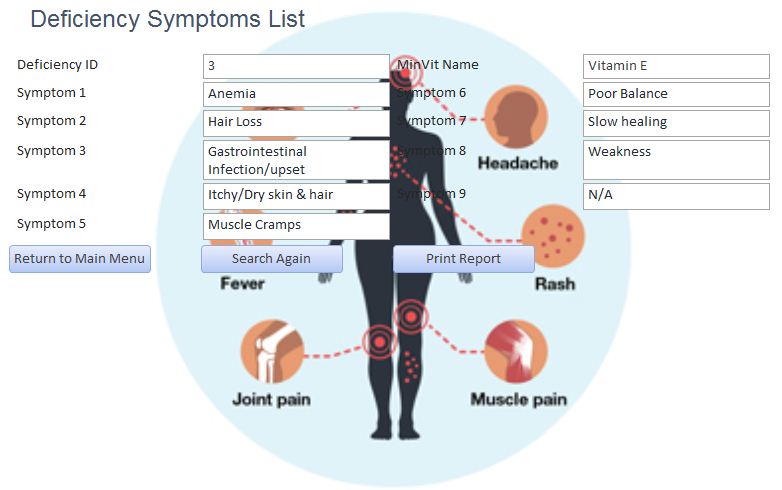
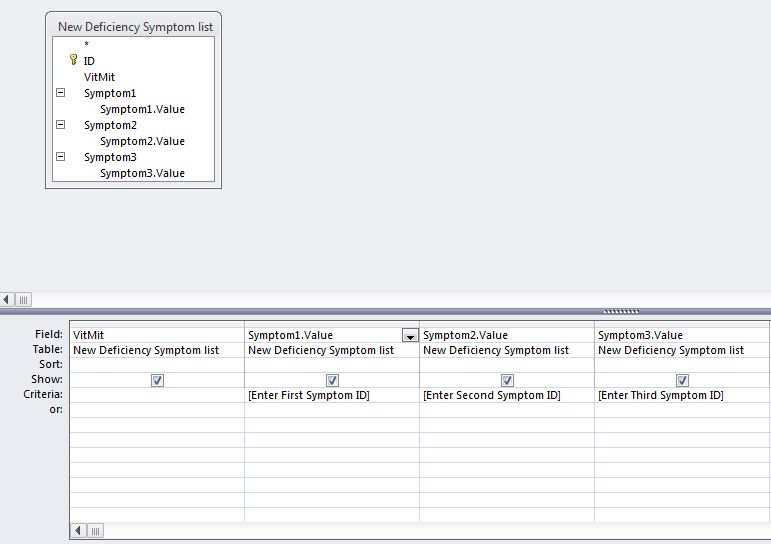


Figure 7-report of vitamin E when user inputs '3' as the deficiency ID

Two other functions use the same form of query. One searches and retrieves information of a mineral/vitamin using its name. The other searches for and displays the information of a symptom through its ID.

Another query prompts user to input multiple values and outputs all records that contain those values. A function that uses this query is ‘Search via Multiple Symptom Input’. Like how we constructed the previous query, the query design page is opened and then the ‘New Deficiency Symptom List’ table is opened to display the following:



The criteria are set for the user to input the three symptom IDs for the search. The first input will be compared with the ID of the symptom stored in the ‘Symptom1’ field, then the second input will be compared with the ID of the symptom stored in the ‘Symptom 2’ field and the same will be done when the third input is compared with the ID of the symptom stored in ‘Symptom 3’.

Execution of this function:

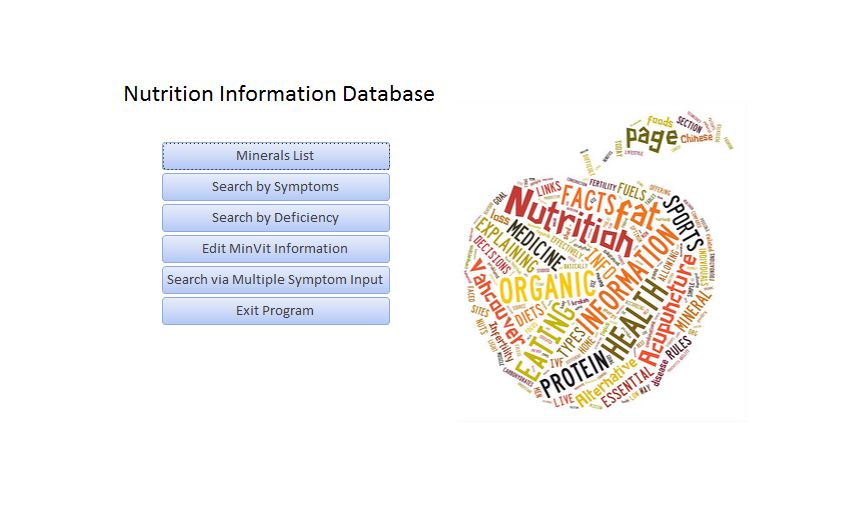


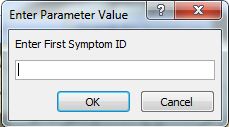
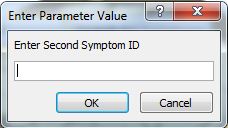
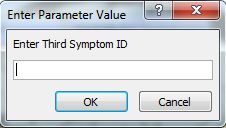
Figure 8-The user first clicks the ‘Search via Multiple Symptom Input’ button

User is directed to the ‘Other List of Deficiencies’ report.



Figure 9-user clicks the ‘Search for Symptoms’ button

The query runs and the user is prompted:

As soon as the user enters all three symptom IDs, the system runs a comparison check to identify deficiencies that show all three symptoms. It then outputs the deficiencies in form of a report

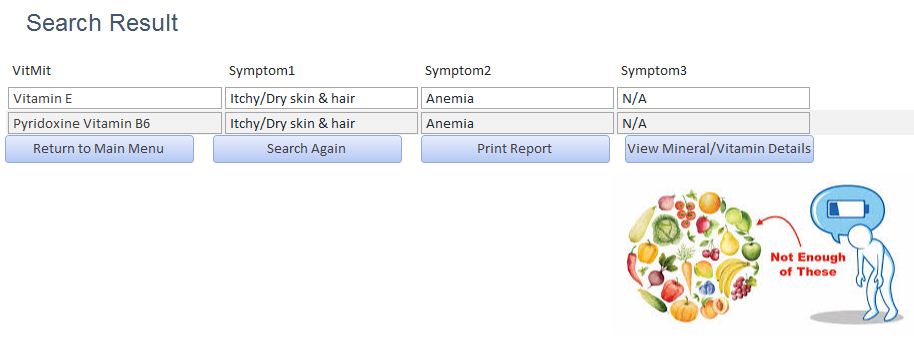


Figure 10-Example. Inputs were '3' (Itchy ID), '14' (Anemia ID) & '45' (N/A ID)

Embedded Macros for Database Navigation, Print Commands and Record Saving Functions

This database contains buttons with embedded macros for different functions. These buttons are placed on forms and reports, to navigate between forms/reports, print report when required, move between records to view and edit data, save data, and exit the database.

The macro that moves the user from one report/form to another involves two functions. First, it would open the next form or report and then close the current.

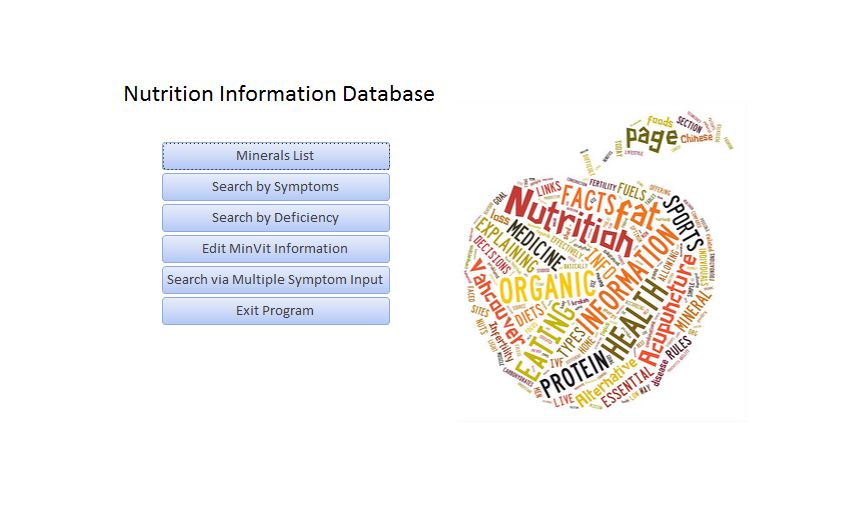


Figure 11-User clicks on 'search by symptom'

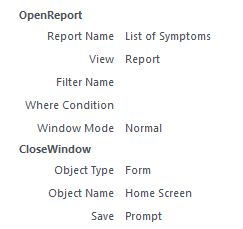


Figure 12- Following macro is executed

This macro first opens ‘List of Symptoms’ and then closes the Home Screen form.



Another embedded macro within this database is the ‘Print Object’ macro.

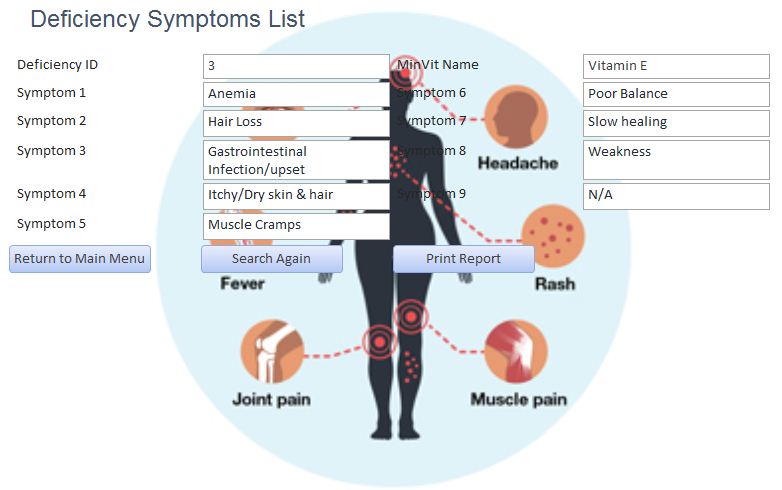


Figure 13-User clicks on ‘print report’



Figure 14-Following 'print report' macro is executed

Moving between records to view and edit data is another set of macros that has been used. One button is assigned to move to a previous record and one to the next. Hence, the user can view and navigate through all the records. This works like:

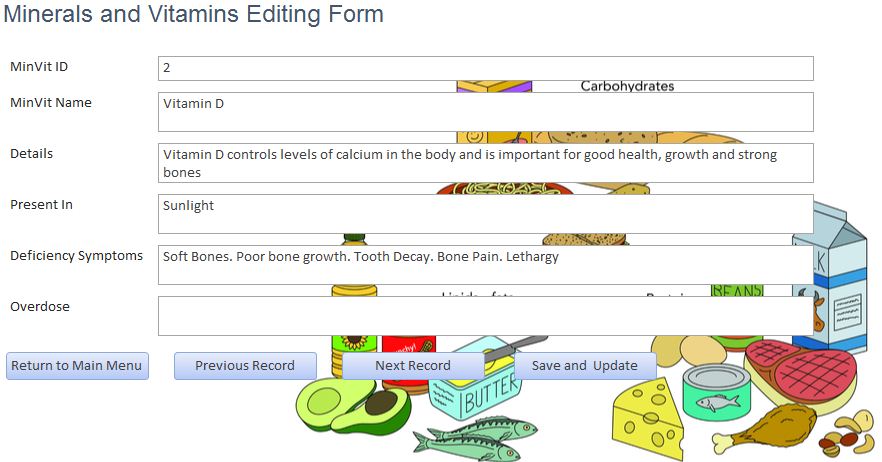


Figure 15-'previous record' and 'next record' to navigate back and forth records

The embedded macros in these are as follows:



Figure 16-macro for moving to previous adjacent record



Figure 17-macro for moving to next adjacent record

Another macro is for saving a record. The data that was viewed and edited in the ‘Edit MinVit Form’ needs to be saved in order to be updated or added to the database. Therefore, the save record macro makes the changes and updates the records in the table.

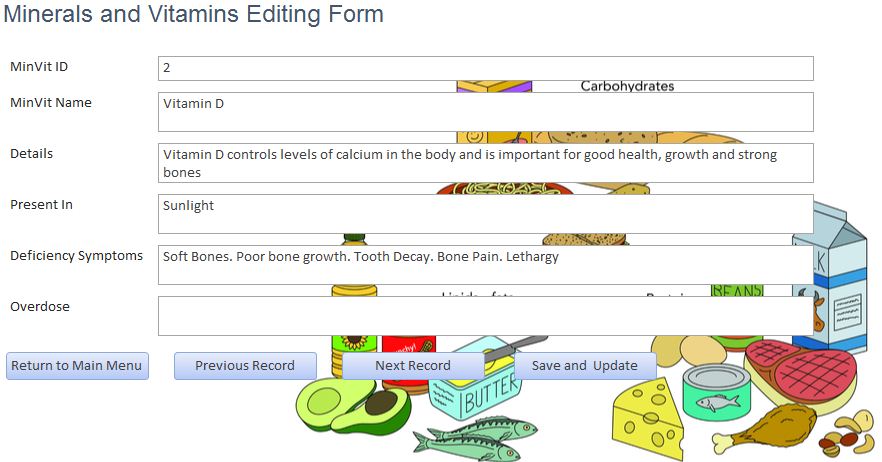


Figure 18-User clicks on 'save and update'



Figure 19-macro embedded in 'save and update' button

The final embedded macro function is the ‘Close Database’ function. This function closes the database. This function can be performed from the Home Screen form only

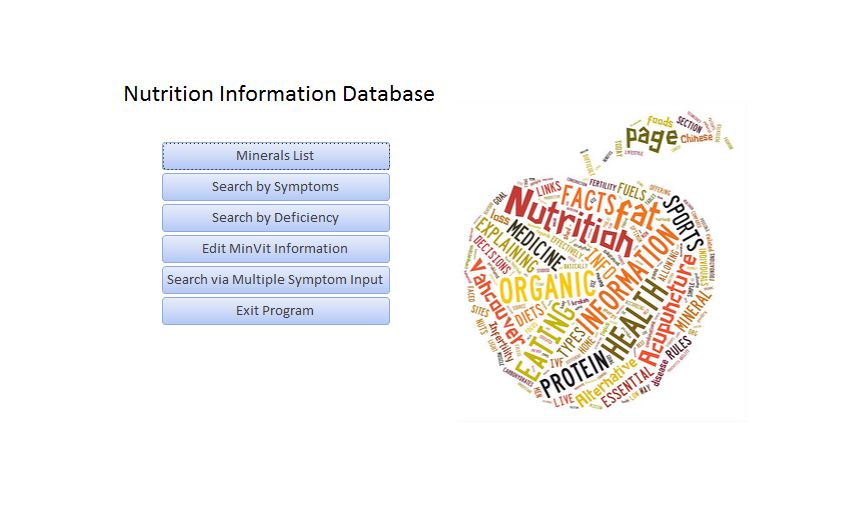


Figure 20-User clicks on 'exit program'

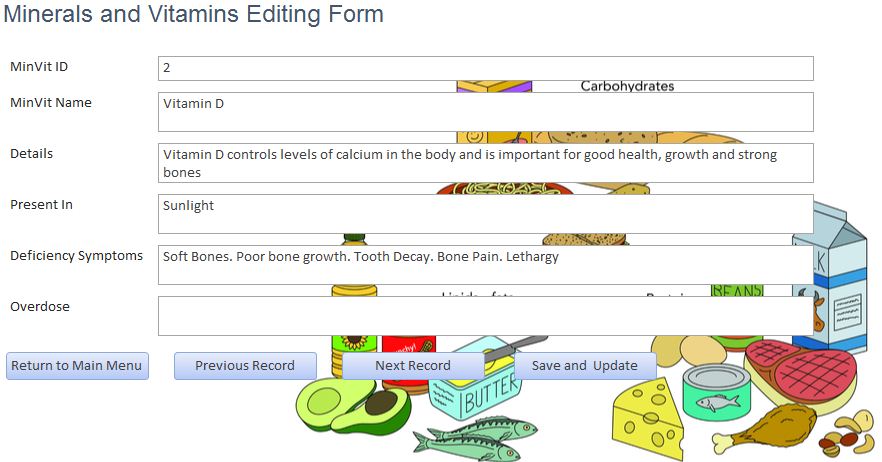


Figure 21-Following macro is executed when user clicks on 'exit program'

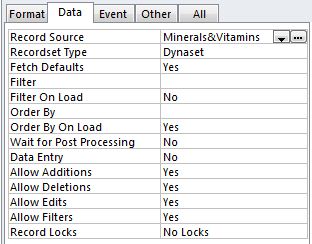
Authentic Data Representation on Forms and Reports through Linking Tables and Queries as Sources

The forms and reports that are created to allow the user to view or edit data from tables and queries, use these tables and queries as sources from where they display the data. Data sources are used in three ways. One; data imported from a table on to a form so that the user may view and edit the information. Two; data is viewed from a table as a report. Three; a data source is used, as the search result of a query being displayed on a report.

The ‘Edit MinVit’ Form displays different fields and alongside them, information from individual records like:



The information of records is displayed individually within their respective fields, as was in the ‘Minerals&Vitamins’ table. Hence, the source of this form is the ‘Minerals&Vitamins’ table and is visible in the form properties under the data heading as such:



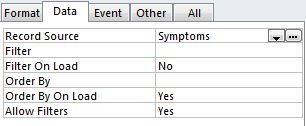
The data in this form that is imported from the ‘Minerals&Vitamins’ table can be edited, added or even deleted. All the changes can be saved back in the ‘Minerals&Vitamins’ table through the ‘Save and Update’ button.

Another way that data types are used in this database are as a list of viewable records in form of a report. Various lists in this database use this type of data representation, where particular records from a source table are displayed in form a report.



Figure 22-'Symptom ID' and 'Symptom Name' are records from the 'Symptoms' table

All the records are listed in chronological order for the fields that were imported from the table, and the reason being is due to the following:



As visible under the Data section of the report properties, the data source for the report is the ‘Symptoms’ table, and thus Symptom IDs and their corresponding Symptom Names are listed as they were in the ‘Symptoms’ table.

The final way data is used here, is the representation of query results in form of reports. Just like records, the source of the report is set as the query and the information from the query would be displayed in the report. However unlike the table, from which each and every record is displayed, the query reports would only contain results that the query outputs after its function. Example:

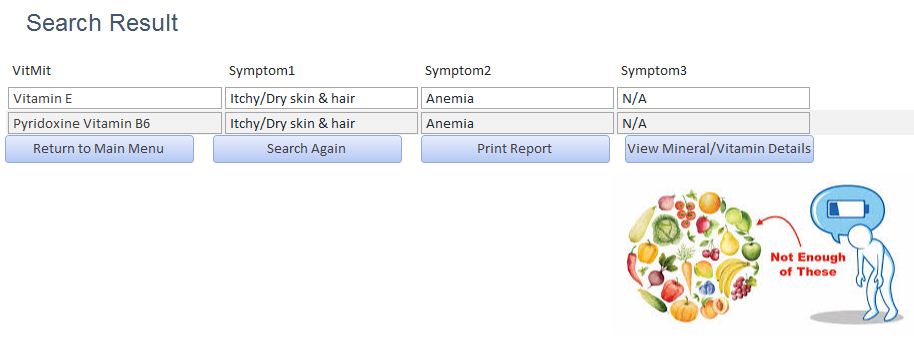
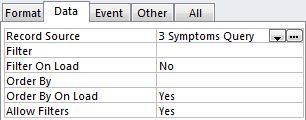


Figure 23-search using multiple inputs discussed earlier in this document generated the following report

Even though this query was made using the ‘New Deficiency Symptom list’ table, it did not display all the records in the table, but only displayed those minerals/vitamins that had the symptoms which correspond to the IDs input earlier.

In the Data section of the ‘3 Symptoms Report’, it can be viewed that the source of the displayed information is the ‘3 Symptoms Query’:



**Words: 1168 (excluding headings, captions and screenshots)**