**Function Definition: def back\_record():**

This line defines a function named back\_record. A function is a reusable piece of code that performs a specific task.

**global current\_record\_id**

This line specifies that current\_record\_id is a global variable. Global variables are accessible from anywhere in the program, not just within the function or block of code where they are defined.

**if current\_record\_id is None:**

This line checks if the current\_record\_id variable has no value (None). This situation would indicate that no record is currently selected.

**tkinter.messagebox.showwarning("Warning", "No record selected.")**

If no record is selected, this line shows a warning message to the user saying "No record selected."

**return**

This line exits the function early if no record is selected, preventing the rest of the code from executing.

**try:**

This line starts a 'try block' where you attempt to run code that might cause an error. It is used for error handling.

**script\_dir = os.path.dirname(os.path.realpath(\_\_file\_\_))**

This line gets the directory of the script or program that's currently running.

**file\_path = os.path.join(script\_dir, file\_name)**

This line joins the script directory path with a file name to create a full file path.

**wb = load\_workbook(file\_path)**

This line opens an Excel workbook file located at the specified file\_path.

**ws = wb.active**

This sets ws to the active worksheet in the workbook, essentially the sheet that's open.

**for row in reversed(list(ws.iter\_rows(min\_row=2))):**

This line loops through the rows of the worksheet in reverse order, starting from the bottom, but skipping the first row (often header row).

**if row[0].value and int(row[0].value) < int(current\_record\_id):**

Inside the loop, this line checks if the first cell in a row has a value and if its numerical value is less than current\_record\_id.

**previous\_record\_id = row[0].value**

If the condition is true, this line sets previous\_record\_id to the value found, identifying the previous record.

**break**

This exits the loop early since the previous record is found.

**if previous\_record\_id is None:**

After the loop, this line checks if no previous record was found.

**response = tkinter.messagebox.askyesno("First Record", "This is the first record. Do you want to move to the last record?")**

If it is the first record, this line asks the user if they want to move to the last record.

**if response:**

This line checks the user's response. If it's 'yes', the following block will execute.

**last\_row = list(ws.iter\_rows(min\_row=2))[-1]**

This gets the last row of the worksheet.

**previous\_record\_id = last\_row[0].value**

This sets previous\_record\_id to the ID of the last record.

**for row in ws.iter\_rows(min\_row=2):**

This line iterates through the rows again to find the previous record.

**if row[0].value == previous\_record\_id:**

This checks if the current row's ID matches the previous\_record\_id.

**current\_record\_id = previous\_record\_id**

If it matches, update current\_record\_id with the previous\_record\_id.

**name\_var.set(row[1].value)**

These lines update variables on the screen with values from the selected record.

**except Exception as e:**

This part catches any errors that might have occurred in the 'try block'.

**tkinter.messagebox.showerror("Error", f"An error occurred: {e}")**

If there's an error, this line shows an error message with the details.

This function essentially navigates to the previous record in a dataset, updating the display with its information or handling various edge cases, like the absence of a selected record or reaching the first record in the dataset.