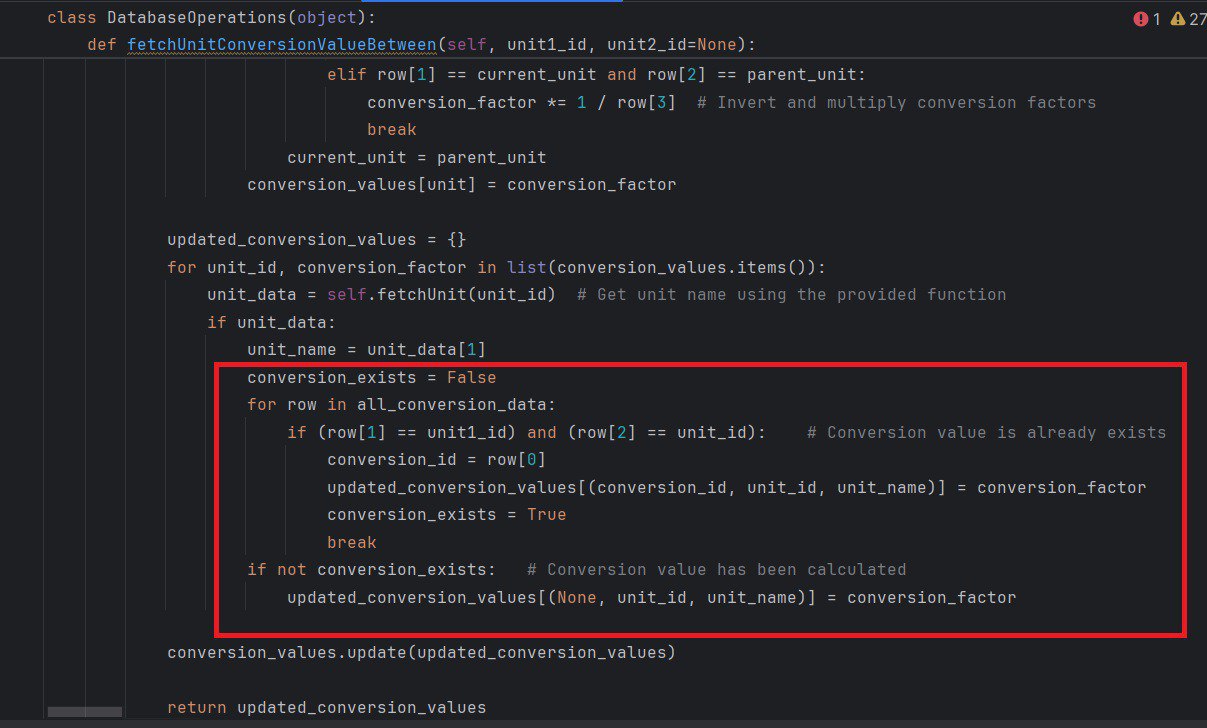
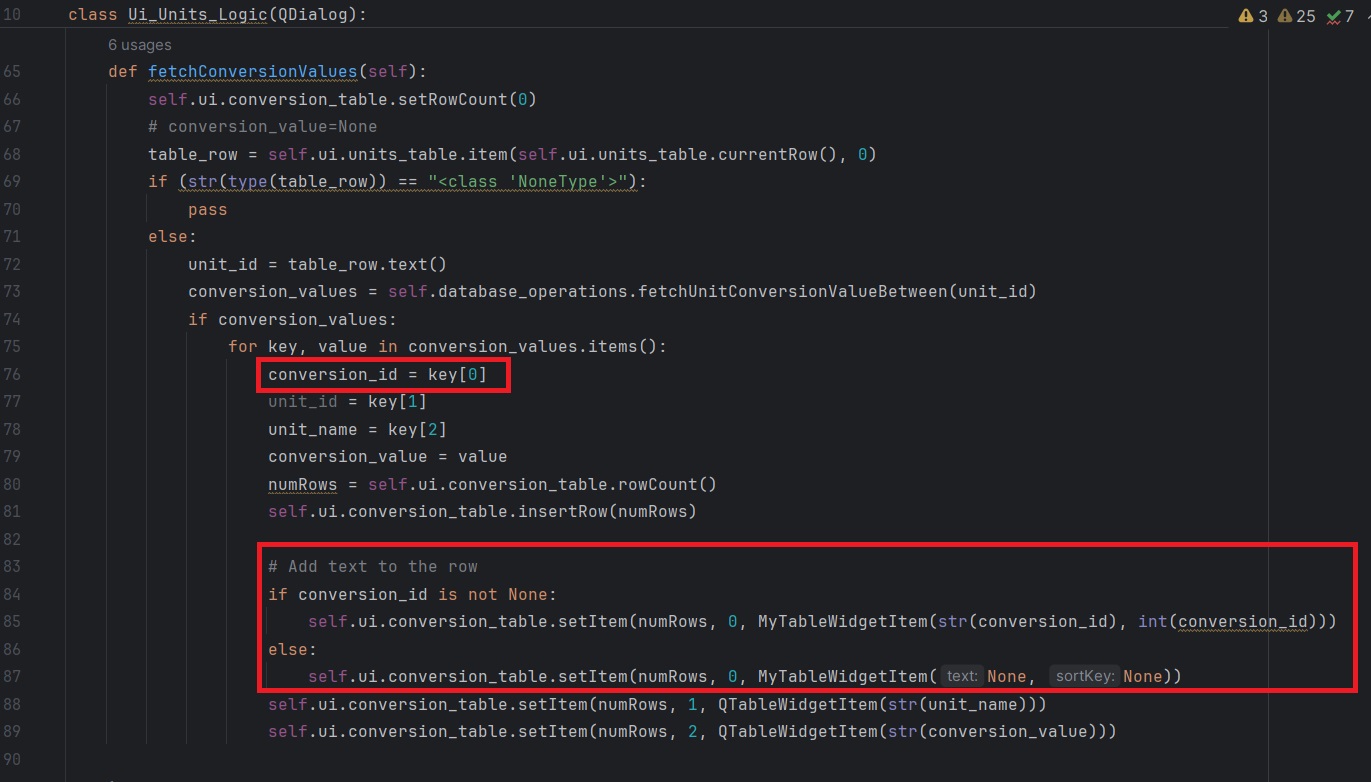
* **Unit Conversion:**

To ensure the right conversion ID is displayed in the Ui, first, in the file **DatabaseOperations.py**, we modified the method **fetchUnitConversionValueBetween**, in the end and after calculating the conversion values between the units, we added a block that checks if the conversion value between the two units does exist in the database as a direct value provided by user and then the **`conversion\_id`** should have the same ID in the database, otherwise the value is calculated by the algorithms and the **`conversion\_id`** should have the value NONE.



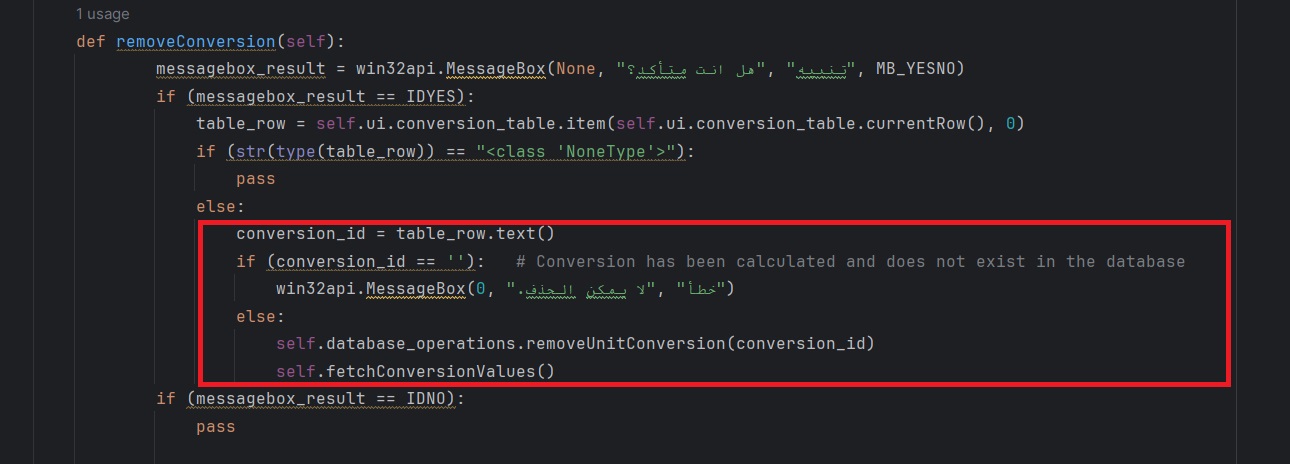
We notice that the returned dictionary (**updated\_conversion\_values**) have now three values in every key item, (**conversion\_id, unit\_id, unit\_name**), so we can now access the conversion ID easily as we will see.

Now in the file **Ui\_Units\_Logic.py**, in the method **fetchConversionValues** we can get the conversion\_id directly from the key of every dictionary item:

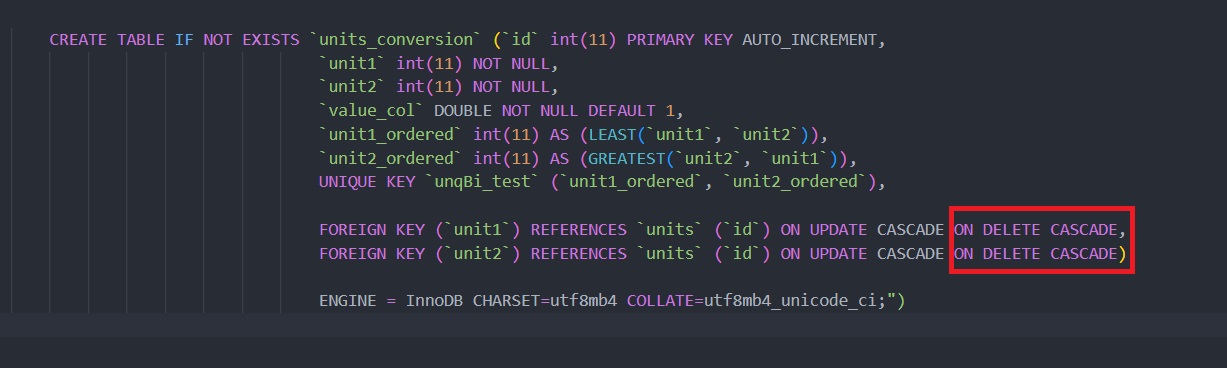


This value will either be an integer number refers to the conversion id that exists in the database, or NONE if the conversion has been calculated.

Also we have modified the **removeConversion** method to check if the deletion is possible when the conversion exists in the database:



Finally, we have modified the files **MySQLConcector.py** and **SQLiteConnector.py**, we updated the Foreign keys restrictions for `unit1`and `unit2`, we set it to **ON DELETE CASCADE**, so now whenever a unit is deleted, all its conversions will be deleted too:



Same restriction has been added to SQLiteConnector.

**Note:**

In the file **Ui\_Units\_Logic.py**, in the method **removeUnit,** we have to refresh the conversion table so we called the method **fetchConversionValues** after the deletion process:

