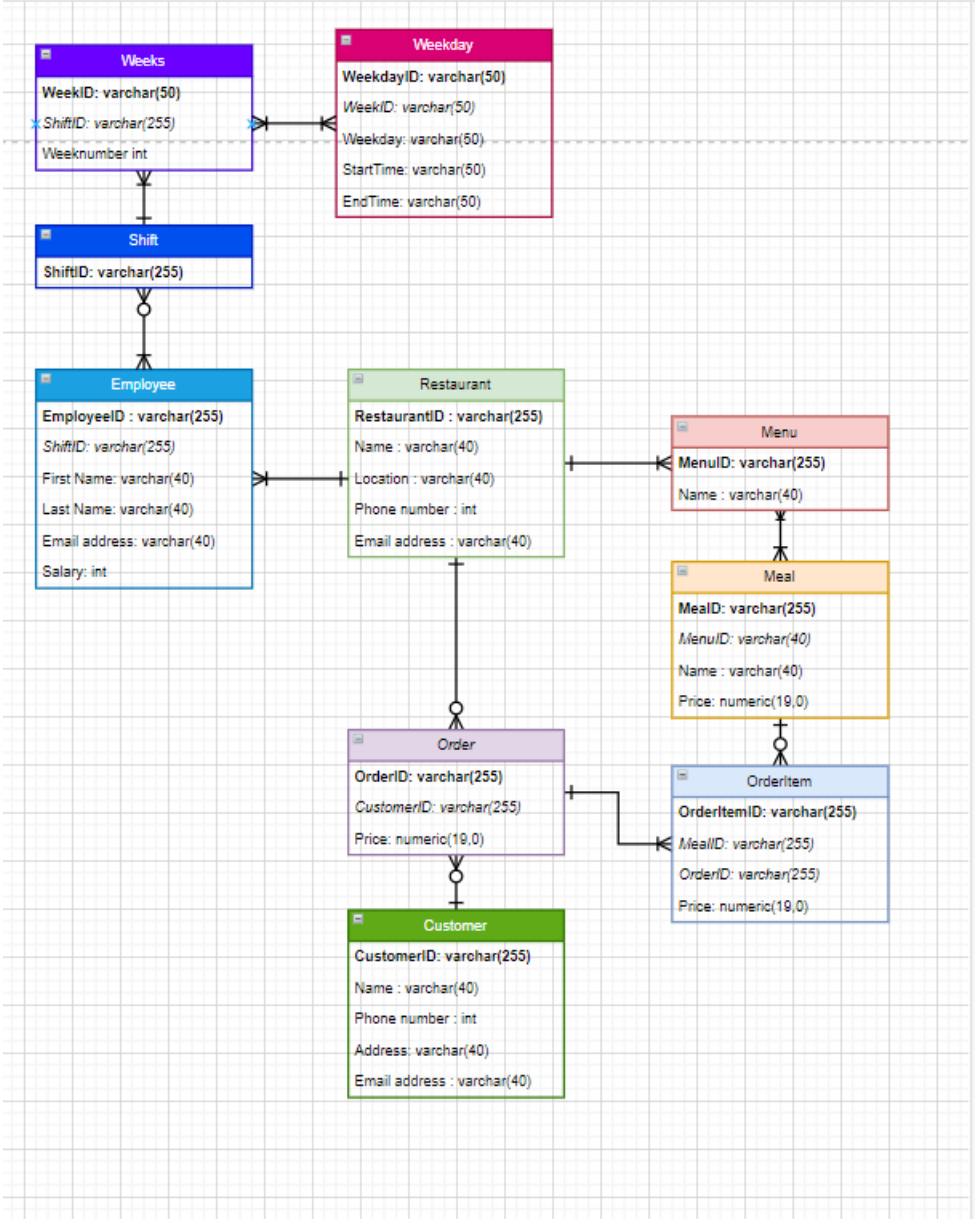
Final Project

Leevi Kämäräinen 0568396

My ER-diagram:



I made the diagram so that one restaurant may have one or more menus and employee, and one restaurant can also have zero or more orders.

One or more menus can contain one or more meals (one meal may be in more than one menu). One meal can be in zero or many ordered items so it means that one specific meal can be ordered zero or many times.

One order may contain one or more order items and one customer may have zero or more orders. You can be customer without ordering anything.

One or many employees may belong to zero or more shifts, which means that if you haven't any shifts you are for example at vacation. This also means that multiple employees may be in the same shift. One shift may contain multiple weeks and multiple weeks may contain multiple weekdays.

My code:

First the subprogram to create connection to SQL database:

```
File  Edit  Format  Run  Options  Window  Help
import sqlite3
from sqlite3 import Error
def create_connection(db_file):
    """ create a database connection to a SQLite database """
    conn = None
    try:
        conn = sqlite3.connect(db_file)
        print(sqlite3.version)
    except Error as e:
        print(e)
    finally:
        if conn:
            conn.close()
```

Here I create the tables if they don't exist yet.

```
def main():
    if __name__=='__main__':
        create_connection(r"D:\Basics of Database systems\Pythons\FinalProject")
    conn = sqlite3.connect("FinalProject.db")
    conn.execute("""CREATE TABLE IF NOT EXISTS Restaurant(
                    RestaurantID varchar(255) NOT NULL,
                    Name varchar(40),
                    Location varchar(40),
                    Phone number varchar(40),
                    Email address varchar(40),
                    PRIMARY KEY (RestaurantID)
                    );""")
    conn.execute("""CREATE TABLE IF NOT EXISTS Menu(
                    MenuID varchar(255) NOT NULL,
                    Name varchar(40),
                    PRIMARY KEY (MenuID)
                    );""")
    conn.execute("""CREATE TABLE IF NOT EXISTS Meal(
                    MealID varchar(255) NOT NULL,
                    MenuID varchar(255),
                    Name varchar(40),
                    Price numeric(19,0),
                    PRIMARY KEY (MealID),
                    FOREIGN KEY (MenuID) REFERENCES Menu(MenuID)
                    );""")
    conn.execute("""CREATE TABLE IF NOT EXISTS OrderItem(
                    OrderItemID varchar(255) NOT NULL,
                    MealID varchar(255),
                    OrderID varchar(255),
                    Price float(19,0),
                    PRIMARY KEY (OrderItemID),
                    FOREIGN KEY (MealID) REFERENCES Meal(MealID),
                    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)
                    );""")
    conn.execute("""CREATE TABLE IF NOT EXISTS Orders(
                    OrderID varchar(255) NOT NULL,
                    CustomerID varchar(255),
                    Price numeric(19,0),
                    PRIMARY KEY (OrderID),
                    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
                    );""")
    conn.execute("""CREATE TABLE IF NOT EXISTS Employee(
                    EmployeeID varchar(255) NOT NULL,
                    ShiftID varchar(255),
                    First Name varchar(40),
                    Last Name varchar(40),
                    Email address varchar(40),
                    Salary int,
                    PRIMARY KEY (EmployeeID),
                    FOREIGN KEY (ShiftID) REFERENCES WorkingHours(ShiftID)
                    );""")
    conn.execute("""CREATE TABLE IF NOT EXISTS Shift(
                    ShiftID varchar(255) NOT NULL,
                    PRIMARY KEY (ShiftID)
                    );""")
```

```
    conn.execute("""CREATE TABLE IF NOT EXISTS Weeks(
                    WeekID varchar(255) NOT NULL,
                    ShiftID varchar(255),
                    Weeknumber int,
                    PRIMARY KEY (WeekID),
                    FOREIGN KEY (ShiftID) REFERENCES Shift(ShiftID)
                    );""")
    conn.execute("""CREATE TABLE IF NOT EXISTS Weekday(
                    WeekdayID varchar(50) NOT NULL,
                    Weekday varchar(50),
                    WeekID varchar(255),
                    Start time varchar(40),
                    End time varchar(40),
                    FOREIGN KEY (WeekID) REFERENCES Weeks(WeekID),
                    PRIMARY KEY (WeekdayID)
                    );""")
    conn.execute("""CREATE TABLE IF NOT EXISTS Customer(
                    CustomerID varchar(255),
                    Name varchar(40),
                    Phone number varchar(40),
                    Address varchar(40),
                    Email address varchar(40),
                    PRIMARY KEY (CustomerID)
                    );""")

    conn.commit()
    conn.close()
    mainmenu()
    print("Thank you for using Restaurant database!")

main()
```

Here is my main menus code where I can control what I want to do next:

```
def mainmenu():
    print("Welcome to main menu!\n")
    choice = 1
    while choice != 0:
        print("""What would you like to do:
            1. Add data
            2. Look at data
            3. Change data
            4. Order food (customer only)
            5. Look at customer orders
            6. Display Employee Hours
            0. Exit
            """)
        choice = input("Your choice: ")
        if choice=="1":
            addData()
            continue
        if choice=="2":
            selectData()
            continue
        if choice=="3":
            changeData()
            continue
        if (choice == "4"):
            orderFood()
            continue
        if (choice == "5"):
            displayOrders()
            continue
        if (choice == "6"):
            displayEmployeeHours()
            continue
        if(choice == "0"):
            break
        else:
            print("Invalid choice, try again!")
            continue
```

This is what it looks like:

```
2.6.0
Welcome to main menu!

What would you like to do:
        1. Add data
        2. Look at data
        3. Change data
        4. Order food (customer only)
        5. Look at customer orders
        6. Display Employee Hours
        0. Exit

Your choice:
```

Here is the code for how I input data to the tables. I made one subprogram where with indexes you can choose the table where you want to input data to, and after that it automatically gives you the variables which you need to give data to. You may think this subprogram that only admins may use, for customers you have order food program.

Questionmark variable here is the sql commands questionmarks (… VALUES(**?,?,?**)) which are used to prevent SQL injection. There are as many question marks in the sql command as there are variables to add.

```
def addData():
    conn = sqlite3.connect("FinalProject.db")
    cursor = conn.cursor()
    table, columns = chooseTable()
    datacount = len(columns)
    questionmark="?"
    i = datacount
    datalist = []
    while i>1:
        i = i-1
        questionmark = questionmark+",?"
    for column in columns:
        data = input("Give a value for "+column+": ")
        datalist.append(data)
    sqlcommand = "INSERT INTO "+table+" VALUES("+questionmark+")"
    try:
        cursor.execute(sqlcommand,datalist)
    except Error as e:
        print(e)
        print("Invalid SQL command, try again!")
    conn.commit()
    cursor.close()
    conn.close()
```

Here is the chooseTable subprogram where I have coded manually each of the table names and their column names. This prevents SQL-injection so the user can't input table or column names which can include SQL code which can compromise your database:

```
def chooseTable():
    table = ""
    choice=""
    while(table==""):
        print("""Choose table:
                    1) Restaurant
                    2) Menu
                    3) Meal
                    4) OrderItem
                    5) Order
                    6) Customer
                    7) Employee
                    8) Shift
                    9 Week
                    10) Weekday
                    """)
        choice=input("Your choice: ")
        if(choice=="1"):
            table="Restaurant"
            columns = ["RestaurantID","Name","Location","Phone number","Email address"]
            return table,columns
        elif(choice=="2"):
            table="Menu"
            columns = ["MenuID","Name"]
            return table,columns
        elif(choice=="3"):
            table="Meal"
            columns = ["MealID","MenuID","Name","Price"]
            return table, columns
        elif(choice=="4"):
            table="OrderItem"
            columns = ["OrderItemID","MealID","OrderID"]
            return table, columns
        elif(choice=="5"):
            table="Orders"
            columns = ["OrderID","CustomerID","Price"]
            return table, columns
        elif(choice=="6"):
            table="Customer"
            columns = ["CustomerID","Name","Phone number","Address","Email address"]
            return table, columns
        elif(choice=="7"):
            table="Employee"
            columns = ["EmployeeID","ShiftID","First Name", "Last Name", "Email address", "Salary"]
            return table, columns
        elif(choice=="8"):
            table="Shift"
            columns = ["ShiftID"]
            return table, columns
        elif(choice=="9"):
            table="Weeks"
            columns = ["WeekID","ShiftID", "Weeknumber"]
            return table,columns
        elif(choice=="10"):
            table="Weekday"
            columns = ["WeekdayID","Weekday","WeekID","Start time","End time"]
            return table,columns
        else:
            print("Invalid choice, try again!")
            continue
```

Examples for inserting data to each of the tables:

```
What would you like to do:
        1. Add data
        2. Look at data
        3. Change data
        4. Order food (customer only)
        5. Look at customer orders
        6. Display Employee Hours
        0. Exit

Your choice: 1
Choose table:
                1) Restaurant
                2) Menu
                3) Meal
                4) OrderItem
                5) Order
                6) Customer
                7) Employee
                8) Shift
                9 Week
                10) Weekday

Your choice: 1
Give a value for RestaurantID: Restaurant111
Give a value for Name: Pizza Place
Give a value for Location: Lappeenranta
Give a value for Phone number: 040555
Give a value for Email address: pizzaplace@email.com
```

```
Your choice: 1
Choose table:
                1) Restaurant
                2) Menu
                3) Meal
                4) OrderItem
                5) Order
                6) Customer
                7) Employee
                8) Shift
                9 Week
                10) Weekday

Your choice: 2
Give a value for MenuID: Menu1234
Give a value for Name: Spring Menu
```

```
Choose table:
                1) Restaurant
                2) Menu
                3) Meal
                4) OrderItem
                5) Order
                6) Customer
                7) Employee
                8) Shift
                9 Week
                10) Weekday

Your choice: 3
Give a value for MealID: Meal1001
Give a value for MenuID: Menu1234
Give a value for Name: Meatballs&Spaghetti
Give a value for Price: 10
```

```
Choose table:
                1) Restaurant
                2) Menu
                3) Meal
                4) OrderItem
                5) Order
                6) Customer
                7) Employee
                8) Shift
                9 Week
                10) Weekday

Your choice: 4
Give a value for OrderItemID: Item111
Give a value for MealID: Meal1001
Give a value for OrderID: Order123
Give a value for Price: 10
```

```
Choose table:
                1) Restaurant
                2) Menu
                3) Meal
                4) OrderItem
                5) Order
                6) Customer
                7) Employee
                8) Shift
                9 Week
                10) Weekday

Your choice: 5
Give a value for OrderID: Order111
Give a value for CustomerID: Customer123
Give a value for Price: 50
```

```
Choose table:
                1) Restaurant
                2) Menu
                3) Meal
                4) OrderItem
                5) Order
                6) Customer
                7) Employee
                8) Shift
                9 Week
                10) Weekday

Your choice: 6
Give a value for CustomerID: Customer123
Give a value for Name: James
Give a value for Phone number: 040100200
Give a value for Address: Street 100
Give a value for Email address: James@email.com
```

```
Choose table:
                1) Restaurant
                2) Menu
                3) Meal
                4) OrderItem
                5) Order
                6) Customer
                7) Employee
                8) Shift
                9 Week
                10) Weekday

Your choice: 7
Give a value for EmployeeID: EmpID200
Give a value for ShiftID: Shift300
Give a value for First Name: Donald
Give a value for Last Name: Duck
Give a value for Email address: donduck@email.com
Give a value for Salary: 2400
```

```
Choose table:
                1) Restaurant
                2) Menu
                3) Meal
                4) OrderItem
                5) Order
                6) Customer
                7) Employee
                8) Shift
                9 Week
                10) Weekday

Your choice: 8
Give a value for ShiftID: Shift300
```

```
Choose table:
                1) Restaurant
                2) Menu
                3) Meal
                4) OrderItem
                5) Order
                6) Customer
                7) Employee
                8) Shift
                9 Week
                10) Weekday

Your choice: 9
Give a value for WeekID: Week50
Give a value for ShiftID: Shift300
Give a value for Weeknumber: 24
```

```
Choose table:
                1) Restaurant
                2) Menu
                3) Meal
                4) OrderItem
                5) Order
                6) Customer
                7) Employee
                8) Shift
                9 Week
                10) Weekday

Your choice: 10
Give a value for WeekdayID: Weekday20
Give a value for Weekday: Monday
Give a value for WeekID: Week50
Give a value for Start time: 08:00
Give a value for End time: 16:00
What would you like to do:
```

Next, I have program to display data from any of the tables:

```
def selectData():
    conn = sqlite3.connect("FinalProject.db");
    cursor = conn.cursor()
    tablename, columns = chooseTable()
    sql = "SELECT * FROM "+tablename
    cursor.execute(sql)
    data = cursor.fetchall()
    for row in data:
        print(row)
    conn.close()
```

Example for displaying different restaurants' data:

```
Your choice: 2
Choose table:
                1) Restaurant
                2) Menu
                3) Meal
                4) OrderItem
                5) Order
                6) Customer
                7) Employee
                8) Shift
                9 Week
                10) Weekday

Your choice: 1
('Rest123', 'Burger Place', 'Lappeenranta', '040111222', 'burgerplace@hotmail.com')
('Restaurant111', 'Pizza Place', 'Lappeenranta', '040555', 'pizzaplace@email.com')
```

Customers can order food with the order food program. The subprogram automatically updates the orders total price to account all the order items in one order. Code:

```
def orderFood():
    conn = sqlite3.connect("FinalProject.db")
    cursor = conn.cursor()
    print("What is your customer ID?")
    customerID = input("Your ID: ")
    cursor.execute("SELECT Name FROM Customer WHERE CustomerID = (?)", (customerID,))
    data = cursor.fetchone()

    try:
        print("Welcome "+data[0]+", which menu would you like to use?")
    except:
        print("This ID doesn't exist yet.")
        return 0
    menus = cursor.execute("SELECT Name FROM MENU")
    menudata = menus.fetchall()
    print(menudata)

    menuchoice = int(input("Select index (0,1,2...)): "))
    menuname = menudata[menuchoice]
    cursor.execute("SELECT MenuID FROM MENU WHERE Name = (?)",(menudata[menuchoice]))
    menuindex = cursor.fetchone()
    print("What would you like to order from "+menuname[0]+"?")
    cursor.execute("SELECT Name, Price,MealID FROM Meal WHERE MenuID = (?)",(menuindex))
    meals = cursor.fetchall()

    index = 0
    for row in meals:
        index += 1
        print(str(index)+". "+row[0]+" -- Price = "+str(row[1]))
    mealchoice = int(input("Your choice: "))-1
    meal = meals[mealchoice][0]
    price = meals[mealchoice][1]
    mealID = meals[mealchoice][2]
    print("You chose "+meal+" with a price of "+str(price)+" euros.")

    orderitemID = input("Give your order item ID: ")
    orderID = input("Give your order ID: ")
    cursor.execute("INSERT INTO OrderItem VALUES(?,?,?,?)",(orderitemID,mealID, orderID, price))
    cursor.execute("SELECT SUM(Price) FROM OrderItem WHERE OrderID = ?", (orderID,))
    totalprice = cursor.fetchone()
    cursor.execute("INSERT OR REPLACE INTO Orders VALUES(?,?,?)", (orderID,customerID,totalprice[0]))
    conn.commit()
    cursor.close()
    conn.close()
```

Example output:

```
What would you like to do:
        1. Add data
        2. Look at data
        3. Change data
        4. Order food (customer only)
        5. Look at customer orders
        6. Display Employee Hours
        0. Exit

Your choice: 4
What is your customer ID?
Your ID: Customer123
Welcome James, which menu would you like to use?
[('Summer Menu',), ('Spring Menu',)]
Select index (0,1,2...)): 1
What would you like to order from Spring Menu?
1. Meatballs&Spaghetti -- Price = 10
Your choice: 1
You chose Meatballs&Spaghetti with a price of 10 euros.
Give your order item ID: Item123
Give your order ID: Order111
```

Then we can look at specific customers orders, lets use the Customer123 we have created.

This is the code for displaying customers' orders:

```
def displayOrders():
    conn = sqlite3.connect("FinalProject.db")
    cursor = conn.cursor()
    print("What is your customer ID?")
    customerID = input("Your ID: ")
    cursor.execute("SELECT OrderID, Price FROM Orders WHERE CustomerID = (?)", (customerID,))
    data = cursor.fetchall()
    try:
        testi = data[0]
    except:
        print("This ID doesn't exist yet.")
        return 0
    for order in data:
        print("You have ordered following items in order "+order[0]+": ")
        cursor.execute("SELECT OrderItemID, Price, MealID FROM OrderItem WHERE OrderID = ?", (order[0],))
        orderitems = cursor.fetchall()
        for orderitem in orderitems:
            cursor.execute("SELECT Name, Price FROM Meal WHERE MealID = ?",(orderitem[2],))
            meal = cursor.fetchone()
            print("Order item: "+orderitem[0]+" --- "+meal[0]+" --- Price: "+str(meal[1])+" euros.")
        print("Total price in order: "+order[0]+" is "+str(order[1])+" euros.")

def displayEmployeeHours():
```

Example output:

```
What would you like to do:
        1. Add data
        2. Look at data
        3. Change data
        4. Order food (customer only)
        5. Look at customer orders
        6. Display Employee Hours
        0. Exit

Your choice: 5
What is your customer ID?
Your ID: Customer123
You have ordered following items in order Order111:
Order item: Item123 --- Meatballs&Spaghetti --- Price: 10 euros.
Total price in order: Order111 is 10 euros.
```

If there are multiple orders for one customer (Customer ID 123 existed already) it will look like tihs:

```
What would you like to do:
        1. Add data
        2. Look at data
        3. Change data
        4. Order food (customer only)
        5. Look at customer orders
        6. Display Employee Hours
        0. Exit

Your choice: 5
What is your customer ID?
Your ID: 123
You have ordered following items in order Order222:
Order item: Item3 --- Fish&Chips --- Price: 6 euros.
Order item: Item4 --- Hamburger --- Price: 8 euros.
Total price in order: Order222 is 14 euros.
You have ordered following items in order Order123:
Order item: Item1 --- Pizza --- Price: 10 euros.
Order item: Item2 --- Hamburger --- Price: 8 euros.
Order item: Item111 --- Meatballs&Spaghetti --- Price: 10 euros.
Order item: Item100 --- Hamburger --- Price: 8 euros.
Total price in order: Order123 is 36 euros.
```

Next we have subprogram to display Employees hours:

```
def displayEmployeeHours():
    conn = sqlite3.connect("FinalProject.db")
    cursor = conn.cursor()
    print("What is your Employee ID?")
    empID = input("Your ID: ")
    cursor.execute("SELECT ShiftID, First Name, Last Name FROM Employee WHERE EmployeeID = (?)", (empID,))
    data = cursor.fetchall()
    try:
        testi = data[0]
    except:
        print("This ID doesn't exist yet.")
        return 0
    print("Employees "+data[0][1]+" "+data[0][2]+" working hours: ")
    cursor.execute("SELECT WeekID, Weeknumber FROM Weeks WHERE ShiftID = (?)", (data[0][0],))
    weekdata = cursor.fetchall()
    for week in weekdata:
        print("Weeks "+str(week[1])+" working hours: ")
        cursor.execute("SELECT Weekday, Start time, End time FROM Weekday WHERE WeekID = (?)", (week[0],))
        days = cursor.fetchall()
        for day in days:
            print("Weekday: "+day[0]+" --- Start time: "+day[1]+" --- End time: "+day[2])

def mainmenu():
```

Lets look at EmpID200:s working hours:

```
What would you like to do:
        1. Add data
        2. Look at data
        3. Change data
        4. Order food (customer only)
        5. Look at customer orders
        6. Display Employee Hours
        0. Exit

Your choice: 6
What is your Employee ID?
Your ID: EmpID200
Employees Donald Duck working hours:
Weeks 24 working hours:
Weekday: Monday --- Start time: 08:00 --- End time: 16:00
```

Lets add rest of the weeks days to week 24 (Week ID: Week50).

This is what empID200s work week 24 looks like now:

```
What would you like to do:
        1. Add data
        2. Look at data
        3. Change data
        4. Order food (customer only)
        5. Look at customer orders
        6. Display Employee Hours
        0. Exit

Your choice: 6
What is your Employee ID?
Your ID: EmpID200
Employees Donald Duck working hours:
Weeks 24 working hours:
Weekday: Monday --- Start time: 08:00 --- End time: 16:00
Weekday: Tuesday --- Start time: 08:00 --- End time: 16:00
Weekday: Wednesday --- Start time: 08:00 --- End time: 16:00
Weekday: Thursday --- Start time: 08:00 --- End time: 16:00
Weekday: Friday --- Start time: 08:00 --- End time: 16:00
```

Now we can use change data subprogram to change Mondays and Fridays start time and end time.

Change data subprograms code:

```
def changeData():
    conn = sqlite3.connect("FinalProject.db");
    c = conn.cursor()
    tablename, columns = chooseTable()
    print("Choose variable to update to from (index: 0,1,2...): ")
    print(columns)
    updatevar = int(input("Your choice: "))
    updatevar = columns[updatevar]
    print(updatevar)
    restrict = int(input("Choose the variable you want to restrict results with from: "))
    restrict = columns[restrict]
    c.execute("SELECT "+restrict+" FROM "+tablename)
    print(c.fetchall())
    restrictvalue = input("What is the restrict variables value:  ")
    newvalue = input("What do you want the new value to be: ")
    c.execute("UPDATE "+tablename+" SET "+updatevar+" = ? WHERE "+restrict+" = ?", (newvalue,restrictvalue))
    print("Updated table "+tablename+" where "+restrict+" is "+restrictvalue+". Updated "+updatevar+" to "+newvalue+".")
    conn.commit()
    conn.close()
```

Change EmpID200s Mondays working hours to 10.00-18.00:

```
Choose table:
                1) Restaurant
                2) Menu
                3) Meal
                4) OrderItem
                5) Order
                6) Customer
                7) Employee
                8) Shift
                9 Week
                10) Weekday

Your choice: 10
Choose variable to update to from (index: 0,1,2...):
['WeekdayID', 'Weekday', 'WeekID', 'StartTime', 'EndTime']
Your choice: 3
Choose the variable you want to restrict results with from: 0
[('Weekday20',), ('Weekday21',), ('Weekday22',), ('Weekday23',), ('Weekday24',)]
What is the restrict variables value:  Weekday20
What do you want the new value to be: 10:00
Updated table Weekday where WeekdayID is Weekday20. Updated StartTime to 10:00.
```

and

```
Choose table:
                1) Restaurant
                2) Menu
                3) Meal
                4) OrderItem
                5) Order
                6) Customer
                7) Employee
                8) Shift
                9 Week
                10) Weekday

Your choice: 10
Choose variable to update to from (index: 0,1,2...):
['WeekdayID', 'Weekday', 'WeekID', 'StartTime', 'EndTime']
Your choice: 4
Choose the variable you want to restrict results with from: 0
[('Weekday20',), ('Weekday21',), ('Weekday22',), ('Weekday23',), ('Weekday24',)]
What is the restrict variables value:  Weekday20
What do you want the new value to be: 18:00
Updated table Weekday where WeekdayID is Weekday20. Updated EndTime to 18:00.
```

Same for the Friday.

```
Choose variable to update to from (index: 0,1,2...):
['WeekdayID', 'Weekday', 'WeekID', 'StartTime', 'EndTime']
Your choice: 3
Choose the variable you want to restrict results with from: 0
[('Weekday20',), ('Weekday21',), ('Weekday22',), ('Weekday23',), ('Weekday24',)]
What is the restrict variables value:  Weekday24
What do you want the new value to be: 10:00
Updated table Weekday where WeekdayID is Weekday24. Updated StartTime to 10:00.
What would you like to do:
        1. Add data
        2. Look at data
        3. Change data
        4. Order food (customer only)
        5. Look at customer orders
        6. Display Employee Hours
        0. Exit

Your choice: 3
Choose table:
            1) Restaurant
            2) Menu
            3) Meal
            4) OrderItem
            5) Order
            6) Customer
            7) Employee
            8) Shift
            9 Week
            10) Weekday

Your choice: 10
Choose variable to update to from (index: 0,1,2...):
['WeekdayID', 'Weekday', 'WeekID', 'StartTime', 'EndTime']
Your choice: 4
Choose the variable you want to restrict results with from: 0
[('Weekday20',), ('Weekday21',), ('Weekday22',), ('Weekday23',), ('Weekday24',)]
What is the restrict variables value:  Weekday24
What do you want the new value to be: 18:00
Updated table Weekday where WeekdayID is Weekday24. Updated EndTime to 18:00.
```

This is what EmpID200:s shift for the week 24 looks like now:

```
What is your Employee ID?
Your ID: EmpID200
Employees Donald Duck working hours:
Weeks 24 working hours:
Weekday: Monday --- Start time: 10:00 --- End time: 18:00
Weekday: Tuesday --- Start time: 08:00 --- End time: 16:00
Weekday: Wednesday --- Start time: 08:00 --- End time: 16:00
Weekday: Thursday --- Start time: 08:00 --- End time: 16:00
Weekday: Friday --- Start time: 10:00 --- End time: 18:00
What would you like to do:
```