

Предмет	МДК 03.01 ТРПО
Дата	12.12.2020
Группа	474
Студенты	Цветков М. В.
Работа	№ 6

1. Настройка интерфейса.

Интерфейс первого окна, в котором находится list. Интерфейс второго окна состоит из выпадающего списка и двух кнопок. (рисунок 1).

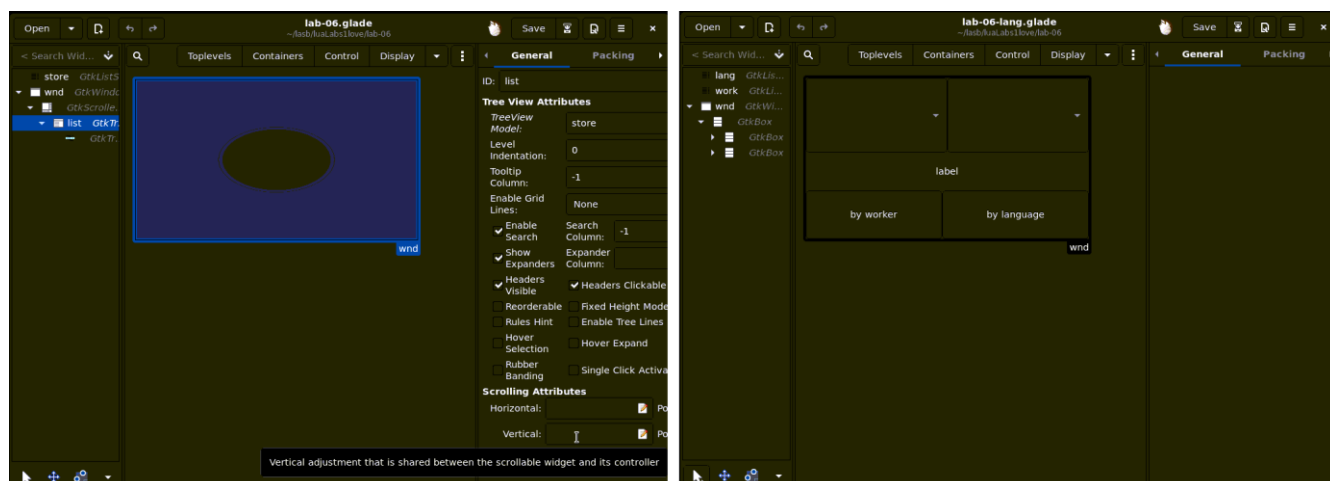


Рисунок 1 – Интерфейс программы.

2. Функционал приложения

Для запуска предлагается 2 варианта приложения: на рисунке 2 приложение выводит список из базы SQL, а так же то же самое с помощью дочернего процесса `io.popen()`.

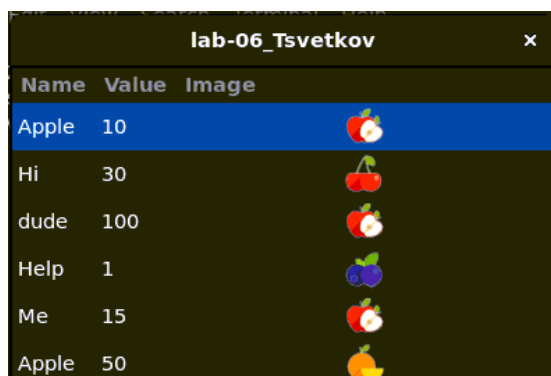


Рисунок 2 –Рабочая программа.

На рисунке 3 приложение используется для ведения учета за сотрудниками фирмы. При выборе сотрудника, показывается, какими языками программирования он владеет, а при выборе языка, показывается, кто им владеет.

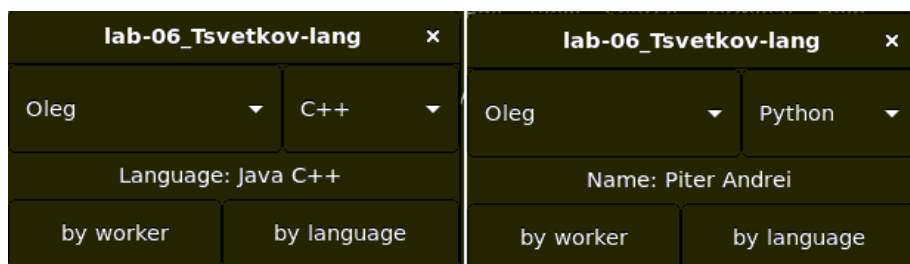


Рисунок 3 –Рабочая программа.

3. Код программы

За первое окно отвечают 2 кода: На рисунке 4.

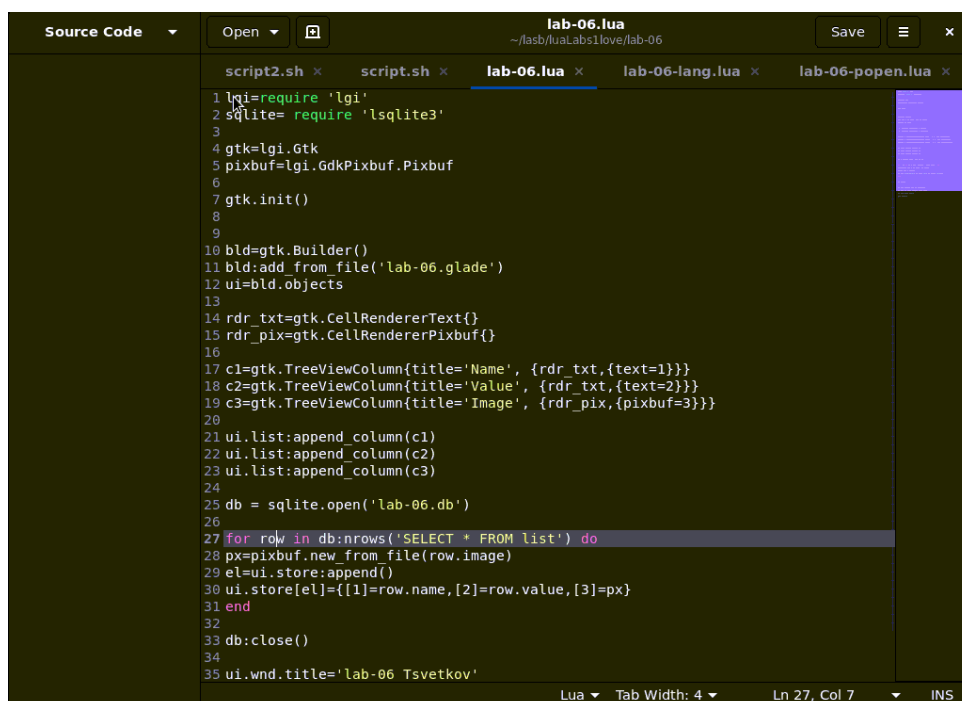
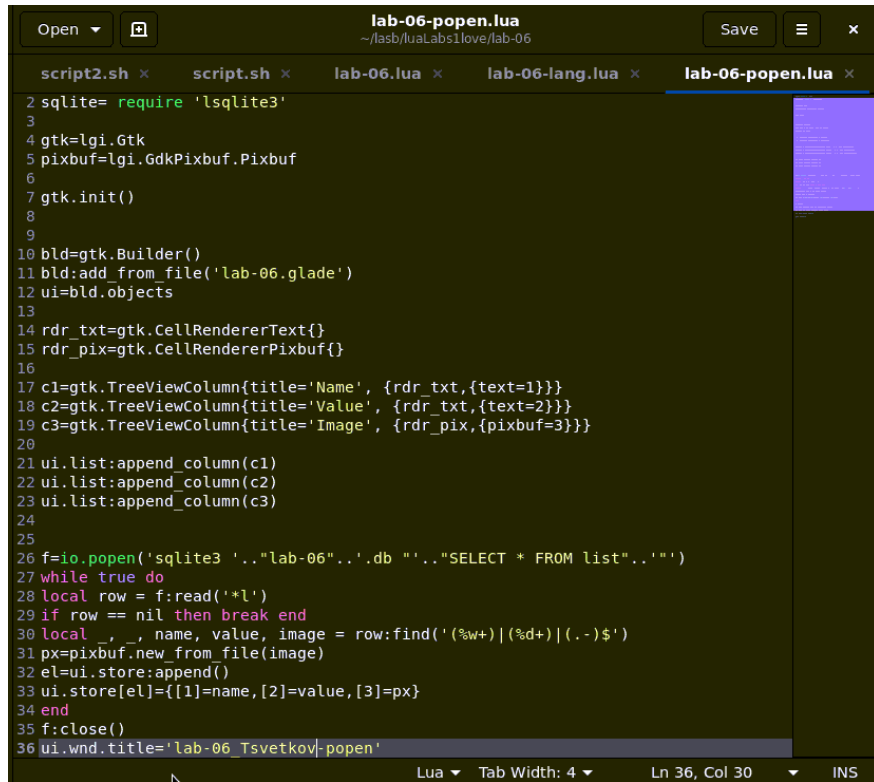


Рисунок 4 –Код lua.

И на рисунке 5

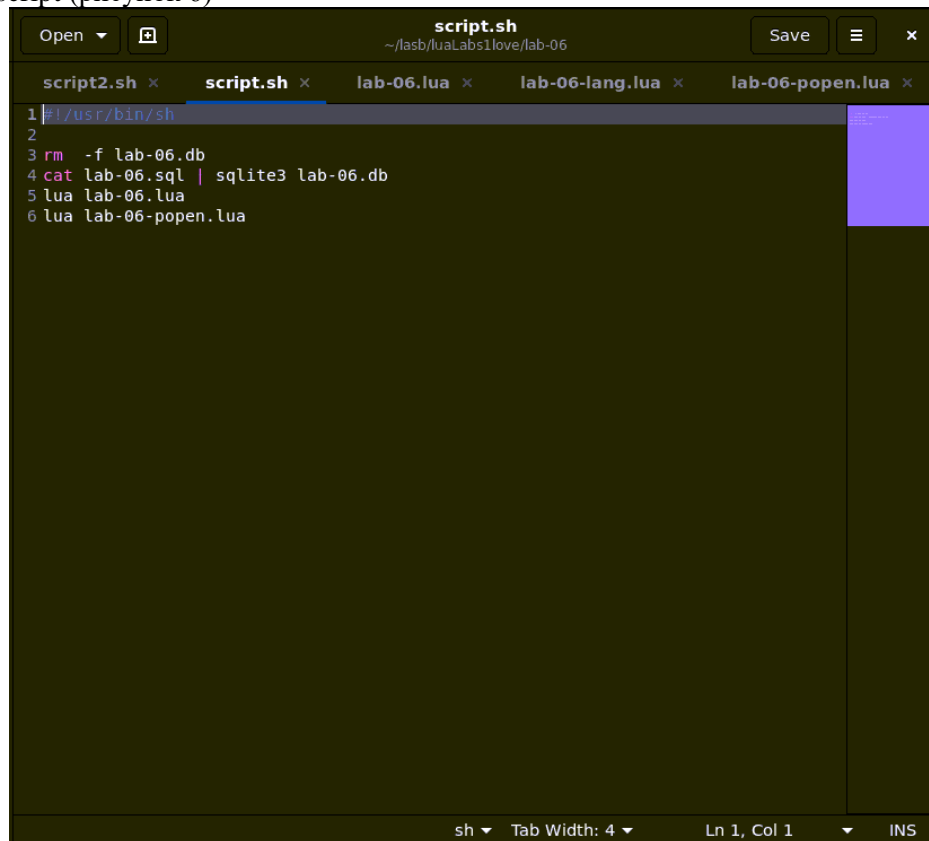


The screenshot shows a code editor with the file 'lab-06-popen.lua' open. The code is written in Lua and uses the Gtk library for a graphical interface. It includes a SQLite database connection and a loop that reads data from the database and displays it in a tree view. The code is as follows:

```
2 sqlite= require 'sqlite3'
3
4 gtk=lg1.Gtk
5 pixbuf=lg1.GdkPixbuf.Pixbuf
6
7 gtk.init()
8
9
10 bld=gtk.Builder()
11 bld:add_from_file('lab-06.glade')
12 ui=bld.objects
13
14 rdr_txt=gtk.CellRendererText{}
15 rdr_pix=gtk.CellRendererPixbuf{}
16
17 c1=gtk.TreeViewColumn{title='Name', {rdr_txt,{text=1}}}
18 c2=gtk.TreeViewColumn{title='Value', {rdr_txt,{text=2}}}
19 c3=gtk.TreeViewColumn{title='Image', {rdr_pix,{pixbuf=3}}}
20
21 ui.list:append_column(c1)
22 ui.list:append_column(c2)
23 ui.list:append_column(c3)
24
25
26 f=io.popen('sqlite3 '.."lab-06"..'.db "'..'SELECT * FROM list"..''')
27 while true do
28   local row = f:read('*l')
29   if row == nil then break end
30   local _, _, name, value, image = row:find('(w+)|(%d+)|(.)$')
31   px=pixbuf.new_from_file(image)
32   el=ui.store:append()
33   ui.store[el]={[1]=name,[2]=value,[3]=px}
34 end
35 f:close()
36 ui.wnd.title='lab-06 Tsvetkov-popen'
```

Рисунок 5 – Код lua

А также shell-script (рисунок 6)

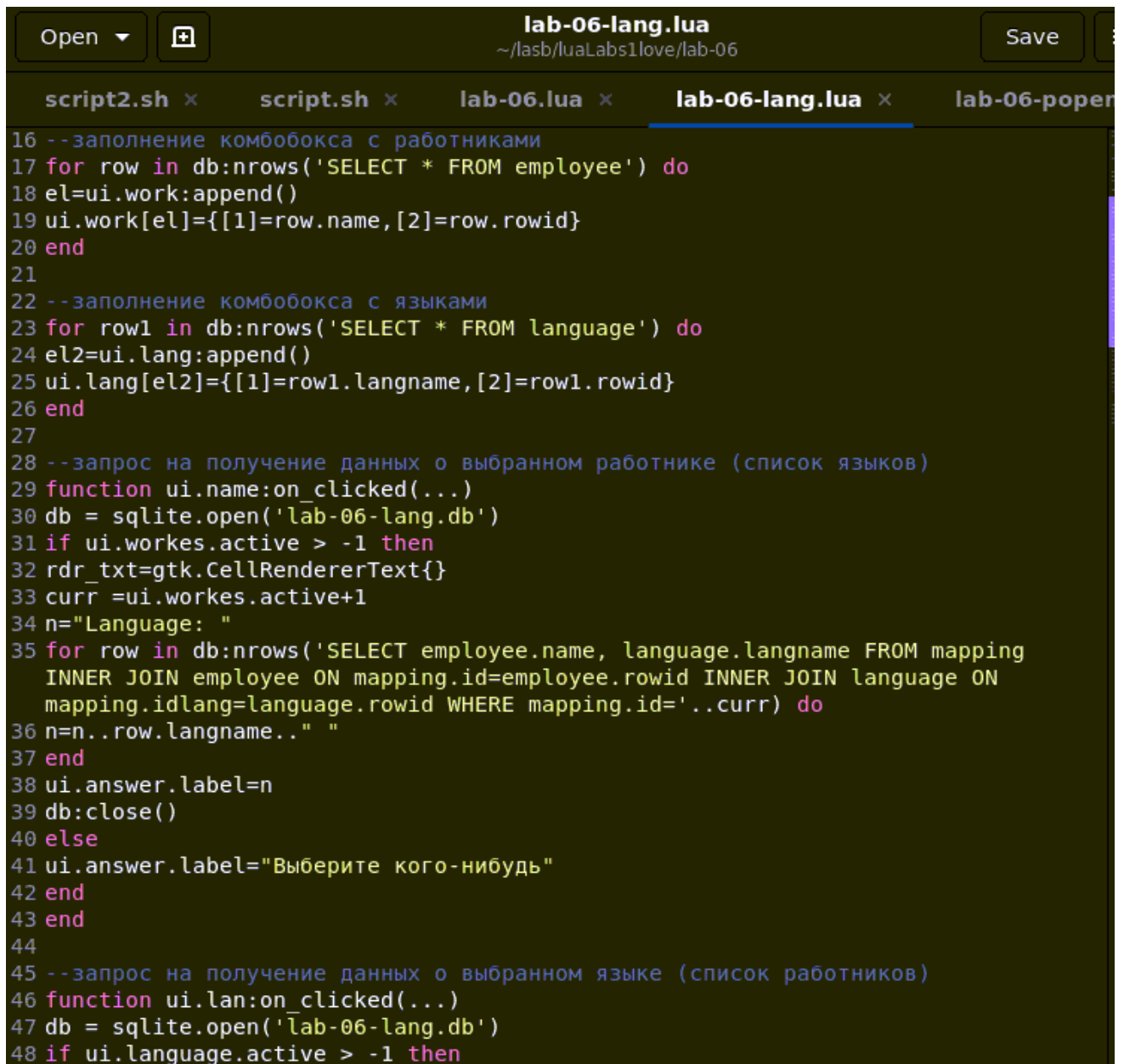


The screenshot shows a code editor with the file 'script.sh' open. The script is a shell script that performs several operations: it removes a file 'lab-06.db', concatenates the contents of 'lab-06.sql' into 'lab-06.db', and then runs 'lab-06.lua' and 'lab-06-popen.lua'. The script is as follows:

```
1 #!/usr/bin/sh
2
3 rm -f lab-06.db
4 cat lab-06.sql | sqlite3 lab-06.db
5 lua lab-06.lua
6 lua lab-06-popen.lua
```

Рисунок 6 – shell-script

За второе окно отвечает код на рисунке 7.



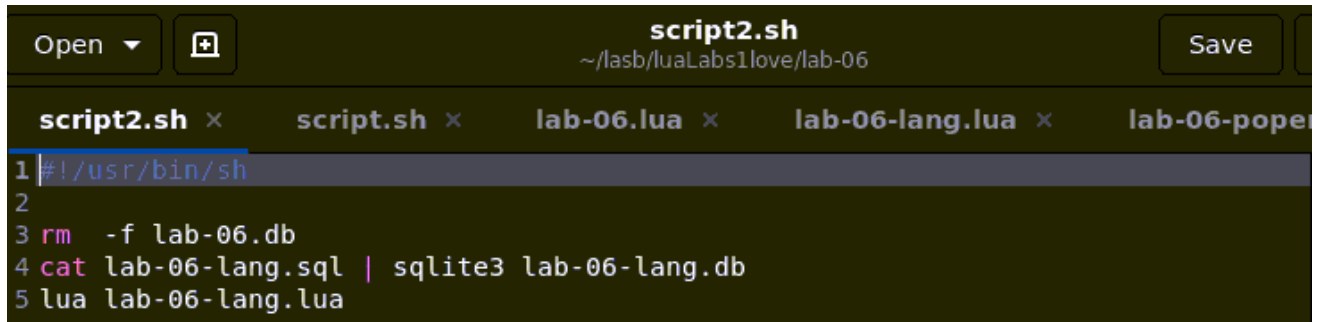
```
lab-06-lang.lua
~/lab/luaLabs1love/lab-06

script2.sh x  script.sh x  lab-06.lua x  lab-06-lang.lua x  lab-06-popen

16 --заполнение комбобокса с работниками
17 for row in db:nrows('SELECT * FROM employee') do
18 el=ui.work:append()
19 ui.work[el]={[1]=row.name,[2]=row.rowid}
20 end
21
22 --заполнение комбобокса с языками
23 for row1 in db:nrows('SELECT * FROM language') do
24 el2=ui.lang:append()
25 ui.lang[el2]={[1]=row1.langname,[2]=row1.rowid}
26 end
27
28 --запрос на получение данных о выбранном работнике (список языков)
29 function ui.name:on_clicked(...)
30 db = sqlite.open('lab-06-lang.db')
31 if ui.workes.active > -1 then
32 rdr_txt=gtk.CellRendererText{}
33 curr =ui.workes.active+1
34 n="Language: "
35 for row in db:nrows('SELECT employee.name, language.langname FROM mapping
INNER JOIN employee ON mapping.id=employee.rowid INNER JOIN language ON
mapping.idlang=language.rowid WHERE mapping.id='..curr) do
36 n=n..row.langname.." "
37 end
38 ui.answer.label=n
39 db:close()
40 else
41 ui.answer.label="Выберите кого-нибудь"
42 end
43 end
44
45 --запрос на получение данных о выбранном языке (список работников)
46 function ui.lan:on_clicked(...)
47 db = sqlite.open('lab-06-lang.db')
48 if ui.language.active > -1 then
```

Рисунок 7 – Код lua для второго окна

А также script-shell (рисунок 8).



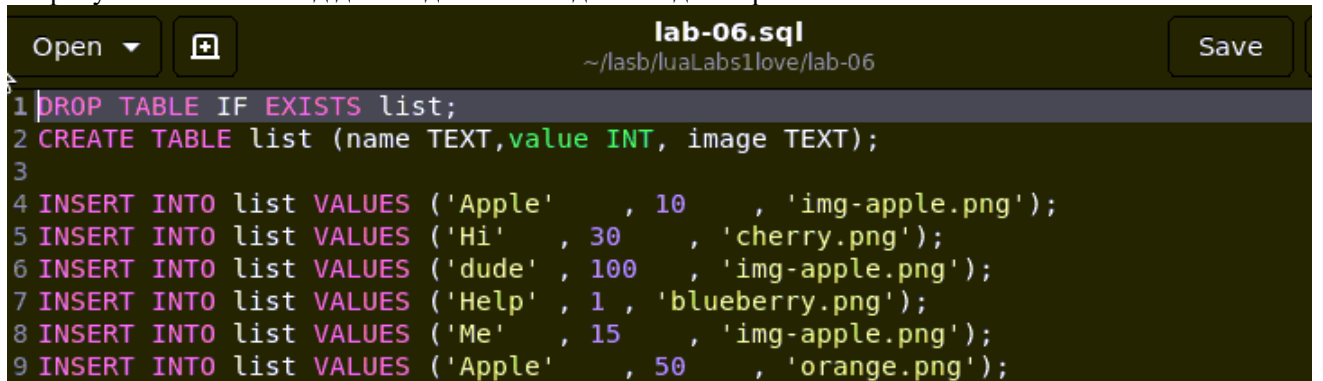
```
script2.sh
~/lasb/luabsllove/lab-06

1 #!/usr/bin/sh
2
3 rm -f lab-06.db
4 cat lab-06-lang.sql | sqlite3 lab-06-lang.db
5 lua lab-06-lang.lua
```

Рисунок 8 – script-shell для второго окна

4. Код базы данных

На рисунке 9 показан код для создания базы данных для первого окна.

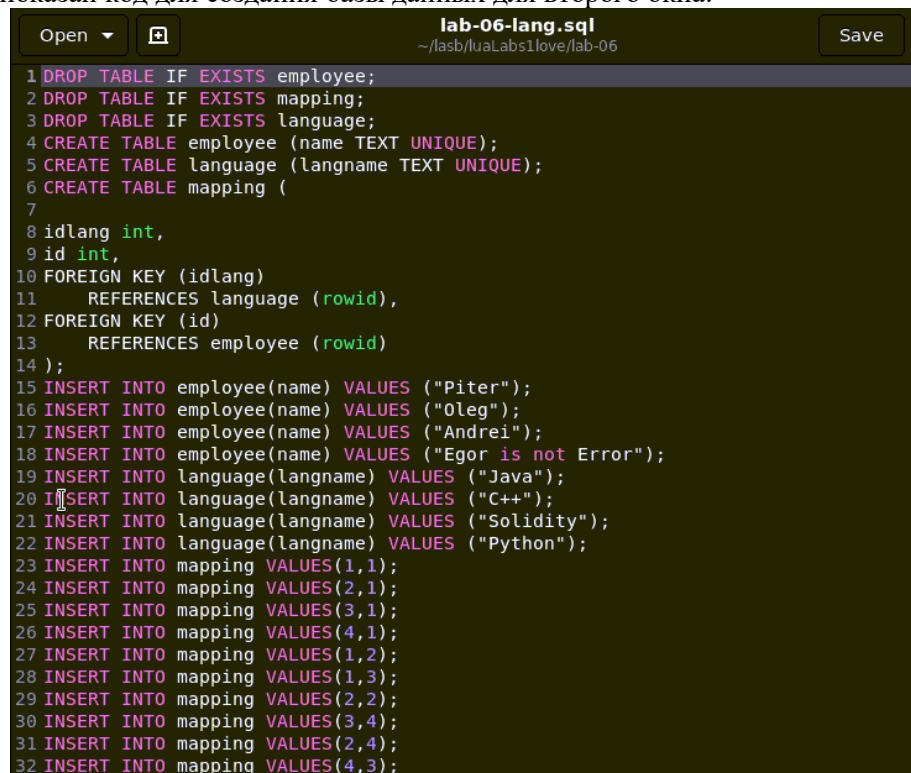


```
lab-06.sql
~/lasb/luabsllove/lab-06

1 DROP TABLE IF EXISTS list;
2 CREATE TABLE list (name TEXT,value INT, image TEXT);
3
4 INSERT INTO list VALUES ('Apple' , 10 , 'img-apple.png');
5 INSERT INTO list VALUES ('Hi' , 30 , 'cherry.png');
6 INSERT INTO list VALUES ('dude' , 100 , 'img-apple.png');
7 INSERT INTO list VALUES ('Help' , 1 , 'blueberry.png');
8 INSERT INTO list VALUES ('Me' , 15 , 'img-apple.png');
9 INSERT INTO list VALUES ('Apple' , 50 , 'orange.png');
```

Рисунок 9 – Скрипт создания таблицы

На рисунке 10 показан код для создания базы данных для второго окна.



```
lab-06-lang.sql
~/lasb/luabsllove/lab-06

1 DROP TABLE IF EXISTS employee;
2 DROP TABLE IF EXISTS mapping;
3 DROP TABLE IF EXISTS language;
4 CREATE TABLE employee (name TEXT UNIQUE);
5 CREATE TABLE language (langname TEXT UNIQUE);
6 CREATE TABLE mapping (
7
8 idlang int,
9 id int,
10 FOREIGN KEY (idlang)
11 REFERENCES language (rowid),
12 FOREIGN KEY (id)
13 REFERENCES employee (rowid)
14 );
15 INSERT INTO employee(name) VALUES ("Piter");
16 INSERT INTO employee(name) VALUES ("Oleg");
17 INSERT INTO employee(name) VALUES ("Andrei");
18 INSERT INTO employee(name) VALUES ("Egor is not Error");
19 INSERT INTO language(langname) VALUES ("Java");
20 INSERT INTO language(langname) VALUES ("C++");
21 INSERT INTO language(langname) VALUES ("Solidity");
22 INSERT INTO language(langname) VALUES ("Python");
23 INSERT INTO mapping VALUES(1,1);
24 INSERT INTO mapping VALUES(2,1);
25 INSERT INTO mapping VALUES(3,1);
26 INSERT INTO mapping VALUES(4,1);
27 INSERT INTO mapping VALUES(1,2);
28 INSERT INTO mapping VALUES(1,3);
29 INSERT INTO mapping VALUES(2,2);
30 INSERT INTO mapping VALUES(3,4);
31 INSERT INTO mapping VALUES(2,4);
32 INSERT INTO mapping VALUES(4,3);
```

Рисунок 10 скрипт создания базы данных для второго окна