

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

09.03.04 – Программная инженерия

Отчет о прохождении производственной практики

Разработка модуля «Отчетность студентов» веб сервера ИМИТ. База данных

Выполнил студент группы 22307

Лисицкий Олег Витальевич

Направление подготовки:

Системное и прикладное программное обеспечение

Место прохождения практики:

Кафедра ИМО, ЦИИ

Сроки прохождения практики:

30.05-09.06

Руководитель практики:

к.т.н., доцент

Богоявленская Ольга Юрьевна

Оценка _____

Дата _____

Петрозаводск — 2023

Перечень терминов

В ходе описания отчёта о практике по научно-исследовательской работе будут использоваться следующие термины:

- ПК – персональный компьютер;
- AutoReport - реализуемое web-приложение;
- парсинг - преобразование данных в нужный вид
- БД - база данных

Оглавление

Перечень терминов	2
Введение	4
1 Проектирование	5
1.1 Функциональные требования	5
2 Подсистема работы с БД	7
2.1 Проект основных структур данных	7
2.2 Проектирование модуля curriculum_importer.py	9
3 Реализация	11
3.1 Инструменты	11
3.2 Создание БД	12
3.3 Реализация модуля curriculum_importer.py	13
Заключение	21
Литература	22

Введение

Цель практики: закрепление и углубление теоретических знаний, полученных в процессе обучения, приобретение навыков и опыта практической работы по разработке веб-приложений. Локальное развёртывание БД и разработка парсера данных xls-файлов для реализации модуля "Отчетность студентов" веб-сервера ИМИТ.

Задачи практики:

1. Разработать проект основных структур данных БД
2. Разработать парсер данных из файлов учебных планов в БД
3. Реализовать локальное развёртывание БД
4. Реализовать парсер данных из файлов учебных планов в БД

Сайт ИМИТ ПетрГУ существует уже много лет и включает в себя много функций, делаая жизнь студентов и преподавателей проще. Сайт продолжает развиваться, его функциональность расширяется для того, чтобы удовлетворять новые потребности его пользователей.

Тема данной научно-исследовательской работы - это разработка в интересах дирекции института нового модуля «Отчетность студентов» веб сервера ИМИТ для автоматизации генерации отчётов об аккредитации студентов.

Глава 1

Проектирование

AutoReport — сервис автоматизации формирования отчетов. Является расширением функциональности сайта ИМИТ ПетрГУ.

Для импорта таблиц Excel в БД существуют следующие программные решения

- Использование SQL Server Import and Export Wizard — инструмента SQL Server, позволяющий произвести импорт данных из таблиц в БД
- SQL Server Integration Services (SSIS) — это компонент программного обеспечения базы данных Microsoft SQL Server, который можно использовать для выполнения широкого круга задач переноса данных.
- Azure Data Factory — облачная служба ETL Azure для масштабируемой бессерверной интеграции и преобразования данных. Она предлагает пользовательский интерфейс без кода для интуитивно понятного создания и мониторинга и управления с одной панели.

Данные решения не наблюдаются возможным внедрить в существующую архитектуру сайта ИМИТ с требуемым к сервису перечнем функциональных требований, с которыми можно познакомиться в следующем параграфе.

1.1 Функциональные требования

Все функции сервиса доступны только ограниченному кругу авторизованных пользователей.

Список функциональных требований для разрабатываемого веб-приложения:

1. Сервис должен предоставлять пользователям возможность загружать со своего персонального компьютера отчёт с предварительным выбором года и семестра (осенний / весенний) для дальнейшей работы с ним;
2. Сервис должен добавлять данные из файлов учебных планов в базу данных;
3. Сервис должен предоставлять пользователям возможность формировать отчёт;
4. Сервис должен предоставлять пользователям возможность самостоятельно проверить и исправить сформированный ранее отчёт;
5. Сервис должен предоставлять пользователям возможность конвертировать итоговый отчёт и добавлять его на сервер ИМИТ;
6. Сервис должен обновлять состояние каждого отчёта за определённый год и семестр.

У разрабатываемого веб-приложения следующий список ограничений:

1. Пользователи должны иметь доступ к сервису только с персонального компьютера. Доступ к сервису со всех остальных устройств должен быть ограничен;
2. Сервис представляет собой расширение функциональности сайта ИМИТ;
3. Сервис реализуется в виде web-приложения;
4. Сервис написан на языке программирования Python с использованием фреймворка Flask;
5. Сервис должен использовать базу данных заказчика (MariaDB);
6. Пользователями сервиса являются авторизованные на сайте ИМИТ пользователи. Остальные пользователи не должны иметь доступ к сервису;
7. Пользовательский интерфейс сервиса на русском языке;
8. Переход на страницу сервиса должен происходить по ссылке на странице сайта ИМИТ «Отчётность студентов»; Файлы учебных планов в формате .xls содержатся в файловой системе на сервере ИМИТ;
9. Сервис определяет местоположение необходимых файлов учебных планов с помощью конфигурационного файла.

Глава 2

Подсистема работы с БД

Подсистема работы с БД должна выполнять следующие задачи:

1. чтение данных из файлов учебных планов
2. добавление данных учебных планов в БД
3. составление запросов для получения данных из БД
4. проставление и поиск статусов формирования отчётов в БД

В рамках НИР были реализовано выполнение первых двух пунктов в модуле импорта `curriculum_importer.py`

2.1 Проект основных структур данных

Для того, чтобы составить проект основных структур данных, необходимо ознакомиться со структурой исходных файлов учебных планов. После анализа предоставленных примеров таких файлов, была сформулирована следующая задача для проектирования БД и структур данных:

Имеется набор из 4 файлов формата `.xls` с учебными планами учебных групп (1 файл на 1 группу), содержащих по 4 листа в каждом: учебный план 1-4 курса учебной группы. В строке таблицы учебного плана помимо нерелевантных данных указывается блок дисциплин (по выбору, практика, просто предмет), к которому относится предмет, название предмета, тип аккредитации в семестре (два набора по 3 ячейки (экзамен зачет, дифференцированный зачёт) для 2 семестров), в котором вписан либо символ «х», либо ячейка пуста. Требуется создать базу данных и найти способ передачи в неё данных об учебных направлениях и предметах, которые они изучают в каждом семестре. БД также должна

хранить следующую информацию о сформированных отчётах: год отчёта, семестр отчёта, состояние отчёта (0 - «сформирован»; 1 - «добавлен»), путь к отчёту в файловой системе. БД должна быть создана в СУБД MariaDB.

9.03.04 - Программная инженерия														
Часть УП	Наименование	Кафедра	1 семестр				Неделя ТО				2 семестр			
			кол-во недель	лекции	практ.	лаб.	Всего	Экз	Зач	ДЗ	КР	КП	кол-во недель	лекции и
Б1.Б	Физическая культура и спорт	О						x						
Б1.Б	Линейная алгебра	МА						x	x					
Б1.Б	Дискретная математика	ПМИК						x						
Б1.Б	История России	ОИ							x					
Б1.Б	Иностранный язык	ИЯЯз							x					x
Б1.Б	Математический анализ	МА						x	x					x
Б1.Б	Основы информатики и программирования	ИМО						x						x
Б1.Б	Геометрия	МА												x
Б1.Б	Введение в архитектуру ЭВМ	ИМО												x
Б1.Б	Операционные системы и оболочки	ИМО												x
Б1.Б	Офисные технологии	ПМИК							x					
Б1.В.ОД	Основы проектирования ПО	ИМО												x
Б1.В.ОД	Основы тестирования ПО	ИМО												x
Б1.Б	Безопасность жизнедеятельности	КБЖ							x					
Б1.Б	Всеобщая история	ОИ												x

Рис. 2.1 – Вид таблицы одного курса исходного xls-файла учебного плана

Создание необходимой базы данных:

Так как мы имеем дело с электронными таблицами, наиболее оптимальным решением будет использование реляционной базы ведь данные в них хранятся в виде таблиц и строк. После анализа содержимого таблиц из файлов учебных планов, была предложена следующая структура базы данных:

Поле	Тип данных	Ключ	Описание
subjectId	int(11)	PK	Идентификатор учебной дисциплины
subjectName	varchar(200)	MUL	Название учебной дисциплины
accred	varchar(12)	MUL	Вид аккредитации (экзамен, зачёт, дифф. зачёт)
semester	tinyint(1)	MUL	номер семестра
pract	tinyint(1)		Флаг принадлежности дисциплины к практикам

Таблица 2.1 – tbl_subjects - информация о предметах

Поле	Тип данных	Ключ	Описание
dirId	int(11)	PK	Идентификатор направления подготовки
dirName	varchar(50)	MUL	Название(код) направления подготовки
adyear	int(11))	MUL	Год поступления

Таблица 2.2 – tbl_directions - информация о направлениях подготовки

Поле	Тип данных	Ключ	Описание
linkId	int(11)	PK	Идентификатор связи
dirId	int(11)	PK	Идентификатор направления подготовки
subjectId	int(11)	PK	Идентификатор учебной дисциплины

Таблица 2.3 – tbl_links - информация о связях учебных дисциплин и направлений подготовки

Поле	Тип данных	Ключ	Описание
repId	int(11)	PK	Идентификатор отчёта
repYear	int(11)		Год отчёта
repSem	int(11)		Семестр отчёта)
repStatus	tinyint(1)		Состояние отчёта (0 - «сформирован»; 1 - «добавлен»)
repPath	varchar(1000)		Путь к отчёту в файловой системе

Таблица 2.4 – tbl_reports - информация о состоянии отчётов

2.2 Проектирование модуля curriculum_importer.py

Была выделена следующая последовательность основных шагов выполнения работы модуля:

1. Установление подключения к БД
2. Чтение xls-файла учебного плана, на этом же этапе получаем название направления, его код и дату поступления группы, передаем их таблицу tbl_directions
3. Чтение таблицы для учебного плана одного курса сверху вниз, если она существует, обрабатывая случаи, когда в ячейки таблицы Excel на месте названия встречается непустое значение
4. Передача данных о дисциплинах в БД
5. Прекращение соединения после выполнения работы программы

Представив возможную реализацию модуля была составлена следующая блок схема работы модуля:

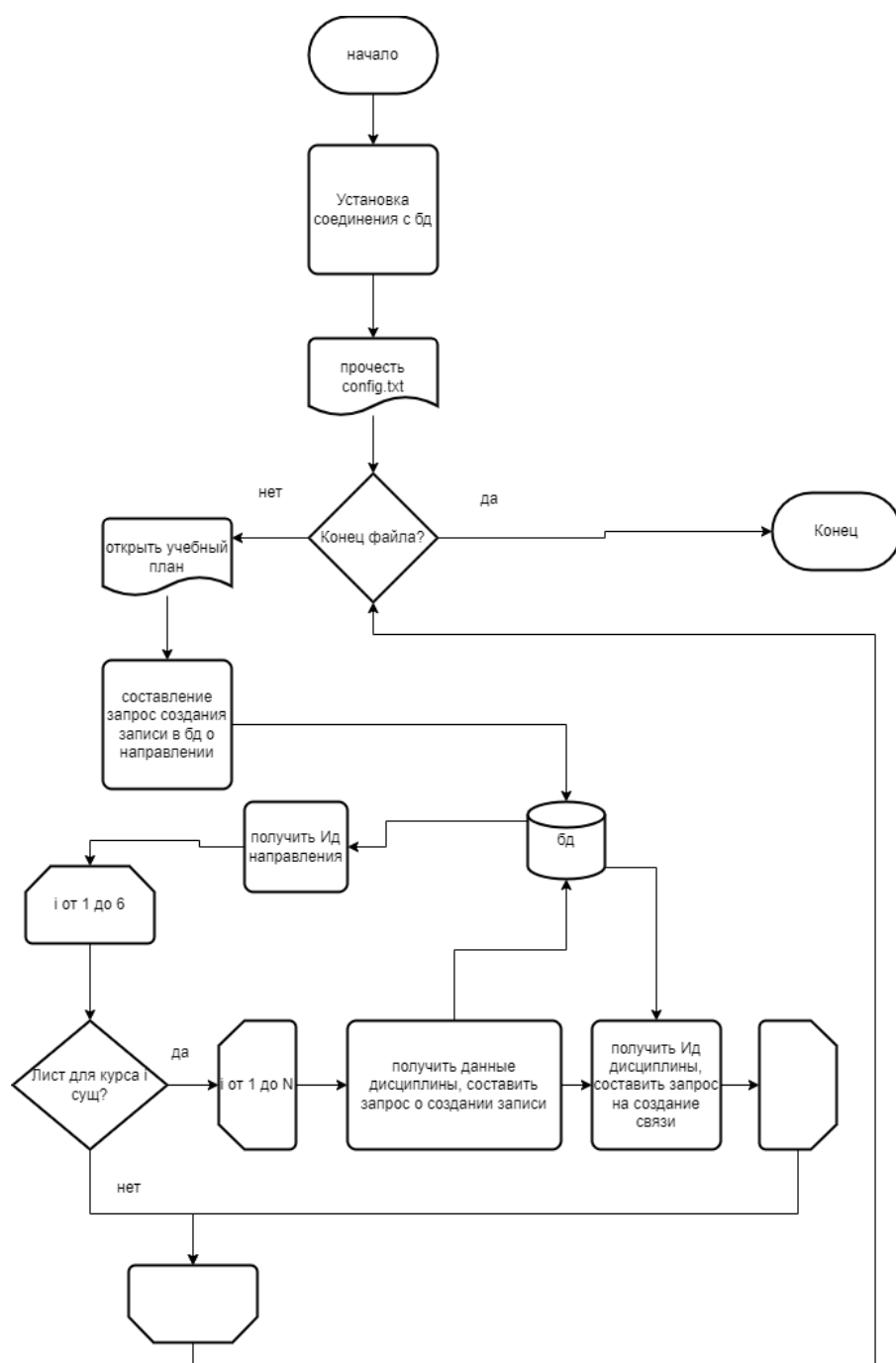


Рис. 2.2 – Блок-схема работы модуля curriculum_importer.py

Глава 3

Реализация

На ПК была локально развернута БД с именем db в СУБД MariaDB, содержащей таблицы, описанных во второй главе данного отчёта. Также был реализован модуль curriculum_importer.py на языке программирования Python, способный подключаться к БД, отправлять ей SQL-запросы, передавая и получая данные из БД.

3.1 Инструменты

Для реализации части кодирования в проекте были оспользованы такие инструменты, как:

- Python - скриптовый язык программирования. Для разработки модуля curriculum_importer.py использовалась библиотека для чтения xls-файлов xlrd и библиотека MariaDB Connector/Python для установления подключения к БД в СУБД MariaDB и взаимодействия с ней
- MariaDB - тветвление от системы управления базами данных MySQL, разрабатываемое сообществом под лицензией GNU GPL.
- SQL - декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.
- DBeaver - клиентское программное приложение SQL и инструмент администрирования базы данных.

3.2 Создание БД

Для начала требуется создать пустую базу данных с именем db в СУБД MariaDb. Для этого достаточно прописать следующий запрос:

Листинг 3.1 – создание пустой базы данных language

```
1 CREATE DATABASE `db`
```

Далее создаем в БД требуемые таблицы:

Листинг 3.2 – создание таблиц БД

```
1 -- db.tbl_directions definition
2
3 CREATE TABLE `tbl_directions` (
4   `dirId` int(11) NOT NULL AUTO_INCREMENT,
5   `dirName` varchar(50) DEFAULT NULL,
6   `adyear` int(11) NOT NULL,
7   PRIMARY KEY (`dirId`),
8   UNIQUE KEY `dirName` (`dirName`,`adyear`)
9 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
10
11 -- db.tbl_subjects definition
12
13 CREATE TABLE `tbl_subjects` (
14   `subjectId` int(11) NOT NULL AUTO_INCREMENT,
15   `subjectName` varchar(200) NOT NULL,
16   `accred` varchar(12) NOT NULL,
17   `semester` tinyint(1) DEFAULT NULL,
18   `pract` tinyint(1) DEFAULT NULL,
19   PRIMARY KEY (`subjectId`),
20   UNIQUE KEY `tbl_subjects_un` (`semester`,`accred`,`subjectName`)
21 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
22
23 -- db.tbl_links definition
24
```

```

25 CREATE TABLE `tbl_links` (
26   `linkId` int(11) NOT NULL AUTO_INCREMENT,
27   `dirId` int(11) NOT NULL,
28   `subjectId` int(11) NOT NULL,
29   PRIMARY KEY (`linkId`),
30   UNIQUE KEY `tbl_links_un` (`dirId`,`subjectId`),
31   KEY `subjectData` (`subjectId`),
32   CONSTRAINT `dirData` FOREIGN KEY (`dirId`) REFERENCES `tbl_directions` (`
    dirId`),
33   CONSTRAINT `subjectData` FOREIGN KEY (`subjectId`) REFERENCES `tbl_subjects`
    (`subjectId`)
34 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
35
36 -- db.tbl_reports definition
37
38 CREATE TABLE `tbl_reports` (
39   `repId` int(11) NOT NULL AUTO_INCREMENT,
40   `repYear` int(11) DEFAULT NULL,
41   `repSem` int(11) DEFAULT NULL,
42   `repStatus` tinyint(1) DEFAULT NULL,
43   `repPath` varchar(1000) DEFAULT NULL,
44   PRIMARY KEY (`repId`)
45 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

Таким образом мы получили базу данных, соответствующую проекту основных структур данных. С помощью DBeaver было создано подключение через порт 3306 к IP-адресу локальной машины localhost.

3.3 Реализация модуля curriculum_importer.py

Для решения проблемы непосредственно экспорта данных рассматривались два варианта передачи:

- Конвертирование xls-файла в тестовый формат .csv, и передачей его содержимого с помощью sql-запросов. Такой метод требует, чтобы исходная таблица xls-файла и

таблица базы данных совпадали по своей структуре. Как упоминалось в постановке задачи, файл с учебными планами помимо требуемых данных содержит ячейки, которые не представляют интереса нам. То есть, понадобится создать некоторый алгоритм чистки файла после преобразования в формат .csv, чтобы подготовить его к передаче БД. Также из-за текстового представления данных представляется сложной задачей поиск необходимых ячеек в файле.

- Использование библиотеки `xlrd` питона для чтения `xls`-файлов. При использовании этой библиотеки обращение к ячейкам и листам файла происходит по тем же индексам, что и в Excel, что в значительной степени облегчает поставленную задачу. Стоит отметить, что эта библиотека работает исключительно с форматом `xls`, более современные форматы не обрабатываются ей.

По итогу был выбран вариант с использованием библиотеки `xlrd`, с её помощью был реализован следующий программный код для импорта данных, принцип его работы соответствует шагам, указанным на этапе проектирования:

1. Устанавливаем подключение к БД, если не получилось, сообщаем об ошибке:

Листинг 3.3 – Реализация подключения к БД language

```
1 # пробуем подключится к БД данные( с локальной бд)
2
3 try:
4
5     conn = mariadb.connect(
6
7         user="root",
8
9         password="1024",
10
11         host="localhost",
12
13         port=3306,
14
15         database="db"
16
```

```

17     )
18
19 # если не удалось подключиться, кидаем ошибку
20
21 except mariadb.Error as e:
22
23     print("Error occurred during connection to db: {e}")
24
25     sys.exit(1)
26
27 # установка, чтобы транзакции по соединению выполнялись сразу после их
    вызова автоматически
28
29 conn.autocommit = True

```

2. Открываем xls-файл (пути файлов храним в текстовом файле config.txt), на этом же этапе получаем название направления, его код и дату поступления группы, передаем их таблицу tbl_directions:

Листинг 3.4 – Открытие файла для чтения и получение данных об учебном направлении language

```

1 # открываем и считываем конфигурационный файл
2
3 if (os.path.isfile("config.txt")):
4
5     configFile = open("config.txt", "r")
6
7     planFiles = configFile.readlines()
8
9     configFile.close()
10
11     cur = conn.cursor() # устанавливаем курсор для соединения позволяет(
    выполнять sql команды)
12

```

```

13     # Залезаем в каждый учебный план, перечисленный в конфиге
14
15     for fname in planFiles:
16
17         fname = os.path.split(fname.strip())[1]
18
19         book = xlrd.open_workbook(filename=fname) # открываем таблицу
эксель
20
21         dirName = str(book.sheet_by_name(                # получаем код
направления и год
22
23             УП')).cell_value(2, 0)).split(" - ")[0]
24
25         adyear = int(fname.split(".")[0].split("_")[2])
26
27         # вставляем данные о направлении в таблицу БД, если их не было
ранее; запоминаем Идентификатор направления
28
29         data = (dirName, adyear)
30
31         cur.execute(
32
33             "INSERT INTO tbl_directions (dirName, adyear) VALUES (%s,%d)
ON DUPLICATE KEY UPDATE dirId=dirId;", data)
34
35         cur.execute(
36
37             "SELECT dirId FROM tbl_directions WHERE dirName = %s and
adyear = %d;", data)
38
39         dirId = cur.fetchone()

```

3. Выбираем в нём таблицу для учебного плана одного курса, если она существует,

затем проходимся по ней построчно сверху вниз, обрабатывая случаи, когда в ячейки таблицы Excel на месте названия встречается непустое значение:

Листинг 3.5 – Чтение учебного плана language

```
1 # обход учебного план igo- курса
2
3     for i in range(1, 6):
4
5         shName = str(i) + 'курс_'
6
7         if (shName in book.sheet_names()): # проверяю, что такой
лист есть в экселе
8
9             sh = book.sheet_by_name(shName)
10
11             for j in range(3, sh.nrows):
12
13                 if (sh.cell_value(j, 1) != ""):
14
15                     subjectName = sh.cell_value(j, 1)
16
17                     # нахождение и связка дисциплин по выбору в пары
18
19                     if Дисциплины(" по выбору" in sh.cell_value(j, 1)
):
20
21                         for l in reversed(range(j+1, sh.nrows)):
22
23                             if (sh.cell_value(j, 1) == sh.cell_value(
1, 1)):
24
25                                 subjectName = str(sh.cell_value(
26
27                                     l+1, 1)) + " или " + str(sh.
```

```

cell_value(1+2, 1))
28
29         if практика("'" in str(sh.cell_value(j, 1))):
30
31             pract = True
32
33         else:
34
35             pract = False
36
37         # если дошли до конца
38
39         if (subjectName == Итого"):
40
41             continue
42
43         for k in range(2):           # обход по семестру
44
45             sem = i*2 - 1 + k
46
47             exCol = 8 + k*10  # № колонка, в которой
находятся метки об экзаменах
48
49             # следующие 2 колонки за ней -- для зачета и
дифф зачета соотв.
50             insertInDataBase(subjectName, "exam", sem,
pract, exCol)
51             insertInDataBase(subjectName, "cred", sem,
pract, exCol + 1)
52             insertInDataBase(subjectName, "diff", sem,
pract, exCol + 2)

```

4. Данные о дисциплинах передаются функции insertInDataBase(subjectName, accredType, semester, pract, accredCol) для составления и отправки запроса БД и создания в ней

новой записей о предмете и его связи с направлением. Если такой предмет уже есть в БД, запрос выполняется без создания дубликата и изменения данных:

Листинг 3.6 – Чтение учебного плана language

```
1  # функция для отправки данных предмета в БД
2
3      def insertInDataBase(subjectName, accredType, semester, pract,
4      accredCol):
5
6          # проверяем наличие указанное типа аттестации в выбранном
7          семестре; запоминаем Идентификатор предмета
8
9          if (sh.cell_value(j, accredCol) != ''):
10
11              # в таблицу БД с предметами вносится запись, если её не
12              было ранее
13
14              data = (subjectName, accredType, semester, pract)
15
16              cur.execute(
17
18                  "INSERT INTO tbl_subjects (subjectName, accred,
19                  semester, pract) VALUES (%s,%s,%d,%d) ON DUPLICATE KEY UPDATE
20                  subjectId=subjectId;", data)
21
22              cur.execute(
23
24                  "SELECT subjectId FROM tbl_subjects WHERE subjectName
25                  = %s and accred = %s and semester = %d and pract = %d", data)
26
27              subjectId = cur.fetchone()
28
29              # в таблицу БД со связями направлениепредмет/ вносится
30              запись с полученными Ид, если её не было ранее.
```

```
24
25         data = (dirId[0], subjectId[0])
26
27         cur.execute(
28
29             "INSERT INTO tbl_links (dirId, subjectId) VALUES (%d
            ,%d) ON DUPLICATE KEY UPDATE linkId=linkId", data)
```

5. После выполнения работы программы прекратить подключение:

Листинг 3.7 – Чтение учебного плана language

```
1 conn.close()
```

Заключение

По итогам работы в этом году мной были реализованы структура БД для работы системы и основной модуль работы с базой данных, способный передавать данные из файлов учебных планов в БД.

На данном этапе разработки БД и работа с ней были реализовано локально. В следующем году планируется развёртывание веб-приложения в выделенной на сервере ИМИТ "песочнице".

Литература

1. Сухов, К. HTML5 - путеводитель по технологии / К. Сухов, - Москва : ДМК Пресс, 2013. 352 с. [Электронный ресурс].
2. Документация на Flask (русский перевод) Выпуск 0.10.1 февр. 24, 2019
3. readthedocs.io: [Электронный ресурс]. - [Б.м], 2022. - URL: <https://xlr.readthedocs.io/en/latest/> (дата обращения: 07.03.2023). - Текст: электронный.
4. Pythonworld.ru: [Электронный ресурс]. - [Б.м], 2022. - URL: <https://pythonworld.ru/> (дата обращения: 10.05.2023). - Текст: электронный.
5. Jinja.palletsprojects.com: [Электронный ресурс]. - [Б.м], 2022. - URL: <https://jinja.palletsprojects.com/> (дата обращения: 28.04.2023). - Текст: электронный.
6. Flask-wtf.readthedocs.io: [Электронный ресурс]. - [Б.м], 2022. - URL: <https://flask-wtf.readthedocs.io/> (дата обращения: 19.05.2023). - Текст: электронный.
7. Teemukanstren.com: [Электронный ресурс]. - [Б.м], 2015. - URL: teemukanstren.com/2015/03/08/import-from-excel-to-mysqlmariadb-and-back (дата обращения: 21.05.2023). - Текст: электронный.
8. Flask.palletsprojects.com: [Электронный ресурс]. - [Б.м], 2022. - URL: <https://flask.palletsprojects.com/> (дата обращения: 21.05.2023). - Текст: электронный.