Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

Специальность 09.02.03 Программирование в компьютерных системах

**ОТЧЕТ**

**ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

**(по профилю специальности)**

Профессиональный модульПМ.02 Разработка и администрирование баз данных

*(наименование профессионального модуля)*

Выполнил:

обучающийся учебной группы № 3 ПКС-120

Б. В. Жарков

*(И.О. Фамилия)*

Проверил:

руководитель практики от колледжа:

И. В. Сибирев

*(И.О. Фамилия)*

**Москва**

**20­­­­23**

Содержание

[ВВЕДЕНИЕ 3](#_Toc122618050)

[ОБЩАЯЧАСТЬ 4](#_Toc122618051)

[Деятельность в сфере IT 4](#_Toc122618052)

[Задание 5](#_Toc122618053)

[ЗАКЛЮЧЕНИЕ 6](#_Toc122618054)

[ИСПОЛЬЗОВАННЫЕРЕСУРСЫ 7](#_Toc122618055)

[ПРИЛОЖЕНИЕ 8](#_Toc122618056)

Разработка программных модулей программного обеспечения для компьютерных систем предприятия ООО «Супериор - Запад».

# ВВЕДЕНИЕ

Описание деятельности организации.

Основные цели проекта:

* Обеспечение военнослужащих по призыву мобильной связью с родственниками и друзьями для ускорения адаптации к условиям военной службы. Родителям военнослужащих предоставляется возможность из любой точки России общаться с сыновьями.
* Предоставление военнослужащим возможности бесплатно позвонить в круглосуточную службу психологической консультации и социальной поддержки. Оказание содействия им в получении оперативной правовой поддержки, психологической помощи в стрессовых ситуациях.

# ОБЩАЯЧАСТЬ

## Деятельность в сфере IT

Для выполнения этих целей необходима обработка, хранение и систематизация персональных данных и данных об услугах операторов мобильной связи, чтобы выполнять эти функции работники организации создают скрипты.

В работе организации используются такие языки как Visual Basic, Shell script, SQL, Python.

Для обмена данными с внешними операторами используется Excel, в нем создаются макросы на Visual Basic, которые конвертируют файл в нужный формат, для импорта в SQL, и систематизируют полученные данные.

В Shell script и Python были написаны скрипты, которые используют функции SQL, для проверки и обработки базы данных.

Скрипты запускаются на сервере консольными командами по расписанию.

Для редактирования скриптов используется текстовый редактор.

## Задание

Во время производственной практики мне нужно было разобрать код на Python, который будет вносить данные в MySQL из текстового файла по одной строке, при этом файл нужно читать по частям и заносить данные в две таблицы.

Листинг программы:

[см. Приложение 1]

Вывод программы:

[см. Приложение 2]

Данные MySQL:

[см. Приложение 3]

Данный код запускает функции SQL, с его помощью в программу заносятся данные из текстового файла при этом они вводятся по одной линии, и для каждой строки создается отдельное подключения к базе данных, чтобы не занимать таблицу на долгий промежуток времени.

# ЗАКЛЮЧЕНИЕ

Во время производственной практики я узнал много об использовании программирования в профессиональной деятельности. Ознакомился с некоторыми консольными командами Unix, изучил структуру и синтаксис Shell script, разобрал новые библиотеки в Python, а также рассмотрел совместную работу функций SQL и скриптов на Python и Shell script на платформе Unix.

# ИСПОЛЬЗОВАННЫЕРЕСУРСЫ

<https://www.shellscript.sh/> (дата обращения: 16.06.2023)

# ПРИЛОЖЕНИЕ

Приложение 1

|  |
| --- |
| import mysql.connector  from itertools import islice  long = 100  b = 0  dbconf\_ext = {'user': 'root','password': '57ag237r','host': '127.0.0.1','database': 'cooldatabase','raise\_on\_warnings': True}  conn1 = mysql.connector.connect(\*\*dbconf\_ext)  cursor1 = conn1.cursor()  # Получение списка существующих столбцов в первой таблице  cursor1.execute("SHOW COLUMNS FROM cooltable")  existing\_columns\_table1 = [column[0] for column in cursor1.fetchall()]  # Получение списка существующих столбцов во второй таблице  cursor1.execute("SHOW COLUMNS FROM table\_name")  existing\_columns\_table2 = [column[0] for column in cursor1.fetchall()]  conn1.close()  cursor1.close()  with open('4405\_DET\_202111\_V211209.tab', 'r') as file:  first\_line = next(file).strip().split(',')  print(first\_line)    # Открытие файла с данными  try:  while True:  with open('4405\_DET\_202111\_V211209.tab', 'r') as file:  next(file)  lines = list(islice(file, b, long))  #lines= file.readline().strip().split(',')    b += 100  long += 100  print(b)  print(long)  file.close()  if lines == []:  break    #print(lines)  # Разделение данных на группы  data\_groups = [lines[i:i+1] for i in range(0, len(lines), 1)]  for i, data\_lines in enumerate(data\_groups):  # Подключение к базе данных  try:  conn = mysql.connector.connect(\*\*dbconf\_ext)  cursor = conn.cursor(buffered=True)  # print(conn)  # Определение соответствия колонок в файле и таблицах  our\_columns\_table1 = [] # список колонок для вставки в первую таблицу  our\_columns\_table2 = [] # список колонок для вставки во вторую таблицу  for block in first\_line:  if block in existing\_columns\_table1:  our\_columns\_table1.append(block)  if block in existing\_columns\_table2:  our\_columns\_table2.append(block)  # Чтение и вставка данных  for line in data\_lines:  data = line.strip().split(',')  data = [value if value != '' else None for value in data]  # Подготовка данных для вставки в первую таблицу  values\_table1 = []  for block in our\_columns\_table1:  if block in first\_line:  index = first\_line.index(block)  if index < len(data):  values\_table1.append(data[index])  else:  values\_table1.append(None)  else:  values\_table1.append(None)  # Подготовка данных для вставки во вторую таблицу  values\_table2 = []  for block in our\_columns\_table2:  if block in first\_line:  index = first\_line.index(block)  if index < len(data):  values\_table2.append(data[index])  else:  values\_table2.append(None)  else:  values\_table2.append(None)      if values\_table2 == [] or values\_table1 == []:  a=1  else:  # SQL-запросы  placeholders\_table1 = ', '.join(['%s' for \_ in range(len(our\_columns\_table1))])  columns\_table1 = ', '.join(our\_columns\_table1)  query\_table1 = f"INSERT INTO cooltable ({columns\_table1}) VALUES ({placeholders\_table1})"  placeholders\_table2 = ', '.join(['%s' for \_ in range(len(our\_columns\_table2))])  columns\_table2 = ', '.join(our\_columns\_table2)  query\_table2 = f"INSERT INTO table\_name ({columns\_table2}) VALUES ({placeholders\_table2})"  # Вставка данных в первую таблицу  cursor.execute(query\_table1, tuple(values\_table1))  # Вставка данных во вторую таблицу  cursor.execute(query\_table2, tuple(values\_table2))  # Фиксация изменений и закрытие соединения  conn.commit()  conn.close()  cursor.close()  except mysql.connector.Error as error:  conn.close()  print(f"Вставка данных группы {i + 1} завершена.")  except mysql.connector.Error as error:  print("Ошибка при работе с базой данных:", error)  print("Перенос данных завершен.") |

Приложение 2

|  |
| --- |
| >>> %Run -c $EDITOR\_CONTENT  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8DF6C3A00>  Вставка данных группы 1 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8DF6C3C40>  Вставка данных группы 2 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8E14DE860>  Вставка данных группы 3 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8DF6C3A00>  Вставка данных группы 4 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8E14DF790>  Вставка данных группы 5 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8DF6C3C40>  Вставка данных группы 6 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8E14DE860>  Вставка данных группы 7 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8DF6C3A00>  Вставка данных группы 8 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8E14DF790>  Вставка данных группы 9 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8DF6C3C40>  Вставка данных группы 10 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8E14DE860>  Вставка данных группы 11 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8DF6C3A00>  Вставка данных группы 12 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8E14DF790>  Вставка данных группы 13 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8DF6C3C40>  Вставка данных группы 14 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8E14DE860>  Вставка данных группы 15 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8DF6C3A00>  Вставка данных группы 16 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8E14DF790>  Вставка данных группы 17 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8DF6C3C40>  Вставка данных группы 18 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8E14DE860>  Вставка данных группы 19 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8DF6C3A00>  Вставка данных группы 20 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8E14DF790>  Вставка данных группы 21 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8DF6C3C40>  Вставка данных группы 22 завершена.  <mysql.connector.connection\_cext.CMySQLConnection object at 0x000001E8E14DE860>  Вставка данных группы 23 завершена.  Перенос данных завершен.  >>> |

Приложение 3

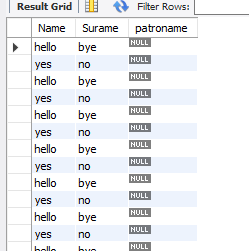


Рисунок 1

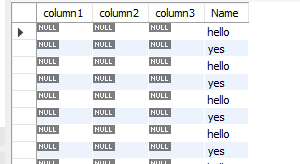


Рисунок 2