Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Операционные среды и системное программирование

ОТЧЕТ

к лабораторной работе №2

на тему

**ИМПОРТ ДАННЫХ (БД)**

Студент М. Ю. Пухов

Преподаватель С. И. Сиротко

Минск 2024

СОДЕРЖАНИЕ

[1 Цель работы 3](#_Toc146752068)

[2 Теоретические сведения 4](#_Toc146752069)

[3 Результат выполнения](#_Toc146752070) 6

[Заключение](#_Toc146752071) 7

[Список использованных источников](#_Toc146752072) 8

[Приложение А (обязательное) Листинг кода](#_Toc146752073) 9

1. **ЦЕЛЬ РАБОТЫ**

Изучение методов и средств обработки текстовой информации, включая регулярные выражения, и использующих их утилит.

1. **ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Входные данные для импорта представлены в формате CSV (Comma-Separated Values). Формат CSV является текстовым файлом, где значения разделены запятыми. Этот формат широко используется для обмена данными между различными приложениями. Важно учитывать особенности формата CSV, такие как обработка кавычек, экранирование символов и т.д.

Важной частью задачи является анализ структуры данных во входных файлах. Обычно информация о структуре таблицы БД содержится в первых строках входного файла. Например, первая строка может содержать имена таблиц, а вторая – имена соответствующих полей в этих таблицах. Это позволяет определить, куда должны быть импортированы данные.

Для импорта данных в БД предлагается использовать стандартные SQL-запросы, что обеспечивает переносимость между различными системами управления базами данных (СУБД). Оператор INSERT используется для добавления данных в таблицу. Важно соблюдать синтаксические правила стандартного SQL для обеспечения совместимости.

При необходимости, можно учесть специфические диалекты SQL для конкретной СУБД (например, MySQL, PostgreSQL, SQLite). Однако, рекомендуется ограничиваться стандартным SQL для обеспечения переносимости кода.

В процессе импорта данных важно предусмотреть механизм обработки ошибок. Это включает в себя проверку правильности данных, наличие необходимых полей и обработку возможных конфликтов при вставке данных в таблицу.

Обеспечение безопасности при импорте данных также является критическим аспектом. Использование параметризованных запросов, проверка наличия необходимых прав доступа и другие меры могут снизить риски связанные с безопасностью данных в процессе импорта.

1. **РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ**

Изначально в файле *sql\_script.sql* хранится 5 записей (рисунок 1).

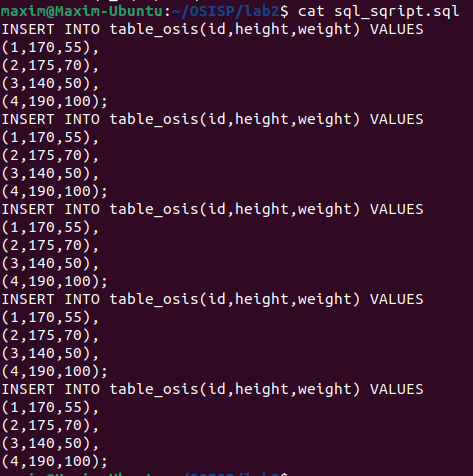


Рисунок 1 – Начальные данные

После выполнения скрипа туда записывается дополнительное количество данных (рисунок 2)

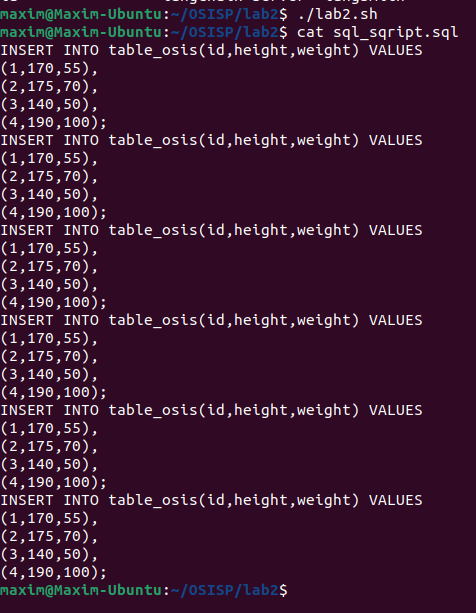


Рисунок 2 – Дополнительные значения

# ЗАКЛЮЧЕНИЕ

В результате рассмотрения задачи по импорту данных из CSV-файла в базу данных с использованием SQL-запросов INSERT, можно отметить важность эффективного управления данными.

Процесс импорта данных начинается с анализа формата CSV, который предоставляет удобный способ представления информации в текстовой форме. Анализ структуры данных в файле, особенно первых строк, содержащих информацию о таблицах и полях, является критическим шагом для правильного направления данных в базу.

Использование стандартных SQL-запросов INSERT обеспечивает переносимость кода между различными системами управления базами данных. Однако, при необходимости, можно учитывать специфические диалекты SQL для конкретных СУБД.

Важным аспектом в решении задачи является обработка ошибок, включая проверку правильности данных и предотвращение возможных конфликтов при вставке данных. Также, обеспечение безопасности данных в процессе импорта через использование параметризованных запросов и контроль доступа к базе данных является неотъемлемой частью решения.

В целом, разработка эффективного механизма импорта данных требует внимания к деталям, соблюдение стандартов SQL и учет особенностей структуры входных файлов для обеспечения успешной и безопасной передачи информации в базу данных.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. CSV File Format [Электронный ресурс]. – Режим доступа: https://www.csvreader.com/.
2. SQL Tutorial [Электронный ресурс]. – Режим доступа: https://www.w3schools.com/.

# ПРИЛОЖЕНИЕ А (обязательное) Листинг кода

Листинг 1 – Файл *lab2.sh*

#!/bin/bash

table\_name=""

counter=0

var1=""

while IFS= read -r line

do

# Первая строка содержит имена таблицы

if [[ "$counter" == 0 ]] ; then

table\_name=${line//[[:blank:]]/} # Удаление пробелов и табуляций из имени таблицы

elif [[ "$counter" == 1 ]] ; then

var1="INSERT INTO $table\_name(${line//[[:blank:]]/}) VALUES"

else

echo "$var1" >> sql\_script.sql

var1='('${line//[[:blank:]]/}'),'

fi

((counter++))

done < table.csv

# Запись последней строки в файл

echo ${var1::-1}';' >> sql\_script.sql # Удаление последней запятой и добавление точки с запятой

Листинг 2 – Файл *sql\_sqript.sql*

INSERT INTO table\_osis(id,height,weight) VALUES

(1,170,55),

(2,175,70),

(3,140,50),

(4,190,100);

INSERT INTO table\_osis(id,height,weight) VALUES

(1,170,55),

(2,175,70),

(3,140,50),

(4,190,100);

Листинг 3 – Файл *table.csv*

*table\_osis*

*id, height, weight*

*1, 170, 55*

*2, 175, 70*

*3, 140, 50*

*4, 190, 100*