UDSQL Database User Manual

Introduction

UDSQL is a simple database program that allows you to create, read, update, and delete records in tables. It uses a custom file format (.udsql) to store data and a metadata file to keep track of table structures. This manual will guide you through the basic operations of UDSQL.

Getting Started

1. **Installation**
    a. Make sure you have Python 3 installed on your system.
    b. Repository: https://github.com/Inaryuta/Database_Foundations_Workshop-3.git
    c. If you want to start with a completely new database, you will need only 4 repository files (Operations folder, main.py, create_table.py and metadata_manager.py).
    d. Now with these files we are going to open a terminal in the folder path where we have them and the first step will be to execute the following command (python3 create_table.py) depending on your case "python3" can be replaced with "py" or simply "python"

```
alejo@alejo-Lenovo-IdeaPad-S145-14API:~/Descargas/Database_Foundations_Workshop-3-main$ python3 create_table.py
Tabla 'Estudiantes' creada con exito
Tabla 'Profesores' creada con exito
Tabla 'Administrativos' creada con exito
Tabla 'Espacios_Academicos' creada con exito
Tabla 'Biblioteca' creada con exito
Tabla 'Carrera' creada con exito
```

    e. Repeat the process but in this case the comand will be (python3 metadata_manager.py)

```
alejo@alejo-Lenovo-IdeaPad-S145-14API:~/Descargas/Database_Foundations_Workshop-3-main$ python3 metadata_manager.py
Metadata actualizada: Tabla 'Estudiantes' añadida
Metadata actualizada: Tabla 'Profesores' añadida
Metadata actualizada: Tabla 'Administrativos' añadida
Metadata actualizada: Tabla 'Espacios_Academicos' añadida
Metadata actualizada: Tabla 'Biblioteca' añadida
alejo@alejo-Lenovo-IdeaPad-S145-14API:~/Descargas/Database_Foundations_Workshop-3-main$
```

2. **Running the Program**
    a. Open a terminal or command prompt in the directory where you saved the files again.
    b. Run the program by typing (python3 main.py) and pressing Enter.

Main Menu

Once the program starts, you will see the main menu with the following options:

1. Insert record
2. Update record
3. Delete record
4. Select records
5. Exit

Operations

1. **Insert Record**
    a. Select option 1 from the main menu.
    b. Enter the name of the table you want to insert into.
    c. Enter the values for each field in the table, one by one.
    d. The program will display a success message with the new record's code.
2. **Update Record**
    a. Select option 2 from the main menu.
    b. Enter the name of the table you want to update.
    c. Enter the code of the record you want to update.
    d. Enter the new values for each field in the record, one by one.
    e. The program will display a success message.
3. **Delete Record**
    a. Select option 3 from the main menu.
    b. Enter the name of the table you want to delete from.
    c. Enter the code of the record you want to delete.
    d. The program will display a success message.
4. **Select Records**
    a. Select option 4 from the main menu.
    b. Enter the name of the table you want to query.
    c. Enter a WHERE clause to filter the records (optional). If you press Enter without typing a clause, all records will be selected.
    d. The program will display the selected records in a table format.

WHERE Clause Syntax

The WHERE clause is used to filter records in a SELECT query. It consists of one or more conditions combined with logical operators.

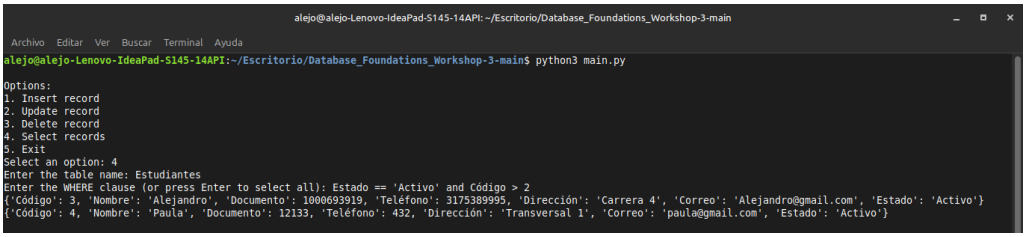
- **Format:** The correct way to introduce the WHERE clauses is the following, the name of the attribute without quotes + operators + 'data to be entered' for example **State == 'Active'**
- **Conditions**
  - A condition compares a field value with a constant value using comparison operators:
    - == (equal to)
    - != (not equal to)
    - > (greater than)
    - < (less than)
    - >= (greater than or equal to)
    - <= (less than or equal to)
  - For example: Status == 'Active'
- **Logical Operators**
  - Conditions can be combined using logical operators:
    - and (both conditions must be true)
    - or (at least one condition must be true)
    - not (negates a condition)
  - For example: Status == 'Active' and Code > 10
- **Parentheses**
  - Use parentheses to group conditions and control the order of evaluation.
  - For example: (Status == 'Active' or Status == 'Pending') and Code < 100

Examples

- Select all active students: Status == 'Active'
- Select students with code greater than 100: Code > 100
- Select students with name "John" and status "Active": Name == 'John' and Status == 'Active'
- Select students with status "Active" or "Pending" and code less than 100: (Status == 'Active' or Status == 'Pending') and Code < 100

Exiting the Program

- Select option 5 from the main menu to exit the program.
- If you can't find a table or record, make sure it exists and that you are typing the name correctly.


alejo@alejo-Lenovo-IdeaPad-S145-14API: ~/Escritorio/Database_Foundations_Workshop-3-main
Archivo Editar Ver Buscar Terminal Ayuda
alejo@alejo-Lenovo-IdeaPad-S145-14API:~/Escritorio/Database_Foundations_Workshop-3-main$ python3 main.py
Options:
1. Insert record
2. Update record
3. Delete record
4. Select records
5. Exit
Select an option: 4
Enter the table name: Estudiantes
Enter the WHERE clause (or press Enter to select all): Estado == 'Activo' and Código > 2
{'Código': 3, 'Nombre': 'Alejandro', 'Documento': 1000693919, 'Teléfono': 3175389995, 'Dirección': 'Carrera 4', 'Correo': 'Alejandro@gmail.com', 'Estado': 'Activo'}
{'Código': 4, 'Nombre': 'Paula', 'Documento': 12133, 'Teléfono': 432, 'Dirección': 'Transversal 1', 'Correo': 'paula@gmail.com', 'Estado': 'Activo'}