SQL Like Query Parser and Executor in C

This project is about building a simplified SQL – like query parser and executor using the C programming language. The program can read a database table stored as a text file, where the first row contains field names and the following rows contain data records. Users can input queries in a restricted SQL Format (e.g., *SELECT name, grade FROM students WHERE age = 20;*), and the program will parse the query validate the fields and table, process the optional *WHERE* conditions, and then display the requested results. The goal of this project is to demonstrate how string parsing, file handling, and conditional filtering can be combined to mimic the functionality of SQL in a basic way.

1st Test Case

Input Query: SELECT name, grade FROM students WHERE age = 20;

Output Query: Query Parsed Successfully!

Results: name grade

Alice A

```
alif@ALIF:~/coding/CSE/database$ cd "/home/alif/coding/CSE/database/"
 && gcc new.c -o new && "/home/alif/coding/CSE/database/"new
Enter Query:
SELECT name,grade FROM students WHERE age=20;
Query Parsed Successfully!
Results:
name    grade
Alice   A
alif@ALIF:~/coding/CSE/database$
```

2nd Test Case:

Input Query: SELECT name, grade FROM students WHERE grade = A;

Output Query: Query Parsed Successfully!

Result: id   name   age   grade

1   Alice   20   A

3   Charlie   19   A

```
alif@ALIF:~/coding/CSE$ cd "/home/alif/coding/CSE/database/" && gcc new.c -o new && "/home/alif/coding/
CSE/database/"new
Enter Query:
SELECT id, name, age, grade FROM students WHERE grade=A;
Query Parsed Successfully!
Results:
id      name    age     grade
1       Alice   20      A
3       Charlie 19      A
alif@ALIF:~/coding/CSE/database$
```

3rd Test Case:

Input Query: SELECT age FROM students WHERE name = Bob;

Output Query: Query Parsed Successfully!

          Results: age

                   21

```
alif@ALIF:~/coding/CSE/database$ cd "/home/alif/coding/CSE/database/"
  && gcc new.c -o new && "/home/alif/coding/CSE/database/"new
Enter Query:
SELECT age FROM students WHERE name=Bob;
Query Parsed Successfully!
Results:
age
21
alif@ALIF:~/coding/CSE/database$
```

4th Test Case:

Input Query: SELECT address FROM students WHERE grade = A;

Output Query: Error: Field 'address' not found in table 'students'.

```
alif@ALIF:~/coding/CSE/database$ cd "/home/alif/coding/CSE/database/"
  && gcc new.c -o new && "/home/alif/coding/CSE/database/"new
Enter Query:
SELECT address FROM students WHERE grade=A;
Error: Field 'address' not found in table 'students'.
alif@ALIF:~/coding/CSE/database$
```

5th Test Case:

Input Query: SELECT name FROM students WHERE height=180;

Output Query: Error: Field 'height' not found in table 'students'.

```
alif@ALIF:~/coding/CSE/database$ cd "/home/alif/coding/CSE/database/"
  && gcc new.c -o new && "/home/alif/coding/CSE/database/"new
Enter Query:
SELECT name FROM students WHERE height=180;
Error: Field 'height' not found in table 'students'.
alif@ALIF:~/coding/CSE/database$
```