ASSIGNMENT 4

In this assignment, you're required to extend the work done by you for the Relational Algebra (RA) subset **{SELECT, PROJECT, CARTESIAN PRODUCT, EQUI-JOIN}** mentioned in Assignment 3.

You're expected to use (and modify, if required) the flex code written by you for Assignment 3, to perform lexical analysis whilst using flex to generate the scanner and bison to generate the parser. The bison parser should generate the target code in C language for your input RA program, considering all database tables to be simply represented by text files in comma separated .csv format. The generated C code should be compilable using gcc. Outputs of all queries should be directed to **stdout** as shown in the examples hereunder.

The Syntax for the operations **{SELECT, PROJECT, CARTESIAN PRODUCT, EQUI-JOIN}** are same as mentioned in assignment-3. However, the expected output for each query is not only the "valid syntax" or "invalid syntax" but also the evaluated query output. Few examples on SELECT operation are given below.

Contents of Employee.csv file are as follows:

```
Id,Name,Salary
1,John,400
2,Jill,30
3,Jack,400
4,John,180
```

Eg 1) SELECT < salary<200 > (Employee)

Output:

```
Id,Name,Salary
2,Jill,30
4,John,180
```

Eg 2) SELECT < name='John' AND salary>2000 > (Employee)

Output:

```
Id,Name,Salary
No rows selected
```

Eg 3) SELECT (Employee) < salary<200 >

Output:

```
Invalid Syntax
```

Further, your program should be able to deal with various other errors (e.g. Table not found, column not found, etc.)