Database Systems and Implementations

Assignment

TODO:-
• Finding the **Closure** of an attribute given a set of Functional Dependencies

• Finding **Candidate key**s given a Relation and a set of Functional Dependencies

• To check **Equivalence** between 2 sets of Functional Dependencies given a Relation

• Find the highest **Normal Form** given a Relation and a set of Functional Dependencies

• **Minimal Cover** given a set of Functional Dependencies

Program is divided into 3 files.

Assignment File
assignment.py is used for user interface. It reads the relation and the set of functional dependencies and displays the result to the user according to the instructions.

Helper File
helpers.py is a file that contains functions namely,
Every
- Returns True if every element of the list returns True.
Some
- Returns True even if a single element of the list returns True.

Func File
func.py file defines 2 classes and other functions.
The 2 classes are:
FunctionalDependencySet
This Class is used to define all the functional dependencies called *__items__* where each fd is a tuple of the form *(lhs, rhs).*
The Class defines functions to add, remove or replace functional dependencies. It also includes a function to calculate the closure of any attribute for given set of functional dependencies.
Relation
Any object of this class consists of 2 attributes. First is the attributes in the database. Second is an instance of FunctionalDependecySet describing the fds of this relation.
The methods included are:
closureSet
Calculates the closure of an attribute given the fds.
validKey
Returns True if given attribute is a super key of the given relation.
candidateKeys
Returns the set of all the candidate keys for the given relation.

cover(fd1, fd2) :- checks if fd1 is covered by fd2.
equivalence(fd1, fd2) :- checks if fd1 and fd2 are equivalent functional dependencies.
isPartialDependency() :- checks if fdItem is a partial functional dependency.

hasPartialDependency() :- checks if given relation has a partial dependency.
isFirstNF(), isSecondNF(), isThirdNF(), isBCNF() checks normal forms according to the definations.

*Note fds are assumed to be in 1NF.

minimalCover() :- Finds the minimal cover of the fd. Redundant fds are removed by finding closures with and without a particular fd.