Database Programs Using Python

Samyak Ahuja

University of Delhi

1. Introduction

List of Programs are as follows :

- **Closure** of an attribute given a set of Functional Dependencies
- **Candidate key**s given a Relation and a set of Functional Dependencies
- **Equivalence** between 2 sets of Functional Dependencies given a Relation
- **Minimal Cover** given a set of Functional Dependencies
- Find the highest **Normal Form** given a Relation and a set of Functional Dependencies

Note Definition of Minimal Cover is often contrieved and confused with Canonical Cover Let's clear that up once and for all. A canonical cover is "allowed" to have more than one attribute on the right hand side. A minimal cover cannot. As an example, the canonical cover may be "A -> BC" where the minimal cover would be "A -> B, A -> C". That is the only difference.

2. Program Structure

Programs are divided into 3 files each having a purpose of its own.

Helper File

helpers.py is a file that contians helper functions namely,

Every

- Returns True if every member of the list when called with function callback return True.

Some

- Returns True if any member of the list when called with function callback returns True.

Driver File

driver.py is used for interacting with the user. It reads the relation and the set of functional dependencies from the user and the displays the menu and the result to the user.

DBMS File

dbms.py file defines 2 different Classes and all the functions listed above.

The 2 classes are:

FunctionalDependencySet

This Class is used to define all the functional dependencies(fd) in a set of fds called *__items__* where each fd is a tuple of the form *(lhs_of_fd, rhs_of_fd).*

The Class defines functions to add, remove, 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/features 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.

The functions provided are:

cover(fd1, fd2)
: Returns True if fd1 is covered by fd2.

equivalence(fd1, fd2)
: Returns True if fd1 and fd2 are equivalent functional dependencies.

isPartialDependency(fdItem, candidates, nonPrimes)
: Returns True if fdItem is a partial functional dependency.

hasPartialDependency(relation)
: Returns True if given relation has a partial dependency

isFirstNF(relation)
: Always returns True, It has been assumed that a relation is always in 1NF.

isSecondNF(relation)
: Returns True if given relation is in 2NF. It does so by checking 2 things.

- Given relation in in 1NF (always true)
- Given relation does not have any partial dependency

isThirdNF(relation)
: Return True if given relation is in 3NF. It does so by checking 2 things.

- Given relation is in 2NF.
- For every functional dependency, Any of following is True.
  - RHS is subset of LHS, i.e. it is a trivial functional dependency.
  - LHS is a valid super-key
  - RHS is a subset of the prime attributes.

isBCNF(relation)
: Return True if given relation is in BCNF. It does so by checking 2 things.

- Given relation is in 3NF.

- For every functional dependency LHS is a valid super-key.

minimalCover(attributes, fdString)

Return an instance of FunctionalDependencySet that is the minimal Cover given the functional dependencies. It forms the minimal cover by doing 3 steps:

- Reduce the RHS by using decomposition rule so that RHS only contains 1 attribute.
- Remove any Functional Dependency if cover of LHS when removed and when isn't are equal.
- Check for Redundancy in the LHS of the functional dependency.

3. Ouput examples