Build and Run

CSC 173 - Project 4: The Relational Data Model

THE COMPLETE PROJECT IS MADE IN THE REPL FORMAT AND GUIDES THE USER THROUGH ALL THE THREE PARTS.
To RUN The file on the command prompt kindly enter these commands:

```
gcc -o -xyz -std=c99 -std=c11 -Wall -Werror main.c LinkedList.c
```

followed by another command:

```
xyz
```

Table Implementation

The table is implemented using a hashmap composed of an array of linkedlists of tuples.

- Hash function sums the ascii values of a tuple’s characters and finds the remainder when divided by the length of the internal array.
- Length of internal array set by global constant “HASH MOD”.
- Tuples are added to the end of the linkedlist at the index the tuple hashes to.
- Instantiating a new hashmap requires a list of attribute names, a list of key names, and the lengths of those lists.
- The table is fully generic with void arguments for its functions.

Insertion

Insertion takes a generic tuple and a hashmap with the same schema to insert into and returns nothing. • Hashes key attributes to find linkedlist of hashmap to insert into.

• Looks for any matching tuples to prevent duplicates. • Adds tuple to end of linkedlist.

Lookup

Looking up tuples takes a generic tuple and a hashmap with the same schema to search and returns a linkedlist of matching tuples.

• Can take “*” as an element of a tuple to search for multiple matches. • Checks if a tuple is hashable to find matching linkedlist in hashmap. • Iterates through matching linkedlist to return all matching tuples.
• Iterates through entire hashmap to return matching tuples otherwise.

Deletion

Deleting tuples takes a generic tuple and a hashmap with the same schema to delete from and returns nothing. • Can take “*” as an element of a tuple to delete multiple matches.

• Checks if a tuple is hashable to find matching linkedlist in hashmap. • Iterates through matching linkedlist to delete all matching tuples.
• Iterates through entire hashmap to delete matching tuples otherwise.

Selection

Selecting rows from a relation takes a hashmap and an attribute index and returns a hashmap with the selected rows. • Goes through entire hashmap to check all tuples of relation.

• Takes user input to compare to target index attribute values. • Checks each tuple for target attribute matching user input.
• Copies matching tuples over to output hashmap.

Projection

Projecting columns from a relation takes a hashmap and a number of columns to project. • Takes user input for target attributes.

• Creates output hashmap with only the target attributes.
• Goes through entire hashmap to check all tuples of relation.
• Copies target attribute values from each tuple into new hashmap.

Join

Joining two relations takes the two hashmaps to be joined and the index of the joining attribute in each hashmap. • Makes a union of the hashmap schemas to make output hashmap schemas.

• Determines keys of output hashmap by merging the keys of the original hashmaps.
• Filters out tuples that are unique to one of the input hashmaps.
• Matches tuples from first hashmap to a linkedlist of tuples fromt the second hashmap. • Merges matching tuples to the output hashmap’s schema.
• Inserts merged tuples into the output hashmap.