PROJECT 4 - THE RELATIONAL DATA MODEL

GROUP MEMBERS:

1. Thabani Dube

2. Nicholas Salem

3. Syed M Qasim Sudais

HOW TO BUILD:

Compile from terminal with command gcc Project_4.c

Run with a or .\a

TESTING THE PROJECT:

Once the project runs, everything will be tested sequentially from part 1 to part 3.

PART 1:

Informative output is produced.

First all data as in the textbook is inserted into the database and the database is printed out.

Then from there, the required Example 8.2 operations are performed.

From there, the read write is performed. Current database is written to file and database is cleared.

Then the data is read from the file back into the current database

Just before part 2, database is filled again to restore to the original before changes by

example 8.2 operations

PART 2:

Query is prompted. Enter test imput by referring to database in book or printed at beginning of part 1

Example for query 1: Name: P. Patty and Course: EE200

Example for query 2: Name: P. Patty, Day: Tu, Hour: 10AM

Relevant answers to queries are outputted.

PART 3:

All operations are performed, and informative output is produced.

IMPLEMENTATION DETAILS

Relations are stored in a hash table. This also applies to those created in part 3 operations.

Each relation for example, CSG will have its own struct for a tuple of that relation. These tuples are stored in a hash table of tuples in a linked list. The database is tables of these tuples.

Tuples are always found through indexing, however if there is no key in the function arguments (i.e * argument on key), then the lookup will take longer by going through the hash table. Insert works by looking up first and if not found, insertion is performed, otherwise ignored. Delete checks if the tuple exists and if not adds it, otherwise, ignores. In these cases, indexing is still used if possible.

The keys are as follows:

CSG – Course-StudentId

SNAP – StudentId

CP – Course-Prerequisite

CDH – Course-Day

CR – Course-Room

In new Tuples created in part 3

S – StudentId

CRDH – Course Day

DH – Day-Hour

Notes:

-When running, you will notice random printing of tuples after output processing… The result however, is what is clearly produced after that. This is due to use of the lookup function which I made to print out the tuples found. The way the look up function works, is it looks up and returns true or false based on whether anything was found. In addition to that, it will store these found tuples an array. This way they can be accessed and used if needed, otherwise it can be just checked if they exist. However, it also prints them as they are being found. This could be changed around easily but the functionality is enough either way.

-As the project runs, a lot of output will be presented. If followed from the beginning, the output is sequential according to the project handout. You may need to scroll around. However, all parts are clearly divided. The code offers a prompt for the part 2 queries.