

MSDS 7330

File Organization and Database Management

Homework Databases

Name: Zach Brown

This is a homework assignment for MSDS 7330, File Organization and Database Management. Print out this document, and unless directed otherwise, hand write your answer to a question in the white space under the question. Scan your completed solutions and submit them in a single pdf file. Be sure to put your last name at the beginning of the filename. For example, for Daniel Engels, the submitted file should be named EngelsMSDS7330HW2.pdf.

Collaboration is expected and encouraged; however, each student must hand in their own homework assignment. To the greatest extent possible, answers should not be copied but, instead, should be written in your own words. Copying answers from anywhere is plagiarism, this includes copying text directly from the textbook. Do not copy answers. Always use your own words. Directly under each question list all persons with whom you collaborated and list all resources used in arriving at your answer. Resources include but are not limited to the textbook used for this course, papers read on the topic, and Google search results. Don't forget to place your name on every page.

Introduction to Databases

Question 1 : A database contains a collection of data relevant to an enterprise. A database-management system (DBMS) provides a set of programs to manage the data and to access the data stored in the enterprise's databases. The database system and the DBMS were designed to overcome the limitations of the typical file processing systems used prior to the invention of the modern database. Identify three broad problems experienced with file processing systems, such as data integrity problems, and discuss how a database system might be able to overcome the problems.

Collaborators:

Resources: Database System Concepts - Sixth Edition by Abraham Silberschatz, Henry F. Korth and S. Sudarshan

One problem experienced with file processing systems is data redundancy and inconsistency. Files and application programs created by different people are bound to be structured differently from one another and contain duplicate data. Database systems overcome this obstacle by providing one consistent data repository that can be referenced by everybody.

Another problem with file processing systems is data accessibility. With file processing systems, ad-hoc requests may require an entirely new application to be built. But with database systems, you simply need to alter the query you run to change the data set that is returned.

As mentioned in the question, integrity problems are also common among file processing systems. If data is stored in more than one location, it is easy for the values to get out of sync. Databases solve this problem by enforcing referential integrity constraints.

Question 2 : Data may be represented at three different layers of abstraction: *physical*, *logical* and *view*. At which level should a database administrator design a database and why?

Collaborators:

Resources: Database System Concepts : Sixth Edition

A database should be designed at the logical layer. This is the layer that describes what data are stored, as well as the relationships between those data.

Question 3 : Provide an answer to each of the following questions.

Collaborators:

Resources: Database System Concepts : Sixth Edition

Name three (3) approaches to storing data.

Three approaches to storing data are the Relational Model, The Entity-Relationship Model and the Semistructured Data Model.

Name three (3) applications in which data is used extensively.

Three applications which use data extensively are web site backends, banking applications and enterprise resource planning applications.

Question 4 : Provide a concise definition for each of the following:

Collaborators:

Resources: Database System Concepts: Sixth Edition

Database Schema

The logical design of a database

Data Manipulation Language (DML)

Language used for retrieval, insertion, deletion and modification of data.

Data Definition Language (DDL)

Language used to define the schema and other attributes of a database.

Question 5 : Database applications are typically partitioned into either two or three tiers. For each of the following statements regarding database applications identify whether the statement is either *True* or *False* and state why you believe the statement to be thus.

Collaborators:

Resources: Online Courseware video 2.5.1 - RDBMS in a Typical System Architecture

In a three-tier architecture the client machine acts as a front end interface to the user and does not contain any direct database calls.

True. The client machine interacts with middleware via an API and does not call the database directly.

In a two-tier architecture the user accesses the application located on the database server.

The application accesses the database system directly.

False. The applications accessed by users in a two-tier architecture typically communicate with the database over a network.

In a three-tier architecture the application functionality is split between the client machine and the database server machine.

False. The application functionality is split between the client machine and the middleware.

The Relational Model

Question 6 : The relational model utilizes its own unique vocabulary. It is important to learn this vocabulary in order to quickly and easily understand any discussion on relational databases. To this end, define each of the following.

Collaborators:

Resources: Database System Concepts: Sixth Edition

Relation

A uniquely named set of attributes, also known as a table

Relation Instance

A snapshot of the data contained within a relation

Tuple

A row within a table/relation.

Attribute

A column within a table/relation

Domain

The set of all values that are possible for a specific attribute

Atomic Domain

A domain where none of the elements can be broken down into more granular values.

Question 7 : Keys are used to distinguish tuples within a relation. In the fundamental relational model, no two tuples are allowed to have the same attribute values for all attributes. This ensures that a subset of the attributes may be used to uniquely identify each tuple. These subsets are referred to as *keys*. Define each of the following types of keys.

Collaborators:

Resources: Database System Concepts: Sixth Edition

Superkey

One or more attributes (possibly containing extraneous attributes) that can collectively uniquely identify a tuple

Candidate Key

A subset of superkeys that do not contain any extraneous attributes.

Primary Key

The candidate key that the DBA chooses to uniquely identify tuples

Foreign Key

An attribute that references another table's primary key

Question 8 : The relational model allows for a set of operations to be performed on relations in accordance to relational algebra. A relational operation is the Natural Join. Describe what the Natural Join does. Identify its inputs and its output. Give a simple example. What is the benefit of using a Natural Join?

Collaborators:

Resources: Database System Concepts : Sixth Edition

The Natural Join combines two relations ~~which~~ which have at least one attribute in common and returns the combinations where the identically named attributes have the same value for both relations.

For Example, a natural join of the following two tables:

Table 1

A	B
1	2
1	3
2	3

Table 2

B	C
1	3
1	4
2	3
3	4

Would return the following table:

A	B	C
1	2	3
1	3	4
2	3	4

Consider the following relational database for the next two questions.

employee (*personName*, *street*, *city*)

works (*personName*, *companyName*, *salary*)

company (*companyName*, *city*)

Question 9 : Consider the relational database above. What are the appropriate primary keys?

Collaborators:

Resources:

The primary key for *employee* would be *personName*.

The primary key for *works* would be (*personName*, *companyName*) and *personName* and *companyName* would also each be foreign keys.

The primary key for *company* would be *companyName*.

Question 10 : Consider the relational database above. Give an expression in the relational algebra to express each of the following queries:

- 1) Find the names of all employees who live in the city of Miami.
- 2) Find the names of all employees whose salary is greater than \$100,000.
- 3) Find the names of all employees who live in Miami and whose salary is greater than \$100,000.

Collaborators:

Resources:

$$1.) \pi_{\text{personName}}(\sigma_{\text{city}=\text{Miami}}(\text{employee}))$$

$$2.) \pi_{\text{personName}}(\sigma_{\text{salary} > 100000}(\text{works}))$$

$$3.) \pi_{\text{personName}}((\sigma_{\text{city}=\text{Miami}}(\text{employee})) \bowtie (\sigma_{\text{salary} > 100000}(\text{works})))$$

Consider the following Python code for the following questions.

```
1 print "Think of a number between 1 and 100."
2 print "Now allow me to guess your number."
3 minNumber = 1
4 maxNumber = 100
5 correctAnswer = False
6 while not correctAnswer:
7     midNumber = (maxNumber + minNumber + 1)/2
8     answer = raw_input('Is your chosen number ' + str(midNumber) + '?')
9     if answer[0] == 'y':
10        correctAnswer = True
11    elif answer.startswith('lower'):
12        maxNumber = midNumber - 1
13    elif answer.startswith('higher'):
14        minNumber = midNumber + 1
15    else:
16        print "I don't understand your answer."
17
18 print 'Yeah! I got it!'
```

Question 11 : The while loop exits when the variable correctAnswer is True. What will cause correctAnswer to be True?

Collaborators: correctAnswer is set to True if the user enters 'y' in response to the prompt.

Resources:

Question 12 : How many times will the program print out 'Yeah! I got it!?

Collaborators: The program will print "Yeah! I got it!" once (assuming the condition to exit the while loop is satisfied at some point).

Resources:

Question 13 : What is purpose of the variable answer?

Collaborators: The answer variable stores the user's response when asked if the correct number ~~was~~ was guessed.

Resources:

Question 14 : The program prints out its guess at line 8. What user response will the program understand in response to its guess?

Collaborators: The only answers that the program will understand are "y", "lower" and "higher".

Resources: Any other answer will result in it printing "I don't understand your answer."

Question 15 : If the program gets the response 'higher', what does that tell the program about its guess?

Collaborators: The response "higher" tells the program that the user's chosen number is higher than its current guess.

Resources:

Question 16 : What are the variables minNumber, maxNumber and midNumber used for?

Collaborators: minNumber is the smallest number that could be correct given the user's previous answers.

Resources: maxNumber is the largest number that could be correct given the user's previous answers.

midNumber is the halfway point between minNumber and maxNumber, which is also used as the program's next guess.

Question 17 : A useful data type built into Python is the dictionary. Dictionaries are indexed by *keys*, which can be any immutable type. Keys are commonly of type string or a number. A dictionary is easily considered to be an unordered set of *key:value* pairs. Keys must be unique within a dictionary. The main operations on a dictionary are storing a value with some key and extracting the value given the key.

Create a Python dictionary that catalogs the courses you are currently taking for the SMU MSDS program. The keys should be the course number, and the values should be the course titles.

Write a function `add_course` that takes 2 arguments (a course number a title) and adds a course to your dictionary. Use this function to add the courses you're taking next term to the dictionary.

Write a function `print_courses` that takes one argument (a search string that is a portion of a course number) and prints out all the courses in your dictionary that match the search string in any part of the course number. Be sure to have an appropriate response for a search string that has no matches.

Attach to the homework pdf a pdf document of your source code and a pdf of the screen capture of an execution of your database and functions. Be sure to include a screen capture that captures all possible functionality for your program.

Collaborators:

Resources: [HTTP://docs.python.org](http://docs.python.org)

Name: Zach Brown

Zach Brown

MSDS 7330 403

HW2_Q17.py

29 Sep 2015

Instantiate dictionary containing courses

```
courses = {'MSDS6371': 'Experimental Statistics I',  
           'MSDS7330': 'File Organization and Database Management'}
```

Function to add a course

```
def add_course(num, title):  
    courses[num] = title
```

Function to print the courses

```
def print_courses(str):  
    # Step through each key:value pair in the dictionary and print  
    # the course title if the string is contained in the course number  
    match = False  
    for i in courses:  
        if (str in i):  
            match = True  
            print courses[i]
```

If no keys match the string, print an error message

```
if (match == False):  
    print 'No matching courses!'
```

Function to prompt user for Course Number

```
def get_course_num():  
    return raw_input('Enter Course Number: ')
```

Name: Zach Brown

Function to prompt user for Course Name

```
def get_course_name():
```

```
    return raw_input('Enter Course Name: ')
```

Function to prompt user for Course Number search string

```
def get_search_string():
```

```
    return raw_input('Enter full or partial Course Number to search for: ')
```

Function to ask user if they want to add a course or print courses

```
def get_option():
```

```
    return int(raw_input('Enter 1 to add a course, 2 to print courses or 3 to exit: '))
```

Main function

```
def main():
```

```
    keep_running = True
```

```
    while(keep_running == True):
```

```
        # Get option from user and make sure it's 1 or 2
```

```
        option = get_option()
```

```
        # Add course if user enters 1
```

```
        if (option == 1):
```

```
            course_num = get_course_num()
```

```
            course_name = get_course_name()
```

```
            add_course(course_num, course_name)
```

```
        # Print course if user enters 2
```

```
        elif (option == 2):
```

```
            search_string = get_search_string()
```

```
            print_courses(search_string)
```

```
        # Exit the program if user enters 3
```

```
        elif (option == 3):
```

```
            keep_running = False
```

```
            return
```

Name: Zach Brown

```
# Print an error message if user enters an invalid option
```

```
else:
```

```
    print 'Please enter a valid option'
```

```
main()
```

Name: Zach Brown

```
IPython console
IP Console 2/A
Python 2.7.10 |Anaconda 2.3.0 (64-bit)| (default, May 28 2015, 16:44:52)
[MSC v.1500 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 3.2.0 -- An enhanced Interactive Python.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
?      -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help    -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
%gui?   -> A brief reference about the graphical user interface.

In [1]: runfile('C:/Users/zach/Google Drive/SMU/MSDS 7330 - File
Organization and Database Management/Week 5/HW2_Q17.py',
wdir='C:/Users/zach/Google Drive/SMU/MSDS 7330 - File Organization and
Database Management/Week 5')

Enter 1 to add a course, 2 to print courses or 3 to exit: 2

Enter full or partial Course Number to search for: foo
No matching courses!

Enter 1 to add a course, 2 to print courses or 3 to exit: 2

Enter full or partial Course Number to search for: MSDS
File Organization and Database Management
Experimental Statistics I

Enter 1 to add a course, 2 to print courses or 3 to exit: 1

Enter Course Number: MSDS6372

Enter Course Name: Experimental Statistics II

Enter 1 to add a course, 2 to print courses or 3 to exit: 5
Please enter a valid option

Enter 1 to add a course, 2 to print courses or 3 to exit: 1

Enter Course Number: MSDS6390

Enter Course Name: Visualization of Information

Enter 1 to add a course, 2 to print courses or 3 to exit: 2

Enter full or partial Course Number to search for: MSDS
File Organization and Database Management
Visualization of Information
Experimental Statistics II
Experimental Statistics I

Enter 1 to add a course, 2 to print courses or 3 to exit: 3

In [2]:

Permissions: RW      End-of-lines: CRLF      Encoding: ASCII
```