CS3200 HW1

Xin Guan

1 Part A

1. Discuss the differences between DDL and DML. What operations would you typically expect to be available in each language?

   Solution:

   DDL is for defining or modifying data structures or data schemas. DDL states the properties including table entities, attributes, relationships required for the application, index, users of a data base and . In DDL and SQL, we are expecting to have statements including **create, drop, alter**

   DML is used for inserting, updating, selecting and deleting data in a database.

2. Describe the difference between data security and data integrity.

   Solution:

   Data security is about protecting the database, the containing data and the management system from internal and external threats. Threats includes stealing data, actions without permissions, and internal abuse of data. However, data integrity is a property of data stored in database. Data integrity requires data to be consistent and correct. Also, to maintain data integrity, we need to do some senity check or set some constraints before operations and manipulations.

3. What advantages do databases provide over early file-based approaches? Give at least three examples.

   Solution:

   (a) File based system is difficult to access since files containing data are distributed on different machiens. While, databases provide a centerlized storage which provides an easier access and smaller duplications. For example, a company need to save duplicated files in different departments so that people from different departments have the access, which might lead to inconsistency and duplications.

   (b) File based system stores data in different formats. While, databases output data in the same format. Therefore, application developers need to parse files with different formats. However, with databases, developers only need to parse interact with one data format, which accelerates the development and reduces bugs and complexcity.

   (c) Databases have fixed queries while file based systems don't. Application developers do not need to implement the algorithm of the data operations and might reuse the queries during the development with databases.

4. What is concurrency control and why does a DBMS need a concurrency control facility?

   Solution:

   Concurrency control is the procedure for managing multiple users' changes to the database simultaneously without conflicting with each other. DBMS need a concurrency because multiple users are reading and writing at the same time and controlling such procedures makes sure user are not doing something conflicts with other users and keep the data correct.

5. What is a transaction? Give an example of a transaction.

   Solution:

   A transaction is a sequence of data operations. we can view transaction as a logical work request to the database.

   For example, updating the score of a student in CS3200 with nuid 123:

   UPDATE cs3200_score
   SET score = 85
   WHERE nuid=123;

6. What is meant by the term 'client-server architecture' and what are the advantages of this approach? Compare the client-server architecture with two other architectures.

   Solution:

   "Client-server architecture" is dividing a system to 2 client and server. Clients manages user interface and run the applications. Server stores the databases and runs the DBMS. They are usually connected by internet.
   The advantages include better performance, wider access to existing databases, possible reduction of client development and hardware cost, reduction in communication costs and increased consistency.

7. In the 2016 InfoWorld article "What eBay looks like under the hood" what problem did eBay have with their product catalog? How is this a database design problem? Compare eBay's approach to the way in which Amazon allows third-party vendors to sell used books.

   Solution:

   eBay is facing the problem of identifying the same products under different circumstances from enormous amount of providers. It is difficult to connect the same products since they are listed by different people with various discriptions.

   They designed a layer structured data on top of the free-form listing data to identify the same products. They have to create a database of all possible products so they can categorize all kinds of products. In terms of books, since books have a unique identifier: ISBN, Amazon can build a database according to existing ISBN data and easily identify the same book from different sellers. However, eBay might need a deeper layer of identification trees to help categorize and identify the same products from various providers.

8. The two assigned articles from the February issue of The Scientist both mention the building of Databases. How do these databases facilitate the scientific research discussed in each article? Is the rational for building a database in each case similar? Justify your answer.

In microbes, Knight and Gilbert is solving the problem of distributed storage of DNA code in different formats. They were acceping samples from all over the world and trying to sequence them under a uniformed protcol and store them in a centerlized database. They are more focusing on tacling the inconsistency and difficulties in protocol transformations. They built this database to regulate and centerlize the data storage and grant public accesses.

In chironomids, Lin is working hard on identify new species and genetic sequences. The article is emphasizing more on contributing to the database. The database helped scientists to compare species with species and possibly identify new ones. The rational for building the database is helping scientists to identify new species.

Both databases have the similar hope that they are helping scientists to access data with better conisstency and larger amount. While, Knight and Gilbert is more focusing on breaking the barriers between data from different researches and help the comparision of data. Lin is focusing on increasing the data and helping detecting new species.

2 Part B

In big data environments, analysts are constantly writing ad hoc database queries to obtain general table statistics for purposes of validation, trend analysis, and to simply gain a deeper understanding of their data. Suppose we had a .CSV text file listing bank customer loan applications. Each line includes data about the customer: age, marital status, education, account balance, and so on. The last column indicates whether the customer was approved for the loan.

The bank manager would like to know if customers with different marital status have, on average, significantly different account balances. In a programming language of your choice (or using pseudocode), write or describe a method that reads through the CSV line-by-line, and outputs the average account balance for customers in each marital status category. The equivalent SQL query, and expected output, is provided below. (Your program will probably require more than three lines of code. Using Python pandas or R dataframes is not allowed – nice try!)

Algorithm 1: Reading CSV and Calculate average balance

```
marital_column = 0
balance_column = 0
// read first line:
csv_file = read_file(’loans.csv’)
attr_line = csv_file.readline()
For i = 0 to length(attr_line)
    If attr_line[i] == ’marital’ :
        marital_column = i
    If attr_line[i] == ’balance’ :
        balance_column = i
map: sum_map // we can access data_map by key
total_line = 0
// Read the data in csv file:
While csv_file not end
    total_line += 1
    line = csv_file.readline()
    marital_stat = line[marital_column]
    If marital_stat not in sum_map :
        sum_map.insert(marital_stat, line[balance_column])
    Else
        sum_map[marital_stat] += line[balance_column]
map avg_map
For key in sum_map.keys
    avg_map.insert(key, sum_map[key] / total_line)
Return avg_map
```