W4111 – Introduction to Databases
Sections 002, V002; spring 2022

Homework 1 – Written Assignment

Instructions

- The homework submission date/time is 06-Feb-2022 at 11:59 PM.
- Submission format is a PDF version of this document with your answers. Place your answers in the document after the questions.
- The name of your PDF must be <UNI>_S22_W4111_HW1_Written.pdf. For example, mine would be dff9_S22_W4111_HW1_Written.pdf
- You must use the Gradescope functions to mark the location of your questions/answers in the submitted PDF. Failure to mark pages will cause point deductions.
- You can use online sources but you must cite your sources. You may not cut and paste text.
- Questions typically require less than five sentences for an answer. You will lose points if your answer runs on and wanders.

  "Verbosity wastes a portion of the reader's or listener's life."

Questions

Question 1: Briefly explain the terms *structured data, semi-structured data* and *unstructured data.* Give an example of each type.

Structured data: Structured data refers to data that has been formatted into a structure before storing. It is based on relational database tables and can be stored in SQL tables with rows and columns. Example: Relational data such as an inventory system for a store.

Semi-structured data: Semi-structured data is more flexible than structured data. It does not necessarily need to be in a relational database but also has some organizational properties. Example: json, XML data.

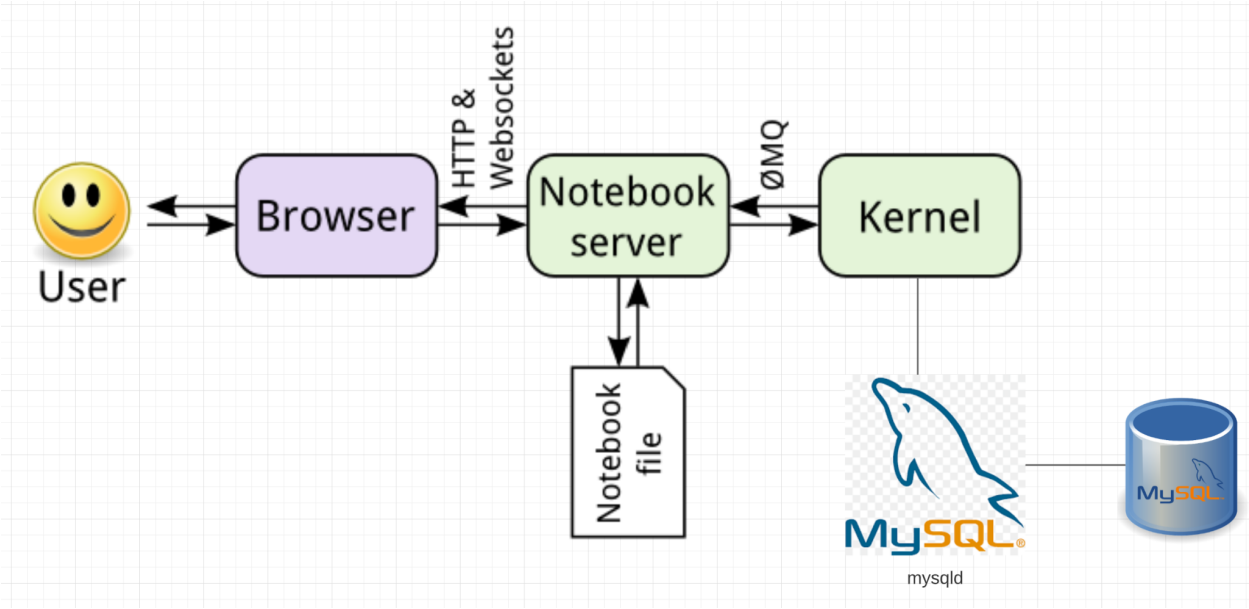
Unstructured data: Unstructured data is data that does not have a predefined structure. It is hard to be put in a relational database and oftentimes needs further processing upon using.
Example: Media logs, PDFs.

https://www.geeksforgeeks.org/difference-between-structured-semi-structured-and-unstructured-data/

Question 2: Briefly explain the concept of *metadata*. For a presentation (PowerPoint, Google Slides), what would be some examples of metadata?

Metadata is data that tells information about other data, but not the content of it. For example, for a presentation, metadata should not contain any content of the actual presentation, but it should give information about things like the file size, the author of the presentation, and the date it was created.

Question 3: The following diagram is an overview of Jupyter Notebook's runtime model when the notebook is using MySQL. Is this a 2-tier application or 3-tier application? Briefly explain why.



This is a 3-tier application. A three-tier application includes the data, the user interface, and a logic process in the middle. In this example, the Kernel, MySQL database, is the data, the user is using the browser as an interface, and the notebook server is the application layer in between these two things.

Question 4: Briefly define and explain procedural and declarative languages. Is SQL procedural or declarative?

Procedural language: in a procedural language, the goal of the program and the set of instructions for the program are provided in advance. The way to process the data does not vary too much even given different file contents, and the data of the program is typically pre-gathered rather than from user input.

Declarative language: On the other hand, for declarative languages, we only tell the computer what problem needs to be solved without giving specific instructions.

SQL is declarative, because it only provides the query but does not specify the steps to get the result.

Question 5: List 4 advantages/differences of database management systems (DBMS) compared to programs and files for data processing. List two disadvantages of DBMS?

Advantages:

1. Ensures data security. DBMS provides a framework that ensures security and privacy.
2. Improved data sharing. DBMS creates an environment for many users to access the data.
3. Minimized data inconsistency. Different versions of the same data can be easily integrated in the DBMS.
4. Improved decision making. DBMS provides a framework to better analyze the dataset.

Disadvantages:

1. DBMS is much more expensive compared to programs/files.
2. Since DBMS is much more complex than normal data files, people need to learn how to efficiently use the system.

http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/462-advantages-and-disadvantages-of-dbms.html

Question 6: In a relational DBMS, columns/attributes should be *atomic.* Briefly explain what this means. If a table has a column *name* of the form “last name, first name”, is this atomic?

Atomic here means that “in each tuple within a relation should consist of a single value” (Codd's original notion from 1969). In other words, “atomic” means something indivisible. “Last name, first name” is not atomic because it can be further divided by the comma to two different data values.

Question 7: Attributes/columns have *types,* e.g. int, varchar(128), timestamp. An attribute/column values must be from a *domain*? What is the difference between a type and a domain (hint: domain constraints)?

Yes, an attribute/column values must be from a domain. Type means built-in data types such as int, varchar(128), string while domain represents “type” in a database with constraints. The difference here is that we can add optional constraints to types to create a new “domain.” For example, the type of a column “weight” is float, but we would say the domain is all positive real numbers.

Question 8: There are four common types of people that interact with a database management system. List and briefly explain each of the four types.

1. Naive Users: people who interact with the system via previously-written programs.
2. Application Programmers: people who write programs for the database.
3. Sophisticated Users: people who access the database using query languages or softwares. They don't write programs.
4. Specialized Users: Write specialized programs for the database. Those programs typically don't fit into the traditional data framework, such as audio, video.

Question 9: Briefly explain the concepts of database *instance* and *schema*?

Instance is the data that is stored in the database at a particular time. Schema is the overall design of the database that can be viewed as a diagram. It contains things such as the table, keys, views, etc.
https://www.javatpoint.com/dbms-data-model-schema-and-instance#:~:text=%E2%86%92%20%E2%86%90%20prev-,Data%20model%20Schema%20and%20Instance,skeleton%20structure%20of%20the%20database.

Question 10: Explain the concept of *physical data independence* and the importance of the concept.

Physical data independence means that we are able to modify the physical level schema without changing the higher level schemas. It is important because it allows a separation of the physical level and the logical or the view level, and we don't need to rewrite the programs when only the physical level is changed. For example, when we want to add or remove files to the system, physical data independence ensures our change remains at the physical level.