Lily Kendall
Homework 1
Due 9/11/24

Problem 1.7: List four significant differences between a file-processing system and a DBMS.

1. DBMS have integrity constraints which enforces the correctness of the data and limits duplicates in the data. File-processing systems do not have the same constraints and are therefore more vulnerable to duplicate data.
2. DBMS are more efficient at getting specific data because of its set programs that come with it. The file-processing system relies on application programs to retrieve data, which make fetching data less efficient.
3. DBMS have built in security procedures that allow only certain users to have access. File-processing systems must have the security procedures built by someone, and so it is less secure.
4. File-processing systems do not have a protocol for when multiple users update the data at the same time, or for when there is a system failure. DBMS have transaction management and recovery systems in place for when there are multiple updates at the same time or the system crashes.

Problem 1.8: Explain the concept of physical data independence and its importance in database systems.

The physical level describes the lowest level of how the data is stored. Physical data independence means that the physical level can be changed without the rest of the data schema also having to change or be affected. Physical data independence is important because a database may have many people working on different parts, so if someone working on the physical level must change something or do maintenance of some kind, the others working on the logical and view levels will not also have to make changes or even know the details of the physical level. Physical data independence also allows developers to make changes easily since it will not affect the rest of the schema, so it is easier to optimize and modify if the data needs.

Problem 1.9: List five responsibilities of a database-management system. For each responsibility, explain the problems that would arise if the responsibility were not discharged.

1. A DBMS is responsible for storing the data. Additionally, the data must be stored well to make fetching and updating convenient. If this responsibility was not discharged the data may end up being inaccurate and hard to fetch from.
2. A DBMS has features that save the data in the event the system fails and crashes. If this responsibility was not discharged then when the system crashes, the data may not be able to be recovered.

3. A DBMS has security systems in place that controls who is allowed access in the database. If this task was not discharged then foreign users may be able to get into the database causing security issues.
4. A DBMS is responsible for letting multiple users access and update the data at the same time without conflicts. If this responsibility was not discharged, two users making updates at the same time may not save correctly.
5. A DBMS has certain constraints on the data that must be met. It is the responsibility of the DBMS to make sure the constraints are met. If this responsibility was not met, there may be duplicates in the data, or inconsistent or incorrect data.

Problem 1.11: Assume that two students are trying to register for a course in which there is only one open seat. What component of a database system prevents both students from being given that last seat?

The concurrency-control manager keeps track of multiple users access simultaneously, and ensures that two updates at the same time do not causes conflicts. In this case, the concurrency-control manager would make sure that only one user gets the spot and the database updates correctly with only one user registered and one user not registered.

Problem 1.12: Explain the difference between two-tier and three-tier application architectures. Which is better suited for web applications? Why?

On a two-tier application architecture, there is the user's view and the server. The user interacts with the interface, which sends the data directly to the server. On a three-tier architecture system, the user interface sends the data to another tier which processes the data and the request to the server. After the processing, the middle tier sends the data to the server. The three-tier architecture is better suited for web applications because there is a third layer to handle processing. This allows for more web traffic because the user end can take on more. The third layer also makes changes to the user end easier because the data processing does not also need to be updated.

Problem 1.14: Explain why NoSQL systems emerged in the 2000s, and briefly contrast their features with traditional database systems.

In the 2000s, social media came about along with a large increase in data analytics. This produced a need for other forms of storing data and applications that had functionality in analytics for large amounts of data. NoSQL allows for different structures of data and data storage, while a traditional database system is only built to store tables. NoSQL also has support for methods such as APIs and various query languages while a traditional database only features SQL for queries.

Problem 1.15: Describe at least three tables that might be used to store information in a social-networking system such as Facebook.

The first table could store user profile information such as name, username, date of birth, and email. Since no two people can have the same username and usernames are easily identifiable, it makes for a good primary key for the table.

The second table could store a user's posts, with attributes such as caption, date of post, username of the user who posted, and an identifier such as a URL to the image. The username in the posts table should match a username in the user table. The primary key could be a combination of post URL and username, or a new ID.

The third table could store the posts that a user has liked. The attributes may be the username of the user who liked the post, the username of the user who posted the post, the URL of the post, and the time it was liked. The usernames should match usernames in the user table. The primary key could be a combination of usernames of who liked and posted the post, along with the image URL, or a new ID.