Databases Homework 1
8/31/23
Jax Lubkowitz

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

DPMS were built with the faults of their predecessor, the file-processing system, in mind. While file-processing systems can have consistency problems in the event of the crash, DBMS implement atomicity properties, allowing either operations to execute fully or not at all to ensure the consistency of the database. DBMS can also have physical data independence, allowing for changes in the physical level without impacting the logical or view level. File-processing systems require hardcoded files and any shift in physical level requires the corresponding programs to be adapted accordingly. DBMS also generate meta-data on the database schema and data. This data dictionary helps when finding or changing information. In addition, DBMS are built to allow for differing models such as relational or object-based data models which was not possible in a file-processing system.

In summary, 4 significant differences between these databases are:

1. Atomicity
2. Physical Data Independence
3. Data Dictionary
4. Data-models

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

Physical data independence refers to the separation between the physical and logical/view levels of a database system. To exhibit physical data independence, shifts and changes at the physical level should not impact or require re-building of higher levels (logical/view) to account for base level (physical) changes. Furthermore, the structure of the physical level need not be considered when developing higher levels if the database is physically independent.

Implementing physical data independence in a database allows for more dynamic development and maintenance over time. PDI allows for expansion or restructuring of the physical level without forcing a complete rebuild of the logical level and programs used within the database saving time and resources.

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 dis-charged.

Database management systems (DBMS) play many important roles and failure can cause many problems. Primarily DBMS are responsible for storing, retrieving, and manipulating data. If any one of these three actions is compromised the database loses virtually all functionality. Without the ability to store data; a database will never have any value. Without the ability to retrieve data; a database is not worth having. Without the ability to manipulate data; a database can only fill so many roles. As technology has grown, it has also become critical that data be accessible from multiple places and people at the same time while still providing accurate and consistent results. Along with this, atomicity arose to ensure that changes on one user's end do not differ from another user due to mid-database use system crashes creating discrepancies in the results (actions must be completed in full or not at all). If this was to fail, the data stored would be compromised and often wrong, voiding its purpose. In addition, DBMS play an important role in data security. Databases contain private as well as proprietary information that can harm many, such as social security numbers or bank statements.

In summary, 5 responsibilities of a DBMS are:

1. Data Storage
2. Data retrieval
3. Data manipulation
4. Maintaining database integrity in the event of a crash
5. Protecting 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?

Concurrency Control Systems are responsible for preventing simultaneously submitted data changes to cause inconsistencies. Without this system, both students might be able to register for the class, breaking the confines on class size and introducing differences in the database as both students cannot be registered.

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

In a two-tier application architecture system, the user runs applications to call the database directly from their machine to the database system. In a three-tier application architecture system, the user calls messages to an application server (which they do not have access or control too) which then interacts with the database. This allows for better scalability as instead of many queries coming from different machines, the application server can manage them into a more efficient workflow with queries coming from only the one source. In addition, this adds another layer of security to the database as users do not have direct access.

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

NoSQL development was driven by a huge increase in data traffic and technological advancements that required more flexible data-type boundaries. NoSQL was implemented by social platforms due to its scalability (as social media was a relatively new concept and growing rapidly). Unlike SQL, NoSQL allows for temporary inconsistency and varying datatypes across the database and would converge at a further time.

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

Social network databases must have many tables within to organize and manage the data. One such table, USER_PROFILE, might consist of usernames, and profile info (hometown, birthday, posts, relationship status, etc.). Another important table is a table of who each account follows, FOLLOWING. In other words, many complex tables are needed to connect all the accounts into an actual network. While these tables are all user oriented, there must also be information on who can access and edit Facebook as a platform for updates and modifications. This requires a table of valid admin users, DEVELOPER_ACCESS, and their passwords to access the system.