Jackie Diep

Assignment 1

CSC 411 G001

2/19/21

Chapter 1

- **1.7**

List four significant differences between a file-processing system and a DBMS

1. A file-processing system can have problems with data redundancy and inconsistency due to file duplication and changes not being made across all applicable files.
2. Data cannot be retrieved in an efficient way like more responsive data-retrieval systems.
3. It is much easier to ensure atomicity in a DBMS than it is in a file-processing system.
4. The ad hoc manner of file-processing systems can make security enforcement more difficult than in a DBMS

- **1.8**

Explain the concept of physical data independence and its importance in database systems

Physical data independence is a layer of abstraction around the possibly more complex physical-level structures within the database to simplify things at the logical level. This also helps database administrators determine what data to keep or remove as the logical layer is much easier to perceive.

- **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. Buffer management: The DBMS would not be able to handle much data from the disk due to the smaller size of main memory.
2. Storage management: The DBMS would not be able to interact with the file manager and store, retrieve, or update data in the database.
3. Query processing: Users would be forced to understand the specific implementation of the physical level of the system.
4. Transaction management: System failure would lead to inconsistent states in data and concurrent interactions with the DBMS would lead to conflicts.
5. Integrity management: Integrity constraints would not be enforced

- **1.10**

List at least two reasons why database systems support data manipulation using a declarative query language such as SQL, instead of just providing a library of C or C++ functions to carry out data manipulation.

1. It is easier to learn for non-programmers
2. It leaves the choice of efficiency up to the system instead of the user

- **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 transaction manager

- **1.13**

List the two features developed in the 2000s and that help database systems handle data-analytics workloads.

Programming frameworks like “map-reduce” to facilitate parallelism and “NoSQL” systems that gave programmers greater flexibility.

- **1.14**

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

They were developed due to new data-intensive applications and the need for rapid development. They are different from traditional DBMS because they use “not only SQL”, which gives programmers greater flexibility when working with new types of data due to a lack of the strict data consistency that traditional DBMS have. They also follow the “eventual consistency” model that allows for distributed copies of data to be inconsistent with the idea that they would eventually become consistent without further updates.

Chapter 2

- **2.14**

Consider the employee database of Figure 2.17. Give an expression in the relational algebra to express each of the following queries:

a. *Find the ID and name of each employee who works for “BigBank”.*

   $\Pi_{\text{ID, person\_name}}$ (employee $\bowtie_{\text{employee.id = works.id}}$ ($\sigma_{\text{company\_name = “BigBank”}}$ (works)))

b. *Find the ID, name, and city of residence of each employee who works for “BigBank”.*

   $\Pi_{\text{ID, person\_name, city}}$ (employee $\bowtie_{\text{employee.id = works.id}}$ ($\sigma_{\text{company\_name = “BigBank”}}$ (works)))

c. *Find the ID, name, street address, and city of residence of each employee who works for “BigBank” and earns more than $10000.*

   $\Pi_{\text{ID, person\_name, street, city}}$ ($\sigma_{\text{company\_name = “BigBank” ^ salary > 10000}}$(employee $\bowtie_{\text{employee.id = works.id}}$ works))

d. *Find the ID and name of each employee in this database who lives in the same city as the company for which she or he works.*

$\Pi_{ID,\ person\_name}\ (\sigma_{company.city\ =\ employee.city}\ (employee \bowtie_{employee.id\ =\ works.id}\ works \bowtie_{works.company\_name\ =\ company.company\_name}\ company))$

- **2.15**

Consider the bank database of Figure 2.18. Give an expression in the relational algebra for each of the following queries:

a. *Find each loan number with a loan amount greater than $10000.*

   $\Pi_{loan\_number}\ (\sigma_{amount\ >\ 10000}\ (loan))$

b. *Find the ID of each depositor who has an account with a balance greater than $6000.*

   $\Pi_{ID}\ (\sigma_{balance\ >\ 6000}\ (account \bowtie depositor))$

c. *Find the ID of each depositor who has an account with a balance greater than $6000 at the "Uptown" branch.*

   $\Pi_{ID}\ (\sigma_{balance\ >\ 6000\ \wedge\ branch\_name\ =\ \text{"Uptown"}}\ (account \bowtie depositor))$

- **2.18**

Write the following queries in relational algebra, using the university schema.

a. *Find the ID and name of each instructor in the Physics department.*

   $\Pi_{ID,\ name}\ (\sigma_{dept\_name\ =\ \text{"Physics"}}\ (instructor))$

b. *Find the ID and name of each instructor in a department located in the building "Watson".*

   $\Pi_{ID,\ name}\ (\sigma_{building\ =\ \text{"Watson"}}\ (department \bowtie_{dept\_name\ =\ dept\_name}\ instructor))$

c. *Find the ID and name of each student who has taken at least one course in the "Comp. Sci." department.*

   $\Pi_{ID,\ name}\ (\sigma_{dept\_name\ =\ \text{"Comp. Sci."}}\ (student \bowtie_{student.id\ =\ takes.id}\ takes \bowtie section \bowtie course))$

d. *Find the ID and name of each student who has taken at least one course section in the year 2018.*

$\Pi_{\text{ID, name}} (\sigma_{\text{takes.year = 2018}} (\text{student} \bowtie_{\text{student.id = takes.id}} \text{takes} \bowtie \text{section}))$

e. *Find the ID and name of each student who has not taken any course section in the year 2018.*

$\Pi_{\text{ID, name}} (\sigma_{\text{takes.year} \neg \text{2018}} (\text{student} \bowtie_{\text{student.id = takes.id}} \text{takes} \bowtie \text{section}))$