

# CIS 560 Database System Concepts

## Fall 2006

### Homework 1 of 10: Problem Set (PS1)

#### Warm-up: Basics, Relations and Relational Algebra

#### Model Solutions

**Note:** For short answer and essay questions, the instructor's solutions are intended as an example only. If you have any questions concerning notation in proofs and other problem solutions, see the instructor during office hours or contact me by e-mail.

1. **(2 points) Database Management Systems (DBMS).** Adapted from Problem 1.8, p. 32 Silberschatz *et al.* 5e. List four responsibilities of a database management system. For each responsibility, explain the problems that would arise if the responsibility were not discharged.

*A database management system mediates storage, manipulation and retrieval of information. It should ensure or facilitate:*

a) **Efficient data manipulation.** *A DBMS should make manipulation of its contents (e.g., through relational joins) more efficient by indexing records. If it implements relational operations in a naïve way, the worst-case running time – proportional to the product of the sizes of the relations – could be many times slower than that of an efficient implementation. Given the current sizes of large databases (millions to billions of records), this could mean a factor of hundreds to thousands.*

b) **Consistency.** *A DBMS must update its contents in a safe way to reflect the results of possibly concurrent transactions. If concurrent transactions are not conducted in a way such that updates are noninterferent (as well as integral) then harmful effects such as invalid values can occur.*

c) **Security and Isolation.** *A DBMS should ensure that only authorized reads and writes occur with respect to its contents. Views should protect private information, malicious changes (data diddling) should be prevented, and unauthorized processes should not be able to access and transmit protected information stored within the database. If this is not done for databases with sensitive information, then serious breaches of security at private, local, national, or international level may occur.*

d) **Integrity and Durability.** *A DBMS should ensure that fields of records in different parts of the database are consistent. In particular, it can be used to ensure **referential integrity**, i.e., that values that appear in one relation for a given set of attributes also appear for another set of attributes in another relation (Section 4.2.5 Silberschatz *et al.* 5<sup>e</sup>, p. 129 – 132). Without integrity constraints, updates can happen that can cause erroneous conditions: overdrafts on financial accounts at banks and departments, for example. The results should also persist, or the user cannot be assured of the reliability of confirmation messages.*

2. **(2 points) Data Manipulation Languages (DMLs).** Problem 1.9, p. 32 Silberschatz *et al.* 5e. 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.

a) **Representation Independence and Transparency (Strong Typing).** A DML lets the programmer manipulate relations as first-class data structures, decreasing the chance of low-level mistakes by hiding details of how relations are represented and transformed.

b) **Efficiency.** It may be more efficient to perform certain operations such as relational joins using sort arrays and hash tables (Silberschatz et al. Section 13.5, p. 542 – 555) instead of iterator patterns and functions. The database itself can keep the sequential storage records needed for bookkeeping (Silberschatz et al. Chapters 11 – 12).

3. **(2 points) Database administration.** Problem 1.11, p. 32 Silberschatz et al. 5e. What are five main functions of a database administrator?

Following Silberschatz et al. Section 1.12.2, p. 27:

a) **Data Definition.** A DBA designs the basic schema and implements it by using a data definition language (DDL) or data definition user interface to create tables.

b) **Initial storage structure design.** A DBA translates the initial schema design (e.g., star/constellation, data warehouse) into a storage design by specifying where tables actually reside and what kind of indexing information is needed for the specified organization. Examples include data dictionaries (Section 11.8, p. 472 – 474) and multilevel indices (Section 12.2.2, p. 485 – 486).

c) **Updating and changing schemas.** A DBA maintains the schema, adding or dropping fields to reflect reorganization of the database, refactoring it into different tables when needed, and changing the actual physical organization of tables.

d) **Administering access privileges.** A DBA defines, grants and revokes privileges for accessing tables, either directly or by means of views.

e) **Backups, restores, and performance tuning.** A DBA administers regular incremental backups of database contents, storage quotas, and restoration of data corrupted or lost in crashes. He or she may also look at usage statistics in order to decide on an acceptable use policy (CPU quotas) and perform basic load balancing.

For problems 4-6, consider the two project options for this semester. **Choose one** for your solution (this need not be your final project choice). Indicate which domain you are discussing.

- a. Preparing a university admissions and grade database for data mining to identify strong predictors of academic probation and dismissal.
- b. Populating a university phonebook database from the K-State White Pages.

4. **(2 points) Practical Databases.** What do you think is a good server platform for the project, and why?

a) An entity-relational (E-R) data model for this database can be developed using Microsoft Access or even UML tools such as Rational Rose or ArgoUML. A relational database server such as ORACLE, Microsoft SQL Server, MySQL, or postgresql can be used to implement queries and provide views. All of these RDBMS platforms support queries for analytical applications, although ORACLE comes with some data mining tools bundled. Once data has been centralized and transformed, exporting specific fields to flat file (or relational) format is straightforward. For an interactive visualization, C# with ODBC .NET or Java with JDBC and JSP are good front-end development platforms. To facilitate compliance with state and federal privacy laws and provide assurance of privacy

to students whose data is stored in this RDB, an industrial-strength RDBMS such as the four named above is recommended.

b) MySQL and postgresql in particular have zero-cost (as opposed to free software) licenses and support the offline processing and data warehousing operations needed to carry out the data transformations. This project has more of an online component than the grad student database one. C# .NET with ODBC or Java Server Pages (JSP) with JDBC and a graphical user interface (GUI) are recommended, although an application with a shorter life cycle could be implemented using only HTML forms. The more strictly an RDBMS observes the ACID properties (Atomicity, Consistency, Isolation, Durability; Chapter 14, Silberschatz et al.) the easier it is to maintain a client form or application for.

5. (2 points) **Bad Database Design.** What is wrong with exporting a single flat file (one table in ASCII form) from the original database? Give one example of data redundancy issue and one of a data integrity issue.

a) **Redundancy:** A monolithic table results in a tremendously large number of replicated instances of student transcripts over time. At a minimum, the student data (which changes slowly and rarely over the course of a student's enrollment) should be isolated from relations about registrations and courses.

**Data integrity:** There are many values shared across replicated instances of records – e.g., students' demographic information – that need to be propagated per semester. Unless the replication is eliminated or integrity constraints added, the correctness of transactions (such as a transfer student registration) cannot be guaranteed.

b) **Redundancy:** A monolithic table results in a large number of replicated instances of person-phone records, possible once per instance of an entity such as office. Making the model hierarchical or objective would work well here.

**Data integrity:** This project has more automated updates (e.g., for transfers, new hires, graduates) and requires consistency to be maintained across parts of the conceptual data model.

6. (2 points) **Query Example.** Give an example in English of a real select query that a user might submit over a web form, and write it in relational algebra.

a) Show the list of all students who went on academic probation and were ultimately dismissed from the MS program along with the undergraduate GPA.

$$\Pi_{\text{student\_name, undergrad\_GPA}} (\sigma_{\text{probation=true, dismissal-reason="probation"}} (\text{MS-students} \bowtie \text{undergraduate-transcripts}))$$

b) Show the list of all faculty members in the CIS department with their office phone numbers.

$$\Pi_{\text{person\_name, office\_phone}} (\sigma_{\text{status="faculty", department="CIS"}} (\text{person} \bowtie \text{office-phone}))$$

For problems 7-8, refer to Sections 2.2 – 2.3 and consider the following relational database, where the primary keys are underlined:

employee (person\_name, street, city)  
 works (person\_name, company\_name, salary)  
 company (company\_name, city)  
 manages (person\_name, manager\_name)

7. **(2 points) Relational Algebra: Queries.** Problem 2.5, parts a, d. Consider the following relational database above, where the primary keys are underlined. Give an expression in the relational algebra to express each of the following queries:

- a. Find the names of all employees who work for First Bank Corporation.

$$\Pi_{\text{person\_name}} (\sigma_{\text{company\_name}=\text{"First Bank Corporation"}} (\text{works}))$$

- d. Find the names of all employees in this database who live in the same city as the company for which they work.

$$\Pi_{\text{person\_name}} \left( \Pi_{\text{person\_name, city}} (\text{employee}) \bowtie \Pi_{\text{person\_name, city}} (\text{works} \bowtie \text{company}) \right)$$

8. **(2 points) Relational Algebra: Updates.** Problem 2.7, parts a, b.

- a. Give all employees of First Bank Corporation a 10 percent salary raise.

$$\begin{aligned} \text{works} \leftarrow & \Pi_{\text{person\_name, company\_name, salary}} * 1.10 (\sigma_{\text{company\_name}=\text{"First Bank Corporation"}} (\text{works})) \cup \\ & \Pi_{\text{person\_name, company\_name, salary}} (\sigma_{\text{company\_name} \neq \text{"First Bank Corporation"}} (\text{works})) \end{aligned}$$

- b. Give all managers in this database a 10 percent salary raise, unless the salary would be greater than \$100,000. In such cases, give only a 3 percent raise.

$$\begin{aligned} \text{managers} \leftarrow & \Pi_{\text{manager\_name}} (\text{manages}) \\ \text{manager-works} \leftarrow & \Pi_{\text{person\_name, company\_name, salary}} (\sigma_{\text{person\_name} = \text{manager\_name}} (\text{managers} \times \text{works})) \\ \text{non-manager-works} \leftarrow & \text{works} - \text{manager-works} \\ \text{works} \leftarrow & \Pi_{\text{person\_name, company\_name, salary}} * 1.10 (\sigma_{\text{salary} * 1.10 < 100000} (\text{manager-works})) \cup \\ & \Pi_{\text{person\_name, company\_name, salary}} * 1.03 (\sigma_{\text{salary} * 1.10 \geq 100000} (\text{manager-works})) \cup \\ & \text{non-manager-works} \end{aligned}$$

9. **(2 points) Relations.** Give an example of a:

- a. One-to-one function between a subset of A and a subset of B that is not onto.

$$\begin{aligned} A &= \{A_1, A_2\} \\ B &= \{B_1, B_2, B_3\} \end{aligned}$$

$$f = \{ (A_1, B_1), (A_2, B_2) \}$$

$f$  is one-to-one because every element of  $A$  maps into exactly one element of  $B$ .

$f$  is **not** onto because there is an element  $B_3$  of  $B$  that is not the mapping of any element of  $A$ .

- b. Onto function between a subset of A and a subset of B that is not one-to-one.

$$\begin{aligned} A &= \{A_1, A_2, A_3\} \\ B &= \{B_1, B_2\} \end{aligned}$$

$$f = \{ (A_1, B_1), (A_2, B_2), (A_3, B_2) \}$$

$f$  is onto because every element of  $B$  is the mapping of some element of  $A$ .  
 $f$  is **not** one-to-one because not every element of  $A$  maps into exactly one element of  $B$ .  $A_2$  and  $A_3$  both map into  $B_2$ .

**10. (2 points) Databases and Data Structures.** Consider a table in a relational database.

- a. What kind of C++ or Java data structure would you use to represent it, and why?

*A linked list of arrays or a hash table of arrays is a simple way to represent a table with variable number of rows and a fixed number of columns. If attributes can be added or deleted using ALTER TABLE ADD A D and ALTER TABLE DROP A, then the width of rows (records) could grow or shrink. In that case, it is more flexible to make rows into dynamic data structures represented using linked lists (or actually B-trees or B+ trees as shown in Chapter 12).*

- b. What is the time complexity of updating a row given a primary key using your scheme?

*For a linked list of arrays, the worst-case time complexity of an update is linear in the number of rows. For a hash table of arrays, the time complexity of an update depends on the hash function: it is worst-case linear in the number of rows but could be in  $O(1)$  in the expected case with a uniform hash function. Note that using arrays to represent rows gives us random access to fields of each record.*