Prof. Li-Yan Yuan

CMPUT 291: File and Database Management Systems

# Final Examination

April 19, 2001

It is a close-book examination and the time for the test is 120 minutes. There are nine (9) questions over four (4) pages. The value of each question is indicated in [ ] and the total is 100. Good luck to all of you.

1. Performance in sorting is often measured in terms of the number of comparisons. Explain why the number of comparisons is not adequate for measuring performance in sorting large files. [4]

2. An airport database contains the following information:

   (a) Each airplane has a registration number, and is of a specific model.

   (b) The airport accommodates a number of airplane models, and each model is identified by a model number(e.g. DC-10) and has a capacity and a weight.

   (c) A number of technicians work at the airport. The information about technicians includes the name, SIN, address, phone number, and salary.

   (d) Each technician is an expert on one or more plane model(s), and each plane model may have one or more expert(s).

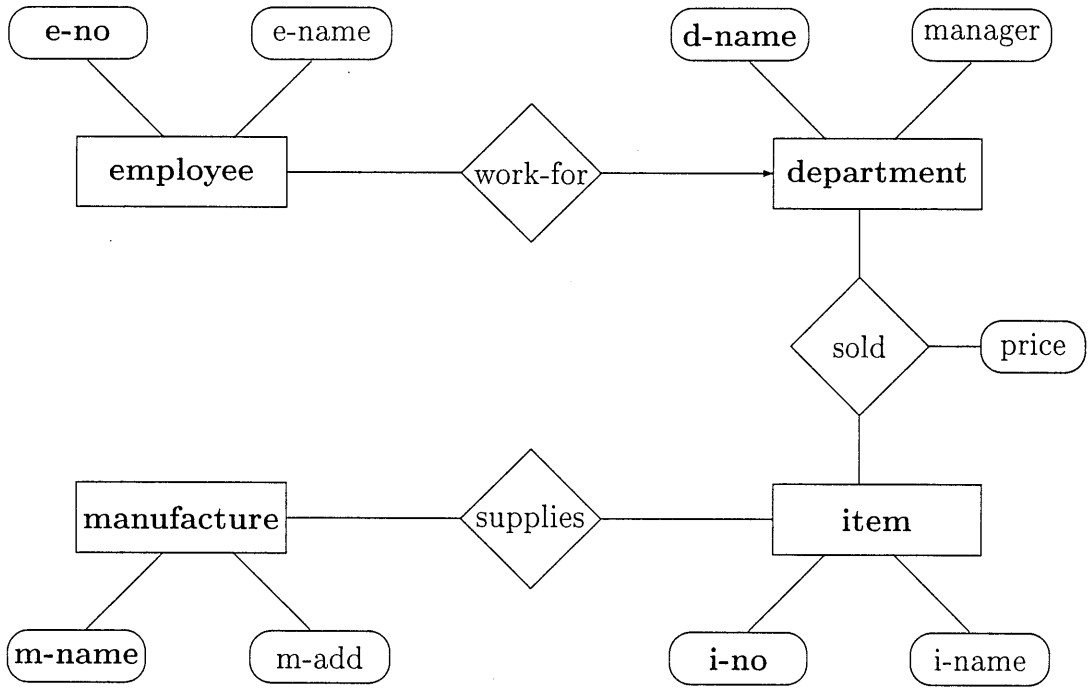
   (e) The airport has a number of tests that are carried out periodically to ensure that airplanes are still air-worthy. Each test has a Federal Aviation Administration (FAA) test number, a name, and a maximum possible score.

   (f) The FAA requires the airport to keep track of each time that a given airplane is tested by a given technician using a given test. For each testing event, the information needed is the date, the number of hours the technician spent doing the test, and the test score that the airplane received on the test.

   Draw an ER diagram for the airport database. Specify the key and participate constraints for each relationship set with the notations used in the textbook and/or lecture notes.

   (You may indicate any reasonable assumptions to justify your design choices.) [8]

3. Consider a company database represented by the following E-R diagram.




Find an appropriate relational database schema (i.e., present a list of table schemas) from the ER diagram. For each table schema, you need to give a list of attributes, and indicate the primary key for the table. No SQL statement, and no data types.

Note that the arrow points to the one side of the corresponding relationship set and the bold indicates the key attributes. State any reasonable assumptions you may have. [8]

4. Consider the following employee database.

```
employee(e_id, e_name, street, city)
work(e_id, c_id, salary)
company(c_id, c_name, city)
manager(e_id, m_id)
```

The first two tables store the information about the e_id, name, street address, city, the employer (c_id) and salary of each employee. The third table stores information about each company, including its id, name, and location; the last one indicates who is whose boss. Both e_id and m_id in the last table must be an e_id of the first table. Note that an employee may work for more than one company.

Use SQL to express the following queries. (You results' may contain duplicate tuples.)

(a) Find the e_id and name of all employees who work for a company that located at St. Albert.

(b) Find the e_id and name of all employees who are a manager for some company.

(c) List the e_id, name and total salary for all employees. (Note that an employee may work for several companies, and his/her total salary is the sum of salaries from all the companies he/she works for.)

(d) The most expensive employee(s) of a company is the employee of the company with the highest salary. List the name, and salary of the most expensive employees of all companies.

(e) Find the name of the company that has the smallest payroll among all the companies in the database. The payroll of a company is the sum of all salaries paid by the company to its employees.

(f) List the name of all the companies whose employees do not work for any other company. [36]

5. Consider a relational database about a university with the following three relations

   teach(Prof, Course)
   take(Student, Course, Grade)
   advise (Prof, Student)

   The first relation indicates the courses a prof teaches; the second tells what courses each student takes and the corresponding grades; and the last indicates advisors of a student. Give relational algebra expressions to

   (a) retrieve all courses taken by students whose advisor is "Peter Jennings";

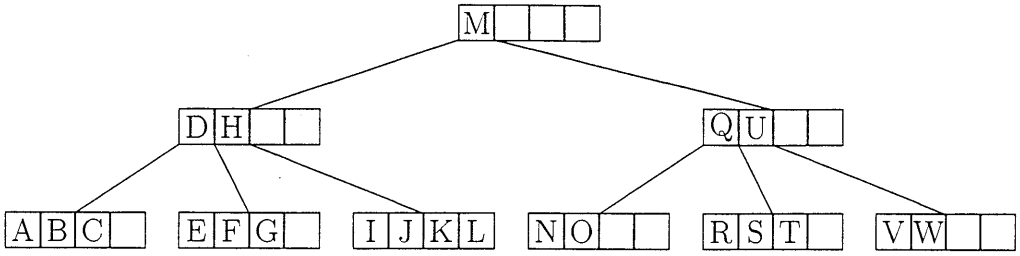
   (b) retrieve all professors who never advise any student who is taking a course taught by him/herself. [12]

6. Show the B-tree of order four that results from loading the following keys in order.

   ```
   3, 7, 16, 24, 14, 19, 21, 15, 1, 5, 2, 8, 12, 6
   ```

   By order four we mean the maximum number of keys in a page is 3, and the minimum number of keys in a non-root page is 1. Note that only the basic split-and-promotion is considered, and the promoted key is from a right page. (You don't have to show your steps, but you will get 0 if there are errors but no steps.)[8]

7. Show, step by step, the trees that result after each of the keys $A$, $B$, $Q$, and $R$ is deleted, in the given order, from the following B-tree of order five. (You may consider plain B-tree operations only, i.e., only redistribution and concatenation.) [8]




8. $B^+$tree is designed for efficient file accesses with both indexed search and sequential access. When designing a file with indexed search as the only access mode, is there any gain by using a $B^+$ tree over a B-tree? Be specific. [8]

9. The following algorithm is used to insert a record into a hash file using the progressive overflow strategy for collision.

   It is assumed that the file has N slots, represented by slot[i] for $i = 0, \ldots N-1$, and that slot[i] = *new* if there never was a record stored in it, and slot[i] = *tombstone* if there once was a record in it but it is now empty.

   In the algorithm, hash(key) represents a hash function returning an integer between 0 and N-1, and INPUT(slot[i]) and OUTPUT(slot[i]) represent the read from and write into slot[i] respectively.

```
Algorithm INSERT(key)
begin
        h:=hash(key);
        k:= 0;
        INPUT(slot[h]);
        while slot[h+k] ≠ new and slot[h+k] ≠ tombstone do
                k:= k + 1;
                INPUT(slot[h+k])
        end while;
        OUTPUT the record into slot[h+k]
        return_message("the key has been inserted");
end
```

   (a) Is there anything wrong with the algorithm? Explain your answer.

   (b) Design an algorithm DELETE(key) to delete a record from the same hash file. Of course, your algorithm must be compatible with INSERT(key). [8]