Dr. Stefan Brass
School of Information Sciences
University of Pittsburgh

January 5, 2000

INFSCI 2710 "Database Management" — Example for Final Exam —

Notes

- These are the exercises from the final exam of the Spring 1999 course.
- There was 1 hour, 45 minutes time to solve these exercises.
- Please note that there will be some new types of exercises, for instance related to concurrency control:
  - Is a given schedule serializable? Would Oracle allow this schedule?
  - Which rows would Oracle lock?
  - Can two given programs get into a deadlock?
- There might also be more exercises related to PL/SQL, e.g.:
  - What is the output of a given program?
  - Is a given program syntactically correct?
- Exercises related to access rights (GRANT, REVOKE) and views.
- I probably will mention other types of possible exercises during the course and especially during the last lecture before the exam.
- The final exam will be on April 26. Please check the web page for changes.
- You can again use books and notes.
- You cannot use notebooks etc. during the exam. Please practice SQL long enough so that you can write SQL queries without feedback from the DBMS. I.e. you must get them right in the first try. SQL is the most important topic in the exam.
- If the midterm exam shows that you are weak in certain topics, there might be another exercise of the same type in the final exam. E.g. this example exam contains again the translation from the ER-model into the relational model.
- I might ask in the final exam for the minimal grade you would be willing to accept before you would prefer a G grade and much (!) additional work. Of course, there is no guarantee that I will respect your wish. I can give G grades only to a very limited number of students. I might also ask for a grade with which you would be quite happy (only for statistical purposes). Please think about these questions before you come to the exam.

Exercise 1 (SQL Queries) 16 Points

You are given a database of a (slightly unusual) web server:

- Page(ID: number(4), Path: varchar(80) not null, Owner: varchar(10) not null, Last_Modified: date not null, text: long not null)
  contains an entry for every web page on the server: An identifying number, a path, the owner, the date of last modification, and the text of the page (the HTML source code). This web server has the data of the web pages stored inside the database, and not in operating system files.

Page				
ID	Path	Owner	Last_Modified	Text
1	/~sbrass/db/index.html	sbrass	26-APR-99	...
2	/~sbrass/db/soft.html	sbrass	24-APR-99	...
3	/~sbrass/db/webdb.html	sbrass	10-APR-99	...
⋮	⋮	⋮	⋮	⋮

- Request(No: Number(5), IP: varchar(15) not null, when: date, ID → Page not null)
  is used to log requests for web pages: Whenever a browser requests a web page, the IP address of the requesting machine, a timestamp (attribute “when”) and the ID of the page are stored in the table “Request”. In addition, since there is no good key, an identifying number for the request is added.

Request			
No	IP	when	ID
10000	136.142.116.203	01-MAR-99	1
10001	130.75.26.3	05-APR-99	1
10002	136.142.116.203	27-APR-99	2
⋮	⋮	⋮	⋮

- Keyword(Word, ID → Page)
  is used for a local search function: It contains keywords (search terms) for each page. A page can be decribed by any number of keywords (0, 1, or many).

Keyword	
Word	ID
Databases	1
Databases	2
Databases	3
WWW	3
⋮	⋮

All attributes are not null. Since the "not null" constraint is implied by the primary key constraint, it is explicitly listed only for the non-key attributes. Of course, the database state is only an example. Your queries must work with any database state.

Formulate the following queries in SQL:

a) Print ID number and path of all web pages referring to both, "Databases" and "WWW" (so they have at least these two entries in the "Keyword" table).

b) Print ID number and path of all web pages which were not requested in 1999, i.e. for which there is no request in the database on `'01-JAN-99'` or later.

c) How many distinct words are in the table "`Keyword`"? (Print only the count.)

d) Print for each web page (ID and Path), which was requested at least 10 times, the number of requests. The output should be sorted by this number, with the most often requested page first.

Exercise 2 (Views) 6 Points

a) Define a view which shows each user the access count for his/her pages including the score 0, if the page was never accessed. The view should have three columns, the ID of the page, the path, and the access count.

b) Is this view updatable? Please explain your answer briefly (one sentence).

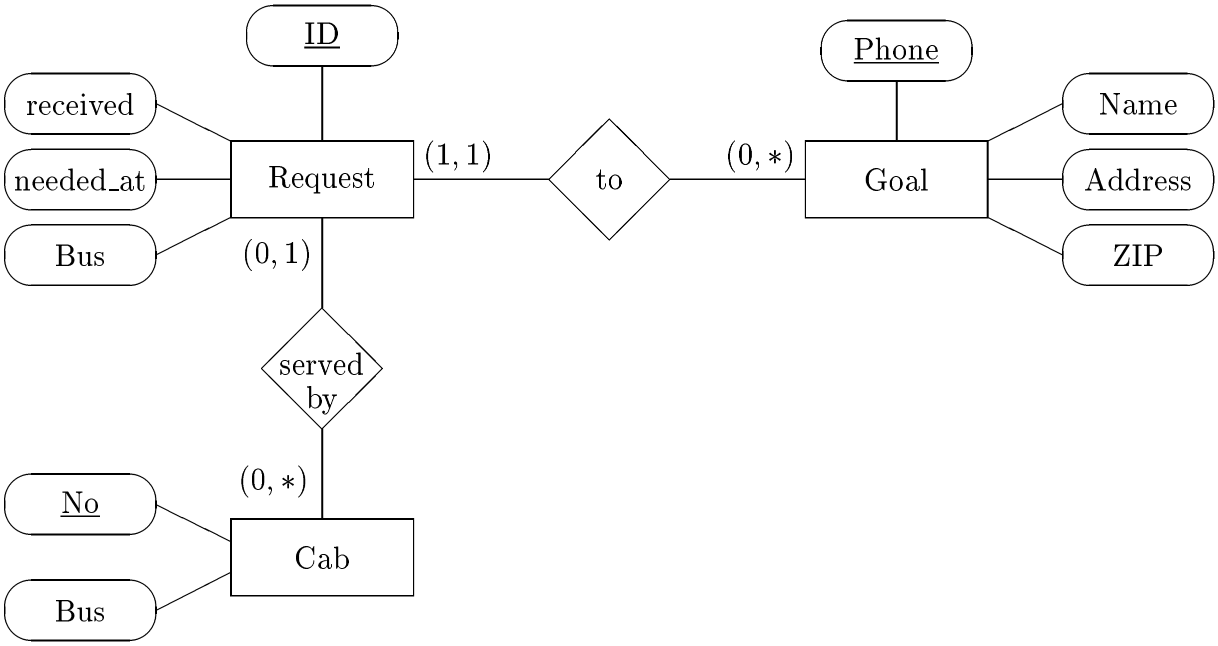
Exercise 3 (ER to Relational Mapping) 10 Points

You are given the following ER-schema for the database of the headquarters of a taxi/cab company. It contains information about:

- Goal addresses: The street address, the ZIP-code, the name of the person who ordered the taxi, and his/her phone number at this address. Phone numbers are used as keys, so when the person requests the next time a cab, the chances are good that the corresponding address is already in the database. The phone number needs 10 digits, the ZIP-code 5, the street address can be up to 40 characters long, and the name up to 20.
- Requests: The timestamp of when the request came in (attribute "received"), and the date and time of when the taxi should arrive (attribute "needed_at"). Both attributes have the type "`date`" in Oracle. It is also possible to request a big taxi

(mini bus) with more space for people and baggage. This is modelled as an attribute "Bus" which can be "Y" or "N". Each request is assigned a unique ID (8 digit number).

- Taxis/Cabs: Each cab has a unique number. In addition we store whether it is a bus or not (as above).
- Whereas a request must have a goal address, we enter only later which cab is scheduled to serve the request.



Translate this schema into the relational model. Please write SQL "`CREATE TABLE`" statements for the resulting tables. These should include the specification of keys, foreign keys, and not null constraints. If further constraints are needed to make the relational schema equivalent to the ER-schema, please explain them in natural language. In addition, please define the following constraints and defaults:

- The time a request is received must be before the time at which the cab is needed.
- The time a request is received is by default the time the tuple is entered into the database ("`SYSDATE`").
- None of the attributes shown in the ER-schema is optional (all must be filled out).

Exercise 4 (Integrity Constraints, Trigger) 5 Points + 3 Extra

An additional constraint for the taxi/cab database is that if a customer requested a bus, this request may only be served by a cab with `Bus='Y'`. Parts a) and b) of this exercise are required, part c) will earn you three points of extra credit.

a) Write an SQL query which finds violations of this integrity rule. Note that it is ok to send a mini bus even if the the customer requested a normal taxi.

b) Which are critical operations for this constraint? I.e. which of insert/update/delete on which table could lead to a violation of the constraint? In case of an update, specify also the attribute(s) which have to be changed. You do not have to consider updates on key attributes. Also the attribute “Bus” of “Cab” is never changed.

c) Write an Oracle trigger which enforces the constraint. It should check only the necessary tuples, and not entire tables. Call `raise_application_error` if the constraint is violated.