CPSC 310: Database Systems / CSPC 603: Database Systems and Applications
Exam 1
October 5, 2005

Name: _____

Instructions:

1. This is a closed book exam. Do not use any notes or books, other than your 8.5-by-11 inch review sheet. Do not confer with any other student. Do not use any computer equipment.
2. Show your work. Partial credit will be given. Grading will be based on correctness, clarity and neatness.
3. I suggest that you read the whole exam before beginning to work any problem. Budget your time wisely—according to the point distribution.
4. There are 5 questions worth a total of 100 points, on 7 pages (including this page).

DO NOT BEGIN THE EXAM UNTIL INSTRUCTED TO DO SO. GOOD LUCK!

Please sign the academic integrity statement:

"On my honor, as an Aggie, I have neither given nor received unauthorized aid on this academic work. In particular, I certify that I have not received or given any assistance that is contrary to the letter or the spirit of the guidelines for this exam."

Signature: _____

References for these problems:

- our textbook
- *Modern Database Management, Seventh Edition,* Hoffer, Prescott, and McFadden, Pearson Prentice Hall, 2005)
- *Database System Concepts, Fifth Edition,* Silbershcatz, Korth, and Sudarshan, McGraw-Hill, 2006)
- Gradiance web site

1. (16 pts total) Short answers.

(a) (4 pts) Give two design guidelines for E/R diagrams.

(b) (4 pts) Define two kinds of anomalies can occur with poorly-structured relations.

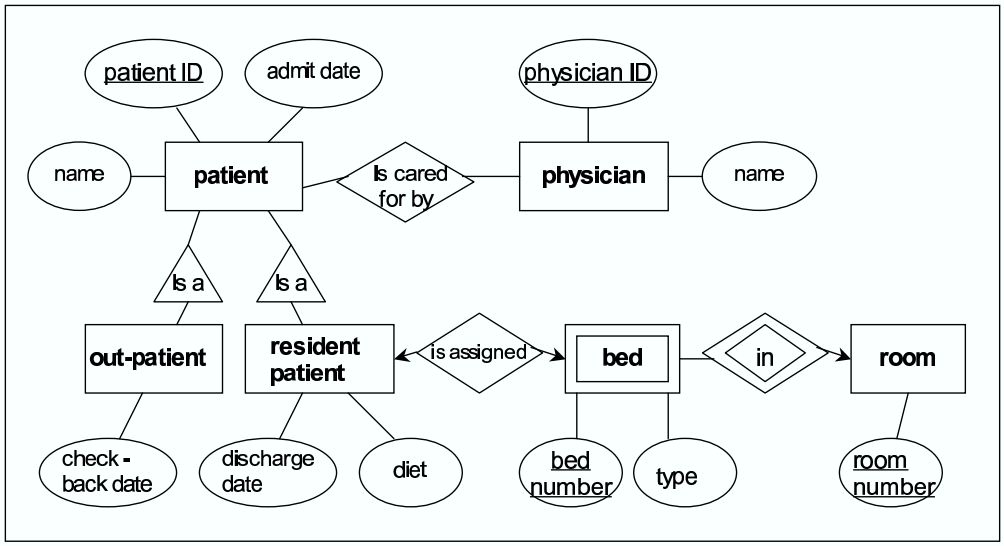
(c) (4 pts) What is the definition of Third Normal Form (3NF)? What is its advantage over Boyce-Codd Normal Form?

(d) (4 pts) In SQL, name one advantage of set semantics over bag semantics and one advantage of bag semantics over set semantics.

2. (25 pts total) E/R Diagrams.

(a) (12 pts) Draw a well-structured E/R diagram for the following application. If you believe that additional reasonable assumptions are needed, state them clearly.

Customers submit orders for products. Each order has a unique ID and a date. A customer can submit any number of orders (including none). Each customer has a unique ID as well as name and address. An order consists of one or more order lines, which are numbered consecutively. Each order line refers to exactly one product and also indicates the quantity of that product that is ordered. Each product has a unique id, a text description and a unit price.



(b) (13 pts) Convert the E/R diagram above into a relational schema. Use the algorithm presented in class and combine relations according to those rules. Handle the is-a hierarchy using the NULLs method.

3. (24 pts total) Normal Forms.

Consider the relation
`Lending(branch_name, branch_city, assets, customer_name, loan_number,amount)`
with the following functional dependencies:

- `branch_name` $\rightarrow$ `assets branch_city`
- `loan_number` $\rightarrow$ `amount branch_name`

(a) (4 pts) How many keys does this relation have? List them.

(b) (5 pts) Compute the closure of the set of attributes { `loan_number` }.

(c) (4 pts) Calculate all additional nontrivial FDs that follow from those given above.

(d) (4 pts) Why is the relation `Lending` not in Boyce-Codd Normal Form?

(e) (7 pts) Convert the relation `Lending` into Boyce-Codd Normal Formal. Be sure to show your work, i.e., explain the process you are using.

4. (14 pts total) Relational Algebra.

Consider the relations $R(A, B)$ and $S(B, C)$ with the following instances:

R:

A	B
0	1
2	3
0	1
2	4
3	4

S:

B	C
0	1
2	4
2	5
3	4
0	2
3	4

(a) (7 pts) Compute $\gamma_{B,AVG(C)}(S)$ (grouping).

(b) (7 pts) Compute $\gamma_{A,MAX(C)}(R \bowtie S)$ (grouping and natural join)

5. (21 pts total) SQL.

Consider the following (informal) database schema for our familiar example from class:

- `Candies(name, manf)` : lists candies and their manufacturers
- `Stores(name, city, addr, phone)` : lists the name of stores, their city, (street) address, and phone number
- `Consumers(name, city, addr, phone)` : lists consumers by their name, city, (street) address, and phone number
- `Likes(consumer, candy)` : indicates which consumers like which candies
- `Sells(store, candy, price)` : indicates which stores sell which candies at which prices (assume each store sells each candy at only one price)
- `Frequents(consumer, store)` : tells which consumers shop at which stores

Write queries to find the following information.

(a) (7 pts) Find the names of all candies, and their prices, sold by the store 'HEB'.

(b) (7 pts) Find the name and phone number of every consumer who likes the candy 'Snickers'.

(c) (7 pts) Find all stores frequented by both 'Moe' and 'Curly'.