CS411 Database Systems

Fall 2004, Prof. Chang

Department of Computer Science
University of Illinois at Urbana-Champaign

Midterm Examination

October 22, 2004

Time Limit: 90 minutes

- Print your name and NetID below. In addition, print your NetID in the upper right corner of every page.

Name: ____________________ **NetID:** ____________________

- Including this cover page, this exam booklet contains **7** pages. Check if you have missing pages.
- The exam is closed book and closed notes. No calculators or other electronic devices are permitted. Any form of cheating on the examination will result in a zero grade.
- Please write your solutions in the spaces provided on the exam. You may use the blank areas and backs of the exam pages for scratch work. Please do not use any additional scratch paper.
- Please make your answers clear and succinct; you will lose credit for verbose, convoluted, or confusing answers. *Simplicity does count!*
- Generally, we think one minute per point is a reasonable allocation of time; so plan your time accordingly. *You should look through the entire exam before getting started, to plan your strategy.*

Problem	1	2	3	4	5	6	7	8			Total
Points	15	20	20	20							75
Score											
Grader											

Problem 1 (*15 points*) Misc. Concepts

For each of the following statements, indicate whether it is *TRUE* or *FALSE* by circling your choice. You will get *1 point* for each correct answer, *-0.5 point* for each incorrect answer, and *0 point* for each answer left blank.

(1) *True* *False*

The relational data model was proposed by Dr. Edgar Codd in early 1970's.

(2) *True* *False*

Schema normalization not only reduces potential data redundancy but also enhances query efficiency.

(3) *True* *False*

SQL stands for *Structured Query Language*.

(4) *True* *False*

Any relationship satisfying referential integrity constraint will satisfy single value constraint as well.

(5) *True* *False*

A schema in BCNF will be in 3NF as well.

(6) *True* *False*

If $A \to B$ and $C \to D$ hold, then $AC \to BD$ also holds.

(7) *True* *False*

Given a relation with attributes A, B, C, D, E : if $AB \to C$, $BC \to AD$, and $D \to E$ hold, then AB is a key.

(8) *True* *False*

A multivalued dependency $\mathsf{A} \twoheadrightarrow \mathsf{B}$ implies functional dependency $\mathsf{A} \to \mathsf{B}$.

(9) *True* *False*

SQL is a *declarative* query language, in which we simply declare *what* we want, but not *how* to compute, in formulating a query.

(10) *True* *False*

With the six basic operations (union, difference, selection, projection, product, and renaming), relational algebra is *Turing complete*. Other operations are just syntactic sugar and can be derived from the basic operations.

(11) *True* *False*

In SQL, a *view* can be used like a stored relation in any operations.

(12) *True* *False*

In SQL, the value *NULL* is ignored in any aggregation.

(13) *True* *False*

For any SQL query, there exists a *unique* translation into relation algebra.

(14) *True* *False*

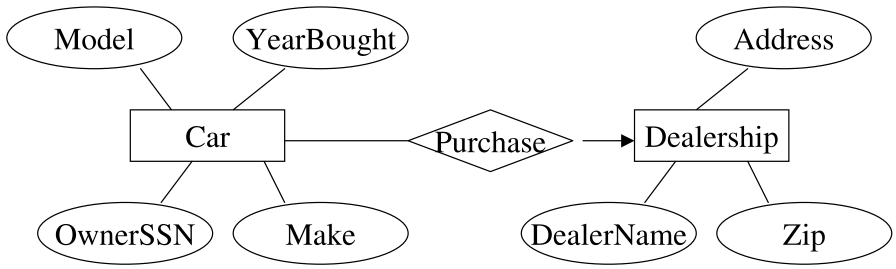
An *aggregate* function, *e.g.*, *Sum* and *Avg*, returns a value computed from a set of values. Thus, *Min* and *Max* are not aggregate, since they only return a single value.

(15) *True* *False*

Unlike *assertions* which must be, in principle, checked any time when data is modified, *triggers* allow user to specify when such checking should occur.

Problem 2 (*20 points*) ER and Schema Design

Consider the following E/R diagram:



(a) For each of the following statement, write a functional dependency (FD) that best captures the statement. (*6 points*)

1. No two makes have cars with the same model name.
2. No owner has two cars of the same model.
3. No two dealerships have the same name.

(b) From the E/R diagram and the set of FDs you listed above, can you derive new FDs? If *no*, explain why not. If *yes*, derive such new FDs– give as many an you can, but no more than three. (*6 points*)

(c) Now suppose the FDs you specified above are true. Translate this diagram into a relational schema in BCNF. (*8 points*)

Problem 3 (*20 points*) Query Languages

Consider our typical "drinker" database with the following relations.

$$\mathsf{Drinker}(\mathsf{drinkerName}, \mathsf{street}, \mathsf{age})$$
$$\mathsf{Bar}(\mathsf{barName}, \mathsf{owner}, \mathsf{street})$$
$$\mathsf{Frequent}(\mathsf{drinkerName}, \mathsf{barName})$$

We ask you to write queries. Please write simple and *non-redundant* queries – Note that we will *really* check if your answers are unnecessarily complex.

(a) In relational algebra, write a query to return the bars that Sally frequents. (*6 points*)

(b) In relational algebra, write a query to return each drinker who frequents *only* bars on the same street that he lives. (*7 points*)

(c) In SQL, write a query to return the bars whose frequent drinkers are "young"– in particular, with average age below 37. (*7 points*)

Problem 4 (*20 points*) Views and Constraints

Consider the following relational schema:

Account(<u>accountNumber</u>, branchName, balance)
Branch(<u>branchName</u>, street, city, assets)
Customer(<u>customerSSN</u>, street, city)
Deposit(customerSSN, accountNumber, Amount)

(a) List all the attributes (in the four tables) that are foreign keys and indicate what attributes they are referencing. (*4 points*)

(b) Define a view *BigBranch* that gives for each branch its branchName, city, and assets. The branch should have more than 50 accounts and the total balance of all accounts is greater than $1,000,000. (*8 points*)

(c) Suppose we want to check that, for each branch, the total balance of all accounts is less than or equal to the assets of the branch. Complete the following SQL statement, by specifying ⟨*condition*⟩. Note, by definition, such an "assertion" statement will enforce the ⟨*condition*⟩ to hold true at all times. (*8 points*)

CREATE ASSERTION *BalanceCheck* CHECK ⟨*condition*⟩