

**Comprehensive Exam (A)**  
**Databases**  
**Spring 2011**

1) (30 points) Circle T or F for each of the following.

- |  |   |   |
|--|---|---|
| — “SQL” stands for Structured Query Language.  | T | F |
| — An advantage of SQL over other query languages is that SQL does not have transactions.   | T | F |
| — SQL was first proposed by IBM in early 2001.   | T | F |
| — One purpose of a query optimizer is to improve query performance.  | T | F |
| — Every candidate key is also a super key.   | T | F |
| — Suppose a one-to-one relationship set R exists between two entity sets E1 and E2. In addition, let K1 and K2 be candidate keys for E1 and E2, respectively. Then the union of K1 and K2 forms a super key for R. | T | F |
| — The <i>division</i> operation is not one of the six basic relational algebraic operators.  | T | F |
| — All relational algebraic expressions are “safe,” by definition.  | T | F |
| — The <i>natural join</i> of two relations with no common attributes is an empty relation.   | T | F |
| — The closure of a set of attributes can be computed using Armstrong’s axioms.   | T | F |
| — Armstrong’s axioms include <i>reflexivity</i> , <i>transitivity</i> and <i>union</i> .   | T | F |
| — Computing the closure of a set of attributes eliminates attributes from that set.  | T | F |
| — The decomposition rule states that if $AC \rightarrow B$ and $A \rightarrow B$ and $C \rightarrow B$ .   | T | F |
| — If a set of relational schemes is in BCNF, then that set also has a loss-less join.  | T | F |
| — Let R be any relational scheme that is not in BCNF. Then R can always be decomposed into a collection of BCNF relational schemes that has a loss-less join.  | T | F |

2) (30 points) Consider the following relational schemes. Note that attributes forming the primary key for each relation have been underlined.

student( <u>student-name</u> , street, city)	-- Basic student information.
offering( <u>department</u> , <u>number</u> , <u>section</u> , time, population)	-- Courses currently offered; for CSE5260 department is "CSE" and number is 5260. Section would be D-1, D-2, E-1, E-2, etc.
titles( <u>department</u> , <u>number</u> , title)	-- Course titles; "CSE5260" is "Database Systems"
enrollment( <u>student-name</u> , <u>department</u> , <u>number</u> , <u>section</u> )	-- Indicates which students are enrolled in which courses and sections.

Give an SQL query for part (a).

(a) The department, number, section, title, and population for every course section.

<u>department</u>	<u>number</u>	<u>section</u>	<u>title</u>	<u>students</u>
CSE	5260	D-2	Database Systems	76
MTH	5100	E-1	Discrete Math	87
PSY	4260	D-3	Abnormal Psychology	65
CHM	2035	D-4	Introduction to Chemistry	98
:				

Give a relational algebraic expression for part (b).

(b) A list of student names along with the number of courses that each student is enrolled in.

<u>student-name</u>	<u>number-of-courses</u>
Jones	2
Smith	4
Brown	3
:	

Give a tuple calculus expression for part (c).

(c) A list of the names of students who are currently enrolled in every class that "Jones" is currently enrolled in (in the same *class*, but not necessarily the same *section*).

3) (10 points) Recall that during database design, an ER is frequently converted to tables.

(a) Describe how a many-to-one relationship is converted to tables.

(b) Describe how a many-to-many relationship is converted to tables.

(c) Describe how a one-to-one relationship is converted to tables.

4) (20 points)

(a) Define 1NF.

(b) Define 3NF.

(c) Define BCNF.

5) Consider the following set of functional dependencies for the relational scheme  $R=(A,B,C,D,E,F,G)$ .

$A \Rightarrow B$

$A \Rightarrow D$

$C \Rightarrow EA$

$F \Rightarrow G$

(a) (2 points) Give the attribute closure of C.

(b) (2 points) Give a candidate key for R.

(c) (2 points) Give one non-trivial functional dependency that is in the closure of the above set of functional dependencies. Use Armstrong's axioms to prove that the functional dependency is in the closure.

(d) (2 points) Consider the decomposition of R into  $R_1=(A,B,F,G)$  and  $R_2=(A,C,D,E)$ . Does this decomposition have a lossless join (yes or no)? If so, then explain why, and if not, then explain why not.

(e) (2 points) Now consider the following set of functional dependencies for the same relational scheme:

$A \Rightarrow B$

$A \Rightarrow D$

$C \Rightarrow EA$

$F \Rightarrow G$

$C \Rightarrow B$

Does the decomposition of R into  $R_1=(A,B,C, F,G)$  and  $R_2=(A,C,D,E)$  have a lossless join (yes or no)? If so, then explain why, and if not, then explain why not.