Database Systems
Comprehensive Exam
Spring 2012

Name:

1) (60 points) Circle T or F for each of the following.

— DDL stands for Database Development Language. T F

— A collection of database operations that are to be treated collectively as a single logical unit, and which either succeed or fail as a whole, is referred to as a transaction. T F

— SQL was invented by Microsoft. T F

— The purpose of ER data modeling is to determine low-level details such as how data items are stored on disk. T F

— The purpose of an index is to save space. T F

— Every super key is also a primary key. T F

— Suppose a one-to-many 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 primary purpose the query optimizer is to manage concurrent transactions. T F

— The six basic relational algebraic operators includes *select*, *project*, *union*, *intersection*, *cartesian product* and *rename*. T F

— An *unsafe* tuple calculus expression has the property that its' result set cannot be computed from the database contents. T F

— The *natural join* operator can be defined in terms of *cartesian product*, *selection*, and *projection*. T F

— The *theta join operator* is the same as the *aggregate operator*. T F

— A tuple calculus expression can contain at most one existential quantifier. T F

— Tuple calculus is an example of a pure, non-procedural query language. T F

— One disadvantage of tuple calculus is that it is impossible to represent a query that requires a natural join. T F

— A *relation* is an ordered collection of tuples. T F

— A universal quantifier is identical to an existential quantifier. T F

— The *division operator* is an example of an unsafe relational algebraic operator. T F

— A foreign key reference from a table T1 to a table T2 must reference the primary key of T2. T F

— An SQL select statement consisting of *select*, *from* and *where* clauses is sufficient to perform a natural join.

T F

— Every SQL statement that contains a *having* clause must also include a *where* clause. T F

— When applied to a numeric column of a table, the SQL aggregate operators *avg*, *min*, *max* and *sum* will ignore all tuples that contain a null value.

T F

— Grouping in SQL (i.e., the group by clause) is always implemented in SQL by performing a sort. T F

— The closure of a set of attributes is the same as the closure of a set of functional dependencies. T F

— The closure of a set of functional dependencies is larger (i.e., contains more functional dependencies), generally, than a canonical cover for that set of functional dependencies.

T F

— Let R be any relational scheme that is in 1NF, but not in 3NF. Then R can always be decomposed into a collection of 3NF relational schemes that have a loss-less join and preserve dependencies.

T F

— Let R be any relational scheme that is in 1NF, but not in BCNF. Then R can always be decomposed into a collection of BCNF relational schemes that have a loss-less join and preserve dependencies.

T F

— If a relational scheme is in BCNF then it must be in 3NF T F

— The union rule states that if A->B and C->B then AC->B. T F

— If a set of relational schemes has a loss-less join, then that set also preserves dependencies. T F

2) (35 points) Consider the following relational schemes. Note that attributes forming the primary key for each relation have been underlined.

branch(branch-name, city, assets)
customer(customer-name, street, city)
loan(loan-number, branch-name, amount)
account(account-number, branch-name, balance)
borrower(customer-name, loan-number)
depositor(customer-name, account-number)

Give an SQL statement, a relational algebra expression, or a tuple calculus expression (your choice, but only give one) for each of the following. Note that your choice may be different for each.

(a) A list of those branches that have at least one account with a balance over $10,000.

(b) A list of the names of those branches that provide both loans and accounts.

(c) A list of the names of those customers who have a loan and an account at the same branch. For example, “Smith” has a loan and an account at the Perryridge branch.

(d) A list of the names of those customers who have an account at every branch where “Jones” has a loan.

3) (5 points) Give an example of a relational scheme, and an associated set of functional dependencies, that is in 3NF but not in BCNF.