Midterm 1 Review

Midterm 1: February 22 on class time

You have just been hired as a consultant for a big airplane manufacturer. Talking with the people in the company, you get the following information.

The database contains information about employees, factories and parts.

Each employee has a SSN, name, and salary. Employees are uniquely identified by their SSN.

Each factory has an id, name and a budget. The id uniquely identifies a factory.

Each part has an id and a name. The id uniquely identifies a part.

Each employee reports to at most one other employee.

Each employee works in at least one factory.

Each part is manufactured in exactly one factory.

Each part is a component of zero or more other parts.

A.1) (13 points) Draw the Entity-Relationship diagram for the above application.

A.2) (13 points) If you were to write SQL statements to create the relations corresponding to the above ER diagram. What are the tables that you will create and what are the primary key(s) and foreign key(s) for each table (You do NOT have to write the exact SQL statements, just mention the table names and the primary/foreign key(s) for each table).

B.1) (16 points) Assume we have two relations R(A,B), and S(B,C). All three attributes (A, B, and C) are integer attributes. Assume that Relation R contains the following tuples: (1,2), (2,3), and (3,4). Assume that Relation S contains the following tuples (2,2), (2,3) and (5,1). Recall that a key is a minimal superkey, and that a key is not a superkey.

a) (4 points) Give an example of an attribute(s) that cannot be a key for relation S.

b)(4 points) How many rows are in the result of the equi-join between R and S on attribute B?

B 1) (16 points) Assume we have two relations R(A,B), and S(B,C). All three attributes (A, B, and C) are integer attributes. Assume that Relation R contains the following tuples: (1,2), (2,3), and (3,4). Assume that Relation S contains the following tuples (2,2), (2,3) and (5,1). Recall that a key is a minimal superkey, and that a key is not a superkey.

c) (4 points) How many rows and columns are in the Cartesian product of R and S

d) (4 points) How many rows and columns are in the result of the following relational algebra:

$$\pi_{A,C}(R \times S)$$

B.2) (8 points) Consider tables R and S created with the following SQL:

```
CREATE TABLE R (C INT PRIMARY KEY, D INT );
CREATE TABLE S (B INT PRIMARY KEY, C INT REFERENCE R(C) ON DELETE CASCADE);
```

Suppose that tables R and S contain r and s tuples, respectively. What would be the maximum number of tuples that can appear in the result of an equi-join of R and S based on attribute C. Justify your answer.

PART C. Relational Algebra (24 points total)

Consider the following relations. Keys are underlined. Attributes with the same name represent a foreign key constraint:

Branch (BranchName, BranchCity, assets)

Customer (CustomerID, CustomerName, CustomerCity, CustomerStreet)

Loan (LoanID, BranchName, amount)

Borrower (CustomerID, LoanID)

Account (AccountID, BranchName, Balance)

Depositor (CustomerID, AccountID)

C.1) (8 points) Write the following query in relational algebra: “Find the ID of each customer of the bank who has an account but not a loan.”

Branch (BranchName, BranchCity, assets)

Customer (CustomerID, CustomerName, CustomerCity, CustomerStreet)

Loan (LoanID, BranchName, amount)

Borrower (CustomerID, LoanID)

Account (AccountID, BranchName, Balance)

Depositor (CustomerID, AccountID)

C.2) (8 points) Write the following query in relational algebra: “Find the ID of each customer who lives on the same city as customer ID 400”

Branch (BranchName, BranchCity, assets)

Customer (CustomerID, CustomerName, CustomerCity, CustomerStreet)

Loan (LoanID, BranchName, amount)

Borrower (CustomerID, LoanID)

Account (AccountID, BranchName, Balance)

Depositor (CustomerID, AccountID)

C.3) (8 points) Write the following query in relational algebra: “Find the name of each branch that has at least one customer who has an account in the bank and live in Minneapolis.

PART D. SQL Queries (26 points total)

Consider the following relations (same exactly as PART C). Keys are underlined. Attributes with the same name represent a foreign key constraint:

Branch (BranchName, BranchCity, assets)

Customer (CustomerID, CustomerName, CustomerCity, CustomerStreet)

Loan (LoanID, BranchName, amount)

Borrower (CustomerID, LoanID)

Account (AccountID, BranchName, Balance)

Depositor (CustomerID, AccountID)

D.1) (8 points) Write the following query in SQL: “Find the ID of each customer of the bank who has an account but not a loan.” (This is the same query as C.1)

Branch (BranchName, BranchCity, assets)

Customer (CustomerID, CustomerName, CustomerCity, CustomerStreet)

Loan (LoanID, BranchName, amount)

Borrower (CustomerID, LoanID)

Account (AccountID, BranchName, Balance)

Depositor (CustomerID, AccountID)

D.2) (9 points) Write the following query in SQL: “Find the name of each branch that has at least one customer who has an account in the bank and who lives in Minneapolis.” (This is the same query as C.3)

Branch (BranchName, BranchCity, assets)

Customer (CustomerID, CustomerName, CustomerCity, CustomerStreet)

Loan (LoanID, BranchName, amount)

Borrower (CustomerID, LoanID)

Account (AccountID, BranchName, Balance)

Depositor (CustomerID, AccountID)

D.3) (9 points) Write the following query in SQL: “Find the IDs of customers with highest loan amount.”