Introduction to Database Systems

Problem Set 1

Due: January 20, 2015 at 11:59 PM

You are designing a database for a bank. It has the following requirements:

- Account holders (or customers) have ids, names, addresses, and phone numbers
- Bank accounts have id numbers, types (e.g., savings, checking), and balances
- Transactions have amounts, timestamps, and an affected account (the amount is negative for debits). If customer A sends money to account holder B, this creates two transaction entries.
- A bank account has one or more owners
- A customer can have any number of accounts
- Each account can participate in any number of transactions

1. Create an entity-relationship diagram to describe these design requirements. Clearly show the relationships as 1:1, 1:N, or N:N.
2. Use relational algebra to pose the following queries:
   - Show me the names of all of the bank account holders with a balance greater than $\$N$.
   - Create a list of savings account ids with greater than n owners.
   - List all of the names of customers that received deposits greater $\$N$.
3. If we have n account holders, $2n$ accounts, and $1000n$ transaction tuples, is your relational algebra plan the most efficient ordering of these operators? Why or why not?
4. Translate the queries in Question 2 to SQL.
5. Identify the functional dependencies associated with this schema.
6. Normalize this schema. Start with a single relation and decompose it as appropriate. Is the result in Boyce-Codd Normal Form? Why or why not?
7. Pick a domain from your own experience, describe it in English, and the develop an E-R diagram for it. Alternatively, interview someone about a domain in their experience (and not yours) and create an E-R diagram for it.