CSE 532 Midterms (100 points)
28\textsuperscript{nd} October: 5:20-6:40pm (80 minutes)
Closed Book/Notes.

Name:
1. **Functional Dependencies (25 points).**

Consider the relation $R(A, B, C, D)$ with a set $F$ of *non-trivial* functional dependencies. All the functional dependencies in $F$ have only one attribute on either (right hand or left hand) side. We are given that $A \rightarrow B$ is in $F$, and $R$ is in BCNF.

(a) If $\{A\}$ is the only key, what is $F$? Give all possible correct values of $F$.

(b) If $\{A\}$ and $\{B\}$ are the only keys of $R$, then *how many* possible values of $F$ are there? Justify your answer (without necessarily listing them all). Note that $F$ is a *set* of functional dependencies.
2. SQL (25 points).

Consider the table Frequents(drinker, bar) where a tuple \((d, b)\) signifies that the drinker \(d\) frequents the bar \(b\). Assume that the table doesn’t have any duplicates.

We define a drinker \(d_1\) to be more social than a drinker \(d_2\) if \(d_1\) frequents every bar that \(d_2\) also frequents. Write an (non-recursive) SQL query to find the most social drinker (i.e., who is more social than every other drinker). You may use any SQL construct (aggregates, subqueries, etc.) you wish.
3. **Datalog (25 points).**

Write a stratified datalog query for the previous question (most social drinker).
4. **Linear Hashing (25 points).**

Suppose we start with an empty linear hash table, with only (empty) buckets for 0 and 1. Our hash function produces $k$ bits for some very large $k$. Each block holds three records. Also, unlike the class example, we use the threshold 100%; that is, we do not add an extra bucket until the number of records exceeds thrice the number of buckets. To be more precise, the addition of a bucket occurs after a request to insert into the table is made, but before the insertion itself is done (thus, the threshold never exceeds 100%).

We insert records into the table, and by strange coincidence, the results of applying the hash function to the first key is 00...000, the result of applying it to the second key is 00...001, for the next 00...010, then 00...011, 00...100, 00...101, and so on. That is, the hash value of the $i^{th}$ key is $i - 1$ in binary.

**Question:** What is the first record that causes an overflow block to be created? Explain your reasoning briefly.