Part One: Short Answer questions

1

a)

RS1
SELECT MAX(Salary) FROM table
WHERE Gender = Female AND School = Science AND Position = Lecturer;

RS2
SELECT AVG(Salary) FROM table
WHERE Gender = Female AND School = Science AND Position = Lecturer;

Since RS1 would be the same as RS2, we know that all female science lecturers in the database have the same salary as that is the only way for the maximum salary to be equivalent to the average salary.

b)

Lecturers, RS1
SELECT SUM(Salary) FROM table
WHERE Position = Lecturer

Non-Lecturers+Dunn, RS2
SELECT SUM(Salary) FROM table
WHERE (Gender = Female AND School = Science AND Position = Lecturer)
OR NOT Position = Lecturer

All, RS3
SELECT SUM(Salary) FROM table

Dunn's salary = **RS2** - (**RS3** - **RS1**)

Since *(RS3 - RS1)* will give us sum of the Non-lecturers salaries we can take that result from the RS2 to give us just Dunn's salary.

2

Random Oracle Model is used for proving systems security when the system uses cryptographic hash functions and requires strong randomness assumptions in the hash functions output. In reality, hash functions do not have the mathematical properties required by

the proof and so a random oracle (a theoretical function that produces truly random output based on its input) must be used in its place to complete the proof.

3

The method of protecting against statistical inference by restricting query set sizes to the following algorithm $k \leq X(C) \leq N - k$ fails because statistical inference could still be made using query sets that are within the bounds of $k \leq X(C) \leq N - k$.

An example of this is assuming you know that the max size of a department is 5 and you want to protect the average salary for any single department. In this example *k* would equal 5. However the protected information could still be discovered through inference using the following queries:

Number of employees: 32

RS1 // *5 <= Query Set Size (~10) <= (32 - 5)*
SELECT SUM(Salary), COUNT(*) FROM employees
WHERE Department = “Dept1” OR Department = “Dept2”

RS2 // *5 <= Query Set Size (~10) <= (32 - 5)*
SELECT SUM(Salary), COUNT(*) FROM employees
WHERE Department = “Dept1” OR Department = “Dept3”

RS3 // *5 <= Query Set Size (~10) <= (32 - 5)*
SELECT SUM(Salary), COUNT(*) FROM employees
WHERE Department = “Dept2” OR Department = “Dept3”

To calculate the number of employees in Dept1

Let a = Count for Dept1, b = Count for Dept2, c = Count for Dept3
Therefore (a + b) = RS1.Count, (a + c) = RS2.Count, (b + c) = RS3.Count

Dept1 Count = (RS1.Count + RS2.Count - RS3.Count) / 2
= ((a+b)+(a+c)-(b+c)) / 2
= ((2a + b + c) - (b + c)) / 2
= 2a / 2
= a

We can use this same formula to calculate the sum of department 1 as well, therefore to get the average wage for the single department all we need to do is take the calculated sum of Dept1 and divide it by the calculated count of Dept1.

4

The least accurate response would be the subset of 500 elements that produced an average furthest from the true average for the 10,000 elements. This would be either the greatest 500 elements in the data or the least 500 elements in the data dependent on how the data is weighted.

5

Banner grabbing is an enumeration technique used to glean information about computer systems on a network and the services running its open ports.

In a banner grabbing attack a connection is established with a system on a network and a bad request is sent. A vulnerable host will than respond with a banner message which may contain information that could be used to further compromise the system. An example of this is on a HTTP server the standard response would contain information such as the server software and version number. A known bug exploit for that particular software version could then be used to compromise the system.