# COMPSCI 351 S1 C : Tutorial 10

Due Date: Sunday 7 June 2020 at 11:59pm
10 marks in total = 1% of the finalgrade

1. Let r and s be relations with no indices, and assume that the relations are not sorted. Assuming infinite memory, what is the lowest-cost way (in terms of I/O operations) to compute $r \bowtie s$? What is the amount of memory required for this algorithm? [1 mark]

2. Consider the bank database of the figure below, where the primary keys are underlined, and the following SQL query:

```
select T.branchname from branch T, branch S
        where T.assets > S.assets and S.branch city= "Brooklyn"
```

*branch*(*branch_name*, *branch_city*, *assets*)
*customer* (*customer_name*, *customer_street*, *customer_city*)
*loan* (*loan_number*, *branch_name*, *amount*)
*borrower* (*customer_name*, *loan_number*)
*account* (*account_number*, *branch_name*, *balance*)
*depositor* (*customer_name*, *account_number*)

Write an efficient relational-algebra expression that is equivalent to this query. Justify your choice.

[2 marks]

3. Background. Most database systems provide a way to view the evaluation plan chosen to executea given query. Many databases support variations of a command "explain<query>", which displays the execution plan chosen for the specified query<query>. The exact syntax varies with different databases. MySQL uses **explain**<query> syntax (`https://dev.mysql.com/doc/refman/5.7/en/using-explain.html`. The estimated costs for the plan are also displayed along with the plan. It is worth noting that the costs are usually not in any externally meaningful unit, such as seconds or I/O operations, but rather in units of whatever cost model the optimizer uses. Use the database in Assignment 2 to try the "explain" feature.

   (a) Write a query with an equality condition on DEPARTMENT.Dname(which does not have an index), and view the plan chosen.

   (b) Create an index on the attribute DEPARTMENT.Dname, and view the plan chosen for the above query.

   (c) Create simple queries joining two relations, or three relations, and view the plans chosen.

   (d) Create queries joining two relations, or three relations, with selection conditions and view the plans chosen.

[2 marks]

4. Consider the relations $r_1(A, B, C)$, $r_2(C, D, E)$, and $r_3(E, F)$, with primary keys A,C,and E, respectively. Assume that$r_1$ has 1000 tuples, $r_2$ has 1500 tuples, and $r_3$ has 750 tuples. Estimate the size of $r_1 \bowtie r_2 \bowtie r_3$, and give an efficient **strategy** for computing the join. [2 marks]

5. Consider the relations $r_1(A, B, C)$, $r_2(C, D, E)$, and $r_3(E, F)$ without primary keys, except the entire schema. Let$V(C, r_1)$ be 900,$V(C, r2)$ be 1100,$V(E, r_2)$ be 50, and $V(E, r_3)$ be 100. Assume that $r_1$ has 1000 tuples, $r_2$ has 1500 tuples, and $r_3$ has 750 tuples. Estimate the size of $r_1 \bowtie r_2 \bowtie r_3$ and give an efficient **strategy** for computingthe join. [2 marks]

6. List the ACID properties. Explain the usefulness of each. [1 mark]