CS4320 Prelim Exam

October 23rd, 2018

(60 minutes working time)

Name: __________________________________ Cornell NETID: _________

I understand and will adhere to the Cornell Code of Academic Integrity.

Signature

Maximum number of points possible: 40. This exam counts for 25% of your overall grade. Questions vary in difficulty. Do not get stuck on one question.

In all problems, whenever you think a problem is underspecified, make assumptions and clearly state them.

Good luck!

Note – you have 60 minutes working time for this exam, NOT 2 hours as on some other prelims.

Prelim Part A) SQL Queries. (20 points)

Consider the database schema created by the following SQL commands:

```
CREATE TABLE Sailors (sid integer PRIMARY KEY,
sname varchar(20), rating integer, age real);

CREATE TABLE Boats (bid integer PRIMARY KEY,
bname varchar(20), color varchar(20));

CREATE TABLE Reserves(sid integer, bid integer,
day date, PRIMARY KEY (sid, bid, day),
FOREIGN KEY (sid) references Sailors (sid),
FOREIGN KEY (bid) references Boats (bid));
```

The database stores information on sailors (table Sailors), on boats (table Boats), and reservations for boats by sailors (table Reserves). The result of the solution queries for A.1 to A.3 must contain only one single column, no duplicate elimination is necessary (i.e., no need to use the SQL keyword DISTINCT).

A.1) Write an SQL query retrieving the names of sailors (attribute sname) who made no reservation for the boat with boat ID (bid) 100. (5 points)

```
SELECT S.sname FROM Sailors S WHERE NOT EXISTS (
SELECT * FROM Reserves R WHERE  S.sid = R.sid and R.bid=100);
```

A.2) Write an SQL query retrieving the names (attribute `sname`) of all sailors with the highest rating (assume that the database contains at least one sailor). (5 points)

```
SELECT S.sname FROM Sailors S WHERE
S.rating >= ALL(SELECT S2.rating FROM Sailors S2);
```

A.3) Write an SQL query retrieving all colors (attribute `color`) for which the database stores at least three boats of that color. (5 points)

```
SELECT B.color FROM Boats B
GROUP BY B.color HAVING COUNT(*) >= 3;
```

A.4) In an effort to make them more attractive, the management decides to paint all boats which received less than three reservations in red. Write an SQL update query that changes the color (attribute `color`) of all boats in the database with less than three associated reservations to red. (5 points)

```
UPDATE Boats B SET color = ‘red’ WHERE
3 > (SELECT COUNT(*) FROM Reserves R WHERE R.bid = B.bid);
```

Prelim Part B) External Sorting. (8 points)

We want to sort a relation that contains 8,000 tuples. Each tuple consumes 20 bytes. We store 8,000 bytes per page and use in total four buffer pages for sorting. Calculate the cost of sorting the relation via the external sort algorithm seen in class, following the same conventions for calculating cost as we did in class.

With 8,000 bytes per page and 20 bytes per tuple, we can fit 8,000/20 = 400 tuples per page. Hence, we need 20 pages to store all 8,000 tuples. Using four buffer pages, we obtain sorted runs of length four pages after pass 0. We need one output buffer and can therefore dedicate three pages as input buffers for the following passes (i.e., we can merge three sorted runs per pass). After pass 1, we have sorted runs of length 3 * 4 = 12 pages. We have sorted all 20 pages after the next run. Each pass requires reading and writing each page once, we do however not count the cost for writing out the final result. Hence, the total cost is 2 * 2 * 20 + 1 * 20 = 100 page I/Os.

Prelim Part C) Query Plans. (12 points)

Consider the database schema created by the following SQL commands:

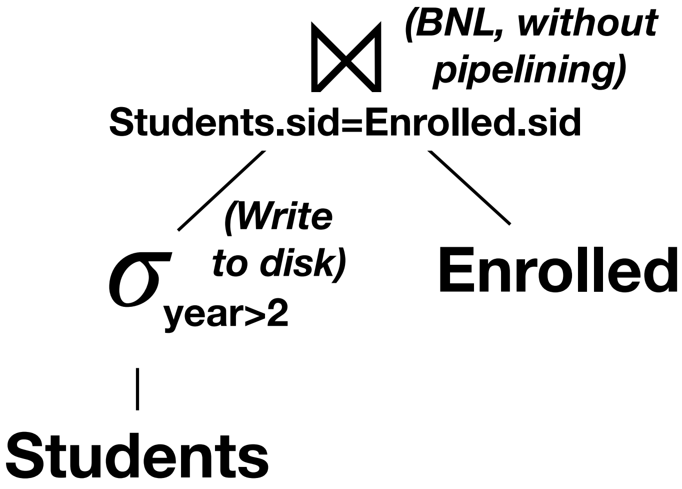
```
CREATE TABLE Students (sid integer PRIMARY KEY,
year integer);

CREATE TABLE Enrolled (sid integer, cid integer,
PRIMARY KEY (sid, cid),
FOREIGN KEY (sid) references Students (sid));
```

This database stores information on students (table `Students`) and which courses they are enrolled in (table `Enrolled`).

Calculate the cost of the query plan below under the following assumptions:

- Each integer field in a tuple consumes four bytes,
- Each disk page stores 8,000 bytes,
- Table `Students` has 10,000 rows,
- Table `Enrolled` has 20,000 rows,
- The BNL join uses all four buffer pages,
- The `year` field contains values between one and five (inclusive),
- For each year value, we store the same number of students,
- Each student is enrolled in two courses.



(You can use this page for the solution of Part C)

We calculate the size of base tables first. Both tables require 8 bytes per tuple. Having 8,000 bytes per page, we can store 1,000 tuples per page. Hence, the Students table consumes 10 pages while the Enrolled table consumes 20 pages.

Next, we calculate the size of intermediate results. We assume that students are uniformly distributed over years with values between 1 and 5 (inclusive). Hence, the selectivity of the predicate on the year is estimated as 3/5. We have 6 pages with students satisfying that predicate.

The plan exploits four buffer pages. For the BNL join, we need one output buffer and one input buffer for the inner relation. Hence, we can exploit two pages to store blocks from the outer relation. Hence, the join performs 6/2 = 3 iterations.

We calculate the cost of the plan. Reading all students to identify the one satisfying the predicate takes 10 page I/Os, writing out the result takes 6 I/Os. Reading the outer operand for the BNL join costs 6 page I/Os, iterating three times over the inner relation costs 3 * 20 = 60 page I/Os. As usual, we do not count the cost of writing out the final result to disk. Hence, the total cost is 10 + 6 + 6 + 3*20 = 82 page I/Os.

CS4320 Prelim Exam

This page will be used for grading your exam. Do not write anything on this page.

SECTION	**QUESTION**	**SCORE**	**SECTION TOTAL**
Part A	A.1 (Max: 5 points)		(Max: 20 points)
	A.2 (Max: 5 points)		
	A.3 (Max: 5 points)		
	A.4 (Max: 5 points)		
Part B	B.1 (Max: 8 points)		(Max: 8 points)
Part C	C.1 (Max: 12 points)		(Max: 12 points)
Total (Max: 40 points)			