CSE-6421: Assignment #2

1. (20 points) **SQL.** *The most books win!*

   Write the following queries against the St. Lawrence Bookseller's database (StL). The schema, and how to access the database under DB2, is on our course web-site.

   For these five queries, submit your SQL on-line, as per instructions on the web-site.

   Each query is four points.

   a. **sales**: ⟨year, month, sales⟩
   The sum of sales (total of sale from **Purchase**) per month.

   b. **multiple**: ⟨cust#, book#, genre#, language#, #copies⟩

   Customers who have bought the same book multiple times over *different* purchases, and the number of different times the person has bought the book (when) is seventy or more.
   Note that a customer might buy several copies of a books (qnty) in a purchase; however, if the person does not buy the same book again later, this is not counted. In each case, the book, its genre and language, and the number of copies bought overall by the customer is reported.

   c. **toppers**: ⟨language, city, sales⟩

   For each language, the city that has the maximum sales (total of sale from **Purchase**)
   If, for a language, there is a tie for maximum over several cities, all cities in the tie are listed.

   d. **sweep**: ⟨cust#, language, genre, publisher⟩

   Customers who have bought *all* the books that StL has available within a language-genre-publisher category that contains more than ten books.
   For each, list the language-genre-publisher category.

   e. **same**: ⟨exemplar, #members, #titles, genre⟩

   A group of customers who have each bought exactly the same books as each other within the same genre category, and the number of books bought in common is ten or greater.
   Identify each group by the smallest cust# in the group. List the number of members of the group (two or more), the number of book titles (#book) they have each commonly bought (ten or more), and the genre category the books are from.

2. (5 points) **Index Logic.** *Take the next index to your left.*

   You are told that the following indexes are available on the table **Employee**:

	key	**type**	**clustered?**
A.	name	tree	no
B.	name, city	tree	yes
C.	city, name	tree	yes
D.	age, salary	hash	no
E.	salary, age	hash	no

   You are suspicious that this information is not correct. Why? Identify three problems with what is reported.

3. (5 points) **Indexes and their uses.** *What's a* cluttered *index?!*

   Dr. Bas recently discovered how great *index intersection* is. He realized that the same idea can be applied to a *single*-index access path when the index is unclustered:

   - collect together the matching data entries using the index;
   - sort the data entries by their *rid*'s (record identifiers); and
   - then retrieve the data records in order.

   He claims that this is then just like using a clustered index; the only extra cost is sorting the data entries.

   When he ran an experiment, however, he found this performed no better than using the unclustered index directly! It still seemed to cost one I/O per data record retrieved.

   Why is Dr. Bas wrong about this being just like using a clustered index (except for the extra sorting step)? Devise a scenario—that is, what was it about the query and the tables in his experiment that led to bad performance—to support your argument.

4. (5 points) **Indexes and Cost.** *An index by any other name...*

```
SELECT * FROM Foo
    WHERE level = 3
      AND 20 < bar AND bar <= 30
      AND 100 < flavour AND flavour <= 250;
```

Table **Foo** contains 1,000,000 records. There are 20 records per page, on average, for 50,000 pages.

- Values of level: 1, 2, 3, 4, & 5.
- Values of bar: $1, 2, \ldots, 100$.
- Values of flavour: $1, 2, \ldots, 1000$.

For each value, assume there are an equal number of records with that value (uniformity assumption). So 200,000 records have level = 2, and 10,000 records have a flavour value between 120 and 130. Assume also that the values of one column are not correlated with the values of another column (independence assumption).

Indexes. All are of alternative 2. Each has a depth of 3, not including the data record / entry pages.

I. A clustered tree index on level, bar.

II. An unclustered tree index on bar, level.

III. An unclustered tree index on flavour.

---

a. (1 point) If we used no indexes, how many I/O's would it cost to evaluate the query?

---

b. (1 point) Are indexes **II** (bar, level) and **I** (level, bar) equivalent for this query? Why or why not?

---

c. (3 points) Estimate the I/O cost for each of the three cases using the index to evaluate the query.

Which is best?

5. (5 points) **Buffer Pool.** *Dirty laundry.*

   You have recently joined the firm VSDB, Inc., who specialize in very small databases. You are working in Dr. Dogfurry's group and are in charge of building a better buffer pool manager.

   Dr. Dogfurry points out that dirty pages (dirty_bit = *true*) cost more than clean pages (dirty_bit = *false*) to replace, because they need to be written back to the disk.

   a. (3 points) Briefly sketch out a design of a replacement strategy based on *clock* that favours replacing clean pages.

      Be careful not to favour picking clean pages as victims too much. Dirty pages should be picked occasionally, even when an unpinned clean page's frame is available. You do not want "hot" clean pages always being replaced while "cold" dirty pages stay.

   b. (2 points) Will this new replacement strategy necessarily be better than clock? Briefly, why or why not?