**CSI2132- 2016**

**Assignment 3**

**Answers**

**A. Relational Algebra**

Note that I used natural or equijoins. There are also other ordering of operations that are correct and/or more efficient, such as using projections earlier.

1.

$$\rho(\text{GirlScience}, \sigma_{\text{camptitle}='\text{GirlScience}'}\ \text{Camp})$$

$$\rho(\text{GirlTech}, \sigma_{\text{camptitle}='\text{GirlTech}'}\ \text{Camp})$$

$$\rho(\text{Science}, \pi_{\text{cname}, \text{email}, \text{fee}}(\text{Camper} \bowtie \text{Signup} \bowtie (\text{GirlScience}))$$

$$\rho(\text{Tech}, \pi_{\text{cname}, \text{email}, \text{fee}}(\text{Camper} \bowtie \text{Signup} \bowtie \text{GirlTech})$$

$$\rho(\text{Result}, \text{Science} \cap \text{Tech})$$

2.

$$\rho(\text{older}, \pi_{\text{cname}}(\sigma_{\text{age} > 10}\ \text{Camper}))$$

$$\rho(\text{Makers}, \pi_{\text{empID}}(\sigma_{\text{camptitle} = \text{MakerTech}}\ \text{Camp} \bowtie \text{Signup} \bowtie \text{Older}))$$

$$\rho(\text{Result}, \pi_{\text{Name}, \text{Startdate}, \text{enddate}}(\text{Makers} \bowtie_{\text{empID}} \text{Camp})$$

3.

$$\rho\,(\text{SevenCamps},\ \pi_{\text{campID}}\,(\text{Signup} \bowtie_{\text{cname}} (\sigma_{\text{age}=7}\ \text{Camper})))$$

$$\rho\,(\text{SandyCamps},\ \pi_{\text{campID}}\,(\text{Camp} \bowtie (\sigma_{\text{name}=\text{Sandy} \wedge \text{Salary}=500}\ \text{Mentor})))$$

$$\rho\,(\text{Result},\ \pi_{\text{cname},\text{email},\text{tshirt}}\,(\text{Camper} \bowtie \text{Signup} \bowtie (\text{SevenCamps} - \text{SandyCamps})))$$

**B. Normal Forms**

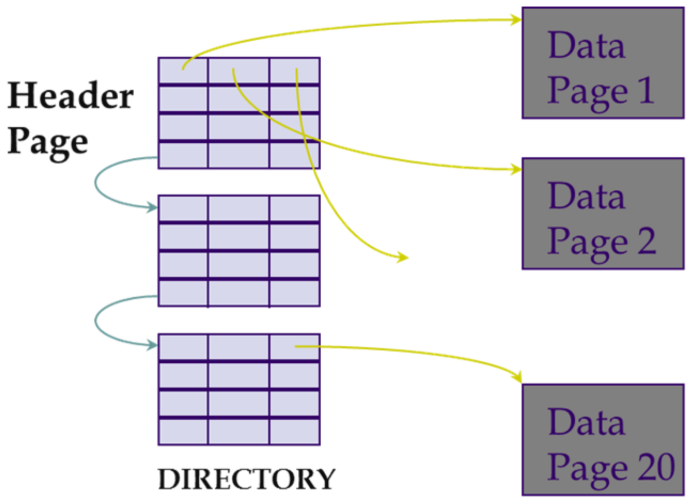
1. Refer to the slides and also Chapter 19 of the textbook. These three problems occur in cases where I have more than one entity together in one table. This leads to repeated values and so-called uncontrolled redundancies. Below are some examples.
   a. Insertion anomalies: If I want to insert a new book, and I do not have the price yet (or if I do not know the Distributor), then I may have to insert "dummy" values.
   b. Deletion anomalies: Deleting the first book will also remove information about Prentice Hall. That is, we will not know that such a Distributor exists.
   c. Update anomalies: If I change the name of say Addison W, then there is at least two rows. If I forget to update one of them, then this may lead to inconsistent and incorrect data in my tables.

2. Refer to the normal form rules for dependencies.

   a. We have B → C, C → DA, so B → ABCD and B is the key. This table is not in 3NF, since we have a transitive dependency. The highest normal form is 2NF (no partial dependencies).
   b. We have A → BC, and BC → D, so A → ABCD and A is the key. This table is not in 3NF, since we have a transitive dependency. The highest normal form is 2NF (no partial dependencies).
   c. We have AB → CD, C → A and D → B, so AB → ABCD and AB is the key. This table is not in BCNF, since we have cycles. The highest normal form in 3NF (no transitive or partial dependencies).

## C. Physical Database Design

1. You were required to some either i) the double-linked list or ii) the directory of pages implementations.
   For the directory implementation, the header page will include the addresses of all the data pages on disk. The Camper table will be stored over 20 disk pages (20,000/1,000 = 20).



   For the double-linked list implementation, we have a header pointing to 20 full pages and another pointing to the free disk pages. (Refer to the slides and textbook for the description.)

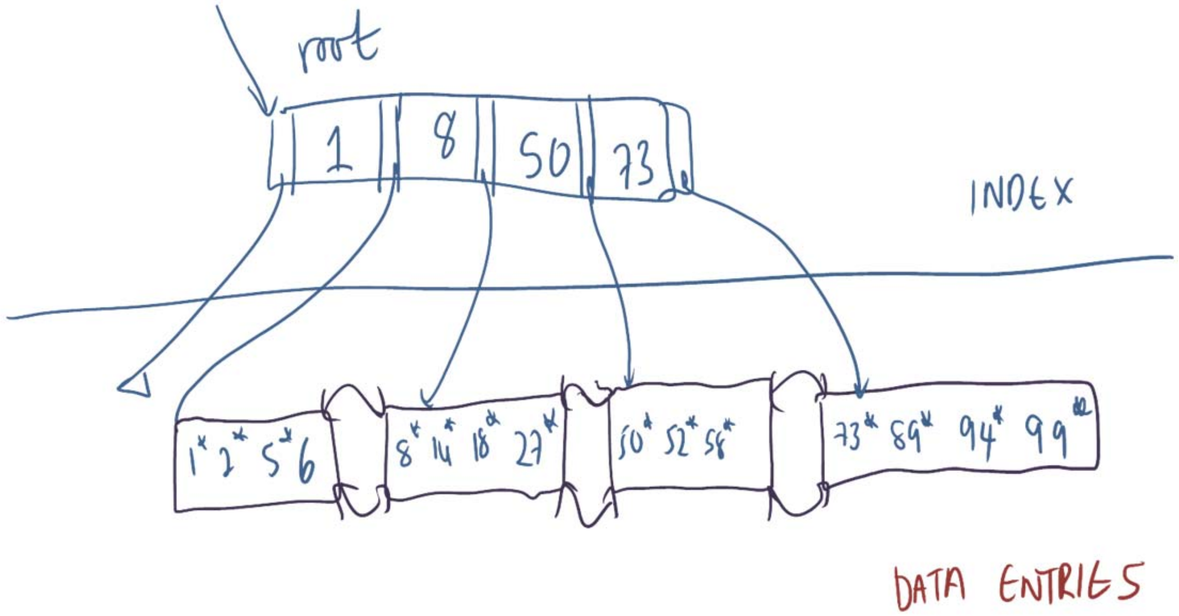
   Note: In general, we prefer the directory since it is more efficient.

2. In this query, we need to access the entire Camper file. This file contains data that span 20 pages on disk. Our buffer has the capacity to hold up to 10 blocks at a time. This implies that we cannot hold the entire Camper table in main memory at the same time, and that we will need some buffer replacement policy.
    a. We will follow a file scan, i.e. reading the entire file.
    b. We locate the start of the file on disk (the first page). This information is obtained from the system catalogue.
    c. The disk space manager will talk with the disk controller. The disk head will be moved to the correct track.
    d. Next, page X (X = page 1 to 20 of Camper) will be read when the sector moves underneath the head.
    e. page X is then read into the buffer pool.
        i. If there is space, the buffer frame is marked as occupied and we are done.
        ii. If not, then a buffer frame will be selected for replacement.
        iii. A buffer replacement policy, such as LRU, clock or MRU, is followed.
        iv. If a page needs to be replaces, we see if it is dirty.
            1. Dirty pages are written back to disk.
            2. Otherwise, the page simply overwrites the frame.
    f. This process is continued until the entire file has been read, and the query has been answered.

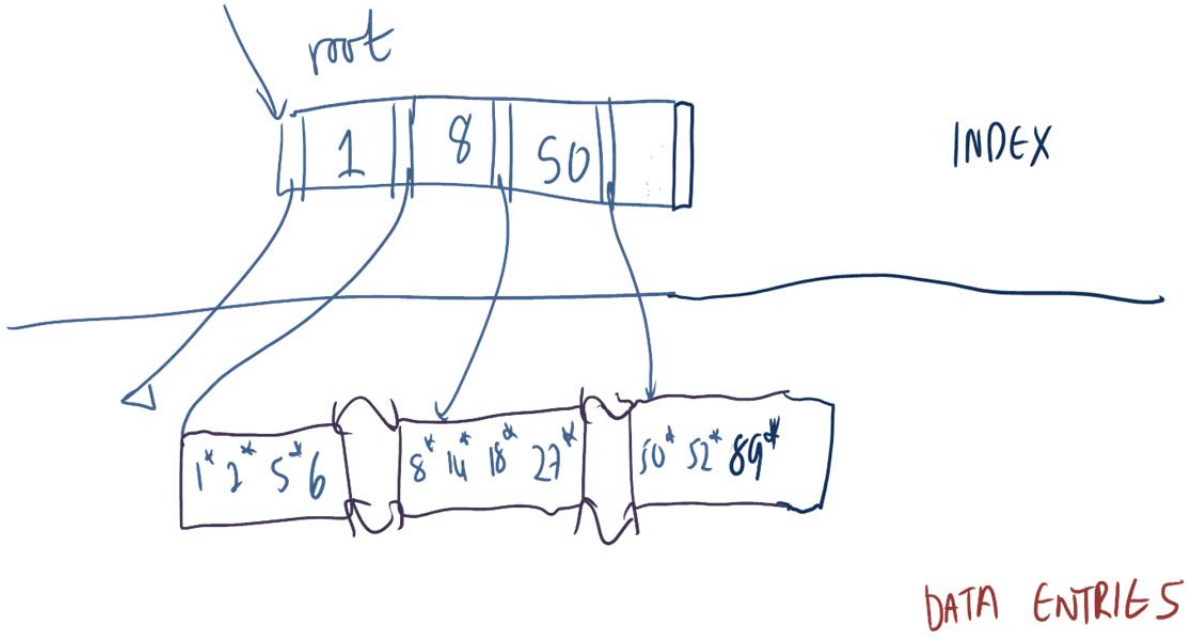
**D. Storage and Indexing**

Note: if we have an order d of 2, then each note has 4 (2*d) entries and we have 5 pointers.

1.



2.


root
1
8
50
INDEX
1* 2* 5* 6
8* 14* 18* 27*
50* 52* 89*
DATA ENTRIES