Question 4 (10 marks)

Consider a DBMS which has a range of possible buffer-management policies. You can specify the total number of buffers to be allocated in the buffer pool. For replacement, unpinned buffers are favoured over pinned buffers. You can specify that the buffer manager should use either least-recently-used (LRU) or most-recently-used (MRU) in determining which buffer to re-use. This DBMS defines recency relative to "the time of the last request or release operation on the buffer", *not* relative to the last time the buffer was accessed, because it is simpler to keep track of the time of request/release operations. The buffer manager also allows you to allocate multiple buffer pools of varying sizes and specify how tables should be associated to pools.

For each of the scenarios below, do the following:

i. Calculate the total numbers of *requests* on each buffer pool

ii. Calculate the total number of *hits* on each buffer pool

iii. Calculate the total number of disk *reads* on each buffer pool

iv. Show the state of the buffer pools after exactly 15 requests; show all pools and all slots. For each slot, show the table name and page number for the current contents of that slot

The following shows what your state should look like (this shows the state after 3 requests for scenario (a)):

```
Pool [0]
Buffers  [0] [1] [2] [3] [4]
Contents  P0  P1  P2  -   -
```

Scenarios:

a. One sequential scan of a single table *P* (with b_P=*20* pages) using a single buffer pool with LRU replacement strategy and 5 buffers. The scan behaves as follows:

```
for i in 0..19 { request page i from P;  process page i;
release page i }
```

b. Two sequential scans of a single table *R* (with b_R=*10* pages) using a single buffer pool with MRU replacement strategy and 5 buffers. The scans behave as follows:

```
for i in 0..9 { request page i from R;  process page i;  release
page i }
for i in 0..9 { request page i from R;  process page i;  release
page i }
```

c. Simple nested loop join on two tables *S* and *T* (with b_S=*5* and b_T=*10*) using a buffer pool with MRU replacement strategy and 10 buffers. The join behaves as follows.

```
for i in 0..4 {
    request page i of S
    for j in 0..9 {
        request page j of T
        process join on page i of S and page j of T
        release page j of T
    }
    release page i of S
}
```

In all scenarios, assume that the buffer pool initially starts empty and that empty slots are used first, before any replacement is considered. Assume also that pools, buffers and pages are indexed starting from 0.

Show enough working; you don't need to show the state after every request/release.

Instructions:

- Type your answer to this question into the file called `q4.txt`
- Submit via: **`give cs9315 sample_q4 q4.txt`**
  or via: Webcms3 > exams > Sample Exam > Submit Q4 > Make Submission

End of Question