CS 564 Problem Set #3

DELIVERABLES

Submit your answers using a **single pdf** file. Upload the file at Canvas (under PS3).

A: THE YELP DATABASE **[70pts]**

Suppose we are given a database with the following schema.

Users (UserID INTEGER, Name CHAR(30), Age INTEGER, ReviewCount INTEGER)

Businesses (BusinessID INTEGER, BName CHAR(30), City CHAR(20), State CHAR(2))

Checkins (BusinessID INTEGER, Weekdays INTEGER, Weekends INTEGER)

Reviews (ReviewID INTEGER, UserID INTEGER, BusinessID INTEGER, Stars REAL)

Reviews (UserID) is a foreign key referring to **Users** (UserID).
Reviews (BusinessID) is a foreign key referring to **Businesses** (BusinessID).
Checkins (BusinessID) is a foreign key referring to **Businesses** (BusinessID).

A page is 8 kB in size. The RDBMS buffer pool has **10,000** pages, all of which are available. Initially, the buffer pool is empty.

The relation instances have the following statistics. Assume there are no NULL values. Each integer or real is 8B, and each character is 1B (so as an example CHAR(20) is 20B). Additionally, the record id of each tuple is 8B.

Relation	Number of Pages	Number of Tuples
Users	75,684	10m
Businesses	41,504	5m
Checkins	19,532	5m
Reviews	488,282	100m

Answer the following questions. *Clearly explain how you obtained your answer for each.*

1. **[10pts]** Name 5 indexes (hash and/or clustered B+ tree) on Users that match the predicate in the following SQL query and explain why each index matches.

```
SELECT  *
FROM    Users
WHERE   NOT ((Name <> "John" AND NOT (Name = "Mary"))
        OR (Age <> 20 AND Age <= 50));
```

2. **[15pts]** Suppose we are given a clustered B+ tree index on Businesses (State, City) with a (constant) fan-out of 100 and fill factor 1. Also, suppose that the index follows the alternative of storing the data records directly in the leaf pages of the index. What is the best possible I/O cost for the following SQL? Exclude the cost of writing the output. Assume that the selectivity of the predicate `State = "WI"` is 2%.

```
SELECT  DISTINCT City
FROM    Businesses
WHERE   State = "WI";
```

3. **[15pts]** Suppose we are given a clustered B+ tree index each on Businesses (BusinessID) and Checkins (BusinessID), both with a (constant) fan-out of 100. Also, suppose that both indexes follow the alternative of storing the data records directly in the leaf pages of the index. Which join algorithm among the following has the lowest I/O cost for a natural join of Businesses and Checkins: Block Nested Loop Join, Index Nested Loop Join, Sort-Merge Join, or Hash Join? Show all of your calculations clearly.

4. **[15pts]** Suppose that there is no index on the Businesses relation. Consider the following SQL query.

```
SELECT    City, COUNT (BusinessID)
FROM      Businesses
GROUP BY  City;
```

   What is the maximum number of cities for which it is possible to implement hash-based aggregation by reading the relation only once? Assume that the fudge factor of the hash table is $f = 1.4$. Show all of your calculations clearly.

5. **[15pts]** Suppose that there are no indexes on any relation and no relation is sorted on any attribute. Propose a physical plan for the following SQL query that does not materialize any intermediate relation, and compute its I/O cost. Assume that the values of `Stars` are real numbers uniformly distributed between 0 and 5 (inclusive), and the values of `Age` are integers uniformly distributed between 10 and 99 (inclusive). Also assume independence of the predicates on `Stars` and `Age`. Show all of your calculations clearly.

```
SELECT  COUNT (UserID)
FROM    Users U, Reviews R
WHERE   U.UserID = R.UserID AND R.Stars < 1 AND U.Age = 18;
```

B: MORE QUESTIONS [30pts]

1. **[15pts]** Suppose we are joining two tables S and R with respective number of pages $4BN_S$ and $8BN_R$, wherein $4BN_S \gg 8BN_R$. The number of buffer pages is $4B + 1$ and the buffer pool is initially empty. We are also given that $2fN_R = 4B - 1$, where f is the hash table fudge factor.

   The distribution of the join attribute values in S and R are such that after the first hash partitioning phase, we get exactly $4B$ partitions of S, each of length N_S pages, but not all partitions of R are of the same length. Suppose R gets partitioned as follows: $2B$ partitions of length N_R pages, B partitions of length $2N_R$ pages, and B partitions of length $4N_R$ pages.

   What is the I/O cost of the regular hash join algorithm discussed in class? Exclude the cost of writing the output of the join. Assume perfect uniform splitting occurs during the recursive repartitioning. Show all of your calculations clearly.

   (Hint: The answer is of the following form: $xBN_S + yBN_R$, where $x \in \{12, 14, 16, 18\}$ and $y \in \{24, 28, 32, 36\}$.)

2. **[15pts]** The *mode* of a list of values is the most frequent value. There can be more than one mode for a list. For example, the list $\{5, 2, 2, 3, 6, 6, 2, 5, 5, 10\}$ has two modes, 5 and 2. Assume that no index is available. Suppose we want to compute the mode of attribute A of a relation R with N pages. What is the best algorithm you can come up to in terms of I/O cost? The analysis of the algorithm should consider the worst case cost.