CSE 562 Midterm *Solutions*

March 12, 2014

Question	Points Possible	Points Earned
A.1	15	
A.2	10	
B.1–8	24	
B.9	10	
C.1	8	
C.2	8	
C.3	8	
C.4	12	
D.1	5	
Total	100	

The Birdwatcher Schema.

```
CREATE TABLE Birds (
  bid integer,
  species string,
  legband char(4),
  PRIMARY KEY (bid),
  UNIQUE (legband)
);

CREATE TABLE Observers (
  oid integer,
  name string,
  PRIMARY KEY (oid)
);

CREATE TABLE Sightings (
  oid integer,
  bid integer,
  when date,
  latitude decimal,
  longitude decimal,
  PRIMARY KEY
    (bid, oid, when),
  FOREIGN KEY (oid)
    REFERENCES Observers,
  FOREIGN KEY (bid)
    REFERENCES Birds
);
```

Birds	bid	species	leg band
	1	Raven	MORB
	2	Raven	MKYB
	3	Blue Jay	MRRK

Observers	oid	name
	1	Alice
	2	Bob
	3	Carol

Sightings	oid	bid	when	lat	long
	3	2	01/03/14	43.17	-77.96
	2	1	01/03/14	42.59	-78.69
	1	1	01/03/14	42.95	-78.66
	1	3	01/01/14	42.68	-78.65
	1	1	01/01/14	43.15	-79.30
	3	3	01/02/14	43.88	-78.62

Relational Algebra Operator Reference

Selection	$\sigma_c(R)$	c : The selection condition
Projection	$\pi_{e_1,e_2,\ldots}(R)$	e_i : The column or expression to project
Cartesian Product	$R_1 \times R_2$	
Join	$R_1 \bowtie_c R_2$	c : the join condition
Aggregate	$\pi_{gb_1,gb_2,\ldots,\texttt{SUM}(e_1),\ldots}(R)$	gb_i : group by columns, e_i : expression
Set Difference	$R_1 - R_2$	
Union	$R_1 \cup R_2$	

Part A. SQL and Relational Algebra

(25 points)

1. (15 points) Write a SQL query that answers the following question (for any data): How many *distinct* **species** of **bird** have *ever* been seen by the **observer** who saw the *most* **birds** on **December 15, 2013**.

```
SELECT COUNT(DISTINCT species) FROM Sightings NATURAL JOIN Birds WHERE oid
= (SELECT oid FROM sightings WHERE d = date('2013-12-15') GROUP BY oid ORDER
BY COUNT(bid) DESC LIMIT 1)
```

or

```
SELECT COUNT(DISTINCT species) FROM Sightings NATURAL JOIN Birds WHERE oid
= (SELECT oid FROM sightings WHERE d = date('2013-12-15') GROUP BY oid HAVING
COUNT(bid) >= ALL (SELECT COUNT(bid) FROM sightings WHERE d = date('2013-12-15')
GROUP BY oid))
```

2. (10 points) Using *set*-relational algebra, write an expression that answers the following question (for any data): Which **observers** have *not* **sighted** at least one **bird** of every **species** recorded.

$\pi_{oid}\{[\pi_{oid}(\texttt{Observers}) \times \pi_{species}(\texttt{Birds})] - [\pi_{oid,\ species}(\texttt{Sightings} \bowtie \texttt{Birds})]\}$

or

$[\pi_{oid}(\texttt{Observers}) - (\pi_{oid,\ species}(\texttt{Sightings} \bowtie \texttt{Birds}) \div \pi_{species}(\texttt{Birds}))]$

Part B. Relational Operators

(34 points)

Consider the following *bag*-relational algebra query:

$$\pi_{R.A,T.E}(\sigma_{R.B<S.B}(R \times (S \bowtie_{S.C=T.C} (T_1 \cup \sigma_{T.D=3} T_2))))$$

(24 points) For each operator listed below, write down the size of the operator's working set and **estimate** the amount of work (in number of tuples) that the operator will need to perform. Be sure to note down the definition of any symbols you use. You may assume:

- The attribute D has 100 distinct values, uniformly distributed across tuples in T_2.
- The attribute C has 50 distinct values, uniformly distributed across tuples in S, T_1, and T_2
- The attribute B has 10 distinct values, uniformly distributed across tuples in R and S.

#	Operator	Working Set Size	Est. Cost
1	$\sigma_{T.D=3}(T_2)$ (no index)	1	$\|T_2\|$
2	$\sigma_{T.D=3}(T_2)$ (B+Tree index on $T_2.D$)	1	$\log_k T_2 + (\|T_2\|/100)$
3	$\bowtie_{S.C=T.C}$ (as Nested Loop Join)	2	$(\|T_2\|/100 + \|T_1\|) \times \|S\|$
4	$\bowtie_{S.C=T.C}$ (as Block Nested Loop Join)	$\|B\| + 1$ (or $2\times\|B\|$)	$(\|T_2\|/100 + \|T_1\|) \times \|S\|$
5	$\bowtie_{S.C=T.C}$ (as Hybrid Hash Join)	$\|S\|$ (or $\|T_1 \cup T_2\|$)	$(\|T_2\|/100 + \|T_1\|) + \|S\|$
6	$\bowtie_{S.C=T.C}$ (as Index Nested Loop Join)	2 + Index	Index Lookup + $(\|S\|)/50 \times (\|T_2/100 + \|T_1\|)$
7	$\times$	2	$\|R\| \times (\|S\|/50) \times (\|T_1\| + (\|T_2\|/100))$
8	$\sigma_{R.B<S.B}$	1	$\|R\| \times (\|S\|/50) \times (\|T_1\| + (\|T_2\|/100))$

Symbol Definitions

$|B|$ - size of block
k - fanout of B+ Tree

9. (10 points) Using the relational equivalencies for Selection, Projection, and Cartesian Products discussed in class, prove that (A_1 and A_2 are sets of attributes, and C is a boolean condition). **Be sure to state any assumptions or conditions under which your proof and/or the equivalence holds.**

$$\pi_{A_1 \cup A_2}(R \bowtie_C S) \equiv (\pi_{A_1}(R) \bowtie_C \pi_{A_2}(S))$$

$$\begin{aligned}
\text{L.H.S} \quad & & \pi_{A_1 \cup A_2}(R \bowtie_C S) \quad (1)\\
\text{Deconstruct join} \quad & \equiv & \pi_{A_1 \cup A_2}\sigma_C(R \times S) \quad (2)\\
\text{Assuming } C \text{ uses a subset of } A_1 \cup A_2 \quad & \equiv & \sigma_C\pi_{A_1 \cup A_2 \cup C}(R \times S) \quad (3)\\
\text{Assuming } A_1 \text{ is local to R and } A_2 \text{ is local to S} \quad & \equiv & \sigma_C(\pi_{A_1}R \times \pi_{A_2}S) \quad (4)\\
\text{Construct join} \quad & \equiv & (\pi_{A_1}(R) \bowtie_C \pi_{A_2}(S)) \quad (5)\\
& \equiv & R.H.S \quad (6)
\end{aligned}$$

■

Part C. Physical Layout and Indexing

(36 points)

The bird watcher database designers are considering which indexes to build over their database and have asked you to consult. After a lengthy information-gathering process, you are able to determine that most of the queries that they will be issuing against their database will follow one of the following query templates (`?` denotes a parameter to the query template).

(A) The number of species that each observer has observed.

```
SELECT COUNT(DISTINCT b.species) FROM Birds b NATURAL JOIN Sightings s
  NATURAL JOIN Observers o WHERE o.name = ?;
```

(B) The number of days where each bird was sighted in a given geographic region.

```
SELECT b.bid, COUNT(DISTINCT s.when)
  FROM Birds b NATURAL JOIN Sightings s
  WHERE (s.latitude BETWEEN ? AND ?) AND (s.longitude BETWEEN ? AND ?)
  GROUP BY b.bid;
```

(C) The northernmost latitude at which a given species has been sighted.

```
SELECT s.latitude FROM Birds b NATURAL JOIN Sightings s
  WHERE b.species = ? ORDER BY s.latitude DESC LIMIT 1
```

(D) Register a new observer.

```
INSERT INTO Observers(name) VALUES (?);
```

(E) Register a new sighting, given an observer's name and a bird's leg-band identifier.

```
INSERT INTO Sightings(oid, bid, when, latitude, longitude)
  SELECT o.oid, b.bid, ?, ?, ? FROM Observers o, Birds b
  WHERE o.name = ? AND B.legband = ?;
```

However, you are unable to determine how frequently each of these queries will be issued. So as to be properly prepared, you have come up with several potential scenarios, listed below.

For each of the following scenarios state (i) the physical layout (heap, sorted, or clustered index) that you would select for each table, and (ii) which index(es) you would create (Hash, or B+Tree). When your answer includes a sort or index, indicate on which attributes should be used to sort or index. Justify your answer **in no more than *two* sentences**.

1. (8 points) Queries **(A)** and **(E)** dominate the workload.

   Birds can be sorted so as to make the COUNT(DISTINCT b.species) efficient.

   Observers can be sorted or clustered.

   Signtings needs no index and hence are stored in a heap.

   We can have Hash index on o.name and B.legband in order to make the queries of the form o.name = ? and B.legband = ? fast.

2. (8 points) Queries **(D)** and **(E)** dominate the workload.

   No index and heap on sightings. Having index will make inserting more costlier.

   Birds can have a clustered index.

   No index and heap for observers or the second acceptable answer is a we can have Hash index on o.name and

   Hash index on B.legband in order to make the queries of the form o.name = ? and B.legband = ? fast.

3. (8 points) Queries **(B)**, **(C)** and **(D)** dominate the workload.

   No index and heap on observers. Having index will make inserts slower.

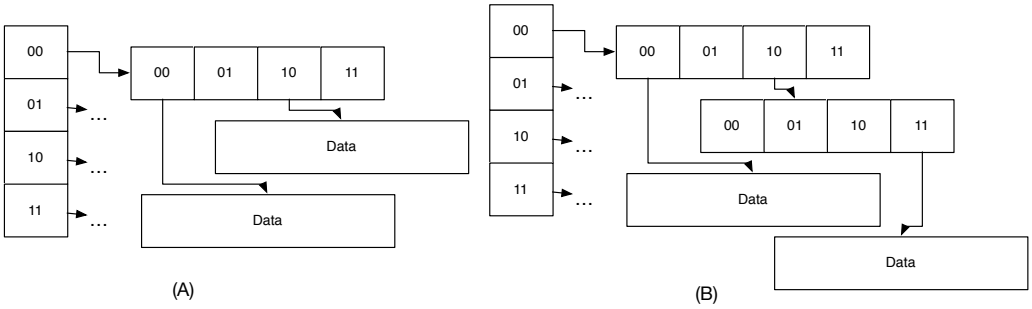
   Birds clustered so that queries like b.species = ? are fast.

   Sightings sorted on latitude and/or longitude.

   B+ tree (preferred) or any index on s.latitude. Any index on b.species.

   A 2 attribute index first on s.latitude and then on s.longitude.

4. (12 points) Consider the following disk-based datastructure discussed in class.



Pages are either *index pages* or *data pages*. A data page stores one or more records. An index page stores a list of pointers to additional pages (which may themselves be either index or data pages). To find a record, a hash function $h_i(record)$ is used: h_1 finds the correct pointer to follow on the initial index page, h_2 finds the correct pointer to follow on the subsequent index page, and so forth, until a data page is reached.

When a data page becomes full, it splits (e.g., (A) $\rightarrow$ (B) in the diagram): The data page is replaced with an index page, and each record being stored is partitioned according to the appropriate hashing function.

Assume that a k-bit hash function is being used (the example diagram uses a 2-bit hash function), that the datastructure contains N records, and that each data page stores exactly 4 records. What is the expected IO cost (in terms of absolute number of read and write IOs) of each of the following operations:

(a) Read all records with attribute $A = 20$

$\log_{2^k} N$

(b) Read all records with attribute $A > 20$

N

(c) Insert a new record.

$\log_{2^k}(N) + 1 + 2^k$

Part D. Class Participation

(5 points)

1. (5 points) What is Prof. Kennedy's favorite response to a question?

   It Depends

Grading Scheme

Part A (Grader: Niccolo): #1 No points deducted for not getting optimized solution. General scheme was 7 points for getting the inner query right and 8 points for getting the outer query right. Partial points were awarded on the basis of student's understanding and the correctness of operator usages like MAX or count, etc.
#2 Partial points awarded on the basis of student's understanding and the correctness of operator usages.
Part B 1-8 (Grader: Vishrawas) : For Working Set Size: #1-#3 1.5 points if the values match; 0 otherwise. #4- 0.5 marks deducted if the answer is 2 but no mention of B; 0 for all others. No marks deducted for missing constants like 1 or 2 in general for all questions. #5 - marks awarded to students who have written 1 hashtable or 1 relation or equivalent to thereof. #6 -0.5 if Index is not mentioned but 2 is written; 0 for all others #7 possible answers 3 and 2; 0 to all others #8 possible answers 2 and 1; 0 to all others. Discretion has been applied as when required.

For Estimated Cost: #1 -1.5 for anything incorrect #2 -0.5 for missing one component #3 onwards - points awarded based on whether student has understood the question and the concept. In general 0.5 points where deducted if one of the component was missing. For hash if + is replaced by $\times$ full points have been deducted for failing to understand the difference. Similarly for Index. In general if there is a mistake made at the start and there has been a cascading of the error, points has been deducted only once. For instance if cost estimate of #7 is wrong but in permissible range of tolerance but #8 is same #7 no points have been deducted for #8.
Part B 9 (Grader: Oliver) : We were looking for 2 specific assumptions: (1) That C relies exclusively on a subset of $A_1 \cup A_2$ (2) That $A_{1,2}$ were attributes that belonged to R and S respectively. Assumption 1 was worth 4 points, Assumption 2 was worth 3 points. A structurally correct proof was worth the final 3 points. Though it was not strictly a proof of *equivalence*, those who noted that the right expression could be safely transformed into the left expression were given the benefit of the doubt on the assumptions. A second common error involved assuming that A_1 and A_2 were disjoint sets. This is not strictly

necessary, but no penalty was applied for this error.

Part C 1-3 (Grader: Shounak) : The following questions ask for two parts namely the physical layout that you would select for EACH table and which index(es) you would create. Each of this part is worth 4 points each. You get 2 points for the right answer and 2 points for a justification to support your answer. For question #2 using a B+ tree instead of a Hash index will fetch a 2 point penalty as it is not the most optimal solution. For question #3 1 point is reserved for the two attribute index. The rest of the answer will give you a maximum of 7 points following the above grading scheme.

Part C 4 (Grader: Oliver) : 4 points per subquestion for a fully correct answer, or for an answer using $\log_k$ instead of $\log_{2^k}$. 2 points for at least recognizing the logarithmic order for part (a), linear order for part (b), and either of the sub terms of part (c).

Part D (Grader: Oliver) : Binary grading. Credit was given for answers that convincingly demonstrated class attendance.