

CS387H Problem Set #3

Due Thursday, October 30th

[1] Consider the following three relations, each having 9 attributes listed in increasing order of selectivity (i.e., the most selective attribute is listed first):

```
relation A { type-a1 A1; // type-a1 is the data type for attribute A1
            ...
            type-an A9;
            };
relation B { type-b1 B1;
            ...
            type-bn B9;
            };
```

For the following questions, assume the MGL graph of Figure 1. Also, assume that the first five

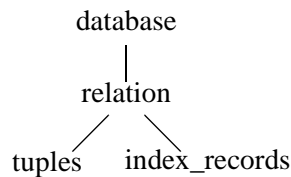


Figure 1: MGL Dag

attributes of relations A and B are indexed. Using the *Degree 2 Record Locking Protocol* presented in class, what locks would be taken for each statement? And how many locks would this be? (Assume n is the number of tuples that fully qualify and let m be the number of tuples that are qualified only by index records). (Recall that the Degree 2 Record Locking protocol locks tuples/index_records in read mode if the tuple/record is read; lock is in write mode if the tuple/record is to be written; also, only *fully-qualified* tuples are locked).

- (a)

```
select *
from A
where A2= 2 and A4 = 4
```
- (b)

```
select *
from A
where A8 = 8
```
- (c)

```
update A
set A1 = a1, A2 = a2
where A8 = 9 and A5 = 5
```
- (d)

```
delete A
where A4 = 4
```
- (e)

```
select *
from A,B
where A3=3 and B4=4 and A2=B2
```

[2] Resolve problem [1] using the *Degree 3 Record Locking Protocol*. (Recall that the Degree 3 Record Locking Protocol differs from the Degree 2 protocol in that all *index-record* qualified tuples are locked, not just “fully-qualified” tuples).

Solutions to PSN#3

[1a] The following $3 + n$ locks would be taken:

```
IR(database), IR(relation A), R(A2=2),  
foreach tuple t that satisfies (A2=2) and (A4=4): R(t)
```

[1b] The following 2 locks would be taken (in the following order):

```
IR(database), R(relation A)
```

Note that the indices don't help in restricting the read set of A; hence a read lock on relation A is needed.

[1c] The following $5 + 3n$ locks would be taken (in the following order):

```
IW(database), IW(relation A), w(A1=a1), W(A2=a2), R(A5=5),  
foreach tuple t that satisfies (A8=9) and (A5=5):  
    W(t), W(A1=t[A1]), W(A2=t[A2])
```

Note that the index records (A1=a1) and (A2=a2) are write-locked because their index records are updated. The index lock (A5=5) is locked in read mode because the record is read, not updated. Each qualified tuple is write-locked for updates.

[1d] The following $3 + n$ locks would be taken:

```
IW(database), IW(relation A), W(A4=4)  
foreach tuple t that satisfies (A4=4):  
    W(t), W(A1=t[A1]), W(A2=t[A2]), W(A3=t[A3]), W(A5=t[A5])
```

Note that each tuple to be deleted is write-locked, along with every index record that references that tuple is write-locked because those index records are to be updated.

[1e] The following $5 + n$ locks are taken:

```
IR(database), IR(relation A), R(A3=3),  
foreach tuple t that satisfies (A3=3): R(t)  
IR(relation B), R(B4=4),  
foreach tuple r that satisfies (B4=4): R(r)
```

[2a] The following $3 + m$ locks would be taken:

```
IR(database), IR(relation A), R(A2=2),  
foreach tuple t that satisfies (A2=2): R(t)
```

[2b] Same as [1b].

[2c] The following $5 + m + 2n$ locks would be taken (in the following order):

```
IW(database), IW(relation A), W(A1=a1), W(A2=a2), R(A5=5),  
foreach tuple t that satisfies (A5=5): W(t), W(A1=t[A1]), W(A2=t[A2])
```

Note that the index records (A1=a1) and (A2=a2) are write-locked because their index records are updated. The index lock (A5=5) is locked in read mode because the record is read, not updated. Each index-record qualified tuple is write-locked for updates. m records are locked (because each record referenced by the index record is read-locked), but only $2n$ (not $2m$) index records are additionally locked because of tuple qualification.

[2d] The following $3 + m$ locks would be taken (note in this example $n = m$):

```
IW(database), IW(relation A), W(A4=4)
foreach tuple t that satisfies (A4=4):
    W(t), W(A1=t[A1]), W(A2=t[A2]), W(A3=t[A3]), W(A5=t[A5])
```

Note that each tuple to be deleted is write-locked, along with every index record that references that tuple is write-locked because those index records are to be updated.

[2e] The following $5 + m$ locks are taken (note in this example $n=m$):

```
IR(database), IR(relation A), R(A3=3),
foreach tuple t that satisfies (A3=3): R(t)
IR(relation B), R(B4=4),
foreach tuple r that satisfies (B4=4): R(r)
```