

CISC 7512X Midterm Exam

For the below questions, use the following schema definition.

```
customer(cid,fname,lname,ssn)
account(aid,cid,type,name)
transaction(tid,aid,tim,amnt)
```

It is a schema for a credit card company (a bank), where customers may have several credit cards (accounts), and can do purchases (positive amounts) and make payments (negative amounts).

Pick the best answer that fits the question. Not all of the answers may be correct. If none of the answers fit, write your own answer.

1. (5 points) Find balance for account 1.
 - (a) select sum(amnt) from transaction where aid=1;
 - (b) select tid,sum(amnt) from transaction group by tid
 - (c) select sum(amnt) from transaction where amnt > 0
 - (d) select max(amnt) from transaction where aid=1
 - (e) Other:
2. (5 points) Find the total borrowed amount for the entire bank.
 - (a) select sum(amnt) from transaction
 - (b) select sum(amnt) from transaction where amnt > 0
 - (c) with stats as (select sum(amnt) tot from transaction where amnt > 0) select sum(tot) from stats
 - (d) with stats as (select sum(amnt) tot from transaction group by aid) select sum(tot) from stats where tot > 0
 - (e) Other:
3. (5 points) Find all accounts with unpaid balance.
 - (a) select * from transaction where sum(amnt) <> 0
 - (b) select sum(amnt) from transaction group by tid having sum(amnt) <> 0
 - (c) select aid from transaction group by aid having sum(amnt) > 0
 - (d) select aid from account a inner join transaction b on a.cid=b.cid group by a.cid having sum(amnt) > 0
 - (e) Other:
4. (5 points) Find all transactions of more than \$10000.
 - (a) select tid from transaction where sum(amnt) > 10000
 - (b) select tid from transaction where amnt>0
 - (c) select max(tid) from transaction where amnt > 10000
 - (d) select sum(amnt) from transaction group by tid having sum(amnt) > 10000
 - (e) Other:

5. (5 points) Find first name and last name of every customer with type 2 account.
 - (a) select fname, lname from customer a inner join account b on a.cid=b.cid and b.type=2
 - (b) select fname, lname from account natural inner join transaction where type=2
 - (c) select fname, lname from customer natural inner join account where type=2 group by fname, lname having type=2
 - (d) select * from customer where type=2
 - (e) Other:

6. (5 points) Find potentially fraudulent accounts (type=13 accounts with over \$10000 balance)
 - (a) select * from account a inner join transaction where a.type=13 group by sum(amnt) having amnt > 10000
 - (b) select * from account where type=13 group by aid having sum(amnt) > 10000
 - (c) select * from account natural inner join transaction group by aid having sum(amnt) > 10000 and type=13
 - (d) select b.aid from account a inner join transaction b using(aid) where a.type=13 group by b.aid having sum(amnt) > 10000
 - (e) Other:

7. (5 points) What is the most appropriate index for customer.cid field?
 - (a) Btree Index
 - (b) Bitmap Index
 - (c) Clustered Index
 - (d) Bitmap Clustered Index
 - (e) Other:

8. (5 points) What is the most appropriate index for customer.lname field?
 - (a) Bitmap Index
 - (b) Btree Index
 - (c) Clustered Index
 - (d) Bitmap Clustered Index
 - (e) Other:

9. (5 points) What is the most appropriate index for account.type field?
 - (a) Bitmap Index
 - (b) Btree Index
 - (c) Clustered Index
 - (d) Bitmap Clustered Index
 - (e) Other:

10. (5 points) The below code:

```
create table dates as with recursive mydates (tdate) as (  
    select cast('19000101' as date) tdate  
    union all  
    select cast(tdate + interval '1 day' as date) tdate  
    from mydates where tdate <= '21000101'  
) select * from mydates;
```

- (a) Is invalid.
- (b) Will return all dates between 19000101 and 21000101
- (c) Will never return.
- (d) Will create a table with all dates between 19000101 and 21000101
- (e) Other:

11. (5 points) The Map operation in MapReduce does:

- (a) Applies a function/transformation on every input.
- (b) Collects identical keys for later aggregation.
- (c) Sorts the input before applying a filter.
- (d) Sorts the input before applying an operation per key and a list of values.
- (e) Other:

12. (5 points) The Reduce operation in MapReduce does:

- (a) Applies a function/transformation on every key.
- (b) Collects identical keys for later aggregation.
- (c) Redistributes/shuffles the records based on the key.
- (d) Sorts the input before applying an operation per key and a list of values.
- (e) Other:

13. (5 points) Find above average transactions:

- (a) `select tid,avg(amnt) from transaction group by tid having amnt > avg(amnt)`
- (b) `select * from (select tid,amnt,avg(amnt) over (partition by tid) avgamnt from transaction) a where amnt > avg(amnt) over (partition by tid)`
- (c) `select * from (select tid,amnt,avg(amnt) over () avgamnt from transaction where amnt>0) a where amnt > avgamnt`
- (d) `select * from (select tid,amnt,avg(amnt) over (partition by tid) avgamnt from transaction) a having amnt > avg(amnt) over (partition by tid)`
- (e) Other:

14. (5 points) Find all type 1 accounts with less than 1 transaction per month for entire 2022.
- select aid from transaction inner join account using (aid) where type=1 and to_char(tim,'YYYY') = '2022' group by aid, to_char(tim, 'YYYYMM') having count(*) < 1
 - select aid from transaction inner join account using (aid) where type=1 and to_char(tim, 'YYYY') = '2022' group by aid having count(*) < 12
 - select aid from transaction inner join account using (aid) where type=1 and to_char(tim, 'YYYY') = '2022' group by aid, to_char(tim, 'YYYYMM') having sum(1.0) < 1.0
 - select aid from transaction inner join account using (aid) where type=1 and to_char(tim, 'YYYY') = '2022' and count(*) < 1.0 group by aid, to_char(tim, 'YYYYMM')
 - Other:
15. (5 points) Find balances by SSN.
- select ssn,sum(amnt) from customer inner join account using (cid) inner join transaction using (aid) group by ssn
 - select ssn,sum(amnt) from customer c, account a, transaction j where c.aid=a.aid and a.aid=j.aid group by ssn
 - select a.ssn,b.amnt from customer a inner join (select aid,sum(amnt) amnt from transaction group by aid) b using aid
 - select aid,sum(amnt) from customer c, account a, transaction j where c.cid=a.cid and a.aid=j.aid group by aid
 - Other:
16. (5 points) For “account inner join transaction”, and no indexes, most modern databases will perform:
- indexed lookup join.
 - inner loop join.
 - merge join.
 - hash join.
 - Other:
17. (5 points) In general, on limited memory system, no indexes, and huge tables, what join type would perform best?
- hash join.
 - indexed lookup join.
 - merge join.
 - inner loop join.
 - Other:

18. (5 points) Data sharding:

- (a) Identical to partitioning except across multiple drives.
- (b) Process of storing a large dataset across multiple machines.
- (c) Creates a replica of the data on multiple computers.
- (d) Can often be replaced with a database connected to a NAS.
- (e) Other:

19. (5 points) Below query is identical to:

```
select a.*,b.val
from T1 a
left outer join T2 b
on a.key=b.key and a.val!=b.val
```

- (a) with TMP as (
select a.*,b.val from T1 a inner join T2 b on a.key=b.key
where a.val!=b.val)
select a.*,b.val from T1 a left outer join TMP b on a.key=b.key
- (b) select a.*,b.val
from T1 a inner join T2 b
on a.key=b.key and a.val!=b.val
- (c) with TMP as (
select a.*,b.val from T1 a left outer join T2 b on a.key=b.key
where a.val!=b.val)
select a.* from TMP where a.val!=b.val
- (d) All of the above queries are identical.
- (e) None of the queries are identical to the question.

20. (5 points) When you write:

```
select * from T1 a inner join T2 b on a.tim between b.start and b.end
```

what is the expected performance?

- (a) Hash join, approximately $O(N \log N)$, where N is the number of records in both T1 and T2.
- (b) Sort merge join, approximately $O(N)$, where N is the number of records in both T1 and T2.
- (c) Distributed hash join, approximately $O(N)$ to distribute data, and $O(N \log N)$ after distribution.
- (d) Inner loop join, approximately $O(N^2)$, where N is the number of records in both tables.
- (e) Other: