

## CISC 7510X Midterm Exam

For the below questions, use the following schema definition.

```
customer(cid, fname, lname, ssn)
account(aid, cid)
product(pid, symbol, type)
journal(jid, tid, tim, aid, pid, qty)
```

It is a schema for an asset custodian (e.g. a bank, brokerage firm, etc.). Customers have accounts. Accounts hold multiple products. Products can be anything, stocks, bonds, options, futures, swaps, other contracts, and cash in any currency. The journal records position changes for any product: negative qty removes product from account. The tid identifies the transaction that may have multiple journal entries: transactions do not create product quantity, therefore each product quantity within a transaction must sum to 0.

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 name of customer with SSN: 123-45-6789.
  - (a) `select fname,lname from customer where ssn='123-45-6789'`
  - (b) `select fname,lname from account where ssn='123-45-6789'`
  - (c) `select custid,ssn from customer where ssn='123-45-6789'`
  - (d) `select fname,lname from customer where custid='123-45-6789'`
  - (e) Other:
  
2. (5 points) Find pid (product id) of stock symbol "IBM".
  - (a) `select pid from product where type='OPTION' and symbol='IBM 191011C00140000'`
  - (b) `select pid from product where type='IBM' and symbol='ABC'`
  - (c) `select pid from product where type='STOCK' and symbol='IBM'`
  - (d) `select pid from account where type='STOCK' and symbol='IBM'`
  - (e) Other:
  
3. (5 points) Find pid (product id) of USD currency.
  - (a) `select pid from product where type='CURRENCY' and symbol='USD'`
  - (b) `select type, symbol from product where type='CURRENCY' and symbol='USD'`
  - (c) `select pid from product where type='BOND' and symbol='USD'`
  - (d) `select aid from account where type='STOCK' and symbol='USD'`
  - (e) Other:
  
4. (5 points) Find all of John Doe's (SSN=123-45-6789) account numbers.
  - (a) `select b.aid from customer natural inner join account where a.ssn='123-45-6789'`

- (b) `select b.aid from customer a inner join account b where a.cid=b.cid and a.ssn='123-45-6789'`
- (c) `select b.aid from account a inner join journal b on using (cid) where ssn='123-45-6789'`
- (d) `select b.aid from customer a inner join account b on a.cid=b.cid where a.ssn='123-45-6789'`
- (e) Other:
5. (5 points) How much money (in USD currency) does John Doe (SSN=123-45-6789) have?
- (a) `select sum(qty) from customer inner join account inner join journal inner join product where ssn='123-45-6789' and type='CURRENCY' and symbol='USD'`
- (b) `select sum(qty) from customer natural inner join account natural inner join journal natural inner join product where ssn='123-45-6789' and type='CURRENCY' and symbol='USD'`
- (c) `select sum(qty) from customer cross join account cross join journal cross join product where ssn='123-45-6789' and type='CURRENCY' and symbol='USD'`
- (d) `select sum(qty) from customer where ssn='123-45-6789' natural inner join account natural inner join journal natural inner join product where type='CURRENCY' and symbol='USD'`
- (e) Other:
6. (5 points) Find all products (and quantities) that John Doe (SSN=123-45-6789) has.
- (a) `select cid,pid,type,symbol,sum(qty) from customer natural inner join account natural inner join journal natural inner join product where ssn='123-45-6789' having sum(qty) != 0`
- (b) `select cid,pid,type,symbol,sum(qty) from customer natural inner join account natural inner join journal natural inner join product where ssn='123-45-6789' group by cid,pid,type,symbol`
- (c) `select cid,pid,type,symbol,sum(qty) from customer inner join account inner join journal inner join product where ssn='123-45-6789' group by cid,pid,type,symbol having sum(qty) != 0`
- (d) `select cid,pid,type,symbol,sum(qty) from customer natural inner join account natural inner join journal natural inner join product where ssn='123-45-6789' group by cid,pid,type,symbol having sum(qty) != 0`
- (e) Other:
7. (5 points) Find all customers who own stock IBM.
- (a) `select cid,pid,type,symbol,sum(qty) from customer natural inner join account natural inner join journal natural inner join product where type='STOCK' and symbol='IBM' having sum(qty) != 0`

- (b) `select cid,pid,type,symbol,sum(qty) from customer natural inner join account natural inner join journal natural inner join product where type='STOCK' and symbol='IBM' group by cid,pid,type,symbol`
- (c) `select cid,pid,type,symbol,sum(qty) from customer inner join account inner join journal inner join product where type='STOCK' and symbol='IBM' group by cid,pid,type,symbol having sum(qty) != 0`
- (d) `select cid,pid,type,symbol,sum(qty) from customer natural inner join account natural inner join journal natural inner join product where type='STOCK' and symbol='IBM' group by cid,pid,type,symbol having sum(qty) != 0`
- (e) Other:

8. (5 points) Find customers who have no accounts.

- (a) `select a.* from customer a left outer join account b on a.id=b.aid where aid is null`
- (b) `select a.* from customer a left outer join account using (aid) where aid is null`
- (c) `select a.* from customer a natural left outer join account where aid is null`
- (d) `select a.* from customer a natural inner join account where aid is null`
- (e) Other:

9. (5 points) Find accounts that did not have any transactions in the last 6 months.

- (a) `select a.aid from account a left outer join journal b on a.aid=b.aid where b.tim>=(now()-interval '6 months') and b.id is null`
- (b) `select a.aid from account a natural left outer join journal b where b.tim>=(now()-interval '6 months') and b.aid is null`
- (c) `select a.aid from account a left outer join journal b using(aid) and b.tim>=(now()-interval '6 months') where b.aid is not null`
- (d) `select a.aid from account a left outer join journal b on a.aid=b.aid and b.tim>=(now()-interval '6 months') where b.aid is null`
- (e) Other:

10. (5 points) Find all customers who had a short (negative) position in stock symbol IBM on January 1st, 2019.

- (a) `select cid,fname,lname,type,symbol, sum(qty) from customer natural inner join account natural inner join journal natural inner join product where type='STOCK' and symbol='IBM' and tim<cast('2019-01-01' as date) group by cid,fname,lname,type,symbol having sum(qty) < 0`

- (b) `select cid,fname,lname,type,symbol, sum(qty) from customer natural inner join account natural inner join journal natural inner join product where type='STOCK' and symbol='IBM' and tim<cast('2019-01-01' as date) having sum(qty) < 0`
- (c) `select cid,fname,lname,type,symbol, sum(qty) from customer natural inner join account natural inner join journal natural inner join product where type='STOCK' and symbol='IBM' and tim<cast('2019-01-01' as date) and sum(qty) < 0 group by cid,fname,lname,type,symbol`
- (d) `select cid,fname,lname,type,symbol, sum(qty) from customer a inner join account b on a.cid=b.cid inner join journal c on b.aid=c.aid inner join product d on c.pid=d.pid and d.type='STOCK' and d.symbol='IBM' and tim<cast('2019-01-01' as date) where sum(qty) < 0 group by cid,fname,lname,type,symbol`
- (e) Other:
11. (5 points) IBM issued a dividend of \$1.62 USD per share. The company sent us a check. As part of tid=1234567 (transaction identifier), we need to add the respective amounts to each customer who has a long (positive) position in IBM.
- (a) `update journal set qty+=1.62 where product.type='CURRENCY' and product.symbol='USD' from product a where a.type='STOCK' and a.symbol='IBM' group by aid having sum(qty) > 0`
- (b) `insert into journal select nextval('journal_jid'), 1234567 as tid, now() as tim, aid, (select pid from product where type='CURRENCY' and symbol='USD'), sum(qty)*1.62 from journal natural inner join product where type='STOCK' and symbol='IBM' group by aid having sum(qty) > 0`
- (c) `insert into journal select nextval('journal_jid'), 1234567 as tid, now() as tim, aid, (select pid from product), sum(qty)*1.62 from journal inner join product where type='STOCK' and symbol='IBM' group by aid having sum(qty) > 0`
- (d) `insert into journal select nextval('journal_jid'), 1234567 as tid, now() as tim, aid, (select pid from product where type='CURRENCY' and symbol='USD'), sum(qty)*1.62 from journal a inner join product b on a.pid=b.pid where b.type='STOCK' and b.symbol='IBM' and sum(a.qty) > 0 group by a.aid`
- (e) Other:
12. (5 points) Identify broken or incomplete transactions (those that do not sum to 0 for a particular product).
- (a) `select tid,pid,sum(qty) from journal where sum(qty) != 0 group by tid,pid`
- (b) `select tid,pid,sum(qty) from journal group by pid,qty having sum(qty) != 0`
- (c) `select tid,pid,sum(qty) from account natural inner join journal group by tid,pid having sum(qty) = 0`

(d) `select tid,pid,sum(qty) from journal group by tid,pid having sum(qty) != 0`

(e) Other:

13. (5 points) What percentage of accounts have a short (negative) position in symbol SPY?

(a) `with tmp as (select aid,sum( qty ) spypos from journal natural inner join product where type='STOCK' and symbol='SPY' group by aid) select 100.0*sum(case when spypos < 0 then 1.0 else 0.0 end)/sum(1.0) from tmp`

(b) `with tmp as (select aid,sum(case when type='STOCK' and symbol='SPY' then qty else 0 end) spypos from journal natural inner join product group by aid) select 100.0*sum(case when spypos < 0 then 1.0 else 0.0 end)/sum(1.0) from tmp`

(c) `with tmp as (select aid,sum(case when type='STOCK' and symbol='SPY' then qty else 0 end) spypos from journal natural inner join product group by aid) select 100.0*percentage(spypos < 0) from tmp`

(d) `select percentile( sum(qty) < 0 ) from journal where type='STOCK' and symbol='SPY' group by aid having sum(qty) < 0`

(e) Other:

14. (5 points) Find customer with the most IBM shares (across all their accounts).

(a) `with tots as (select cid, sum(qty) tot from account natural inner join journal natural inner join product where type='STOCK' and symbol='IBM' group by cid having sum(qty) != 0),  
rnk as (select cid, tot, dense_rank() over (order by tot desc)  
r from tots)  
select * from rnk where r=1`

(b) `with tots as (select cid, rank(sum(qty)) r from account natural inner join journal natural inner join product where type='STOCK' and symbol='IBM' having sum(qty) != 0 group by cid))  
select * from tots where r=1`

(c) `with tots as (select cid, sum(qty) tot from account inner join journal inner join product where symbol='IBM' and type='STOCK' having sum(qty) != 0 group by cid),  
rnk as (select cid, tot, rank(tot) over () r from tots)  
select * from rnk where r=1`

(d) `with tots as (select cid, sum(qty) tot from customer natural inner join journal natural inner join product where type='STOCK' and symbol='IBM' having sum(qty) != 0 group by cid),  
rnk as (select cid, tot, rank(tot) over (order by tot desc)  
r from tots)  
select * from rnk where r=1`

(e) Other:

15. (5 points) Find accounts that have both IBM & MSFT.
- select aid from journal inner join product where type='STOCK' and symbol in ('IBM','MSFT') and sum(case when stock='MSFT' then 1.0 else 0.0 end)>0 and sum(case when symbol='IBM' then 1.0 else 0.0 end)>0
  - select aid from journal natural inner join product where symbol in ('IBM','MSFT') and type='STOCK' and sum(case when symbol='MSFT' then 1.0 else 0.0 end)>0 and sum(case when symbol='IBM' then 1.0 else 0.0 end)>0 group by aid
  - select aid from journal inner join product where symbol in ('IBM','MSFT') and sum(case when symbol='MSFT' then 1.0 else 0.0 end)>0 and sum(case when symbol='IBM' then 1.0 else 0.0 end)>0 group by aid
  - select aid from journal natural inner join product where type='STOCK' and symbol in ('IBM','MSFT') group by aid having sum(case when symbol='MSFT' then 1.0 else 0.0 end)>0 and sum(case when symbol='IBM' then 1.0 else 0.0 end)>0
  - Other:
16. (5 points) What the average number of distinct products per account?
- with tmp as (select aid,count(distinct pid) dcnt from journal group by aid having sum(qty)!=0) select avg(dcnt) from tmp
  - with tmp as (select aid,count(\*) cnt from journal group by aid having sum(qty)!=0) select avg(cnt) from tmp
  - with tmp as (select aid,count(pid) dcnt from journal having sum(qty)!=0) select avg(dcnt) from tmp
  - with tmp as (select aid,count(\*) dcnt from journal where sum(qty)!=0 group by aid) select avg(dcnt) from tmp
  - Other:
17. (5 points) Create a table of currency journal transfers (transactions with 2 or more distinct currencies).
- create table currency\_journal as with tids as (select distinct tid from journal natural inner join product where type='CURRENCY' and count(distinct symbol) > 1 group by tid ) select \* from journal natural inner join tids
  - create table currency\_journal as with tids as (select distinct tid from journal natural inner join product where symbol='USD' and type='CURRENCY' group by tid having count(distinct symbol)>1 ) select \* from journal natural inner join tids
  - create table currency\_journal as with tids as (select distinct tid from journal natural inner join product where type='CURRENCY' group by tid having count(distinct symbol)>1 ) select \* from journal natural inner join tids

(d) create table currency\_journal as select \* from journal where 1=0; insert into currency\_journal with tids as (select distinct tid from journal natural inner join product where type='CURRENCY' group by tid having count(distinct symbol)>1 ) select \* from journal a inner join tids b on a.aid=b.aid

(e) Other:

18. (5 points) Using table from previous question, create a table of unusual currency journal transfers. We define unusual as absolute qty being above 3 standard deviations of previous 60 currency transactions.

(a) create table unusual\_currency as with tmp as (select a.\*, avg(abs(qty)) over (partition by pid order by tim rows between 60 preceding and current row) avg60, stddev(abs(qty)) over (partition by pid order by tim rows between 60 preceding and current row) sd60 from currency\_journal a) select jid, tid, tim, aid, pid, qty, (qty-avg60)/(case when sd60=0 then null else sd60 end) z from tmp where (qty-avg60)/(case when sd60=0 then null else sd60 end)>3

(b) create table unusual\_currency with tmp as (select a.\*, avg(abs(qty)) over (partition by pid order by tim preceding and current row) avg60, stddev(abs(qty)) over (partition by pid rows between 60 preceding and current row) sd60 from currency\_journal a), sigma as ( select jid, tid, tim, aid, pid, qty, (qty-avg60)/(case when sd60=0 then null else sd60 end) z from tmp ) select \* from sigma where abs(z) > 3

(c) create table unusual\_currency with tmp as (select a.\*, avg(abs(qty)) avg60, stddev(abs(qty)) sd60 from currency\_journal a) select abs(qty)-avg60/sd60 from sigma where abs(abs(qty)-avg60)/sd60 > 3

(d) create table unusual\_currency as with tmp as (select a.\*, avg(abs(qty)) over (partition by pid order by tim rows between 60 preceding and current row) avg60, stddev(abs(qty)) over (partition by pid order by tim rows between 60 preceding and current row) sd60 from currency\_journal a), sigma as ( select jid, tid, tim, aid, pid, qty, (abs(qty)-avg60)/(case when sd60=0 then null else sd60 end) z from tmp ) select \* from sigma where abs(z) > 3

(e) Other:

19. (5 points) As custodians we are required to monitor customer activity for potential money laundering. Using table from previous question, identify instances where an *unusual* transfer, and a *reverse* happen within 24 hours of each other (e.g. customer using USD to buy GBP and within a day buying back USD).

(a) select \* from unusual\_currency a inner join unusual\_currency b on a.aid=b.aid and a.pid=b.pid where a.tid<b.tid and b.tim-a.tim < interval '24 hour'

- (b) `select * from unusual_currency natural inner join unusual_currency where a.tid<b.tid and b.tim-a.tim < interval '24 hour'`
- (c) `select * from unusual_currency a inner join unusual_currency b where a.aid=b.aid and a.pid=b.pid and a.tid<b.tid and b.tim-a.tim< interval '24 hour'`
- (d) `select * from unusual_currency a left outer join unusual_currency b on a.aid=b.aid and a.pid=b.pid where a.tid<b.tid and b.tim-a.tim < 24`
- (e) Other:

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