CISC 7512X Final Exam

For the below questions, use the following schema definition.

```
customers(cid, fname, lname, email)
accounts(aid, cid, accttype)
securities(sid, ticker, cname, stype)
accountevents(eid, aid, sid, etype, qty, px, cashamt, ets)
```

This schema models a simplified brokerage system. Customers own one or more brokerage accounts, accounts interact with tradable securities, and all financial activity is recorded in a single accountevents ledger table.

- The customers table stores client identity information using fields such as cid, fname, lname, and email.
- The accounts table stores brokerage account metadata, including aid, the owning cid, and accttype.
- The securities table stores tradable instruments using sid, ticker, cname, and stype.
- The accountevents table is the core transactional ledger and records all account activity. Important fields include
  - aid identifying the affected account,
  - sid identifying the traded security,
  - etype describing the event type such as:
    - DEPOSIT — cash added to account ($cashamt > 0$, no sid)
    - WITHDRAW — cash removed from account ($cashamt < 0$, no sid)
    - BUY — purchase of security (sid, qty, px, $cashamt < 0$)
    - SELL — sale of security (sid, qty, px, $cashamt > 0$)
  - qty and px describing traded shares and execution price,
  - cashamt recording the resulting cash movement,
  - ets storing the event timestamp,

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. SQL to get full name and email address of all customers?
   (a) SELECT fname, lname, email FROM accounts;
   (b) SELECT fname, lname, email FROM customers;
   (c) SELECT cid, accttype FROM customers;
   (d) SELECT fname, lname FROM securities;

2. SQL to list all brokerage accounts along with the customer's last name?

   (a) SELECT accounts.aid, customers.lname FROM accounts JOIN customers ON accounts.cid = customers.cid;

   (b) SELECT accounts.aid, securities.lname FROM accounts JOIN securities ON accounts.sid = securities.sid;

   (c) SELECT accounts.cid, customers.sid FROM accounts JOIN customers ON accounts.cid = customers.cid;

   (d) SELECT accounts.aid, customers.email FROM securities JOIN customers ON accounts.cid = customers.cid;

3. SQL to get all BUY and SELL events along with the associated security ticker?

   (a) SELECT e.eid, s.ticker FROM accounts e JOIN securities s ON e.sid = s.sid;

   (b) SELECT e.eid, c.ticker FROM accountevents e JOIN customers c ON e.cid = c.cid;

   (c) SELECT e.eid, s.ticker FROM accountevents e JOIN securities s ON e.aid = s.sid;

   (d) SELECT e.eid, s.ticker FROM accountevents e JOIN securities s ON e.sid = s.sid WHERE e.etype IN ('BUY','SELL');

4. SQL to get number of accounts per customer?

   (a) SELECT cid, COUNT(*) FROM customers GROUP BY cid;

   (b) SELECT cid, COUNT(aid) FROM securities GROUP BY cid;

   (c) SELECT c.cid, COUNT(a.aid) FROM customers c JOIN accounts a ON c.cid = a.cid GROUP BY c.cid;

   (d) SELECT cid, COUNT(*) FROM accountevents GROUP BY cid;

5. SQL to get total cash deposited (DEPOSIT events only) per account?

   (a) SELECT aid, SUM(qty) FROM accountevents WHERE etype = 'DEPOSIT';

   (b) SELECT aid, SUM(cashamt) FROM accountevents WHERE etype = 'DEPOSIT' GROUP BY aid;

   (c) SELECT cid, SUM(cashamt) FROM accounts GROUP BY cid;

   (d) SELECT aid, SUM(px) FROM accountevents WHERE etype = 'BUY';

6. SQL to get the first and last name of customers who have at least one BUY event?

   (a) SELECT c.fname, c.lname FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid WHERE e.etype = 'SELL';

   (b) SELECT c.fname, c.lname FROM customers c JOIN accountevents e ON c.cid = e.cid WHERE e.etype = 'BUY';

   (c) SELECT c.fname, c.lname FROM securities s JOIN accounts a ON s.sid = a.sid JOIN customers c ON c.cid = a.cid WHERE s.etype = 'BUY';

   (d) SELECT c.fname, c.lname FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid WHERE e.etype = 'BUY';

7. SQL to get customers who do NOT own any account that has traded AAPL?

   (a) SELECT DISTINCT c.cid
   FROM customers c LEFT JOIN accounts a ON c.cid = a.cid LEFT JOIN accountevents e ON a.aid = e.aid LEFT JOIN securities s ON e.sid = s.sid
   GROUP BY c.cid
   HAVING MAX(CASE WHEN s.ticker = 'AAPL' THEN 1 ELSE 0 END) = 0;

   (b) SELECT DISTINCT c.cid
   FROM customers c LEFT JOIN accounts a ON c.cid = a.cid LEFT JOIN accountevents e ON a.aid = e.aid LEFT JOIN securities s ON e.sid = s.sid
   WHERE s.ticker IS NULL OR s.ticker <> 'AAPL';

   (c) SELECT DISTINCT c.cid
   FROM customers c JOIN accounts a ON c.cid = a.cid JOIN securities s ON a.sid = s.sid
   WHERE s.ticker <> 'AAPL';

   (d) SELECT c.cid
   FROM securities s LEFT JOIN customers c ON c.cid = s.sid
   WHERE s.ticker = 'AAPL';

8. SQL to get securities that have never been traded?

   (a) SELECT s.sid FROM securities s JOIN accountevents e ON s.sid = e.sid WHERE e.sid IS NULL;

   (b) SELECT e.sid FROM accountevents e LEFT JOIN securities s ON e.sid = s.sid WHERE s.sid IS NULL;

   (c) SELECT s.sid FROM securities s JOIN accounts a ON s.sid = a.aid;

   (d) SELECT s.sid FROM securities s LEFT JOIN accountevents e ON s.sid = e.sid WHERE e.sid IS NULL;

9. SQL to get customers who currently hold at least one share of AAPL?

   (a) SELECT DISTINCT c.cid
   FROM customers c
   JOIN securities s ON c.cid = s.sid
   WHERE s.ticker = 'AAPL';

   (b) SELECT c.cid
   FROM customers c
   JOIN accounts a ON c.cid = a.cid
   WHERE s.ticker = 'AAPL';

   (c) SELECT DISTINCT c.cid
   FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid JOIN securities s ON e.sid = s.sid
   WHERE s.ticker = 'AAPL'
   GROUP BY c.cid
   HAVING SUM(CASE WHEN e.etype = 'BUY' THEN e.qty ELSE -e.qty END) > 0;

   (d) SELECT DISTINCT c.cid
   FROM customers c
   JOIN accountevents e ON c.cid = e.cid WHERE e.etype = 'AAPL';

10. SQL to get top 100 customers who currently hold the most AAPL shares, ranked in descending order of shares held?

    (a) SELECT c.cid, SUM(CASE WHEN e.etype = 'BUY' THEN e.qty ELSE -e.qty END) AS shares,
    RANK() OVER (ORDER BY SUM(CASE WHEN e.etype = 'BUY' THEN e.qty ELSE -e.qty END) DESC) AS rnk
    FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid JOIN securities s ON e.sid = s.sid
    WHERE s.ticker = 'AAPL'
    GROUP BY c.cid
    HAVING rnk <= 100;

    (b) WITH positions AS (SELECT c.cid, SUM(CASE WHEN e.etype = 'BUY' THEN e.qty ELSE -e.qty END) AS shares
    FROM customers c JOIN accounts a ON c.cid = a.cid LEFT JOIN accountevents e ON a.aid = e.aid LEFT JOIN securities s ON e.sid = s.sid
    WHERE s.ticker = 'AAPL'
    GROUP BY c.cid),
    ranked AS (SELECT cid, shares, RANK() OVER (ORDER BY shares DESC) AS rnk
    FROM positions
    WHERE shares > 0)
    SELECT cid, shares, rnk
    FROM ranked
    WHERE rnk <= 100;

    (c) SELECT c.cid, SUM(e.qty) AS shares, RANK() OVER (PARTITION BY c.cid ORDER BY e.qty DESC) AS rnk
    FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid JOIN securities s ON e.sid = s.sid
    WHERE s.ticker = 'AAPL'
    GROUP BY c.cid HAVING rnk <= 100;

    (d) SELECT c.cid, SUM(e.cashamt) AS shares, RANK() OVER (ORDER BY SUM(e.cashamt) DESC) AS rnk
    FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid JOIN securities s ON e.sid = s.sid
    WHERE s.ticker = 'AAPL'
    GROUP BY c.cid;

11. SQL to get total number of distinct customers who have made at least one account event?

    (a) SELECT COUNT(*) FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid;

    (b) SELECT COUNT(DISTINCT c.cid) FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid;

    (c) SELECT COUNT(DISTINCT e.eid) FROM accountevents e;

    (d) SELECT COUNT(*) FROM accounts a;

12. SQL to get total number of accounts per account type?

    (a) SELECT accttype, COUNT(*) FROM accounts GROUP BY accttype;

    (b) SELECT accttype, SUM(aid) FROM accounts GROUP BY accttype;

    (c) SELECT accttype, COUNT(DISTINCT cid) FROM customers GROUP BY accttype;

    (d) SELECT accttype, COUNT(*) FROM accountevents GROUP BY accttype;

13. SQL to get all customers and the number of brokerage accounts they own (including customers with zero accounts)?

    (a) SELECT c.cid, COUNT(a.aid) FROM customers c JOIN accounts a ON c.cid = a.cid GROUP BY c.cid;

    (b) SELECT c.cid, SUM(a.aid) FROM customers c LEFT JOIN accounts a ON c.cid = a.cid GROUP BY c.cid;

    (c) SELECT c.cid, COUNT(a.aid) FROM customers c LEFT JOIN accounts a ON c.cid = a.cid GROUP BY c.cid;

    (d) SELECT c.cid, COUNT(DISTINCT e.eid) FROM customers c LEFT JOIN accountevents e ON c.cid = e.eid GROUP BY c.cid;

14. SQL to get the net position (BUY minus SELL) per customer per security, including only customers with non-zero positions?

    (a) SELECT c.cid, s.sid, SUM(CASE WHEN e.etype = 'BUY' THEN e.qty ELSE -e.qty END) AS position
    FROM customers c
    JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid JOIN securities s ON e.sid = s.sid
    GROUP BY c.cid, s.sid
    HAVING SUM(CASE WHEN e.etype = 'BUY' THEN e.qty ELSE -e.qty END) <> 0;

    (b) SELECT c.cid, s.sid, SUM(e.qty) AS position
    FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid
    GROUP BY c.cid, s.sid
    HAVING e.etype = 'BUY';

    (c) SELECT c.cid, s.sid, SUM(e.cashamt) AS position
    FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid
    GROUP BY c.cid, s.sid;

    (d) SELECT c.cid, s.sid, COUNT(e.qty) AS position
    FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid
    GROUP BY c.cid, s.sid;

15. SQL to get customers whose portfolio value (SUM qty * px for BUY/SELL net) is the highest?

    (a) SELECT c.cid, SUM(e.qty) AS portfolio_value
    FROM customers c JOIN accountevents e ON c.cid = e.cid
    GROUP BY c.cid;

    (b) SELECT c.cid, SUM(e.cashamt) AS portfolio_value
    FROM customers c
    GROUP BY c.cid;

    (c) SELECT c.cid, AVG(e.qty * e.px)
    FROM customers c JOIN securities s ON c.cid = s.sid
    GROUP BY c.cid;

    (d) SELECT c.cid, SUM(CASE WHEN e.etype = 'BUY' THEN e.qty * e.px ELSE -e.qty * e.px END) AS portfolio_value
    FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid
    GROUP BY c.cid
    ORDER BY portfolio_value DESC;

16. SQL to get securities that have only been sold (never bought)?

    (a) SELECT s.sid
    FROM securities s JOIN accountevents e ON s.sid = e.sid
    WHERE e.etype = 'SELL';

    (b) SELECT s.sid
    FROM securities s JOIN accountevents e ON s.sid = e.sid
    GROUP BY s.sid
    HAVING SUM(CASE WHEN e.etype = 'BUY' THEN 1 ELSE 0 END) = 0 AND SUM(CASE WHEN e.etype = 'SELL' THEN 1 ELSE 0 END) > 0;

    (c) SELECT s.sid
    FROM securities s
    WHERE e.etype = 'SELL';

    (d) SELECT s.sid
    FROM accountevents e
    GROUP BY s.sid
    HAVING e.etype = 'SELL';

17. SQL to get customers whose cash balance is negative?

    (a) SELECT c.cid
    FROM customers c JOIN accounts a ON c.cid = a.cid
    WHERE e.cashamt < 0;

    (b) SELECT c.cid
    FROM customers c
    GROUP BY c.cid
    HAVING COUNT(e.cashamt) < 0;

    (c) SELECT c.cid
    FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid
    GROUP BY c.cid
    HAVING SUM(e.cashamt) < 0;

    (d) SELECT c.cid
    FROM accountevents e
    WHERE SUM(e.cashamt) < 0;

18. Suppose the following query is frequently executed:

```
SELECT * FROM accountevents WHERE aid = ?  AND ets BETWEEN ? AND ?;
```

Which index is most appropriate to optimize this query?

(a) CREATE INDEX idx_events_ets ON accountevents(ets);

(b) CREATE INDEX idx_events_aid_ets ON accountevents(aid, ets);

(c) CREATE INDEX idx_events_sid ON accountevents(sid);

(d) CREATE INDEX idx_events_etype ON accountevents(etype);

19. Which query correctly computes the running cash balance per account ordered by event time?

(a) SELECT aid, ets, SUM(cashamt) FROM accountevents GROUP BY aid;

(b) SELECT aid, ets, AVG(cashamt) OVER (PARTITION BY ets ORDER BY aid) FROM accountevents;

(c) SELECT aid, ets, SUM(cashamt) OVER (PARTITION BY aid ORDER BY ets) AS running_balance FROM accountevents;

(d) SELECT aid, SUM(cashamt) OVER (ORDER BY aid) FROM accountevents GROUP BY ets;

20. SQL to get the last 10 transactions for each customer ordered by event timestamp?

(a) SELECT c.cid, e.eid, e.ets
FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid
WHERE ROW_NUMBER() OVER (PARTITION BY c.cid ORDER BY e.ets DESC) <= 10;

(b) SELECT c.cid, e.eid, e.ets
FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid
GROUP BY c.cid, e.eid, e.ets
HAVING ROW_NUMBER() OVER (PARTITION BY c.cid ORDER BY e.ets DESC) <= 10;

(c) WITH STATS AS (
SELECT c.cid, e.eid, e.ets, ROW_NUMBER() OVER (PARTITION BY c.cid ORDER BY e.ets DESC) AS rn
FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid)
SELECT * FROM x WHERE rn <= 10;

(d) SELECT c.cid, e.eid, e.ets
FROM customers c JOIN accounts a ON c.cid = a.cid JOIN accountevents e ON a.aid = e.aid
ORDER BY e.ets DESC
LIMIT 10;