

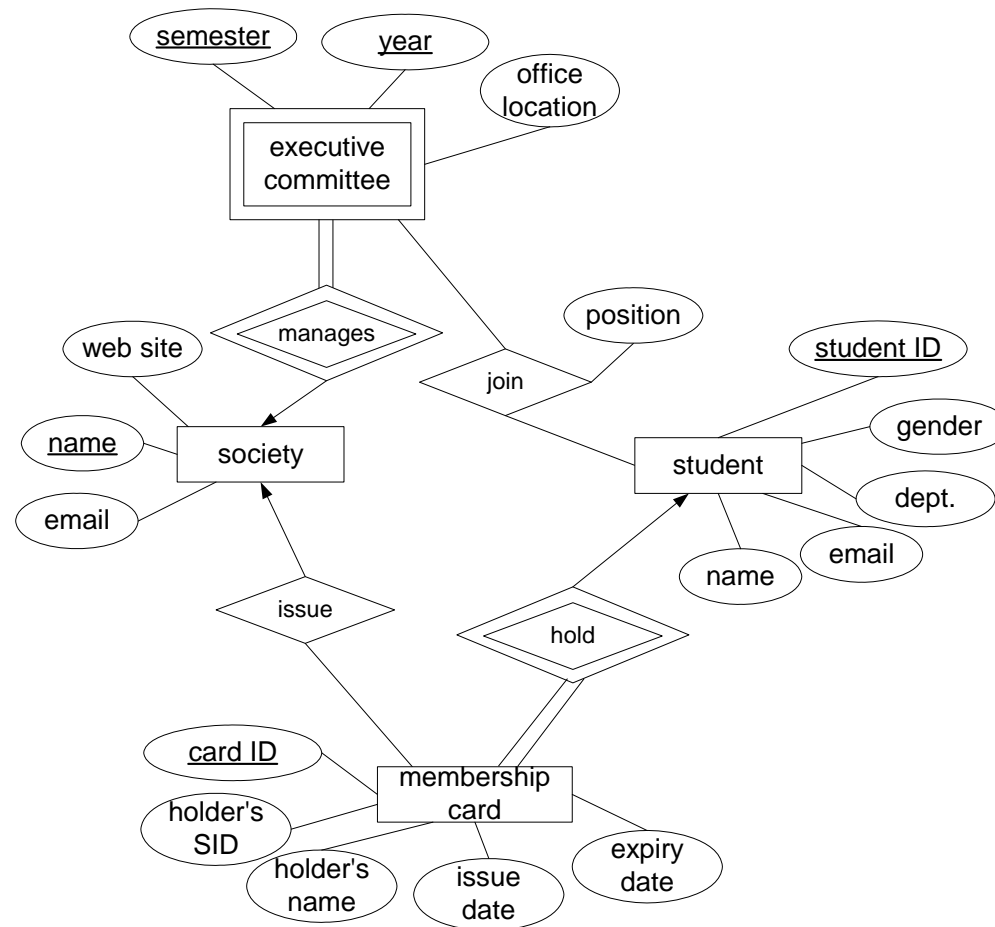
《数据库系统原理》 期末试题答案 (B)

Question 1 Multiple Choices (10 points, 2 for each)

(a) B (b) DE (c) A (d) AD (e) AB

Question 2 Database Design (16 points)

(a) (10 points)



(b) (6 points)

society(name, email, web site)

executive-committee(society name, start date, end date, office location)

student(SID, name, email, gender, department)

membership-card(CID, holder's SID, holder's name, issue date, expiry date)

issue(name, card ID)

join (SID, society name, start date, end date, position)

Question 3 RA and SQL (18 points)

(a)

(i) RA (4 points)

$\pi_{TITLE} (\sigma_{LAST-NAME = 'Sipser'} (BOOK \text{ JOIN } AUTHOR.AID = BOOK.AID \text{ } AUTHOR))$

(ii) SQL (4 points)

```
SELECT B.TITLE
FROM BOOK B, AUTHOR A
WHERE A.LAST-NAME = 'Sipser' AND A.AID = B.AID
```

(b)

(i) RA (5 points)

$\pi_{CID}CUSTOMER - \pi_{CID}ORDER$

(ii) SQL (5 points)

```
SELECT C.CID
FROM CUSTOMER C
WHERE NOT EXISTS (
    SELECT *
    FROM ORDER
    WHERE C.CID = ORDER.CID)
```

Question 4 Constraints and Normalization (16 points)

(a) (6 points)

3 for customer-name $\rightarrow X$

X is any two attributes of {customer-name, customer-street, customer-city}

3 for branch-name $\rightarrow Y$

Y is any two attributes of {branch-city, assets, branch-name }

3 for account-number $\rightarrow Z$

Z is any two attributes of {branch-name, balance, account-number }

(b) (4 points)

(Case 1)

- Check if the account number of the inserted tuple is unique in the account table (reject the insertion if not) and the branch name is in the branch table (reject the insertion if not).

(Case 2)

- Check if the account number is currently used in the depositor table (reject the deletion if it is the case or do the cascade delete).

(c) (6 points)

No. Because the attribute assets is not prime and branch-city is not a superkey of the branch schema.

Decomposition:

Obtain the canonical form of all FDs {branch-name \rightarrow branch-city, branch-city \rightarrow assets}

The key is branch-name

{branch-name, branch-city} from branch-name \rightarrow branch-city

{branch-city, asserts} from branch-city \rightarrow assets

Question 5 Query Optimization (16 points)

(a)

Solution part I: (2 points)

The number of blocks for CUSTOMERS $b_c = 10,000 / 20 = 500$ blocks

The number of blocks for BRANCHES $b_b = 2,000 / 40 = 50$ blocks

The number of blocks for ACCOUNTS $b_a = 12,000 / 30 = 400$ blocks

Solution part II: (6 points)

i. Cost = $b_c + b_a = 500 + 400 = 900$ block accesses

ii. Cost = $b_c + b_c * b_a = 500 + 500 * 400 = 200,500$ block accesses

iii. Cost = $\lceil b_c / (M - 2) \rceil * b_a + b_c = (500 / 20) * 400 + 500 = 10,500$ block accesses

(b) (8 points)

(If the conclusion is wrong, will deduct all the points)

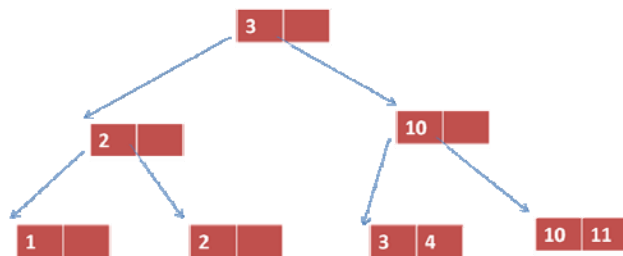
Strategy 2 is better. (4 points)

Because in Strategy 1, CUSTOMERS \bowtie BRANCHES is equal to the cross-product of the two relations and the size of the join result will become as large as $10,000 * 2,000 = 20,000,000$ tuples. This intermediate result is very large and later when joining this intermediate result with ACCOUNTS, the cost is also large

In comparison, in Strategy 2, CUSTOMERS \bowtie ACCOUNTS has only 12,000 tuples. And later when joining this intermediate result with BRANCHES, the cost is also small. (Just need to answer the key points).

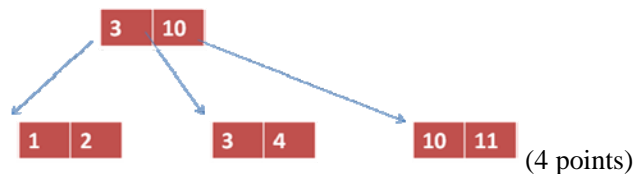
Question 6 Indexing and Storage structure (16 points)

(a) (8 points)



(b) (8 points)

Step 1: sort the keys: 1,2,3,4,10,11 (1 points)

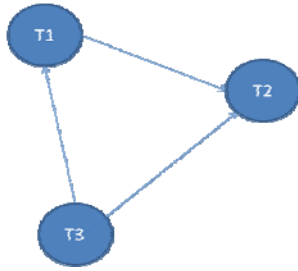


Advantage: (3 points)

- 1) Fewer I/Os during build.
- 2) Leaves will be stored sequentially (and linked, of course).
- 3) Can control “fill factor” on pages.

Question 7 Transaction Management (8 points)

(a) (5 points)



The dependency contains no circular loop. Therefore, S is conflict-serializable.

(b) (3 points)

$T3 \rightarrow T1 \rightarrow T2$