

《SE-304 数据库系统》期末试题答案(A)

(考试形式：开卷 考试时间：2 小时)



《中山大学授予学士学位工作细则》第六条

考试作弊不授予学士学位

方向：_____ 姓名：_____ 学号：_____

出卷：冯剑琳、郑贵锋_____ 审核：_____

注意：答案一定要写在答卷中，写在本试题卷中不给分。本试卷要和答卷一起交回。

Question 1. (20 marks) Assume a simple case of non-1NF relations defined as follows: all values are non-empty sets of integers. For example, the following is a schema of A, B, C and D.

A	B	C	D
{1,2}	{2,4}	{1,2}	{1,4}
{2,3}	{4,5}	{2,3}	{2,3}
{1,3}	{1,3}	{2,3}	{2,3}

Suppose an FD $X \rightarrow Y$ (X and Y are sets of attributes) is newly defined as follows: any pair of tuples t_1 and t_2 , if for all attributes A in X , $t_1[A]$ and $t_2[A]$ have common elements (如果所有属性均有公共元素) then there exists an attribute B in Y such that $t_1[B]$ and $t_2[B]$ have common elements. For example, we have $A \rightarrow C$ in the above table but not $A \rightarrow B$.

举例来说，{1,2} {2,4} {1,2} {1,4} 这一行就是一个 tuple，{1,2} 是一个 attribute，{1,2} 里面的 1 和 2 就是两个 elements。

A	B	C	D
{1,2}	{2,4}	{1,2}	{1,4}
{2,3}	{4,5}	{2,3}	{2,3}
{1,3}	{1,3}	{2,3}	{2,3}

- (i) (5 marks) Prove or disprove if decomposition rule (If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$) of FDs holds in the new setting.

分解定律不成立。在题目所给的例子中， $A \rightarrow BC$ 成立， $A \rightarrow C$ 也成立，但 $A \rightarrow B$ 不成立。

再看另一个例子。这个例子中 $A \rightarrow BC$ 成立， $A \rightarrow B$ 也成立。但 $A \rightarrow C$ 不成立。

A	B	C	D
{1}	{1}	{1}	{1}
{1}	{1}	{2}	{2}

- (ii) (5 marks) Prove or disprove if transitivity rule (If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$) of FDs holds in the new setting.

传递律不成立。反例如下：

A	B	C	D
{1}	{1}	{1}	{1}
{1}	{1}	{2}	{2}

$A \rightarrow BC$ 和 $BC \rightarrow D$ 均成立。

但是 $A \rightarrow D$ 不成立。

($BC \rightarrow D$ 成立的原因可由“蕴含 $a \Rightarrow b$ 中若 a 为假，则 $a \Rightarrow b$ 为真”推出)

- (iii) (10 marks) Prove or disprove if union rule (If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$) of FDs holds in the above setting.

标准答案：Union rule holds.

To prove union rule: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$

For any two tuples t_1 and t_2 , such that $t_1[A] \cap t_2[A] \neq \Phi$ for all A in X

Since $X \rightarrow Y$ holds, we have $t_1[A] \cap t_2[A] \neq \Phi$ for some A in Y .

A is in YZ and thus $X \rightarrow YZ$ also holds.

因为 $X \rightarrow Y$ ，所以对于 X 里面的所有属性，它们都有公共元素；且对于 Y 里面存在某些属性具有公共元素。

同理因为 $X \rightarrow Z$ ，所以对于 X 里面的所有属性，它们都有公共元素；且对于 Z 里面存在某些属性具有公共元素。

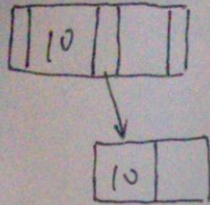
所以我们可以得出结论：“对于 X 里面的所有属性，它们都有公共元素；对于 YZ 里面存在某些属性相互之间是有公共元素的”，这就可以推出 $X \rightarrow YZ$ 了。

Question 2. (20 marks) Consider constructing a B+ tree of order $d=1$ (i.e., every node contains m entries, where $d \leq m \leq 2d$).

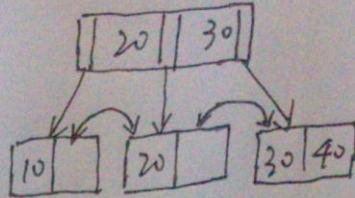
- (a) (10 marks) Show the resulting tree after inserting keys 10, 20, 30, 40, 50, 60 in this order, one by one. Draw the initial tree and the trees after every node split. (初始树、每次节点分裂后的树).
- (b) (10 marks) Is it possible, with the same set of keys, to construct a “shorter” tree— i.e., one that has a smaller height? If no, explain why not. If yes, show an order of inserting the keys and the resulting tree.

Sol: (a)

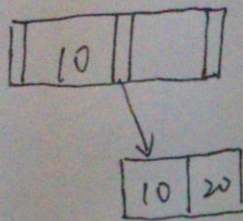
插入 10



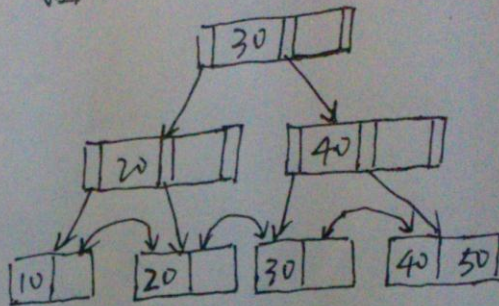
插入 40



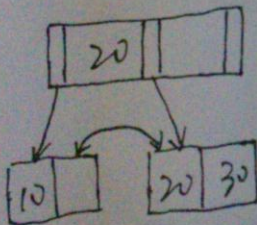
插入 20



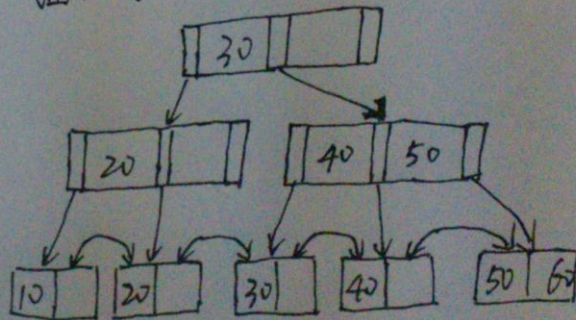
插入 50



插入 30

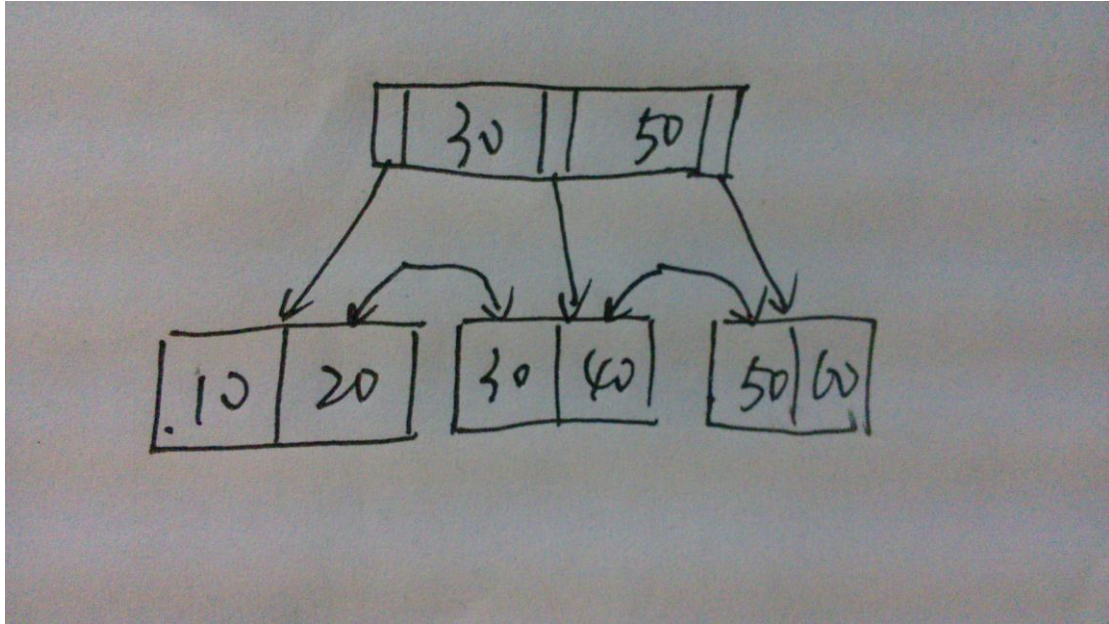


插入 60

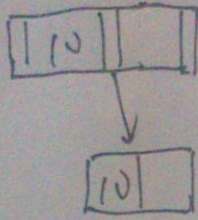


(b)

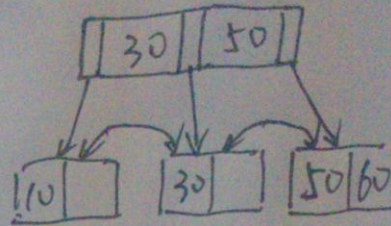
输入码的顺序 (其中一种): 10, 30, 50, 60, 40, 20



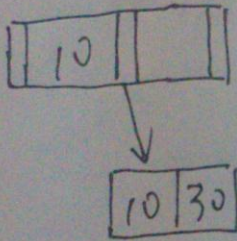
插入 10



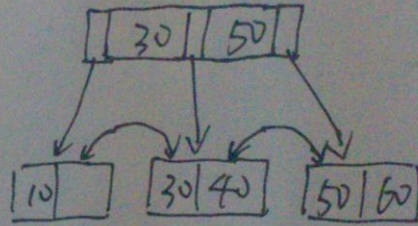
插入 60



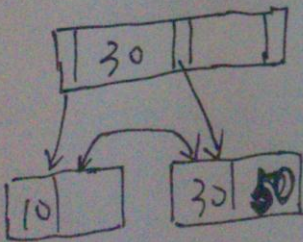
插入 30



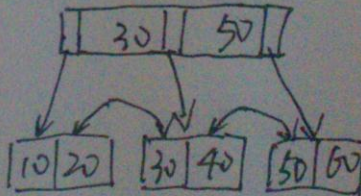
插入 40



插入 50



插入 20



Question 3. (25 marks) Consider the following two relations R and S. The number of records in R and S is also given.

- R(A, B, C, D): 20,000 records
- S(A, E): 36,000 records

Assuming the size of each memory page is 4K bytes and the memory buffer has 30 pages. For simplicity, we take 4K as 4000; we assume all attribute values and pointers, if needed to be considered, are 10 bytes. Consider the following SQL query:

Query: SELECT B, E FROM R, S WHERE R.A=S.A

Give estimation of the following questions. All the computations steps and their reasoning should be clearly explained.

(a) (10 marks) What is the size of R and S in terms of pages?

Each R record is $(10 \times 4) = 40$ bytes and each S record is $(10 \times 2) = 20$ bytes. There are $4000/40 = 100$ R records per page and $4000/20 = 200$ S records per page.

Therefore, R contains $20000/100 = 200$ pages and S contains $36000/200 = 180$ pages.

(b) (15 marks) Assume using *sort-merge join* to process the query as follows:

Step 1: Sort R on R.A and store the sorted relation in disk, assuming we discard irrelevant attributes as soon as possible.

Step 2: Sort S on S.A but compare with R (sorted already in Step 1) once a sorted page of S is generated in the final pass, assuming we *discard irrelevant attributes* as soon as possible.

Step 3: Transfer sorted R page by page from disk to memory buffer to compare the sorted pages of S generated in Step 2.

Estimate the cost of each step and then compute the total cost of sort-merge join.

Step 1: We have $\lceil 200/30 \rceil = 7$ sorted runs. At each sorted run 30 R pages are read and 15 pages are written due to discarding attribute C and D. One more pass can merge all. Cost of sorting R = $200 + 100$ (first pass) + $100 + 100$ (final pass) = 500 pages.

Step 2: We have $\lceil 180/30 \rceil = 6$ sorted runs. At each sorted run 30 S pages are read and 30 pages are written (no attribute can be discarded). One more pass can merge all and the generated pages are used for direct comparison. Cost of sorting R = $180 + 180$ (first pass) + 180 (half of the final pass due to immediate merging with R pages) = 540

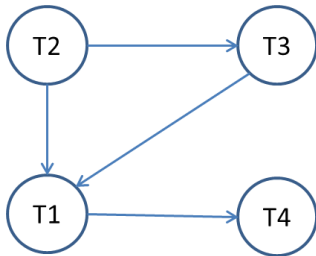
Step 3: Cost of transfer sorted R = 100

Total cost = $500 + 540 + 100 = 1140$ pages

Question 4. (15 marks) Consider the schedule S that consists of four transactions as follows: S = $\langle T1_R(X), T1_W(X), T2_R(Z), T3_R(Z), T2_R(Y), T3_W(Z), T4_R(X), T4_W(X), T2_W(Y), T1_W(Z), T3_R(Y) \rangle$. The notation is self-explanatory. For example, $T2_R(X)$ means that transaction T2 reads item X.

T1	T2	T3	T4
R(X)			
W(X)			
	R(Z)		
	R(Y)	R(Z)	
		W(Z)	
	W(Y)		R(X)
			W(X)
W(Z)			
		R(Y)	

- (a) (9 marks) Construct the precedence graph of S. Explain why or why not the schedule is conflict-serializable.



It's conflict-serializable , no circle.

- (b) (6 marks) If you find that S is serializable in (a), write down all equivalent serial schedules of S.

T2,T3,T1,T4

Question 5 (20 marks) The relational database schema for a Car-insurance company is given below:

person (driver-id, name, address)

car (license, year, model)

accident (report-number, location, date)

owns (driver-id, license)

participated (report-number, driver-id, license, damage-amount)

employee (person-name, street, city)

works (person-name, company-name, salary)

company (company-name, city)

manages (person-name, manager-name)

- (i) (10 marks) Give an expression in the **relational algebra** to formulate each of the following queries:

a. Find the names of all employees who work for the company “Bank China”.

$\pi_{person-name} (\sigma_{company-name="Bank\ China"}(works))$

b. Find the names and cities of residence of all employees who work for “Bank China”.

$\pi_{person-name, city} (employee \bowtie \sigma_{company-name="Bank\ China"}(works))$

(ii) (10 marks) Formulate an expression in **SQL** for each of the following queries.

a. Modify the database so that Zhang San now lives in “WaiHuan Dong, PanYu” (i.e. the street changes to WaiHuan Dong and city changes to PanYu for Zhang San).

```
update employee
```

```
set city = “PanYu” and street = “WaiHuan Dong”
```

```
where person-name = “ Zhang San”
```

b. Give all managers of “Bank China” a 10 percent raise.

```
update works
```

```
set salary = salary * 1.1
```

```
where person-name in (select manager-name  
from manages)
```

```
and company-name = “ Bank China”
```