中山大学软件学院 2011 级软件工程专业（2012 学年春季学期）

《SE-304 数据库系统原理》期中试题答案

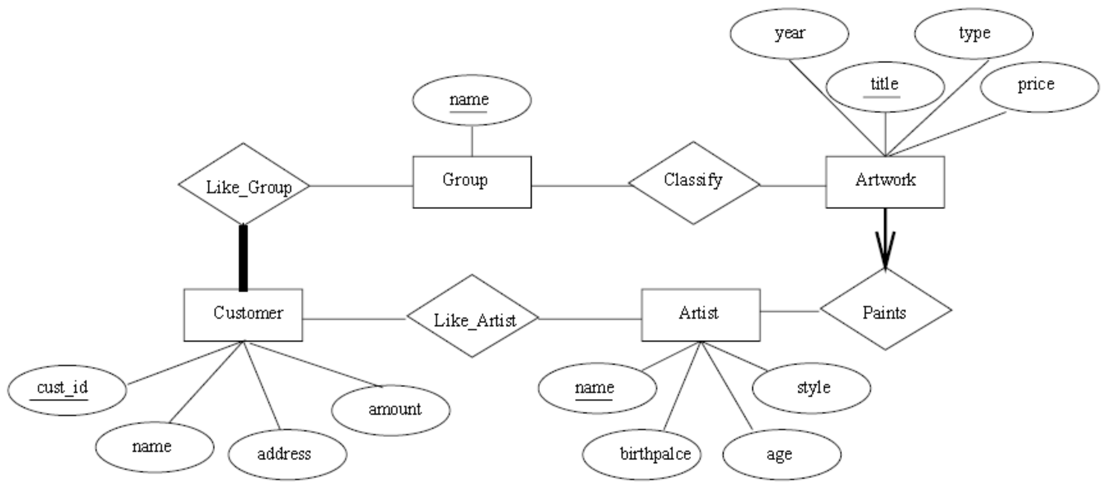
（考试形式： 闭 卷 考试时间：2 小时）

Problem 1 ER Diagram (20 points)

Suppose that you have set up a database company called “ArtBase” in order to build a product for art galleries. The core of this product is a database with a schema that captures all the information that galleries need to maintain.

- Galleries keep information about artists, their names (which are unique), birthplaces, age, and style of art.
- For each piece of artwork, the artist, the year it was made, its unique title, its type of art (e.g., painting, lithograph, sculpture, photograph), and its price must be stored. Note that each piece of artwork is painted by only one artist.
- Pieces of artwork are also classified into groups of various kinds (e.g., portraits, still lifes, works by Picasso, or works of the 19th century); a given piece of artwork may belong to more than one group. Each group is identified by a name (like those just given) that describes the group.
- Finally, galleries keep information about customers. For each customer, galleries keep that person’s unique id, name, address, total amount of dollars spent in the gallery (very important!), and the artists and groups of art that the customer tends to like. Note that each customer come to the galleries for the reason that they like at least one group of artworks.

(1) Draw the ER diagram for the database. (10 points)



(2) Translate the ER diagram into a relational schema. (10 points)

Artist (<u>name:string</u>, birthplace:string, age:integer, style:string)
Artwork (<u>title:string</u>, year:string, type:string, price:real)
Customer (<u>cust_id:integer</u>, name:string, address:string, amount:real)
Group (<u>name:string</u>)
Paints (<u>title:string</u>, name:string)

Classify (title:string, name:string)
Like_Group (cust_id:integer, name:string)
Like_Artist (cust_id:integer, name:string)

注：Paints和Artwork可以合并成
Paint_Artwork (title:string, year:string, type:string, price:real, name:string)

Problem 2 Functional Dependencies and Keys (24 points)

1. Suppose you are given a relation R with four attributes ABCD. Assume that the following FDs are the only dependencies that hold for R.
   We have the FD set: F = {AB→A, AB→C, AB→D, C→A, D→B}

(1) Compute the attribute closure AB^+ and A^+ with respect to F with detailed explanation. (4 points)

AB^+ = ABCD

AB^+内一定包含AB。因为AB→C,所以AB^+内包含C。因为AB→D,所以AB^+内包含D。

A^+=A

因为从A无法推出其他属性。所以A^+只包含A。

(2) Identify the candidate key(s) for R. Briefly explain the reason. (4 points)

A^+ = A,
B^+= B,
C^+=CA,
D^+=DB,
AB^+=ABCD,
AC^+=AC,
AD^+=ABCD,
BC^+=ABCD,
BD^+=BD,
CD^+=ABCD

所以R的候选码有AB，AD，BC，CD。

(3) Find out the trivial functional dependencies in F and prove that they are trivial FDs. (4 points)

AB→A

因为AB→A中，函数依赖右边的属性A是左边的属性AB的子集，所以它是平凡函数依赖。

2. Suppose you are given a relation R with attributes ABCDEFGHIJK. Assume that the following FDs are the only dependencies that hold for R.
   We have the FD set: FD = {ABD→E, AB→G, B→F, C→J, CJ→I, G→H}

(1) Compute the attribute closure AB^+ and C^+ with respect to FD with detailed explanation. (4 points)

AB^+ = ABFGH

AB^+内一定包含AB。因为B→F,所以AB^+内包含F。因为AB→G,所以AB^+内包

含G。因为G→H,所以AB^+内包含H。

C^+ = CJI

C^+内一定包含C。因为C→J,所以C^+内包含J。因为CJ→I,所以C^+内包含I。

(2) **Identify the candidate key(s) for R. Briefly explain the reason. (4 points)**

L = ABCD

R = EFHI

N = K

LR = GJ

计算LN集合，即LN = ABCDK，求出$ABCDK^+$=ABCDEFGHIJK，已经包含R的全部属性。所以R的候选码为ABCDK。

(3) **Write down a minimal cover for the set F. Briefly explain the reason. (4 points)**

a. 将FD中的所有依赖右边化为单一元素
   FD = {ABD→E, AB→G, B→F, C→J, CJ→I, G→H}已经满足。
b. 去掉FD中的所有依赖左边的冗余属性.
   在依赖CJ→I中,因为C^+=CJI,其中包含I，所以J是冗余的，可去除。
   FD = {ABD→E, AB→G, B→F, C→J, C→I, G→H}
c. 去掉FD中所有冗余依赖关系.
   做法为从FD中去掉某关系,如去掉X→Y,然后在FD中求X^+,如果Y在X^+中,则表明X是多余的.需要去掉.
   如果FD去掉ABD→E, FD将等于{ AB→G, B→F, C→I, G→H },而ABD^+=ADBFGH,其中不包含E.所以ABD→E不是多余的.
   同理，AB→G , B→F , C→I, G→H不是多余，故不能去掉.
   所以所求最小函数依赖集为{ ABD→E, AB→G, B→F, C→J,C→I, G→H }.

Problem 3 Normal Forms (12 points)

1. Suppose you are given a relation R with four attributes ABCD. Assume that the following FDs are the only dependencies that hold for R.
   We have the FD set: F = { C→D, C→A, B→C}
   Identify the best normal form that R satisfies (1NF, 2NF, 3NF and BCNF). If R is not in 3NF, decompose it into a set of 3NF relations that preserve the dependencies. (6 points)
   2NF. （R的码为B，B→C, C→D和B→C, C→A都有传递依赖）
   3NF分解过程：
   F的最小覆盖为{C→D, C→A, B→C}，使用保持依赖的算法得到关系模式CD、AC、BC，合并后得到ACD，BC。
2. Suppose you are given a relation R with four attributes ABCDEG. Assume that the following FDs are the only dependencies that hold for R.
   We have the FD set: F = { A→B, B→C, AD→G, D→E}
   Identify the best normal form that R satisfies (1NF, 2NF, 3NF and BCNF). If R is not in BCNF, decompose it into a set of BCNF relations that preserve the dependencies. (6 points)

1NF。（因为R的码为AD，A→B违反了部分依赖，所以R甚至不符合2NF。所以它只符合1NF）

BCNF分解过程：

① 违反函数依赖的有A→B，D→E，A→C

按A→B分解成AB,ACDEG

ACDEG按D→E分解成DE,ACDG

ACDG按A→C分解成AC,ADG

所以最后结果是AB, DE, AC,ADG （分解顺序不同也都是这个结果）

② 违反函数依赖的有A→B，D→E，B→C

按B→C分解成BC,ABDEG

ABDEG按A→B分解成AB,ADEG

ADEG按D→E分解成DE,ADG

所以最后结果是BC, AB, DE,ADG

Problem 4 Relational Algebra, Calculus and SQL (36 points)

Consider the following relations containing airline flight information:

- Flights(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time)
- Aircraft(aid: integer, aname: string, cruising_range: integer)
- Certified(eid: integer, aid: integer)
- Employees(eid: integer, ename: string, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft (otherwise, he or she would not qualify as a pilot), and only pilots are certified to fly.

Write the following queries in relational algebra (RA), tuple relational calculus (TRC) and SQL.

(1) Find the names of pilots certified for the aircrafts named “Boeing”. (12 points)

RA:

$$\pi_{ename}((\sigma_{aname='Boeing'}Aircaft) \bowtie Certified \bowtie Employees)$$

TRC:

$$\{T|\exists E \in Employees(\exists C \in Certified(\exists A \in Aircarft(A.aid = C.aid \wedge A.aname =' Boeing' \wedge E.eid = C.eid)) \wedge T.ename = E.ename)\}$$

SQL:

```
SELECT E.ename
FROM   Aircraft A, Certified C, Employees E
WHERE  A.aid = C.aid AND A.aname = ‘Boeing’ AND E.eid = C.eid
```

(2) Find the names of pilots certified for all the aircrafts. (12 points)

RA:

$$\pi_{ename}((Certified\ /\ \pi_{aid}Aircraft) \bowtie Employees)$$

TRC：

$$\{T | \exists E \in \text{Employees}(\exists C1 \in \text{Certified}(\forall A \in \text{Aircarft}(\exists C2 \in \text{Certified}(C2.\text{aid} = A.\text{aid} \wedge C2.\text{eid} = C1.\text{eid})) \wedge E.\text{eid} = C1.\text{eid}) \wedge T.\text{ename} = E.\text{ename})\}$$

SQL:

```
SELECT E.ename
FROM   Employees E
WHERE E.eid IN ( SELECT C1.eid
                 FROM   Certified C1
                 WHERE NOT EXISTS (SELECT A.aid
                                   FROM   Aircraft A
                                   WHERE NOT EXISTS ( SELECT C2.eid
                                                      FROM   Certified C2
                                                      WHERE C2.eid = C1.eid
                                                        AND C2.aid= A.aid)))
```

(3) Find the eids of employees who make the highest salary. (12 points)

RA:

$$\rho(E1, Employees)$$
$$\rho(E2, Employees)$$
$$\rho(E3, \pi_{E2.eid}(E1 \bowtie_{E1.salary > E2.salary} E2)$$
$$(\pi_{eid} E1) - E3$$

TRC:

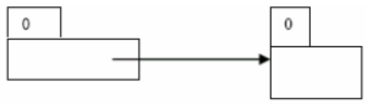
$$\{T | \exists E1 \epsilon \text{Employees} \wedge \neg(\exists E2 \epsilon \text{Employees}(\text{E2.salary} > E1.salary)) \wedge T.eid = E1.eid\}$$

SQL:

```
SELECT E.eid
FROM   Employees E
WHERE  E.salary = ( Select MAX (E2.salary)
                    FROM Employees E2 )
```

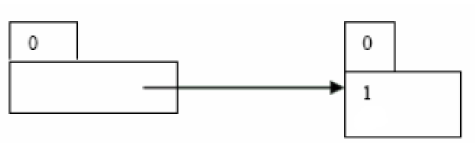
Problem 5 Indexing and Storage Structure (8 points)

Suppose that we are using extendable hashing index that contains the following search-key values K: 1,4,5,7,8,2,20. Assuming the search-key values arrive in the given order (i.e. 1 being the first coming key and 20 being the last one). The initial configuration of the structure and the binary form (assuming the choice of bits starting from Least Significant Bit (LSB)) of all keys are given in the diagram below.

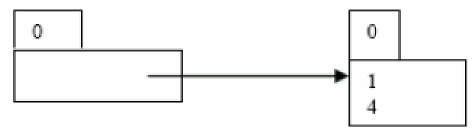


1	4	5	7	8	2	20
001	100	101	111	000	010	100

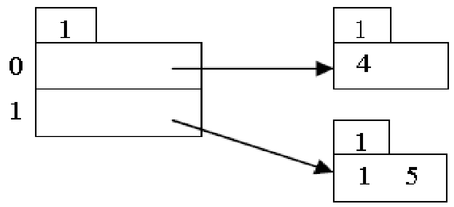
Show the extendable hash structure for each insertion of the above key values file if the hash function is *h(x) = x mod 8* and buckets (桶) can hold two keys.
EXAMPLE:
Insert 1:



Insert 4:

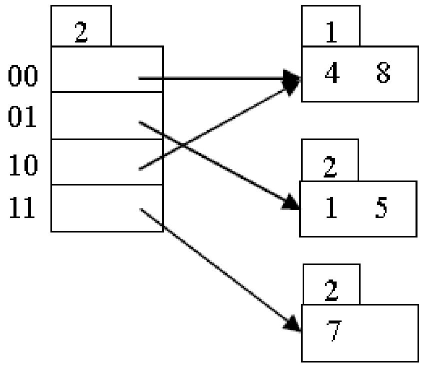


Insert 5:

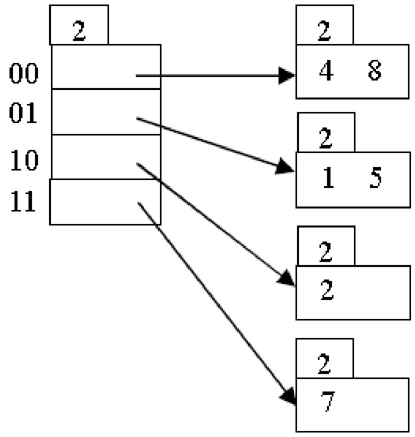


Please show the extendable hash structure for the following insertions (7, 8, 2, and 20). (8 points)

Insert 7, 8:



Insert 2:



Insert 20: