中山大学软件学院 **2009** 级软件工程专业(2010 学年春季学期)

《SE-304 数据库系统原理》期末考试试卷(B)

(考试形式：开卷　　考试时间：2 小时)

警示

《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：＿＿＿＿＿＿

Question 1 Logical Databases (20 marks)

You are required to construct an ER Diagram for a restaurant which maintains data about the following three entities: *dish, staff* and *transaction*. The three relationships are *bill*, *cook* and *processed-by* and the details of the relationships are described as follows:

- A record in **dish** is uniquely identified by a five-digit *dish-code*, and contains the information for the *name* of the dish, the *price* of the dish;
- A **staff** member has *staff-id*, *name*, and *working-time-info*;
- Each dish is cooked by exactly one staff member, but a staff member can **cook** more than one dishes, assume that all the staff members can cook for simplicity;
- A **transaction** can be uniquely identified by the following information: *date*, *time*, and the responsible *staff* who has **processed** the transaction.
- More than one units of the same dish could be sold and should be recorded by a distinct transaction, this relationship is called **bill**.

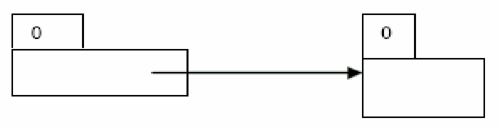
(a) (15 marks) Draw the ER diagram that consists of the three entities and the three relationships. You should write down the assumptions clearly if you think that the above description of the three entities and three relationships is not adequate for drawing the ER Diagram. You should also indicate all important information of the ER Diagram such as keys, weak entities and cardinalities.

(b) (5 marks) Convert the above ER Diagram into the corresponding relational schemas. Underline all the attribute(s) of the primary key.　Specify clearly whether a schema is obtained from an entity, a weak entity or a relationship.

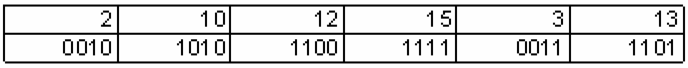
Question 2 Indexing Structure (18 marks)

Suppose that we are using extendable hashing index that contains the following search-key values K: 2, 10, 10, 12, 15, 15, 3, 13. Assuming the search-key values arrive in the given order (i.e. 2 being the first coming key and 13 being the last one). Show the extendable hash structure for each case of inserting a group of the above key values if the hash function is $h(x) = x \bmod 16$ and buckets can hold ***two*** keys. The initial configuration of the structure and the binary form of all keys are given in the diagram below.

Convention requirement: You should use the bits starting from the **most-significant one** or **MSB** (the left-hand side) as the hashing index.



Initial hash table Empty bucket



2	10	12	15	3	13
0010	1010	1100	1111	0011	1101

Binary values of all keys K

a) After inserting 2, 10, 10;

b) After inserting 12;

c) After inserting 15, 15, 3, 13.

Question 3 (22 marks) Join Algorithms

Consider the following two relations R and S. The number of records in R and S is also given.

- R(A, B, C, D): 5,000 records
- S(A, E, F, G): 18,000 records

The sizes of all attribute values are given in the table.

A	B	C	D	E	F	G
10 bytes	20 bytes	30 bytes	20 bytes	50 bytes	20 bytes	20 bytes

Assume the size of each memory page is 4K bytes and the memory buffer has 25 pages. All the computations should show the steps. Consider the following SQL query:

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

(a) (4 marks) What is the size of R and S in terms of pages?

(b) (18 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*. (Note: You may need to round up to whole page after removing the irrelevant attributes in the first pass.)

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 again we discard irrelevant attributes as soon as possible. (Note: S pages generated in the final pass do not need to write back to disk.)

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

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

Question 4 (20 marks) Physical Database Design

Consider the following schemas with usual meaning: Students and professors belong to some departments. Students are enrolled in the courses offered by some professors. The size of all the attribute values is assumed to be the same in each schema.

TABLES:	**No of records**	**No of pages**
DEPARTMENT(D_ID, D_NAME, *HEAD_ID*)	100	20
PROFESSOR(P_ID, P_NAME, SALARY, *D_ID*)	1,000	100
STUDENT(S_ID, S_NAME, *D_ID*)	10,000	1,000
COURSE(C_ID, C_NAME, *D_ID*)	1,000	100
OFFERING(O_ID, *C_ID*, YEAR, SEMESTER, *P_ID*)	10,000	1,000
ENROLL(*S_ID, O_ID*, GRADE)	100,000	10,000

General information: There are 500 different professor names, 1,000 different student names, and 100 different department names. On the average, each student is enrolled in 10 offerings and each offering has 10 students. Each department offers 10 courses. Assume that professor salaries are uniformly distributed in the range from 10,000 to 110,000.

Index information: All index pages are stored in disk. We assume B-trees with height 3 for clustered index stored and there are no overflow buckets for hash index. For indexes on a non-candidate key, you may have multiple entries with the same search key entry. We also ignore the cross-page factor for multiple records with same search key in a file if the number of records is within the page limit.

We optimize the queries in (a) and (b) as follows.

(a) *Query 1:* Display the names of all professors whose salary is in the range from 30,000 to 35,000.

Approach 1: Build a clustered index on professor.salary.

Approach 2: Build a multi-attribute index on <salary, p_name> and do index-only scan.

Which approach is better? Your answer should be based on the cost estimation with clear explanations.

(b) *Query 2:* Given a professor's name, find the name of the department that the professor belongs to.

Approach 1: Build one hash index on professor. p_name and another hash index on department.d_id.

Approach 2: Build one hash index on professor.p_name and a hash file organization on department.d_id where the department records are partitioned in 5 buckets.

Which approach is better? Your answer should be based on the cost estimation with clear explanations.

Question 5 Transaction Management (20 marks)

(a) (2 marks) Give two reasons why a serial schedule for database transactions is not practical for commercial DBMSs.

In the following questions, the notation is self-explanatory. For example, R1(X) means transaction T1 reads item X. Similarly, “W” means WRITE and “C” means “Commit”.

(b) (18 marks) For each of the following schedules, indicate whether it is conflict serializable and recoverable. If it is conflict serializable, then write out one equivalent serial schedule. The following is an example of the expected answer**.**

[Example]
R1(X), R2(X), W2(Y), C2, R1(Y), C1.
Answer:
It is serializable.
It is recoverable.
The equivalent serial schedule is T2, T1.

(i) R1(X), W1(X), R2(X), W2(X), R1(Y), W1(Y), R2(Y), W2(Y), C2, C1.

(ii) R1(X), R1(Y), R2(X), W2(Y), W2(X), C2, W1(Z), C1.

(iii) R1(X), R2(X), R3(X), W2(Y), W3(X), R3(Y), R1(Y), C3, C2, W1(X), C1