装 订 线 专业：_____ 班级：_____ 学号：_____ 班内序号：_____ 姓名：_____ 封 线 密

北京邮电大学 2022—2023 学年第一学期

《数据库系统原理》期末考试试题（A 卷）

考试注意事项	一、学生参加考试须带学生证或学院证明，未带者不准进入考场。学生必须按照监考教师指定座位就坐。 二、书本、参考资料、书包等物品一律放到考场指定位置。 三、学生不得另行携带、使用稿纸，要遵守《北京邮电大学考场规则》，有考场违纪或作弊行为者，按相应规定严肃处理。 四、学生必须将答题内容做在试题答卷上，做在试题及草稿纸上一律无效。 五、填空题用英文答，中文答对得一半分。

考试课程	数据库系统原理				考试时间			2022 年 12 月 20 日 13:30~15:30			
题号	一	二	三	四	五	六	七	八	九		总分
满分	22	20	15	13	12	18					100
得分											
阅卷教师											

1. (22 points) Here is the schema diagram for the **Banking** database(银行数据库). The table *branch* describes the name, the city located and the assets(资产) of the bank's branches (支行) . The customers of branches are represented in the table *customer*. A customer may have an *account* (存款账户) in a branch. His account is uniquely identified by the attribute *account_number*, and the attribute *balance* (存款额) records the amount of money in this account; A customer may also have a *loan* (借款账户) in a branch. His loan is solely identified by *loan_number*, and the amount of money he loans is given by the attribute *amount* (借款额). The relationships between customers and their accounts or loans are modelled as *depositor* or *borrower* respectively.

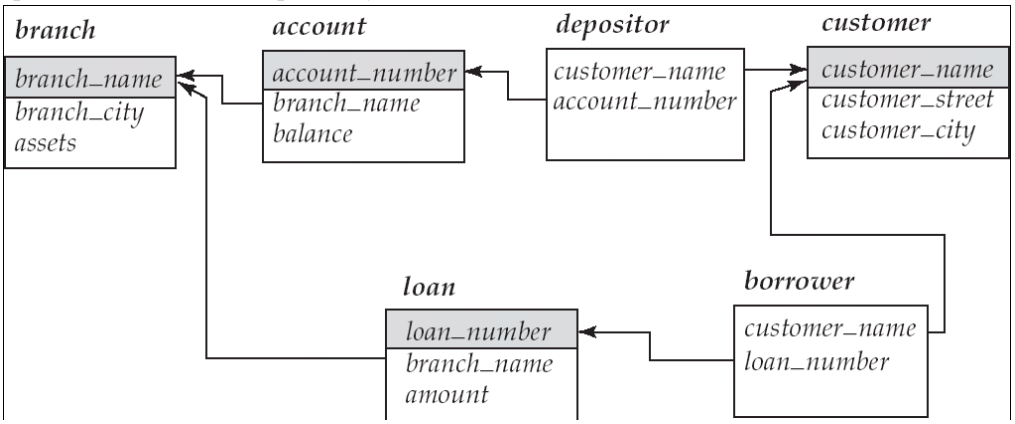

Figure 1 Schema diagram for Banking database

For the following queries, give **relational algebra** expressions for (1), and **SQL statements** for (2)~(5)

(1) Find the names of all customers who have an account at the "Haidian" branch, but at any branch have no loan whose amount is greater than 1000. (4 points)

(2) Create the table *account*, in which *account_number* is the primary key, *branch_name* is a foreign key that defines a referential integrity constraint from *account* to *branch*. It is also required that *branch_name* is not permitted to be null, and the balance of an account is not below 0. (5 points)

(3) Use one or more SQL statements to verify whether or not the functional dependency *account_number→balance* is satisfied by the table *account*. (4 points)

(4) A customer BUPT paid off all of his loans in Haidian branch, and accordingly, use SQL statements to delete related information in database. (4 points)

(5) For each branch located in Beijing, calculate the total number of the customers who have accounts in the branch. For the branch that have more than 100 account customers (i.e. depositors), list its name, the total number of the depositors, as well as the sum of the balances, in descending order of balance sum. (5 points)

2.(20 points) A cake chain store(蛋糕连锁店) needs to manage its stores, products(cakes), employees and customers, and needs to manage the following information:

(1) Stores（店铺）: including Store_ID， Address （including province, city, street, zip code), Contact telephone number and other information. Store_ID is unique, and a Store can have multiple contact numbers;

(2) Cakes（蛋糕）: including Cakes_ID , Name, Price, and the Cakes_ID is unique;

(3) Employees(员工): including Employee_ID, Name, Gender, Age, Telephone Number and Employment Date, and the Employee_ID is unique;

(4) Customers(客户): including the Customer_ID, Name, Gender, Age, Contact number. The Customer_ID is unique;

(5) Customer types: including the Type ID, Type Description. The Type ID is unique;

(6) One store has many employees, and one employee can only work in one store；

(7) A customer can only belong to one customer type，and one customer type can contain multiple customers;

(9) Sales information: Record the information about customers' purchase of cakes in a store, including the date and quantity of purchase.

Design a relational database to managing the information mentioned above:

(1) Design the E/R diagram. It is required that mapping cardinality of each relationship and participation of each entity to the relationship should be given in the diagram. (10 points)

(2) Convert the E-R diagram to the proper relational schemas, and give the primary key of each relation schema by underlines. (10 points)

3. (15 points) The functional dependency set F={A→C, B→A, C→DE, D→AC, B→E} holds on the relation schema R = (A, B, C, D, E),

(1) Compute $(AB)^+$. (2 points)

(2) List all the candidate keys of R. (1 points)

(3) What is the highest normal form of R, and why? (3 points)

(4) Is R1=(A, B, C) and R2=(C, D, E) a lossless-join decomposition of R, and Why? (3 points)

(5) Compute the canonical cover F_c. (2 points)

(6) Give a lossless-join and dependency-preserving decomposition of R into 3NF. (4points)

4. (13 points) Considering the **Banking** database in Figure 1, answer the following questions.

(1) The attribute *account_number* in the table *account* is defined as the primary key and a clustering index is created on it. Some tuples/records are then inserted into the table *account*, the records in data file storing *account* are organized as a heap file, sequential file, or multitable clustering file, and why? (3 points)

(2) For the table *account*, after *account_number* has been defined as the primary key, another index BNIdx is defined on the attribute *branch_name* and organized as a B+-tree. Then, the index BNIdx is a primary/clustering index or secondary/non-clustering index, and why? (3 points)

(3) For the following SQL query, in addition to the existing primary indices on the primary keys of the tables, on which attributes the indices can be further defined to speed up the query? (3 points)

```
select   branch_city, sum(amount)
from     branch inner join loan on branch_name
where    assets>1000
group by branch_city
```

(4) Give a SQL statement to define a composite index on combined search key (*branch_name*, *amount*) on the table *loan*.

Can this index be efficiently used for the following query, and why? (4 points)

```
select *
from loan
where amount>100
```

5. (12 points) Consider the **Banking** database given in Figure 1.

(1) Give a SQL statement to find the customer meeting the following requirements, and list his name: (i) the customer lives in Beijing; and (ii) he has an account that is with the balance more than 100 and belongs to a branch located in Tianjing. (3 points)

(2) For the SQL statement in (1), use heuristic optimization scheme to give an optimized query tree. (9 points)

6. (18 points) Consider the concurrent transactions T_1, T_2 ,T_3 and T_4 under the schedule S

(1) Construct the precedence graph for S. Is *S* a serializable schedule? If not, give the reason. If it is, give all serial schedules that are equivalent to S. (5 points)

(2) Is S a recoverable schedule? and why? (3 points)

(3) Is S a cascading or cascadeless schedule? and why? (3 points)

(4) Does S obey the two-phase locking protocol? and why? (3 points)

(5) Does S obey the strict two-phase locking protocol? and why? Does S obey the rigorous

two-phase locking protocol? and why? (4 points)

T_1	T_2	T_3	T_4
			Lock_X(A) Lock_X(C) Read(A) A:=A-100 Write(A) Unlock(A)
Lock_S(B) Read(B) Lock_S(Q)			
	Lock_S(A) Lock_X(B) Read(A) Read(B) B:=B-A Write(B) Lock_S(Q) Unlock(B)		
		Lock_S(B) Read(B) Lock_S(A)	
			Read(C) C:=C+100 Write(C)
	Read(Q) Unlock(A) Unlock(Q)		
		Read(A) Lock_X(C) Read(C) C:=A+B Write(C) Unlock(B) Unlock(C)	
Read(Q) Lock_X(C) Read(C) C:=C+Q Unlock(Q)			
			Unlock(C) commit
	commit		
Write(C) Unlock(C) Unlock(B) Commit			
		Unlock(A) commit	