参考答案

I B B C A B

II. Query Expression (4 points×10=40 points)

(1) $\Pi_{account\_number}(\sigma_{customer\_name='张三'}(\text{customer} \bowtie \text{account}))$

(2) $\Pi_{ID,\ customer\_name}(\text{customer}) - \Pi_{ID,\ customer\_name}(\text{customer} \bowtie \text{loan})$

(3) $_{ID}\mathcal{G}_{sum(balance)\ as\ s1}\ \text{Account}$

(4) $\Pi_{ID,branch\_name}(\text{loan}) \div \Pi_{branch\_name}(\text{branch})$

(5) SELECT ***account_number*** FROM account natural join **customer** WHERE customer_name like '%君%';

(6) SELECT *ID* FROM **loan** WHERE ID **not in** (SELECT ID FROM **account**);

(7) SELECT ***branch_name, count(distinct ID)*** FROM **loan** GROUP BY branch_name ORDER BY quantity **desc**;

(8) SELECT *ID* FROM **account** GROUP BY *ID* HAVING sum(*balance*)>=10000;

(9) SELECT *ID* FROM **account** GROUP BY ID HAVING sum(*balance*)>=**all** (SELECT sum(*balance*) FROM **account** GROUP BY ID);

(10) select *ID* from **loan** X where not exists(select * from **loan** Y where ID='A101' and not exists(select * from **loan** Z where *Z.id=X.id* and *Z. branch_name=Y. branch_name*))

III.

1-①(3 分): **候选码：A**

1-②(3 分): **不满足 BCNF.如 B→C，非平凡，左部不含码，违反 BCNF 条件**

1-③(4 分): **不是。Fc={ A →B, B→CE, C→D }**

2-①(3 分): {sno→sname, sno→deptname, pno→pname, pno→pfund, (sno, pno)→reward}

2-②(2 分): 候选码: (sno,pno)

2-③(5 分): 不满足 3NF，如 sno→sname 违反了 3NF 条件；用合成法进行分解:

Fc={ sno→sname, deptname, pno→pname,pfund, (sno, pno)→reward }

R1: student(sno,sname,deptname), F1={ sno→sname,deptname }

R2: project(pno,pname,pfund), F2={ pno→pname,pfund }

R3: participate(sno, pno,reward), F3={(sno, pno)→reward }

因为 R1 ∩ R3→R1 R2 ∩ R3→R2 所以分解无损

因为 R1 ∩ R3→R1，R2 ∩ R3→R2 所以分解无损

IV.

(1) 结论：S1 是冲突可串行化调度(1 分)

理由：T1 的 read(B)指令与 T2 的 read(B)可交换顺序(1 分)

结论：等价于 T1,T2(1 分)

(2) S1 是可恢复调度，当 T2 读了 T1 所写的数据 A，有 T2 的 commit 在 T1 的 commit 之后。(1 分)

S1 是无级联调度，当 T2 读了 T1 所写的数据 A，有 T2 的 read(A)在 T1 的 commit 之后。(1 分)

(3) T1:lock-X(A) read(A) writ(A) lock-S(B) read(B) commit unlock(A) unlock(B)

T2: lock-S(B) read(B) lock-S(A) read(A) commit unlock(A) unlock(B)

(4) 结论：不会(1 分)理由(1 分)：因为 T1 只对数据项 A 申请排他锁，没有对数据 B 申请排他锁，T2 只对数据项 A，数据项 B 申请读锁， 不会造成互相等待的情况。

V. Database Design (10 points*2=20 points)

1.

Customer(customer_id , name, address);

Car (license_no, model, customer_id); //Fks: customer_id ->customer(customer_id)

Policy(policy_id);

Premium_payment (policy_id , payment_no, due_date, amount, received_on); //Fks: policy_id ->policy(policy_id)

Accident(report_id, date, place) ;

Participated (license_no, report_id); //Fks: policy_id ->policy(policy_id), report_id ->accident(report_id)

Covers(license_no,policy_id) // Fks: license_no ->car(license_no), policy_id ->policy(policy_id)

2.