浙江大学 2008–2009 学年 春 季学期

《数据库系统原理》课程期末考试试卷

参考答案

1. Relational Algebra (10 points, 5 points each)

(1) $\prod_{name}((\sigma_{gender='F'}(Employee)) \bowtie EmpSkill \bowtie (\sigma_{job='DEV'}(JobSkill)))$

(2) $\prod_{name}(Employee \bowtie (EmpSkill \div \prod_{skill}(\sigma_{job='DEV'}(JobSkill))))$

2. SQL Query (20 points, 5 points each)

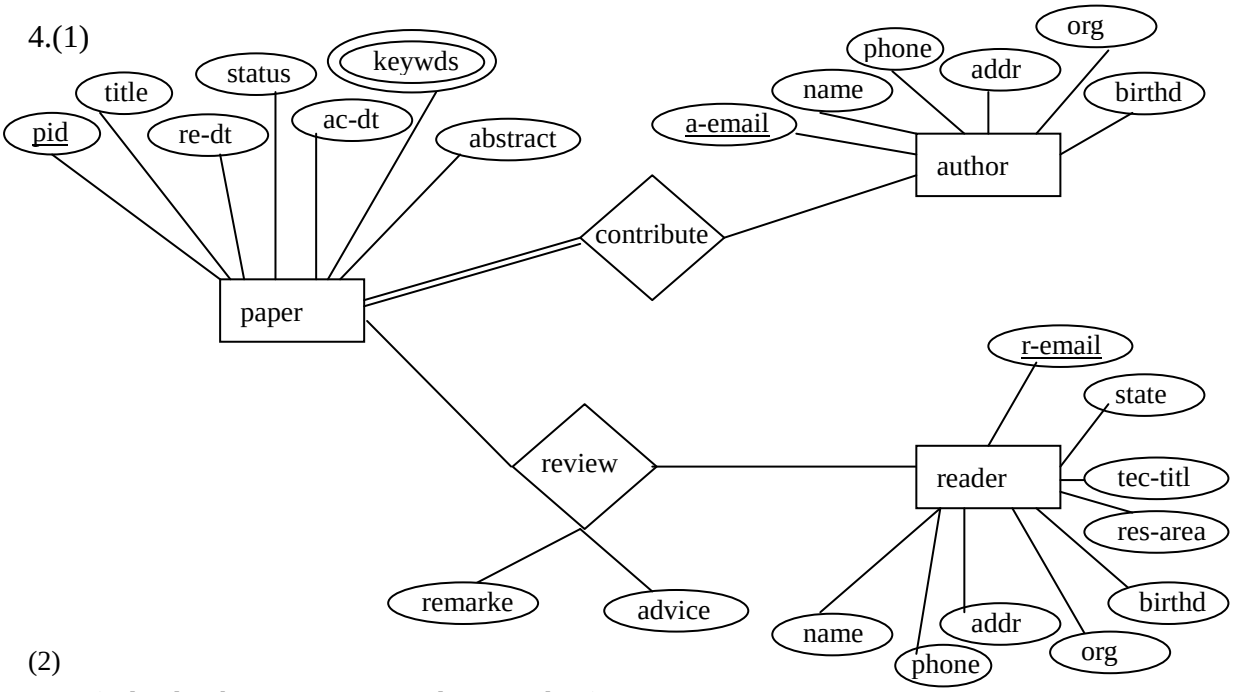
```
(1) select * from Employee where id not in ( select id from EmpSkill );
(2) select A.job from JobSkill A, JobSkill B
        where A.job=B.job and A.skill='Java' and B.skill='C++';
(2)' ( select job from JobSkil    whereskill='Java' )
      Intersect
    ( select job from JobSkil    whereskill='C++' );
(3) select name from Employee where id in ( select id from EmpSkill
                                            group by id having count(skill) >= all
                                          (select count(skill) from EmpSkill group by id ) );
(4)   select * from Employee E where not exists
          ( (select skill from JobSkill where job='DEV' or job='UAT' )
          except
          (select skill from EmpSkill K where K.id=E.id) );
(4)'   select * from Employee E where not exists
          (select * from JobSkill J where (job='DEV' or job='UAT') and not exists
          (select * from EmpSkill K where K.id=E.id and K.skill=J.skill )
```

3. Embedded SQL (10 points)

```
(1) select skill from EmpSkill where id= :id
(2) OPEN skill_cursor
(3) FETCH skill_cursor INTO :skill
(4) SQLCA.STATE = '02000'
(5) CLOSE skill_cursor
```

4. E-R Model (20 Points, 10 points each)

4.(1)



(2)

Paper(<u>pid</u>, title, abstract, status, rec-date, acc-date)

Paper-keywords(<u>pid, kword</u>);

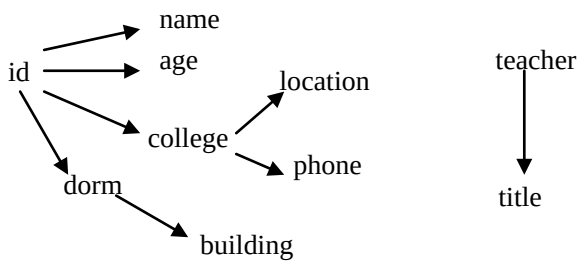
Author(<u>a-email</u>, name, birthdate, phone, address, org)

Reader(<u>r-email</u>, name, birthdate, phone, address, org, tech-title, res-area, state)

Contribute(<u>pid, a-email</u>);

Review(<u>pid, r-email</u>, remark, advice);

5. Relational Formalization (20 points, 5 points each)



(1) candidate key = (id, teacher, year)

(2) Neither BCNF and 3NF, because the left side of all function dependencies in F are not super-key, e.g. id→name, id→ age, dorm→building, …

(3) R1=(id, name), R2=(id, age), R3=(dorm, building), R4=(id, dorm), R5=(college, location), R6=(college, phone), R7=(id, college), R8=(teacher, title), R9=(id, teacher, year);

(4) the above decomposition is dependency preserving, because all function dependencies in F are preserved.

6. XML (20 points, 5 points each)

(1) relational schemas:

Project(pid, pname, budget, from, to, manager, members);

Developer(did, dname, age);

(2)

```
<!DOCTYPE research-proj [
    <!ELEMENT research (project +, developer + )>
        <!ELEMENT project (pname, budget, from, to )>
            <!ATTLIST project
                pid    ID       # REQUIRED
                members        IDREFS       # REQUIRED >
        <! ELEMENT pname (#PCDATA)>
        <! ELEMENT budget (#PCDATA)>
        <! ELEMENT from (#PCDATA)>
        <! ELEMENT to (#PCDATA)>
        <!ELEMENT developer(dname,age)>
            <!ATTLIST developer
                did    ID       # REQUIRED >
        <! ELEMENT dname (#PCDATA)>
        <! ELEMENT age (#PCDATA)>
        ]>
```

(3) Xiao Zhao
Xiao Sun

(4)

```
<talent>
    <dname>Xiao Sun</dname>
    <pname>Unstructured Data Management</pname>
</talent>
```