Database System Principles

Test One

Class_____No_____Name_____

1. ((10points) Fill in blanks

(1)The collection of information stored in the database at a particular moment is called an _instance_ of the database.

(2)The database system provides users with three levels of data abstraction, the _view_ level of abstraction describes only part of the entire database.

(3) Database design involves the following phases: requirements analysis, _Conceptual_ schema design, logical design and physical design.

(4) _Data model_ is a collection of conceptual tools for describing data, data relationships, data semantics, and data constraints.

(5) As human-machine interfaces, the database language consists of two parts, i.e the data definition language (DDL) and DML (data manipulation language).

(6) With respect to integrity mechanisms in DBS, the _trigger_ defines actions to be executed automatically when some events occur and corresponding conditions are satisfied.

(7) An entity set that does not have a primary key is referred to as a _weak entity set_.

(8) For a relation r(R), among schema, the attributes the values of which can be used to uniquely identify the tuples in r(R) is called the key/superkey/ primarykey/candidate key of R.

(9) The six fundamental operations in the relational algebra are select, project , union , set difference, _____, and rename.

(10) Let $r_1(R_1)$ and $r_2(R_2)$ be relations with primary keys K_1 and K_2 respectively, the subset α of R_2 is called the foreign key referencing K_1 in relation r_1, if for every t_2 in r_2 there must be a tuple t_1 in r_1 such that $t_1[K_1] = t_2[\alpha]$

2. (5 points) Given a table Employees and some SQL queries on it, why are these queries wrong?

Employees(employee-id, employee-name, *company-id*, employee-city, age, salary)

It is assumed that each employee has an unique id and name.

(1)
```
create table Employees
( employee-id  char(20),
  employee-name  char(20),
  company-id  char(20),
  employee-city char(20),
  age  integer,
  salary  integer,
  primary key (employee-id),
  primary key (employee-name),
  check (age >0)
)
```
不能有 2 个主键

(2)
```
select employeeid, sum(salary)
 from Employees
 group by company-id
   having avg(salary)>1000
```
employeeid 不是分组属性，不能出现在 select 子句中

3. (10 points) **给出下列关系代数操作对应的 SQL 语句**

(1) $\sigma_p(r)$ **(2)** $\Pi_{A1, A2, .., Am}(r)$

(3) $r \infty s$ **,**

，假设 r(A, B, C), s(C, E, F)

Answers：

(1) select * from r where P

(2) select A1, A2, .., Am from r

(3) select * from r natual join s

或者：

select * from r where r.C = s.C

4. (10 points) 给出下列 SQL 语句对应的关系代数表达式

(1) **select *branch-name*, max *(salary)***

from *pt-works*

group by *branch-name*

假设 pt-works(employee-name, branch-name, salary)

(2) **insert into *r***

select $A_1, A_2, \ldots, A_m$

from $r_1, r_2, .., r_n$

where P

(3) **update *loan***

set *amount* = amount *1.2

where *amount* > 1000

Answers:

(1) $_{\text{branch\_name}}G_{\text{max(salary)}}$ (pt-works)

(2) $r \leftarrow r \cup \Pi_{A1,A2,..,Am}(\sigma_p(r_1 \times r_2 \times \ldots \times r_n))$

(3) $T1 \leftarrow \Pi_{\text{loan-number, branch\_name, amount*1.2}}\sigma_{amount > 1000}$ (loan)

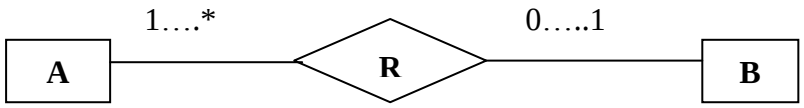
$T2 \leftarrow \sigma_{amount \le 1000}$ (loan)

$loan \leftarrow T1 \cup T2$

5. (5 points) For the entity sets **A** and **B** and the relationship set **R** among them in the following figure,

(1) point out the participation constraints of **A** and **B** in **R**

(2) what is the mapping cardinality form *A* to *B*



Answers:

(1) A: total B: partial

(2) one-to-many

6. (5points) Convert the entity set **"学生"**, of which the attribute "老乡" is a multivalued attribute, in Fig.4 into relational **tables**

student-id	籍贯	老乡	性别	年龄
07494	北京	07596, 07611	男	20
07498	河北	07320, 07321	女	19

Fig.4

Answers:

student-id	籍贯	性别	年龄
07494	北京	男	20
07498	河北	女	19

student-id	老乡
07494	07596
07494	07611

07498	07320
07498	07321

7. (20 points) There are four relations in a ***Student*** database.

Student(studentID, studentname, sex, age, birthday, schoolID)

Teacher(teacherID, teachername, sex, birthday, schoolID)

Course(courseID, coursename, teacherID)

Score(studentID, courseID, grade)

School(*SchoolID*, schoolname, dean, location)

Give SQL statements for the following queries:

(1) (5 points) Use a SQL statement to define the relational table *Student*, in which {studentID} is the primary key, and {studentname} is the candidate key and is not permitted to be null; there also exists the referential integrity between the table *Student* and *School*. It is required that the student's age is larger than 10.

create table *student*

(studentID integer primary key /*也可以放在后面用单独 primary key 语句定义

studentname varch(50) /*也可以采用其它长度的 varch、char 类型

sex varch(50)

age int,

birthday date /*或：varch(50)，

schoolID integer

primary key (studentID) /*也可以在前面 studentID 处定义主键

unique (studentname), not null,

foreign key (schoolID) references *School*,

check (age>10)

)

主键、候选键、外键、check 定义，每个 0.5 分。其它 1 分。

(2) (5points) Find the student who takes a course and the grade he gets on this course is higher than the average grade of this course. List the student's name, the name of the course he takes and gets higher grade, and his grade on this course.

Select studentname, coursename, grade
From *Score* as A, *Student*, *Course*
Where A.studentID=Student.studentID and A.courseID= *Course*.courseID
A.grade >
(Select avg(grade)
From *Score* as B
Where A.courseID=B.courseID
)

关键部分答出，2 分。

(3) (5points) Find the courses that are taught by the teachers in School of Computer Science (its SchoolID is SCS) and are taken by at least 5 students. For these courses, list their courseIDs, coursenames, average grades maximal grades and minimal grades, in descending order of the average grades.

Select courseID, coursename, avg(grade) as avggrade,
max(grade) as maxgrade, min(grade) as mingrade
From *Course* Natural Join *Score*
Where courseID in
{Select courseID
From *Teacher* Natural Join *Course*
Where schoolID=SCS
}
Group by courseID
Having count(studentID)>=5

关键部分答出，2 分。

(4) (5 points) Find all students who take the course “Database System Principles”, and increase their scores on this course by 5 points.

Update *Score* as A
Set grade=grade+5
From *Course* as B
Where A.CourseID=B.CourseID and B.coursename= Database System Principles

或者：

Update *Score* as A
Set grade=grade+5
Where CourseID in

{ Select courseID
From Course
Where coursename= Database System Principles
}

8. (5 points) Given *R* (A, B, C, D, E, F, G, H, I), and *F* = {A→F, B→E, BE→F, E→C, A→G, G→CD, I→E} holding on *R*, find out all candidate keys of *R*
(利用求候选键算法，给出计算过程)

Answers：

参见讲义附录部分的解法

9. (10 points) The relation schema R(U,F),U=(A,B,C,D,E,F,G), The functional dependencies set F ={ E→G, AC→D,FG→E,AFG→B }:

（1） Compute $(ACF)^+$ (2 points)

$(ACF)^+$={A，C，F，D} 一个 0.5

（2） Consider the decompositionρ={R1(A,C,D), R2(A,B,C,E,F,G)}, is this decomposition lossless or lossy? Why? (4points)

是 lossless。 （2 分）
因为 R1∩R2 →R1 （2 分）

10. (20points) The functional dependency set F={A→C, C→A, B→A, D→AC, B→E} holds on the relation schema R = (A, B, C, D, E),

(1) Compute $(AB)^+$ (3points)

(2) List all the candidate keys of R. (5points)

(3) What is the highest normal form of R, and why? (2 points)

(4) Compute the canonical cover F_c (5points)

(5) Give a lossless and dependency-preserving decomposition of R into 3NF. (5 points)

Answers:

(1) $(AB)^+$=ABCE

(2) 唯一候选键 BD

(3) R 所属最高范式为 1NF （1 point）

因为：

存在非主属性 A 对候选键 BD 的部分依赖 B→A（或 D→A），

或者：非主属性 E 对键 BD 的部分依赖 B→E；

非主属性 C 对键 BD 的部分依赖 D→C

(4) Fc={A→C, C→A, B→AE, D→A}

或者：Fc={A→C, C→A, B→AE, D→C}

答对 1 个即可，Fc 不完全正确，扣 1 分

(5) 分解 1：根据 Fc={ A→C, C→A, B→AE, D→A }

R1(AC), R2(ABE), R3(AD), R4(BD)

分解 2：根据 Fc={A→C, C→A, B→AE, D→C}

R1(AC), R2(ABE), R3(CD), R4(BD)