《数据库系统原理》期中测验

班级:_____ 学号:_____姓名:_____

【考试交卷时，试题连同答题纸一起上交】

1.(35 points). A travel agency management database contains the following tables.

tourist (*tid*, *tname*, *sex*, *age*, *telephone*, *email*, *address*, *registration_time*, *team_id*)
tourism_team (*team_id*, *team_name*, *start_date*, *end_date*, *line_id*)
tourism_line (*line_id*, *line_name*, *start_city*, *description*)
scenic_spot (*spot_id*, *spot_name*, *city*, *price*)
spot_line (*spot_id*, *line_id*)

(1) A *tourist*（游客）is identified by *tid*，and has attributes *tname*, *sex*, *age*, *telephone*, *email*, *address*, *registration_time*. Each tourist participates in a tourism_team with *team_id.*
(2) Each *tourism_team*（旅行团）is identified by *team_id*, and has attributes *team_name*, *start_date*, *end_date*. Each tourism_team only travels a tourism_line with *line_id.*
(3) Each *tourism_line*（旅游线路）is identified by *line_id*, and has attributes *line_name*, *start_city*, *line_description*.
(4) A *scenic_spot*（景点）is identified by a *spot_id*, and has attributes *spot_name*, *city*, *price*.
(5) Each scenic_spot with *spot_id* can belong to multiple tourism lines with *line_id*, and each tourism_line with *line_id* consists of multiple scenic spots with *spot_id*.

For the following queries, give **relational algebra** expressions for (1), and **SQL statements** for (2)~(6).

(1) Find the tourists who are male and participate in the tourism team from “2021-5-1” to “2021-5-3”, and they travel the tourism line with the name “Hangzhou-Shanghai”. List the tourists’ *name* and the *description* of the tourism_line they have travelled.

$$\Pi_{tname,description}[\sigma_{(sex="male")and\ (start\_date="2021-5-1")\ and\ (end\_date="2021-5-3")and(line\_name="Hangzhou-Shanghai")}(tourist \bowtie tourism\_team \bowtie tourism\_line)]$$

(2) Find the *name* and *price* of the scenic_spot and the *name* of the tourism_line. It is required that the *price* of the scenic_spot is below $50, the *start_city* of tourism_line is “Beijing”, and the tourism_team starts on “2021-8-1” and ends on “2021-8-10”.
答案：

```
select   spot_name, price, line_name
from     scenic_spot natural join spot_line natural join tourism_line natural join
```

tourism_team
where start_date="2021-8-1" and end_date="2021-8-10" and price<50 and start_city="Beijing"

(3) Create the table *tourist*, in which *tid* is the primary key; there exists a referential integrity constraint from *tourist* to *tourism_team*. It is also required that a tourist's *age* is over 20 years old and his/her *telephone* is not null.

答案：

```
create table tourist (tid    varchar(5),    /*类型也可以为数值型
                      tname varchar(10),
                      sex char(5),
                      age int,
                      telephone varchar(11) not null,
                      email varchar(20),
                      address varchar(30),
                      registration_time varchar(10),    /*类型也可以为日期型
                      team_id varchar(5),
                      primary key (tid),
                      foreign key (team_id) references tourism_team(team_id),
                      check (age >20)
                      )
```

(4) Delete all the information of the tourism spots whose price are more than $100.

答案：

```
delete from spot_line
where spot_id in
(select spot_id from scenic_spot
  where price>100);

delete from scenic_spot
where price>100;
```

(5) Find the tourism-line meeting the following requirements: the tourism_team starts on "2021-10-1" and ends on "2021-10-5", the *start_city* of the tourism-line is "Beijing", and the total number of the tourism_team which travels this line is more than 150. List the *line_id* of the tourism_line and the total number of the tourism_team in a descending order.

答案：

```
select    line_id, count(team_id) as teamnumber
from      tourism_team natural join tourism_line
where     start_date = “2021-10-1” and end_date=“2021-10-5” and start_city =“Beijing”
group by line_id
having count(team_id)>150
order by teamnumber desc
```

(6) Use one or more SQL statements to verify on the table *scenic_spot* (*spot_id*, *spot_name*, *city*, *price*), whether or not the functional dependency *spot_name→city* holds on the table, according to the query results of one or more SQL statements.

答案 1：判断下述语句是否为真

```
create assertion spotname2city check
   (
      not exists
         (select spot_name
           from scenic_spot
           group by spot_name
                  having count(distinct city) > 1
         )
    )
```

方法 2：如果查询结果大于 1，则函数依赖不成立

```
select       max(count(distinct city))
from           scenic_spot
group by      spot_name
```

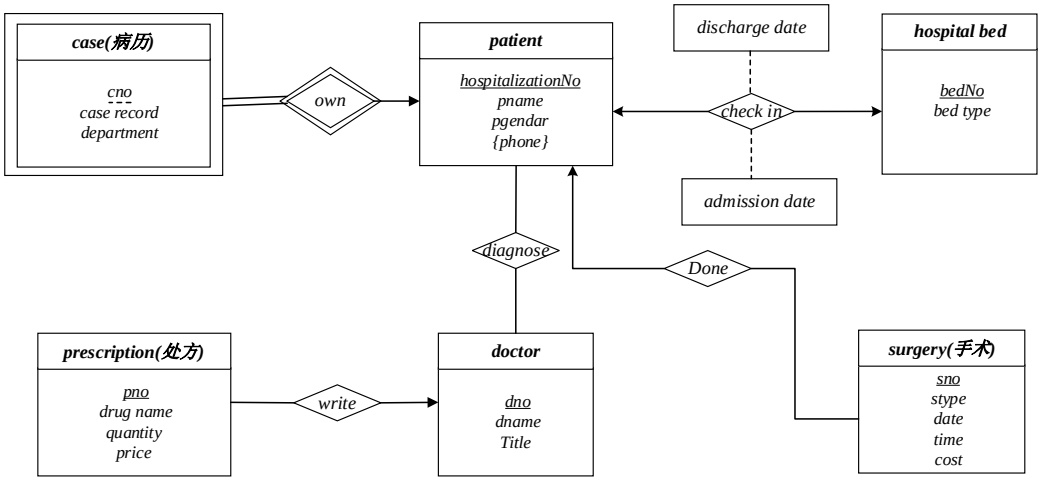
```
select max(numTuple)
 from {
   select    spot_name, count(city) as numTuple
        from       scenic_spot
        group by spot_name
 }
```

方法 3：　下述查询语句为空，则函数依赖成立

```
  select    *
    from    scenic_spot as A, scenic_spot as B
    where A. spot_name =B. spot_name
```

and (A.*city*<>B.*city*)

2. (20 points) Convert the following E-R diagram about the hospital management information system to the relation schemas, and identify the primary key of each relation by underlining the primary attributes.



答案：

弱实体集 case：case(hospitalizationNo, cno, case record, department)

方案 1：实体 patient、一对一联系 check-in、hospital bed，check-in 归并到一端 patient

patient (hospitalizationNo, pname, pgendar, bedNo, discharge date, admission date)

hospital bed (bedNo, bed type)

方案 2：实体 patient、一对一联系 check-in、hospital bed，check-in 归并到一端 hospital bed

patient (hospitalizationNo, pname, pgendar)

hospital bed (bedNo, bed type, hospitalizationNo, discharge date, admission date)

patient 的多值属性 patientphone:

patientphone (hospitalizationNo, phone)

实体 doctor：doctor(dno, dname, title)
多对多联系 diagnose：diagnose(dno, hospitalizationNo)

处方 prescription：prescription(pno, drug name, quantity, price, dno)
手术 surgery：surgery(sno, stype, date, time, cost, hospitalizationNo)

注意：
多对一联系 Done、write，由于多端非完全参与，这 2 个联系可以单独转换为关系表
Done(hospitalizationNo, Sno)
Write(pno, dno)

3. (15 points) Given *R* (A, B, C, D, E, H), and *F*={A→BC, CD→E, B→D, C→AB},
(1) find out all candidate keys of *R*
(2) what is the highest normal form of R? Why?
答案：
(1)
Step1.
L = {}
R = {E}
N = {H}
LR = {A, B, C, D}
X_set = L ∪ N = {H}
X_set^{+}=H<>R，H 不是候选键。
Step2. 由于 H 没有出现在依赖的左部、右部，下面针对 *R*1=(A, B, C, D, E)和 F，找出 *R*1 的候选键，该候选键加上属性 H，就是 R 的候选键。
2.1 从 LR = {A, B, C, D}中挑出一个属性，判断是否为 R1 的候选键。
A^{+}=R1，A 是 R1 的候选键，其它包含 A 的属性集都不是 R1 的候选键。
B^{+}=BD, B 不是 R1 的候选键。
C^{+}=R1，C 是 R1 的候选键，其它包含 C 的属性集都不是 R1 的候选键。
D^{+}=D, D 不是 R1 的候选键。

2.2 从 LR = {A, B, C, D}中拿出 2 个属性，且不包含 A 或 C，判断是否为 R1 的候选键。
$(BD)^{+}$=BD, BD 不是 R1 的候选键。

2.3 从 LR = {A, B, C, D}中拿出 3 个属性，且不包含 A 或 C，判断是否为 R1 的候选键。
找不到满足条件的个属性。
因此，A、C 是 R1 的两个候选键，AH、CH 是 R 的两个候选键。

(2) R 最高满足 1NF（4 分）。
在非主属性 B、D、E 中，对 B，从 A→BC 可知 A→B，非主属性 B 部分依赖于候选键 AH，不满足非主属性完全依赖于候选键的 2NF 要求。（3 分）
（或：从 C→AB，可知 C→B，非主属性 B 部分依赖于候选键 CH）

4. **(30 points)** The functional dependency set F={AB→C, A→DEI, B→FH, F→GH, D→IJ } holds on the relation schema R = (A, B, C, D, E, F, G, H, I, J),
(1) Compute $(AF)^+$
(2) List all the candidate keys of R.
(3) Compute the canonical cover F_c
(4) Give a lossless and dependency-preserving decomposition of R into 3NF. (10 points)
Answer:
(1) $(AF)^+$ = ADEFGHIJ
(2) L-set={A, B}
R-set={C, E, G, H, I, J}
N-set={}
LR-set={D, F}
因为 $(AB)^+$ =ABCDEFGHIJ=R，所以 AB 为 R 的唯一候选键。
(3) Fc={AB→C , A→DE, B→F, F→GH, D→IJ}
(4) 针对 Fc 中 5 条函数依赖，构造 5 个子模式。
R_1(A, B, C)
R_2(A, D, E)
R_3(B, F)
R_4(F, G, H)
R_5(D, I, J)
由于候选键 AB 已经包含在 R1 中，因此以上 5 个子模式构成 R 的 3NF 分解。