# 浙江大学 2018－2019 学年 春夏 学期

## 《数据库系统》课程期末考试试卷（A 卷）

## 参考答案及评分细则

**Answers of Problem 1：**
**(16 points, 4 points per part)**

**1)** $\Pi_{Title}(\sigma_{director="Yimou\ Zhang"}(movie) \bowtie \sigma_{grade>=4}(comment))$

评分细则：
错一处扣 4 分

**2) Update comment set grade=0 where grade is null**

评分细则：
错一处扣 4 分，grade=null, grade is null 等类似答案均给分

**3)**
```
Select type from movie, comment
Where movie.title=comment.title
Group by title
Having avg(grade) >=all (Select avg(grade)
                         From movie, comment
                         Where movie.title=comment.title
                         Group by title)
```

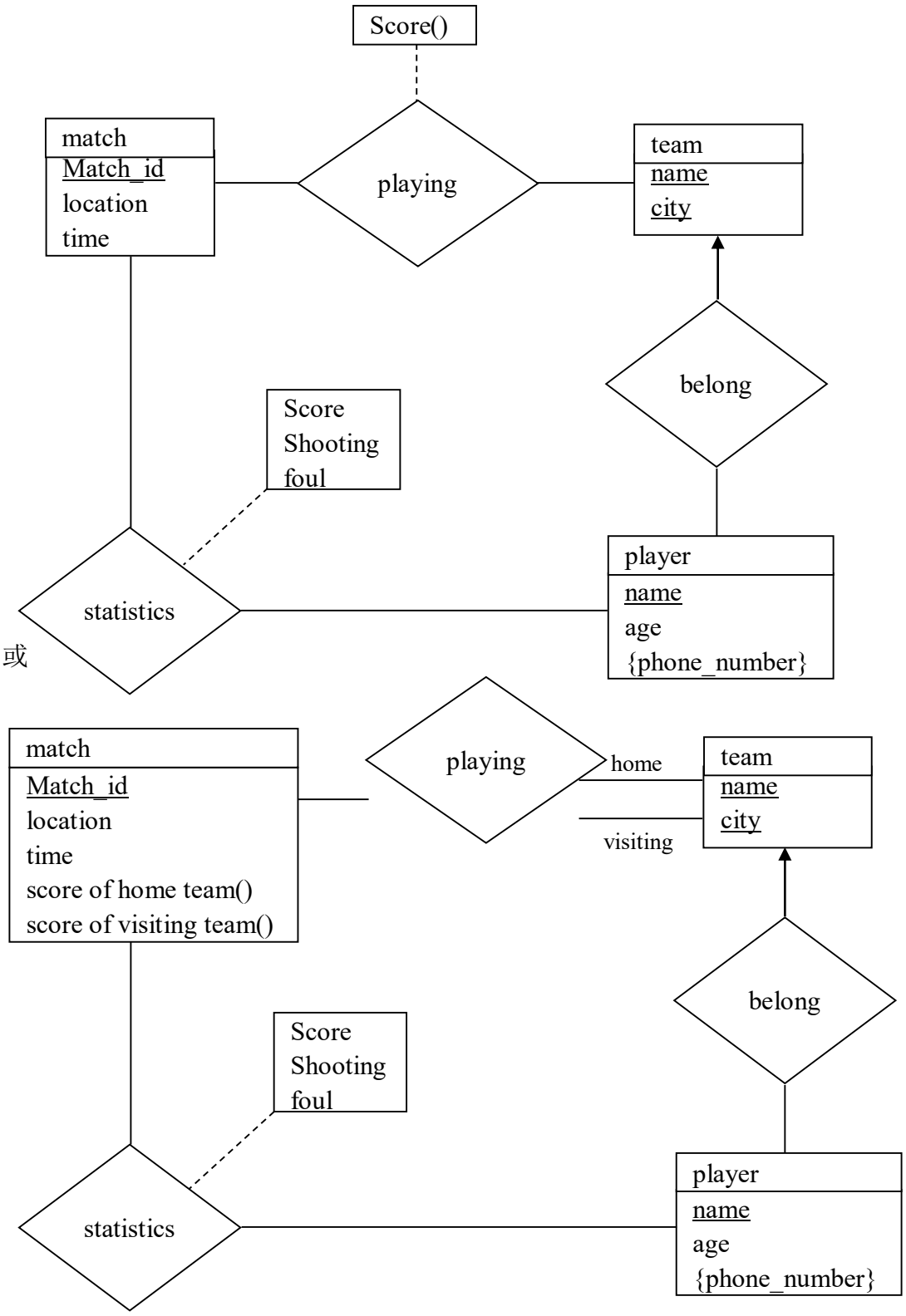
评分细则：
写出 having……且对均给 2 分，用其他 SQL 语句写出相同效果均给分

**4)**
```
Select title from movie
Except
Select title from movie
Where exists ( select *
               From comment A, comment B
               Where A.title=movie.title and A.user_name = B.user_name
               And B.title=' the avenger'
               And A.grade <=B.grade )
```

评分细则：
写出 4 个正确条件给 2 分，全对给 4 分

## Problem 2: E-R Model (9 points)

**1) (5 points)**




评分细则：

**联系写成实体扣 1 分，没写联系扣 1 分，属性写错漏写扣 1 分，少写实体或联系扣 1 分，扣完为止。**

**2) (4 points)**

**Match(match_id, location, time)**

**Team(name, city)**

**Playing(match_id, name, score)**

**Player(name, age)**

**Phone(player_name, phone_number)**

**Statistics(match_id, player_name, score, shooting, foul)**

**或者其中的 match 和 playing 改为：**

**Match(match_id, location, time, home_team_name, visiting_team_name, score_of_home_team, score_of_visiting_team)**

**Each match has one home team and one visiting team.**

**评分细则：**

**每个关系没有主键或者写错扣 1 分，扣完为止**

## Problem 3: Relational Formalization (12 points, 4 points each)

**1) {C E}**

**评分细则：**

**写对一个键给两分，多写一个扣 1 分。诸如写{ACE,CE,BCE}的不得分**

**2) Decompose R into R1(A, B) and R2(A, C, D, E), decompose R2 into R21(A, C) R22(C, D, E), and further decompose R22 into R221(C, D) and R222(C, E)**

**评分细则：**

**由于根据不同模式分出来的步骤可能不一，但是由于最终关系为{C->A,C->B,C->D}，所以最终结果一定是诸如{A,C}{B,C}{C,D}{D,E}等二元组，根据实际情况没分彻底的如(B,C,D)每个 0.5 分，分彻底的 1 个 1 分，分错但是结果正确的酌情给 2-3 分。**

**3) The decomposition is dependency preserving.**

**评分细则：**

**(2)中答案正确并且此处正确的给 4 分，**
**(2)中答案错误并且根据实际拆分情况，若判断一致此处给 3 分**
**(2)中答案错误并且此处正确的给 2 分，（不给出解释扣 1 分）**
**其余情况不给分**

## Problem 4: XML (12 points, 4 points each)

**1)**

**<!DOCTYPE movie_comment[**

```
        <!ELEMENT    movie_comment ( movie*)>
        <!ELEMENT    movie (type, director, comment+)>
        <!ATTLIST          movie title ID #REQUIRED>
        <!ELEMENT    type (#PCDATA)>
        <!ELEMENT    director (#PCDATA)>
        <!ELEMENT    comment (user_name, grade)>
        <!ELEMENT    user_name (#PCDATA)>
        <!ELEMENT    grade (#PCDATA)>
    ]>
```

评分细则：

错 1-2 处扣 1 分，较多错误酌情扣 2-3 分

**2) /movie_comment/movie[type="action" and ./comment/user_name="Alice" and ./comment/grade=5]/@title**

评分细则：

漏一个条件扣 1 分，路径错误扣 1 分

**3) for $p in /movie_comment/movie[director="Yimou Zhang"]**
**where count($p/comment[grade=5])>=1**
**return $p/@title**

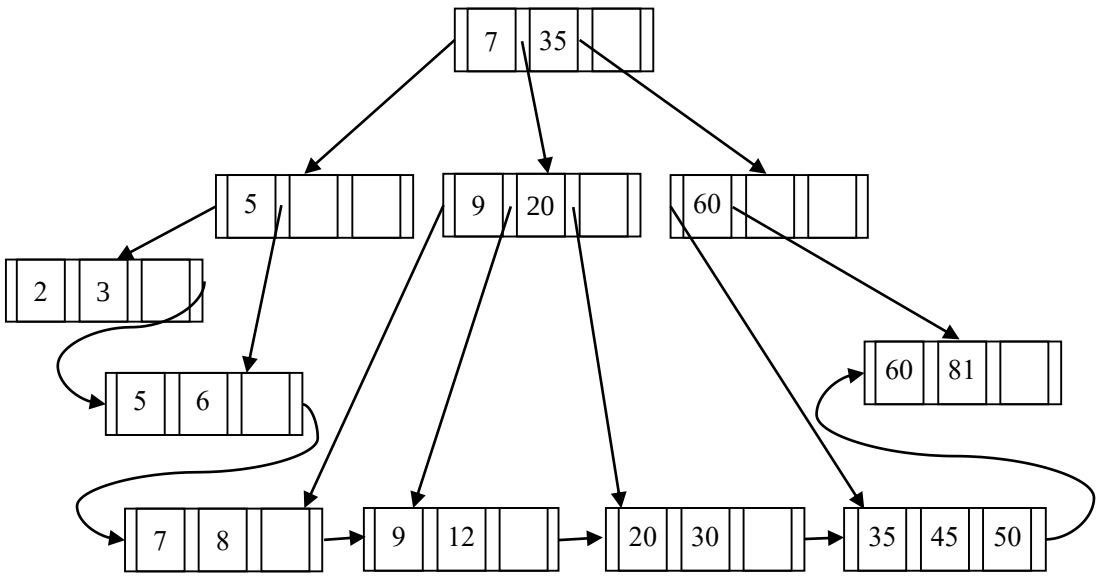
评分细则：

少一个条件扣一分，逻辑错误扣 2-3 分

## Problem 5: B$^+$-Tree (12 points, 3 points each)

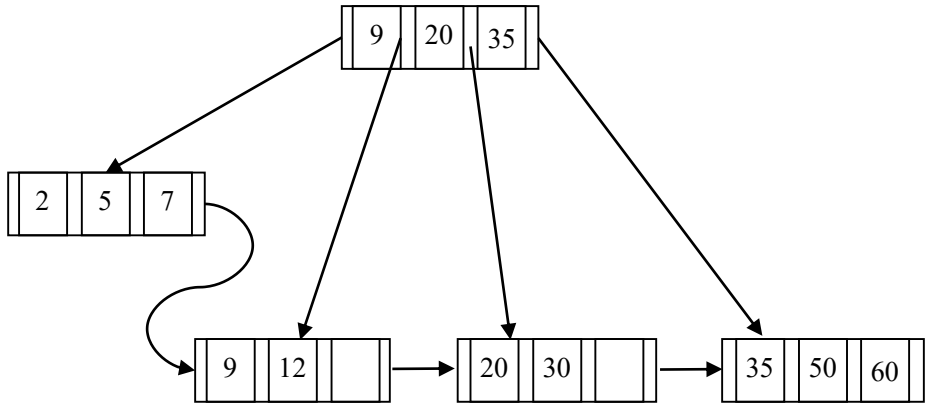
**1)**

**After inserting 8, 6 and 3:**



评分细则：
每个 value 错扣 1 分，最多扣 2 分
叶子正确最少得 1 分

**2)**
**After deleting 81 and 45:**




评分细则：
索引 merge 错误扣 1 分，插入叶子错误扣 1 分
叶子正确最少得 1 分

**3) Maximal number of key values: 4*4*4*4*3=768**
**Minimal number of key values: 2*2*2*2*2=32**
评分细则：
公式列正确即给分

**4) (3 + 1) + 1 = 5 或 (3 + 1) + 2 = 6**
评分细则：
错，扣 3 分

## Problem 6: Query Processing (12 points, 4 points each)

**1) 5,000/500/5 = 2**
评分细则：
2,4 均可，没有计算扣 2 分
400 扣 1 分，10 扣 1 分

**2) Number of blocks of movie is 5000/50=100**

**Number of blocks of comment is 1,000,000/100=10,000**

**Since the equi-join attribute title forms a key on inner relation, we can stop inner loop on the first match.**

**Assign 10 blocks to comments, 1 block to movies, and 1 block for output.**

**Number of block accesses: (10000/10)*100+10000 = 110000 或**

**10000 * 100/10 + 100 = 100100**

**Number of seeks: 2*10000/10=2000**

评分细则：

**Movie，comment 各 1 分**

**Block 110000,100100 均给 1 分**

**Seeks 2000,20 均给分**

有公式答案错酌情扣分

**3) Minimal height = $\log_{60}(5000)$ → 3**（向上取整）

**Max height = $\log_{30}(5000)$ →3**（向上取整）

**So, the height of the $B^+$-tree index on movie(title) is 3.**

**Number of block accesses: 10000+1000000/500*3+1**

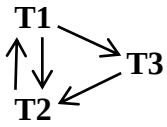
**Number of seeks: 10000+1000000/500*3+1**

评分细则：

不是白卷且答案合理均给分

## Problem 7: Concurrency Control (12 points, 4 points each)

**1)**




**The schedule is not serializable, because there are cycles in the graph.**

评分细则：

少一个依赖扣 1 分，

如前趋图错，若冲突串行化与画出图一致，也给全分

**2) The schedule is not cascadeless.**

评分细则：

结论错，论述正确得 2 分

结论对，论述错得 3 分

结论对，论述正确得 4 分

其余不给分

**3) No. This is because the schedule in 1) exists cycles.**

评分细则：
结论错，论述正确得 2 分
结论对，论述错，酌情得 2-3 分
结论对，论述正确得 4 分
其余不给分

## Problem 8: Aries Recovery Method (15 points, 3 points each)

**1) 1002**

评分细则：
多答扣 1-3 分

**2) 1010**

评分细则：
多答扣 1-3 分

**3) T4**

评分细则：
（T4,1013）也给分，其余不给分

**4) “102.1” = 62, “102.2” = 73**

评分细则：
错一个扣 1 分，错 2 个扣完
多一个扣 1 分，多 2 个不给分

**5)**

**1015: <T4, 102.1, 62>**
**1016: <T4, abort>**
评分细则：
见（4）