1. Single-choice Questions (24 points, 3 points*8)

Answer

A A B D D A A C

2. True or False Questions (12 points, 2 points *6)

Answer

T F F F T T

3. SQL query (16 points)

(1) **select distinct group_name**

from user

where user.gender = "male";

语法/语义错写、漏写每个扣一分，如没写male条件、distinct没写

(2) **select user.user_id, name**

from user, followship

where user.user_id = followship.follower_id and followship.user_id = "1001" and user.age > 18;

4分，where中每个查询条件一分，select和from子句写对一分，

select中user_id没指出表名扣一分；

where中join条件写错扣一分；

(3) 语法/语义错写、漏写每个扣一分，如没写female条件、group by条件写错、直接写max函数（而不是写在select里面）、limit 1扣一分

select group_name

from user

where user.gender = ‘female’

```
group by group_name
having count(user_id) >= all (
    select count(user_id)
    from user
    where user.gender = ‘female’
    group by group_name);
```

或

```
select group_name
from user
where user.gender = ‘female’
group by group_name
order by count(user_id) desc
limit 1;
```

(这种答案算部分对，因为只输出了一个结果，实际可能有多个，去年这种算对的)

或使用with 子句

```
with T(group_name, female_count) as (select group_name, count(user_id)
    from user
    where user.gender = ‘female’
    group by group_name)
select group_name
from T
where T.female_count = (select max(female_count) from T);
```

(4)

语法/语义错写、漏写每个扣一分，如没写game条件、group by条件写错、join条件写错、直接写avg函数（而不是写在select里面）、limit 1扣一分

```
select user.user_id, name, follower_count
from user, (select user.user_id, count(follower_id) as follower_count
            from user, followship
```

```
               where user.user_id = followship.user_id and user.group_name = ‘game’
               group by user.user_id) as T
where user.user_id = T.user_id and T.follower_count > (select avg(follower_count) from T);
```

with 子句类型

```
with T(user_id, follower_count) as (select user.user_id, count(follower_id)
               from user, followship
               where user.user_id = followship.user_id and user.group_name = ‘game’
               group by user.user_id)
with R(avg_follower_count) as (select avg(follwer_count) from T)
select user.user_id, name, follower_count
from user, T, R
where user.user_id = T.user_id and T.follower_count > R. avg_follower_count;
```

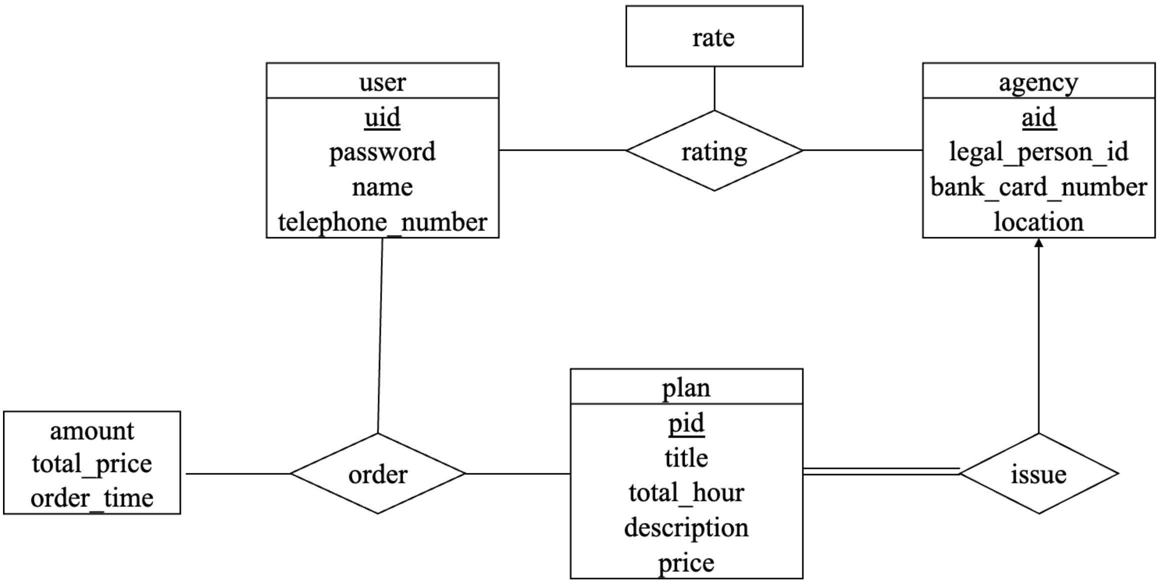
另有一解：

```
select user_id
from user natural join followship
where group_name = ‘game’
group by user_id
having count(follower_id) >= ((
     select count(follower-id)
     from user natural join followship
     where user.group_name = ‘game’)    /
(select count(*)
  from user
  where group_name =’game’));
```

4. Database design (16 points)

Answer

(1)



评分细则：
每个实体（user、agency、plan）、关系（rating 的多对多关系、order 的多对多关系、issue 的多对多关系）、rating 和 order 关系衍生出的属性各一分，共 8 分；
每少写/错写减一分。
关系的名称随意，但是注意关系的类型和箭头；
注意 order 和 rating 每个多对多关系可以写成两个一对多关系，两个一对多关系需要注意箭头方向正确（指向一的实体）。
(2)
The relational schemas are as follows.
user(uid, password, name, telephone_number)
agency(aid, legal_person_ID, bank_card_number, location)
travel_plan(pid, title, total_hour, description, price, agency_ID, issue_time)
order(oid, user_ID, plan_ID, amount, total_price, order_time)
rating(user_ID, agency_ID, rate)

The primary key of each table is underlined. (rating 表如果有单独主键也给分，外键写明白即可)
The foreign keys are as follows:

- the agency_ID of the travel plan table references the aid of the agency table
- the user_ID of the order table references the uid of the user table
- the plan_ID of the order table references the pid of the travel plan table
- the user_ID of the user table references the uid of the user table
- the agency_ID of the rating table references the aid of agency table

评分细则：
一共 8 分

- user 和 agency 表正确，各一分
- travel_plan 表正确，主键和属性一分，外键一分；

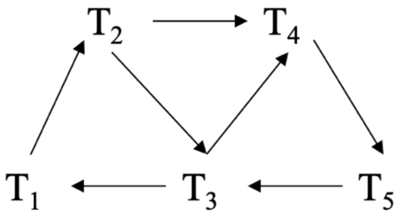
- order 表正确，主键和属性一分，外键一分；
- rating 表正确，主键和属性一分，外键一分；

没有指出外键属性-3 分；直接写 create 语句而不是 schema 扣 1 分。

5. Concurrency Control (8 points)

Answer

(1)



评分细则：图中的边多/少一条扣 1 分，全对给 3 分

(2) S is not conflict serializable，because there are cycles in the graph :

$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_1$

$T_4 \rightarrow T_5 \rightarrow T_3 \rightarrow T_4$

$T_1 \rightarrow T_2 \rightarrow T_4 \rightarrow T_5 \rightarrow T_3 \rightarrow T_1$

评分细则：答案错误扣 3 分，指出一个环算全对，全对给 3 分；没指出环扣 1 分。

(3) No，because every schedule generated by 2PL is serializable. :
评分细则：答案错误扣 2 分，理由错误扣一分，全对给 2 分

6. ARIES Recovery. (12 points)

Answer:

(1) **(5235, 8010, 8010)**. Log 8010 modifies page 5235, which should be added to the DirtyPageTable.

评分标准：2 分。

(2) **8003**, which is the smallest RecLSN in the DirtyPageTable.

评分标准：2 分。

(3) **<T4, 5235.1, 60>, <T2, 3462.1, 40>, <T2, 3462.1, 30>.**

The active transaction table after the analysis pass contains T2 and T4. The related update records should be redone.

评分标准：4 分，写出 3 个各得 1 分；<T2, 3462.1, 40>在<T2, 3462.1, 30>前面得 1

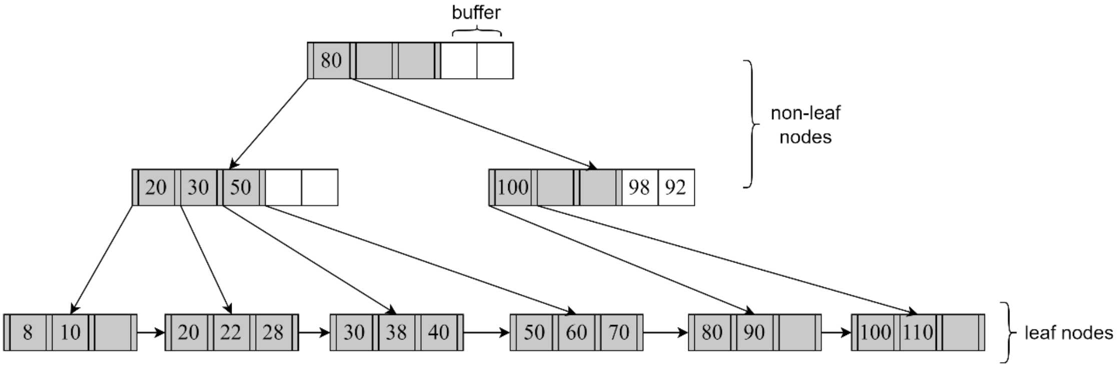
分；每多写一个扣 1 分。

(4) **30, 80**. Location 3462.1 is modified by T2, which is undone. Location 6421.1 is modified by T1, which is committed.
评分标准：4 分，1 个 2 分。

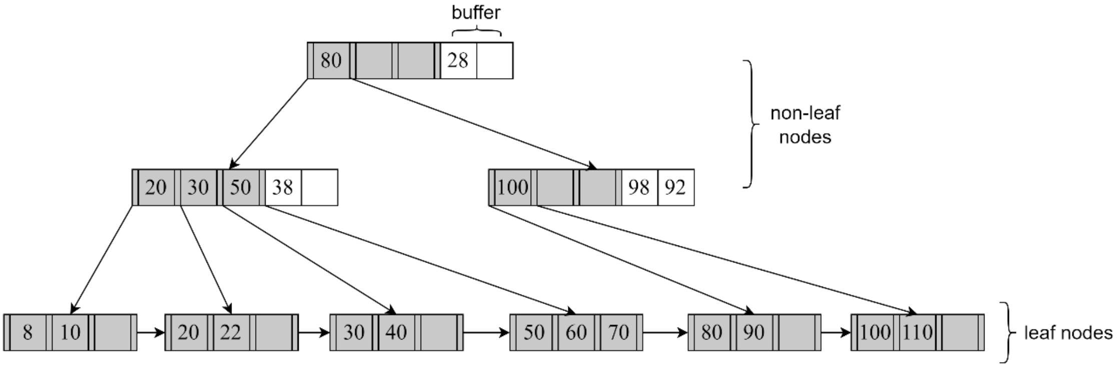
7. Buffer Tree. (12 points)

Answer:

(1) **4.** First, an equality search on 30 requires accessing the root node, the leftmost node on the second level, and the second leaf node on the left. Then, one additional leaf node needs to be searched to collect all items from 30 to 50.
评分标准：数量答对得 1 分，解释合理给 2 分（只要体现出 range query 的查询思路，就可以得分）。
(2)
答案一：按照题目要求，如果 buffer 溢出，将所有元素下推。



答案二：也可以只下推前两个元素，将剩下的元素放进原来的 buffer 里。



评分标准：6 分。根节点错扣 1 分，第二层两个节点错一个扣 1 分，第三层左边三个节点错一个扣 1 分，右边三个节点错任何一个扣 1 分，扣完为止。第二层右边的节点 buffer 如果写成 92 98，算对。两种答案，以令本大题得分较高的为准。

(3) 若(2)为答案一，则为 **0, 0, 5**；若(2)为答案二，则为 **0, 0, 4**。

解析：

插入	描述
38	只插到根节点 buffer
92	只插到根节点 buffer
28	需要修改第二层两个 buffer、左边三个（或者两个）叶子节点，第二层左边节点的指针，一共是 5 个（或者 4 个）节点

评分标准：3 分，1 个 1 分。