**Answers of Problem 1(每小题3分)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **(1)** | **(2)** | **(3)** | **(4)** | **(5)** | **(6)** | **(7)** | **(8)** | **(9)** | **(10)** | **(11)** | **(12)** |
| **B** | **A** | **D** | **D** | **A** | **C** | **D** | **B** | **C** | **A** | **B** | **C** |

***第（9）小题答A给2分***

**Answers of problem 2(每小题4分)**

***（每错一处语法或语义，扣1分）***

1)

select \* from course

where department=’CS’ and

credits =( select max(credits) from course where departmet=’CS’);

或：

select \* from course

where department=’CS’ and

credits >=all ( select credits from course where departmet=’CS’);

或：

select \* from course

where department=’CS’

order by credits desc

limit 1;

2)

select department

from course

group by department

having sum(credits)>=all( select sum(credits) from course group by department);

或：

select department

from course

group by department

order by sum(credits) desc

limit 1;

3)

select title

from course C1

where exists (select \* from course C2 where C2.title=C1.title and C2.course-id !=

C1.course-id)

或:

select C1.title

from course C1, course C2

where C1.title=C2.title and C1.course-id!= C2.course-id;

或

select C1.course-id, C2.course-id, C1.title

from course C1，course C2

where C1.title=C2.title and C1.course-id!= C2.course-id;

或:

select title

from course

group by title

having count(\*)>1;

4)

select course.course-id, count(\*)

from course, prereq

where course.course-id= prereq. prereq-id or course.course-id not in

(select prereq-id from prerequisite )

group by course.course-id

或：

select course.course-id, count(prereq-id) // 或count（prereq.course-id）

from course left outer join prereq on (course.course-id= prereq. prereq-id)

group by course.course-id

**Answers of problem 3.**

（1）8分

***（每个Entity或Relationship或Mapping Cardinality错扣一分）***

conference

cid

title

date

city

rating

comments

review

paper-sub

paper-id

paper-title

abstract

date

file-name

status

c-s

user

write

reviewer

level

author

{subject}

uid

password

name

email

rank

email

affiliation

或：

***reviewer和author不单列，它们的属性level和 {subject}合并到user中，也对。***

conference

cid

title

date

city

rating

comments

review

paper-sub

paper-id

paper-title

abstract

date

file-name

status

c-s

user

write

uid

password

name

email

level

{subject}

rank

email

affiliation

（2）8分，

***（每个 relation错扣1分）***

user(uid, password, name , email)

\*author(uid)

author-subject(uid, subject)

reviewer(uid, level)

conference(cid, title, date ,city)

paper-submission(pid, title, abstract ,file-name, status, date, cid)

write(uid, pid, rank, affiliation, email)

review(pid,uid, rating, comments)

🡪

user(uid, password, name , email, level)

author-subject(uid, subject)

conference(cid, title, date ,city)

paper-submission(pid, title, abstract ,file-name, status, date, cid)

write(uid, pid, rank, affiliation, email)

review(pid,uid, rating, comments)

**Answers of problem 4:**

***给出正确的调度得4分，***

***说明非2PL(如画出precedence graph 、转化成串行调度)得2分***

***说明串行调度（如给出一个等价的串行调度），得2分。***

***用树形协议的调度例子来说明也对。***

**下面是调度的一个例子。**

|  |  |  |
| --- | --- | --- |
| **T1** | **T2** | **T3** |
| **lock-x(A)** |  |  |
| **write(A)** |  |  |
| **unlock(A)** |  |  |
| **lock-s(B)**  **read（B）** | **lock-x(A)**  **write(A)**  **unlock(A)** |  |
| **Unlock(B)** |  |  |
|  |  | **lock-x(B)**  **write(b)**  **unlock(B)** |
|  |  |  |

**T1 doesn’t follow 2PL.**

**The precedence graph is following**

T1

T3

T3

**T1 doesn’t follow 2PL. But the schedule is conflict serializable.**

**The serial order is T1🡪T2🡪T3 , or T1🡪T3🡪T2**

**Answers of problem 5:（每小题3分）**

**1)**

|  |  |  |
| --- | --- | --- |
| **PageID** | **PageLSN** | **RecLSN** |
| 5001 | 2009 | 2003 |
| 5002 | 2013 | 2006 |
| 5003 | 2014 | 2014 |

***（错一行扣1分）***

1. **T4**
2. **5001.2： 222**

**5002.2： 666**

***（错一个扣1分，错两个扣3分）***

1. **2016： <T4, 5003.1, 77>**

**2017： <T4, 5001.2, 222 >**

**2018： <T4, abort >**

***（错一个扣1分）***

**Answers of problem 6（每小题4分）**

1. 13 block transfers + 1 seek

***（transfers 、seek错各扣2分）***

1. (0+3+6)/3=3, 3 block transfers + 3 seeks

***（transfers 、seeks错各扣2分）***

***分别说明三种情况的代价估计也全对。***

***但是三种情况的代价简单求和扣1分***

1. The size (number of nodes) of is (30/2)+15/3+5/3+1 = 15+5+2+1=23***（ 1分）***

9 block reads + 1 seek for reading . ***（1分）***

9 block reads + 1 seek for reading ***（1分）***

23 block writes +1 seek for writifg  ***（1分）***

Total cost to construct is 41 block transfers and 3 seeks.