1. Multiple Choice Questions (single answer, 12*3 points)

(1)

For the table **student(<u>id</u>, name, age, gender, dept_name)**,

and the following query：

```
select  s1.id, s2.id
from student s1, student s2
where s1.dept_name=s2.dept_name and s1.age<18 and
        s2.age>28,
```

What algebra expression is ***not equivalent*** to above query?

(A) $\Pi_{s1.id,\ s2.id}(\sigma_{s1.dept\_name=s2.dept\_name \land s1.age<18 \land s2.age>28}(\rho_{s1}(student) \times \rho_{s2}(student)))$

(B) $\Pi_{s1.id,\ s2.id}(\sigma_{s1.age<18 \land s2.age>28}(\rho_{s1}(student) \bowtie \rho_{s2}(student)))$

(C) $\Pi_{s1.id,\ s2.id}(\sigma_{s1.dept\_name=s2.dept\_name}(\sigma_{age<18}(\rho_{s1}(student)) \times \sigma_{age>28}(\rho_{s2}(student))))$

(D) $\Pi_{s1.id,\ s2.id}(\sigma_{s1.dept\_name=s2.dept\_name}((\sigma_{age<18}(\rho_{s1}(student))) \times \Pi_{id,\ dept\_name}(\sigma_{age>28}(\rho_{s2}(student)))))$

(2)

The table **r (<u>pid</u>, qid)** is defined as following:

```
create table r
 (      pid  char(2)   primary key,
        qid  char(2),
        foreign key (qid)  references r
                  on delete cascade
                  on update cascade );
```

An instance of ***r*** is as following：

pid	qid
11	
22	11
33	22
44	22
55	11

Executing the following SQL statements one by one:

a) **insert into r values (‘11’,’55’);**
b) **delete from r where pid=‘22’;**
c) **update r set pid= ‘22’ where pid =’11’;**
d) **select count(*) from r where qid=’22’;**

What is the result of the statement **d**） above ?

(A) 1　　（B）2　　（C）3　　（D）4

(3)

The functional dependency set **F={A→B, B→C, A→CD, BD→C }** holds on the relation **R(A, B, C, D, E)**.

What is the Canonical Cover of F?

(A) F={A→BCD , B→C, BD→C }
(B) F={A→BCD , B→C}
(C) F={A→B , A→D, B→C}
(D) F={A→BD , B→C}

(4)

The functional dependency set **F={A→B, B→CD}** holds on the relation **R(A, B, C, D, E)**.

What decomposition of R is ***not*** dependency preserving?

(A) R1(B,C,D) ,R2(A,B), R3(A,E)
(B) R1(A,B), R2(A,E), R3(B,C,D)
(C) R1(B,C), R2(B,D), R3(A,B),R4(A,E)
(D) R1(A,B), R2(A,C,D), R3(A,E)

(5) Which statement about data storage is ***incorrect***?

(A) **LRU** is the most suitable replacement strategy for buffer manager in any cases.

(B) For **heap file organization**, records can be placed anywhere in the file where there is free space.

(C) If the needed block is not in the buffer, the **buffer manager** will replace some other block, if required, to make space for the new block.

(D) The **system catalog** of a database stores metadata, such as Information about relations.

(6) What is ***not*** the benefit of **column-oriented storage**?

(A) **Reduced IO** if only some attributes of a relation are accessed

(B) **Improved CPU cache** performance when a query processor fetches the contents of a particular attribute

(C) **Reduced cost** of tuple deletion and update

(D) **Improved compression** of data of the same attribute

(7) Which statement about **index** is ***incorrect***?

(A) **Indexing mechanisms** are used to speed up access to desired data.

(B) **Range query** returns records with an attribute value falling in a specified range of values.

(C) In a **dense index**, index record appears for every search-key value in the file.

(D) **Secondary index** is an index whose search key specifies an order same as the sequential order of the file.

(8)

Assuming the height of the B+-tree index on the table **student(ID)** is 4. Considering the query: **select * from student where ID='2020160008',** the estimated cost for evaluating above query using the **Index Scan algorithm** is:

(A) 4 block transfers + 4 seeks.

(B) 5 block transfers + 5 seeks.

(C) 3 block transfers + 3 seeks.

(D) 5 block transfers + 4 seeks.

(9)

Assuming the table **r** has **160 blocks,** and buffer memory size is **10 blocks**. In the process of sorting **r** with the **External Merge Sort** algorithm, **2 buffer blocks** are allocated to each input run and to the output run, and t**he sorted result of the final pass is *written back* to disk.** The estimated cost for sorting **r** is:

(A) 800 block transfers + 272 seeks.

(B) 800 block transfers +512 seeks.

(C) 960 block transfers + 352 seeks.

(D) 320 block transfers +2 seeks.

(10)

Assuming the table **r** and **s** has **256** and **1024** blocks respectively. Each block has **4K bytes**. To do natural join **r** and **s** with the **Hash Join** algorithm, what is the **approximate minimum memory** needed to avoid recursive partitioning?

(A) 64K bytes （B）128K bytes (C) 1M bytes (D) 2M bytes

(11)

What equivalence rule of the relational algebra is ***incorrect***?

(A) $\sigma_{\theta1}(\sigma_{\theta2}(E)) \equiv \sigma_{\theta2}(\sigma_{\theta1}(E))$

(B) $\sigma_{\theta}(E_1 \cup E_2) \equiv \sigma_{\theta}(E_1) \cup E_2$

(C) $(E_1 \bowtie E_2) \bowtie E_3 \equiv E_1 \bowtie (E_2 \bowtie E_3)$

(D) $(E1 \cup E2) \bowtie E3 \equiv (E1 \bowtie E3) \cup (E2 \bowtie E3)$

(12)

There are following assumptions about the table **instructor** and **teaches**:

- The number of records in **instructor** is **4000**.
- The number of records in **teaches** is **8000**
- The number of distinct values of **dept_name** in **instructor** is **20**
- The value of **salary** in **instructor** is between **10000** and **90000**.

Please estimate the size returned by the following query:

select * from instructor natural join teaches on ID
where dept_name='CS and salary >=70000;

(A) 25 （B）50 （C）100 （D）200

2. SQL query(16 points，4 points per part)

Consider the following relational schema in a university:

course (<u>course-id</u>, title, department, credits)
prereq(<u>course-id, prereq-id</u>)

Write SQL statements to answer following queries:

(1) What course in CS department has the most credits?

(2) What department offers the most total credits of courses?

(3) Are there any courses with the same title?

(4) Calculate the number of times of each course being the prerequisites for other courses.

3. **Database Design(16 points)**

Every year scientist submit papers to various academic conferences. After peer review, a part of papers could be accepted and posted at the conferences. A database for a simplified Conference Management System (CMS) must be able to keep track of conferences, paper submissions and paper reviews.

- CMS maintains profile for each **user**: ID, password, name, and email.
- Users have multiple roles: author and/or reviewer, etc.
- An **author** includes a set of research subjects. A r**eviewer** includes an experience level.
- A **conference** includes conference ID, title, date, and city.
- A paper could be submitted to only one conference. Information of **paper submission** includes paper ID, title, abstract, submit date， status (i.e. accept, reject), and a file name of paper full text.
- Every author of a paper submission must contain rank, affiliation, and email.
- A paper submission could be reviewed by several reviewers. Information of **review** includes rating and comments.

(1) Please draw the E-R diagram for CMS database.

(2) Transform the E-R diagram into relational schemas.

4. **Two-phase locking Protocol(8 points)**

Please give a schedule with three transactions and at least a pair of conflict operations, to illustrate that the two-phase locking protocol is not a necessary condition for conflict serializability. The schedule should include **read**, **write**, **lock**, and **unlock** operations.

5. **Aries Recovery Method (12 points, 3 points per part)**
   A DBMS uses Aries algorithm for system recovery. Following is a figure of a log file after system crashes. The log file consists of 15 log records with LSN from 2001 to 2015. Assuming that the last completed checkpoint is the log record with LSN 2011.
   (1) When the Analysis Pass is complete, what is the contents of the DirtyPageTable?
   (2) Which transactions should be undone?
   (3) After recovery, what values are the data items identified by “5001.2” and “5002.2”？
   (4) After recovery, what additional log records appended to the log fie?

2001： <T1 begin>

2002： <T2 begin>

2003： <T1 , 5001.1, 11, 111>

2004： <T2 , 5001.2, 22, 222>

2005： <T3 begin>

2006： <T3, 5002.1, 33, 333 >

2007： <T4 begin>

2008： <T2 commit>

2009： <T4, 5001.2, 222, 444 >

2010： <T1, 5002.2, 55, 555>

2011：

Txn	LastLSN
T1	2010
T3	2006
T4	2009

PageID	PageLSN	RecLSN
5001	2009	2003
5002	2010	2006

2012： <T1 commit>

2013： <T3, 5002.2, 555, 666>

2014： <T4, 5003.1, 77, 777>

2015： <T3 commit>

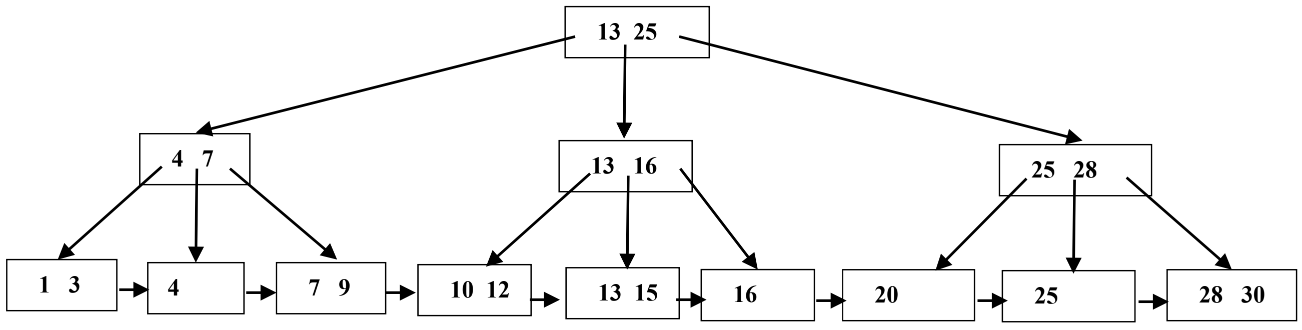
6. LSM-Tree index (12 points, 4 points per part)

Following is an example of a variant of the LSM tree which has multiple trees at each level. L0 is a B+-tree index at main memory. When L0 tree reaches its maximum size (i.e.13 blocks), it is written to disk as L_0^1 at level 0 using only sequential I/O operations. L_0^2 is another B+-tree index that has been written to disk at level 0 using only sequential I/O operations. When the number of B+-tree indexes at level 0 reaches its upper limit (i.e. 2), all the trees L_0^1 and L_0^2 are merged together into one combined new B+-tree at disk as L_1^1 at next level 1: the leaves of L_0^1 and L_0^2 are read sequentially, and the keys merged in sorted order, and the B+-tree L_1^1 at level 1 is constructed using standard techniques for bottom-up construction of B+-trees. For the sake of the exam, we assume the followings: (a) fan-out (n) of B+-tree is 3, (b) node size of B+-tree is the same as block size, (c) buffer space is big enough to load all leaf nodes of L_0^1 and L_0^2 simultaneously .

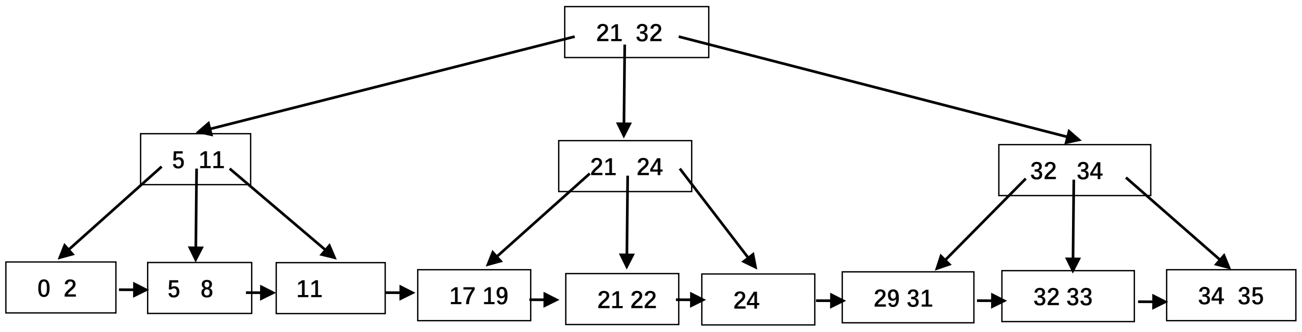
(1) Please estimate block transfers and seeks that are needed to write L_0^1.

(2) Please estimate block transfers and seeks that are needed to lookup an index entry at leaf nodes by first searching the B+-tree at main memory and then searching the B+- trees at the level 0(i.e. L_0^1 and L_0^2) .

(3) Please estimate the block transfers and seeks that are needed to construct L_1^1 by merging L_0^1 and L_0^2。



L0 : B+-tree at main memory
L_0^1 : B+-tree written to disk (level 0)



L_0^2: B+-tree written to disk (level 0)

Answers of problem 1：

Please fill the right choice into the blank for each question.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)

Answers of problem 2：

Answers of problem 3：

Answers of problem 4：

Answers of problem 5：

Answers of problem 6：