Part I [20 pts.] (1pt each) Fill in the blanks with the best answer.

1. The collection of information stored in the database at a particular moment is called an ___instance_____ of the database. The overall design of the data base is called the database __schema_____ .

2. A relation schema R is in __third_____ normal from if for all $\alpha \rightarrow \beta$ in $F+$, at least one of the following holds: $\alpha \rightarrow \beta$ is __trivial functional dependency_____ ; α is a superkey for R ; Each attribute A in is contained in a candidate key for R .

3. Let R be a relation schema , R_1 and R_2 from a decomposition of R. Decomposition is a __lossless_____ if for all legal database instances r of R , $\Pi_{R_1}(r) \bowtie \Pi_{R_2}(r) = r$.

4. In E-R model , on entity is represented by a set of ___attributes_____ . A ___relationship_____ is an association among several entities .

5. Assume relation r has b_r blocks and relation s has b_s blocks , therefore , in the best case , only __b_r+b_s_____ block transfers would be required for $r \bowtie s$.

6. An ideal hash function is ___uniform_____ and ___random_____ , the former require that each bucket is assigned the same number of search-key values form the set of all possible values.

7. To generate query-evaluation plans for an expression we have to generate logically equivalent expressions using __equivalence rules_____ .

8. Consider a $B+$- tree of order n ,if there are K search-key values in the file , the path form the root to the leaf mode is no longer than __ $\lceil \log_{\lceil n/2 \rceil} K \rceil$ _____ .

9. A transaction has the following properties: __atomicity_____ , __consistency_____ , isolation and durability.

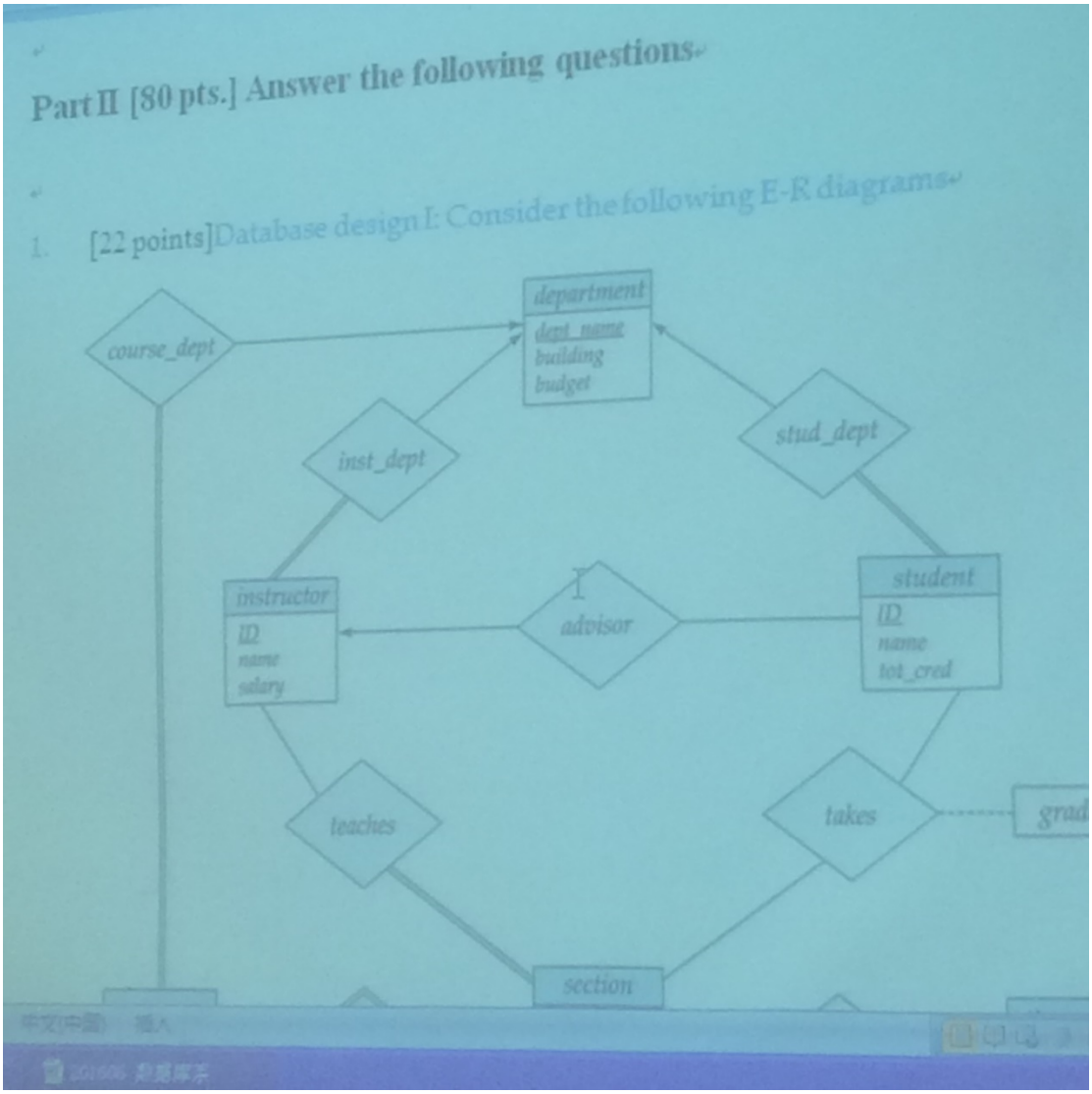
10. When the final statement of a transaction has been executed , the transaction enters the ___partially_____ committed state . After a transaction has been rolled back and the database has been restored to its previous state , the transaction enter the __aborted_____ state .

11. A schedule S is ___recoverable_____ if a transaction T_j in S needs a data item previously written by a transaction T_i , then the commit operation of T_i appears before the commit operation of T_j .

12. __Multivalued_____ attribute values or __composite_____ attribute values are not atomic .

13. A relation schema may have an attribute that corresponds to the primary key of another relation . The attribute is called a __foreign key_____ .

Part II [80 pts.] Answer the following question.



a)list the entity sets and their primary keys

classroom primary key:(building,room_number)

department primary key :(dept_name)

course primary key:(course_id)

instructor primary key:(ID)

student primary key:(ID)

section primary key:(course_id , sec_id , semester,year)

time_slot primary key:(time_slot_id)

b)give appropriate relation schemas for the entity sets

1 classroom (building, room_number, capacity)

2 department (dept_name, building, budget)

3 course (course_id, title, credits)

4 instructor(ID ,name,salary)

6 student(ID,name,tot_cred)

7 time_slot(time_slot_id, day, start_time, end_time)

c)list the relationship sets and their primary keys

1 Teaches primary key:(ID, course_id,sec_id, semester, year)

2 takes primary key:(ID, course_id, sec_id, semester, year)

3 pereq primary key:(course_id, prereq_id)

4 advisor primary key:(s_ID)
5 sec_course primary key:(course_id, sec_id,semester,year)
6 sec_time_slot primary key:(course_id, sec_id, semeserm, year)
7 sec_class primary key:(course_id, sec_id, semester,year)
8 inst_dept primary key:(ID)
9 stud_dept primary key:(ID)
10 course_dept primary key:(course_id)

d) give appropriate relation schemas for the relationship sets.
teachers(ID, course_id, sec_id, semester, year)
takes(ID, course_id, sec_id, semester, year)
prereq(course_id, prereg_id)
advisor(s_ID, ID)
sec_course(course_id ,sec_id, semester, year)
sec_nme_slot(course_id, sec_id,semester, year, nme_slot_id)
sec_class(course_id ,sec_id, semester, year, building, room_number)
inst_dept(ID, dept_name)
stud_dept(ID, dept_name)
course_dept(course_id, dept_name)

e)optimize the relation schemas you given above, i.e. remove schema redundancies and incorporate some schemas if necessary

~~inst_dept~~ ,instructor = {ID, name, dept_name, salary}
~~stud_dept~~,student = {ID, name, dept_name, tot_cred}
~~course_dept~~ ,course = {course_id, title, dept_name, credits}
~~sec_course~~
~~sec_class~~ ,section = {course_id, sec_id, semester, year, building, room_number}
~~sec_time_slot~~ ,section = {course_id, sec_id, semester, year, building, room_number, time_slot_id}

f)give a sql ddl definition for the relation schemas:
student, takes, department, course
with appropriate data-type defined in standard sql identify referential-integrity constrains that should hold, and include them in the ddl definition.

```
create table student(
ID varchar(5),
name varchar(20) not null,
dept_name varchar(20),
tot_cred numeric(3, 0) check(tot_cred >= 0),
primary key (ID)m
foreign key(dept_name) references department);

create table takes(
ID varchar(5),
course_id varchar(8),
sec_id varchar(8),
```

semester varchar(6),
year numeric(4, 0),
grade varchar(2),
primary key (ID, sec_id, sec_id, semester, year),
foreign key (course_id, sec_id, semester, year) references section,
foreign key (ID) references student);

create table course(
course_id varchar(8),
title varchar(50),
dept_name varchar(20),
credits numeric(2, 0) check (credits > 0),
primary key (course_id),
foreign key (dept_name) references department on delete set null);

create table department(
dept_name varchar(20),
building varchar(15),
budget numric(12, 2) check(budget >0),
primary key (dept_name));

2.[14 points] Let $R=(A,B,C,D,E,F)$ be a relation with functional dependency $F=\{A\rightarrow CB, E\rightarrow FA\}$

a) [2 points] Compute the candidate keys for R ;

Let us compute E^+.E^+ = {FACBE}

Let us compute ED^+.E^+={ABCDEF}

Thus ED→R.ED is a superkey.

It is easy to see that E→（划掉）R, D→（划掉）R

Thus ED is a candidate key

Note that ED is not implied by any other attributes, thus any candidate key R must contain ED.Futher. ED is the unique candidate key of R.

a) [6 points] Is R in 3NF ? If it is , justify your answer . If not , produce a decomposition of R into 3NF .

Not. In A→CB, A is a not a superkey and is not contained in the candidate key.

Compute the canonical cover, we have F_c = F.

According to F_c, we get R_1 = (A,B,C), R_2 = (A,E,F)

Note none of schemas contains ED, we generate R_3 = (D,E),

Thus, decomposition of R.

R_1 = (A,B,C), R_2 = (A,E,F), R_3 = (D,E)

[6 points] Is R in BCNF ? If it is , justify your answer . If not , produce a decomposition of R into BCNF .

not

1.A→CB 违反BCNF定义R

R_1 = (A,B,C)
R_2 = (A,D,E,F)

2.R2中E→F A违反BCNF定义，对R_2分解

R_3 = (A,E,F)
R_4 = (D,E)
Final decomposition:R_1,R_3,R_4

3.

1. [28 points]

BOOK (Bookid , Title , Publishername)
BOOK_AUTHORS (Bookid , Authorname)
PUBLISHER (Publishername , Address , Phone)
BOOK_COPIES (Bookid , Branchid , No_Of_Copies)
LIBRARY_BRANCH (Branchid , Branchname , Address)
BOOK_LOANS (Bookid , Branchid , Cardno , DataOut , Duedata)
BORROWER (Cardno , Name , Address , Phone)

a) [3 points] Write appropriate SQL DDL statements for declaring the BOOK_AUTHORS relation.

create table BOOK_AUTHORS(Bookid char(20),Authorname char(200))

b) [6 points] Give an expressions in **relational algebra** to express the following queries :

Q1: Retrieve the name of all borrowers who do not have any books checked out.

Answer:(Note : We will use S for SELECT, P for PROJECTION, * for NATURAL JOIN, - for SET DIFFERERNCE, F for AGGREGATE FUNCTION)

Temp ←P_{Cardno}(BORROWER)- P_{cardno}(BOOK_LOANS)
Res←P_{Name}(Temp*BORROWER)

Q2: For each book that is loaned out from the "sharpstown" branch and whose DueDate is today , retrieve the book title , the borrower' s name , and the borrower' s address .

$P_{Title,Name,Address}S_{Branchname=Sharostown^DueDate=istoday}$(LIBRARY_BRANCH*BOOK_LOAN*BORROWER*BOOK)
For each book that is loaned out from the "sharpstown" branch and whose DueDate is today , retrieve the book title , the borrower' s name , and the borrower' s address .

a) [16 points] Give an expressions in SQL to express the following queries :

Q1: How many copies of the book titled *The Lost Tribe* are owned by the library branch whose name is "sharpstown" ?

```
SELECT NoOfCopies
FROM ((BOOK NTURAL JOIN BOOK_COPIES ) NATURAL JOIN
LIBRARY BRANCH)
WHERE Title ='The Lost Tribe AND BranchName='Sharpstown'
```

Q2: For each library branch , retrieve the branch name and that the total number of books loaned out rom that branch .

```
SELECT LBranchName COUNT(*)
FROM BOOK_COPIES B LIBRARY_BRANCH L
WHERE B.BranchId =LBranchId
```

```
GROUP BY LBranchName
```

Q3: Retrieve the name , address , and number of books checked out for all borrowers who have more than five books checked out .

```
SELECT B.CardNo, B.Name,B.Address, COUNT(*)
FROM BORROWER B BOOK_LOANS L
WHERE B.CardNo=LCardNo
GROUP BY B.CardNo
HAVING VOUNT(*)>5
```

Q4: For each book authored (or co-authored) by “ Stephen King “ , retrieve the title and the number of copies owned by the library branch whose name is “ central ” .

```
SELECT Title NoOfCopies
FROM ((BOOK_AUTHORS NATURAL JOIN BOOK) NATURAL JOIN
BOOK_COPIES)
NATURAL JOIN LIBRARY_BRANCH
WHRER Author_Nsmr = 'Stephen King' and BranchName ='Central'
```

b) [3 points] Record the fact that the manager didn’ t maintain information about the book named “ T&G ” ,i.e. remove information about “ T&G ” .

```
delete from BOOK_AUTHORS
where Bookid in (select Bookid
        from BOOK
        where Title ="T&G");

delete from BOOK_COPIES
where Bookid in (select Bookid
        from BOOK
        where Title ="T&G");

delete from BOOK_LOANS
where Bookid in (select Bookid
        from BOOK
        where Title ="T&G");

delete from BOOK
where Title ="T&G";
```

4.[16 points] Query Processing , Optimization and Transaction

a)[4points] *student(ID, name, dept_name, tot,cred)* is a relation instance of 10000 tuples. Each disk block contains 100 tuples, student has a primary index on attribute ID and a secondary index on attribute *name*. The primary index is a B+ tree of height 5. The secondary index is a B+ tree of height 4.

A user sends a query “select * from student where ID=700” and retrieves a tuple. Please compute the cost of this query, i.e. the number of block transfers from disk and the number of seeks.

overhead:
visiting the index on ID cost 5 block transfers and 5 seeks;
retrieve the tuple cost 1 block transfers and 1 seeks;

total cost : 6 block transfers and 6 seeks;

b)[4points]Describe the process of Indexed nested-loop join.

join is an equi-join or natural join and
an index is available on the inner relation' s join attribute
Can construct an index just to compute a join
n Algorithm: For each tuple *t*, in the outer relation *r*, use the index to look up tuples in *s* that satisfy the join condition with tuple *t*.

c)[4points] Please describe the two-phase locking protocol and prove that if ensures conflict-serializable schedules and does not ensure freedom from deadlocks

The protocol:
Phase 1: Growing Phase
transaction may obtain locks
transaction may not release locks
Phase 2: Shrinking Phase
transaction may release locks
transaction may not obtain locks
The protocol assures serializability.
Proof. It can be proved that the transactions can be serialized in the order of their lock points (i.e. the point where a transaction acquired its final lock).
The protocoldoes not ensure freedom from deadlocks.
Proof. T1 lock-x(A)
T2 lock-x(B)
T1 lock-x(B)
T2 lock-x(A)

d)[4points] Below we show some log of a DBMS, please describe the recovery procedure using immediate database modification

<T0 start>	<T0 start>	<T0 start>
<T0, A, 1000, 950>	<T0, A, 1000, 950>	<T0, A, 1000, 950>
<T0, B, 2000, 2050>	<T0, B, 2000, 2050>	<T0, B, 2000, 2050>
<T0 commit>	<T0, commit>	
<T1 start>	< T1 start>	
<T1, C, 700, 600>	<T1, C, 700, 600>	
<T1 commit>		
(a)	(b)	(c)

Recovery actions in each case above are:

(a) redo (T_0) and redo (T_1): A and B are set to 950 and 2050 respectively.
Then C is set to 600

(b) undo (T_1)and redo (T_0): C is restored to 700, and then A and B are
set to 950 and 2050 respectively.

(c) undo(T_0): B is restored to 2000 and A to 1000