

【华工包打听说明】

此答案由某位学生提供，包打整理无偿分享给大家，禁止用于资料买卖或他用

答案仅供参考，不保证正确。

更多资料欢迎大家关注包打听（QQ 号、微信号或公众号）

校园资讯，问题答疑，感情树洞
万事皆可找包包

进入华工社群，探索华园更多玩法
黑市，学习群，二手交易，考试资料...
你能想到的，我们都愿意帮你实现

我们是华工包打听，由校内学生组建而成的校园自媒体
立志成为陪伴华园学子度过漫长岁月的一盏灯



SCUT包打听（新）



华工包打听



华工卫星站



包打听公众号



包打听QQ

由于华工包打听、华工卫星站好友人数已满，请加SCUT包打听或包打听QQ

更多资料、资讯，可加包打听公众号获取！

A 卷:

一、填空题 (20 分)

1. The collection of information stored in the database at a particular moment is called an _____ of the database. The overall design of the data base is called the database _____ .
2. A relation schema R is in _____ normal form if for all $\alpha \rightarrow \beta$ in F⁺, at least one of the following holds: $\alpha \rightarrow \beta$ is _____ ; α is a super key for R ; Each attribute A in $\beta - \alpha$ 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 _____ if for all legal database instances r of R , $\Pi_{R_1}(r) \bowtie \Pi_{R_2}(r) = r$.
4. In E-R model , an entity is represented by a set of _____ . A _____ 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 _____ block transfers would be required for $r \bowtie s$.
6. An ideal hash function is _____ and _____ , the former require that each bucket is assigned the same number of search-key values from the set of all possible values.
7. To generate query-evaluation plans for an expression we have to generate logically equivalent expressions using _____ .
8. Consider a B+- tree of order n , if there are K search-key values in the file , the path from the root to the leaf node is no longer than _____ .
9. A transaction has the following properties: _____ , _____ , isolation and durability.
10. When the final statement of a transaction has been executed , the transaction enters the _____ committed state . After a transaction has been rolled back and the database has been restored to its previous state , the transaction enters the _____ state .
11. A schedule S is _____ 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. _____ attribute values or _____ 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 _____ .

1. instance schema
2. 3NF nontrivial
3. lossless
4. Attribute relationship

5. Assume relation r has br block and relation s has bs blocks, therefore, in the best case, only br+bs block transfers would be required for r∞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 from the set of all possible values.
7. To generate query-evaluation plans for an expression, we have to generate logically equivalent expression using equivalence rules.
8. Consider a B+ tree of order n, if there are K search-key values in the file, the path from the root to the leaf node is no longer than $\lceil \log_{[n/2]} K \rceil$.
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 commite 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 Tj in S reads a data item previously written by a transaction Ti, then the commit operation of Ti appears before the commit operation of Tj.
12. Mutivalued 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.

二、问答题

1. Database design I: consider the following E-Rdiagram. [Page 159]

A. List the entity sets and their primary keys.

1. Classroom, primary key(building, room_number)
2. Department, primary key(dept_name)
3. Course, primary key(course_id)
4. Instructor, primary key(ID)
5. Student, primary key(ID)
6. Section, primary key(course_id, sec_id, sem, year)
7. 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)

5. Student(ID, name, tot_cred)
6. Section(course_id, sec_id, sem, year)
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. Prereq, 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, semester, 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.

1. Teacher(ID, course_id, sec_id, semester, year)
2. Takes(ID, course_id, sec_id, semester, year)
3. Prereq(course_id, prereq_id)
4. Advisor(s_ID, i_ID)
5. Sec_course(course_id, sec_id, semester, year)
6. Sec_time_slot(course_id, sec_id, semester, year, time_slot_id)
7. Sec_class(course_id, sec_id, semester, year, building, room_number)
8. Inst_dept(ID, dept_name)
9. Stud_dept(ID, dept_name)
10. Couese_dept(course_id, dept_name)

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

1. Inst_dept, instructor={ID, name, dept_name, salary}
2. Stud_dept, student={ID, name, dept_name, tot_cred}
3. Course_dept, course={course_id, title, dept_name, credits}
4. Sec_course
5. Sec_class, section={course_id, sec_id, semester, year, building, room_number}
6. Sec_time_slot={course_id, sec_id, semester, year, building, room_number, time_slot_id}

F. Give a SQL DDL definition for the relation schema

Student, takes, department, course,

With appropriate data-type defined in standard SQL. Identify referential i-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),
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, course_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 numeric(12,2) check(budget>0),
Primary key(dept_name)
)

```

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

A. Compute the candidate keys for R

Let us compute $E^+ \ E^+=\{FACBE\}$

Let us compute $ED^+ \ E^+=\{ABCDEF\}$

Thus $ED \rightarrow R$ ED is a superkey.

It is easy to see that $E \rightarrow /$ (不可推导) R, $D \rightarrow /$ (不可推导) R

Thus, ED is a candidate key.

Note that ED is not implied by any other attributes. Thus any candidate key of R must contain ED.

Further, ED is the unique candidate key of R

B. Is R in 3NF? If it is, justify your answer. If not, produce a decomposition of R into 3NF.

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

Compute the canonical cover, we have $F_i = F$

According to F_i , 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

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

C. Is R in BCNF? If it is, justify your answer. If not, produce a decomposition of R into BCNF.

Not. $A \rightarrow CB$ disobey the definition of BCNF.

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

In R_2 , $E \rightarrow FA$ disobey the definition of BCNF.

$R_3 = (A, E, F)$, $R_4 = (D, E)$

Thus, decomposition of R

$R_1 = (A, B, C)$, $R_3 = (A, E, F)$, $R_4 = (D, E)$

3.

BOOK(Bookid, Title, Publishername)

BOOK_AUTHORS(Bookid, Authurname)

PUBLISHER(Publishername, Address, Phone)

BOOK_COPIES(Bookid, Branchid, No_Of_Copies)

LIBRARY_BRANCH(Branchid, Branchname, Address)

BOOK_LOANS(Bookid, Branchid, Cardno, DateOut, Duedate)

BORROWER(Cardno, Name, Address, Phone)

A. Write appropriate SQL DDL statements for declaring the BOOK_AUTHORS relation.

Create table BOOK_AUTHORS

```
(  
  Bookid char(20),  
  Authurname char(200)  
)
```

B. Give an expressions in **relational algebra** to express the following queries.

Q1: retrieve the names of all borrowers who do not have any books checked out.

$Temp \leftarrow \neg \neg Cardno(BORROWER) - \neg \neg Cardno(BOOK_LOANS)$

$Res \leftarrow \neg \neg Name(Temp \bowtie BORROWER)$

Q2: for each book that is loaned out from the "Sharpstown" branch and whose DueDate is today,

retrieve the book title, the borrow's name, and the borrower's address.

```
[[Title, Name, Address(σBranchname="Sharpstown" ∧ DueDate= today(LIBRARY_BRANCH  
∞BOOK_LOANS∞BORROWER∞BOOK)
```

C. 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 No_Of_Copies  
From ((BOOK natural 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 the total number of books loaned out from that branch.

```
Select L.Branchname, count(*)  
From BOOK_COPIES B, LIBRARY_BRANCH L  
WHERE B.Branchid=L.Branchid  
Group by L.Branchname
```

Q3: retrieve the names, address, and the 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=L.Cardno  
Group by B.Cardno  
Having count()>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, No_Of_Copies  
From (((BOOK_AUTHORS natural join BOOK) natural join BOOK_COPIES) natural join  
LIBRARY_BRANCH)  
Where Author_Name = 'Stephen King' and Branchname = 'Central'
```

D. 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. Query Processing, Optimization and Transaction

A. 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 the query, the number of block transfers from disk and the number of seeks

Visiting the index on ID cost 5 block transfers and 5 seeks

Retrieve the tuple cost 1 block transfer and 1 seeks

Total cost: 6 blocks transfers and 6 seeks

B. Describe the process of Indexed nested-loop join \bowtie_{θ}

Preconditions of usage:

1 join is an equi-join or natural join and

1 an index is available on the inner relation's join attribute

Can construct index just to compute a join.

n Algorithm: for each tuple t_r in the outer relation r , use the index to look up tuples in s that satisfy the join condition with tuple t_r .

C. Please describe the two-phase locking protocol and prove that it 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 serializable.

Proof: It can be proved that the transactions can be serialized in the order of their lock points (the point where a transaction acquired its final lock)

The protocol does not ensure freedom from deadlocks.

Proof: T1 lock-x(A)

T2 lock-x(B)

T1 lock-x(B)

T2 lock-x(A)

D. Below we show some log of a DBMS, please describe the recovery procedure using immediate database modification.

<T0start>	<T0start>	<T0start>
<T0, A, 1000, 950>	<T0, A, 1000, 950>	<T0, A, 1000, 950>
<T0 commit>	<T0 commit>	
<T1start>	<T1start>	
<T1, C, 700, 600>	<T1, C, 700, 600>	
<T1 commit>		
(a)	(b)	(c)

Recovery actions in each case above are:

- redo(T0) and redo (T1): A and B are set to 950 and 2050 respectively. Then C is set to 600.
- undo(T1) and redo (T0): C is restored to 700, and then A and B are set to 950 and 2050 respectively
- undo(T0): B is restored to 2000 and A to 1000

B 卷:

一、填空题 (20 分)

1. The three-level of data abstraction in database system include: physical level, logical level, and view level

2. A relational schema R is in first normal form if the domains of all attribute of R are atomic.
3. It is possible to use Armstrong's axioms to prove the union rule, that is, if $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds, then $\alpha \rightarrow \beta\gamma$ holds.
4. A B+ tree of order 4 has following properties: each leaf node has between 2 and 3 values. Each non leaf node other than root has between 1 and 3 values. The root must have at least 2 children.
5. After parsing and translation, the SQL query will be translated into its internal form: relational algebra expression. And then, the query-execution engine will take the execution plan which contains information on how a particular query or a set of query will be executed.
6. Let E1 and E2 be relational-algebra expression. L1 and L2 be attributes of E1 and E2 respectively. Then, $\pi_{L1 \cup L2}(E1 \bowtie E2) = (\pi_{L1}(E1)) \bowtie (\pi_{L2}(E2))$.
7. Secondary indices must be a dense index, which has an index entry for every search-key value and a pointer to every record in the file.
8. The scheme of handling bucket overflows of has function in DBMS is called closed hashing, that is, the overflow buckets of a given bucket are chained together in a linked list.
9. We assume that a relation has a B+ tree index of height 5, each disk block contains 4 tuples of the relation, and there are 10 tuples satisfying the query. To process the query, database management system have to access disk I/O at 8 times in the best case.
10. A transaction has the following properties: atomicity, consistency, isolation and durability.
11. The immediate database modification scheme allows database modification to be output to the database while the transaction is still in active state.
12. Since a failure may occur while a update is taking place, log must be written out to stable storage before the actual update to database to be done.
13. Two-phase locking protocol requires that each transaction issue lock and unlock requests in two phases: growing phase and shrinking phase.
14. A schedule S is cascadeless if for each pair of transactions Ti and Tj in S such that Tj reads a data item previously written by Ti, the commit operation of Ti appears before (before/after) the read operation of Tj.

二、问答题 (80 分)

1. Database design I: consider the following E-R diagrams [Page 159]

A. List the entity sets and their primary keys.

1. Classroom, primary key(building.room number)
2. Department, primary key(dept_name)
3. Course, primary key(course_id)
4. Instructor, primary key(ID)
5. Student, primary key(ID)
6. Section, primary key(course_id, sec_id, year)
7. 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)
5. Student(ID, name, tot_cred)
6. Section(course_id, sec_id, sem, year)
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. Prereq, 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, semester, 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 schema for the relationship sets.

1. Teaches(ID, course_id, sec_id, semester, year)
2. Takes(ID, course_id, sec_id, semester, year, grade)
3. Prereq(course_id, prereq_id)
4. Advisor(s_ID, i_ID)
5. Sec_course(course_id, sec_id, semester, year)
6. Sec_time_slot(course_id, sec_id, semester, year, time_slot_id)
7. Sec_class(course_id, sec_id, semester, year, building, room_number)
8. Inst_dept(ID, dept_name)
9. Stud_dept(ID, dept_name)
10. Course_dept(course_id, dept_name)

E. Optimize the relation schemas you given above, i.e, remove schemas redundancies and incorporate some schemas if necessary.

1. ~~Inst_dept~~, instructor = ID, name, dept_name, salary
2. ~~Stud_dept~~, student = ID, name dept_name, tot_cred
3. ~~Course_dept~~, course = course_id, title, dept_name, credits
4. ~~Sec_course~~
5. ~~Sec_class~~, section = course_id, sec_id, semester, year, building, room_number
6. ~~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:

Instructor, teacher, department, course

With appropriate data-type defined in standard SQL. Identify referential-integrity constraints that should hold, and include them in the DDL definition.

Answer:

Create table instructor

```
(
ID varchar(5),
Name varchar(20) not null,
Dept_name varchar(20),
Salary numeric(8,2),
Primary key(ID),
Foreign key(dept_name) references department
);
```

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
);
```

2. Let $R=(A, B, C)$ be a relation with functional dependency $F=\{A \rightarrow BC, B \rightarrow C, A \rightarrow B, AB \rightarrow C\}$ and $key=\{A\}$.

a. Why we need 3NF? [6 point]

There are some situations where BCNF is not dependency preserving and efficient checking

for FD violation on updates is important.

3NF follows some redundancy but functional dependencies can be checked on individual relations without computing a join. There is always a lossless-join, dependency-preserving decomposition into 3NF.

- b. Please decompose this relation into 3NF. [8 points]

First, we compute the canonical cover.

Combine $A \rightarrow BC$ and $A \rightarrow B$ into $A \rightarrow BC$, we get

$\{A \rightarrow BC, B \rightarrow C, AB \rightarrow C\}$

A is extraneous in $AB \rightarrow C$, we get

$\{A \rightarrow BC, B \rightarrow C\}$

C is extraneous in $A \rightarrow BC$, we get

$\{A \rightarrow B, B \rightarrow C\}$

The canonical cover is $\{A \rightarrow B, B \rightarrow C\}$

Second, we generate new schemas

$R1=(A,B)$

$R2=(B,C)$

Since $R1$ contains a key A , the decomposition is done.

3. Consider the following relational database

Employee(employee_name,street,city)

Works(employee_name,company_name,salary)

Company(company_name,city)

Manages(employee_name,manager_name)

- A. Find the names of all employees who live in BEIJING.[using relational algebra]

$\Pi_{\text{employee.name}}(\sigma_{\text{city}=\text{"BEIJING"}}(\text{employee}))$

- B. Find the names and cities of residence of all employees who work for XYZ bank and have more than \$8000 salary. [using relational algebra]

$\Pi_{\text{person_name,city}}(\sigma_{\text{company_name}=\text{"XYZ"} \wedge \text{salary}>8000}(\text{works} \bowtie \text{employee}))$

- C. Find the name of company that has the most employees.

Select company_name

From works

Group by company_name

Having count(distinct employee_name) \geq all

(select count(distinct employee_name)

From works

Group by company_name)

- D. Find the names of all employees in this database who live in the same city as the company for which they work [using SQL]

Select employee_name
 From works,employee,company
 Where works.employee_name=employee.employee_name
 and works.company_name=company.company_name
 and employee.city=company.city

E. Find the names, street address, and cities of residence of all employees whose manager lives in SHANGHAI. [using SQL]

Select e1.employee.name, e1.street, e1.city
 From manages g, employee e1, employee e2
 Where g.employee_name=e1.employee_name
 And g.manager_name=e2.manager_name
 And e2.city='SHANGHAI'

F. Give all manager in this database a 12 percent salary raise. [using SQL]

Update works
 Set salary=salary*1.12
 Where employee_name in (select manager_name
 From manages)

4. Query Processing, Optimization and Transaction.

A. 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 name=WangMing” and retrieves 3 tuples. Please compute the cost of this query at the worse case, i.e, the number of block transfer from disk and the number of seeks.

Overhead:

Visiting the index on ID cost 4 blocks transfers and 4 seeks.

Retrieving 3 tuples costs 3 blocks transfers and 3 seeks.

Total cost: 7 blocks transfers and 7 seeks.

B. Describe the process of Merge-join.

C. Please describe the two-phase locking protocol and prove that it ensures conflict-serializable schedules and does not ensure freedom from deadlocks.

D. Below we show some log of a DBMS, please describe the recovery procedure using deferred database modification.

<T0 start>	<T0 start>	<T0 start>
<T0, A, 950>	<T0, A, 950>	<T0, A, 950>
<T0, B, 2050>	<T0, B, 2050>	<T0, B, 2050>
<T0 commit>	<T0 commit>	
<T1 start>	<T1 start>	

<T1, C, 600>

<T1, C, 600>

<T1 commit>

(a)

(b)

(c)