Quiz1

1. **According to the levels of data abstraction, database systems have several schemas, which are as follows:**

**Physical schema**, **Logical schema**, and **View schema**.

1. **List the names of data model used in database systems, and explain briefly how or when to use it.**

(i) **E-R model:** It is used in the conceptual database design, which describes the real world objects with entities, and relationships among these objects.

(ii) **Relational model:** It is used in the logical database design. It is the basis of the most DBMS products in the database world for more than 30 years, and it uses tables to represent both data and the relationships among those data.

(iii) **Object-oriented model:** It is heavily influenced by object-oriented programming language and can be understood as an attempt to add DBMS functionality to a programming language environment. It can be regarded as extending the E-R model with notions of encapsulation, methods and object identity, and support complex data structure. The model can be used in logical database design in most cases, and can also be used in conceptual design step.

(iv) **Object-relational model:** It extends the relational model with features of object-oriented model.

(v) **Semistructured-data model:** It is used in the logical database design steps. In this model, XML is widely used to represent semistructured data.

(vi) **Network model:** Data are represented by collections of records, and relationships among data are represented by links, which can be viewed as pointers. The records in the database are organized as collections of arbitrary graphs. It was only used in old database systems.

(vii) **Hierarchical model:** It is similar to the network model in the sense that data and relationships among data are represented by records and links, respectively. But it differs from the network model in that the records are organized as collections of trees rather than arbitrary graphs. It was used only in old database systems.

1. **What are the physical data independence and logical data independence in database system?**

**Physical Data Independence:** the ability to modify the physical schema without changing the logical schema, as well as application programs, that is, the modification in the physical schema does not affect the logical schema, there application program need not be changed.

**Logical data independence:** Protect application programs from changes in *logical* structure of data, i.e., the modification in the logical schema does not affect the view schema, there application program need not be changed.

Quiz2

**Problem 1. Relational algebra**

***Student*(*Sid*, *Name*, *Age*)**

**1 2 3**

***Project*(*ProjectName*, *Sid*, *Score*)**

**5 6 7**

**⋈15,26,37**

× 15 16 17 25 26 27 …

**(1) Find the names of students who are in the project with project name ‘MiniSQL’.**

Π*name*((*student*) **⋈** (σ*projectName*=‘MiniSQL’(*project*)))

**(2) Find the Sid of students who have not been in any project team yet.**

Π*sid*(*student*) − Π*sid*(*project*)

**(3) Find the names of students who are the youngest.**

Method 1: Π*name*(*student*) − Π*name*(σ*student.age*>*st2.age*(*student* × (ρ*st2*(*student*))))

Method 2: Temp ← gmin(*Age*)(*student*);

Π*name*(σ*age*=*minage*(*student* × (ρT(*minage*)(Temp))))

**Problem 2. Write SQL statement for the following queries.**

***Student*(*Sid*, *Name*, *Age*)**

***Project*(*ProjectName*, *Sid*, *Score*)**

**(1) Find the names of students who get score more than 85 in the project.**

SELECT *Name*

FROM *Student* *S*, *Project* *P*

WHERE *S.Sid* = *P.Sid* and Score > 85

**(2) Find the names of students who get the maximum score in each project.**

Method 1: SELECT *Name*

FROM *Student S*, *Project P*

WHERE *S.Sid* = *P.Sid* and *Score* = (SELECT max(*Score*)

FROM *Project*

WHERE *ProjectName* = *P.ProjectName*)

Method 3: SELECT *Name*

FROM *Student S*, *Project P*

WHERE *S.Sid* = *P.Sid* and (*ProjectName*, *Score*) in (SELECT *ProjectName*,

max(*Score*)

FROM *Project*

GROUP BY *ProjectName*)

Method 2: SELECT *Name*

FROM *Student*

WHERE *Sid* in (SELECT *Sid*

FROM *Project P*

WHERE *Score* >= ALL (SELECT *Score*

FROM *Project*

WHERE *ProjectName* = *P.ProjectName*))

Quiz3

**Problem 1.** Consider a SQL table ***T*(*A int unique*, *B int*)*.*** Assume there are no NULL values. As specified, attribute *A* is a key. Consider the following three SQL queries:

**Q1: Select *B* From *T***

**Where *B* >= some (Select *B* From *T*);**

**Q2: Select *B* From *T* as *T1***

**Where *B* > all (Select *B* From *T* as *T2* where *T2*.A < > *T1*.A);**

**Q3: Select max(*B*) From *T*;**

Which of the queries above are equivalent? Please show a smallestsingle instance of *T* you can come up with that demonstrates your answer.

Case 1: Q2 and Q3 are equivalent, if the maximum of *B* is only once.

Case 2: None are equivalent, if the maximum of *B* is multiple times.

|  |  |
| --- | --- |
| A | B |
| 1 | 5 |
| 2 | 7 |
| 3 | 3 |

|  |  |
| --- | --- |
| A | B |
| 1 | 7 |
| 2 | 5 |
| 3 | 7 |

Case 1 Case2

Q1: 5 Q1: 5

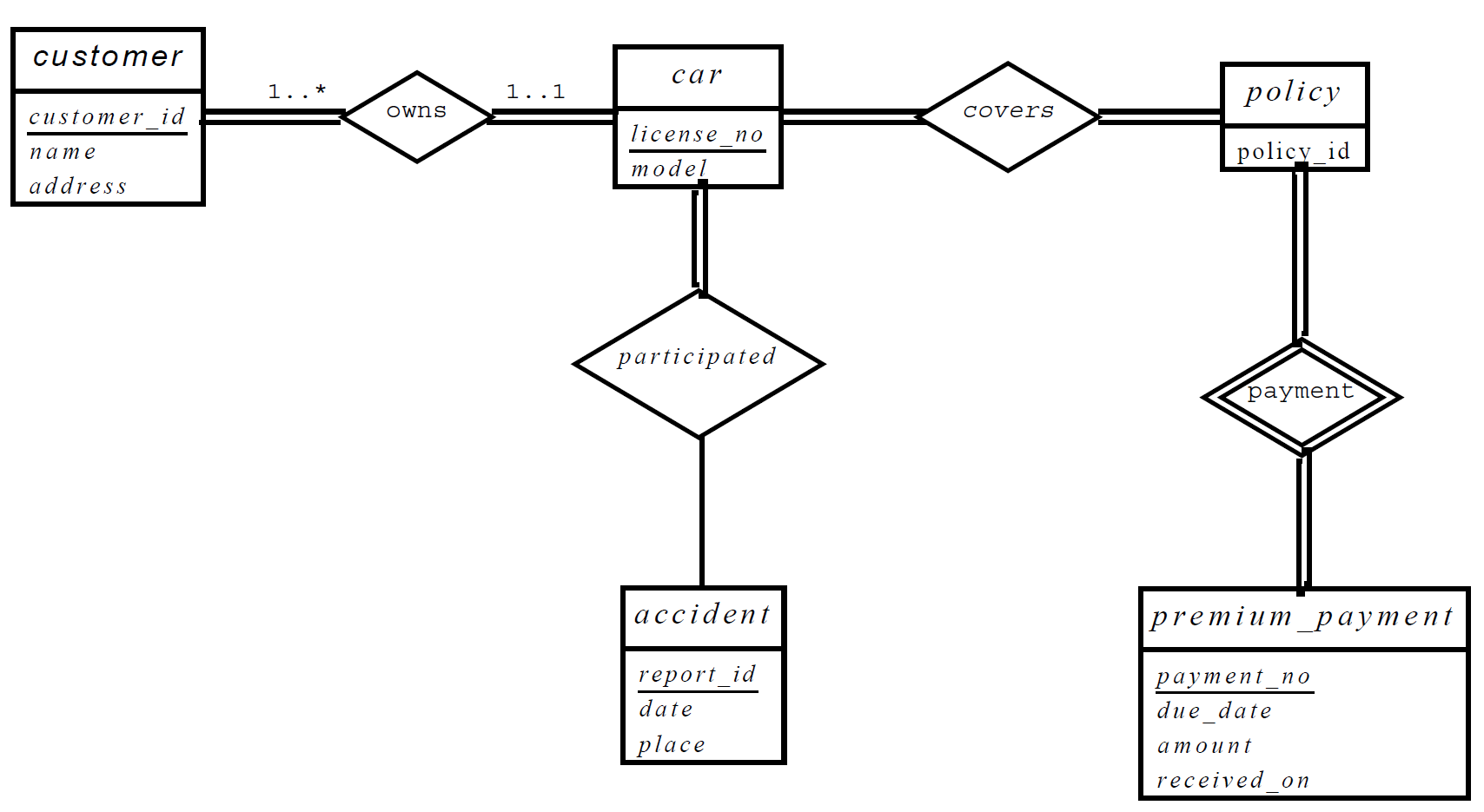
7 7

3 3

Q2: 7 Q2:

Q3: 7 Q3: 7

7.1 Construct an E-R diagram for a car insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents. Each insurance policy covers one or more cars, and has one or more premium payments associated with it. Each payment is for a particular period of time, and has an associated due date, and the date when the payment was received.



**Problem 2.** Consider the following relational schemas describing ***books***, ***publishers***, ***readers***, and reader ratings of the books:

***Book*(*bid*, *title*, *author*, *price*) // *bid* is a primary key**

***Reader*(*name*, *age*, *gender*, *profession*) // *name* is a primary key**

***Rating*(*name*, *bid*, *date*, *score*) // (*name*, *bid*) is a primary key**

***Publisher*(*pid*, *pname*, *location*, *phone*) // *pid* is a primary key**

***PublishedBy*(*bid*, *pid*, *year*) // *bid* is a primary key**

1. Draw an E-R diagram from which these relational schemas could have been produced. Your diagram should be fully connected, and it should be as detailed as possible from the information you have.

**…**

**BOOK**

**Rating**

**READER**

**PublishedBy**

**PUBLISHER**

1. Please make necessary formalization of the relational schemas above, to get a minimum number of relation schemas.

*Book*(*bid*, *title*, *author*, *price*, *pid*, *year*) // *bid* is a primary key

*Reader*(*name*, *age*, *gender*, *profession*) // *name* is a primary key

*Rating*(*name*, *bid*, *date*, *score*) // (*name*, *bid*) is a primary key

*Publisher*(*pid*, *pname*, *location*, *phone*) // *pid* is a primary key

1. Write SQL data definition statements for the relation schemas from the issue/step (2), and give necessary integrity constraints on them.

Create table *Publisher*(*pid* char(10) primary key,

*pname* varchar(30),

*location* varchar(50),

*phone* varchar(20));

Create table *Book*(*bid* char(10) primary key,

*title* varchar(50),

*author* varchar(50),

*price* real,

*pid* char(10),

*year* date,

Foreign key(*pid*) references *publisher*);

Create table *Reader*(*name* varchar(15) primary key,

*age* int,

*gender* char(1) not null,

*profession* varchar(30),

check (*gender* in (‘M’, ‘F’)));

Create table *Rating*(*name* varchar(15),

*bid* char(10),

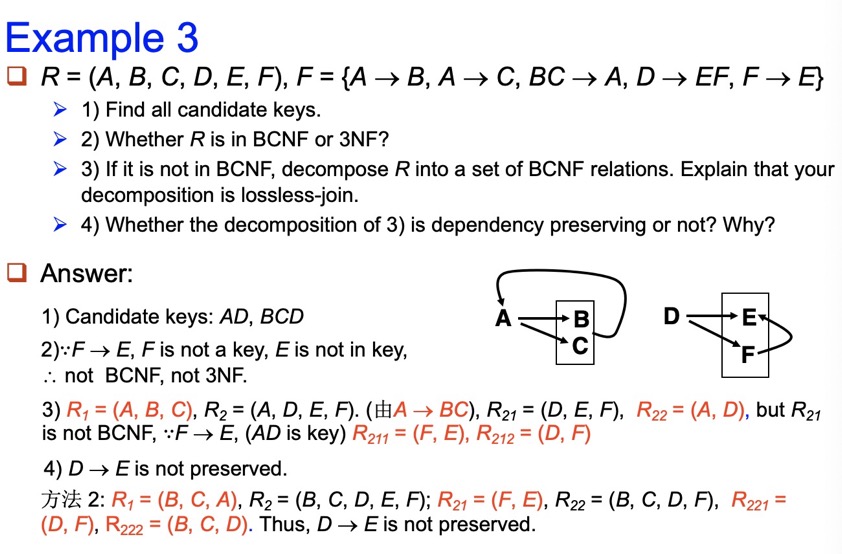
*date* date,

*score* real,

primary key (*name*, *bid*),

foreign key (*name*) references *reader*,

foreign key (*bid*) references *book*);



判断是否是 3NF 的条件： 对于 R 上的每个函数依赖 X->A （X 是关系 R 属性的一个子集，A 是 R 的一个属性） ,以下条件中的一个成立：  
1 X ∈A  
2 X 是超码  
3 A 是 R 的码的一部分

判断是否是 BCNF 的条件：对于 R 上的每个函数依赖 X->A（X 是关系 R 属性的一个子集，A 是 R 的一个属性） ,以下条件中的一个成立：  
1 X ∈A  
2 X 是超码

Quiz4

**Problem 1.** For the following B+ tree (n=3):

P

F

T

Y

F

H

P

T

X

Y

A

1. Draw the B+ tree after insert an index item with key ‘W’ to the given tree.

1.找到子节点，插入

2.如果需要拆分，依次向上，绝不向下。

Solution:

P

X

F

Y

F

H

P

T

W

X

A

T

Y

1. Draw the B+ tree after delete an index item with key ‘A’ from the original tree.

1.删除该节点,若不满足B+树性质则2

2.往上找父节点p,与p的右边兄弟节点合并

3.判断当前节点父节点符不符合B+树性质

若符合，结束

若不符合，

可拆分，拆分

不可拆分，重复2

Solution:

T

P

Y

F

H

P

T

X

Y

1. Assume that the B+ tree contains 1000 index items, please estimate the height of the B+ tree.
2. Assume that the B+ tree contains 1000 index items, please estimate the size (i.e. the number of nodes) of the B+ tree.

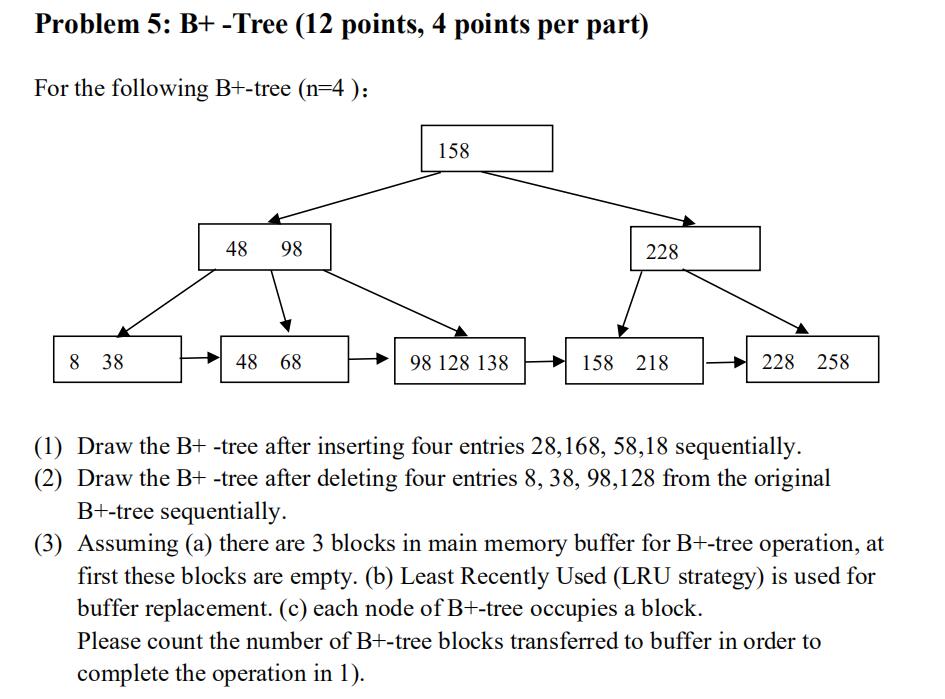
二叉树 1000 + 500 +250+ 125+ 62+31 +15+7+3+1

1750+125+62+57

1994

B+树 500 + 167+56+19+7+3+1

753



**Problem 2.** Consider the following relational schema and SQL query:

**product(pid: char(10), name: char(20), producer: char(20), price: integer)**

**customer(cid: integer, name: char(20), age: integer; city: char(20))**

**order(cid: integer, pid: char(10))**

**select customer.name, product.name**

**from customer, order, product**

**where customer.cid= order.cid and product.pid = order.pid**

**customer.city =‘Hangzhou’ and product.price>=200;**

1. Identify a relational algebra tree (or a relational algebra expression if you prefer) that reflects the order of operations that a decent query optimizer would choose.
2. What indexes might be of help in processing this query? Explain briefly.

Since the pid (in product and order) and cid (in customer and order) is crux of the join, it’s helpful to create B+tree indexes in these attributes. (Different answer is OK if it makes sense.)

**12.3** Let relations *r*1(*A*, *B*,*C*) and *r*2(*C*, *D*, *E*) have the following properties: *r*1 has 20,000 tuples, *r*2 has 45,000 tuples, 25 tuples of *r*1 fit on one block, and 30 tuples of *r*2 fit on one block. Estimate the number of block transfers and seeks required, using each of the following join strategies for

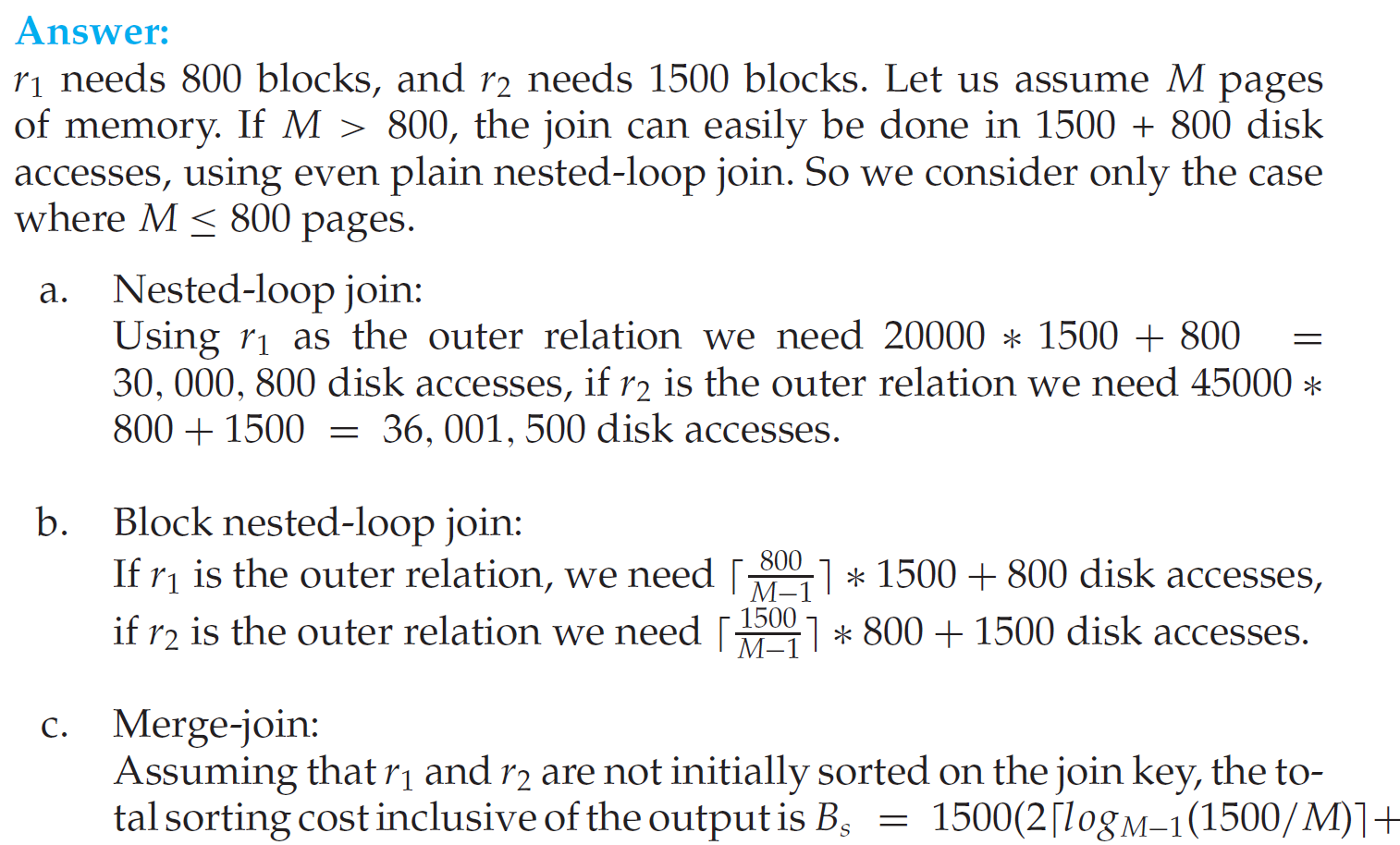
*r*1 ⋈*r*2:

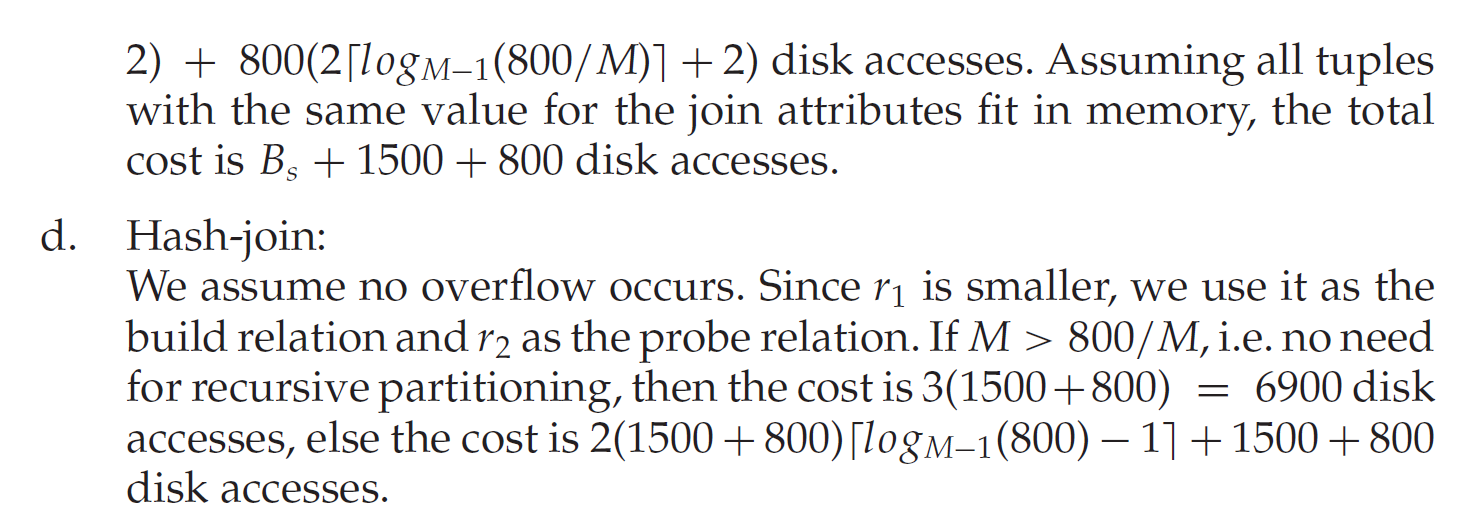
a. Nested-loop join.

b. Block nested-loop join.

c. Merge join.

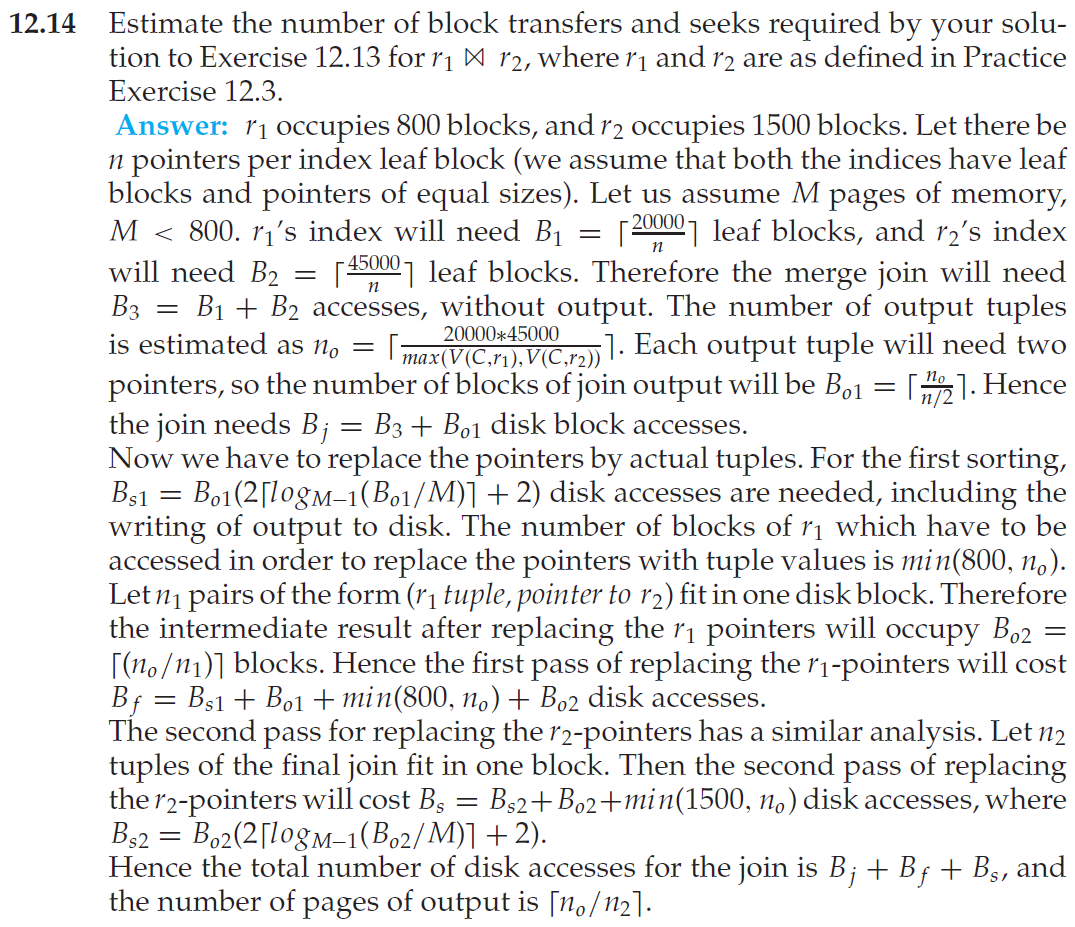
d. Hash join.





**12.13** Design a variant of the hybrid merge-join algorithm for the case where both relations are not physically sorted, but both have a sorted secondary index on the join attributes.

**Answer:** We merge the leaf entries of the first sorted secondary index with the leaf entries of the second sorted secondary index. The result file contains pairs of addresses, the first address in each pair pointing to a tuple in the first relation, and the second address pointing to a tuple inthe second relation. This result file is first sorted on the first relation’s addresses. The relation is then scanned in physical storage order, and addresses in the result file are replaced by the actual tuple values. Then the result file is sorted on the second relation’s addresses, allowing a scan of the second relation in physical storage order to complete the join.



**Problem 7: Concurrency Control**

Consider following three transactions:

T1: read(A)

Read(B)

Write(B)

T2: read(B)

Read(A)

Write(A)

T3: read(A)

Write(A)

1) For the following schedule, please draw the precedence graph, and explain whether it is conflict serializable.

|  |  |  |
| --- | --- | --- |
| T1 | T2 | T3 |
| Read(A) |  |  |
| Read(B) |  |  |
|  |  | Read(A) |
|  |  | Write(A) |
|  | Read(B) |  |
|  | Read(A) |  |
| Write(B) |  |  |
|  | Write(A) |  |

2) For the following schedule, please explain whether it is cascadeless.

|  |  |  |
| --- | --- | --- |
| T1 | T2 | T3 |
| Read(A) |  |  |
| Read(B) |  |  |
| Write(B) |  |  |
|  | Read(B) |  |
|  | Read(A) |  |
|  | Write(A) |  |
|  |  | Read(A) |
| Abort |  |  |

3) Please explain whether the two-phase locking protocol can be used to implement the schedule in 1).

**Problem 8: Aries Recovery Method (15 points, 3 points each)**

A DBMS uses Aries algorithm for system recovery. Following figure is a log file just after system crashes. The log file consists of 14 log records with LSN from 1001 to 1014. The figure does not show PrevLSN and UndoNextLSN in log records. Assume that last completed checkpoint is the log record with LSN 1008.

1001: <T1, begin>

1002: <T1, 101.1, 11, 21>

1003: <T2, begin>

1004: <T2, 102.1, 52, 62>

1005: <T2, commit>

1006: <T3 begin>

1007: <T3, 102.2, 73, 83>

1008: checkpoint

|  |  |
| --- | --- |
| Tx | LastLSN |
| T1 | 1002 |
| T3 | 1007 |

|  |  |  |
| --- | --- | --- |
| PageID | PageLSN | RecLSN |
| 101 | 1002 | 1002 |
| 102 | 1007 | 1004 |

1009: <T1, 101.2, 31, 41>

1010: <T4 begin>

1011: <T3, 102.2, 73>

1012: <T3, abort>

1013: <T4, 102.1, 62, 64>

1014: <T1, commit>

Please answer following questions：

1) Which log record is the start point of Redo Pass?

2) Which log record is the end point of Undo Pass?

3) After Analysis Pass, what is the undo list?

4) After recovery, what is the value of data items identified by “102.1” and “102.2”, respectively?

5) What additional log records are appended to log file during recovery?

**第一次：数据库系统引论**

**主要内容**：

数据库是现代信息社会的基石,绝大多数计算机应用都依赖于数据库系统的支持。数据库广泛应用于企业生产制造、金融、医疗、教育、社会服务、互联网等各个领域。

介绍数据库基本概念、数据库主要特征、数据库与文件系统的区别、数据模型、数据库系统三级模式结构、数据库用户和系统管理员、数据库系统结构等基本概念和专业术语。

介绍关系数据库的基本概念;讲述SQL语言基础。

**课外学习**：数据库系统有什么特征？它和文件系统比较有什么优势？

**第二次：关系数据模型**

**主要内容**：

数据模型是数据库系统的一个根本特性。关系数据模型因为其简单有效而在数据库领域占据主导地位。讲授关系模型的数学模型，包括关系模型的数据结构、数据完整性、数据操作；重点讲授关系代数及基本关系操作及附加关系操作，学习用关系代数表达式表达数据查询要求。

**课外学习**：关系代数有哪些基本操作？关系代数的查询表达能力如何？

**第三次：SQL语言(1)-表定义、基本SQL查询**

**主要内容：**

SQL(Structured Query Language) 是关系数据库标准语言,包括数据定义、数据操纵、数据控制一体化管理功能。SQL是一种陈述式的语言。讲授SQL的表定义，包括：SQL基本数据类型、primary key、foreign key和check定义，以及SQL DML语句的基本用法，包括select、from、where、group by、having各子句。

**课外学习**：SQL作为一种陈述式语言，它和过程式语言（如C）有什么区别？SQL的数据完整性定义功能有什么优点？

**第四次：SQL语言(2)-SQL嵌套查询、数据更新、视图、索引**

**主要内容：**

讲授SQL嵌套子查询构成的复杂查询。

讲授SQL数据更新语句，包括insert、delete、update语句.

SQL视图(view)和索引(index)分别对应数据库三级模式中的用户模式和物理模式。用户可以象查询基本表一样查询视图中的数据，在特定情况下可通过视图更新基本表中数据。索引可以加快数据库查询处理的效率。讲授视图的语法和用法，以及可更新视图的概念。通过具体例子讲解视图的优点。讲授索引的作用、索引的类型SQL索引定义的语法。

**课外学习**：1)构想一个数据库应用，用SQL定义数据库中的表，并构思若干查询要求，写出对应的SQL语句。2)举例说明视图的优点。3)如何选择在数据库表上建立哪些索引？

**第五次：SQL语言(3) - 数据完整性、安全性和事务**

**主要内容：**

数据完整性包括primary key、foreign key、check和断言(Assertion，即一个数据库必须满足的条件谓词)。触发器(Trigger)是数据库表更新时自动触发执行的动作，也是维护数据完整性的一种手段。安全性控制是数据库中不可缺少的功能。SQL中的安全性控制包括用户身份鉴别、权限管理和审计三方面。事务（transaction）是构成一个完整的逻辑工作单元的数据库操作的集合，是数据库系统进行并发控制和恢复的基本手段。Basic operations, additional operations, extended operations三者的关系，是否可替代？

**课外学习**：1)触发器在数据库系统中有哪些用处?　2)数据库安全控制有哪些方面？3)SQL 中和事务相关的语句有哪些？

**第六次：SQL语言(4)-嵌入式SQL、ODBC、 JDBC**

**主要内容：**

嵌入式SQL是应用程序调用数据库的一种方式。SQL和高级程序设计语言(如C语言)存在着基本数据类型和执行方式两方面的不匹配。嵌入式SQL通过游标(cursor)等方式处理这些不匹配的问题。讲授嵌入式SQL的基本原理，包括不带cursor的SQL语句和带cursor的嵌入式SQL语句、静态嵌入式SQL、动态嵌入式SQL、ODBC、JDBC。

**课外学习**：SQL和C语言在数据类型和执行方式上存在哪些不匹配的地方？嵌入式SQL如何处理这些不匹配? ODBC比起嵌入式SQL有什么优缺点？

**第七次：数据库设计和ER模型**

**主要内容：**

实体-联系模型（Entity-Relationship Model）是一种概念模型，用于数据库分析阶段为现实世界建模。它使用ER图描述现实世界的实体（Entity）以及实体之间的联系（Relationship）。实体用以描述现实世界中可以区分的对象。实体所具有的特征称为实体的属性（Attribute）。实体之间存在着各种联系。

通过实体-联系方法得到现实世界的一个抽象模型，但这一模型并不能为数据库管理系统接受。要完成从现实世界到信息世界的转化，还必须将实体—联系方法所得的ER图转化为关系模式，并用SQL语句定义相应的表。

讲授实体-联系模型的各种要素，重点掌握采用实体-联系方法为现实世界建模的一般过程和要点。讲授ER模型中实体（包括弱实体）和联系（包括一对一、一对多、多对多联系）等的转换方法。

**课外学习**：1)采用实体-联系方法为现实世界建模时，有些信息既可以用实体来表达，也可以用联系来表达，这两种方式的选择依据是什么？2）一对多的联系既可以转换成一个独立的关系，也可以和实体对应的关系合并，这两种转换方法各有什么优缺点？

**第八次：关系数据库设计和关系规范化**

**主要内容：**

一个不好的关系数据模式会产生数据冗余、数据更新异常等问题。通过函数依赖的概念分析关系模式的规范化程度，并把不规范的关系模式分解为规范化的关系模式。讲授函数依赖的概念、Armstrong公理系统、关系模式的候选关键字（candidate key）以及关系模式分解的原则，即无损连接的分解和保持函数依赖的分解。

BCNF是函数依赖范畴内规范化程度最高的关系模式，而3NF是比BCNF低的规范化形式。一个关系模式总能无损连接地分解为BCNF 的关系模式，但不一定能保持函数依赖；若要求分解既是无损连接的又是保持函数依赖的，则保证可以分解为3NF。 通过考察多值依赖，还可以获得更高规范化的关系模式，即4NF。讲授函数依赖的相关概念 ，以及3NF和BCF的定义**、**分解为BCNF和3NF的算法；介绍多值依赖及4NF的概念。

**课外学习**：1）一个不好的关系模式会产生哪些问题？如何消除这些问题？ **2）**3NF和BCNF有何区别？如何把关系模式分解为BCNF的关系模式？ 举例说明4NF。

**第九次 数据存储**

**主要内容：**

数据持久存放于以磁盘为代表的存储设备中，处理时需读入主存。磁盘和主存之间存在着巨大的访问速度鸿沟。讲授以块为单位的内外存数据传输、缓冲区管理与替换策略、记录在块中的存放方式，以及数据文件组织的主要形式。

**课外学习**：1）磁盘和主存之间存在着多大的访问速度鸿沟？为克服这样的鸿沟，数据库管理系统采取了哪些措施？2）比较行存和列存的优缺点。

**第十次 索引**

**主要内容：**

索引是数据库管理系统提高数据访问速度的主要措施之一。介绍稠密索引和稀疏索引的原理和区别，重点讲解B+树索引, 以及面向写优化的索引(buffer tree、LSM tree)。

**课外学习**：1）假如已知B+-树索引项的数目，如何估算B+-树的高度和结点总数？2）LSM树索引相比B+树索引有什么优势？

**第十一次 查询处理**

**主要内容：**

查询处理是数据库管理系统的主要功能之一。介绍关系数据库管理系统查询处理的主要过程，关系数据库基本操作（选择，连接，排序等）的算法实现及代价估算，以及关系表达式的求值方式。Nest join, hash; merge

**课外学习**：1）关系表达式求值的流水线方式有什么优点？是否所有的关系操作都可以采用流水线方式处理？

**第十二次 查询优化**

**主要内容：**

查询优化是关系数据库管理系统的核心功能之一。讲解查询优化的两个阶段的基本步骤，即代数优化和物理优化；介绍关系代数表达式的等价变换规则，以及基于代价估算的查询优化的基本原理。

**课外学习**： 1）查询优化为什么要用到数据库的统计信息？2）数据库应用的运行会带来统计信息的变化，这对查询优化会产生怎样的影响？

**第十三次 事务管理**

**主要内容：**

事务是数据库管理系统进行并发控制和恢复的基本单位。讲解事务的基本概念、事务的ACID性质、事务并发执行（日志记录规则）的好处和潜在问题，以及并发事务的可串行性和可恢复性。

**课外学习**：1）事务的并发执行有哪些好处？2）如果不进行恰当的并发控制，多个事务并发执行可能产生哪些潜在的问题？

**第十四次 并发控制**

**主要内容：**

并发控制保证多个事务并发执行如同串行调度一样获得正确的运行结果。讲解基于锁的并发控制协议的主要思想、两阶段封锁协议（2PL）、死锁及解决办法、多粒度锁, 以及数据删除和插入情况下的并发控制。什么是死锁预防，什么是死锁避免

**课外学习**：1）事务的并发执行有哪些好处？2）如果不进行恰当的并发控制，多个事务并发执行可能产生哪些潜在的问题？3）证明2PL是保证事务调度冲突可串行性的充分条件，而非必要条件。

**第十五次 数据库恢复**

**主要内容：**

数据库管理系统确保在系统发生各种故障的情况下，数据库能恢复到正常状态。讲解各种故障类型、基于日志的恢复策略、提高恢复效率的checkpoint方法,以及业界采用的ARIES恢复算法。不同的阶段（Phase）意义;脏表更新和checkpoint的关系；每一个phase的buffer里面存什么

**课外学习**： ARIES算法如何在DBMS恢复效率和系统正常运行时效率两方面取得平衡？