# 分布式数据库习题

## 第一章

1.1 请用自己的语言定义下列分布式数据库系统中的术语：

**全局/局部数据**

全局数据是指参与全局应用，可被多个站点上的应用访问的数据；

局部数据是指只提供本站点的局部应用所需要的数据。

**全局/局部用户（应用）**

在分布式数据库系统中，一个用户或一个应用如果只访问他注册的那个站点上的数据称为局部用户（应用）。

如果访问涉及两个或两个以上站点中的数据，称为全局用户（应用）。

**全局/局部 DBMS**

全局数据库系统是协调全局事务的，协调各局部 DBMS 以完成全局应用，保证数据库的全局一致性，执行并发控制，实现更新同步，提供全局恢复功能的数据库管理系统。

局部数据库管理系统位于局部场地上，是为建立和管理局部数据库，提供场地自治能力，执行局部应用及全局查询的子查询的数据库管理系统。

**全局/局部 DB**

全局数据库（GDB）是指从整个系统角度出发，由全局数据库管理系统进行管理的数据库，它由各个局部数据库逻辑组合而成；

局部数据库（LDB）是指从各个站点的角度出发，由局部数据库管理系统进行管理的数据库，它由全局数据库的某种逻辑分割而成。

**全局外模式**

是全局应用的用户视图，是全局概念模式的子集，也称全局视图。

**全局概念模式**

描述分布式数据库中全局数据的逻辑结构和数据特性，是分布式数据库的全局概念视图。

**分片模式**

描述全局数据的逻辑划分。每一个全局关系可以分为若干不相交的部分，每一部分称为一个片段，分片模式定义片段以及全局关系到片段的映像。

**分配模式**

分配模式定义片段的存放结点。根据选定的数据分布策略，定义各片段的物理存放站点，即定义片段映像的类型，确定分布式数据库是冗余的还是非冗余的，以及冗余的程度。

**局部概念模式**

是全局概念模式的子集，对每个站点来说，在该站点上全部物理映像的集合就称为该站点上的局部概念模式。

1.2 采用分布式数据库系统的主要原因是什么？

分布式数据库系统适合于单位分散的部门，允许各个部门将其常用的数据存储在本地，实施就地存放本地使用，从而提高响应速度，降低通信费用。采用分布式数据库的原因主要有两方面：（1）集中式数据库系统的不足：数据按实际需要已经在网络上分布存储，如果再采用集中式处理，势必造成附加成本和通信开销；应用程序集中在一台计算机上运行，一旦该计算机发生故障，将会影响整个系统的运行，可靠性不高；集中式处理导致系统的规模和配置都不够灵活，系统的可扩展性较差。（2）分布式数据库系统的优点：具有灵活的体系结构；适应分布式的管理和控制机构；经济性能优越；系统的可靠性高、可用性好；局部应用的响应速度快；可扩展性好，易于集成现有系统。

1.3 分布式数据库系统可分为那些类？

（1）按局部数据库数据管理系统的数据模型分类：

同构型 DDBS （包括两种：同构同质型。同构异质型），异构型 DDBS

（2）按分布式数据库系统的全局控制系统类型分类：

全局控制集中型 DDBS，全局控制分散型 DDBS，全局控制可变型 DDBS

1.5 分布式 DBMS 具有哪些集中式 DBMS 不具备的功能？

（1） 物理分布性：分布式数据库中的数据不是存储在一个站点上，而是分散存储在由计算机网络联结起来的多个站点上。

（2） 逻辑整体性：分布式数据库中的数据物理上是分散在各个站点中的，但这些分散的

数据逻辑上却是一个整体。

（3） 站点自治性：站点自治性也称场地自治性，各站点上的数据由本地的 DBMS 管理，具有资质处理能力，完成本站点的应用（本地应用）。

（4） 集中与自治相结合的控制机制：同一站点上的用户可共享本站点上局部数据库中的数据，以完成局部应用；分布式数据库系统上的用户都可共享在分布式数据库系统的各个站点上存储的数据，以完成全局应用。

（5） 适当增加数据冗余度：在集中式数据库中，尽量减少冗余度是系统目标之一。而在分布式系统中却通过冗余数据提高系统的可靠性、可用性和改善系统性能。

（6） 事务管理的分布性：数据的分布性必然造成事务执行和管理的分布性。即，一个全局事务的执行可分解为在若干个站点上子事务（局部事务）的执行。

1.6 请用自己的语言解析“什么时候需要进行数据分片和数据复制”。

数据分片：全局数据库是由各个局部数据库逻辑组合而成，数据库中的一个关系描述了某些数据之间的逻辑相关性，但不同站点的用户需要该关系中的元组可能不同。这就需要对这个关系进行分割，并将分割后的片段存放在相应的站点上。

数据复制：当一个分布式数据库中用户数量较大，地理分布较广，而且需要实时地访问相同数据时可以采用数据复制技术，它通过将这些共享数据复制到位于不同地点的多个数据库中，从而实现数据的本地访问，减少了网络符合，并提高了数据访问的性能，而且通过对数据库中的数据定期同步的更新，从而确保了所有的用户使用同样的、最新的数据。

1.7 在分布式数据库系统中，为什么要对数据进行分片？什么是关系的片段？关系的片段有哪些主要类型？

对数据分片的目的是产生一个对全局数据合适的划分方案，使用这种方案得到的片段作为分布式数据库中数据的分配和存储单位时，不但能够减少应用中的操作量，而且能够对于应用具有最大可能的本地性，即使得各片段位于其使用最多的站点。

关系的片段：对数据库管理系统中的关系进行分割，将分割后得到的各部分元组，就称为该关系的逻辑片段。

关系的片段主要类型有:

（1）水平片段：按一定的条件把全局关系的所有元组划分成若干不相交的子集，每个子集

为关系的一个片段。也是通过对一全局对象的实例（或元组）进行选择得到的子集构。

（2）垂直片段：把一个全局关系的属性集分成若干子集，并在这些子集上做投影运算，每个投影为垂直分片。也是通过将全局对象在其属性子集上进行投影得到的。

（3）混合分片：将水平分片与垂直分片方式综合使用则为混合分片。

# 第二章

2.1 概述分布式数据库系统的创建方法、方法特点和适用范围。

分布式数据库系统的创建方法即分布式数据库系统的实现方法，大致可分为两种：组合法和重构法。

（1）采用组合法创建分布式数据库系统

组合法也称集成法，这是一种自底向上(bottom—up)的创建方法。它是利用现有的计算机网络和独立存在于各个站点上的现存数据库系统，通过建立一个分布式协调管理系统，将它们集成为一个统一的分布式数据库系统。

特点：这种方法由于是利用现存的网络和现存的数据库系统，仅仅需要建立一个分布式协调管理系统。因此，相对来说，如果该系统不是很大的话，其工作量可能会比较小，实现的周期会短些，花费的人力、物力会少些，用户也比较容易接受，因为它有利于保护现有的投资。

适用范围：采用组合法的分布式数据库系统往往是异构或同构异质的分布式数据库系统。

（2）采用重构法创建分布式数据库系统

重构法是根据系统的实现环境和用户需求，按分布式数据库系统的设计思想和方法，采用统一的观点，从总体设计做起，包括各站点上的数据库系统，重新建立一个分布式数据库系统。

特点：这种方法的优点在于，可以按照统一的思想来考虑分布式数据库系统中的各种问题，有效地解决分布式数据库系统的数据一致性、完整性和可靠性。但是花费的人力、物力会比较多，研制周期也比较长，系统建设的代价会比较大。在实际应用中，究竟应该采用哪种方法，要根据具体情况做具体分析后决定。

适用范围：采用重构法创建的分布式数据库系统，通常是同构异质，甚至是同构同质的分布式数据库系统。

2.2 分布式数据库设计的主要目标是什么？

分布式数据库设计的目标除包括集中式数据库设计中的目标外，还要包括以下几点：

（1）分布式数据库的本地性或近地性

分布式数据库系统中最重要的目标(至少在使用广域网的情况下，以及某些使用局域网的情况下)是尽量减少对网络的利用，即尽可能减少站点之间的通信次数和通信量。所以，

分布式数据库设计中的一个主要原则是使数据和应用实现最大程度的本地性。本地性的优点不仅仅减少了远程访问的次数，而且简化了对该应用执行的控制。

（2）控制数据的适当冗余

控制数据的适当冗余是分布式数据库系统设计的又一个目标。分布式系统在可用性和可靠性方面优于非分布式系统，其原因之一是因为分布式数据库系统中存在数据的适当冗余。

（3）工作负荷分布

分布式计算机系统的一个重要特征是把工作负荷分布在网络中的各个站点上。分布工作负荷的目的是充分利用每个站点计算机的能力和资源，以提高应用执行的平行程度，从而提高系统的性能。

（4）存储的能力和费用

数据库的分布会受到各站点的存储能力的影响。在网络中可以有专门用于存储数据的站点，也可以有完全不支持大容量存储的站点。

上述的设计目标都要达到是非常困难的，因为这会使优化模型变得非常复杂。因此，可以将上面的某些特征考虑为约束条件而不是目标。

2.3 概述分布式数据库设计的关键问题及其解决方法。

作为数据库系统设计的核心部分是数据库设计，数据库设计的主要问题是模式(也称概念模式，描述数据库应用所使用的全部数据)和内模式(也称物理模式，描述概念模式映射到存储区域，并确定合适的存取方法)的设计。这两个问题在分布式数据库中变为全局模式设计和每个站点的局部数据库设计的问题，其中的关键是数据库的全局模式应如何划分：并映射到合适站点上。这就产生了分布数据库设计所特有的两个新问题：数据的分片设计和片段的位置分配设计。

分布式数据库的分布设计要求确定数据的分片和片段的分配。分片是指把一全局对象(实体或关系)细分成若干逻辑片段的过程；分配是指把各片段映射到一个或多个站点的过程，片段是最合适的数据分配单位。

2.6 数据分片应遵守哪些基本原则？数据分片有哪些基本类型和方法？

分片方法，必须遵守如下规则：

若 R={R1，R2，…，Rn}满足：

1)完整性(completeness)条件：

如果 a∈R，则必有 a∈Ri，i=1，2，…，n

2)可重构(reconstructed)条件：

R=∪ Ri，(水平分片)或 R=∞Ri，(垂直分片)

3)不相交(disjoint)条件：

Ri∩ Rj=ϕ，i≠j，i，j：=1，2，…，n(水平分片)

Ri∩Rj=主键属性，i，j=1，2，…，n(垂直分片)

有两种基本的数据分片方法：

（1）使用水平分片方法得到水平片段，水平片段是通过对一全局对象的实例(或元组)进行选择得到的子集构成。

水平分片是对全局关系执行“选择”操作，把具有相同性质的元组进行分组，构成若干个不相交的子集。水平分片的方法可归为初级分片(primary fragmentation)和导出分片(derivation fragmentation)两类。

（2）使用垂直分片方法得到垂直片段，垂直片段是通过将全局对象在其属性子集上进行投影得到的。

一个全局关系的垂直分片是通过“投影”操作把它的属性分成若干组。确定一全局关系 R 的垂直分片需要根据应用以“同样方式”(例如具有相同的使用频率)访问的属性来进行分组。这里把垂直分片问题和垂直群集(vettical clustering)问题区分开来，垂直分片的组必须只焦基个键属性上重叠,其他属性不可重叠,而垂直群集的组在其他属性上也可以重叠。

通过交替水平分片与垂直分片，可以产生混合分片。

2.7 为什么说在关系型分布式数据库中使用导出式水平分片，使关系之间的连接变得更加容易？试举一例。

全局关系的导出式水平分片不是以其自身的属性性质为基础,而是从另一个关系的属性性质或水平片段推导出来的。采用导出分片可使片段与片段之间的“连接”(join)变得更容易。

例： 设全局关系 SC(S#. C#，SCORE)

S(S#，SNAME，AGE，SEX)

若要将 SC 划分为男生的各门课成绩和女生的各门课成绩。这就不可能从 SC 本身的属性性质来执行选择，必须从关系 S 的属性性质或水平片段来导出。

define fragment SC 1 as

```
select SC. S#, C#, SCORE from SC, S
where SC. S#=S.S# and SEX=' M'
define fragment SC2 as
select SC.S#, C#, SCORE from SC, S
where SC.S#=S.S# and  SEX=' F'
```

如果 S 已经进行水平分片，分为 SF、和 SM，分别为男生全体和女生全体，则上述的片段定义可以基于片段 SF 和 SM 导出：

```
define fragment SC 1 as
select*from SC where S# in(select SF.S# from SF)
define fragment SC2 as
Select * from SC where S# in(select SM.S# from SM)
```

由此可见，使用导出式水平分片，使关系之间的连接变得更加容易。这是因为可将连接条件代之以子查询，从而使它变为一般的判别条件。

2.8 采用 DATAID-D 方法的分布式数据库设计与传统的集中式数据库设计在步骤和内容上有什么不同？

DATAID—D 是作为集中式数据库设计 DATAID 一 1 方法论的扩充而构造的，后者分成四个阶段：需求分析、概念设计、逻辑设计和物理设计。DATAID—D 要求对其增加两个阶段：分布要求分析阶段和分布设计阶段

1)分布要求分析阶段：需要这一阶段是为了收集关于分布的信息，如水平分片的划分谓词，每一应用在各站点激活的频率等。为了收集关于数据和应用分布的信息，必须从概念设计阶段的某些结果出发来收集关于分布要求。因此，分布要求分析阶段将位于概念设计阶段之后。

2)分布设计阶段：这一阶段始于全局数据库模式的规格说明和所收集的分布要求，然后产生全局数据的分片模式和片段的位置分配模式，分配模式描述了分配在各站点上的数据情况。

2.9 考虑图 2.12 所示的公司数据库的分片和分布。假设该公司有三个计算机站点，站点 B 和 C 分别属于部门 2 和 3.现在希望在站点 B 和 C 上分别频繁访问 EMPLOYEE 和 PROJECT 表中有关工作在该部门的雇员和该部门管辖的项目信息。雇员信息主要是指 EMPLOYEE 表的

NAME，ESSN，SALARY 和 SUPERSSN 属性。站点 A 供公司总部（部门 1）使用，经常存取为保险目的而记录的 DEPENDENT 信息外，还定期地存取所有雇员和项目的信息。根据这些给出的要求，对 COMPANY 关系数据库中关系进行分片和分布。

EMPLOYEE

| FNAME | MINIT | LNAME | ESSN | BDATE | ADDRESS | SEX | SALARY | SUPERSSN | DNO |
|---|---|---|---|---|---|---|---|---|---|

DEPARTMENT

| DNAME | DNO | MGRSSN | MGRSTARTDATE |
|---|---|---|---|

DEPT_LOCATION

| DNO | DLOCATION |
|---|---|

PROJECT

| PNAME | PNUMER | PLOCATION | DNO |
|---|---|---|---|

WORKS_ON

| ESSN | PNO | HOURS |
|---|---|---|

DEPENDENT

| ESSN | DEPENDENT NAME | SEX | BDATE | RELATIONSHIP |
|---|---|---|---|---|

图 2.12 COMPANY 关系数据库模式的一个可能的关系数据库状态（主码用下标线标出）

解：根据给出的要求，图 2.12 中的整个数据库可以被存储在站点 A。为了确定将要复制到站点 2 和 3 上的片段，先根据 DEPARTMENT 表的主码 DNO 的值进行水平分片，然后基于外码部门号(DNO)将导出的片段应用到关系 EMPLOYEE, PROJECT 和 DEPT_LOCATIONS 土，再在刚才得到的 EMPLOYEE 片段上进行垂直分片，得到只含属性{NAME , ESSN , SALARY * SUPERSSN , DNO)的片段。图 2. 13 给出了 EMPD2 和 EMPD3 的混合分片，它包括了分别满足条件 DNO = 2 和 DNO = 3 的 EMPLOYEE 元组。类似地，PROJECT,DEPARTMENT 和 DEPT- LOCATIONS 都按部门编号进行水平分片，这些片段根据其相应的部门号分别存储在站点 B 和 C 上，如下图所示。

| EMPD5 | FNAME | MINIT | LNAME | ESSN | SALARY | SUPERSSN | DNO |
|---|---|---|---|---|---|---|---|
| | John | B | Simth | 123456789 | 20000 | 333445555 | 2 |
| | Franisa | T | Wong | 333445555 | 10000 | 888665555 | 2 |
| | Reminia | K | Naraya | 666884444 | 15000 | 333445555 | 2 |
| | Joyee | A | English | 453453453 | 20000 | 333445555 | 2 |

| DEP5 | DName | DNO | MGRSSN | MGRSTARTDATE |
|---|---|---|---|---|
| | Research | 2 | 333445555 | 2003-05-22 |

| DEP5_LOC5 | DNO | LOCATION |
|---|---|---|
| | 2 | Bellaire |
| | 2 | Sugarlnd |
| | 2 | Housten |

| PROJS55 | PNAME | PNUMBER | PLOCAION | DNO |
|---|---|---|---|---|
| | Product X | 1 | Bellaire | 2 |
| | Product Y | 2 | Sugarlnd | 2 |
| | Product Z | 3 | Housten | 2 |

| DEP5_LOCS | ESSN | PNO | HOURS |
|---|---|---|---|
| | 123456789 | 1 | 32.5 |
| | 123456789 | 2 | 7.5 |
| | 666884444 | 3 | 40.0 |
| | 453453453 | 1 | 20.0 |
| | 453453453 | 2 | 20.0 |
| | 333445555 | 2 | 10.0 |
| | 333445555 | 3 | 10.0 |
| | 333445555 | 10 | 10.0 |
| | 333445555 | 20 | 10.0 |

〔a)站点 B 上的对应部门 2 的关系片段

现在需要对 WORKS_ON 关系进行分片，并且确定哪些 WORKS_ON 片段要存储在站点 B 和 C 上口这时面临这样一个问题，WORKS_ON 中的属性没有一个可以直接指明每个元组所属的部门。实标上 WORKS_ON 中的每个元组与雇员 e 和项日 P 相联系。我们既可以以 e 工作的部门也可以以管辖 P 的部门 d’来分片 WORKS_ON 表。如果对表 WORKS_ON 的所有元组增加一个 d_d’的限制，分片就会变得更加容易，即雇员只能为那些他所在部门管辖的项目工作。然而，在

图 2. 12 中的数据库没有给出这样的限制。所以，对此可以先以雇员工作的部门(用条件 C 表述)来分片 WORKS_ON 然后进一步以管辖雇员工作项目的部门来分片，如下图所示。

| EMPD4 | FNAME | MINIT | LNAME | SSN | SALARY | SUPERSSN | DNO |
|---|---|---|---|---|---|---|---|
| | Ahcia | J | Zelaya | 999887777 | 25000 | 987654321 | 4 |
| | Jennifer | S | Wallace | 987654321 | 43000 | 888665555 | 4 |
| | Ahmad | V | Jabbar | 999887777 | 25000 | 987654321 | 4 |
| | | | | | | | |

| EMPD4 | FNAME | MINIT | LNAME | ESSN | SALARY | SUPERSSN | DNO |
|---|---|---|---|---|---|---|---|
| | John | B | Simth | 123456789 | 20000 | 333445555 | 2 |
| | Franisa | T | Wong | 333445555 | 10000 | 888665555 | 2 |
| | Reminia | K | Naraya | 666884444 | 15000 | 333445555 | 2 |
| | Joyee | A | English | 453453453 | 20000 | 333445555 | 2 |

| DEP4_LOCS | DNAME | DNUMBER | LOCATION | MGRSSN | MGRSTARTDATE |
|---|---|---|---|---|---|
| | Administr | 4 | Statland | 987654321 | 2003-01-01 |

| DEP4_LOCS | ESSN | PNO | HOURS |
|---|---|---|---|
| | 333445555 | 10 | 10.0 |
| | 999887777 | 30 | 30.0 |
| | 999887777 | 10 | 10.0 |
| | 987987987 | 10 | 35.0 |
| | 987987987 | 30 | 5.0 |
| | 987654321 | 30 | 20.0 |
| | 987654321 | 20 | 15.0 |

| PROJS55 | PNAME | PNUMBER | PLOCAION | DNO |
|---|---|---|---|---|
| | Computer | 10 | Statland | 4 |
| | Newbener | 30 | Statland | 4 |

(b)站点 B 上的对应部门 4 的关系片段

在上图中，片段 G1,G2 和 G3 的并集给出了为部门 2 工作的雇员的所有 WORKS_ON 元组。

同样地，片段 G4,G5 和 G6 的并集给出了为部门 3 的雇员的所有 WORKS_ON 元组。另一方面，片段 GI !LI`I 和 G7 的并集给出了被部门 2 管辖的项目的所有 WORKS_ON 元组。因此，可以把片段 G1,G2,G3,G4 和 G 的并集放到站点 B 上，把片段 G5,G5,G6,G2 和 G8 的并集放在站点 C 上，可以看到片段 G2 和 G4 在两个站点上均被复制。

这就清楚地论证了对于大型数据库而言，数据的分片和分配是一个多么复杂的问题。

# 第三章

3.3 概述基于关系代数等价变换的查询优化算法的基本原理和实现步骤。

基本原理：把查询问题转变为关系代数表达式,分析得到查询树(语法树),进行从全局到片段的变换得到基于片段的查询树,然后利用关系代数变换规则的优化算法,尽可能先执行选择和投影操作.这样,一方面可以减少其后操作的操作量,另一方面可以减少操作次数.对该查询树进行优化,从而达到查询优化的目的.

实现步骤：

1、 将一个查询问题转换成关系代数表达式

2、 从关系代数表达式到查询树的变换：对一个关系代数表达式进行语法分析，可以得到一颗语法书，即树的叶子是已知的关系（或片段）、树的结点是关系操作符、节点是按代数表达式中的操作顺序组成的一组关系操作符。

3、 从全局查询到片段查询的变换：在具有分片透明性的系统中，这个变换的典型方法是：把基于全局关系的查询树中的全局关系名用其重构该全局关系的各片段名替换，变换成相应在片段上的查询树。

4、 利用关系代数等价变换规则的优化算法，对片段上的查询树进行优化处理，最后达到优化查询的目的。

3.4 概述基于半连接算法查询优化的基本原理和使用情形。

原理：经半连接操作，可减少操作关系的数据量，从而减少站点间数据的传输量。

适用情况：如果只需要一个关系中的一小部分元组参与和另一个关系连接的话，这是一个使数据传输量最小化的非常有效的方案。此时，T 半<T 全，采用半连接方案是合适的。

3.5 概述基于直接连接算法查询优化的基本原理和适用情形。

基本原理：基于直接连接算法的查询优化处理中的站点依赖算法、分片和复制算法。站点依赖和数据复制算法，以及 Hash 划分算法，主要是使得连接操作的数据传送量小（最好是无数据传送）和负载均衡，从而达到优化的目的。

使用情形：直接连接查询优化的算法有四种，通过比较，站点依赖算法的数据传送量最小（实际上没有），连接数据量最小切利用索引，因而能获得最佳性能。其次是 Hash 划分算法，最后是片段和复制算法。尽管如此，站点依赖算法只有在给出适当的可用语义信息时才使用；Hash 划分算法要求相对均匀的数据分布来得到良好的性能，最坏的情况下，两个关系可能被映射到同一个站点，而达不到负载均衡；在某个关系已经复制在包含其他关系片段的站点中，可在本地立即进行无数据传送处理，这种场合下使用分片和复制算法要好于 Hash 划分算法。在高速的局域网中，本地处理的代价也必须考虑在内，那么使用全连接是比较合适的。此时，查询优化策略就是去确定选择执行全连接的最佳方法。

3.6 设有关系 R，S，T 如图 3.13 所示。

（1）计算连接 R∞S∞T。

（2）计算半连接 R∝S，S∝R，S∝T，T∝R，T∝S，R∝T

R

| A | B | C |
|---|---|---|
| 2 | 3 | 5 |
| 5 | 3 | 6 |
| 1 | 6 | 8 |
| 3 | 4 | 6 |
| 5 | 3 | 5 |
| 2 | 6 | 8 |

S

| B | C | D |
|---|---|---|
| 3 | 5 | 6 |
| 3 | 5 | 9 |
| 6 | 8 | 3 |
| 5 | 9 | 6 |
| 4 | 1 | 6 |
| 5 | 8 | 4 |

T

| D | E | I |
|---|---|---|
| 6 | 6 | 9 |
| 8 | 7 | 8 |
| 8 | 5 | 6 |
| 3 | 8 | 9 |

图 3-13

（1）R∞S∞T

| A | B | C | D | E | I |
|---|---|---|---|---|---|
| 2 | 3 | 5 | 6 | 6 | 9 |
| 1 | 6 | 8 | 3 | 8 | 9 |
| 5 | 3 | 5 | 6 | 6 | 9 |
| 2 | 6 | 8 | 3 | 8 | 9 |

(2) R∝S

| A | B | C |
|---|---|---|
| 2 | 3 | 5 |
| 1 | 6 | 8 |
| 5 | 3 | 5 |
| 2 | 6 | 8 |

S∝R

| B | C | D |
|---|---|---|
| 3 | 5 | 6 |
| 3 | 5 | 9 |
| 6 | 8 | 3 |

S∝T

| B | C | D |
|---|---|---|
| 3 | 5 | 6 |
| 6 | 8 | 3 |
| 5 | 9 | 6 |
| 4 | 1 | 6 |

T∝R 为空

R∝T 为空

T∝S

| D | E | I |
|---|---|---|
| 6 | 6 | 9 |
| 3 | 8 | 9 |

3.7 如果习题 3.6 中的三个关系 R，S，T 分别位于三个不同的站点 X，Y，Z。若采用基于半连接算法计算连接 R∞S∞T，请选择使得传输代价最少的连接执行的站点和确定半连接序列。

假设每个属性域长度均为 1B，考虑所有的半连接

| 方案 | 半连接 | P | 得益 | 费用 | 传送属性 |
|---|---|---|---|---|---|
| P1 | R∞S | 2/3 | 1/3*3*6 | 2*5 | S.B S.C |
| P2 | S∞R | 1/2 | 1/2*3*6 | 2*4 | R.B R.C |
| P3 | S∞T | 2/3 | 1/3*3*6 | 1*3 | T.D |
| P4 | T∞S | 1/2 | 1/2*3*6 | 1*4 | S.D |

1、 选择得益高于 P2 进行优化，得到新的 R,S',T,并对受到影响的方案重新计算得意和费用。

新的 R,S',T 如下：

P

| A | B | C |
|---|---|---|
| 2 | 3 | 5 |
| 5 | 3 | 6 |
| 1 | 6 | 8 |
| 3 | 4 | 6 |
| 5 | 3 | 5 |
| 2 | 6 | 8 |

S'

| B | C | D |
|---|---|---|
| 3 | 5 | 6 |
| 3 | 5 | 9 |
| 6 | 8 | 3 |

T

| D | E | I |
|---|---|---|
| 6 | 6 | 9 |
| 8 | 7 | 8 |
| 8 | 5 | 6 |
| 3 | 8 | 9 |

对受到影响的方案重新计算得益和费用

| 方案 | 半连接 | P | 得益 | 费用 | 传送属性 |
|---|---|---|---|---|---|
| P1 | R∞S' | 2/3 | 1/3*3*6 | 2*5 | S'.B　S'.C |
| P2 | S∞R | 1/2 | 1/2*3*6 | 2*4 | R.B　R.C |
| P3 | S'∞T | 2/3 | 1/3*3*6 | 1*3 | T.D |
| P4 | T∞S' | 1/2 | 1/2*3*6 | 1*4 | S'.D |

2、 选择得益最高的 P4 进行优化，得到新的 R,S',T'，并对受到影响的方案重新计算得益和费用。

新的 R,S',T'如下：

| A | B | C |
|---|---|---|
| 2 | 3 | 5 |
| 5 | 3 | 6 |
| 1 | 6 | 8 |
| 3 | 4 | 6 |
| 5 | 3 | 5 |
| 2 | 6 | 8 |

| B | C | D |
|---|---|---|
| 3 | 5 | 6 |
| 3 | 5 | 9 |
| 6 | 8 | 3 |

| D | E | I |
|---|---|---|
| 6 | 6 | 9 |
| 3 | 8 | 9 |

对受到影响的方案重新计算得益和费用

| 方案 | 半连接 | P | 得益 | 费用 | 传送属性 |
|---|---|---|---|---|---|
| P1 | R∞S' | 2/3 | 1/3*3*6 | 2*5 | S'.B　S'.C |

| | | | | | |
|---|---|---|---|---|---|
| ~~P2~~ | ~~S∞R~~ | ~~1/2~~ | ~~1/2*3*6~~ | ~~2*4~~ | ~~R.B R.C~~ |
| P3 | S'∞T | 2/3 | 1/3*3*6 | 1*3 | T.D |
| ~~P4~~ | ~~T∞S'~~ | ~~1/2~~ | ~~1/2*3*6~~ | ~~1*4~~ | ~~S'.D~~ |

3、 依照 1 或者 2 的方法得到新的 R',S',T'如下：

R'= S'= T'

| A | B | C |
|---|---|---|
| 2 | 3 | 5 |
| 1 | 6 | 8 |
| 5 | 3 | 5 |
| 2 | 6 | 8 |

| B | C | D |
|---|---|---|
| 3 | 5 | 6 |
| 3 | 5 | 9 |
| 6 | 8 | 3 |

| D | E | I |
|---|---|---|
| 6 | 6 | 9 |
| 3 | 8 | 9 |

对受到影响的方案进行重新计算得益和费用得到：

| 方案 | 半连接 | P | 得益 | 费用 | 传送属性 |
|---|---|---|---|---|---|
| ~~P1~~ | ~~R∞S'~~ | ~~2/3~~ | ~~1/3*3*6~~ | ~~2*5~~ | ~~S'.B S'.C~~ |
| ~~P2~~ | ~~S∞R~~ | ~~1/2~~ | ~~1/2*3*6~~ | ~~2*4~~ | ~~R.B R.C~~ |
| P3 | S'∞T | 2/3 | 1/3*3*6 | 1*3 | T.D |
| ~~P4~~ | ~~T∞S'~~ | ~~1/2~~ | ~~1/2*3*6~~ | ~~1*4~~ | ~~S'.D~~ |

4、 选择得益最高的 P3 进行优化，得到 X，Y，Z 站点上最终的 R',S',T'..X,Y,Z 站点上最终的 R',S'',T'如下：

R'= S''= T'=

| A | B | C |
|---|---|---|
| 2 | 3 | 5 |
| 1 | 6 | 8 |
| 5 | 3 | 5 |
| 2 | 6 | 8 |

| B | C | D |
|---|---|---|
| 3 | 5 | 6 |
| 6 | 8 | 3 |

| D | E | I |
|---|---|---|
| 6 | 6 | 9 |
| 3 | 8 | 9 |

所以选择各站点做连接的代价：

X 站点代价=2*3+2*3=12

Y 站点代价=4*3+2*3=18

Z 站点代价=4*3+2*3=18

故选 X 站点作为收集站点代价最低。

由简化过程得知半连接过程为：

1. S'=S∝R
2. 将 S'传送给 T，做半连接 T∝S'得到 T'。
3. 将 S'传送给 R，做半连接 R∝S'得到 R'.
4. 将 T'传送给 S'，做半连接 S' ∝T'得到 S"

即：（R∝(S∝R)）∞((S∝R) ∝(T∝(S∝R)) ∞(T∝(S∝R)）

（1）在站点 Y 上作关系 S 在 R 和 S 公共属性集 B，C 上的投影 π[B,C](S)；把该结果送到站点 X；

（2）在站点 X 上计算半连接，设其结果为 R',则 R'＝R∝S;

（3）在站点 Z 上作关系 T 在 S 和 T 公共属性 D 上的投影 π[D](T)；把结果送到站点 X；

（4）在站点 Y 上计算半连接，设其结果为 S'，则 S'＝S∝T；结果送到站点 X；

（5）在站点 X 上计算半连接，得到最后结果。

3.8 设某公司的雇员关系为 employee(name，address，salary，plant-number)，按 plant-number 水平分片这个关系，每个片段都有两个副本：一个副本存放在 New York 站点，另一个副本存放在工厂的站点。请为在 Toronto 站点提出的下列查询设计一个好的处理策略。

（1）找出 Boce 厂的所有雇员。

（2）找出所有雇员的平均工资。

（3）找出在如下每个站点工资最高的雇员姓名：Toronto，Edmonton，Vancouver，Montreal。

（4）找出该公司中工资最低的雇员姓名。

**答：**（1）找出 Boce 厂的所有雇员。

直接从 Boce 的 employee 分段中投影出 employee 的 name 属性，再将其发送到 Toronto 站点，呈现给所需用户。

（2）找出所有雇员的平均工资。

在 New York 站点将所有分段数据中的 employee 的 salary 相加，取平均值，计算完毕之后再将该值发送到 Toronto 站点。

（3）找出在如下每个站点工资最高的雇员姓名：Toronto，Edmonton，Vancouver，Montreal。

对各个站点的各自的工资进行降序排列（以 salary 的值为标准），选出 salary 的 max 值，对其 name 属性做投影，将数据发送到 Toronto 站点。

（4）找出该公司中工资最低的雇员姓名。

各站点中降序排列（salary），选出各站点的 min（salary）发送到 Toronto 站点，再将各自的 salary 作比较，对最小值的 employee 的 name 属性做投影。

（1）找出 Boce 厂的所有雇员。

答：将 New York 站点上的副本传至 Toronto 站点

（2）找出所有雇员的平均工资。

答：在 New York 站点上球平均工资，传至 Toronto 站点

（3）找出在如下每个站点工资最高的雇员姓名：Toronto，Edmonton，Vancouver，Montreal。

答：Toronto，Edmonton，Vancouver，Montreal 求最高工资，传至 Toronto 汇总。

（4）找出该公司中工资最低的雇员姓名。

答：各站点中降序排列（salary），选出各站点的 min（salary）发送到 Toronto 站点，再将各自的 salary 作比较，对最小值的 employee 的 name 属性做投影。

# 第四章

4.7 请用自己的语言描述两阶段提交的过程。

第一阶段：表决阶段，目的是形成一个共同的决定。协调者向所有参与者发出“准备提交”信息。如果某个参与者准备提交，就回答“就绪” 信息，否则回答撤销信息。参与者在回答之前应把有关信息写入子自己的日志中。协调者在发出准备提交信息前也要把有关信息写入自己的日志中。如果在规定时间内协调者收到了所有参与者“就绪”的信息，则将做出提交的决定，否则撤销。

第二阶段：执行阶段，目的是实现这个决定，协调者将有关决定的信息先写入日志，然后把这个决定发送给所有参与者。参与者收到命令之后首先往日志中写入“收到提交或撤销”决定的信息，并向协调者发送“应答”消息，最后执行有关决定。协调者收到所有参与者的应答消息后，一个事务的执行到此结束，有关日志信息可以脱机保存。

4.8 为什么说两阶段提交协议在不丢失运行日志信息的情况下，可从任何故障恢复？

因为在执行过程中维护了事务日志，记录了执行恢复所需要的信息。

4.9 在分布式数据库系统中对多副本数据的更新通常采用什么方法？快照方法的优点和缺点是什么？

答：对多副本数据更新通常有主文本更新法、移动主文本法和快照方法。

快照方法的优点是：可完成复杂的查询，而又不阻止更新；不必考虑数据的辅文本，只关心每一数据的主文本和在这些主文本上定义的任意多个快照；避免了某些并发控制的开销，又便于复杂查询完成，可提高系统可用性。

快照方法的缺点是：快照为了与主文本保持同步，不许定时刷新，快照是一个制度关系，其中数据只能读而不能写，对更新操作无效。

# 第五章

5.2 描述分布式事务的可串行化理论的一些定义：事务、冲突操作、并发调度、串行调度、

一致性调度、两个调度等价、可串行化调度。

**事务**：在分布式系统中，事务是一个分布式操作的序列，被操作的数据分布在不同的站点上。

**冲突操作**：如果两个操作 P 和 Q，对同一个数据 X 操作，其中至少有一个是写操作 W(X)则 P 和 Q 称为冲突操作。

**并发调度**：并发事务的一个调度简称并发事务。

**串行调度**：若一个调度 S，其每个事务的执行均有 Ti<Tj，即事务 Ti 所有操作都先于事务 Tj 操作，每个事务相继执行，这样的调度 S 成为串行调度。

**一致性调度**：执行一个调度可以使得数据库从一个一致性状态转变为另一个一致性状态，则称调度为一致性调度。

**可串行化调度**：如果一个调度等价于某个串行调度，则该调度称为可串行调度。

**两个调度等价**：冲突等价以及视图等价。

5.5 什么是两阶段封锁协议？它如何保证可串行性？为什么人们经常更愿意采用严格两阶段封锁和严酷两阶段封锁？

**两阶段封锁协议** ： 一个事务所有的封锁操作（读写）都在第一个解锁操作之前，则该事务遵守两阶段封锁协议。这样一个事务被分成两个阶段：

上升阶段(成长阶段)：只能获取新锁，而不能释放已有的锁

收缩阶段(衰退阶段)：只能释放已有的锁，而不能获得新锁

保守 2PL：要求事务在开始执行前就持有所有它要访问的数据项上的锁。

严格 2PL：事务提交或撤销之前，绝对不释放任何一个写锁；在事务结束时，同时释放所有的锁。

严酷 2PL：事务在提交或撤销之前，不能释放任何一个锁。

**如何保证可串行性**：如果事务 T 稍后必须封锁数据项 Y，哪么在它使用完数据项 X 之后，获得数据项 Y 之前，不可以释放数据项 X 上的锁。因此 T 必须一直持续有 X 的锁，直到该事务需要读或写的所有数据项都被他自己封锁，然后，T 才可以释放 X 上的锁。透明合适，即时 T 已经完成 X，另一个要访问 X 的事务也可能会被强制等待。同样对于数据项 Y，如果 T 封锁了 Y 则其他事务也要等待。因此就保证了可串行性。

**为什么人们经常更愿意采用严格两阶段封锁和严酷两阶段锁**，因为在严格两阶段封锁中事务 T 在提交或撤销之前，绝对的不释放任何一个排他锁;在事务结束时，同时释放所有锁。因此，除非事务 T 已经提交，否则，任何其他的事务都不可以读或写由事务 T 所写的数

据项，从而产生了一个对可恢复性而言的调度。改变有数据处理器所发出的操作命令不分，这对于仅当操作被提交或撤销时保证锁的什邡市很必要的。而在严酷 2PL 中，事务 T 在提交或撤销之前，不鞯释放任何一个锁，因此比严格两阶段锁更容易实现。

# 第六章

6.3 概述分布式可靠性协议的组成以及它们各自的使用范围。

答：分布式数据库系统的可靠性协议包括提交协议、终结协议和恢复协议

提交协议：详细说明了提交协议时如何被执行的，它保持了分布式事务的原子性。

终结协议：在执行一个分布式事物时，其中一个站点失效了，我们希望其他站点也停止该事务，终结协议是用来处理这种情况的。

恢复协议：一个站点失效了，终结协议确定可未失效站点如何处理该失效事件，而恢复协议确定失效站点重新启动后，其进程恢复他的状态的过程。

6.4 讨论两阶段提交协议的终结协议和两阶段提交协议的恢复协议。

答：两阶段提交协议的终结协议：终结协议在协调者和参与者的定时器超时发挥作用，超时发生在目的站点在期望的时间内没有从发送站点得到所期望的消息时。处理超时的方法依赖于失效发生的时间和失效的类型。协调者可以再三种状态中发生超时：等待、提交和撤销。在后两种状态发生超时是按相同的方式处理的，所以只需考虑两种情况：

① 等待状态发生超时时，协调者正在等待参与者的局部决定。协调者不能单方面提交事务，因为不满足全局提交规则。然而，他可以决定全局撤销事务，此时，他在日志中写入撤销记录，并向所有参与者发送“全局撤销”消息。

② 在提交状态或撤销状态发生超时时，协调者不能确定是否在所有参与者站点上本地恢复管理程序都执行完提交或撤销过程。因此。协调者重复发出“全局提交”命令或“全局撤销”，命令给没有响应的站点，并等待确认。

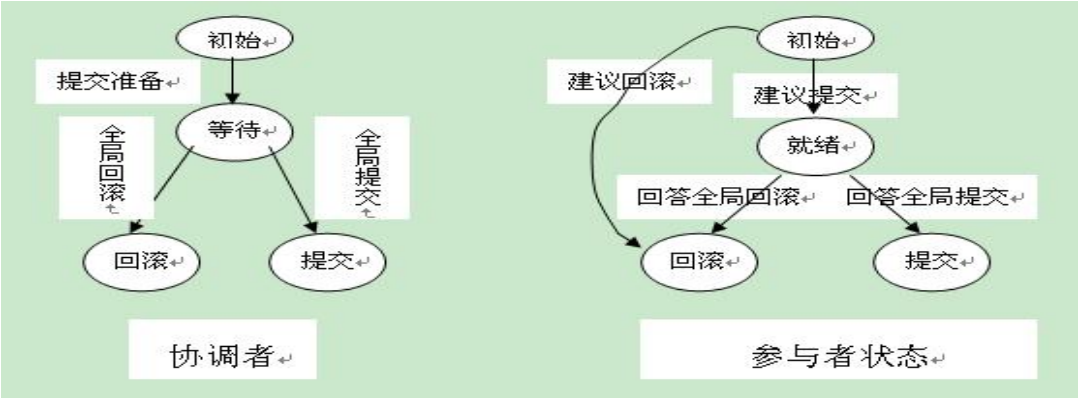
两阶段提交协议的恢复协议可分为三种情形：协调者站点失效，参与者站点失效，还有其他的附加情形。

① 协调者站点失效：1）协调者在初始状态失效发生在协调者初始化提交过程之前，

因此，它将在恢复启动时提交过程。2）协调者在等待状态失效时，协调者已经发送了“准备”命令，恢复时，协调者将从头开始启动提交过程，再次发送“准备”消息。3）协调者在提交状态或撤销状态失效时，协调者可能已经把它的决定通知了参与者，并终结了事务。于是在恢复时如果它已经收到所有的确认消息，它就不需要做任何事情。否则，就要启动终结协议。

② 参与者站点失效：1）一个参与者在初始状态失效，在恢复时，该参与者应该单方面撤销事务。2）一个参与者在就绪状态失效时，协调者已经收到失效站点在失效前发送的肯定性决定，在恢复时，失效站点的参与者认为是在就绪状态发生了超时，于是启动终结协议来处理该事务。3）一个参与者在提交状态或撤销状态失效，在恢复时，参与者不需要采取任何专门的措施。

③ 附加情形：1）协调者在开始提交记录写入日志之后，“准备”消息发送之前失效。协调者按照在等待失效时的情况处理，在恢复时重新发送“准备”命令。2）一个参与者在就绪记录写入日志之后，发送“建议提交”消息之前失效，失效的参与者把这种情况按上面讨论的参与者失效情形 2）对待。3）一个参与者在撤销记录写入日志之后，发送“建议撤销”消息之前失效。这是唯一一种没有被前面讨论的基本情形包含情况，然而，在这种情况下，参与者在恢复时不需要做任何事情。协调者处于等待状态并超时，并在改状态下的终结协议全局撤销该事务。4）协调者在它最后的决定写入日志之后，在给参与者发送“全局撤销”消息或“全局提交”消息之前失效，协调者按第三种情形处理该情况，参与者按在就绪状态超时的情形处理。5）一个参与者在撤销记录或者提交记录写入日志之后，在给协调者发送确认消息之前失效。参与者按第三种情形处理该情况，协调者按在提交状态或撤销状态超时的情形处理。



**两阶段提交协议的终结协议：**

1.协议者超时

（1）等待状态发生超时：在等待状态，协调者正在等待参与者的局部决定，协调者不能单方面得提交事务，因为不满足全局提交规则。然而，它可以决定全局撤销事务，此时，它在日志中写入撤销记录，并向所以投参与者发送“全局撤销”消息

（2）在提交状态或撤销状态发生超时：在这种情况下，协调者不能确定是否在所有参与者站点上恢复管理程序都执行完提交所有或撤销过程。因此，协调者重复发出“全局提交”命令或“全局撤销”命令给没响应得站点，并等待确认。

2.参与者超时

（1）在初始状态发生超时：可以单方面撤销事务。如果参与者稍后收到“准备”消息，他可以按两种可能处理方式之一处理。或者参与者在检查日志的时候发现了撤销记录，就送回一条“建议撤销”消息。

（2）在就绪状态发生超时：参与者在该状态下正提交事务，但不知道协调者的全局决定，参与者不能单方面做出决定。

**两阶段提交协议的恢复协议：**

1.协调者站点失效

（1）协调者在初始状态失效：它将在恢复时启动提交过程

（2）在等待状态失效：协调者从头开始启动提过程，再次发送“准备”

在提交状态或者撤销状态

在恢复时如果它已经收到所有的确认消息他就不需要做任何事情。否则，就要启动终结协议。

2.参与者站点失效

（1）在初始状态失效：在恢复时请参与者单方面撤销事务

（2）在就绪状态失效：失效站点的的参与者启动终结协议来处理

（3）提交或撤销状态失效：不需要爱去任何专门的措施

6.5 什么是三阶段提交协议？讨论三阶段提交协议的终结协议和三阶段提交协议的恢复协议。

**三阶段提交协议**:在 2PC 的等待状态和提交状态之间增加一个状态，作为一个缓冲，用

于在准备提交但是还没有提交的时候。因为从初始状态到提交状态之间有三次状态转换，所以称为三阶段提交协议。

**三阶段提交协议的终结协议：**

1.协调者超时

在等待状态时超时，协调者让该事物全局回滚。

在准备提交状态超时，协调者给参与者发送“准备提交”消息，使他们进入准备提交状态并将提交记录记入日志。

在提交状态超时，此时，终结协议应该通过选举新的协议者来终结事务。

在回滚状态超时，此时，终结协议应该通过选举新的协议来终结事务。

2.参与者超时

在初始状态时，参与者单方面放弃。

在就绪状态超时，通过选举新的协议来终结事务。

在准备提交状态超时，通过选举新的协议来终结事务。

**三阶段提交协议的恢复协议：**

1.协议者在等待状态失效

参与者已经终结的事物，隐藏，在恢复时，协议者必须向他们询问以确定如何终结事务。

2.协议者在预备提交状态失效

终结协议再一次知道可操作参与者的终结事务。由于在这个过程中可以从预备提交状态转换到撤销状态，协调者必须询问以确定如何终结事物。

3.一个参与者在预备提交状态失效

它必须询问以确定其他参与者如何终结事物。