第3章

分布式数据库设计

在分布式数据库系统设计中，最基本的问题就是数据的分布问题，即如何对全局数据进行逻辑划分和实际物理分配。数据的逻辑划分称为数据分片。由于数据分片是通过关系代数的基本运算来实现的，因此本章首先对关系代数进行简要介绍；然后介绍按照自上而下的设计策略进行的数据分布设计，主要包括分片的定义和作用、水平分片和垂直分片、分片的设计原理以及分片的表示方法和分配设计模型，并以关系数据库为例加以说明。本章内容是进行分布式数据库设计的基础。

3.1 关系数据库管理系统的关系运算

20 世纪 80 年代以后推出的 DBMS 几乎都支持关系数据模型，非关系系统的产品也大都增加了与关系模型的接口。关系数据库不仅简单，而且还有严谨的数学理论基础。关系数据模型用二维表格表示现实世界实体集及实体集之间的联系。关系模式由一个关系名以及它的所有属性名构成。一般形式是 $R(A_1,A_2,\cdots,A_n)$，其中 R 是关系名，$A_1,A_2,\cdots,A_n$ 是该关系的属性名。

一个关系数据库是多个关系的集合，这些具体关系构成了关系数据库的实例。由于每个关系都有一个模式，所以，构成该关系数据库的所有关系模式的集合构成了关系数据库模式。在关系数据库中，对数据库的查询和更新操作都归结为对关系的运算。

关系代数是一种抽象的查询语言，是关系数据操纵语言的一种传统表达方式，它是用对关系的运算来表达查询的。掌握好关系代数，将有助于提高读者思考问题的能力，写出正确、高效的查询。

任何一种运算都是将一定的运算符作用于一定的运算对象上，得到预期的运算结果。所以，运算对象、运算符、运算结果是运算的三大要素。关系运算按其表达查询方式不同分为关系代数和关系演算。关系代数的运算按运算符的不同可分为传统的集合运算和专门的关系运算两类。传统的集合运算包括并、交、差和笛卡儿积；专门的关系运算包括选择部分数据的运算和组合两个关系的操作。其中传统的集合运算将关系看成元组的集合，其运算是从关系的“水平”方向即行的角度进行。而专门的关系运算不仅涉及行而且涉及列。比较运

算符和逻辑运算符是用来辅助专门的关系运算符进行操作的。图 3-1 列出了各运算的关系。

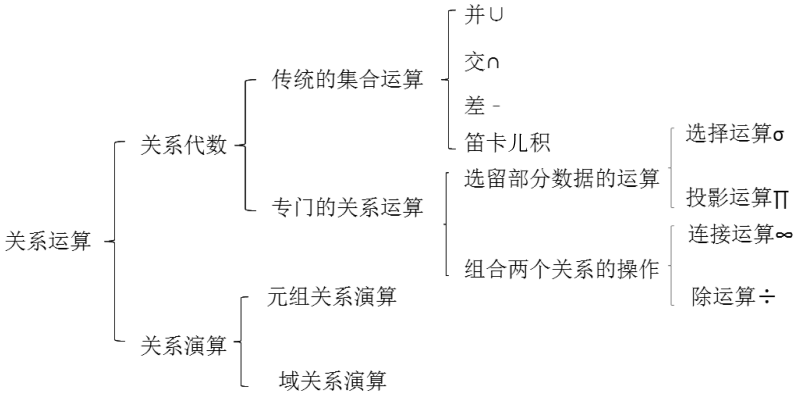

图 3-1　各运算的关系

关系代数的运算对象是关系，运算结果亦为关系。关系代数用到的运算符包括集合运算符(∪、∩、-、×)、关系运算符(σ、∏、∞、÷)、逻辑运算符(¬、∧、∨)、算术比较运算符(>、≥、<、≤、=、≠)。

关系是二维表，实际上是表中的元组(记录、行)的集合。无论是传统的集合运算，还是专门的关系运算，它们都是面向集合的操作，即参与运算的运算量是关系(集合)，运算得到的结果也是关系(集合)。

3.1.1　传统的集合运算

传统的集合运算包括并、交、差和笛卡儿积，表 3-1 和表 3-2 分别列出了两个学生基本信息表 S1 和 S2，下面以此为例讲解并、交、差运算。

表 3-1　学生基本信息表 S1

学　号	姓名	性别	年级	学　院	专　业
0501001	张昊	男	2020	计算机	计算机科学与技术
0501010	李颖	女	2020	计算机	计算机科学与技术
0501206	王婷	女	2020	计算机	计算机科学与技术

表 3-2　学生基本信息表 S2

学　号	姓名	性别	年级	学　院	专　业
0501008	赵娜	女	2020	计算机	计算机科学与技术
0501019	李浩	男	2020	计算机	计算机科学与技术
0501206	王婷	女	2020	计算机	计算机科学与技术

1. 并

假设关系 R 和关系 S 的并(union)运算产生一个新的关系 R',则 R'由属于关系 R 或 S 的所有不同元组组成,记为 $R'=R\cup S$。其中 R 和 S 的属性个数相同,且相应属性分别有相同的值域。$R\cup S$ 由如图 3-2 所示的部分元组组成。

S1∪S2 的结果如表 3-3 所示,它由 S1 和 S2 去掉重复元组后的所有元组组成。

表 3-3 学生基本信息表 S1∪S2

学 号	姓名	性别	年级	学 院	专 业
0501001	张昊	男	2020	计算机	计算机科学与技术
0501010	李颖	女	2020	计算机	计算机科学与技术
0501206	王婷	女	2020	计算机	计算机科学与技术
0501008	赵娜	女	2020	计算机	计算机科学与技术
0501019	李浩	男	2020	计算机	计算机科学与技术

2. 交

假设关系 R 和关系 S 的交(intersection)运算产生一个新的关系 R',则 R'由既属于 R 又属于 S 的元组组成,记为 $R'=R\cap S$。其中 R 和 S 的属性个数相同,且相应属性分别有相同的值域。$R\cap S$ 由如图 3-3 所示两圆相交部分的元组组成。

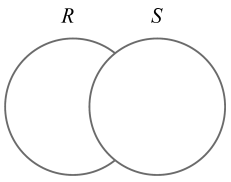

图 3-2 $R\cup S$

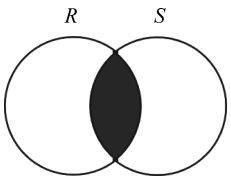

图 3-3 $R\cap S$

S1∩S2 的结果如表 3-4 所示,它由关系 S1 和关系 S2 中完全相同的元组构成。

表 3-4 学生基本信息表 S1∩S2

学 号	姓名	性别	年级	学 院	专 业
0501206	王婷	女	2020	计算机	计算机科学与技术

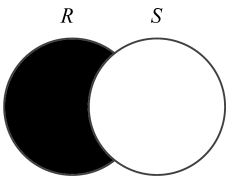

图 3-4 $R-S$

3. 差

假设关系 R 和关系 S 的差(difference)运算产生一个新的关系 R',则 R'由属于 R 但不属 S 的元组组成,记为 $R-S$。其中 R 和 S 属性个数相同,且相应属性分别有相同的值域。$R-S$ 的组成如图 3-4 所示。

表 3-5 给出的是表 3-1 和表 3-2 中两个表 S1 和 S2 进行差运算后的结果。它由属于 S1 但不属于 S2 的元组构成。

表 3-5　学生基本信息表 S1－S2

学　号	姓名	性别	年级	学　院	专　业
0501001	张昊	男	2020	计算机	计算机科学与技术
0501010	李颖	女	2020	计算机	计算机科学与技术

4. 笛卡儿积

设 R 为 m 元关系，S 为 n 元关系，R 和 S 的笛卡儿积(Cartesian product)产生一个新关系 R'，记为 $R'=R\times S$。R' 由 R 和 S 的所有元组连接而成的具有 $m+n$ 个分量的元组组成，新关系中元组的前 m 个分量为 R 的一个元组，后 n 个分量为 S 的一个元组。

设学生信息表 S3 和课程信息表 S4 分别如表 3-6 和表 3-7 所示。则 S3×S4 的结果如表 3-8 所示，它的每个元组由 3+3 个属性构成。

表 3-6　学生信息表 S3

学　号	姓　名	年　级
0501001	张昊	2020
0501010	李颖	2020

表 3-7　课程信息表 S4

课程代码	课程名称	教　室
0501001	数据库原理	B01
0501002	C 语言	B02

表 3-8　S3×S4

学　号	姓　名	年　级	课程代码	课程名称	教　室
0501001	张昊	2020	0501001	数据库原理	B01
0501001	张昊	2020	0501002	C 语言	B02
0501010	李颖	2020	0501001	数据库原理	B01
0501010	李颖	2020	0501002	C 语言	B02

3.1.2　专门的关系运算

1. 选择运算

选择运算是一个单目运算，它从一个关系 R 中选取满足给定条件的元组构成一个新的关系，选择运算记为 $\sigma_F(R)=\{t\mid t\in R\wedge F(t)='真'\}$，其中 σ 是选择运算符，F 表示选择条件，是由逻辑运算符$\neg$、$\wedge$、$\vee$等连接算术表达式组成的条件表达式。$F(t)$是一个逻辑表达式，结果取逻辑值'真'或'假'。

算术表达式的基本形式为 $X\theta Y$，其中 X、Y 是属性名、常量或简单函数，属性名也可以用它的序号来代替。θ 是比较运算符，$\theta\in\{>、\geqslant、<、\leqslant、=、\neq\}$。

选择运算实际上是从关系 R 中选取使逻辑表达式 F 为真的元组。这是从行的角度进行的运算。

例如,从学生信息表 S1 中选择学号为 0501010 的学生,表示为 $\sigma_{学号=\text{“0501010”}}(S1)$,其结果由 S1 表中满足学号=“0501010”的元组构成,如表 3-9 所示。

表 3-9 $\sigma_{学号=\text{“0501010”}}(S1)$结果关系表

学 号	姓 名	性 别	年 级	学 院	专 业
0501010	李颖	女	2020	计算机	计算机科学与技术

2. 投影运算

投影运算也是一个单目运算,它从一个关系 R 中选取所需要的列组成一个新关系,投影运算记为:$\Pi_A(R)=\Pi_{i_1,i_2,\cdots,i_k}(R)=\{t[A]\mid t\in R\}$。其中$\Pi$是投影运算符,$A$ 为关系 R 属性的子集,$t[A]$为 R 中元组相应于属性集 A 的分量,$i_1,i_2,\cdots,i_k$ 表示 A 中属性在关系 R 中的顺序号。

投影运算是从列的角度进行的运算,投影取消了原关系中的某些列后,可能出现重复行,投影后也会取消这些完全相同的重复行。

例如从学生信息表 S1 中投影学号、姓名、年级,表示为$\Pi_{学号,姓名,年级}(S1)$,其结果如表 3-10 所示。

表 3-10 $\Pi_{学号,姓名,年级}(S1)$结果关系表

学 号	姓 名	年 级
0501001	张昊	2020
0501010	李颖	2020
0501206	王婷	2020

3. 连接运算

从两个关系 R 和 S 的广义笛卡儿积中选取满足给定条件 F 的元组组成新的关系的操作称为 R 和 S 的连接(join),其形式如下:

JON 关系名 1 AND 关系名 2　WHERE 条件

记作 $R\underset{F}{\infty}S$,其中,条件 $F=A\theta B$ 是由算术比较符 $\theta\in\{>、\geqslant、<、\leqslant、=、\neq\}$和属性名或列号组成的条件表达式。$A$ 和 B 分别代表 R 的第 A 列和 S 的第 B 列属性。

当连接运算的条件为等号时,连接称为等值连接。连接后的结果包括 R 和 S 的所有字段,即结果中有重复字段。

当连接运算中的比较符为“=”,且参与比较的两个关系中用于比较的两个属性相同时,该连接称为自然连接(natural join),自然连接运算所产生的新关系由参与连接运算的两个关系中的所有属性组成,但在两个关系中都含有的作为等值比较对象的两个属性只

出现一次，所以它不同于一般的等值连接。对于自然连接，无须标明条件表达式 F，只需在结果中把重复的属性去掉，如关系 R 和关系 S 的自然连接记为 $R \infty S$。

假设有学生基本信息表 S5 和学生选课信息表 S6 分别如表 3-11 和表 3-12 所示，则 S5 和 S6 的自然连接 S5∞S6 结果关系如表 3-13 所示。

表 3-11　学生基本信息表 S5

学　号	姓　名	年　级
0501001	张昊	2020
0501010	李颖	2020
0501206	王婷	2020

表 3-12　学生选课信息表 S6

学　号	课　程	教　室
0501001	数据库	B01
0501030	C 语言	B02
0501010	C 语言	B02

表 3-13　S5∞S6 结果关系表

学　号	姓名	年级	课程	教室
0501001	张昊	2020	数据库	B01
0501010	李颖	2020	C 语言	B02

4. 除运算

除运算的含义是给定关系 $R(X,Y)$ 和 $S(Y,Z)$，其中 X、Y、Z 为属性组。R 中的 Y 与 S 中的 Y 可以有不同的属性名，但必须出自相同的域集。R 与 S 的除运算得到一个新的关系 $P(X)$，P 是 R 中满足下列条件的元组在 X 属性列上的投影，元组在 X 上分量值 x 的象集 Yx 包含 S 在 Y 上投影的集合。

例如有两个关系 R1 和 R2，结构与元组分别如表 3-14 和表 3-15 所示，则 R1÷R2 的结果关系如表 3-16 所示。

表 3-14　R1

A	B	C	A	B	C
a1	b1	c2	a4	b6	c6
a2	b3	c7	a2	b2	c3
a3	b4	c6	a1	b2	c1
a1	b2	c3			

表 3-15　R2

B	C	D
b1	c2	d1
b2	c1	d1
b2	c3	d2

表 3-16 R1÷R2

A
a1

R1÷R2 分析,在关系 R1 中,A 可以取 4 个值{a1,a2,a3,a4},其中:

a1 的象集为:{(b1,c2),(b2,c3),(b2,c1)};

a2 的象集为:{(b3,c7),(b2,c3)};

a3 的象集为:{(b4,c6)};

a4 的象集为:{(b6,c6)};

R2 在(B,C)上的投影为{(b1,c2),(b2,c3),(b2,c1)}。

显然只有 R1 的象集 a1 包含 R2 在(B,C)属性组上的投影,所以 R1÷R2={a1}。

3.2 设计方法与分布设计的目标

分布式数据库的设计有两种设计方法:一种是自上而下(Top-Down)的设计方法;另一种是自下而上(Bottom-Up)的设计方法。Bottom-Up 设计方法是多数据库集成的核心研究内容,分布式数据库的设计主要是与 Top-Down 设计方法相关的内容。

3.2.1 Top-Down 设计过程

Top-Down 设计过程是从需求分析开始,进行概念设计、分布设计、物理设计以及性能调优等一系列设计过程。Top-Down 设计过程是系统从无到有的设计与实现过程,适用于新设计一个数据库系统,如图 3-5 所示。

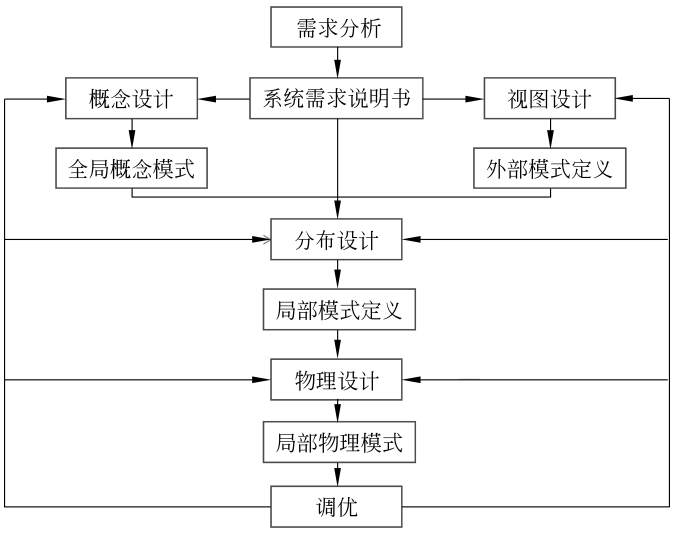

图 3-5 Top-Down 设计过程

第一步,系统需求分析。首先,根据用户的实际应用需求进行需求分析,形成系统需

求说明书。该系统说明书是所要设计和实现系统的预期目标。

第二步，依据系统需求说明书中的数据管理需求进行概念设计，得到全局概念模式，如 E-R 模型。同时根据系统说明书中的应用需求，进行相应的外模式定义。

第三步，依据全局概念模式和外模式定义，结合实际应用需求和分布式设计原则，进行分布设计，包括数据分片和分配设计，得到局部概念模式以及全局概念模式到局部概念模式的映射关系。

第四步，依据局部概念模式实现物理设计，包括片段存储、索引设计等。

第五步，进行系统调优。确定系统设计是否最好地满足系统需求，包括同用户沟通、系统性能模式测试等，可能需要进行多次反馈，以使系统能最佳地满足用户的需求。

3.2.2　Bottom-Up 设计过程

Bottom-Up 设计方法适合于已存在的多个数据库系统，并需要将它们集成为一个数据库的设计过程，Bottom-Up 设计方法属于典型的数据库集成的研究范围，有关异构数据库集成方法中，有基于集成器或包装器的数据库集成策略和基于联邦的数据库集成策略等。构建模式间映射关系的基本方法主要有两种，即 GAV(Global As View)方法和 LAV(Local As View)方法。

下面给出一种基于集成器的多数据库集成系统的设计过程，如图 3-6 所示。首先，各异构数据库系统经过相应的包装器转换为统一模式的内模式；接着集成器将各内模式集成为全局概念模式，集成过程中需要定义各内模式到全局模式的映射关系以及解决模式间的异构问题；最后，全局概念模式即为采用 Bottom-Up 设计策略设计的分布式数据库系统的全局概念模式。

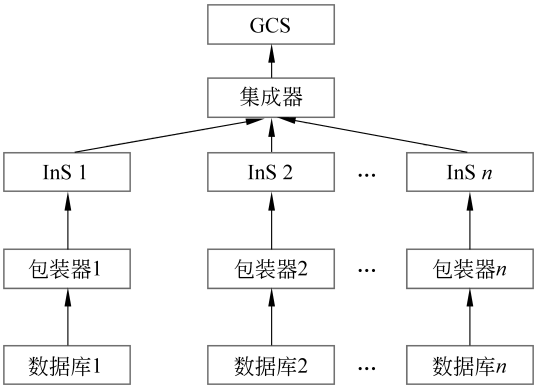

图 3-6　Bottom-Up 设计过程

3.2.3　数据库分布设计的目标

分布式数据库设计中数据库分布设计包括数据库分片设计与片段分配设计，这两者是紧密相关的。数据库分布设计应考虑以下目标。

1. 降低费用

使用数据库的单位在组织上往往是分布的(部门、科室),在地理上也是分布的。分布式数据库系统的结构符合这种分布的要求。允许用户在自己的本地录用、查询、维护等操作,实行局部控制,降低通信代价,提高响应速度。

2. 提高系统可靠性

将数据分布于多个场地,并增加适当的冗余度可以提供更好的可靠性。在一些可靠性要求高的系统中,这一点尤其重要。这避免了因为某个场地的故障而造成全部瘫痪的后果。

3. 处理局部性

数据分布应以尽量满足局部操作为主,即使得大部分操作在局部场地完成。这就要求划分数据,并将数据片段尽量放置在访问它们最频繁的场地或最接近的场地上,以减少通信开销。可以按存取方式将应用分成局部存取和远程存取两类,一旦应用的原场地已知,则存取的局部性和远程性只依赖于数据的分布。最好的完全局部化应用是请求完全在原场地执行的应用。若处理局部性高,则系统的可用性与可靠性也高,而且能减少远程通信与系统控制代价,缩短响应时间,从而提高系统性能。

4. 易于扩展处理能力和系统规模

当一个企业增加了新的部门时,分布式数据库系统的结构可以很容易地扩展系统。在分布式数据库中增加一个新的节点,不影响现有系统的正常运行。

5. 负载分布

合理地分配负载于网络的各个场地,以便能充分发挥各地计算机的能力和提高各应用执行的并行度。负载分布与处理局部性可能相冲突,所以在数据分布设计时必须全面权衡。

分片的定义及分类

3.3 分片的定义及分类

数据分片是指将 DDB 的全局关系划分成相应的逻辑片段(逻辑关系)。数据分片有利于按照用户的需求较好地组织数据的分布,也有利于控制数据的冗余,下面对数据分片的有关概念做一介绍。

3.3.1 分片的定义和作用

分布式数据库中数据的存储单位称为片段。对全局数据库的划分叫作分片。划分的结果就是片段。每个片段可以保存在一个以上的场地(服务器)上。

对数据进行分片存储,便于分布地处理数据,对于提高分布式数据库系统的性能至关

重要。分片的主要作用体现在以下 4 个方面。

1. 减少网络传输量

网络上的数据传输量是影响分布式数据库系统中数据处理效率的主要代价之一，为减少网络上的数据传输代价，分布式数据库中的数据允许复制存储，目的是可就近访问所需数据副本，减少网络上的数据传输量。因此，在数据分配设计时，设计人员需要根据应用需求，将频繁访问的数据分片存储在尽可能近的场地上，减少网络上的数据传输量。

2. 增大事务处理的局部性

数据分片按需分配在各自的局部场地上，可并行执行局部事务，就近访问局部数据，减少数据访问的时间，增强局部事务的处理效率。

3. 提高数据的可用性和查询效率

就近访问数据分片或副本，可提高访问效率。同时当某一场地出现故障时，若存在副本，非故障场地上数据副本均可使用，保证了数据的可用性和完整性以及系统的可靠性。

4. 负载均衡

有效利用局部数据处理资源，就近访问局部数据，可以避免访问集中式数据库所造成的数据访问瓶颈，有效提高整个系统效率。

3.3.2　分片设计过程

分片过程是将全局数据进行逻辑划分和实际物理分配的过程。全局数据分片成各个片段数据，各个片段分配到不同的场地（服务器）上。分片设计过程从全局数据库→片段数据库→物理数据库。分片过程如图 3-7 所示。其中 GDB（Global DB）为全局数据库，FDB（Fragment DB）为片段数据库，PDB（Physical DB）为物理数据库。分片模式定义从全局模式到片段模式的映射关系，分配模式定义从片段模式到物理模式的映射关系，1∶N 时为复制，1∶1 时为分割。

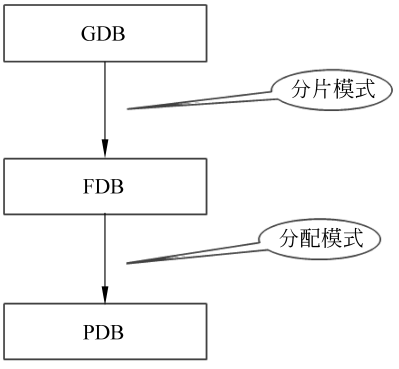

图 3-7　分片过程

3.3.3　分片原则

在设计分布式数据库时，设计者必须考虑数据如何分布在各个场地上，也就是全局数据应该如何进行逻辑划分和物理划分。哪些数据应该分布式存放？哪些数据不需要分布式存放？哪些数据需要复制？对系统进行全盘考虑，使系统性能最优。但是无论如何分片都应该遵循以下原则。

1. 完备性条件

必须把全局关系的所有数据映射到片段中,不允许有属于全局关系的数据却不属于它的任何一个片段。

2. 可重构条件

划分所采用的方法必须确保能够由全局关系的各个片段来重建该全局关系。重构条件的必要性是显而易见的。事实上,仅是全局关系的各个片段(而不是全局关系本身)存储在分布式数据库中。因此,一旦需要,必须能够通过这种重构操作,用全局关系的各个片段来重建该全局关系。

3. 不相交条件

要求一个全局关系被分割后所得的各个数据片段互不重叠。之所以要施加这个限制,其目的是为了在数据分配时易于控制数据的复制。

3.3.4 分片的类型

分布式数据库系统按系统实际需求对全局数据进行分片和物理分配,分片种类有以下 4 种。

1. 水平分片

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。

2. 垂直分片

把一个全局关系的属性集分成若干子集,并在这些子集上进行投影运算,每个投影称为垂直分片。

3. 导出分片

导出分片又称为导出水平分片,即水平分片的条件不是该关系属性的条件,而是其他关系属性的条件。

4. 混合分片

混合分片是水平分片、垂直分片、导出分片的混合。可以先进行水平分片,再进行垂直分片,或先进行垂直分片,再进行水平分片,或采用其他形式,但它们的结果是不相同的。

3.3.5 分布式数据库数据分布透明性

分布透明性的定义:指用户或用户程序使用分布式数据库如同使用集中式数据库那

样，不必关心全局数据的分布情况，包括全局数据的逻辑分片情况、逻辑片段的站点位置分配情况，各站点数据库的数据模型等情况对用户和用户程序是透明的。

分布透明性的三个层次。

1. 分片透明性

分片透明性是分布透明性中的最高层，位于全局概念模式与分片模式之间。

2. 位置透明性

位置透明性是分布透明性的中间层，位于分片模式和分配模式之间。

3. 局部数据模型透明性

局部数据模型透明性是分布透明性的最底层，位于分配模式与局部概念模式之间。

3.4　水 平 分 片

水平分片是按照一定的条件对全局关系元组的划分，即把全局关系的所有元组划分成若干不相交的子集。

3.4.1　水平分片的概念

定义 3.1：设有一个关系 R，$\{R_1,R_2,\cdots,R_n\}$为 R 的子关系的集合，如果$\{R_1,R_2,\cdots,R_n\}$满足以下条件，则称其为关系 R 的水平分片，称 R_i 为 R 的一个水平片段。

(1) $R_1,R_2,\cdots,R_n$ 与 R 具有相同的关系模式。

(2) $R_1\cup R_2\cup\cdots\cup R_n=R$。

(3) $R_i\cap R_j=\phi(i\neq j,1\leqslant i\leqslant n,1\leqslant j\leqslant n)$。

从水平分片的定义可以看出，所谓水平分片，就是按某种特定条件把一全局关系的所有元组划分成若干不相交的子集。每个水平片段由关系中的某个属性上的条件来定义，该属性称为分片属性，该条件称为分片条件。不相交的子集满足完备性条件、可重构条件和不相交条件。

例 3.1　有一个全局关系模式为 student(snum,name,college)，其中 snum 为学生编号，name 为学生姓名，college 为学生所在的学院，并假定学生所在的学院只有两个，即“计算机”和“数学”。按下面的条件进行水平分片：

student1：满足 college＝"计算机"的所有元组

student2：满足 college＝"数学"的所有元组

在该分片中 college 为分片属性，分为两个片段 student1 和 student2，用选择操作可以表示为

$$\text{student1}=\sigma_{\text{college}=\text{“计算机”}}(\text{student})$$

$$\text{student2}=\sigma_{\text{college}=\text{“数学”}}(\text{student})$$

全局关系 student 的这种水平分片如图 3-8 所示。student 的水平分片 student1、

student2 满足完备性条件、可重构条件和不相交条件。

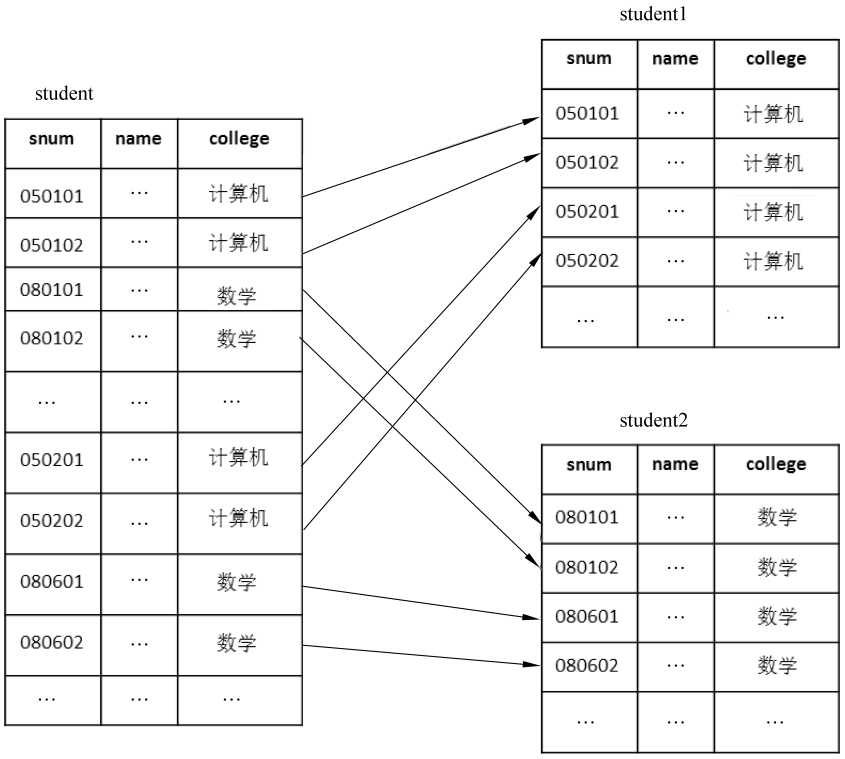

图 3-8 student 的水平分片

(1) 满足完备性条件。

由于“计算机”与“数学”是 college 属性的所有可能取值,所以上面的分片无疑是满足完备性条件的。如果 college 的属性还可能有其他取值,则上述的分片就不满足完备性条件。因为这些其他 college 值的元组属于全局关系 student,但既不属于 student1 也不属于 student2。

(2) 满足可重构条件。

重构条件是易于验证的,因为总是能通过下列运算来重构 student 全局关系:student= student1∪student2。

(3) 满足不相交条件。

student 的水平分片 student1 和 student2 总是满足不相交条件的。因为 snum 作为全局关系 student 的关键字,它唯一地标识了一个学生。这个学生的 college 值或取“计算机”,或取“数学”,因此,student 关系中的每一个元组只能分在一个片段中。

通过例 3.1 可见,为了对一个全局关系 student 进行水平划分,要通过对它施加选择运算来实现。我们把该选择运算中所使用的谓词叫作相应片段的限定(也称分片谓词)。例如在上面的例子中,其限定如下:

q1: college="计算机"

q2: college="数学"

由此，可得出如下结论，即为了满足完备性条件，所有片段的限定集合必须是完全的（或至少关于所允许的值的集合是完全的）。重构条件总是能通过并运算予以满足，不相交条件要求各限定必须是互斥的。

例 3.2　设有雇员关系 EMP{ENO，ENAME，SALARY，DNO}，其中 ENO 为雇员编号，ENAME 为雇员姓名，SALARY 为雇员工资，DNO 为雇员所在部门的部门编号。其元组如表 3-17 所示。

表 3-17　EMP 关系表

ENO	ENAME	SALARY	DNO
001	张颖	2000	101
002	李强	3000	201
003	王丽	4000	301

按下面的条件进行水平分片：

E1：满足 DNO＝"101"的所有元组

E2：满足 DNO＝"201"的所有元组

E3：满足 DNO＝"301"的所有元组

雇员关系 EMP 的水平分片 E1、E2、E3 用选择操作描述如下：

$$E1=\sigma_{DNO="101"}(EMP)$$

$$E2=\sigma_{DNO="201"}(EMP)$$

$$E3=\sigma_{DNO="301"}(EMP)$$

从上面的分片可知，将关系 EMP 分成了三个子关系，部门编号 DNO 等于 101 的元组 E1、部门编号 DNO 等于 201 的元组 E2、部门编号 DNO 等于 301 的元组 E3。

分片属性为部门编号 DNO。

分片条件如下：

E1：DNO＝"101"

E2：DNO＝"201"

E3：DNO＝"301"

各子关系的内容分别如表 3-18 至表 3-20 所示。

表 3-18　子关系 E1 的内容

ENO	ENAME	SALARY	DNO
001	张颖	2000	101

表 3-19　子关系 E2 的内容

ENO	ENAME	SALARY	DNO
002	李强	3000	201

表 3-20 子关系 E3 内容

ENO	ENAME	SALARY	DNO
003	王丽	4000	301

根据水平分片的定义,满足:

(1) E1、E2、E3 和 EMP 具有相同的关系模式;

(2) E1∪E2∪E3=EMP;

(3) E1∩E2=ϕ,E1∩E3=ϕ,E2∩E3=ϕ。

因此,E1、E2、E3 是 EMP 的水平分片。

3.4.2 水平分片的操作

水平分片是针对该关系的选择操作,用 σ 表示,假设选择条件为分片谓词 q,则关系 R 的分片操作可表示为 $\sigma_q(R)$。

例 3.1 的水平分片,具体操作可以表示如下:

$$student1=\sigma_{college="计算机"}(student)$$
$$student2=\sigma_{college="数学"}(student)$$

对应的 SQL 语句可以表示如下:

```
student1: SELECT * FROM student WHERE college="计算机"
student2: SELECT * FROM student WHERE college="数学"
```

例 3.2 的水平分片,具体操作可以表示如下:

$$E1=\sigma_{DNO="101"}(EMP)$$
$$E2=\sigma_{DNO="201"}(EMP)$$
$$E3=\sigma_{DNO="301"}(EMP)$$

对应的 SQL 语句可以表示如下:

```
E1: SELECT * FROM EMP WHERE DNO="101"
E2: SELECT * FROM EMP WHERE DNO="201"
E3: SELECT * FROM EMP WHERE DNO="301"
```

3.4.3 水平分片的原理

对全局进行水平分片时,必须遵守完备性、可重构性和不相交性条件,以保证分布式数据库中数据的完整性和一致性。由于全局关系的水平分片可以由选择运算中的限定的集合(即谓词集)唯一决定,因此,谓词集 P 也必须遵守完备性、可重构性和不相交性条件。

如上所述,全局关系的水平分片是由选择运算来决定的,而选择运算的谓词都是基于全局关系的属性,所以,全局关系中某些属性根据其值的不同可以构成关系的水平分片。

显然，这些属性具有分类的作用。同一片段中每一元组对于这些属性来说都具有相同的性质。

定义 3.2：若全局关系 R 中属性 X 具有地理分布特征或属性 X 的域的任一划分都构成全局关系的元组的不同的聚集，则称属性 X 具有分类特征。

定义 3.3：若全局关系 R 中的属性 X 满足：

(1) DOM(X)是可数有限集合；

(2) 属性 X 具有分类特征；

则称属性 X 为关系 R 的分类属性。

例 3.3　设有以下几个全局关系：

全局关系模式 student(snum，name，college)，其中 snum 为学生编号，name 为学生姓名，college 为学生所在的学院。

雇员关系 EMP(ENO，ENAME，SALARY，DNO)，其中 ENO 为雇员编号，ENAME 为雇员姓名，SALARY 为雇员工资，DNO 为雇员所在部门的部门编号。

EMP 关系中的 DNO、student 关系中的 college 都符合分类属性的定义，可以认为是合适的分类属性。而 name 在通常情况下就不是分类属性。

通过对分类属性域的划分，即选择运算的限定条件包含分类属性的所有域值，可以唯一地确定全局关系的水平分片。因此，对水平分片的讨论可以转换为对水平分片谓词集的讨论。

命题 3.1：对于关系 R 的水平分片谓词集 P，如果对 P 中出现的分类属性集$\{X_1, X_2, \cdots, x_n\}$的域 DOM($X_1$)，DOM($X_2$)，…，DOM($X_n$)构成划分，则谓词集 P 对分类属性集$\{X_1, X_2, \cdots, x_n\}$是完备的。

显然，要使命题成立，首先要求谓词集 P 中出现的所有属性都是分类属性，另外如果属性中存在着相关性的话，则对域构成划分就是总体意义上的，而不是局部意义上的，因为对单个域都构成划分时，在其总和域上不一定有这种性质。

命题 3.2：如果谓词集 $P=\{P_1, P_2, \cdots, P_n\}$中的谓词两两互斥，即 $P_i \wedge P_j = \text{FALSE}$ $(i \neq j)$，且 $Pi(1 \leqslant i \leqslant n)$不为永假，则每一谓词 P_i 都构成一个片段。

很明显，如果谓词两两互斥，则按照这种谓词进行选择运算的结果一定是不相交的，即以不同谓词选择的结果中无相同元组。谓词不为永假，保证了选择运算的结果不会永远为空。

例 3.4　student(snum，name，college)中属性 college 的域为 DOM(college)＝{"计算机","数学"}，则 student 的水平分片可以划分如下：

$$\text{student1} = \sigma_{\text{college}=\text{"计算机"}}(\text{student})$$

$$\text{student2} = \sigma_{\text{college}=\text{"数学"}}(\text{student})$$

在这里，谓词如下：

P_1：college＝"计算机"

P_2：college＝"数学"

谓词集 $P=\{P_1, P_2\}$。

显然，谓词集 P 对分类属性 college 是完备的且 P_1 和 P_2 互斥，谓词 P_1 和 P_2 构成

了两个片段 student 1 和 student 2。

定理 3.1 如果谓词集 $P=\{P_1,P_2,\cdots,P_n\}$是基于关系 R 中分类属性集$\{X_1,X_2,\cdots,X_n\}$ 的,且 P 中的谓词两两互斥并对$\{X_1,X_2,\cdots,X_n\}$是完备的,则谓词集 P 决定关系 R 的一种水平分片。

证明: 设谓词 $P_1,P_2,\cdots,P_n$ 构成的关系 R 的各片段为 $R_1,R_2,\cdots,R_n$。

(1) 显然,$R_1,R_2,\cdots,R_n$ 与 R 具有相同的关系模式。

(2) 因谓词集 P 对分类属性集$\{X_1,X_2,\cdots,x_n\}$是完备的,故对任意元组 $t\in R$,必定存在而且只存在某一 $P_i(1\leqslant i\leqslant n)$为真,使得 $t\in\sigma_{P_i}(R)$,即 $t\in R_i$ 成立。因此有 $R=\sigma_{P_1}(R)\cup\sigma_{P_2}(R)\cup\cdots\cup\sigma_{P_n}(R)$,即 $R=R_1\cup R_2\cup\cdots\cup R_n$;

(3) 因 P 中谓词是两两互斥的,故 $\sigma_{P_i}(R)\cap\sigma_{P_j}(R)=\phi(i\neq j)$,即任意两个片段的交运算的结果为空:$R_i\cap R_j=\phi(i\neq j,i,j=1,2,\cdots,n)$。

因此,$R_1,R_2,\cdots,R_n$ 是关系 R 的水平分片,即谓词集 P 决定关系 R 的水平分片。

从定理的证明中可以看出,由谓词集 P 决定的水平分片符合完备性、可重构性和不相交性规则。另外,定理还指出了对全局关系的水平分片存在着多种不同的谓词集,从而说明全局关系的水平分片存在着不止一种。

水平分片谓词集 P 可以由下述方法生成:

(1) 根据查询模型选取关系 R 中合适的分类属性集$\{X_1,X_2,\cdots,X_n\}$,并确定各自的域 DOM(X_1)、DOM(X_2)、…、DOM(X_n);

(2) 根据查询对分片的要求,选取一个适当的谓词 P_1,令 $P=\{P_1\}$;

(3) 选取新的适当谓词 P_i,P_i 与 P 中谓词互斥,置 $P\leftarrow P\cup\{P_i\}$,直至 P 构成 DOM(X_1)、DOM(X_2)、…、DOM(X_n)的划分。

例 3.5 对全局关系 teacher{tnum,name,age,sex,college}进行水平划分,假定选取 age 和 college 为分类属性,设:

$$\text{DOM(age)}=\{20,21,\cdots,60\}$$

$$\text{DOM(college)}=\{\text{"计算机","数学"}\}$$

选取谓词:

$$P_1:\ \text{age}\leqslant 40$$

$$P_2:\ 40<\text{age}\leqslant 50\wedge\text{college}=\text{"计算机"}$$

$$P_3:\ 40<\text{age}\leqslant 50\wedge\text{college}=\text{"数学"}$$

$$P_4:\ \text{age}>50$$

从 teacher 关系的定义与说明可知,谓词集 $P=\{P_1,P_2,P_3,P_4\}$构成 DOM(age)和 DOM(college)的划分,故 P 对分类属性集{age, college}是完备的。谓词集 $P=\{P_1,P_2,P_3,P_4\}$中谓词是两两互斥的,因此谓词集 P 确定了全局关系 teacher 的水平分片如下:

$$\text{teacher1}=\sigma_{P_1}(\text{teacher})$$

$$\text{teacher2}=\sigma_{P_2}(\text{teacher})$$

$$\text{teacher3}=\sigma_{P_3}(\text{teacher})$$

$$teacher4=\sigma_{P_4}(teacher)$$

实际上，如已构成期望的谓词集 P'：$(P_1,P_2,\cdots,P_{n-1})$，但 P'对分类属性集$\{X_1,X_2,\cdots,X_n\}$不是完备的，则可取 $P_n=\neg P_1\vee\neg P_2\vee\cdots\vee\neg P_{n-1}$，那么 $P=P'\cup\{P_n\}$对分类属性集$\{X_1,X_2,\cdots,X_n\}$一定是完备的。这种方法是比较方便和行之有效的，而且也适用于分类属性域是无限集的情况。

3.5　导出水平分片

若一个关系的分片不是基于关系本身的属性，而是根据另一个与其有关联的属性来划分，这种划分为导出水平划分。

3.5.1　导出水平分片的概念

定义 3.4：如果一个关系的水平分片的分片属性属于另一个关系，则该分片称为导出水平分片。

例 3.6　有雇员关系 EMP{ENO，ENAME，SALARY，DNO}，其中 ENO 为雇员编号，ENAME 为雇员姓名，SALARY 为雇员工资，DNO 为雇员所在的部门编号。其元组如表 3-17 所示。关系 WORKS{ENO，PRJNO，HOURS}，其中 ENO 为雇员编号，PRJNO 为雇员参与的项目编号，HOURS 为雇员参与项目的小时数，其元组如表 3-21 所示。

表 3-21　WORKS 元组内容

ENO	PRJNO	HOURS
001	1	200
002	1	300
003	2	500

要求将 WORKS 按 DNO 进行水平分片，得到的导出水平分片记为 W1、W2、W3，要求如下：

W1：满足 DNO="101"的所有元组，即 $W1=\Pi_{ATTR(WORKS)}(\sigma_{DNO="101"}(WORKS\infty EMP))$

W2：满足 DNO="201"的所有元组，即 $W2=\Pi_{ATTR(WORKS)}(\sigma_{DNO="201"}(WORKS\infty EMP))$

W3：满足 DNO="301"的所有元组，即 $W3=\Pi_{ATTR(WORKS)}(\sigma_{DNO="301"}(WORKS\infty EMP))$

其中，ATTR(WORKS)为 WORKS 的属性组。

分片属性为部门编号 DNO。

分片条件为

W1：DNO="101"

W2：DNO="201"

W3：DNO="301"

各子关系的内容分别如表 3-22 至表 3-24 所示。

表 3-22 子关系 W1 的内容

ENO	PRJNO	HOURS
001	1	200

表 3-23 子关系 W2 的内容

ENO	PRJNO	HOURS
002	1	300

表 3-24 子关系 W3 的内容

ENO	PRJNO	HOURS
003	2	500

根据水平分片的定义,满足:

(1) W1、W2、W3 和 WORKS 具有相同的关系模式;

(2) $W1 \cup W2 \cup W3 = WORKS$;

(3) $W1 \cap W2 = \phi$, $W1 \cap W3 = \phi$, $W2 \cap W3 = \phi$。

因此,W1、W2、W3 满足完备性条件、可重构条件和不相交条件,是 WORKS 的水平分片。由于该分片属性为 DNO,是 WORKS 关系相关联关系 EMP 的属性,因此该水平分片为导出水平分片。

3.5.2 导出水平分片的操作

导出水平分片的操作不是基于关系本身的属性,而是根据另一个与其有关联关系的属性来划分的。因此,导出水平分片可以用连接操作和选择操作来表示。

例 3.6 中的导出水平分片,具体操作表示如下。

(1) 求出 WORKS 中的 DNO,采用自然连接∞。

令 $W' = WORKS \infty EMP$, $W' = (ENO, PRJNO, HOURS, ENAME, SALARY, DNO)$。

(2) 根据 DNO 对 W′进行水平分片。

$$W1' = \sigma_{DNO="101"}(W') = \sigma_{DNO="101"}(WORKS \infty EMP)$$

$$W2' = \sigma_{DNO="201"}(W') = \sigma_{DNO="201"}(WORKS \infty EMP)$$

$$W3' = \sigma_{DNO="301"}(W') = \sigma_{DNO="301"}(WORKS \infty EMP)$$

(3) 只保留 WORKS 的属性。

$$W1 = \Pi_{ATTR(WORKS)}(W1') = \Pi_{ATTR(WORKS)}(\sigma_{DNO="101"}(WORKS \infty EMP))$$

$$W2 = \Pi_{ATTR(WORKS)}(W2') = \Pi_{ATTR(WORKS)}(\sigma_{DNO="201"}(WORKS \infty EMP))$$

$$W3 = \Pi_{ATTR(WORKS)}(W3') = \Pi_{ATTR(WORKS)}(\sigma_{DNO="301"}(WORKS \infty EMP))$$

3.5.3　导出水平分片的作用

在两个关系间存在相关属性并满足关联完整性约束时，一个关系的水平分片常常可以导出另一个关系的水平分片，导出分片可以用来简化片段间的连接运算。

数据分布状况下的连接称为分布连接，如果关系 R 和 S 均有水平分片时，R 和 S 做连接运算就需要比较 R 和 S 中的所有元组。因此，从原则上讲，比较 R 中全部片段 R_i 与 S 中全部片段 S_j 是需要的。但当某一 R_i 与 S_j 的相关属性不相交时，部分连接 $R_i \infty S_j$ 为空。

分布连接可以用连接图表示。例如关系 R 和 S 的连接可用图 $G(N, E)$ 表示，其中 N 是节点集，表示 R 和 S 中的所有片段，E 是边集。每条边表示 R 和 S 的一个部分连接。为简单起见，在节点集中不包括孤立节点。图 3-9 给出了一个连接图的例子。

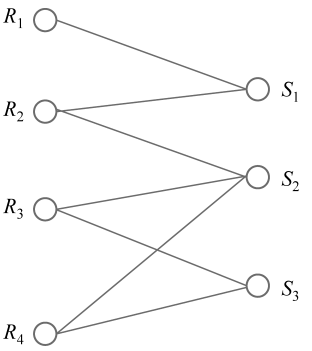

图 3-9　连接图

当连接图的边集包括了所有节点间的边时，该连接图是完全的。当某些 R 的片段与某些 S 的片段间的边省略时，则连接图是简化的，所表示的连接为简化连接。

简化连接有两种特殊的情形：

(1) 如果连接图是由两个以上不相连的子图组成的，则称连接图是分割的(partitioned)，其连接为分割连接，如图 3-10 所示。

(2) 如果分割的连接图中的每一个子图都只有一条边，则称为简单的连接图，其连接称为简单连接，如图 3-11 所示。

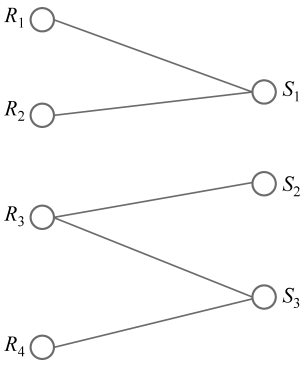

图 3-10　分割连接

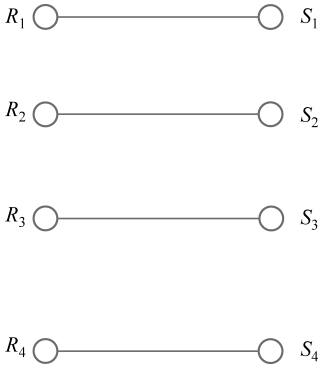

图 3-11　简单连接

在分布式数据库设计中，确定全局关系中的连接是简单连接或是分割连接是十分重要的，这可以降低分布连接的代价。尤其对于简单连接，如把需要连接的两片段都放在相同的场地上，可大大增加处理的局部性和减少网络传输量。

导出水平分片可以简化分布连接。根据导出分片的定义，如果关系 R 的分片是由关系 S 的水平片段导出的，则有 $R_i = R \infty S_i$。当导出分片的定义要求满足时，连接 $R \infty S$