

分布式数据库系统

讲解人：李鸿岐

分布式数据库的设计

分布式数据库的设计

□ 在系统设计中，最基本问题就是数据的分布问题，即如何对分布式数据库进行逻辑划分和实际物理分配。数据的逻辑划分称数据分片。本部分主要针对数据分布进行介绍，以关系数据库为例来说明。

- 两种设计策略（Top-down 和 Bottom-up）

- 分片的定义

- 水平分片

- 水平分片的设计

- 垂直分片的设计

- 分片的表示方法

- 分配设计

- 数据复制技术

- 大数据库的分布存储策略

适于新设计一个数据库系统

已存在许多数据库系统，并将它们集成为一个数据库。典型的数据集成研究

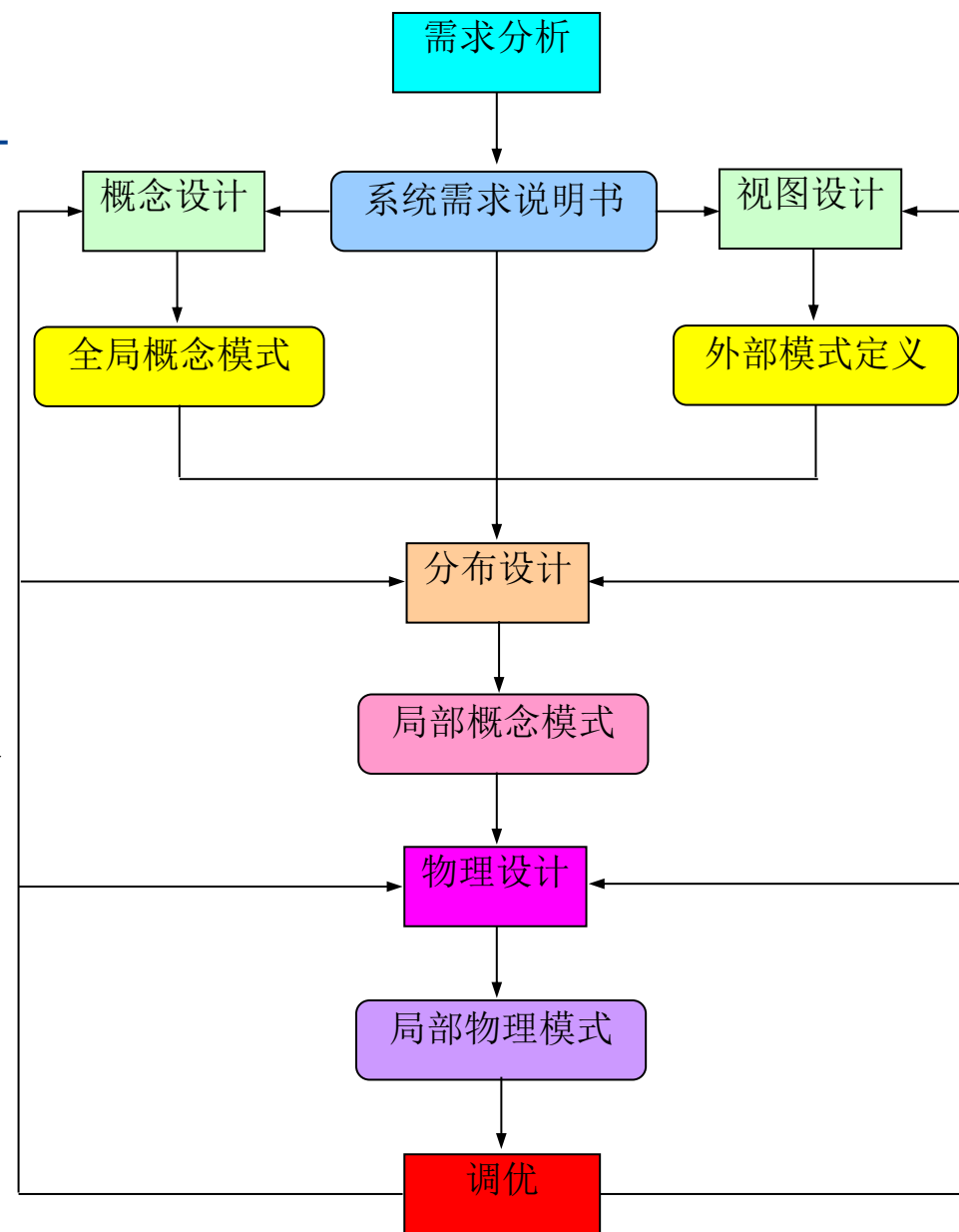
分布式数据库的设计

● 两种设计策略

□ Top-down 设计过程

基于五步递归实现：

1. 需求分析（**Requirement analysis**）
2. 概念设计、视图集成、**E-R** 表示、转换到关系模式
3. 分布设计、数据分片和分配
4. 物理设计
5. 性能调优（**Tuning**）

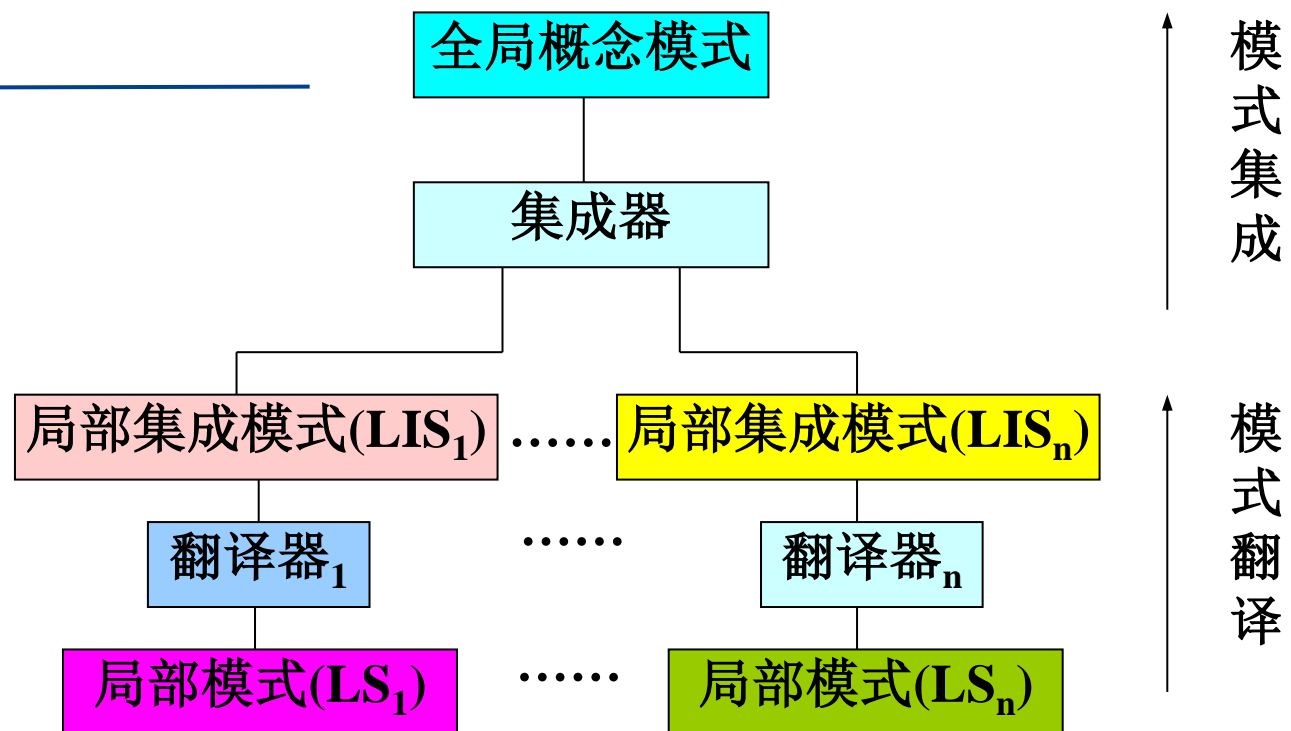
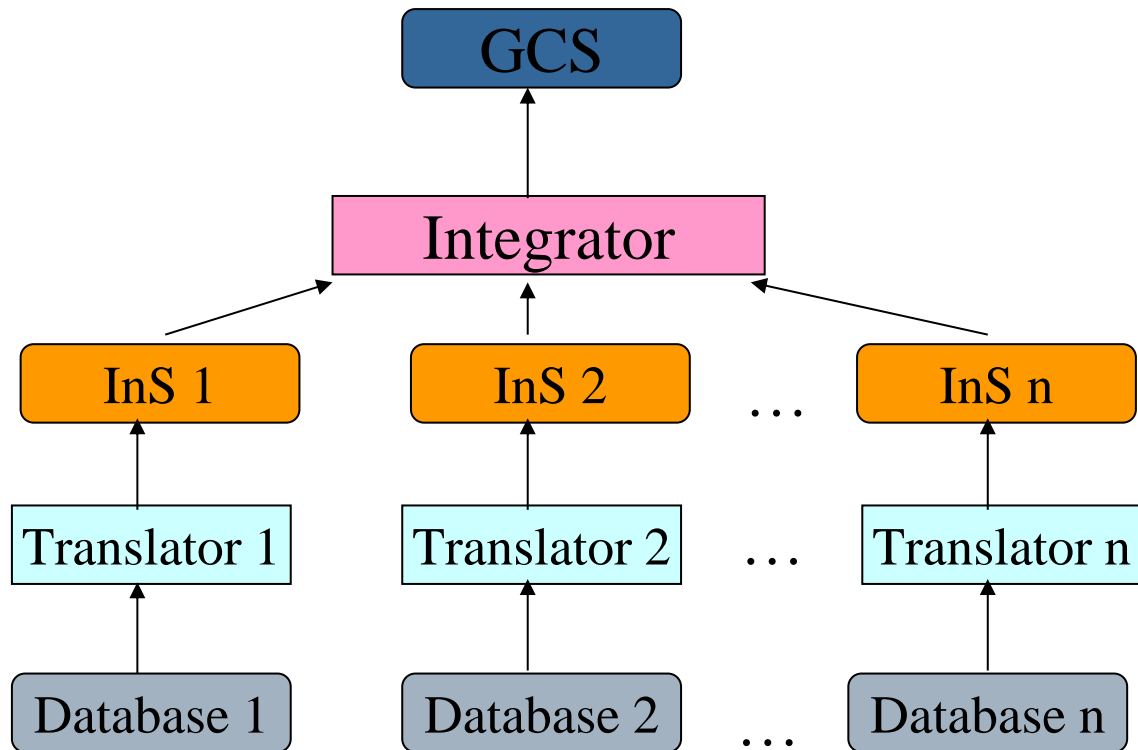


□ Top-down

分布式数据库的设计

● 两种设计策略

□ Bottom-up

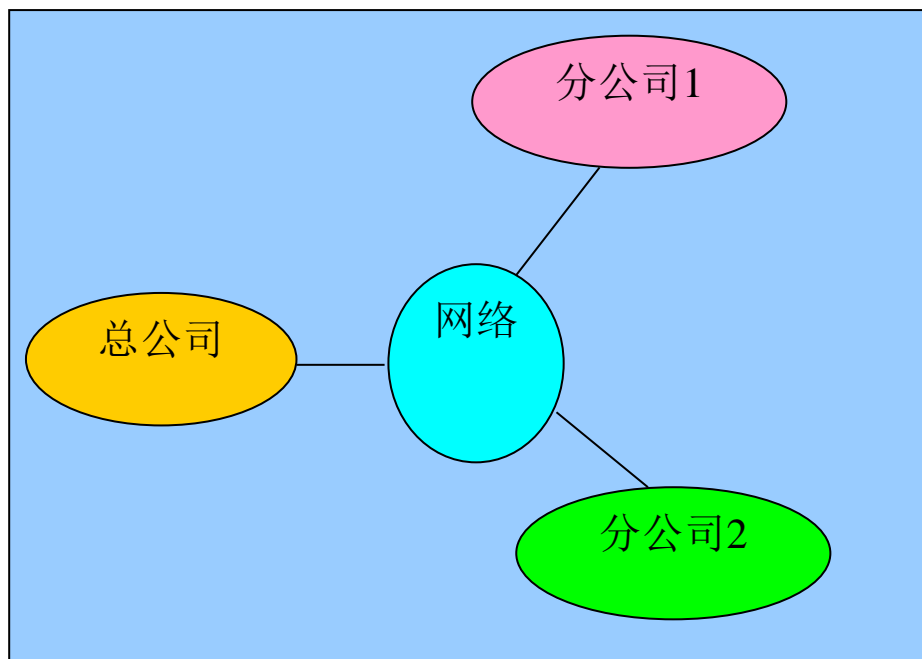


- 多数据库集成系统
- 数据和数据库系统已存在
- 支持集成的访问各数据库中的异构数据
- 通过限制数据管理能力（如限定只读）
可将局部数据库系统数量扩展到数百个

分布式数据库的设计

● 分片的定义

- 例：某集团公司由地理位置分别在不同城市的总公司和两个下属分公司组成，彼此之间靠网络相连接，业务管理由分布式数据库系统完成。其网络结构图如图所示：



假设：人事系统中，有职工关系，定义为：EMP {ENO, EName, Salary, DNO}。

场地定义：总公司为场地0，职工关系为EMP0

分公司1为场地1，职工关系为EMP1

分公司2为场地2，职工关系为EMP2

EMP=EMP0+EMP1+EMP2为全局数据

分布式数据库的设计

● 分片的定义

□ 数据分布要求

方案1： 公司总部保留全部数据。

方案2： 各单位只保留自己的数据。

方案3： 总公司保留全部数据，各分公司只保留自己单位的数据。

系统采用以上不同方案，对应需求不同的数据分配方案。如：

	方案1	方案2	方案3
场地0	总部（EMP）	总部（EMP0）	总部（EMP）
场地1	分公司1（无）	分公司1（EMP1）	分公司1（EMP1）
场地2	分公司2（无）	分公司2（EMP2）	分公司2（EMP2）

□ 上面三种方案中，除方案1外，均须进行分片定义。方案3中，分公司的数据信息除本场地存储外，总部场地也存储一份相同信息，存在在不同场地上互称副本。

分布式数据库的设计

● 分片的定义

□ 定义

- 片段 (**Fragment**)：是分布式数据库中数据的存储单位。
- 分片 (**Fragmentation**)：对全局数据的划分。
- 片段：划分的结果称为片段。
- 分配 (**Allocation**)：对片段存储场地的指定，称为分配。当片段存储在一个以上场地时，称为数据复制 (**Replication**) 存储。如果每个片段只存储在一个场地，称为数据分割 (**Partition**) 存储。

分布式数据库的设计

● 分片的定义

□ 分片的作用

- **减少网络传输量。**如：采用数据复制，可就近访问所需信息。需频繁访问的数据分片存储在本地场地上。
- **增大事务处理的局部性。**局部场地所需的数据分片分配在各自的场地上，减少数据访问的时间，增强局部事务效率。
- **提高数据的可用性和查询效率。**就近访问数据分片或副本，可提高数据的访问效率。同时当某一场地发生故障时，若存在副本，非故障场地上的数据副本均是可用的。保证了数据的可用性、数据的完整性和系统的可靠性。
- **使负载均衡。**减少数据访问瓶颈，提高整个系统效率。

分布式数据库的设计

$$GDB = \sum FDB_i$$

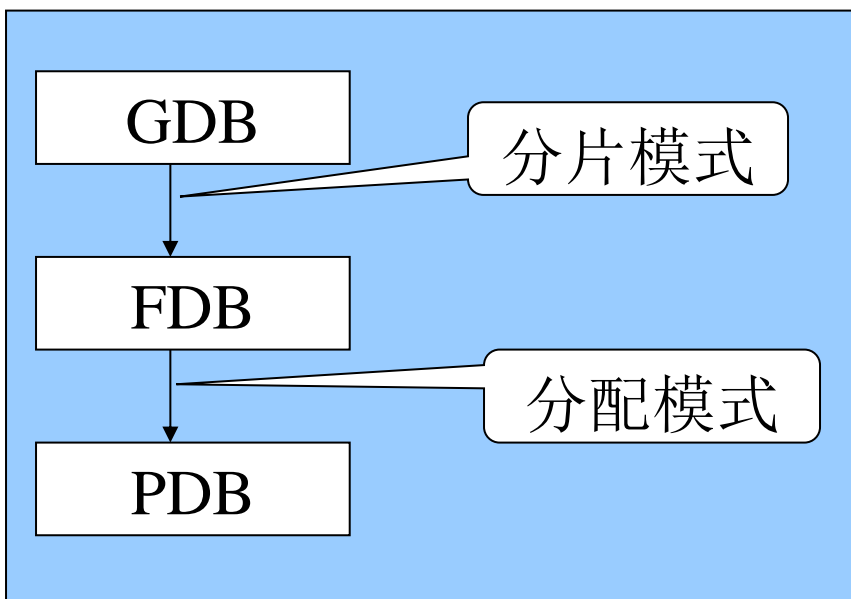
● 分片的定义

$$F(GDB) = FDB \Leftrightarrow F^{-1}(FDB) = GDB$$

□ 分片设计过程

$$P(FDB) = PDB \Leftrightarrow P^{-1}(PDB) = FDB$$

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。全局数据由分片模式定义分成各个片段数据，各个片段数据由分配模式定义存储在各场地上。分片过程如下图所示。



GDB: 全局数据库 (Global DB)

FDB: 片段数据库 (Fragmentation DB)

PDB: 物理数据库 (Physical DB)

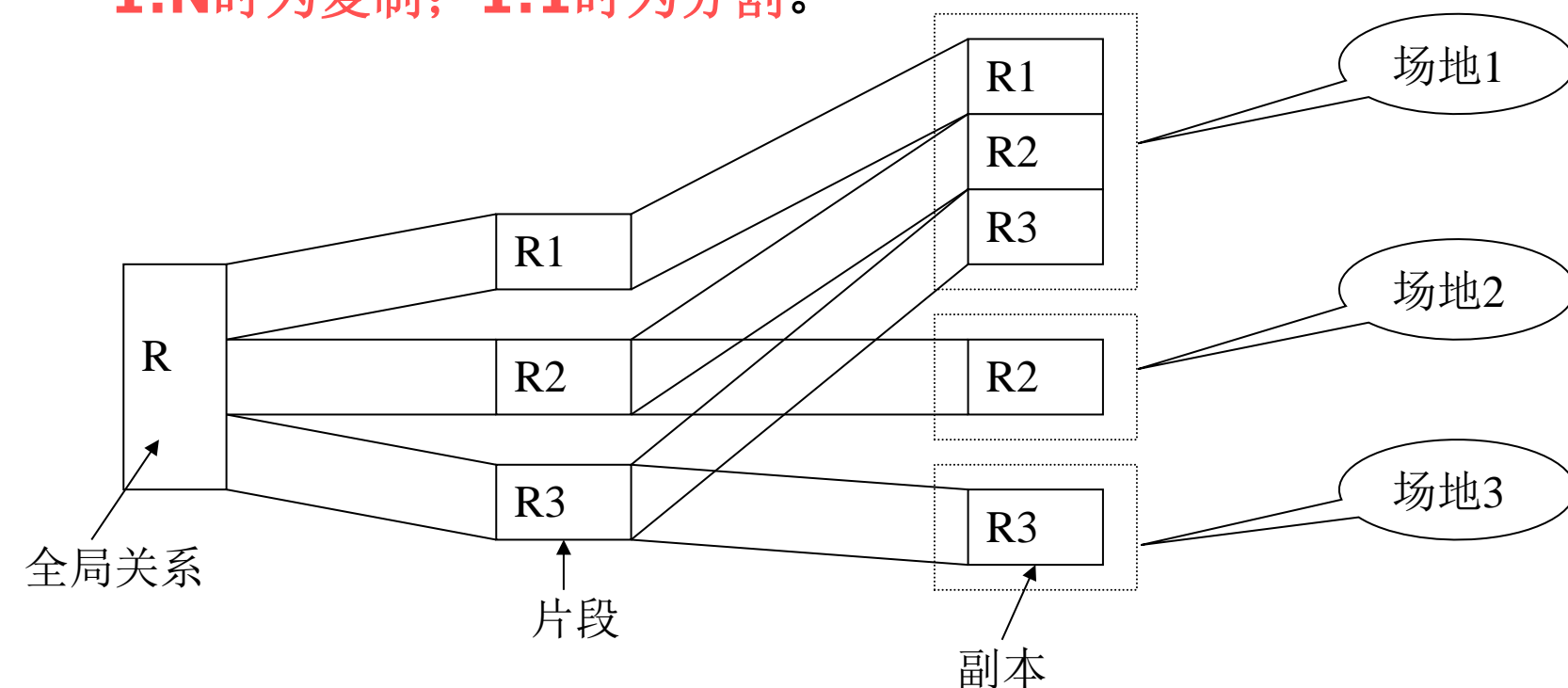
分布式数据库的设计

● 分片的定义

□ **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。



□ 分片的原则

在构成分布式数据库系统时，设计者必须考虑数据如何分布在各个场地上，即全局数据如何进行逻辑划分和物理分配问题。**哪些数据须分布存放、哪些数据不需要分布存放、哪些数据需要复制等**，对系统进行全盘考虑，使系统性能最优。但无论如何进行分片，必须遵循下面原则：

■ **完备性**：所有全局数据必须映射到某个片段上。

■ **可重构性**：所有片段必须可以重新构成全局数据。

■ **不相交性**：划分的各片段所包含的数据不允许相同。

分布式数据库的设计

● 分片的定义

□ 分片的原则 如果全局关系**R**划分的片段为**R₁, R₂, ..., R_n**,

□ 完备性:

对于**R**中任意数据项**d**(**d**∈**R**), 一定存在**d**∈**R_i**(**1**≤**i**≤**n**)。

□ 可重构性:

存在关系运算**ψ**, 使得**R = R₁ ψ R₂ ψ ... ψ R_n**。

□ 不相交性

任意两个不同片段的交集为空, 即**R_i ∩ R_j = φ (i ≠ j, 1 ≤ i, j ≤ n)**

分布式数据库的设计

● 分片的定义

□ 分片的种类

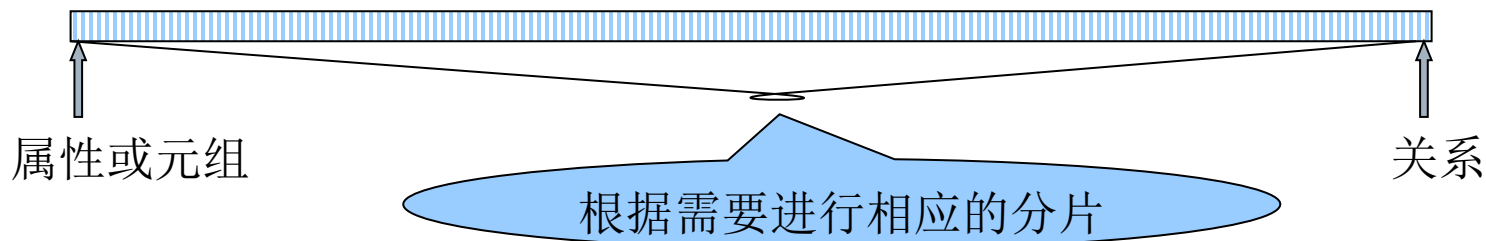
分布式系统按系统实际需求对全局数据进行分片和物理分配。

分片的种类有三种。

■ **水平分片**：按元组进行划分，由分片条件决定。

■ **垂直分片**：按关系属性划分，除关键字外，其他属性不允许出现在多个分片中。

■ **混合分片**：既包括水平分片也包括垂直分片。



分布式数据库的设计

● 水平分片的设计

□ 水平分片的定义

水平分片是将关系的元组集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**。

■ 定义

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

- (1) R_1, R_2, \dots, R_n 与 R 具有相同的模式
- (2) $R_1 \cup R_2 \cup \dots \cup R_n = R$
- (3) $R_i \cap R_j = \Phi \ (i \neq j, 1 \leq i, j \leq n)$

分布式数据库的设计

● 水平分片的设计

□ 例如：设有雇员关系 **EMP{ENO, ENAME, SALARY, DNO}**

ENO	ENAME	SALARY	DNO
001	张三	1500	201
002	李四	1400	202
003	王五	800	203

按下面分片条件进行分片：

E1：满足（**Dno=201**）的所有分组；

E2：满足（**Dno=202**）的所有分组；

E3：满足（**Dno < 201 AND Dno < 202**）的所有分组。

分布式数据库的设计

● 水平分片的设计

□ 将关系**EMP**分成了三个子关系，部门编号**Dno**等于**201**的元组（**E1**）、部门编号**Dno**等于**202**的元组（**E2**）和其他元组（**E3**）。

分片属性：部门编号**Dno**

分片条件：

E1: Dno=201

E2: Dno=202

E3: Dno <> 201 AND Dno <> 202

ENO	ENAME	SALARY	DNO
001	张三	1500	201
002	李四	1400	202
003	王五	800	203

E1:	001	张三	1500	201
E2:	002	李四	1400	202
E3:	003	王五	1200	203

根据水平分片定义，满足：

- E1,E2,E3和EMP关系模式相同；
 - $E1 \cup E2 \cup E3 = EMP$
 - $E1 \cap E2 = \Phi$, $E1 \cap E3 = \Phi$, $E2 \cap E3 = \Phi$
- 因此，E1、E2和E3是EMP的水平分片。

分布式数据库的设计

● 水平分片的设计

■ 水平分片的表示

水平分片实际上是关系的选择操作，因此，用选择操作 σ 表示，选择条件为分片谓词 q ，则 R 的片段可表示为： $\sigma_q(R)$ 。

例：
 $E1 = \sigma_{Dno=201}(EMP)$
 SQL: SELECT * FROM EMP WHERE DNO=201
 $E2 = \sigma_{Dno=202}(EMP)$
 SQL: SELECT * FROM EMP WHERE DNO=202
 $E3 = \sigma_{Dno \neq 201 \text{ AND } Dno \neq 202}(EMP)$
 SQL: SELECT * FROM EMP WHERE Dno \neq 201
 AND Dno \neq 202

求证 $(Dno=201) \cup (Dno=202) \cup (Dno \neq 201 \cap Dno \neq 202) = T$

E1:	001	张三	1500	201
E2:	002	李四	1400	202
E3:	003	王五	1200	203

水平分片正确性判断

分片必须遵循完备性、可重构性和不相交性三个原则。通过验证满足这三个特性判断分片的正确性。例如，上例的验证过程：

□ 完备性证明：

□ 可重构性证明：

□ 不相交性证明：

分布式数据库的设计

● 水平分片的设计

■ 水平分片的表示

水平分片实际上是关系的选择操作，因此，用选择操作 σ 表示，选择条件为分片谓词 q ，则 R 的片段可表示为： $\sigma_q(R)$ 。

例：
 $E1 = \sigma_{Dno=201}(EMP)$
 SQL: SELECT * FROM EMP WHERE DNO=201
 $E2 = \sigma_{Dno=202}(EMP)$
 SQL: SELECT * FROM EMP WHERE DNO=202
 $E3 = \sigma_{Dno \neq 201 \text{ AND } Dno \neq 202}(EMP)$
 SQL: SELECT * FROM EMP WHERE Dno \neq 201
 AND Dno \neq 202

求证 $E1 \cup E2 \cup E3 = EMP$

E1:	001	张三	1500	201
E2:	002	李四	1400	202
E3:	003	王五	1200	203

水平分片正确性判断

分片必须遵循完备性、可重构性和不相交性三个原则。通过验证满足这三个特性判断分片的正确性。例如，上例的验证过程：

□ 完备性证明：

□ 可重构性证明：

□ 不相交性证明：

分布式数据库的设计

● 水平分片的设计

■ 水平分片的表示

水平分片实际上是关系的选择操作，因此，用选择操作 σ 表示，选择条件为分片谓词 q ，则 R 的片段可表示为： $\sigma_q(R)$ 。

例：
 $E1 = \sigma_{Dno=201}(EMP)$
 SQL: SELECT * FROM EMP WHERE DNO=201
 $E2 = \sigma_{Dno=202}(EMP)$
 SQL: SELECT * FROM EMP WHERE DNO=202
 $E3 = \sigma_{Dno \neq 201 \text{ AND } Dno \neq 202}(EMP)$
 SQL: SELECT * FROM EMP WHERE Dno \neq 201
 AND Dno \neq 202

求证 $E1 \cap E2 = \Phi$, $E1 \cap E3 = \Phi$, $E2 \cap E3 = \Phi$

E1:	001	张三	1500	201
E2:	002	李四	1400	202
E3:	003	王五	1200	203

水平分片正确性判断

分片必须遵循完备性、可重构性和不相交性三个原则。通过验证满足这三个特性判断分片的正确性。例如，上例的验证过程：

□完备性证明：

□可重构性证明：

□不相交性证明：

□ 导出水平分片

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关系的属性来划分，这种划分为**导出水平划分**。

- **定义：**如果一个关系的水平分片的分片属性属于另一个关系，则该分片称为另一个关系的导出分片。

- **例如：**

关系**WORKS{ENO, PRJNO, HOURS}**

关系**EMP{ENO, ENAME, SALARY, DNO}**

将**WORKS**按**DNO**进行水平分片，分片条件同**EMP**的水平分片条件，得到的导出水平分片为**W1**、**W2**和**W3**。

□ 导出水平分片

关系 **WORKS**{**ENO**, **PRJNO**, **HOURS**}

关系 **EMP**{**ENO**, **ENAME**, **SALARY**, **DNO**}

则，导出水平分片定义如下：

令： **W' = WORKS** \bowtie **EMP**

W': { **ENO**, **PRJNO**, **HOURS**, **ENAME**, **SALARY**, **DNO** }

根据 **DNO** 对 **W'** 进行水平分片

如： **W1** = $\sigma_{DNO=201}$ (**W'**)

= $\sigma_{DNO=201}$ (**WORKS** \bowtie **EMP**)

= **WORKS** \bowtie $\sigma_{DNO=201}$ (**EMP**)

= **WORKS** \bowtie **E1**

只保留 **WORKS** 的属性

W1 = $\pi_{attr(WORKS)}$ (**W1**)

= $\pi_{attr(WORKS)}$ (**WORKS** \bowtie **E1**)

= **WORKS** \bowtie **E1**

称为半连接。

同理： **W2** = **WORKS** \bowtie **E2**

W3 = **WORKS** \bowtie **E3**

通过上述三个步骤得出按关系 **EMP** 的 **DNO** 属性对 **WORKS** 进行水平划分，得出 **WORKS** 的导出水平分片 **W1**、**W2** 和 **W3**。

分布式数据库的设计

● 垂直分片的设计

□ **垂直分片**是将一个关系按属性集合分成不相交的子集（主关键字除外），属性集合称为分片属性。即**垂直分片**是将关系按列纵向以属性组划分成若干片段。

□ 定义：如果R的子关系 $\{R_1, R_2, \dots, R_n\}$ 满足以下条件，则称其为R的垂直分片。

$$(1) \text{Attr}(R_1) \cup \text{Attr}(R_2) \cup \dots \cup \text{Attr}(R_n) = \text{Attr}(R)$$

其中， $\text{Attr}(R)$ 表示关系R的属性集

(2) $\{R_1, R_2, \dots, R_n\}$ 是关系R的无损分解。

$$(3) \text{Attr}(R_i) \cap \text{Attr}(R_j) = P_k(R) \quad (i \neq j, \text{ 且 } 1 \leq i, j \leq n)$$

其中， $P_k(R)$ 表示R主关键字

分布式数据库的设计

● 垂直分片的设计

□ 例：一雇员关系

EMP {ENO, ENAME, BIRTH, SALARY, DNO}

ENO, ENAME, BIRTH, SALARY, DNO为雇员编码、

雇员姓名、出生日期、工资、部门编号，其元组为：

ENO	ENAME	BIRTH	SALARY	DNO
001	张三	1960. 5. 2	1500	201
002	李四	1957. 3. 5	1400	202
003	王五	1985. 2. 4	1200	203

分布式数据库的设计

● 垂直分片的设计

- 假设：E1 { ENO, ENAME, BIRTH }
E2 { ENO, SALARY, DNO }
则E1和E2元组分别为：

E1元组:		
ENO	ENAME	BIRTH
001	张三	1960. 5. 2
002	李四	1957. 3. 5
003	王五	1985. 2. 4

E2元组:		
ENO	SALARY	DNO
001	1500	201
002	1400	202
003	1200	203

根据垂直分片条件，可知：

E1和E2是EMP的无损分解。

$\text{Attr}(E1) \cup \text{Attr}(E2) = \text{Attr}(EMP)$

$\text{Attr}(E1) \cap \text{Attr}(E2) = \{ \text{ENO} \}$

因此，E1和E2是EMP的垂直分片。

分布式数据库的设计

● 垂直分片的设计

■ 垂直分片的表示方法

垂直分片是指指定属性集上的投影操作。用投影运算 Π 表示，投影属性为分片属性。如上例E1、E2表示为：

$E1 = \pi_{ENO, ENAME, BIRTH} (EMP)$,

SQL: SELECT ENO, ENAME, BIRTH FROM EMP

$E2 = \pi_{ENO, SALARY, DNO} (EMP)$,

SQL: SELECT ENO, SALARY, DNO FROM EMP

- $Attr(E1) \cup Attr(E2) = Attr(EMP)$
- $E1 \bowtie E2 = EMP$
- $Attr(E1) \cap Attr(E2) = PK(EMP)$

垂直分片正确性判断

分片必须遵循完备性、可重构性和不相交性三个原则。通过验证满足这三个特性判断分片的正确性。例如，上例的验证过程：

□ 完备性证明：

□ 可重构性证明：

□ 不相交性证明：

分布式数据库的设计

● 混合分片

混合分片是既包括水平分片，又包括垂直分片的分片过程，称为混合分片。

例：一雇员关系 **EMP {ENO, ENAME, BIRTH, SALARY, DNO}**

其元组同前

混合分片示意图为：

ENO	ENAME	BIRTH	SALARY	DNO
E1			E21	
			E22	
			E23	

先进行垂直分片，分为**E1**和**E2**。然后，将**E2**进行水平分片，分为**E21**、**E22** 和**E23**。

分片表示为：

$$\mathbf{E1} = \pi_{\text{ENO, ENAME, BIRTH}} (\mathbf{EMP})$$

$$\mathbf{E2} = \pi_{\text{ENO, SALARY, DNO}} (\mathbf{EMP})$$

$$\mathbf{E21} = \sigma_{\text{Dno}=201} (\mathbf{E2})$$

$$\mathbf{E22} = \sigma_{\text{Dno}=202} (\mathbf{E2})$$

$$\mathbf{E23} = \sigma_{\text{Dno} <> 201 \text{ AND } \text{Dno} <> 202} (\mathbf{E2})$$

分布式数据库的设计

● 分片的表示方法

- 采用直观的图形表示法外，还提供了分片树表示法。

■ 图形表示法

用图形直观描述。说明：

- 整体全部表示为全局关系；
- 部分表示分段；
- 按水平划分表示水平分段；
- 按垂直划分表示垂直分段；
- 混合划分既有水平划分，又有垂直划分。

分布式数据库的设计

● 分片的表示方法

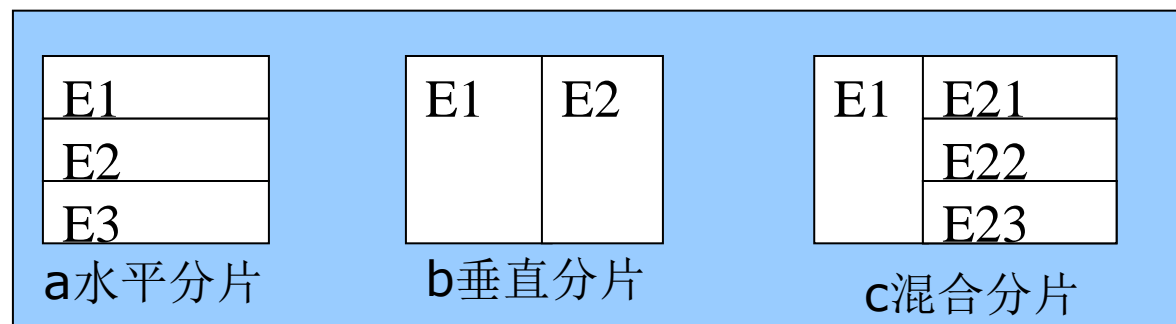
■ 图形表示法

具体图形表示见下图。

□ 关系E水平分片为E1、E2和E3。

□ 关系E垂直分片为E1、E2。

□ 关系E混合分片为E1（垂直分片）和对垂直分片E2的水平分片E21、E22和E23。



分布式数据库的设计

● 分片的表示方法

■ 分片树表示方法

一个分片映射可用分片树表示。先了解分片树的构成。

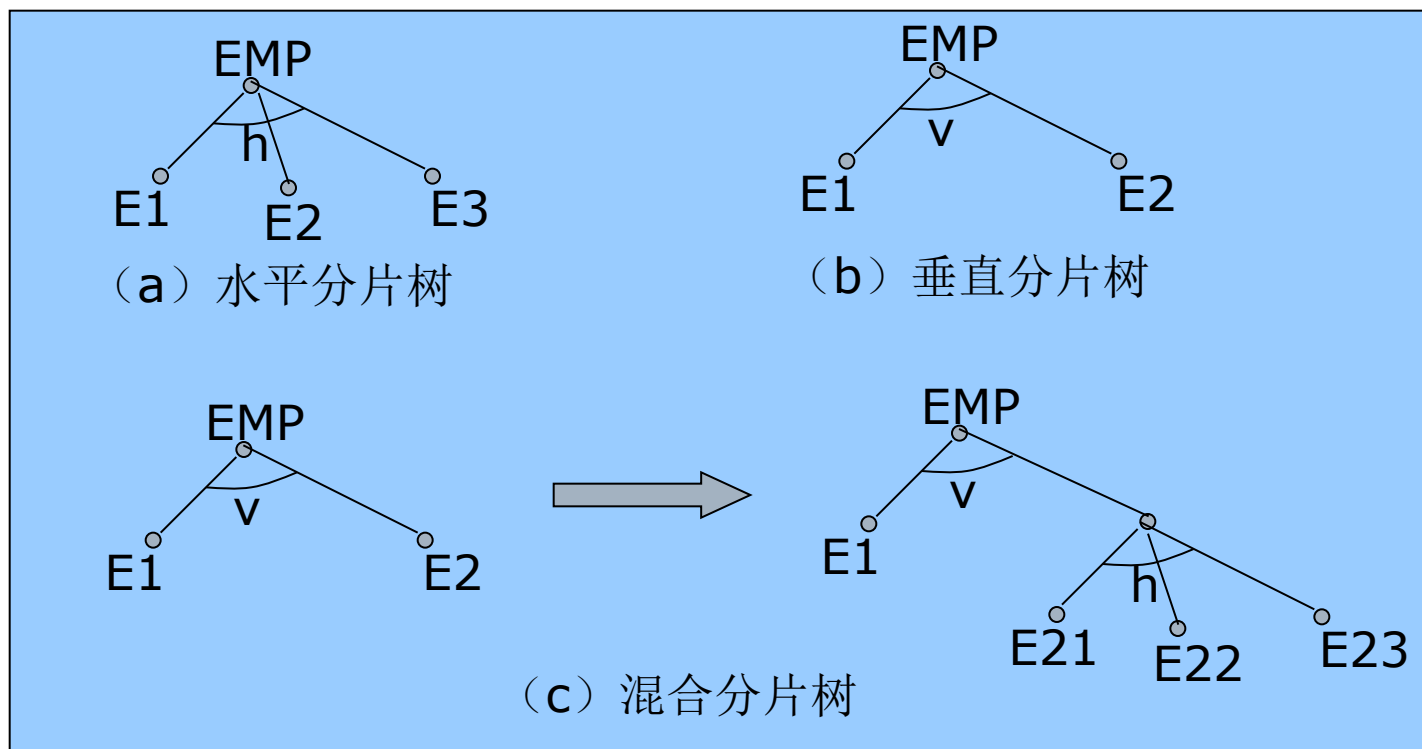
定义：一个分片树由以下几部分组成：

- 根节点表示全局关系；
- 叶子节点表示最后得到的片段；
- 中间节点表示分片过程的中间结果；
- 边表示分片，并用**h**（水平）和**v**（垂直）表示分片类型
- 节点名表示全局关系名和片段名

分布式数据库的设计

● 分片的表示方法

■ 分片树表示方法 例：



分布式数据库的设计

● 分配设计

□ 全局数据经过分片设计，得到各个划分的片段，片段到物理场地的存储映射过程称为分配设计。

□ 分配类型

■ 非复制分配

如果每个片段只存储在一个场地上，称为分割式分布，对应的分布库，称为全分割式分布库。

■ 复制分配

✓ 如果每个片段在每个场地上存有副本，称为全复制分配，对应的分布库称为全复制分布库。

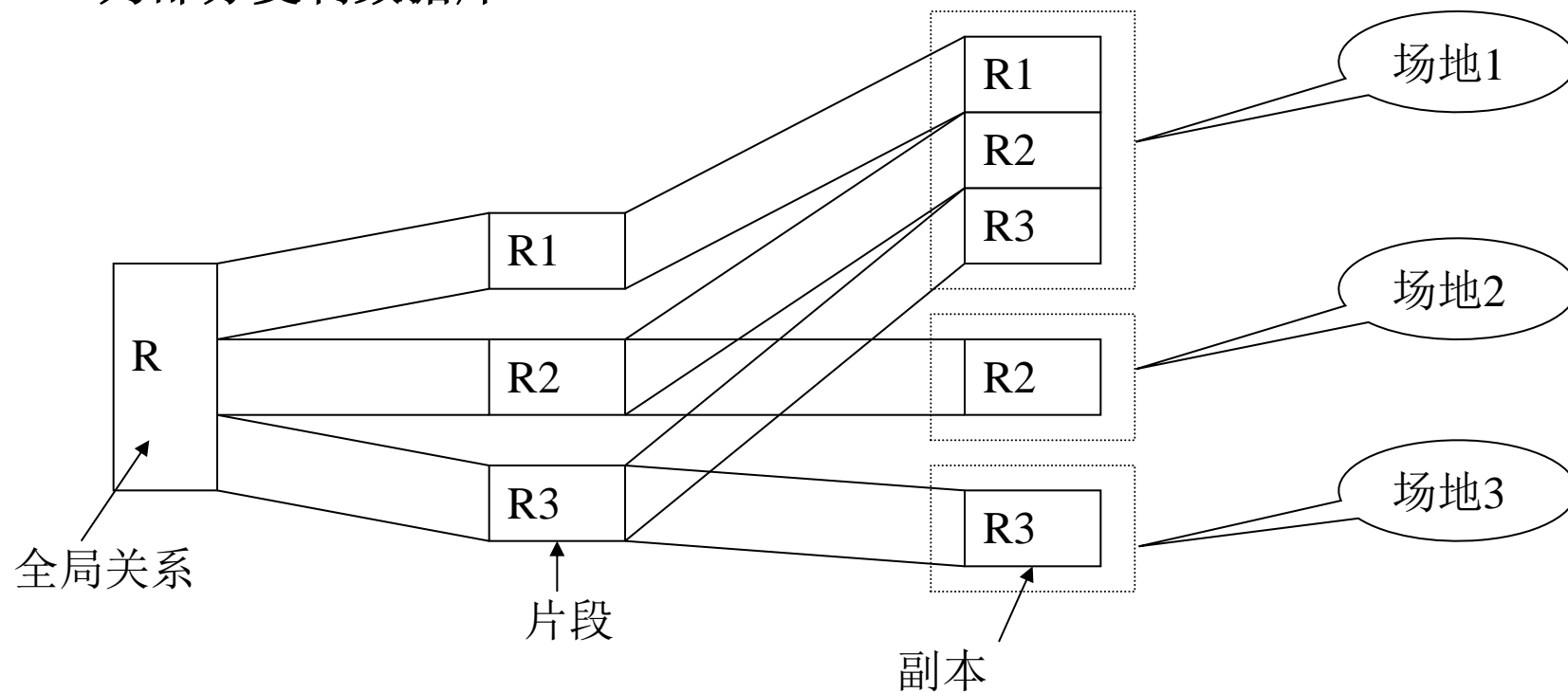
✓ 如果每个片段只在部分场地上存有副本，称为部分复制分配，对应的分布库称为部分复制分布库。

分布式数据库的设计

● 分配设计

□ 例：设 R 为全局关系， $R1$ 、 $R2$ 、 $R3$ 为划分的片段。

■ a 为部分复制数据库



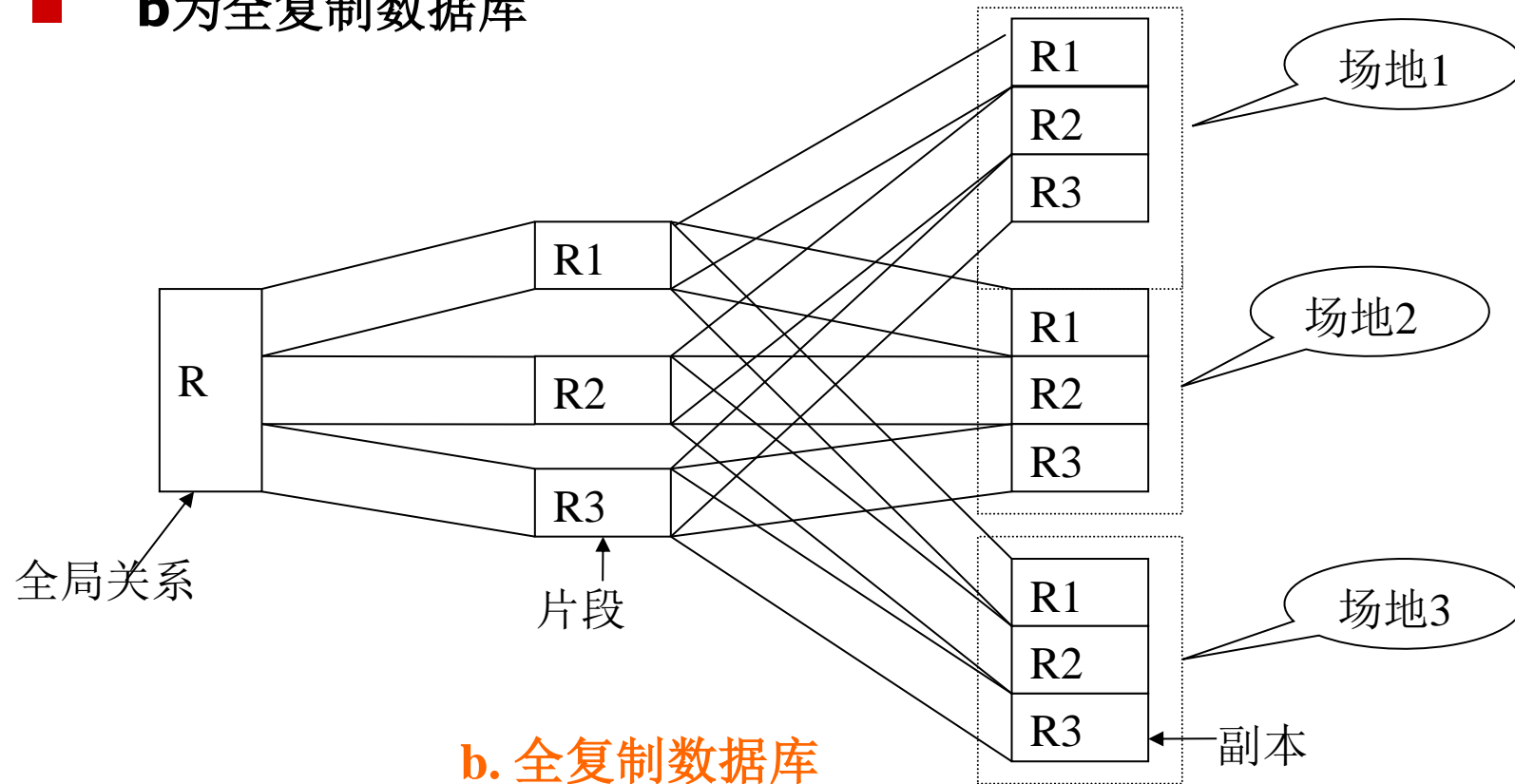
a. 部分复制数据库

分布式数据库的设计

● 分配设计

□ 例：设 R 为全局关系， $R1$ 、 $R2$ 、 $R3$ 为划分的片段。

■ b 为全复制数据库

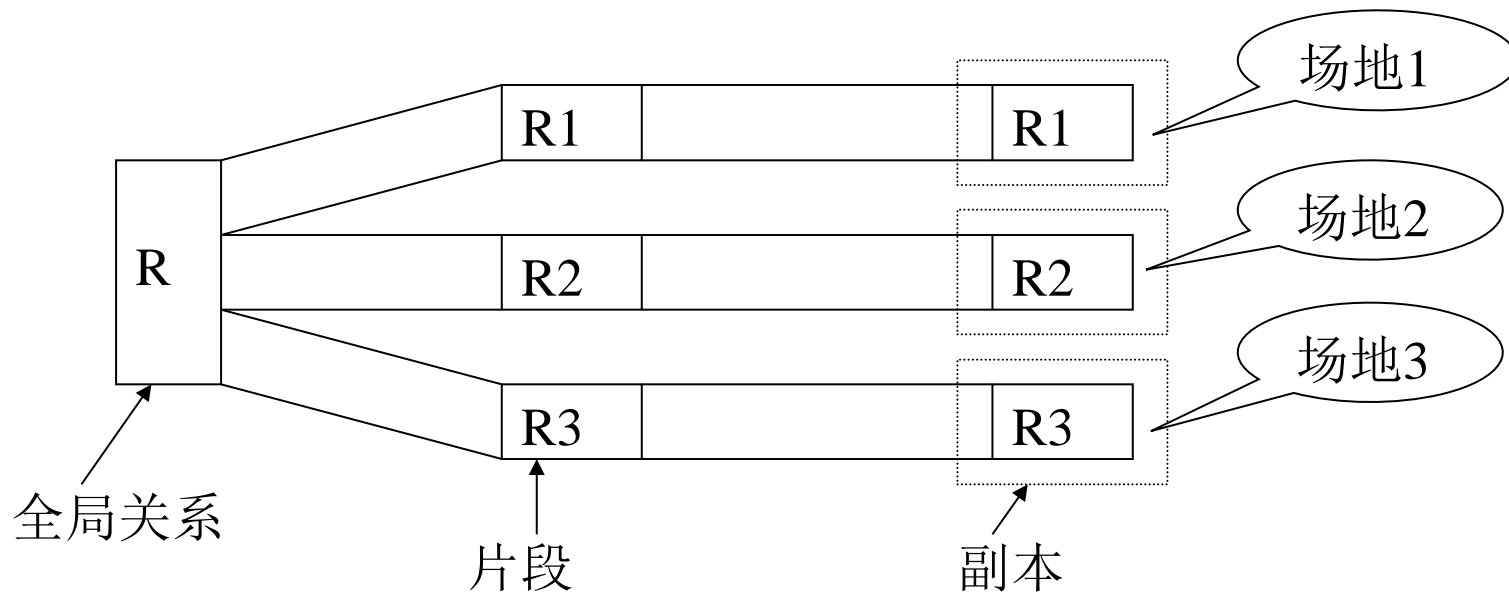


分布式数据库的设计

● 分配设计

□ 例：设 R 为全局关系， $R1$ 、 $R2$ 、 $R3$ 为划分的片段。

■ c 为全分割数据库



c. 全分割数据库

分布式数据库的设计

● 分配设计

□ 分配类型

系统是采用全分割数据库或全复制数据库或部分复制数据库，需根据应用需求及系统运行效率等因素来综合考虑。

一般，从应用角度出发，须考虑：

- 增加事务处理的局部性；
- 提高系统的可靠性和可用性；
- 增加系统的并行性。

从系统角度出发，须考虑：

- 降低系统的运行和维护开销；
- 使系统负载均衡；
- 方便一致性维护。

	全复制	部分复制	全分割
查询处理	易	适中	适中
并发控制	适中	难	易
可靠性	很高	高	低
目录管理	易	适中	适中
可行性	一般	通用	一般

分布式数据库的设计

● 数据复制

□ 数据复制的优势

- 减少网络负载。就近访问，减少网络传输代价。
- 提高系统性能。利用本地资源，并行处理。
- 更好地均衡负载。负载分布到多个节点。

□ 采用复制式存储技术也增加了数据一致性的维护代价。如同步数据时，如何有效地解决冲突等。

分布式数据库的设计

● 数据复制

□ 数据复制的分类

■ 根据更新传播方式不同，分为同步复制和异步复制

- ✓ 同步复制方法是指所有场地上的副本总是具有一致性。

优点：同步复制方法的优势是实时保证了副本数据的一致性。

不足：1) 需要场地间频繁通信并及时完成事务操作；

2) 导致冲突增加，增加了事务响应时间。

适用场合：对于一致性要求较高的商业应用中。

- ✓ 异步复制方法是指各场地上的副本不要求实时一致性，允许在一定时间内是不一致的。

优点：降低了通信量和冲突概率，缩短了事务响应时间，提高了系统效率。

不足：系统不能显示实时的结果，同时也存在潜在的数据冲突，增加了事务回滚的代价。

适用场合：侧重提高系统效率的应用，如大用户群实时访问大数据量的查询应用。

分布式数据库的设计

● 数据复制

□ 数据复制的分类

■ 根据参与复制的节点间的关系不同，分为主从复制和对等复制。

- ✓ 主从复制也称单向复制。主场地上的副本称为主副本，从场地上的副本称为从副本。更新操作只能在主场地上进行，并同步到从场地的从副本上。

分析：主从复制实现简单，易于维护数据一致性。但由于数据只能在主场地上更新，降低了系统的自治性。

- ✓ 对等式复制也称双向复制。各个场地的地位是平等的，可修改任何副本。被修改的副本临时转换为主副本，其它为从副本。主副本所在场地为协调场地，协调同步所有从场地上的从副本。

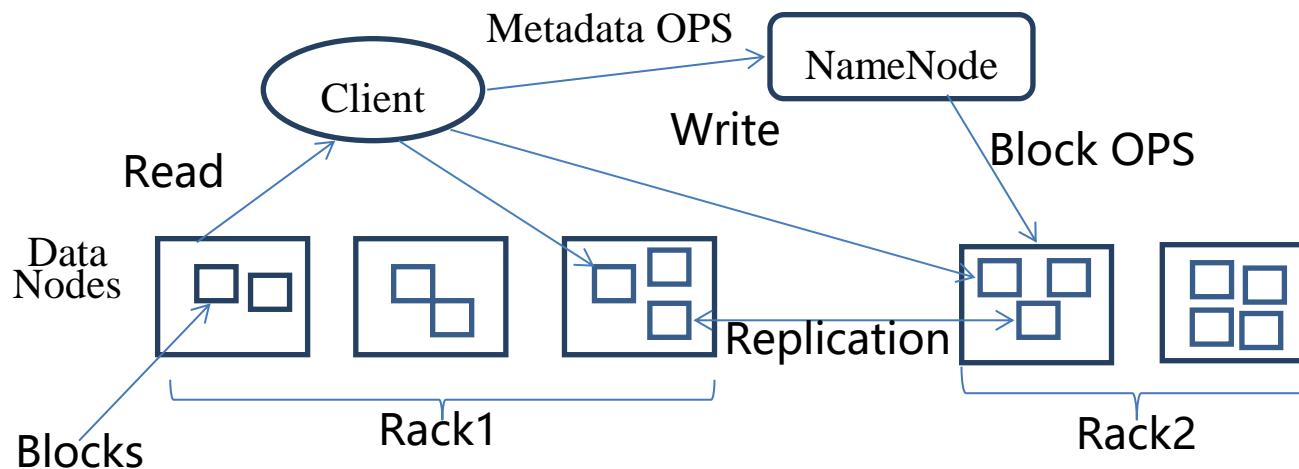
分析：对等式复制中，各场地具有高度自治性，系统可用性好。但由于允许更新任何副本，会引起事务冲突，需要引入有效的冲突解决机制，处理复杂，系统开销大。

分布式数据库的设计

● 大数据的分布存储策略

□ 分布式文件系统-HDFS

- **HDFS:** 分布式文件系统，适合存储超大文件（几百MB、GB甚至TB级别的文件）。将超大文件分割成多个块（block），block大小默认为64MB，每一个block会在多个数据节点（DataNode）上存储多份副本，默认是3份。HDFS采用一次写入、多次读取的流式数据访问模式。



HDFS基本架构图

分布式数据库的设计

● 大数据的分布存储策略

□ 基于**SSTable**的数据存储结构

■ 基本的**SSTable**数据存储结构

- SSTable (Static Search Table)：存储数据的文件格式，其内部提供了一个一致性的、有序的从“键”(Key)到“值”(value)的不可变映射(MAP)。
- 每个SSTable内部包含一系列的**数据块**(通常每个块是64KB大小，且其大小是可配置的)和一个**数据块索引**(保存在SSTable的尾部)，用来定位数据块，当SSTable打开时该索引会被加载到内存。

分布式数据库的设计

● 大数据

□ 基于SSTable

■ 基本的

□ S

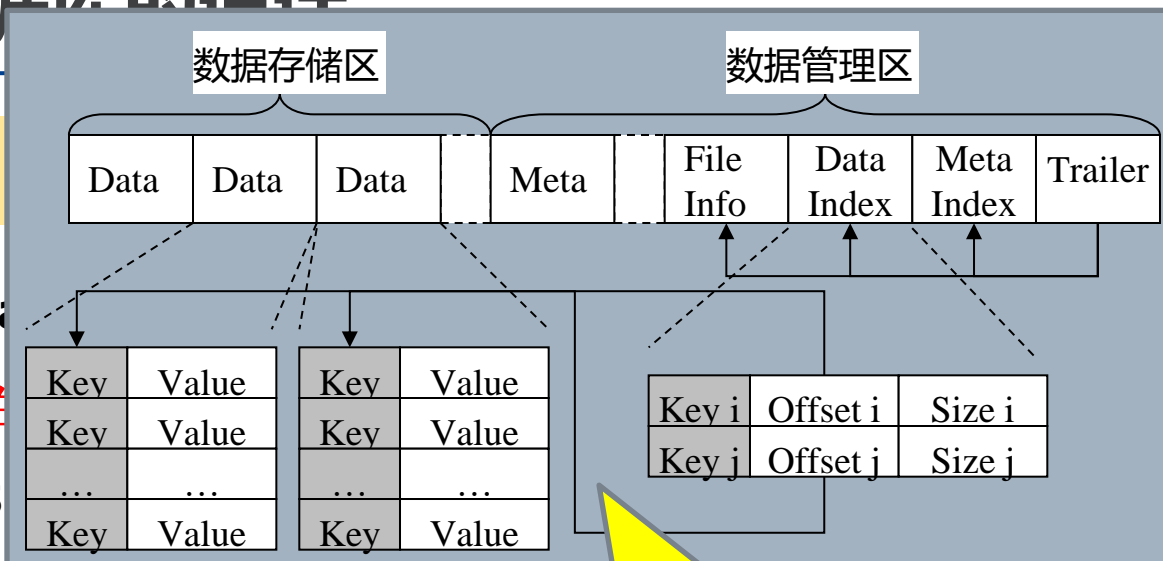
一个一致性的、有序的从

数据存储结构

□ 每个SSTable内部包含

可配置的)和一个数据块索引

SSTable打开时该索引会被

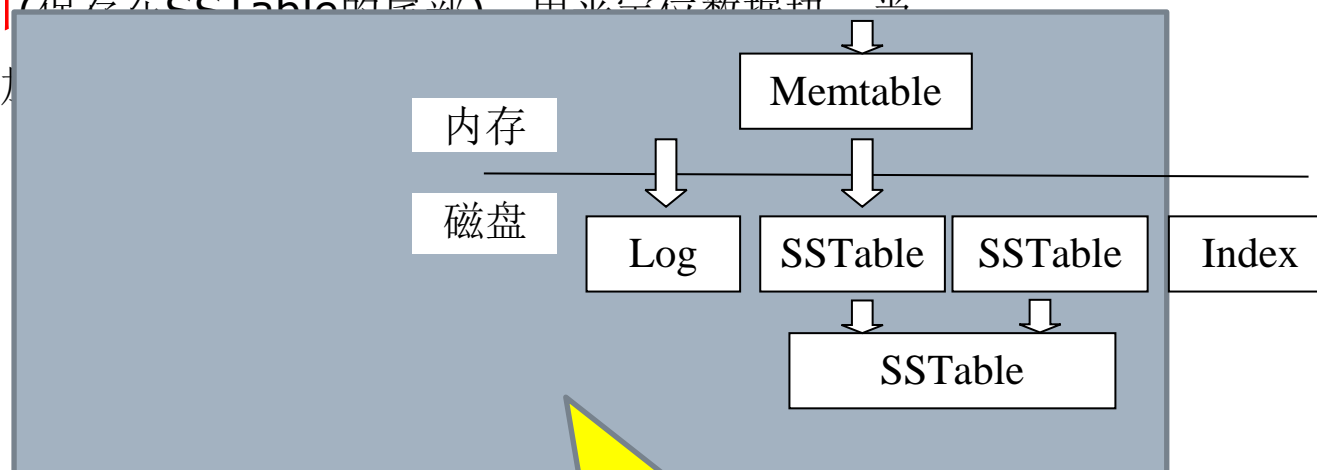


内部提供了一

值”(value)的不可变映射 (MAP)。

常每个块是64KB大小, 且其大小是

用来定位数据块。



文件组织结构

分布式数据库的设计

文档模型

● 大数据的分布存储策略

□ 大数据存储模型

□ 关系模型：行存储模型；列存储模型

□ 键值模型

□ 文档模型

```
{id: 0000000000000001,
 ENO: '002',
 Work:{
  ENAME: 'Bob',
  SALARY: 3500,
  DNO: '103'
 },
 Contact:{
  Address: 'XXX, Road',
  Tel: '11002233',
  Email: 'bob@xx.com'
 }
}
```

Key 1	Value 1: Byte[]
Key 2	Value 2: Byte[]
...
...
Key n	Value n: Byte[]

键值模型

Column Family: EMP			
Key(ENO)	Columns		
002	Column Key	Column Value	Timestamp
	"ENAME"	"Bob"	4
	"SALARY"	"3500"	4
	"DNO"	"103"	4
003	Column Key	Column Value	Timestamp
	"ENAME"	"Lucy"	6
	"SALARY"	"2500"	6
	"DNO"	"102"	6
	"TEL"	"81455588"	6

Column Family: EMP			
Key(ENO)	Super Columns		
002	Key	Columns	
	"WORK"	Column Key	Column Value
		Timestamp	
		"ENAME"	"Bob"
		"SALARY"	"3500"
		"DNO"	"103"
	"Contact"	Column Key	Column Value
		Timestamp	
		"Address"	"XXX, Road"
		"Tel"	"11002233"
		"Email"	"bob@xx.com"

超级列族模型

列族模型

关于本讲内容



祝各位学习愉快!

感谢观看！

讲解人：李鸿岐