非关系式数据库原理

张元鸣 （博士、副教授）
zym@zjut.edu.cn
计算楼B408#

教师主页:http://www.homepage.zjut.edu.cn/zym/

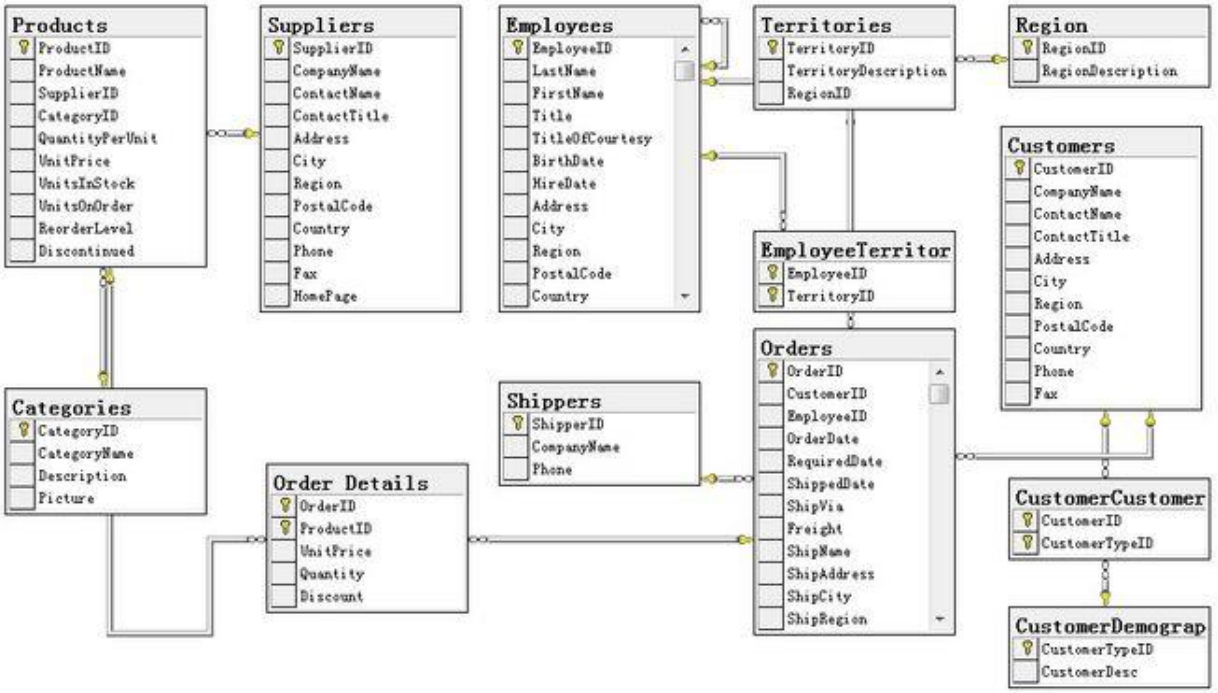
课程概述

- 数据库无处不在（天猫、淘宝、高德、学校、银行）
- 数据库是一座金矿（微信、微博、QQ、日志）
- 数据库没有门槛（计算机专业、非计算机专业）
- 数据库要求很高（组织、管理能力）

2020 年，中共中央、国务院印发《关于构建更加完善的要素市场化配置体制机制的意见》，将数据定义为一种新型生产要素，与土地、劳动力、资本、技术要素并列，共五大生产要素，大数据是重要生产力，数据的价值越来越重要！！！

课程概述

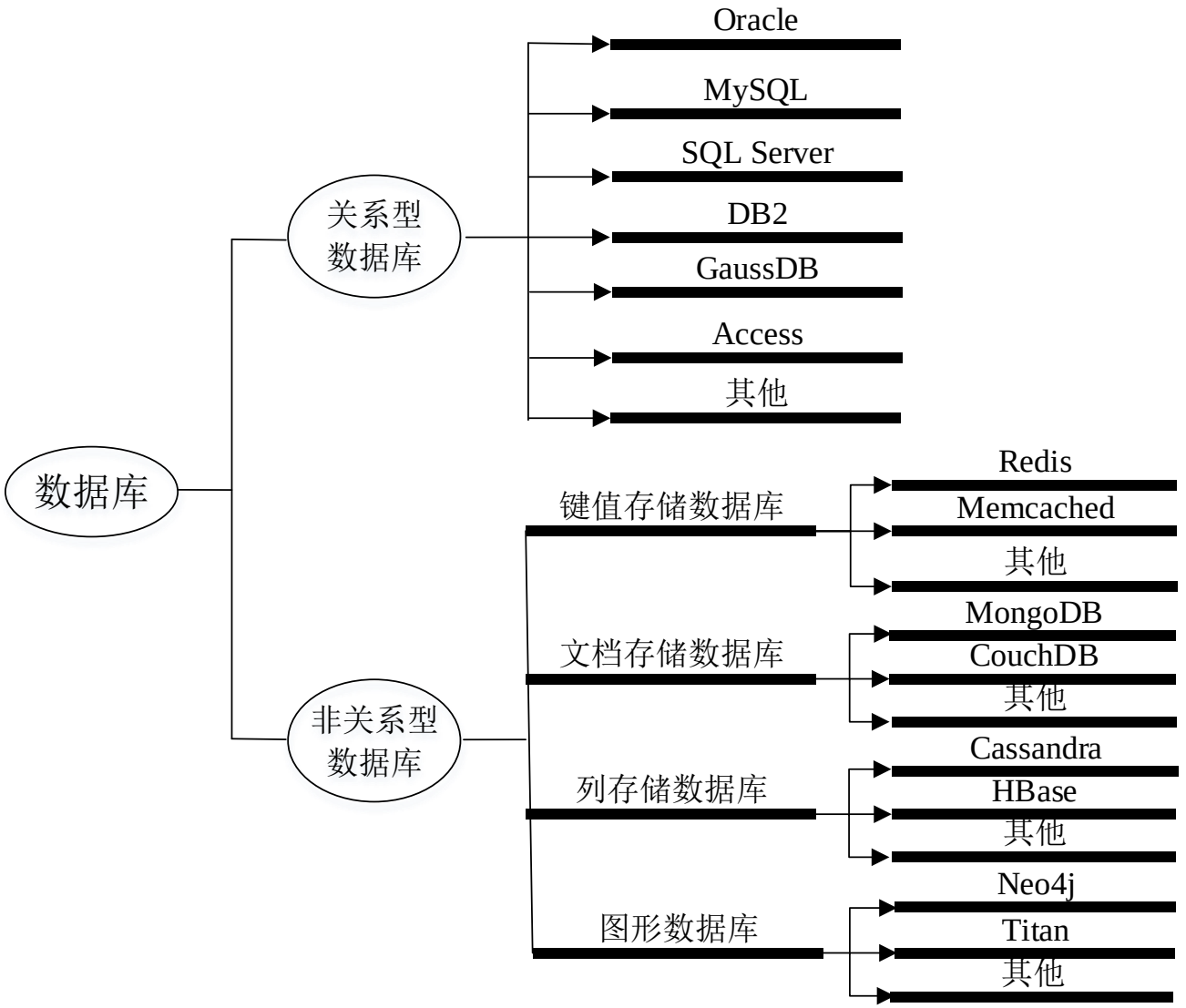
- SQL：Structured Query Language，关系式数据库，用“关系”(也称为“表”)数据结构存储数据（实体和关系），是当前用于存储数据的主流技术。



课程概述

- NoSQL: Not only SQL or no SQL，非单纯的关系式数据库或非关系式数据库，能够弥补关系式数据库的缺点和不足。

课程概述


数据库
关系型数据库
Oracle
MySQL
SQL Server
DB2
GaussDB
Access
其他
非关系型数据库
键值存储数据库
Redis
Memcached
其他
文档存储数据库
MongoDB
CouchDB
其他
列存储数据库
Cassandra
HBase
其他
图形数据库
Neo4j
Titan
其他

课程概述

分类	数据模型	举例	典型应用场景	优点	缺点
键值	键值对	Redis	主要用于大量数据的高访问负载	查找速度快	数据无结构化，通常只被当作字符串或者二进制数据
文档型数据库	文档结构	CouchDB, MongoDb	Web应用	数据结构要求不严格	查询性能不高，而且缺乏统一的查询语法。
列存储数据库	列簇表格	Cassandra, HBase	分布式文件系统	查找速度快，可扩展性强	功能相对局限
图形数据库	图结构	Neo4J, InfoGrid	社交网络，推荐系统等，专注于构建关系图谱	利用图结构相关算法。	很多时候需要对整个图做计算才能得出需要的信息

教学目标

- 掌握NoSQL的基本概念和基本原理，包括数据一致性、可扩展性，CAP、ACID与BASE原理，能够区分非关系式数据库和关系数据库。
- 掌握四种NoSQL数据库的特点及原理，能够从工程化角度理解和运用非关系式数据库。
- 掌握至少一种NoSQL数据库（Neo4j）。
- 区分不同种类非关系式数据库的适用场景，能够根据具体应用选择适当的非关系式数据库。

课程教材

- 教材：

张元鸣，NoSQL数据库技术，清华大学出版社，2023.02

- 参考教材：

Dan Sullivan著，爱飞翔译，NoSQL实践指南：基本原则、设计准则及实用技巧，机械工业出版社：2016年3月

群聊：2024非关系式数据库

教学内容

课程性质：考查课（24学时理论+8学时上机）

授课计划：

- 第1章　基本原理　4
- 第2章　键值数据库　2
- 第3章　文档数据库　6
- 第4章　列族数据库　6
- 第5章　图数据库　6

课程考试

- 成绩评定
  - 课堂参与(10%)
  - 书面作业(20%)
  - 课内实验(20%)
  - 期末考试(50%)

第一章 非关系数据库基本原理

主要内容

1.1　数据管理发展的历史

1.2　非关系式数据库产生的原因

1.3　分布式数据库基本概念

1.4　CAP、ACID与BASE原理

1.5　非关系式数据库概述

1.1 数据管理的发展历史

自计算机被人类发明以来，人们就面临着如何在计算机上管理数据的问题，从简单数据到复杂数据，从少量数据到海量数据，从单用户到多用户。

- 人工管理数据
- 文件系统管理数据
- 数据库系统管理数据
- 大数据技术

1.1 数据管理的发展历史

第一阶段：人工管理数据

数据不能保存、不共享，20世纪50年代中期以前。

- 计算机主要用于科学计算

  数据量小、结构简单，如高阶方程、曲线拟和等。

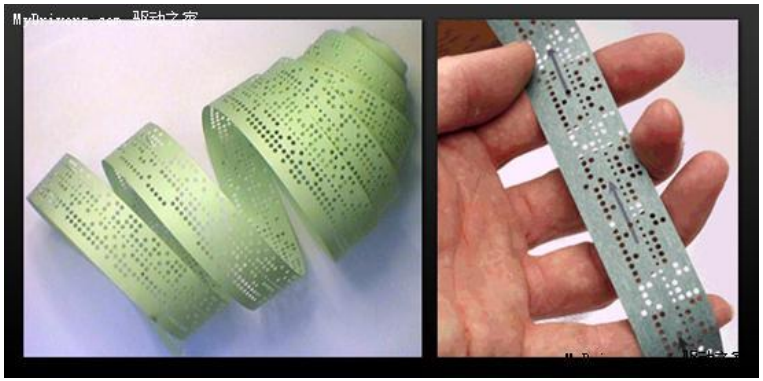
- 外存为顺序存取设备

  磁带、卡片、纸带，没有磁盘等直接存取设备。

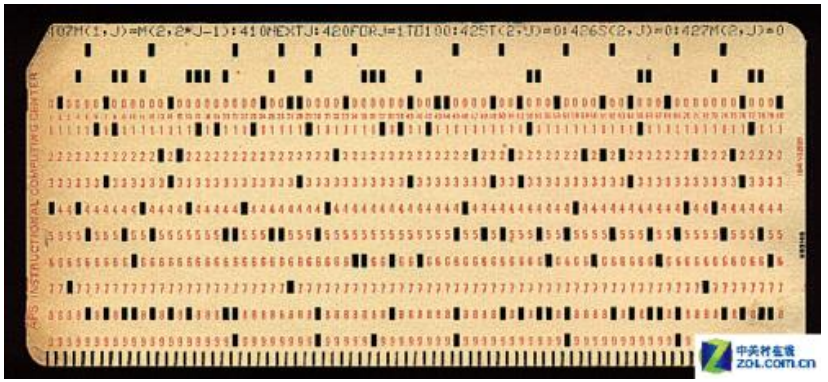
- 没有操作系统，没有数据管理软件

  用户用机器指令编码，通过纸带机输入程序和数据，程序运行完毕后，由用户取走纸带和运算结果，再让下一用户上机操作。

1.1 数据管理的发展历史



穿孔纸带



数据卡片

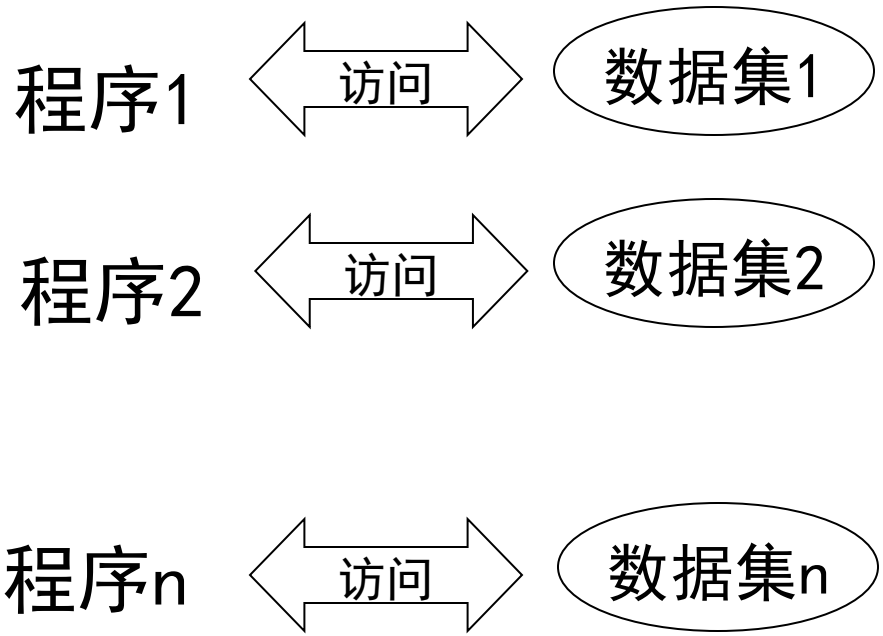
磁带

程序员管理数据

1.1 数据管理的发展历史

- 人工管理阶段的特点：
  1）数据不保存
  2）程序员(人工)管理数据
  3）数据不共享
  4）数据和程序不具有独立性。

程序与数据之间的关系



1.1 数据管理的发展历史

第二阶段 平面文件系统管理数据

从20世纪50年代后期到20世纪60年代中期。

- 存储媒介：

1）磁带：顺序存取

2）磁盘：随机存取

- 特点

1）数据可以长期保存；

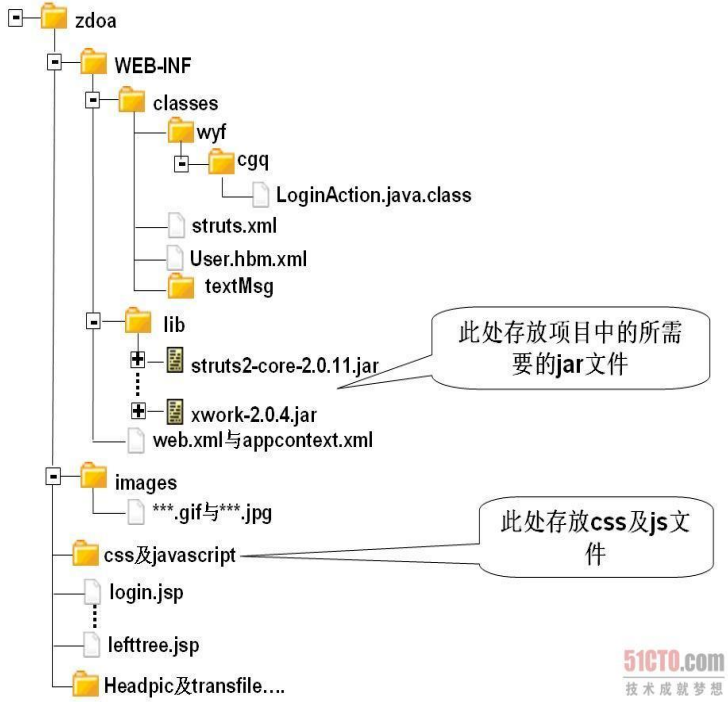
2）文件多样化和结构化；

3）文件系统管理数据。

- 缺点：

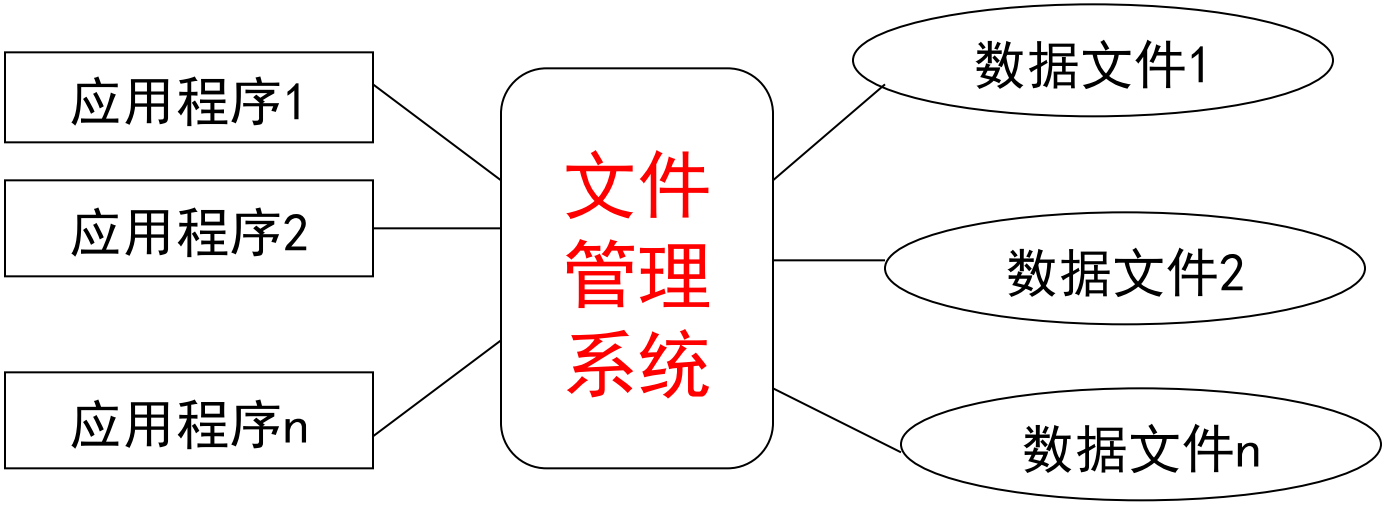
文件系统比人工管理阶段有了很大的改进，但仍存在数据冗余度大、数据安全性较差和数据一致性差等缺点。

1.1 数据管理的发展历史



通过文件系统管理数据

1.1 数据管理的发展历史



应用程序与数据文件的对应关系

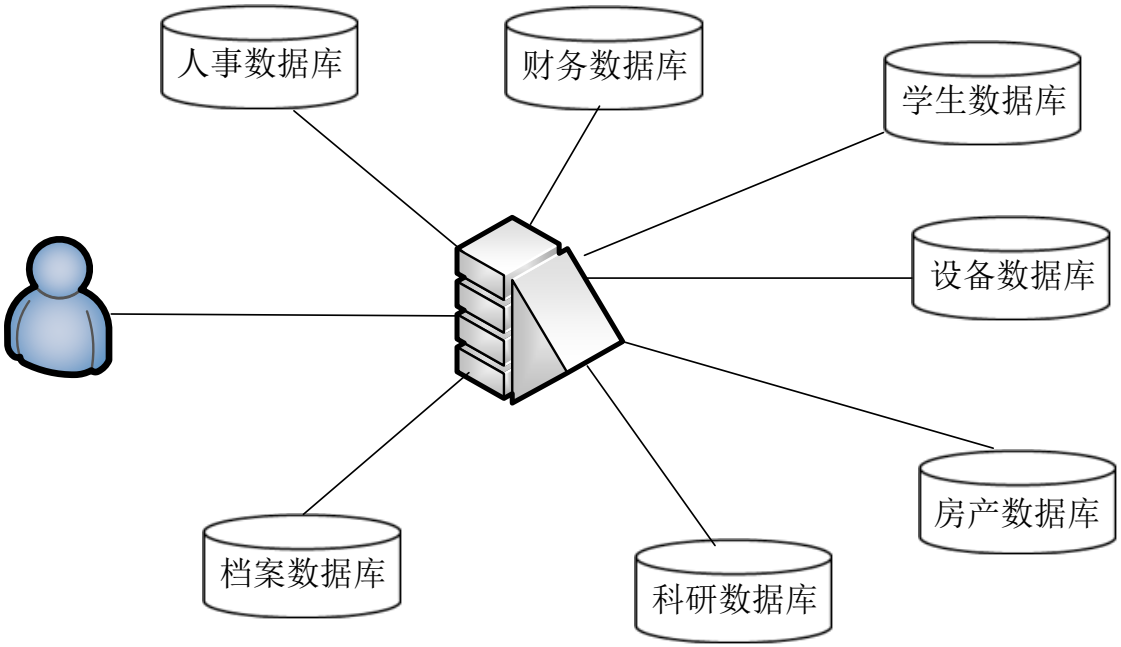
1.1 数据管理的发展历史

第三阶段 数据库系统管理数据

从20世纪60年代后期至今。

以数据模型为主线，数据库系统逐步完善：

- 层次数据库
- 网状数据库
- 关系数据库

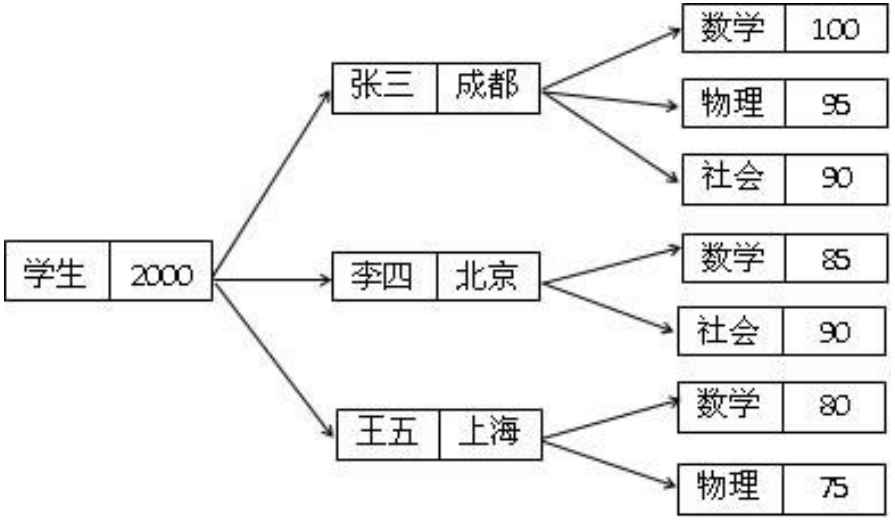


1.1 数据管理的发展历史

1）层次数据库

层次数据库采用树形数据结构组织数据，在各条记录之间创建链接，并确保只有一个节点没有没有父节点，其他节点都只有一个父节点。

- 优点：
  - 与文件系统相比，其搜索的效率更高。
- 缺点：
  - 不适合于表达m:n的关系，存在数据冗余。

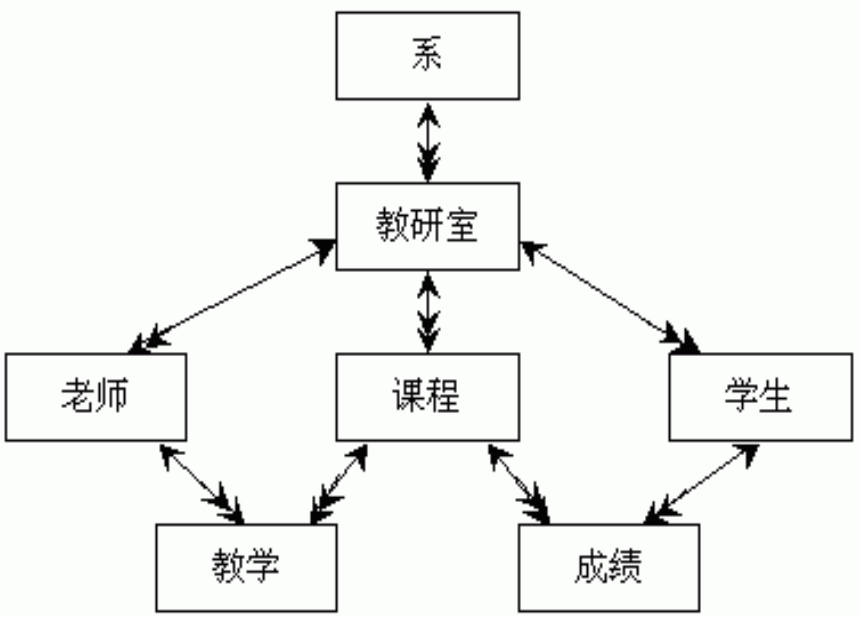


层次数据库

1.1 数据管理的发展历史

2）网络数据库

采用有向无环图数据结构表达数据的关系，一条记录可以有多个父节点。



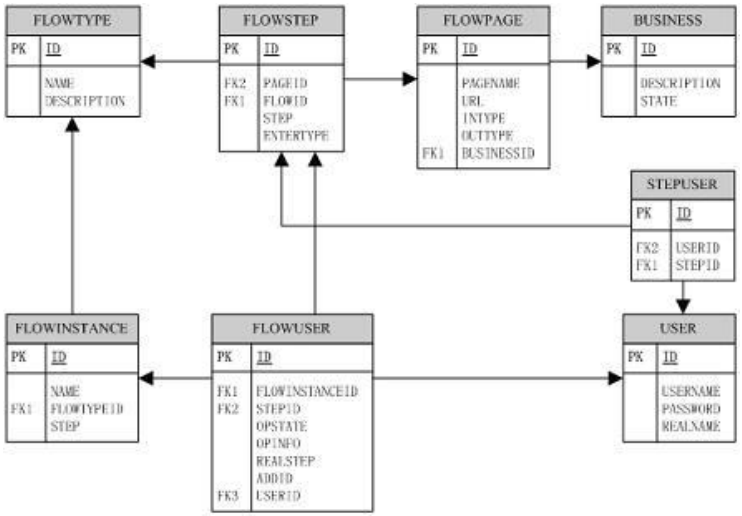
- 关键组件：
  - 纲要：描述节点之间的关系；
  - 数据库：根据纲要保存数据；
- 优点：
  - 能够表达丰富的联系；
- 缺点：
  - 需要维护节点间的大量链接关系

1.1 数据管理的发展历史

3）关系型数据库技术

基于一套形式化的数学模型描述数据及其联系，这种模型有严格的数学理论基础，概念简单、清晰，易于用户理解和使用。

	A	B	C	D	E	F	G
1	系别	学号	姓名	考试成绩	实验成绩	总成绩	
2	信息	991021	李新	74	16	90	
3	计算机	992032	王文辉	87	17	104	
4	自动控制	993023	张磊	65	19	84	
5	经济	995034	郝心怡	86	17	103	
6	信息	991076	王力	91	15	106	
7	数学	994056	孙英	77	14	91	
8	自动控制	993021	张在旭	60	14	74	
9	计算机	992089	金翔	73	18	91	
10	计算机	992005	杨海东	90	19	109	
11	自动控制	993082	黄立	85	20	105	
12	信息	991062	王春晓	78	17	95	
13	经济	995022	陈松	69	12	81	
14	数学	994034	姚林	89	15	104	
15	信息	991025	张雨涵	62	17	79	
16	自动控制	993026	钱民	66	16	82	
17	数学	994086	高晓东	78	15	93	
18	经济	995014	张平	80	18	98	
19	自动控制	993053	李英	93	19	112	
20	数学	994027	黄红	68	20	88	
21							



1.1 数据管理的发展历史

1）关系型数据库的组成

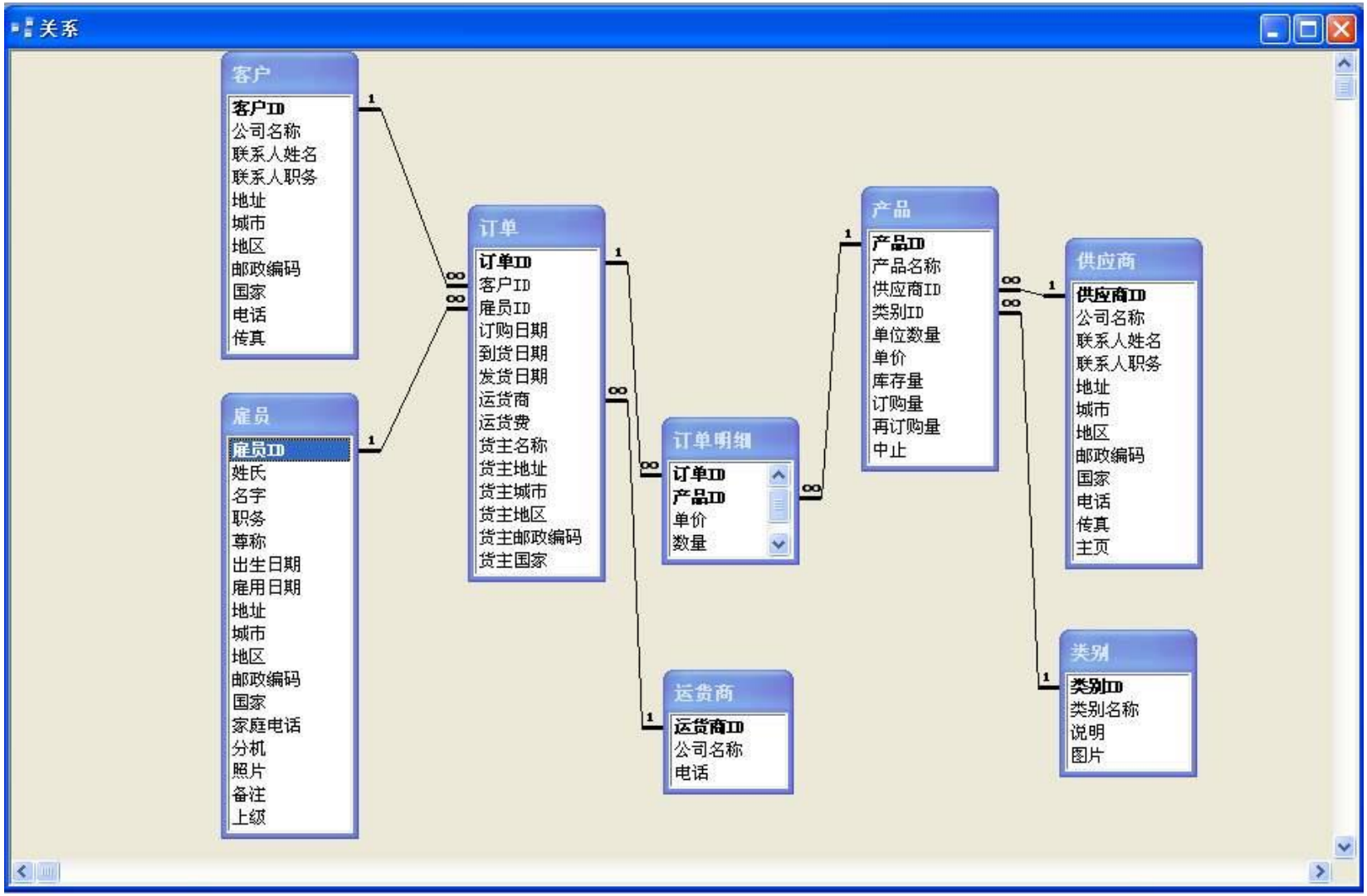
- 存储介质管理程序:记录数据的存取位置
- 内存管理程序：从磁盘读取数据到内存
- 数据字典：保存库结构、表、列、索引、约束、视图等。
- 查询语言：数据定义、数据操纵、数据控制。

1.1 数据管理的发展历史

2）关系型数据库的优点：

- 可以表达复杂的关系；
  - 一切皆关系！
- 数据整体结构化；
  - 通过范式理论进行模式分解
- 具有较高的数据独立性；
  - 数据库与程序相对独立
- 数据冗余度小；
  - 通过外键建立表之间的联系
- 完整的数据控制功能。
  - 实体完整性、参照完整性、用户定义完整性

1.1 数据管理的发展历史



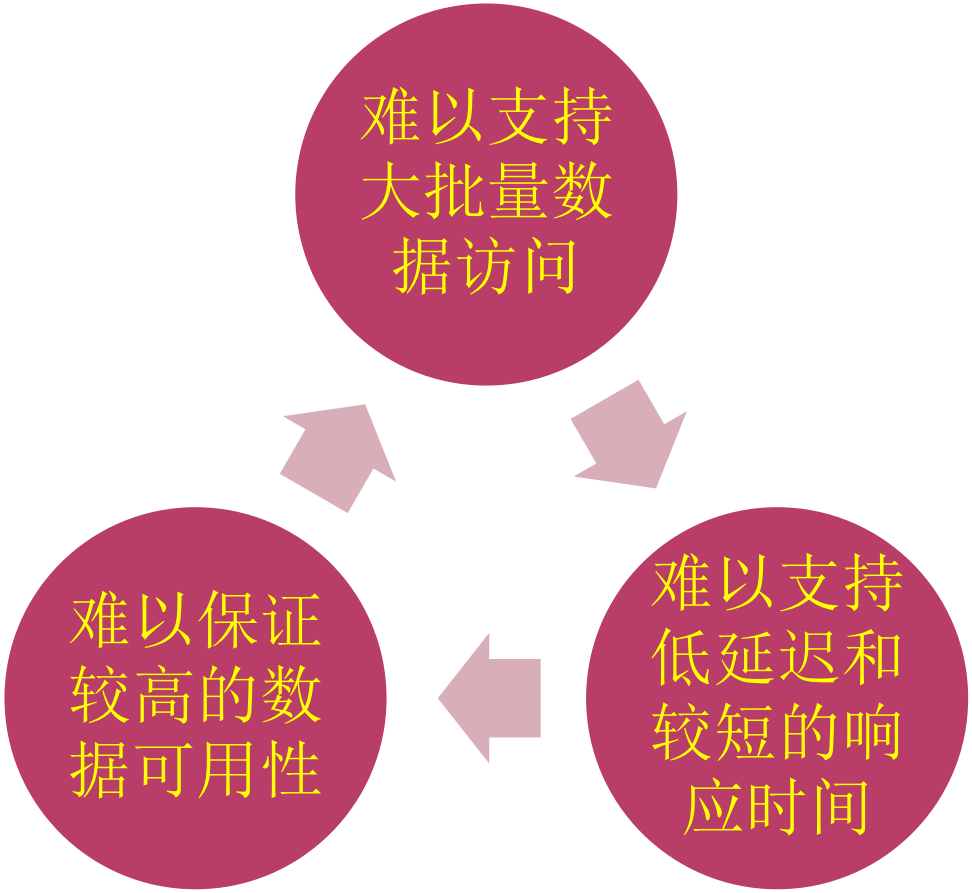
1.1 数据管理的发展历史

3）关系型数据库的局限性：

数据量极大、访问量极大！

1.1 数据管理的发展历史

3）关系型数据库的局限性：

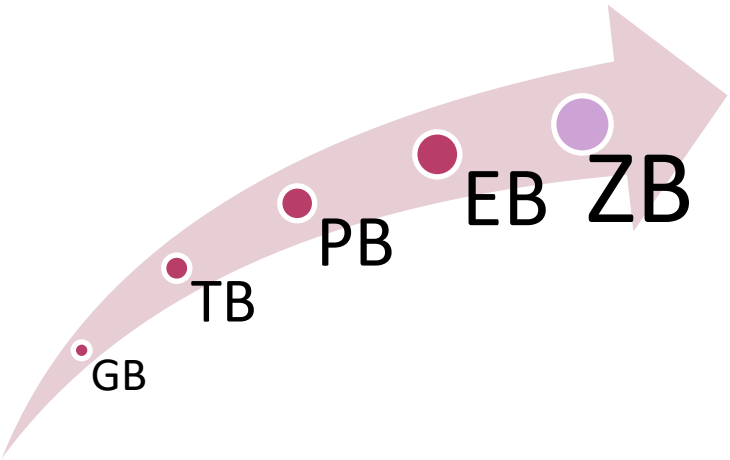


1.1 数据管理的发展历史

第四阶段 大数据技术

数据总量以级数级快速增加：

- 在2006年全球一共新产生了约180EB的数据；
- 在2011年这个数字达到了1.8ZB。
- 到2020年，整个世界的数据总量将会增长44 倍，达到35.2ZB（1ZB=10 亿TB）！



KB→MB→GB→TB→PB→EB→ZB→YB→NB→DB

现有的主流计算机技术已经无法在合理的时间内处理这些庞大的数据！

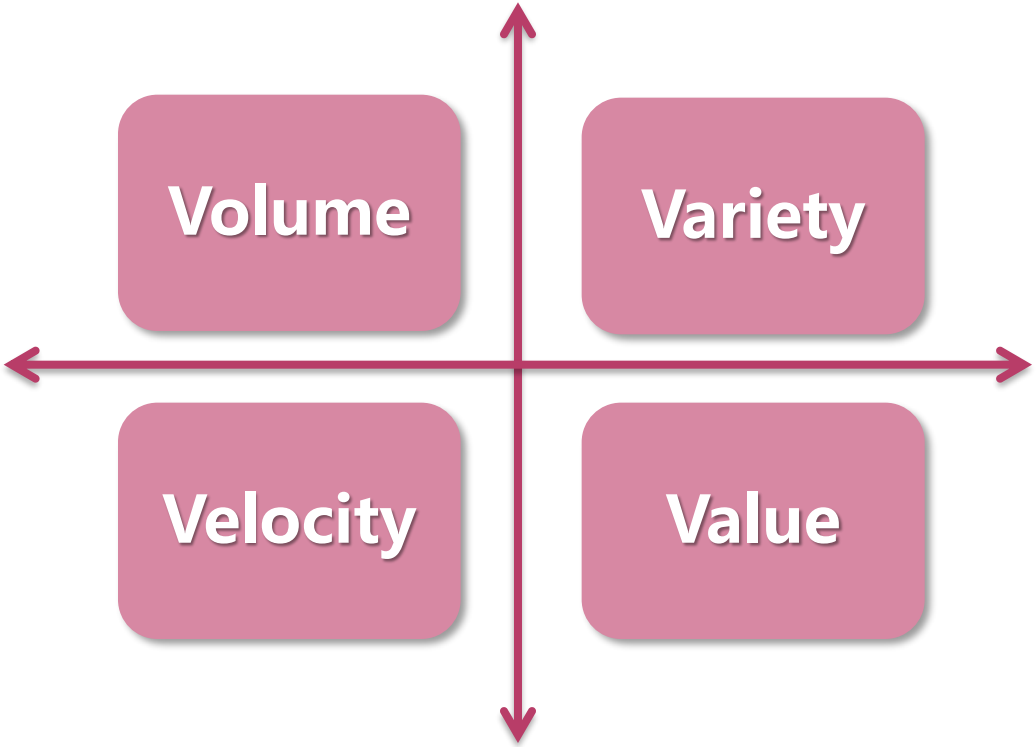
1.1 数据管理的发展历史

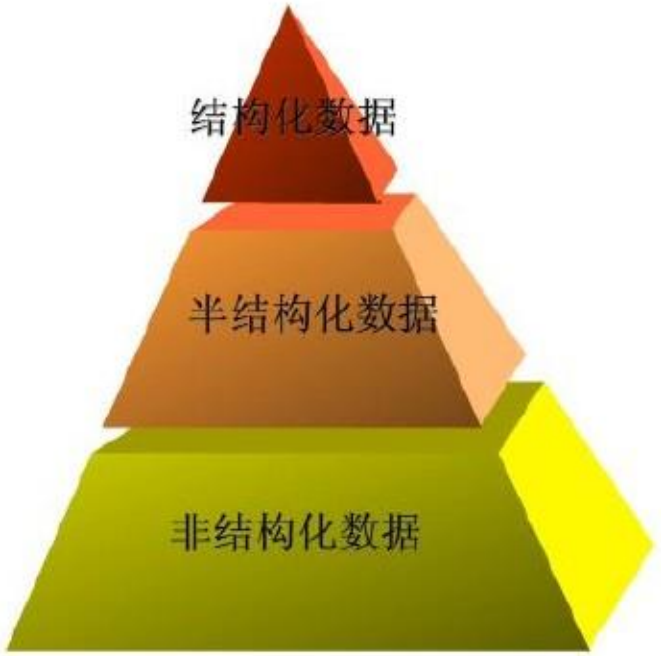
什么是大数据（big data）？

- 维基百科的定义：

  大数据是由数量巨大、结构复杂（非结构化、半结构化）、类型（模态多）众多数据构成的数据集合，是基于云计算的数据处理与应用模式，通过数据的整合共享，交叉复用形成的智力资源和知识服务能力。

1.1 数据管理的发展历史





大体量(Volume)、多样化(Variety)、快速化(Velocity)、价值密度低（Value）是大数据的显著特征。

1.1 数据管理的发展历史

大数据在各领域中的应用：

- 互联网：网页搜索、日志分析、微薄内容分析等。
- 金融领域：交易数据分析、诈骗检验、IT风险管理和自助服务。
- 电信领域：客户数据分析，精准刻画以寻找目标客户，制定有针对性的营销计划、产品组合或商业决策，提升客户价值。
- 政务领域：建设智慧城市，推进公共服务个性化和政府决策智能化。
- 医疗领域：快速检测传染病，进行全面的疫情监测，并通过集成疾病监测和响应程序，快速进行响应。

。。。。。。

1.2 非关系式数据库产生的原因

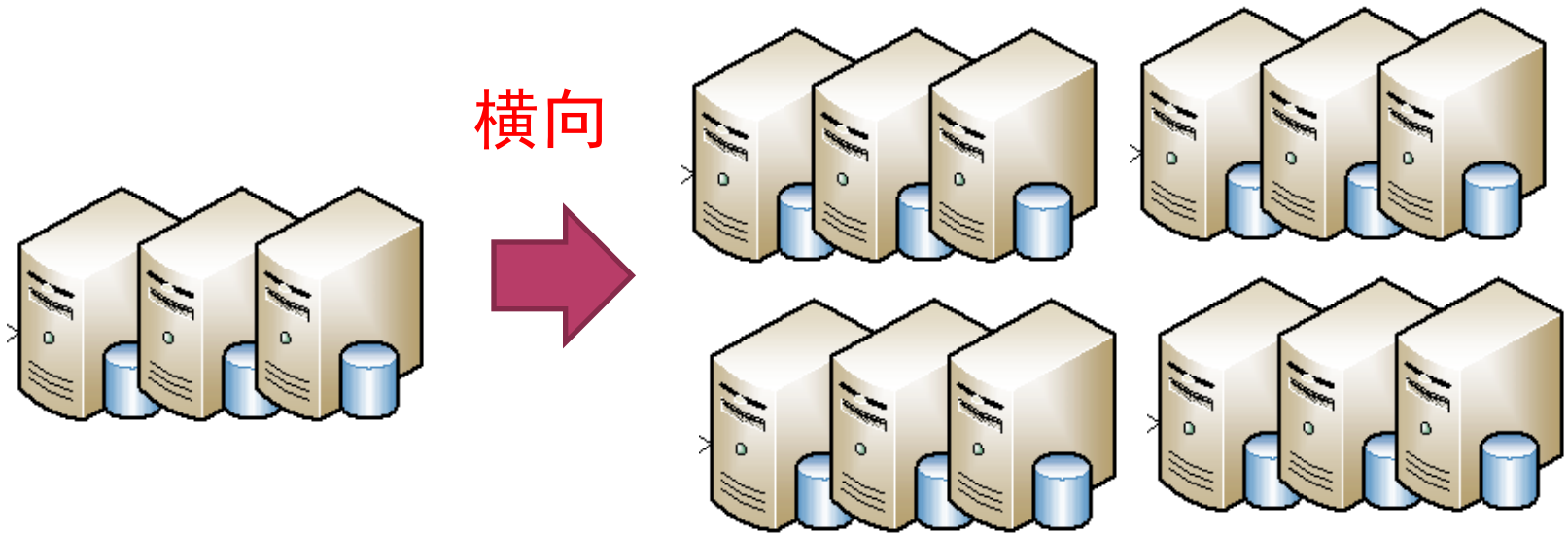
互联网环境下大规模数据(数据量极大)，数以百万计的用户（访问量极大），以及复杂多样的需求（需求千变万化），要求数据库系统具有高可伸缩性、低成本开销、高度的灵活性和高度的可用性。关系型数据库难以适应这些新的要求，催生了NoSQL！！

1.2 非关系式数据库产生的原因

1）可伸缩性

可伸缩性是指有效应对负载变化的能力。而关系型数据库的可伸缩性不高。

– 横向扩展：一般地，通过添加服务器的方式提高负载能力，此为横向扩展。容易，成本低！



1.2 非关系式数据库产生的原因

– 纵向扩展：对现有的数据库服务器进行升级，配置更多的处理器、内存、网络资源等。

不容易，成本高！

纵向扩展

1.2 非关系式数据库产生的原因

2）成本开销

– 关系型数据库不开源，成本高昂，成本难以把控。

– 非关系型数据库基本上是开源软件，成本低。

redis

riak

Cassandra

Apache CouchDB

"Time to relax"

APACHE HBASE

mongoDB

HYPERTABLE

1.2 非关系式数据库产生的原因

3）灵活性

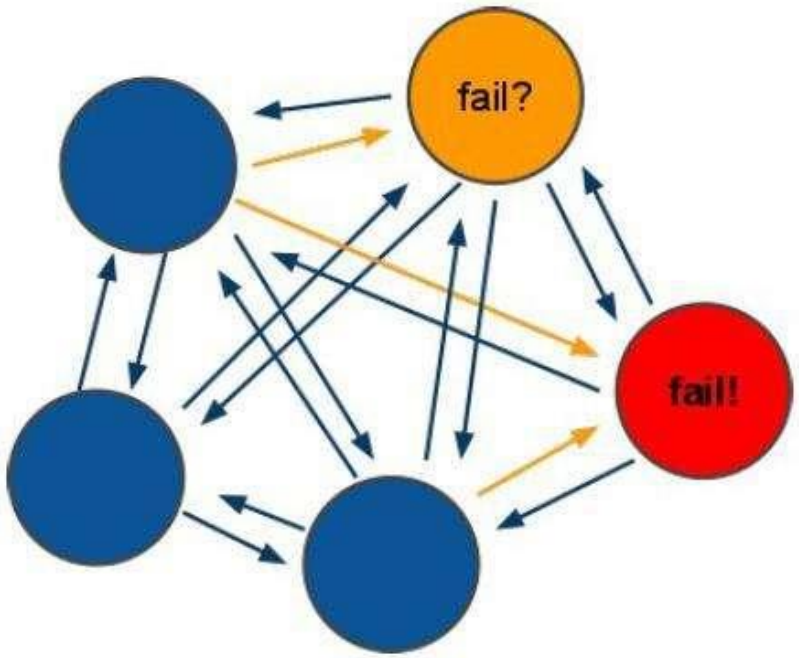
– 有些需求其数据库结构无法提前预设计。

结构在何方?

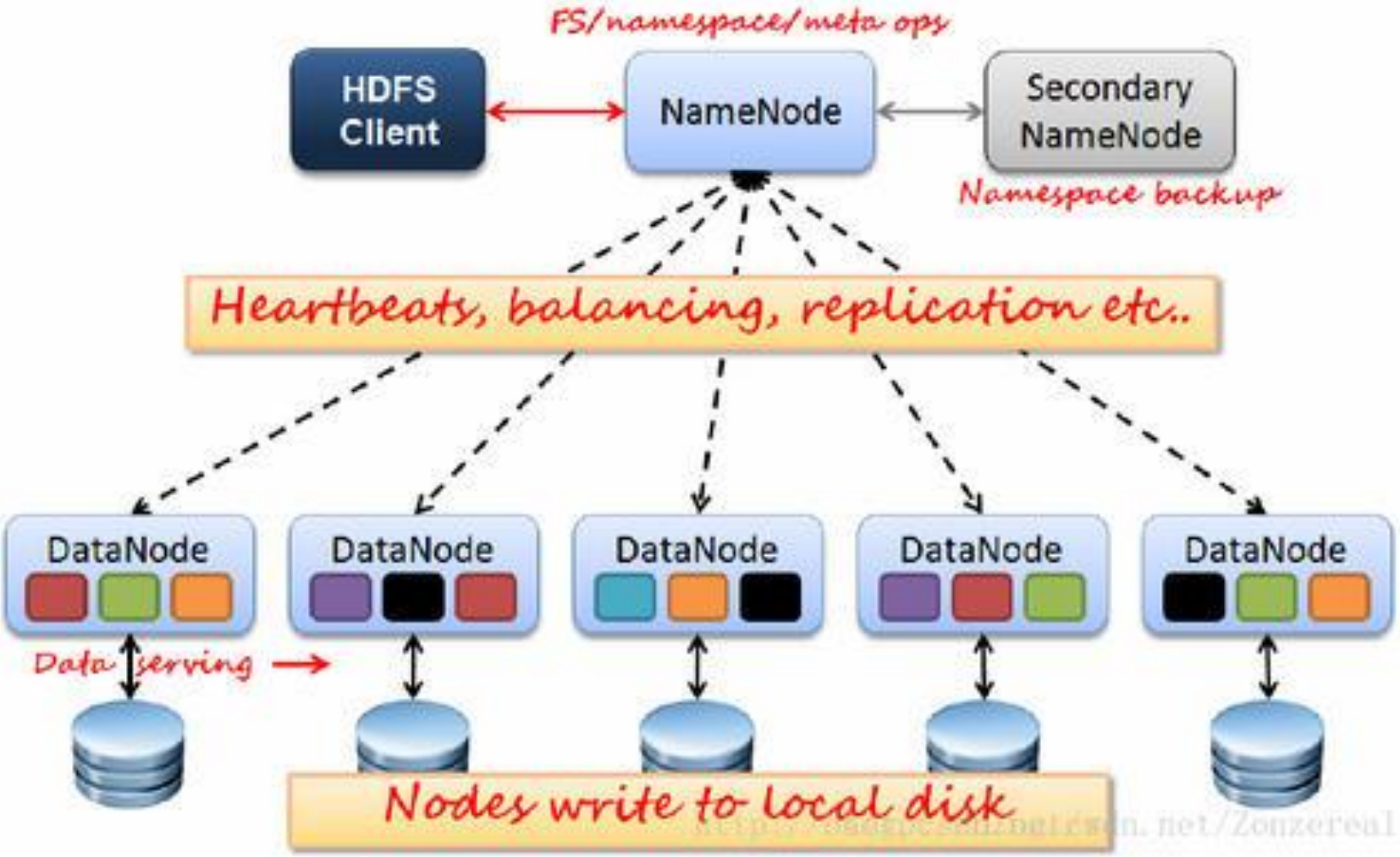
1.2 非关系式数据库产生的原因

4）可用性

– 机器宕机是不可避免的！但是，某台机器宕机不能够影响数据的正确性和软件的可用性。

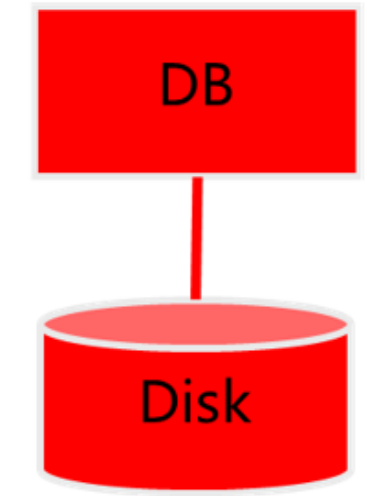


1.2 非关系式数据库产生的原因

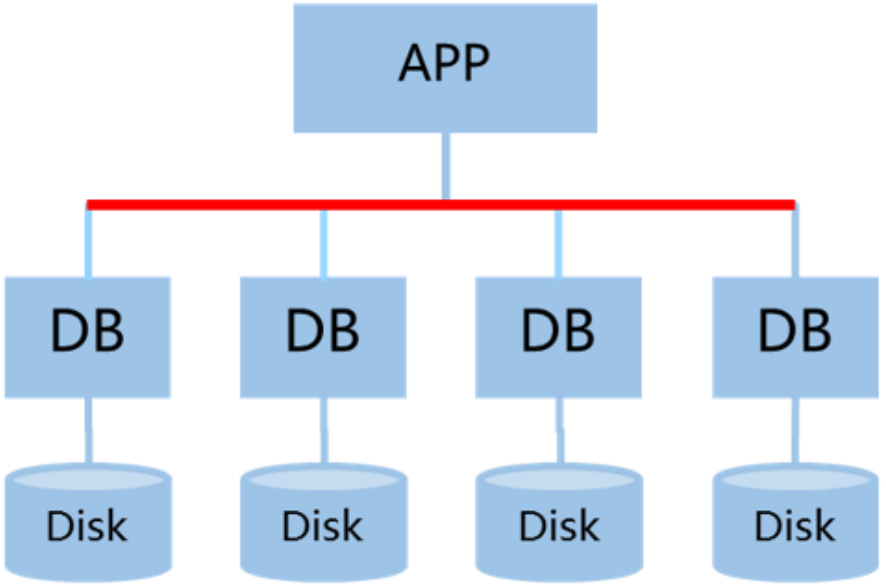


1.3 分布式数据库基本概念

为了满足数据量极大，访问量极大的新需求，NoSQL一般需要部署在多台服务器上（集群架构），由此构成分布式数据库系统。



常规单机数据库



分布式数据库

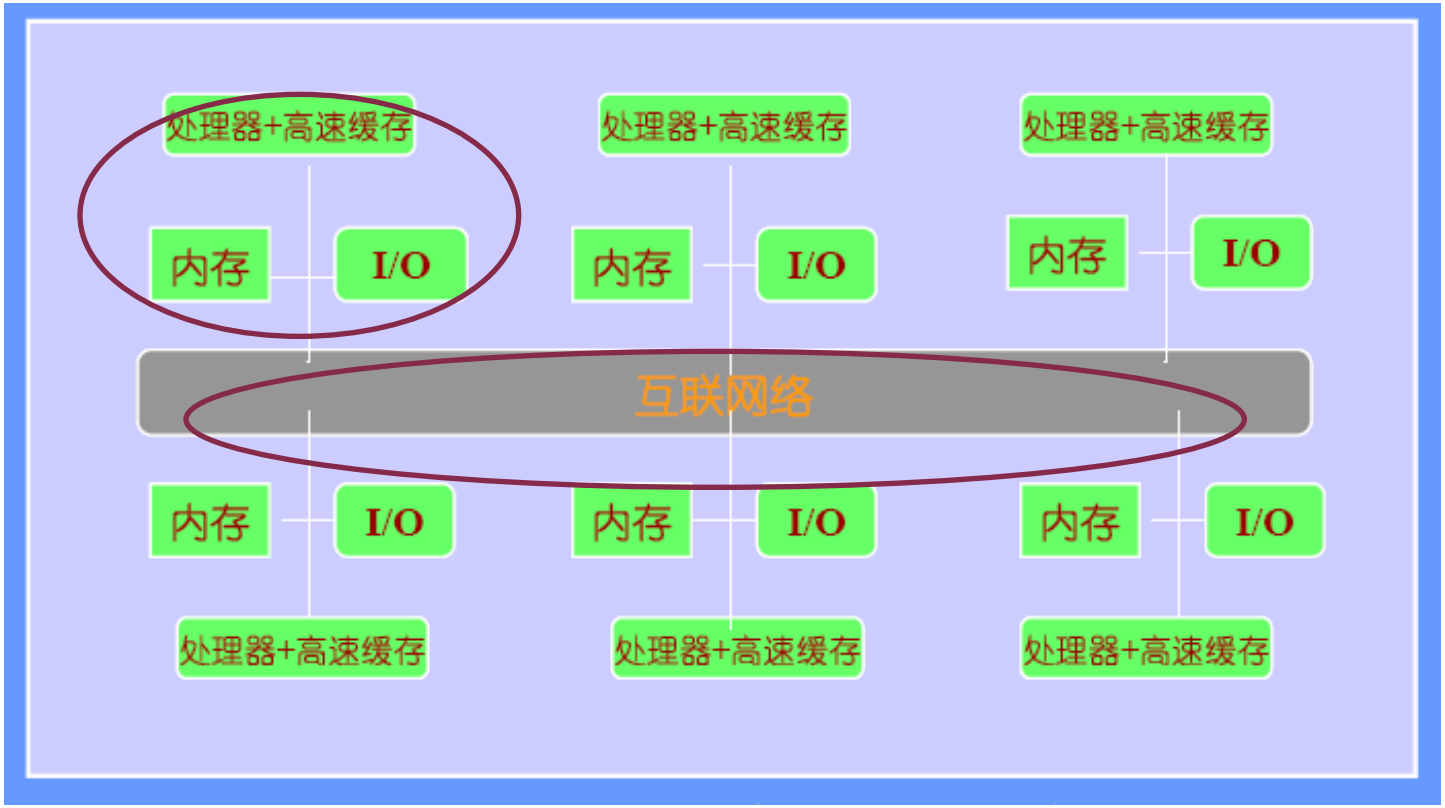
1.3 分布式数据库基本概念

1、集群架构

集群（cluster）是互相连接的多个独立计算机的集合，这些计算机可以是单机或多处理器系统（PC、工作站或SMP），每个结点都有自己的存储器、I/O设备和操作系统，对用户来说是一个单一的系统，它可以提供低价高效的高性能环境和快速可靠的服务。

1.3 分布式数据库基本概念

1、集群架构



1.3 分布式数据库基本概念

1. 价格低廉、易于构建、可扩缩性极强。
2. 由多台同构或异构的独立计算机通过高性能网络或局域网互连在一起。
3. 从用户的角度来看，机群就是一个单一、集中的计算资源。

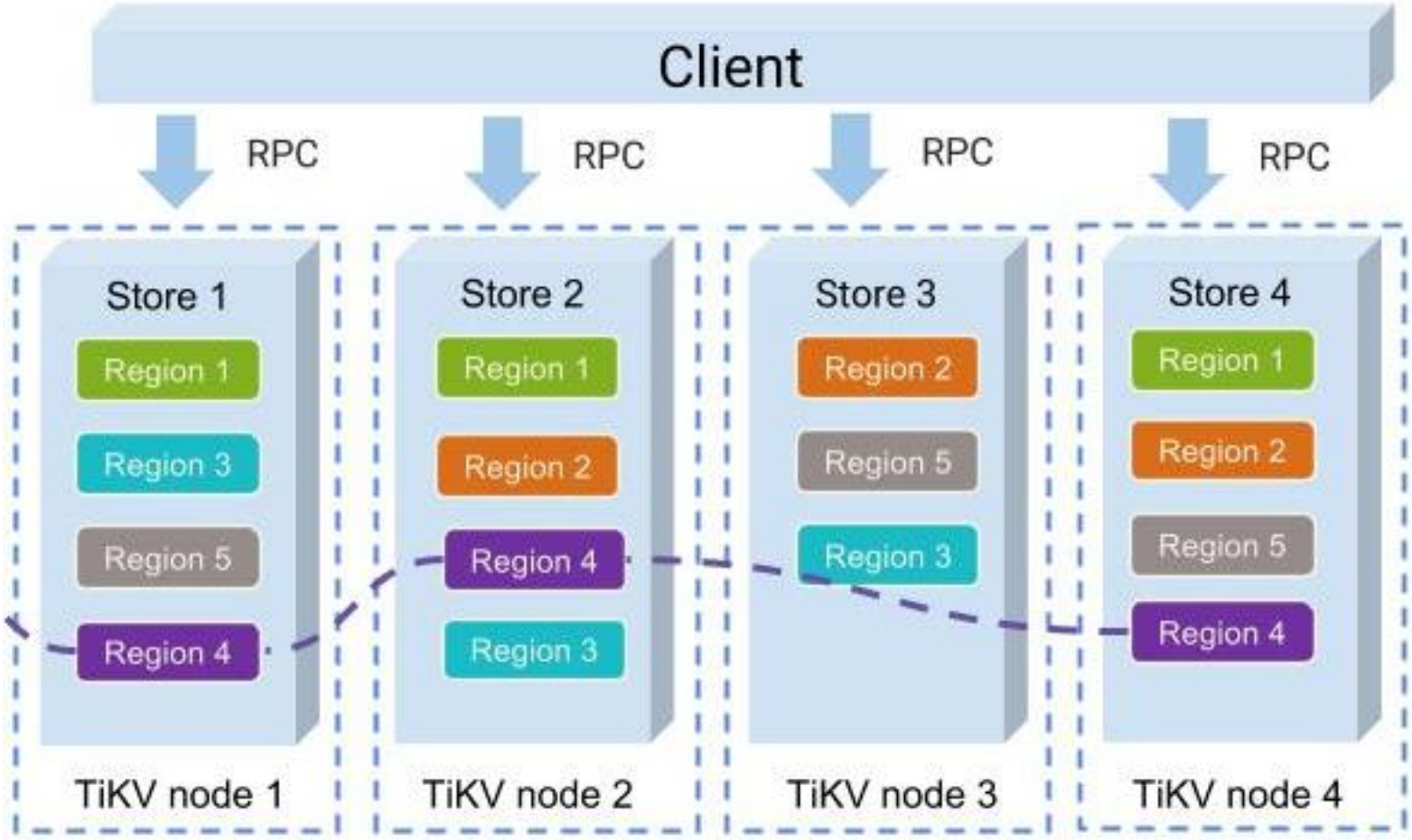
1.3 分布式数据库基本概念

2、分布式数据库管理的任务

1）持久地存储数据

- 数据必须持久地存储起来。
- 数据只有被存储在磁盘、闪存、磁带等长期存储设备上才可以称为持久化存储数据。
- 在分布式数据库中，为了保证数据的可用，同一份数据有多个数据副本。

1.3 分布式数据库基本概念

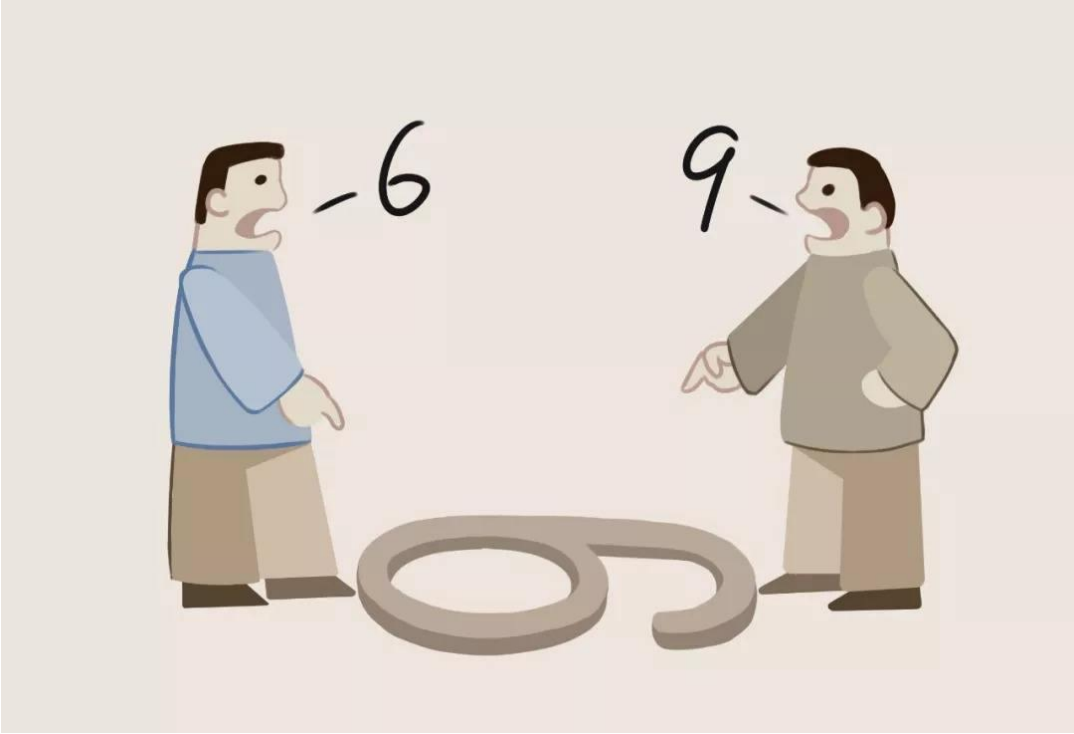


多个数据副本

1.3 分布式数据库基本概念

2）维护数据的一致性

– 用户所查询的数据与数据库中实际存储的数据的一致性，以及不同数据副本之间的一致性。



1.3 分布式数据库基本概念

2）维护数据的一致性

根据一致性的标准高低，可以分为以下几种：

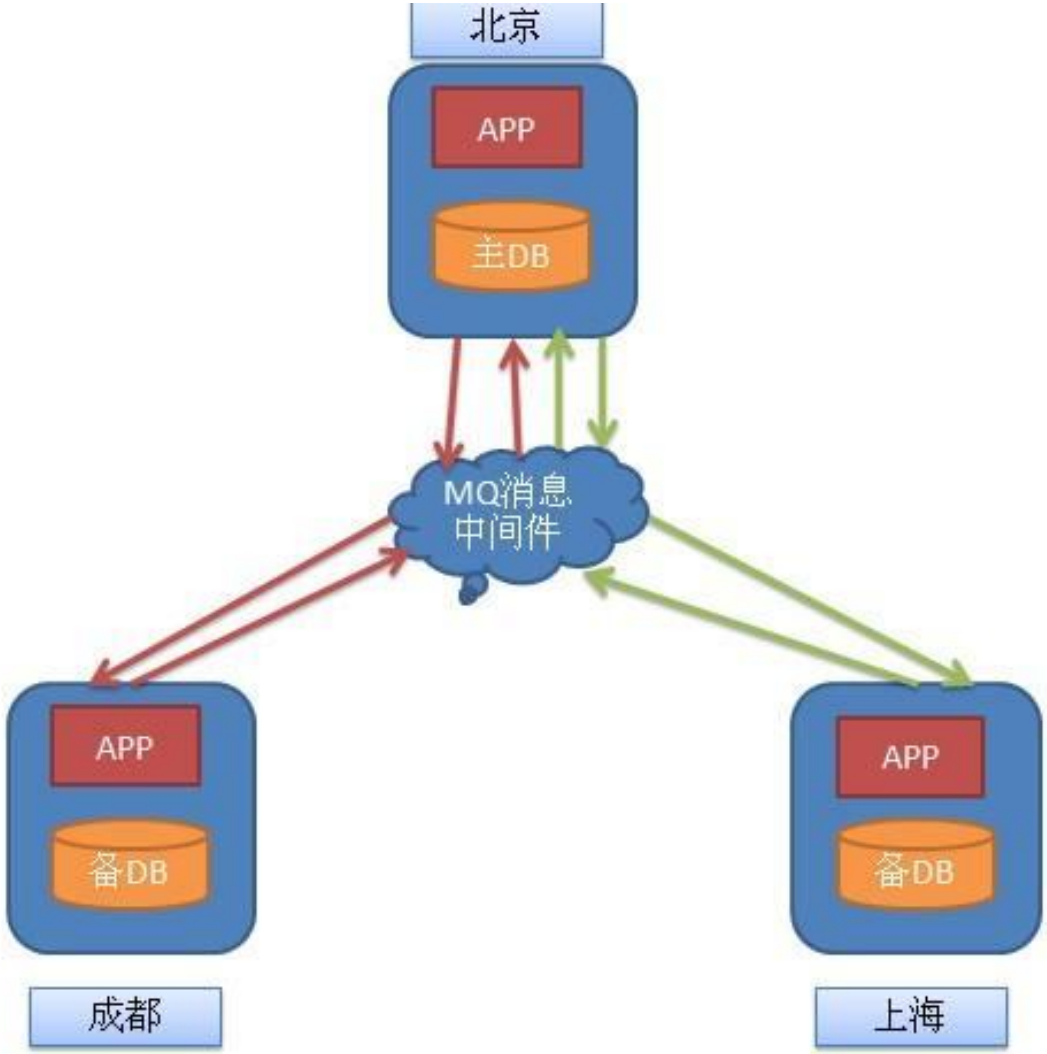
① 强一致性（Strong Consistency）：

任何时刻所有的用户查询到的都是最近一次成功更新的数据。这是要求最高的标准，也是最难实现的，关系型数据库采用该标准。

② 弱一致性（Weak Consistency）：

某个用户更新了某数据副本，但是系统不能保证后面的用户能够读取到最新的值。

1.3 分布式数据库基本概念



1.3 分布式数据库基本概念

③ 最终一致性（Eventual Consistency）：

最终一致性是弱一致性的一种特例。某用户更新了副本的数据，如果没有其他用户更新这个副本的数据，系统最终一定能够保证后续用户能够读取到该用户写的最新值。

最终一致性存在一个不一致性的窗口，也就是用户写入数据到其他用户读取所写的新值所用的时间。

更新数据副本所花费的时间取决于副本数量、系统负载、网络速度等因素。

1.3 分布式数据库基本概念

最终一致性又可以根据更新数据后各用户访问到数据的时间和方式的不同，进一步分为以下5种：

a） 因果一致性（Causal Consistency）

如果用户A通知用户B它已更新了一个数据项，那么用户B的后续访问将返回更新后的值，且一次写入将保证取代前一次写入。与用户A无因果关系的用户C的访问遵守一般的最终一致性规则。

b） 读写一致性（Read-Your-Writes Consistency）

当用户A自己更新一个数据项之后，它总是访问到更新过的值，绝不会看到旧值。这是因果一致性模型的一个特例。

1.3 分布式数据库基本概念

c）会话一致性（Session Consistency）

把访问系统的所有用户放到会话的上下文中，只要会话还存在，系统就保证读写一致性。如果由于某些失败情形令会话终止，就要建立新的会话，而且系统保证不会延续到新的会话。

d）单调读一致性（Monotonic Read Consistency）

如果用户已经看到过数据对象的某个值，那么任何后续访问都不会返回在那个值之前的值。

e）单调写一致性（Monotonic Write Consistency）

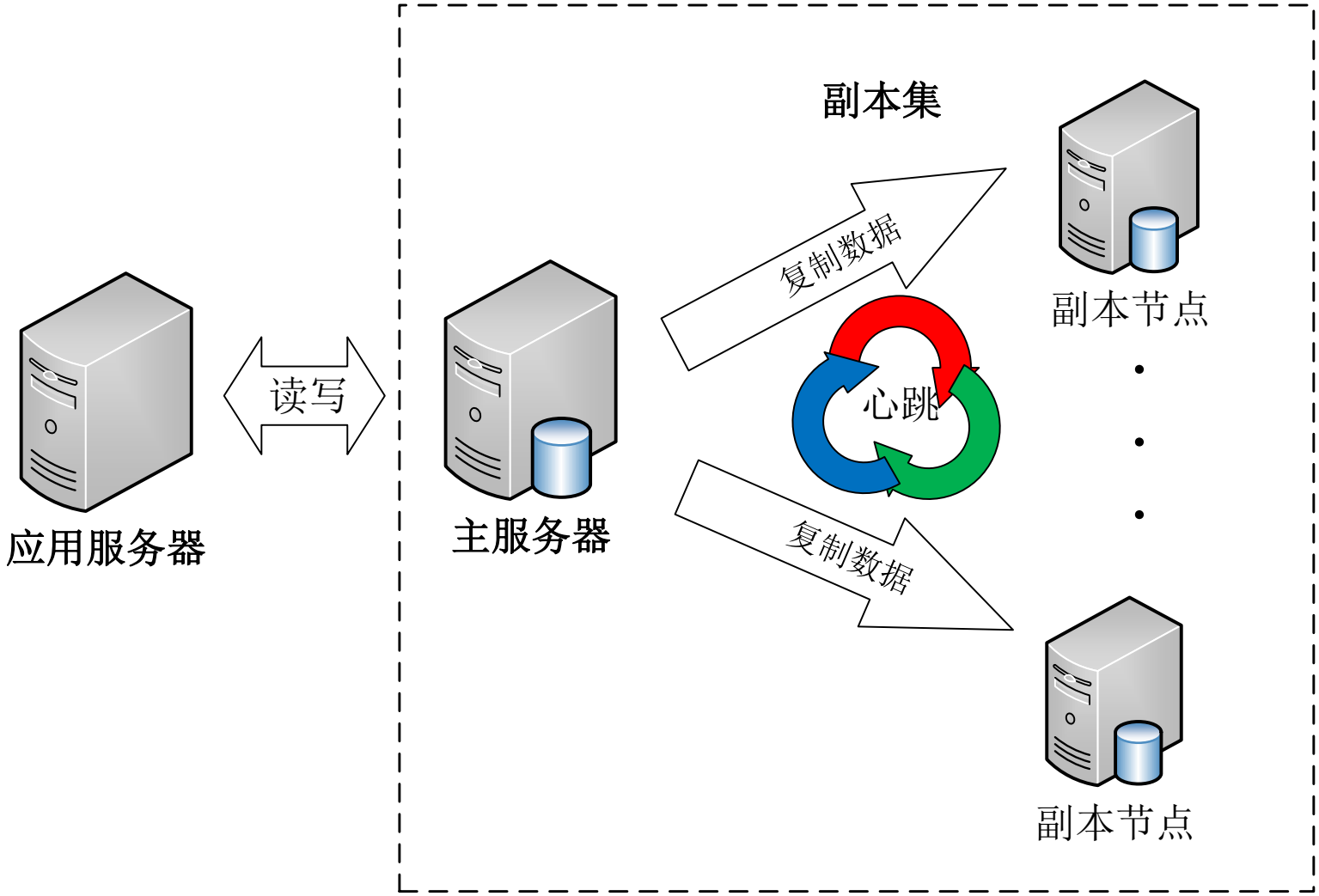
系统保证来自同一个用户的写操作按顺序执行。

1.3 分布式数据库基本概念

3）保证数据的可用性

- 数据库可用性指数据库应该随时可供使用，由于硬件故障难以避免，只依赖一台服务器的数据库很难保障其可用性。
- 一般通过设置主数据库服务器和备份服务器，并确保备份服务器与主服务器中数据的一致性。
  - 通过两阶段提交策略：先写主服务器，再写备份服务器。
  - 两个服务器都要求正确写入，但是如果存在多个备份备份服务器，其开销将增加。

1.3 分布式数据库基本概念



1.4 CAP、ACID与BASE原理

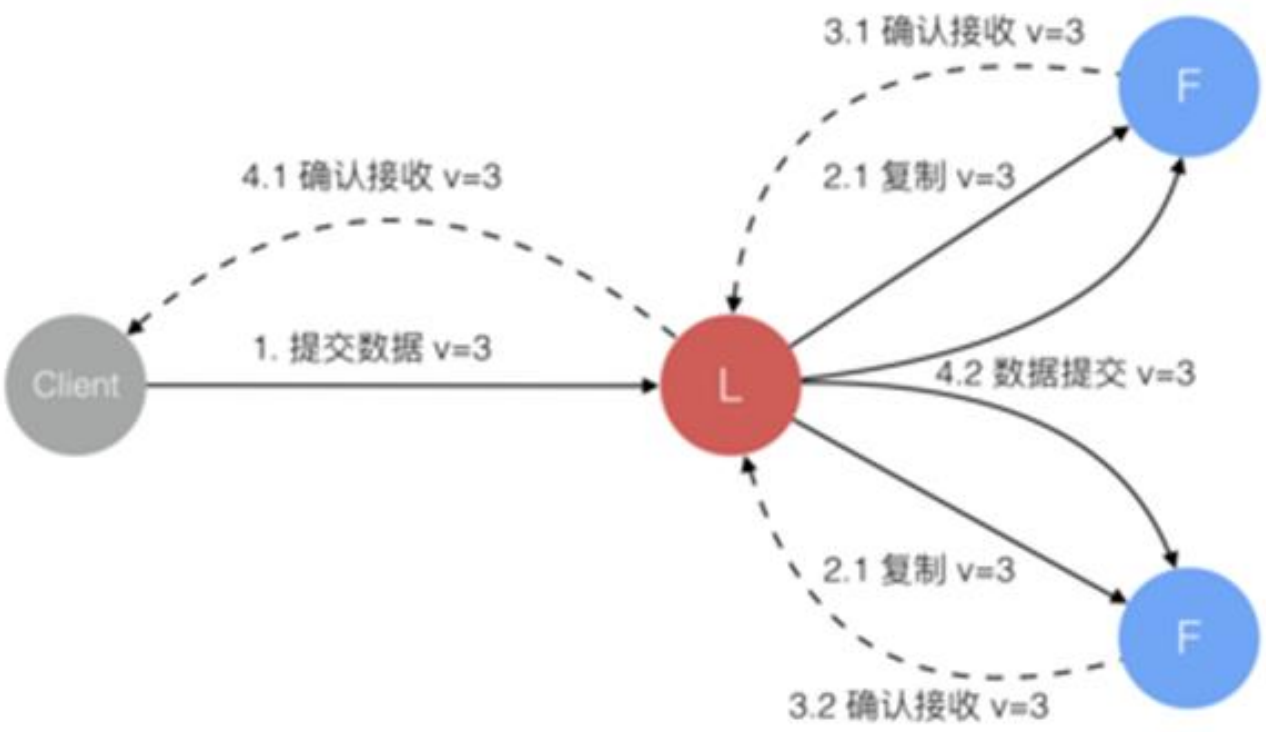
1、CAP理论

该理论由Eric Brewer提出，也称为Brewer理论。

在一个分布式系统中，一致性（Consistency）、可用性（Availability）、分区容错性（Partition tolerance）三个基本需求最多只能同时满足两个，不可能三者兼顾。

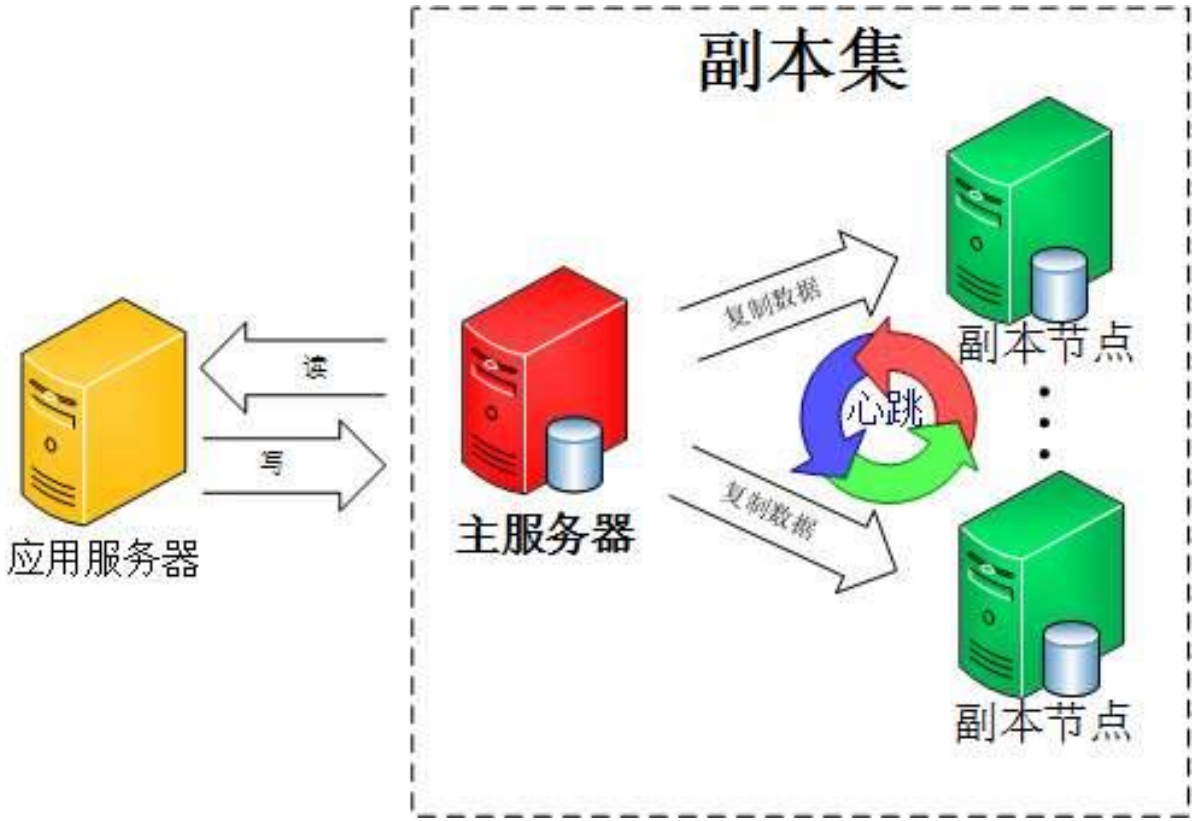
1.4 CAP、ACID与BASE原理

- 一致性（C）：各服务器中的数据副本要彼此相同，当一个节点的数据更新完成后，要求其他副本的数据与此相同。



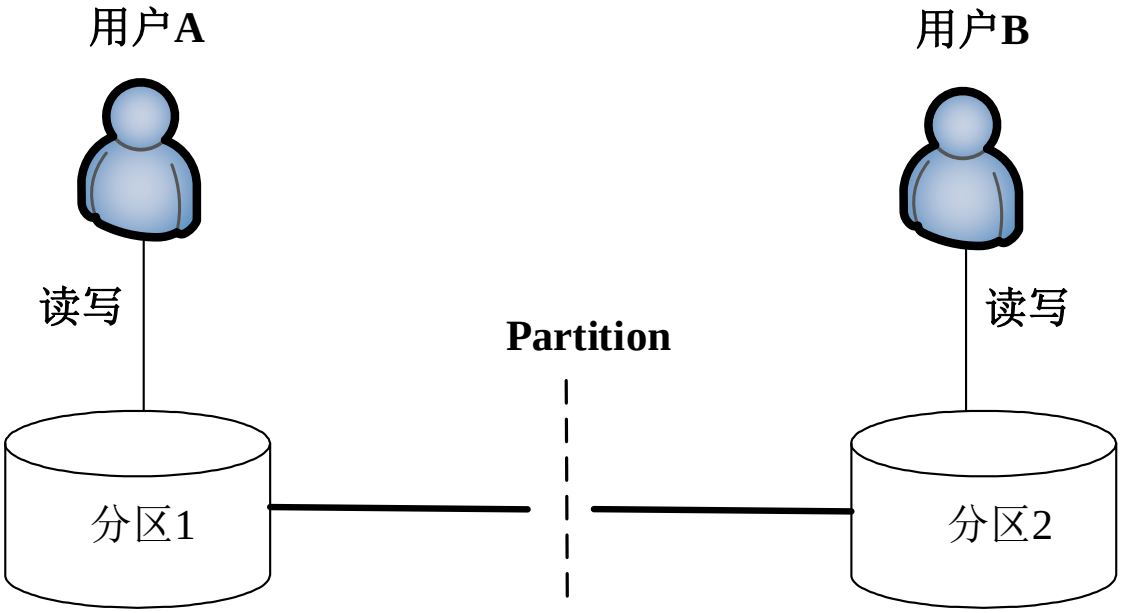
1.4 CAP、ACID与BASE原理

- 可用性（A）：系统能够持续不断地提供相应服务，响应任意的查询请求。



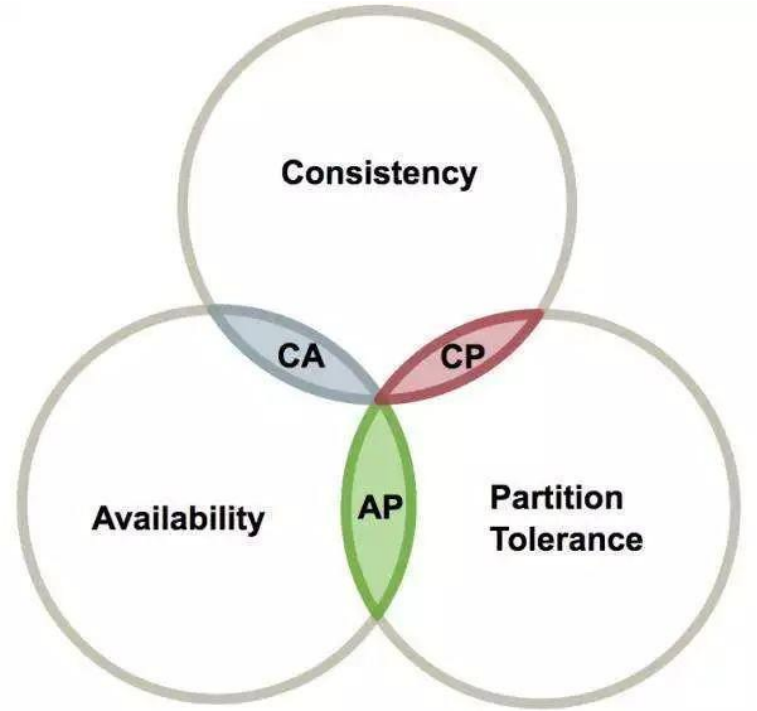
1.4 CAP、ACID与BASE原理

- 分区容错性（P）： 由于网络故障，集群被切分成了若干个孤立的区域，这就是分区。分区容错性要求各分区之间虽然不能够通信，但各分区依然可以提供数据服务。



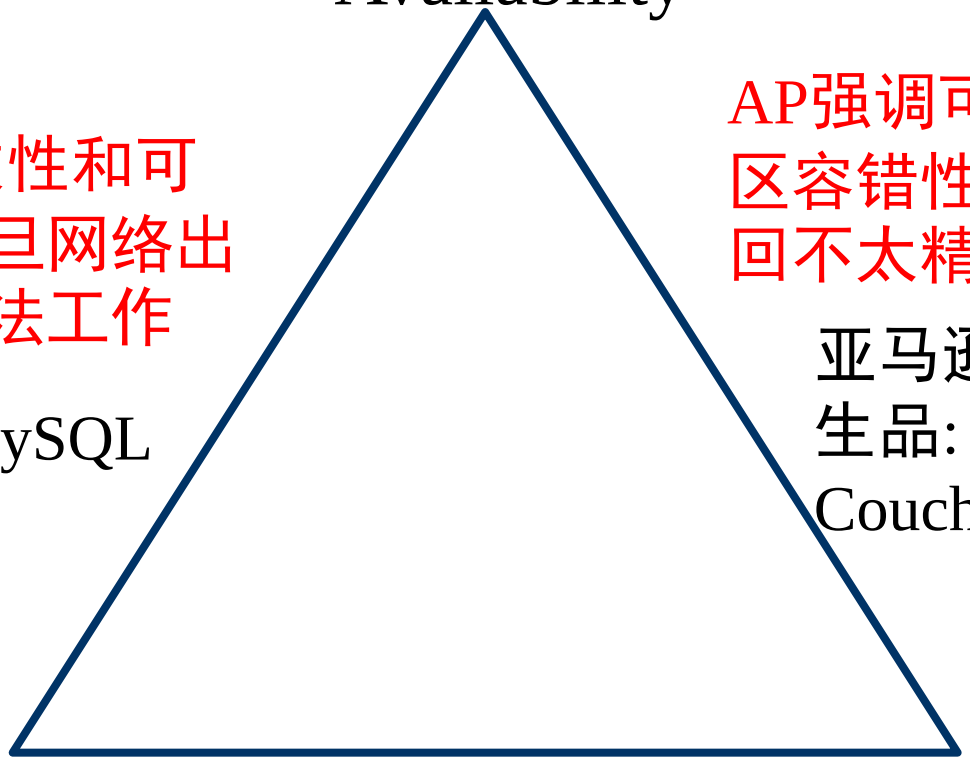
1.4 CAP、ACID与BASE原理

一般地，分布式数据库必须尽力在网络发生故障的情况下继续工作，因此分区容错性是基本要求，接下来要权衡可用性和一致性，即C和A二选一。



CAP理论的精髓就是要么AP，要么CP，要么AC，但是不存在CAP。

1.4 CAP、ACID与BASE原理



CA强调一致性和可用性，但一旦网络出现故障则无法工作

SQL Server、MySQL

AP强调可用性和分区容错性，但可能返回不太精确的数据。

亚马逊Dynamo的衍生品: Cassandra、CouchDB、Riak

CP强调一致性和分区容错性，如果节点发生故障，部分数据不可用。

Neo4j、Bigtable、MongoDB、Hbase、Redis

1.4 CAP、ACID与BASE原理

2、事务ACID特性

ACID是指事务必须具备的四个特性:

(1)原子性(Atomicity)：每个事务是不可分割的数据库逻辑工作单位。

(2)一致性(Consistency)：事务的执行结果必须使数据库从一个一致性状态变到另一个一致性状态。

(3)隔离性(Isolation)：并发执行的各个事务之间不能相互干扰。

(4)持续性(Durability)：持续性也称为永久性，指一个事务一旦提交，它对数据库中数据的改变应该是永久性的。

1.4 CAP、ACID与BASE原理

3、BASE原理

BASE原理与ACID原理不同，它满足CAP原理，通过牺牲强一致性获得可用性，通过达到最终一致性来尽量满足业务的绝大多数需求，是NoSQL的性质：

- BA：Basically Available 基本可用，如果分布式系统中的某些服务器出现了故障，其他服务器仍然可以继续提供服务。
- S：Soft state 软状态，数据库中的数据最终会被新值所覆盖。
- E：Eventually consistency 最终一致性，数据副本有可能在短时间内出现彼此不一致的现象，但最终将完成所有副本的更新，保持一致性。

1.5 非关系式数据库概述

1、非关系式数据库基本特征

1）不需要预定义模式：每条记录都可能有不同的属性和格式，其模式在插入数据时动态定义。

2）弹性可扩展：可以动态增加或者删除集群节点，数据可自行迁移。

3）数据分区存放：将数据进行分区之后存储在多个节点上，节点间自动进行数据复制，制作多个数据副本。

4）异步复制数据：将数据先写入一个节点，再根据数据备份的数量复制数据，这样可以尽快地写入一个节点。

5）BASE：相对于事务严格的ACID特性，NoSQL数据库保证的是BASE特性。

1.5 非关系式数据库概述

1、非关系式数据库基本特征

非关系式数据库适用的范围：

1）数据模型比较简单；

2）需要灵活性更强的集群系统；

3）关注更高的性能；

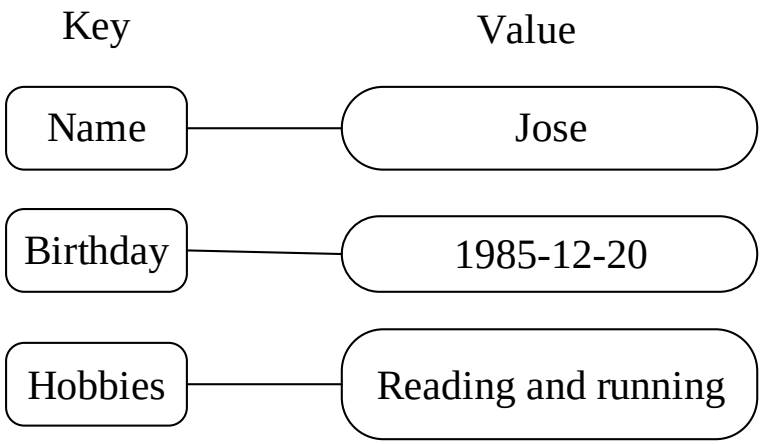
4）不需要保证强一致性。

1.5 非关系式数据库概述

2、键值数据库

键值数据库将数据存储为键值对集合，其中键作为唯一标识符，指向特定的数据。

键值模型对于IT系统来说的优势在于简单、易部署、高并发。



桶1	
S01	张杰
S02	李玮
S03	王韩

桶2	
S01	岑水
S03	蒋珊
S05	陆阳

桶3	
S01	黄河
S04	王杰
S07	李飞

1.5 非关系式数据库概述

键值数据库采用关联数组作为数据结构，也称为Hash表，即（Key，Value）

- 键（Key）：用来查询值的唯一标识符，必须互不相同。
- 值（Value）：是一个实例，必须与一个唯一的Key相关联，可以是任意的数据类型。
- 命名空间键(namespace)：键值对构成的集合，称为桶（bucket）或数据库(database)。
  - 同一个命名空间中的键不允许相同，不同命名空间中的键可以相同。
- 分区：根据键名，把数据分割成不同的单元，存储在集群中的不同服务器上，实现负载均衡。

1.5 非关系式数据库概述

3、文档数据库

文档数据库将数据以文档模式存储，若干个文档形成一个集合，若干个集合构成了一个独立的文档数据库。

每个文档都是自包含的数据单元，是一系列数据项的集合，每个数据项都有一个名称与对应的值，值既可以是简单的数据类型，如字符串、数字和日期等；也可以是复杂的类型，如有序列表和关联对象。

1.5 非关系式数据库概述

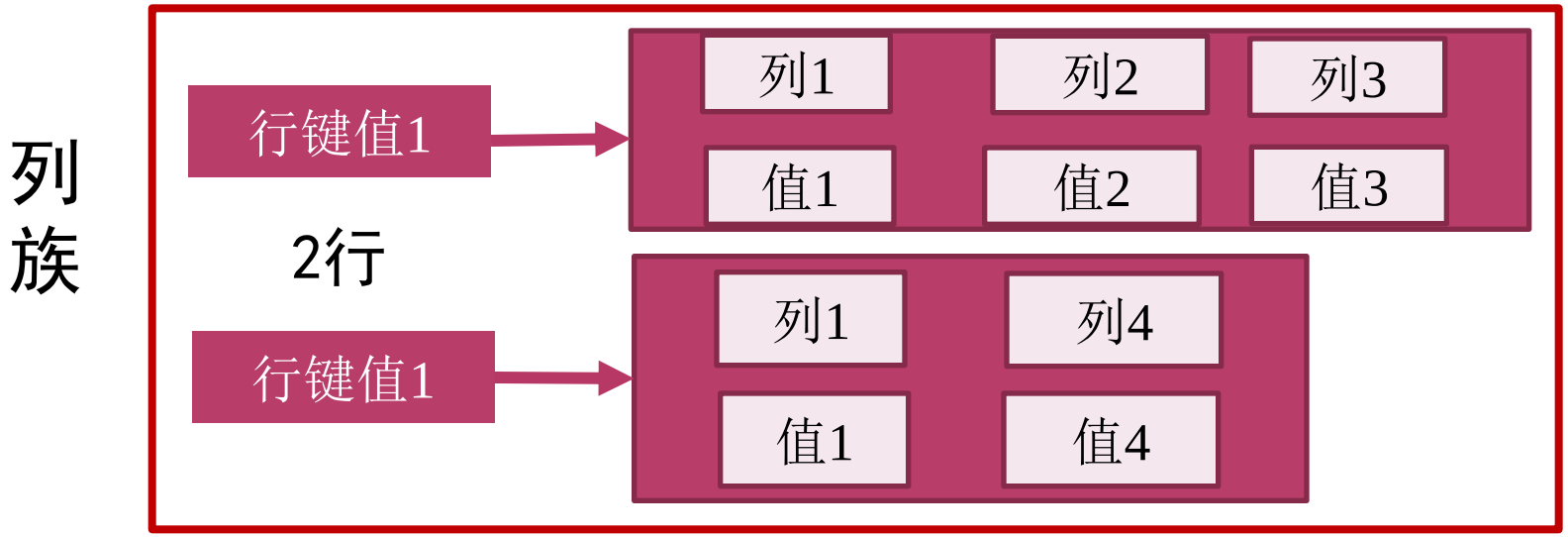
id	user_name	email	age	city
1	Mark Hanks	mark@abc.com	25	Los Angeles
2	Richard Peter	richard@abc.com	31	Dallas

```
{
    "_id": ObjectId("5146bb52d8524270060001f3"),
    "age": 25,
    "city": "Los Angeles",
    "email": "mark@abc.com",
    "user_name": "Mark Hanks"
}
{
    "_id": ObjectId("5146bb52d8524270060001f2"),
    "age": 31,
    "city": "Dallas",
    "email": "richard@abc.com",
    "user_name": "Richard Peter"
}
```

1.5 非关系式数据库概述

4、列族数据库

列族数据库是一种采用列式存储的数据库，面向大体量数据，具有高可靠性、高性能、可伸缩等特点。



Keyspace:Keyspace1

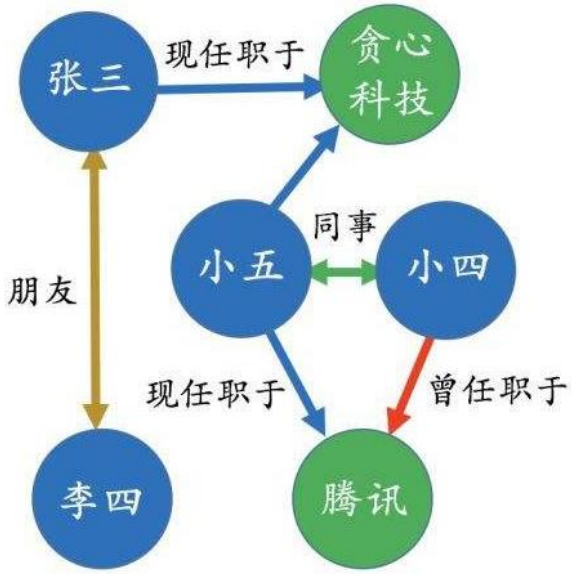
ColumnFamily:Standard2

Key	Columns
studentA	Columns name | value | timestamp "age" | "18" | 1270694041669000 "height" | "172cm" | 1270694041669000
classA	SuperColumns Key | Columns studentX | Columns name | value | timestamp "age" | "20" | 1270694041669000 "height" | "182cm" | 1270694041669000

1.5 非关系式数据库概述

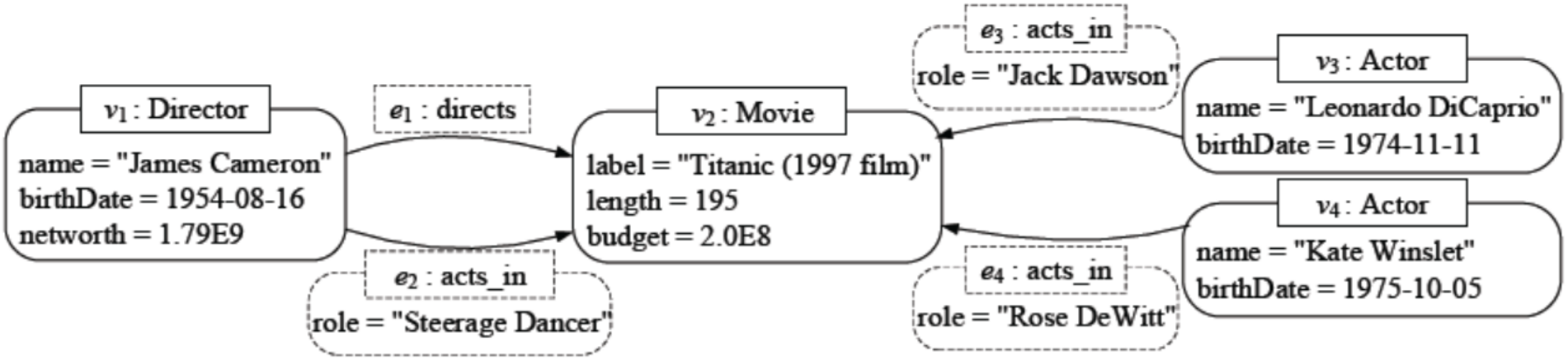
5、图数据库

图数据库是一种基于图论的新型NoSQL数据库，它的数据存储结构和数据查询方式都是以图论为基础。



- 使用节点、边、属性等对象表示和存储数据以及数据之间的关系。

1.5 非关系式数据库概述



- 图有节点和边组成；
- 节点可以有属性，也可以有一个或多个标签；
- 关系有名字和方向，并总是有要给开始节点和一个结束节点，也可以有属性。

1.5 非关系式数据库概述

6、NoSQL数据库适用场景

（1）键值数据库

键值数据库的特点在于高效，适用于读取操作频繁，但写操作较少的场合，其采取的键值对模型非常简单，不适合于存储复杂结构的数据。

以下情况可以考虑选取键值数据库：

- 存储用户配置、会话信息、购物车等信息，这些信息都和一个ID（键）关联。
- 存储图像、音频等大型对象文件。
- 缓存数据，改善性能。

1.5 非关系式数据库概述

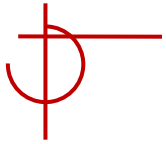
6、NoSQL数据库适用场景

（2）文档数据库

文档数据库的特点是灵活、高效和易用。如果应用系统需要存储大量的数据，而且这些数据的格式不固定、灵活多样、不强调规范化，那么可以选取文档数据库。

以下情况可以考虑选取文档数据库：

- 属性不确定且多变的数据对象，如商品数据。
- 记录各种数据的元数据信息。
- 使用JSON结构存储数据。
- 通过去规范化在大结构中嵌套小结构的数据存储。
- 需要功能强大的查询功能，能够根据文档属性进行查询和统计。



1.5 非关系式数据库概述

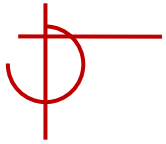
6、NoSQL数据库适用场景

（3）列族数据库

列族数据库适合于大体量的数据读写，一般是部署在分布式集群中，具有高可用性和高性能等特点。

以下情况可以考虑选取列族数据库：

- 处理数据量巨大，频繁读写。
- 字段经常变化，且不确定。
- 要求高可用，允许数据不一致现象。
- 适合于搜索引擎、网络通信日志、社交服务等领域。



1.5 非关系式数据库概述

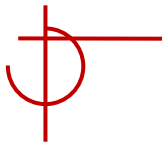
6、NoSQL数据库适用场景

（4）图数据库

图数据库适合于社交网络、交通网络、知识图谱等需要关联分析的领域，然后在此模型基础上进行个性化推荐、智能问答和知识推理等。

以下情况可以考虑选取图数据库：

- 网络与IT基础设施管理。
- 产品推荐与服务。
- 社交网络分析。
- 交易数据分析。



Chapter over