课程内容提要

希赛

- 规范化与反规范化（★★★）
- 分布式数据库（★★★）
- NoSQL（★★★）
- 内存数据库（★★★）
- 其他数据库扩展知识（★★★）

课程内容提要

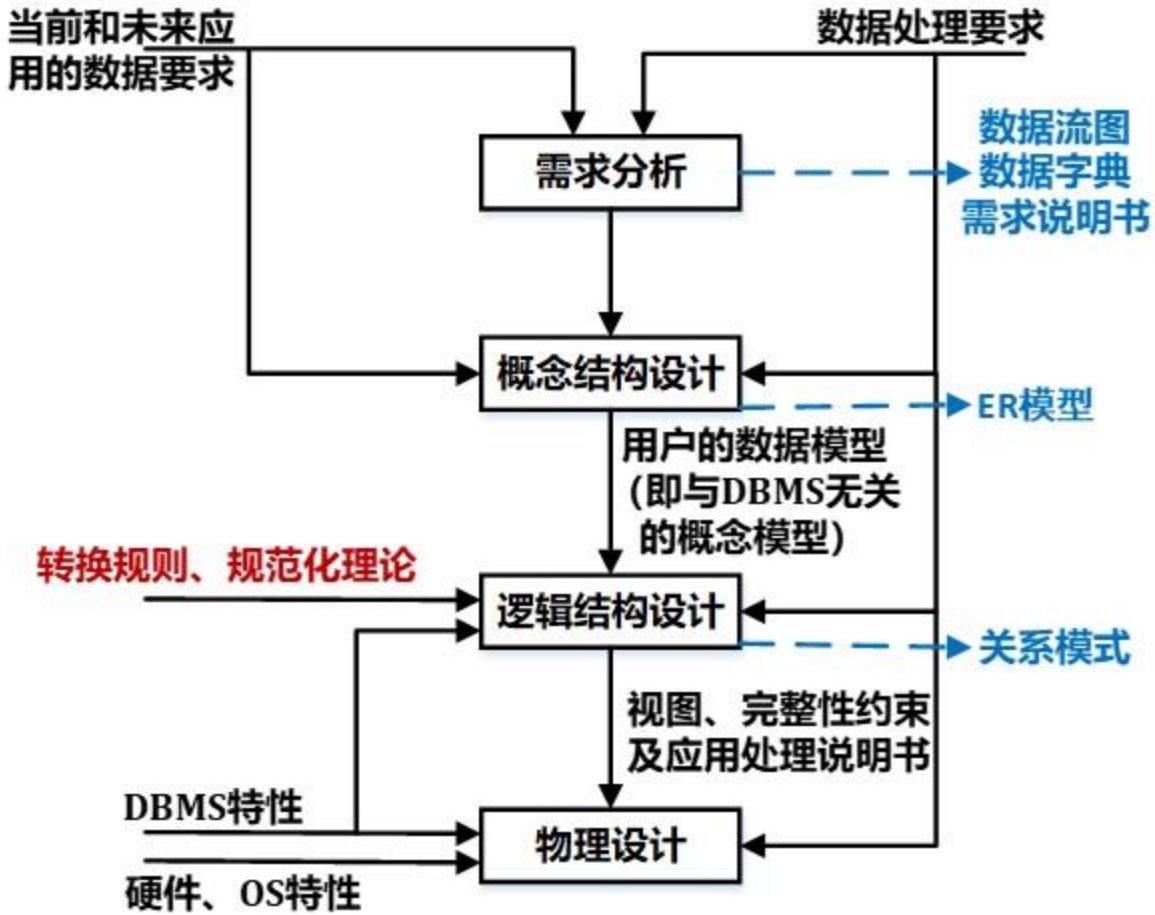
希赛

数据库设计关注的问题：

性能　　数据一致性　　安全

规范化 – 数据库设计过程

希赛

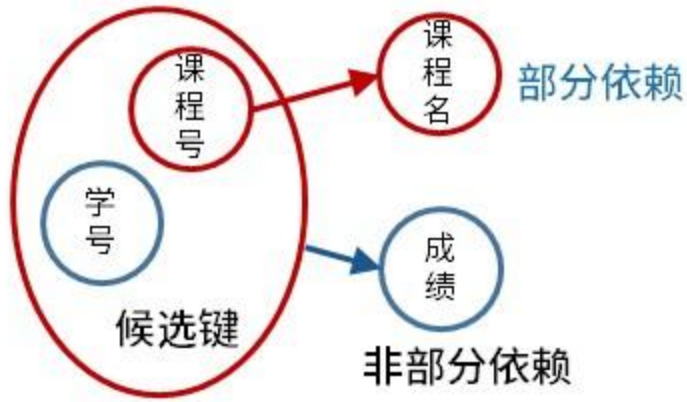


规范化 – 范式

希赛

逐步优化，以解决：
数据冗余、插入、修改、删除异常

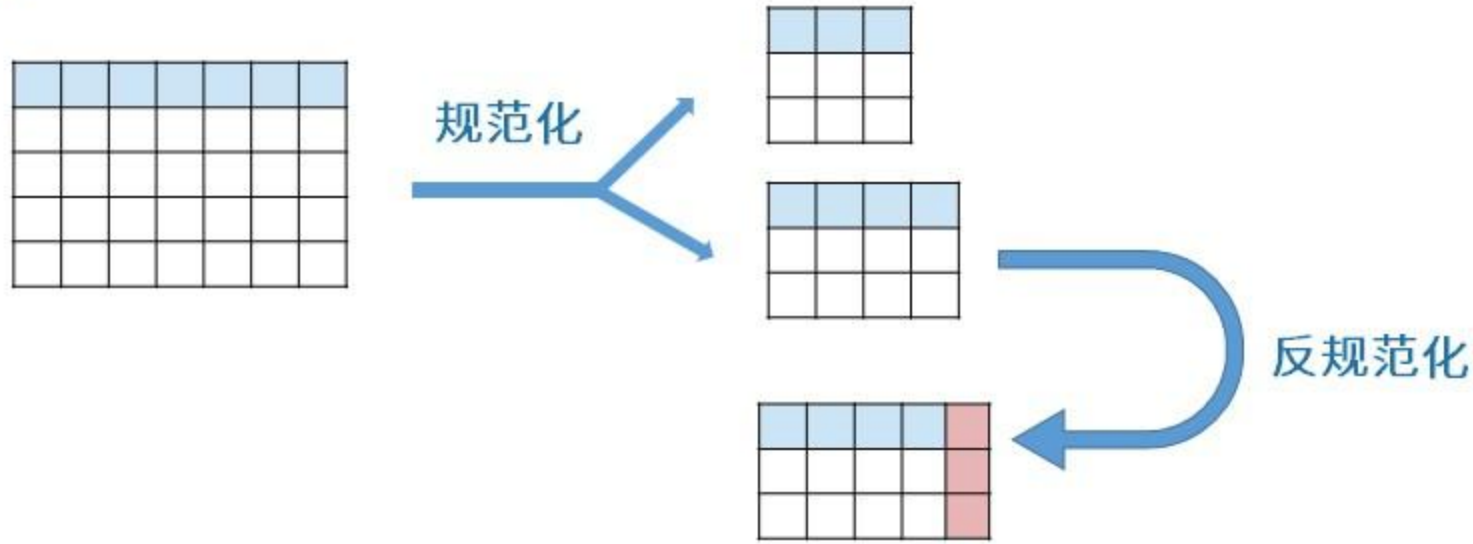
- 1NF：属性值都是不可分的原子值
- 2NF：消除非主属性对候选键的部分依赖
- 3NF：消除非主属性对候选键的传递依赖
- BCNF：消除主属性对候选键的部分和传递依赖



思考：范式级别提升带来了什么负面影响？

反规范化

希赛



技术手段	说明
增加派生性冗余列	已有单价和数量列，增加“总价”列
增加冗余列	已有学号列，增加“姓名”列
重新组表	把拆分的表重新组表
分割表	把用户表做水平分割，长沙的用户存在长沙，上海的用户存在上海

反规范化

希赛

反规范化的**优点**：连接操作少，检索快、统计快；需要查的表减少，检索容易。

反规范化的缺点	解决方案
数据冗余，需要更大存储空间	无解
插入、更新、删除操作开销更大	无解
数据不一致 可能产生添加、修改、删除异常	1、触发器数据同步 2、应用程序数据同步
更新和插入代码更难写	无解

案例分析练习题1

希赛

某集团公司在各省均设有分公司，现欲建立全国统一的销售管理信息系统，以便总公司及时掌握各分公司的销售情况。公司成立专门的项目组进行该系统的研发工作，其中张工负责其中的数据库设计工作。

张工和需求分析小组紧密合作，在设计出数据流图和数据字典的基础上，给出了数据库关系模式和相应的索引设计。同时考虑到未规范化关系模式可能引起的各类数据错误，对关系模式进行了全面的规范化处理，使所有关系模式均达到了3NF或BCNF。

在项目实施过程中，应用开发小组认为该设计方案未考虑应用功能的实际需求。如果严格按照设计方案实施，会对应用系统中整体性能产生较大影响。主要的原因在于进行数据查询时，会产生大量的多表连接操作，影响性能。而设计方案中的索引设计，并不能完全满足数据查询的性能要求。

应用开发小组还认为，该设计方案未考虑到信息系统中核心销售数据处理的特点：各分公司在使用该信息系统时只能操作自己分公司的销售数据，无权操作其它分公司的销售数据；只有总公司有权利操作所有销售数据，以便进行统计分析。

应用开发小组要求，在数据库设计方案中，必须针对实际应用功能的实现来考虑关系模式的规范化，必要时需要采用逆规范化或解除规范化的方法来保证性能要求。

案例分析练习题1

希赛

【问题1】（8分）

系统需要管理供应商和货物等信息，具体包括供应商姓名、地址以及货物名称、价格等，供应商可以提供0~n种货物，其公司地址也可能发生变化。请以供应商关系模式supplier(name,address,product,price)为例，解释不规范的关系模式存在哪些问题。

【问题2】（6分）

应用开发小组认为张工的规范化设计虽然解决了未规范化关系模式带来的问题，但实际实现功能时会造成系统性能的下降，请解释其原因。

【问题3】（5分）

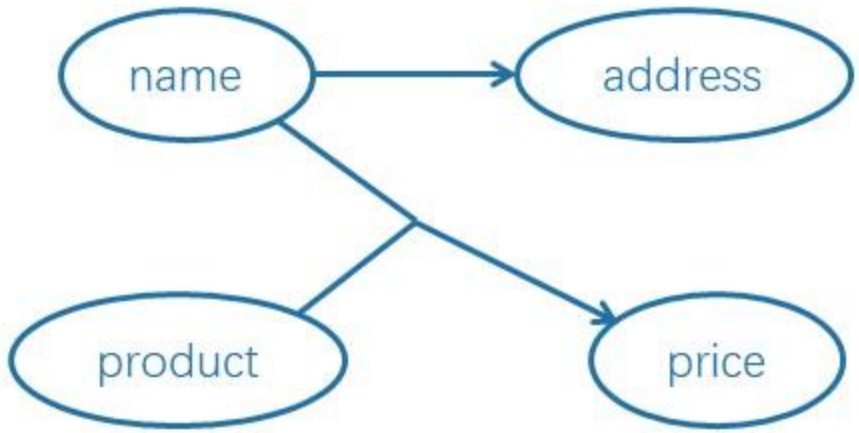
请解释逆规范化方法，说明其优缺点。

【问题4】（6分）

针对该信息系统中核心销售数据处理的特点，如采用关系表水平分割的逆规范化方法，请给出具体的解决方案，并说明该方案存在的问题。

案例分析练习题1

希赛



案例分析练习题1- 参考答案

希赛

【问题1】（8分）

（1）数据冗余：关系模式中多次重复记录了同一供应商的地址。

（2）插入异常：如果还未确定一个供应商有哪些货物，只是想添加一个供应商的地址信息，则会产生产品与价格均为空的记录。

（3）修改异常：当修改一个供应商的地址时，需要将多条记录同时更新，若未同时更新，则数据产生不一致。

（4）删除异常：当删除一个供应商的货物时，其地址信息被一并删除。

【问题2】（6分）

数据库规范化的过程，实际是对数据表的不断拆分，以达到更高的规范程度。这样处理，带来的问题是：系统中大量查询不能通过单表完成，而需要将多表进行连接查询，所以表拆分得越多，查询性能也就越差。

【问题3】（5分）

规范化设计后，数据库设计者希望牺牲部分规范化来提高性能，这种从规范化设计的回退方法称为反规范化技术。

逆规范化方法优点：提高统计、查询效率。

逆规范化方法缺点：增加了数据冗余，浪费存储空间，增、删、改操作的效率降低，可能导致数据不一致，可能产生添加、修改、删除异常。

【问题4】（6分）

解决方案：将各省的数据存放于各省分公司。

该方案主要问题：

（1）在于总公司进行全国数据统计时，需要从各省服务器调取数据，效率较低。

（2）执行应用功能时需要动态选择分公司的数据库表，增加了应用程序的复杂度。

数据库索引

希赛

数据库索引：提升查询效率，降低添加、修改、删除效率。采用B树，B+树等。

数据表

学号	姓名	性别
001	张三	男
102	李四	女
008	王五	男
007	赵六	男
235	钱七	女
417	李华	男
907	张明	女
…	…	…
018	龙四	女

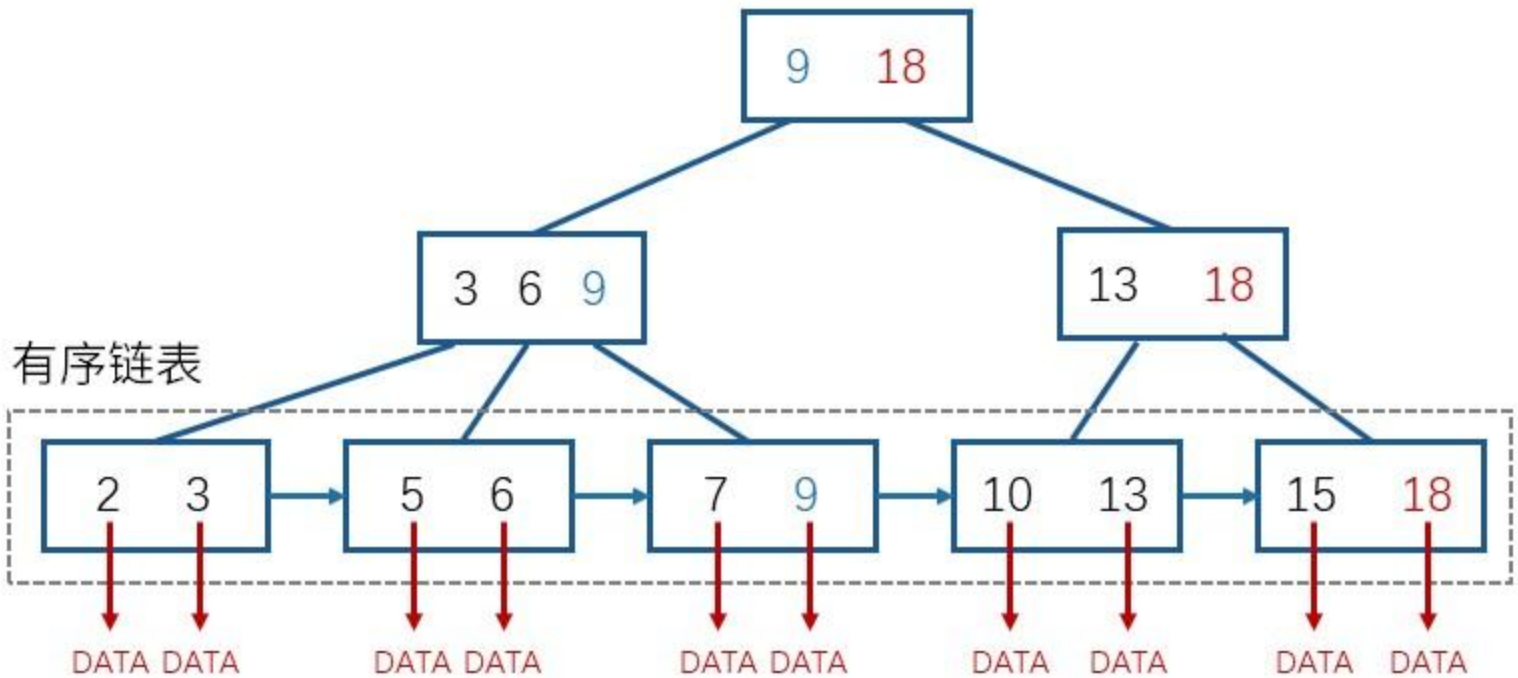
索引表

学号	记录指针
1	data
7	data
8	data
18	data
102	data
235	data
417	data
907	data
…	…

数据库索引

希赛

B+树示例



数据库视图

希赛

视图(View)并不在数据库中实际存在，而是一种虚拟表。

学生信息表

学号	姓名	性别
N001	张三	男
N002	李四	女

课程信息表

课程号	课程名	授课老师
C001	数据结构	王老师
C002	Java语言	李老师

选课关系表

学号	课程号	成绩
N001	C001	85
N002	C002	74

学员选课视图

学号	课程号	姓名	课程名	成绩
N001	C001	张三	数据结构	85
N002	C002	李四	Java语言	74

数据库视图

希赛

视图的**优点**：

1、视图能**简化**用户的**操作**

2、视图机制可以使用户以不同的方式查询同一数据

3、视图对数据库重构提供了一定程度的**逻辑独立性**

4、视图可以对机密的数据提供**安全保护**

物化视图：将视图的内容物理存储起来，其数据随原始表变化，同步更新。

案例分析练习题

希赛

【说明】

某软件企业开发一套类似于淘宝网上商城业务的电子商务网站。该系统涉及多种用户角色，包括购物用户、商铺管理员，系统管理员等。

在数据库设计中，该系统数据库的核心关系包括：

产品（产品编码，产品名称，产品价格，库存数量，商铺编码）

商铺（商铺编码，商铺名称，商铺地址，商铺邮箱，服务电话）

用户（用户编码，用户名称，用户地址，联系电话）

订单（订单编码，订单日期，用户编码，商铺编码，产品编码，产品数量，订单总价）

不同用户角色有不同的数据需求，为此该软件企业在基本数据库关系模式的基础上，定制了许多视图。其中，有很多视图涉及到多表关联和聚集函数运算。

案例分析练习题

希赛

【问题1】（8分）

商铺用户需要实时统计本商铺的货物数量和销售情况，以便及时补货，或者为商铺调整销售策略。为此专门设计了可实时查看当天商铺中货物销售情况和存货情况的视图，商铺产品销售情况日报表（商铺编码，产品编码，日销售产品数量，库存数量，日期）。

数据库运行测试过程中，发现针对该视图查询性能比较差，不满足用户需求。

请说明数据库视图的基本概念及其优点，并说明本视图设计导致查询性能较差的原因。

【问题2】（8分）

为解决该视图查询性能比较差的问题，张工建议为该数据建立单独的商品当天货物销售、存货情况的关系表。但李工认为张工的方案造成了数据不一致的问题，必须采用一定的手段来解决。

1）说明张工方案是否能够对该视图查询性能有所提升，并解释原因。

2）解释说明李工指出的数据不一致问题产生的原因。

【问题3】（9分）

针对李工提出的问题，常见的解决手段有应用程序实现，触发器实现和物化视图实现等，请用300字以内的文字解释说明这三种方案。

案例分析练习题- 参考答案

希赛

【问题1】

视图是虚表，是从一个或几个基本表（或视图）中导出的表，在系统的数据字典中仅存放了视图的定义，不存放视图对应的数据。

视图的优点：

1、视图能简化用户的操作

2、视图机制可以使用户以不同的方式查询同一数据

3、视图对数据库重构提供了一定程度的逻辑独立性

4、视图可以对机密的数据提供安全保护

查询性能较差的原因是视图中“日销售产品数量”需要针对订单表做统计分析，订单表中有数量庞大的历史销售记录，所以这种操作极为耗时。

【问题2】

1）张工方案能够对该视图查询性能有所提升，因为这样做能极大的减少统计分析的数据量，对小数据量进行统计，性能是能得以保障的。

2）由于当日订单数据既存储在订单表中，又存储在单独的当天货物销售、存货情况表中。同一数据存储了两份，一旦出现修改，未同步修改，则会造成数据不一致。

【问题3】

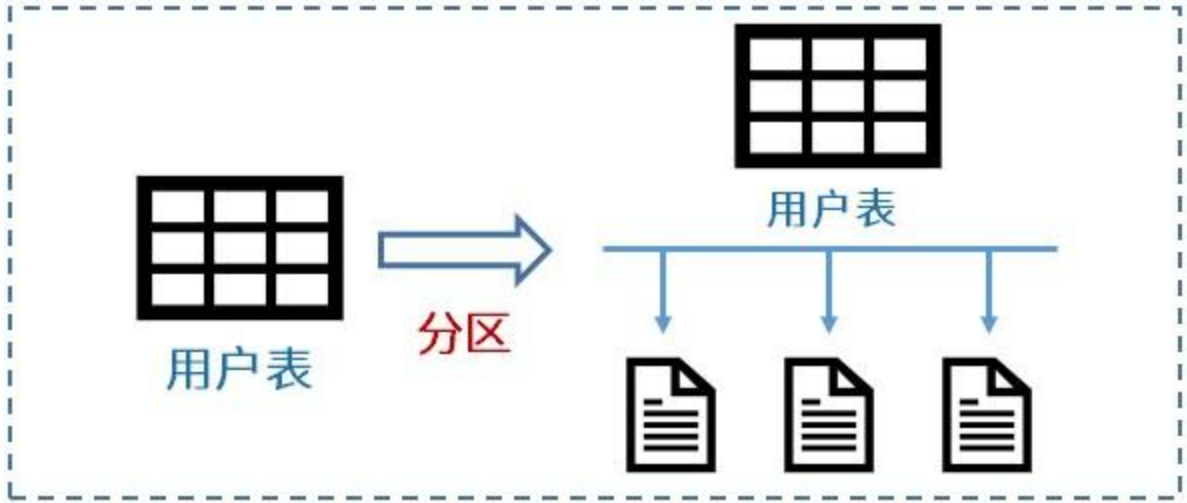
应用程序实现：在进行订单的添加、修改、删除操作时，从应用程序中，控制对两个数据表都进行相关操作，以保障数据的一致性。

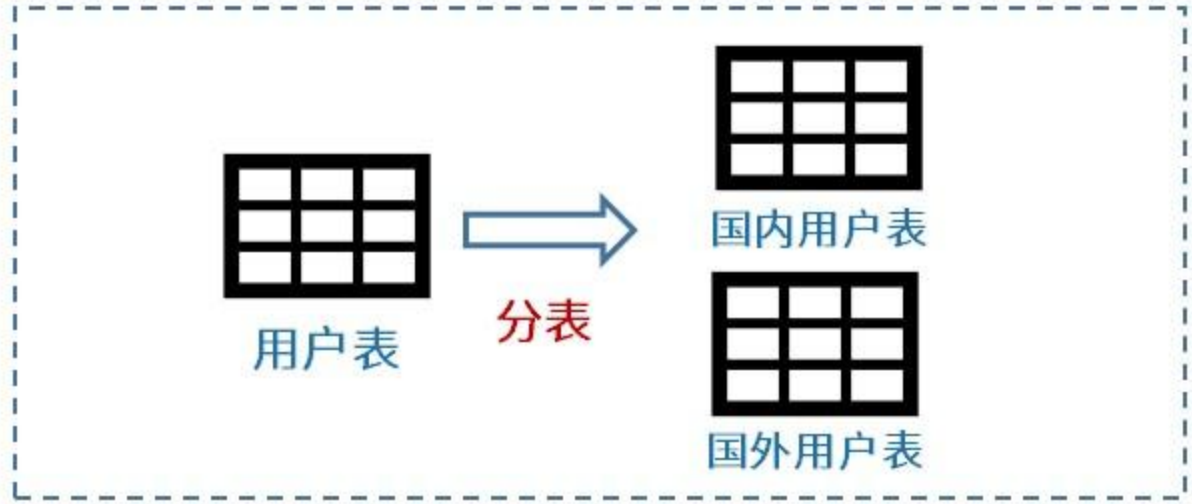
触发器实现：在应用程序中，只对订单表进行操作。但写触发器，当订单表发生变化时，把当日订单内容同步更新到当天货物销售、存货情况表中。

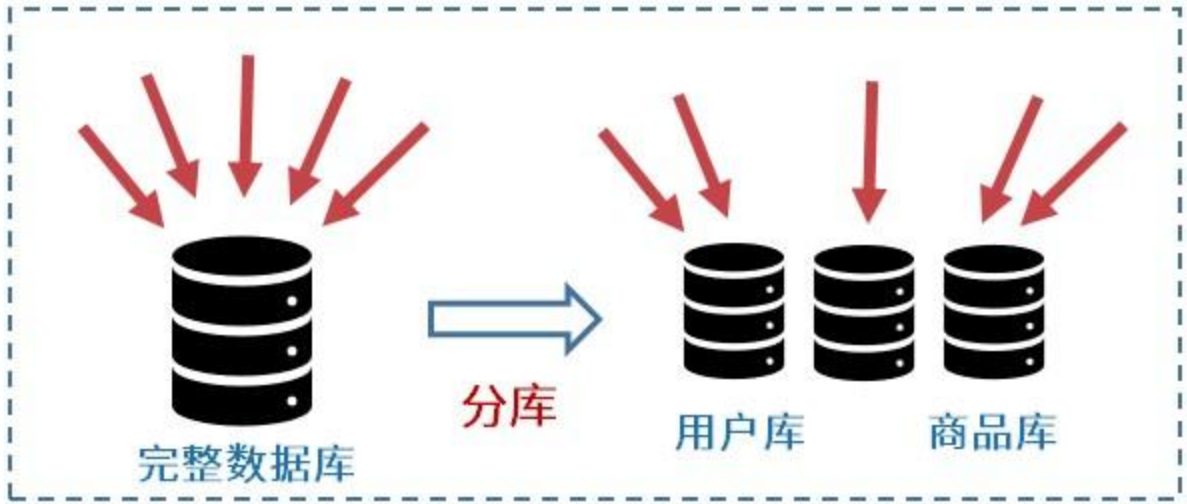
物化视图实现：建立“当天货物销售、存货情况”的物化视图，物化视图会把相应的数据物理存储起来，而且在订单表发生变化时，会自动更新。

数据库 - 分区分表分库

希赛







数据库 - 分区分表分库

希赛

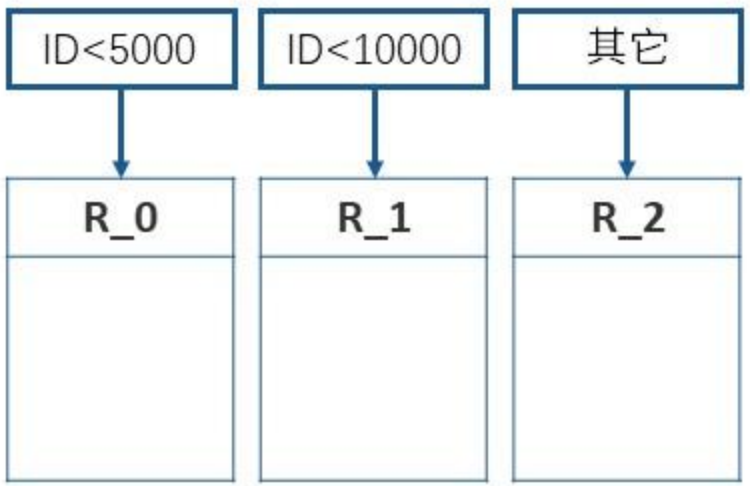
	分区	分表
共性	1、都针对数据表 2、都使用了分布式存储 3、都提升了查询效率 4、都低数据库的频繁I/O压力值	
差异	逻辑上还是一张表	逻辑上已是多张表

数据库 - 分区分表分库

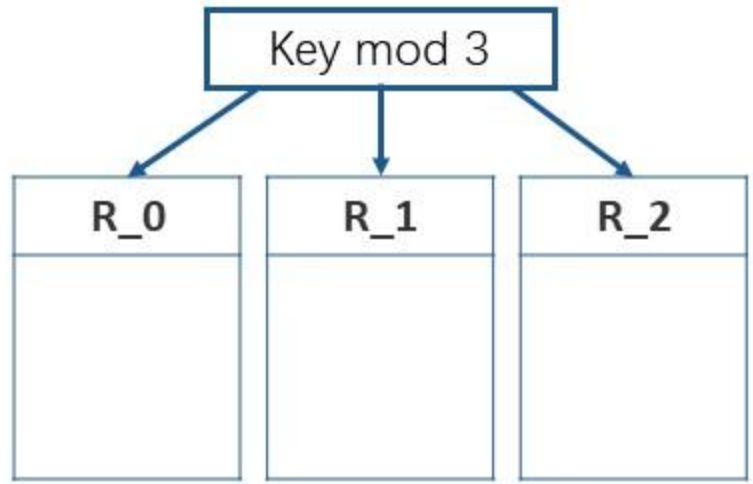
希赛

分区的常见方式

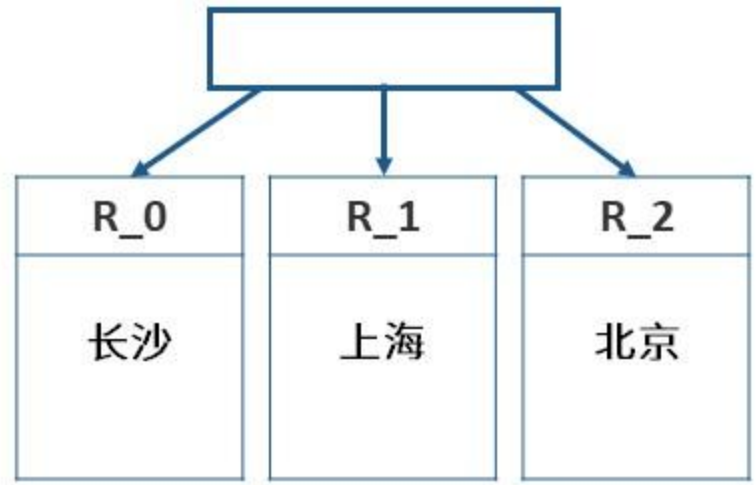
范围分区



哈希分区



列表分区



数据库 - 分区分表分库

希赛

分区策略	分区方式	说明
范围分区【RANGE】	按数据范围值来做分区	例：按用户编号分区，0-999999映射到分区A；1000000-1999999映射到分区B。
散列分区【HASH】	通过对key进行hash运算分区	例：可以把数据分配到不同分区，这类似于取余操作，余数相同的，放在一个分区上。
列表分区【LIST】	根据某字段的某个具体值进行分区	**例：长沙用户分成一个区，北京用户一个区**

数据库 - 分区分表分库

希赛

分区的优点

1、相对于单个文件系统或是硬盘，分区可以存储更多的数据。

2、数据管理比较方便，比如要清理或废弃某年的数据，就可以直接删除该日期的分区数据即可。

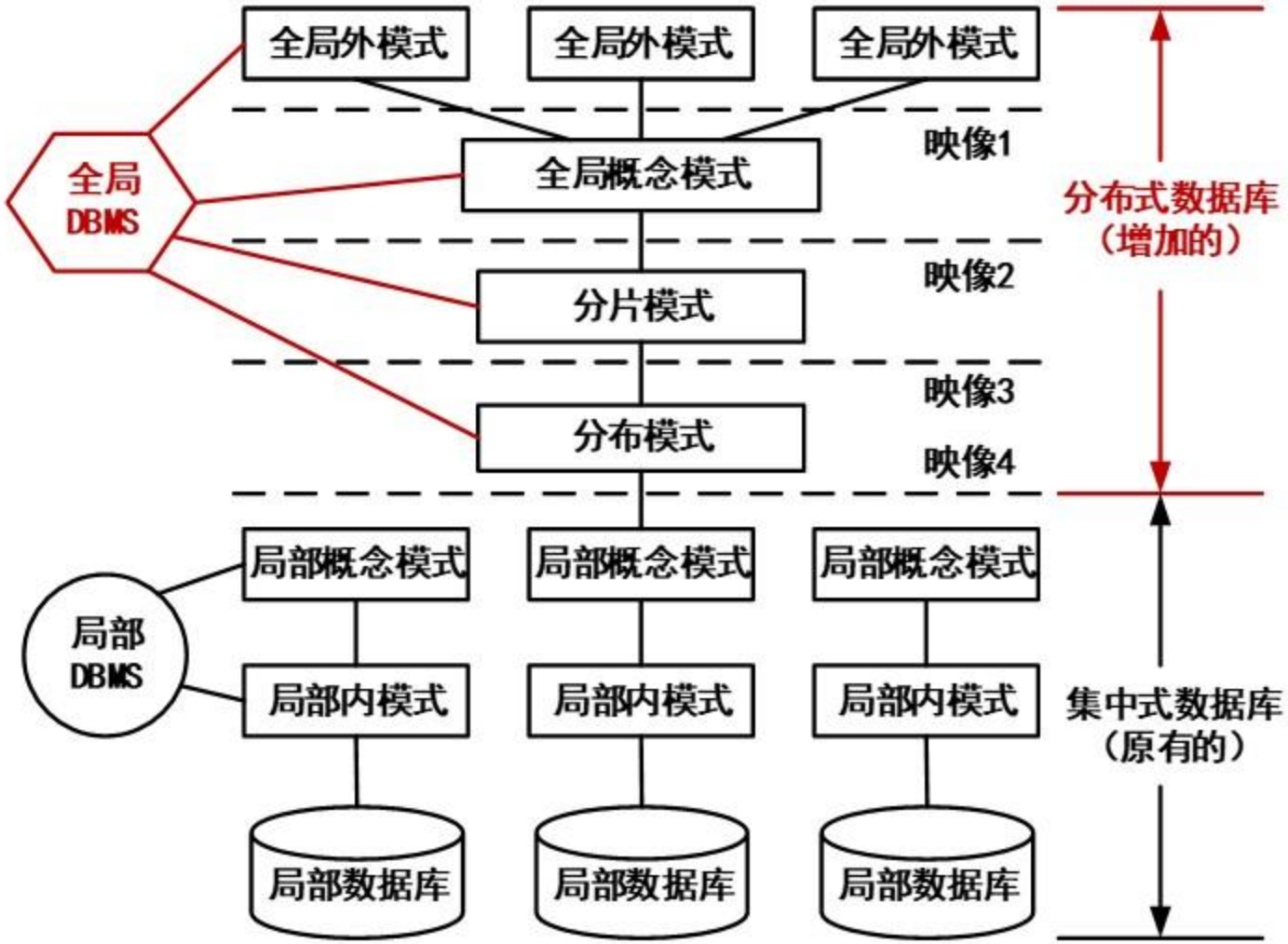
3、精准定位分区查询数据，不需要全表扫描查询，大大提高数据检索效率。

4、可跨多个分区磁盘查询，来提高查询的吞吐量。

5、在涉及聚合函数查询时，可以很容易进行数据的合并。

分布式数据库系统

希赛


全局外模式
全局外模式
全局外模式
全局
DBMS
映像1
全局概念模式
分布式数据库
（增加的）
映像2
分片模式
映像3
分布模式
映像4
局部概念模式
局部概念模式
局部概念模式
局部
DBMS
局部内模式
局部内模式
局部内模式
集中式数据库
（原有的）
局部数据库
局部数据库
局部数据库

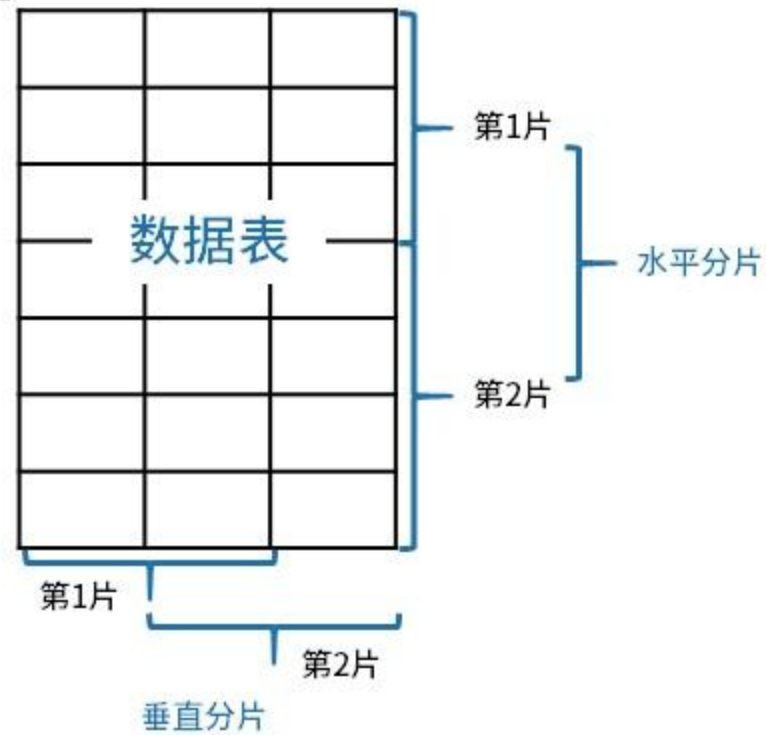
分布式数据库系统

希赛

➢ 分布透明性

- 分片透明性：分不分片，用户感受不到
- 位置透明性：数据存放在哪里，用户不用管
- 局部数据模型透明性（逻辑透明）：用户不用关系局部数据模型

- 水平分片：按记录分
- 垂直分片：按字段分
- 混合分片



➢ 分布式数据库管理系统–组成

- LDBMS
- GDBMS
- 全局数据字典
- 通信管理（CM）

➢ 分布式数据库管理系统–结构

- 全局控制集中的DDBMS
- 全局控制分散的DDBMS
- 全局控制部分分散的DDBMS

NoSQL

希赛

✓ NoSQL（Not-only SQL）：不仅仅只是SQL，泛指非关系型的数据库。

对比维度	关系数据库	NoSQL
应用领域	面向通用领域	特定应用领域
数据容量	有限数据	海量数据
数据类型	结构化数据【二维表】	非结构化数据
并发支持	支持并发、但性能低	高并发
事务支持	高事务性	弱事务性
扩展方式	向上扩展	向外扩展

NoSQL

希赛

分类	典型应用场景	数据模型	优点	缺点	举例
键值(key-value)	内容缓存，主要用于处理大量数据的高访问负载，也用于一些日志系统等等。	Key指向Value的键值对，通常用hash table来实现	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	Redis, Tokyo Cabinet/Tyrant, Voldemort, Oracle BDB
列存储数据库	分布式的文件系统	以列簇式存储，将同一列数据存在一起	查找速度快，可扩展性强，更容易进行分布式扩展	功能相对局限	HBase, Cassandra, Riak
文档型数据库	Web应用（与Key-Value类似，Value是结构化的，不同的是数据库能够了解Value的内容）	Key-Value对应的键值对，Value为结构化数据	数据结构要求不严格，表结构可变，不需要像关系型数据库一样需要预先定义表结构	查询性能不高，而且缺乏统一的查询语法。	CouchDB, MongoDb
图形数据库(Graph)	社交网络，推荐系统等。专注于构建关系图谱	图结构	利用图结构相关算法。比如最短路径寻址，N度关系查找等	很多时候需要对整个图做计算才能得出需要的信息，而且这种结构不太好做分布式的集群方案。	Neo4J, InfoGrid, Infinite Graph

案例分析练习题

希赛

某软件企业开发了一套新闻社交类软件，提供常见的新闻发布、用户关注、用户推荐、新闻点评、新闻推荐、热点新闻等功能，项目采用MySQL数据库来存储业务数据。系统上线后，随着用户数量的增加，数据库服务器的压力不断加大。为此，该企业设立了专门的工作组来解决此问题。

张工提出对MySQL数据库进行扩展，采用读写分离，主从复制的策略，好处是程序改动比较小，可以较快完成，后续也可以扩展到MySQL集群，其方案如图4-1所示。李工认为该系统的诸多功能，并不需要采用关系数据库，甚至关系数据库限制了功能的实现，应该采用NoSQL数据库来替代MySQL，重新构造系统的数据层。而刘工认为张工的方案过于保守，对该系统的某些功能，如关注列表、推荐列表、热搜榜单等实现困难，且性能提升不大；而李工的方案又太激进，工作量太大，短期无法完成，应尽量综合二者的优点，采用Key-Value数据库+MySQL数据库的混合方案。

经过组内多次讨论，该企业最终决定采用刘工提出的方案。

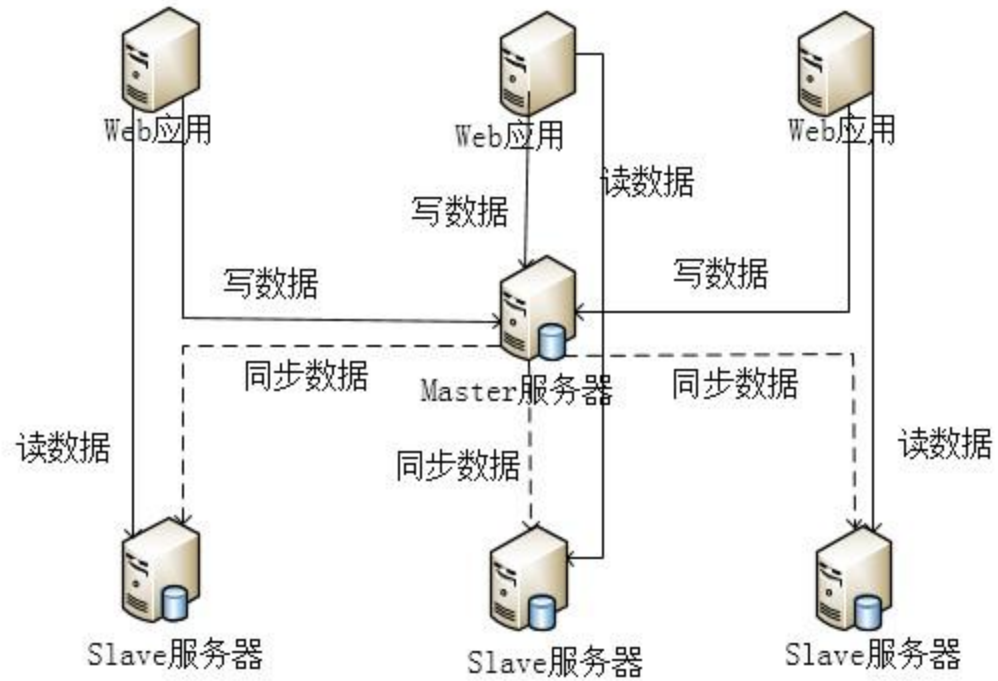


图4-1 张工方案示意图

案例分析练习题

希赛

问题1（8分）

张工方案中采用了读写分离，主从复制策略。其中，读写分离设置物理上不同的主/从服务器，让主服务器负责数据的（a）操作，从服务器负责数据的（b）操作，从而有效减少数据并发操作的（c）,但却带来了（d）。因此，需要采用主从复制策略保持数据的（e）。

MySQL数据库中，主从复制是通过binary log来实现主从服务器的数据同步，MySQL数据库支持的三种复制类型分别是 （f） 、 （g） 、（h）。

请将答案填入（a）～（h）处的空白，完成上述描述。

案例分析练习题

希赛

问题2（8分）

李工方案中给出了关系数据库与NoSQL数据的比较，如表1所示，以此来说明该新闻社交类软件更适合采用NoSQL数据库。请完成表1中的（a）～（d）处空白。

表1关系数据库与NoSQL数据库特征比较

特征	关系数据库	NoSQL数据库
数据一致性	实时一致性	(a)
数据类型	结构化数据	(b)
事务	高事务性	(c)
水平扩展	弱	强
数据容量	有限数据	(d)

问题3（9分）

刘工提出的方案采用了Key-Value数据库+MySQL数据库的混合方案，是根据数据的读写特点将数据分别部署到不同的数据库中。但是由于部分数据可能同时存在于两个数据库中，因此存在数据同步问题。请用200字以内的文字简要说明解决该数据同步问题的三种方法。

案例分析练习题- 参考答案

希赛

问题1

(a)写　　(b)读　　(c)锁争用　　(d)数据冗余　　(e)一致性

(f)(g)(h) 基于SQL语句的复制（statement-basedreplication, SBR），基于行的复制（row-based replication, RBR），混合模式复制（mixed-based replication, MBR）

问题2

(a)最终一致性　　(b)非结构化数据　　(c)柔性事务性　　(d)海量数据

问题3

1、实时同步方案，先查缓存，查不到再从DB查询，并保存到缓存；更新缓存时先更新数据库，再将缓存设置过程期更新缓存。

2、异步队列方式同步，可采用消息中间件处理。

3、通过数据库插件完成数据同步。

4、利用触发器进行缓存同步。

联邦数据库系统

希赛

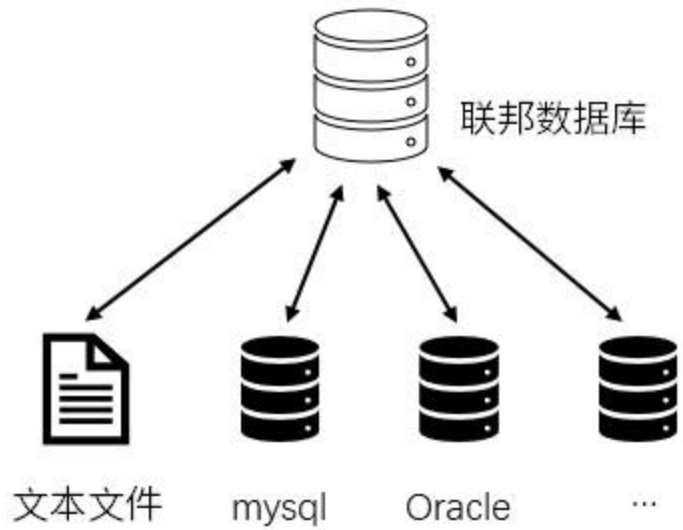
✓ 联邦数据库系统（FDBS）是一个彼此协作却又相互独立的成员数据库（CDBS）的集合，它将成员数据库系统按不同程度进行集成，对该系统整体提供控制和协同操作的软件叫做联邦数据库管理系统（FDBMS）

➢ 联邦数据库特征

- 分布性
- 异构性
- 自治性
- 透明性

➢ 联邦数据库分类

- 紧耦合
- 松耦合



数据库性能优化

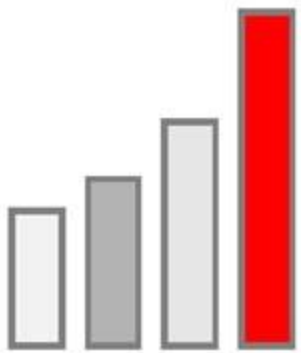
希赛

- 集中式数据库优化
  - 硬件系统：CPU，内存，I/O（硬盘，阵列），网络
  - 系统软件：参数，如进程优先级，CPU使用权，内存使用
  - 数据库设计
    - 分区、分表、分库
    - 物化视图
    - 索引：常查询-建索引，常修改-避免索引
    - SQL优化：以不相干子查询替代相干子查询
只检索需要的列
用带IN的条件子句等价替换OR子句
经常提交COMMIT，以尽早释放锁
尽可能减少多表查询
  - 应用软件：数据库连接池
- 分布式数据库优化
  - 通信代价：全局查询树的变换
多副本策略
查询树的分解
半连接与直接连接

大数据

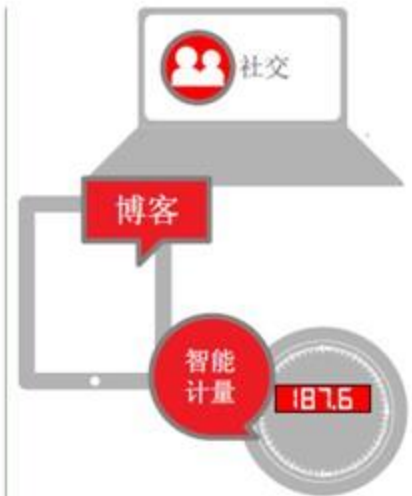
希赛

4V?



数据量
Volume

速度
Velocity



多样性
Variety

值
Value

大数据

希赛

比较维度	传统数据	大数据
数据量	GB或TB级	PB级或以上
数据分析需求	现有数据的分析与检测	深度分析（关联分析、回归分析）
硬件平台	高端服务器	集群平台

- **大数据处理系统应该具有的重要特征**
  - 高度可扩展性
  - 高性能
  - 高度容错
  - 支持异构环境
  - 较短的分析延迟
  - 易用且开放的接口
  - 较低成本
  - 向下兼容性