11 章 数据仓库

1、定义

一种面向主题的、集成的、不可更新的、随时间不断变化的数据集合，用于支持企业或组织的决策分析处理.本质上和数据库一样，是计算机内有组织、可共享的数据集合.

2、特点

- 面向主题 ：数据按照主题来建模、优化、组织.
- 集成性 ：数据来自企业内、外的多个数据源，经过清洗、转化、集成后存入数据仓库.
- 不可更新：数据仓库的数据一般不执行更新操作，对事务数据库的更新，通过数据集成过程反映到数据仓库.
- 时变性：不断有新数据追加到数据仓库；不断有超过存储期限的数据被删除，引起数据仓库中与时间有关的聚集数据的重新计算.。

3、数据组织

- 外部数据源：构建一个数据仓库，必然要有充足的数据来源，从外部为数据仓库系统提供进行分析的“原材料”——数据，这些数据来源称为数据仓库的外部数据源。外部数据源并不局限于传统数据库，可以是非结构化的信息，如文本文件，也可以是网络资源。要保证数据仓库进行的分析能得出正确的、有价值的结论，就必须保证外部数据源的完整、正确。
- ETL：Extract/Transformation/Load

  针对外部数据源进行的数据抽取、转换、装载

  针对加载的数据，进行不同粒度的汇总和综合

  a. 数据抽取：是否有用；是否冗余，是否主题相关

  b. 数据清洗：各类脏数据的清洗：数据空缺、噪声数据、不一致数据；外部

数据缺陷。

c. 数据转换：由于文件格式差异、数据库平台差异和数据结结构的差异导致

d. 不同粒度的数据综合：早期细节级—当前细节级—轻度综合级—高度综合级

4、多维数据与星型建模

- 多维数据模型是数据分析时用户的数据视图，是面向分析的数据模型，用于给分析人员提供多种观察的视角和面向分析的操作
- 星型模式: 包含两种类型的表

a. 事实表：描述实际发生的定量业务数据（如销售量、订单），包含组合关键字和事实数据。

b. 维表：维度：人们总是从不同角度来观察事实，每个观察角度被称为一个维度，维表就是对维度的描述；它包含：简单的主关键字和若干非主属性

c. 事实表和维表的关系：一对多：一个事实表周围环绕多个维表，形成星型结构，代表人们从多个维度观察同一事实数据

5、OLAP 操作

OLAP 操作（联机分析处理）有哪些：（多维分析操作）

切片（slice）：在某一个维上，选定一个维成员的动作 slice for time=“Q2”

切块（dice）：在某一个维上，选定某一区间的动作

下钻（drill - down）：展开细节

上卷（roll-up）：在一定粒度上，取消细节，向上汇总

旋转（pivot）：旋转维度或维度的某个层次

12 章 NoSQL 数据库

1、概念演变和基本理念

最初表示“反 SQL”运动，用新型的非关系数据库取代关系数据库，现在表示关系和非关系型数据库各有优缺点 彼此都无法互相取代。

2、NoSQL 基本理念

- 不一定遵循关系数据库的一些基本要求，例如：关系表结构、标准 SQL 语言、事务的 ACID 特性等。
- 相比传统数据库， NoSQL 数据存储被简化，更灵活。

3、主要数据模型（文档、图、列簇、键值）

类型	特点	部分代表
列存储	**按列存储数据** 适于结构化和半结构化数据，方便数据压缩，对按列查询有非常大的I/O优势	**Hbase** Cassandra Hypertable
key-value 存储	**按key-value对存储** 可以通过key快速查询到对应的value	Tokyo Cabinet / Tyrant Berkeley DB MemcacheDB **Redis**
文档存储	**用类似 json 的格式存储文档内容** 可对某些字段建立索引，实现关系数据库的某些功能	**MongoDB** CouchDB
图存储	**图形关系的最佳存储** 关系数据库在图数据设计和访问等方面很不方便	**Neo4J**、FlockDB、InfoGrid
对象存储	采用类似面向对象语言的语法，来操作数据库，访问数据对象	db4o Versant
xml数据库	存储XML数据，支持XML的内部查询语法，比如XQuery,Xpath	Berkeley DB XML BaseX

4、内存数据库概念和特点

- 概念（MMDB：Main Memory Database）

  内存数据库是将内存作为主存储设备，将数据放在内存中，直接操作的数据库

- 特点

  “主拷贝”或“工作版本”常驻内存，直接的内存访问；与磁盘数据库同步，具有数据恢复机制；并发处理能力；高性能：微妙级的查询响应；高吞吐率和低访问延迟；硬件相关性

- Redis 数据库的概念、特点、数据结构（五种）、适用性

1) Redis 数据库概述：Remote Dictionary Server 的缩写；开源高性能数据库：C 语言开发；内存数据库：是基于内存的，亦可持久化；键值数据库：采用 key-value 数据模型；分布式数据库：支持分布式集群，具有扩展性

2) Redis 数据库特性（记住每个特性名）

持久化：可以将内存中的数据保存到磁盘上，重启时可以再次加载并使用

两种持久化方式：a. RDB：在指定时间间隔内，生成数据集的时间快照；b. AOF：记录服务器执行的所有写操作，并在服务器启动时，重新执行这些命令来还原数据集。

高性能：读写速度大约 10 万次 / 秒；SET 操作每秒 11 万次；GET 操作每秒 8 万次

主从复制：支持 Master 和多个 Slave 的主从关系；Slave 会自动同步 Master 服务器的数据；主服务器宕机，从服务器可替代主服务器为客户端服务；可以把持久化任务配置在从服务器上，减轻主服务器的压力；用途：实现读写分离、数据备份、灾难恢复等

提供多种键值数据类型，适应不同场景下的存储需求：字符串类型：String；散列类型： Hash（ 一个 key，多个 field）；列表类型：list （双向链表）；集合类型：Set （元素不重复）；有序集合类型：Sorted set

扩展性强：支持集群架构；通过数据分片（sharding）实现数据的集群分布；集群没有中心节点或代理节点，线性可扩展；部分节点失效或无法通讯时，仍然可继续处理请求；分片方式：键空间拆分成 16384 个槽位，各节点负责其中部分槽位。

客户端资源：提供多种语言的开源客户端类库:支持 Java，Python，C/C++，C#，PHP，JavaScript，Perl，Object-C，Ruby，Erlang 等语言;提供上述语言的 Redis 接口方法。如：面向 string、list、hash、set、sorted set 等

数据结构的各类操作；提供了灵活的客户端连接。

3) 数据结构：

String，Hash（类似于字典，有 key-field-value），List（存储有序的字符串列表, 重要数据结构之一），Set（集合数据，自动排重），Sorted set（有序且不重复的集合数据））

4) 适用性

请求量大：QPS（每秒请求数）一般至少 2000 以上；数据规模不大时，可以支持 9-10 万 的 QPS

读写频繁：适用于读写频繁的应用情境；充分利用 Redis 内存数据库的快速读写优势

数据结构复杂：可以通过 string、hash、list、set、sorted set 等数据结构，完成一些典型应用场景