**《NoSQL数据库技术》教材课后习题参考答案（部分）**

**第一章习题**

**2、NoSQL数据库产生的主要原因是什么？**

答：主要原因包括：

1. 应对海量用户和低延迟的挑战；
2. 应对模式多样性和灵活性的挑战；
3. 应对高可扩展性和高可用性的挑战；
4. 应对高可扩展性和高可用性的挑战；
5. 应对知识建模的挑战。

**5、试阐述CAP定理的具体含义。**

答：CAP定理中的C指的是一致性（Consistency），即多个数据副本在同一时刻的数据总是一致的；A指的是可用性（Availability），即在集群中一部分节点发生故障后，其他节点仍然能够响应用户请求；P（Partition tolerance）指的是分区容错性，即在出现网络故障后，各个被隔离的网络分区虽然不能通信，但仍然保持可用。

CAP定理是指分布式数据库系统最多只能同时满足一致性、可用性、分区容错性中的两个特性，不可能三者兼顾。

根据CAP定理，一个分布式数据库系统要么满足可用性和分区容错性（AP），要么满足一致性和分区容错性（CP），要么满足可用性和分区容错性（AC），但是不可能同时满足CAP。

**第二章习题**

**1、键值数据库的数据模型是什么？**

答：键值数据库根据键名（Key）来读写键值（Value），其采用的数据模型是关联数组。

**2、键值数据库的键名设计需要遵循什么规范，这样做有什么好处？**

答：键名设计一般应遵循以下几个规则：

（1）键名应该包含有意义的字符 ；

（2）将标识符放入键名，如ID、日期或整数值等，用以唯一确定键名；

（3）键名由多个部分组成，每个部分之间插入分隔符，一般采用“：”；

（4）在不影响语义含义的前提下，尽量把键名设计得简短一些，减少存储空间。

在键值数据库中，键名是指向键值的引用，它与地址的概念相类似，提供了查找键值的方式，是唯一的。规范命名可以看出该键所代表的实际含义，在实际应用中有规律可循。

**6、将2.6节中的数据全部存储到Redis键值数据库中，再根据以下要求写出相应的命令：**

1）查询学号是“S01”的学生姓名。

get Student:S01:Sname

1. 查询学号是“S01”的学生年龄。

get Student:S01:Sage

1. 查询学号是“S01”的学生选修课程号和成绩。

hgetall Reports:S01:Cno:Grade

1. 查询课程号是“C01”的课程名。

get Courses:C01:Cname

1. 查询部门编号是“D01”的部门名。

get Dept:D01:name

**第三章习题**

**2、如何对一对多和多对多的文档关系进行建模？**

答：根据文档与文档之间的逻辑关系，可以使用嵌入式文档或引用标识符对文档结构进行建模。

通过嵌入式文档所构成的数组能够灵活地表示一对多的文档关系。

通过嵌入式文档和文档标识符能够对文档的多种关系进行建模，具体采用何种方式取决于查询文档的效率以及维护文档的方便性，可以灵活设计。

**4、文档数据库是无模式数据库，请说明具体含义。**

模式在数据库中是一个重要的概念。一般地，在数据库中模式指数据的“结构”或“纲要”。

文档数据库不需要预先指定模式结构，属于无模式数据库或去规范化数据库。这个特点使得文档数据库具有更高的灵活性，由应用程序在使用过程中灵活地定义具体的模式，可以随时向集合中添加文档，并定义键值对。

实际上，文档数据库的同一个集合可以存储不同类型的文档，也可以具有不同的结构，没有限定性规范来约束文档的结构。文档数据库的这种无模式结构为应用程序提供了灵活性。

**6、将第二章2.6节的三个基本表中的数据全部存入MongoDB，并完成以下操作：**

1. 查询姓名是“王建平”的文档。

db.students.find({“Sname”:”王建平”}).pretty()

1. 查询年龄大于20岁的学生文档。

db.students.find({“Sage”:{$gt:20}}).pretty()

1. 查询选修了“C01”且及格的学生文档。

db.students.find({Courses.Cno:”C01”,Courses.Grade:{$gte:60}}).pretty()

4) 在姓名上创建一个升序的唯一索引。

Db.students.createIndex({“Sname”:1},{unique:true})

5）查询各门课程的平均分数。

Db.students.aggregate([$unwind:”$Courses”},{$group:{\_id:”$Course.Cno”,avg:{$avg:”$Courses.Grade”}}}]).pretty()

6）查询姓”刘”的同学的文档信息。

db.students.find({“Sname”:/刘/}).pretty()

**第四章习题**

**1、列族数据库的列与关系型数据库的列有什么不同？**

**答：**在列族数据库中，列通常是一个键值对（key-value pair）的集合，其中每个键对应于一个列族（column family），而每个值对应于该列族中的一个特定列。

在关系型数据库中，列是一个数据表中的垂直部分，每一列都有一个特定的数据类型和名称，并且在数据表中有明确定义的位置。

因此，列族数据库的列具有动态性和灵活性，可以轻松地添加、删除和修改列。相反，关系型数据库中的列是静态的，必须在创建表时就定义好，而且难以动态修改。另外，列族数据库通常被用于处理大规模、非结构化的数据，而关系型数据库则更适合处理结构化的数据。

**5、Cassandra采用的数据复制策略有哪些？请说明具体含义。**

**答：**Cassandra设置了三种数据复制策略：

(1)SimpleStrategy复制策略：该策略将多个数据副本存储在同一个机架中，且是连续存储。假设副本数为3，某节点已存储了1个副本，则另外2个副本存储在与该节点相邻的两个节点中。

(2)OldNetworkTopologyStrategy复制策略：该策略需要考虑机架的因素，假设第一个数据副本已经存储在某个节点中，则第二个接下来先找一个与第一个数据副本不在同一个数据中心的节点放置第二个副本；然后再继续找与第二个备份节点位于同一个数据中心，但是不同机架的节点进行备份；接下来所有的备份节点寻找策略就按照SimpleStrategy的备份策略继续寻找。

(3)NetworkTopologyStrategy复制策略：将M个副本放置到其他数据中心，将N-M-1个副本放置在同一数据中心的不同机架中。

**6、假设有一个关系型数据库包含三个基本表，其中Students表保存了学生信息，Courses表保存了课程信息，Reports表保存了选课信息，如下述表格所示。**

(1)创建 StudentMis 键空间，复制策略是“SimpleStrategy”，副本数是 1。

create keyspace StudentsMis with

replication={'class':'SimpleStrategy','replication\_factor':1} and durable\_writes='true';

(2)创建两个表（列族），分别存储学生及其选修课信息、课程信息。

create table students(

Sno text,

Sname text,

Ssex text,

Sage int,

College text,

Reports map<text,int>,

primary key(Sno)

);

create table courses(

Cno text,

Cname text,

Cterm int,

Credits int,

primary key(Cno)

);

(3)将数据插入所创建的两个表中。

Insert into students(Sno,Sname,Ssex,Sage,College,Reports)

values('S01','张利','女',22,'信息学院',{'C01':92,'C03':84});

Insert into students(Sno,Sname,Ssex,Sage,College,Reports)

values('S02','王芳','女',20,'信息学院',{'C01':90,'C02':94,'C03':82});

Insert into students(Sno,Sname,Ssex,Sage,College,Reports)

values('S03','范诚欣','女',19,'计算机学院',{'C02':90});

Insert into students(Sno,Sname,Ssex,Sage,College,Reports)

values('S04','李铭','男',21,'计算机学院',{'C03':75});

insert into students(Sno,Sname,Ssex,Sage,College) values('S05','黄佳宇','

男',21,'理学院');

insert into students(Sno,Sname,Ssex,Sage,College) values('S06','仇星星','

男',22,'理学院');

insert into course(Cno,Cname,Cterm,Credits) values('C01','高等数学',1,4);

insert into course(Cno,Cname,Cterm,Credits) values('C02','英语',2,4);

insert into course(Cno,Cname,Cterm,Credits) values('C03','离散数学',3,3);

insert into course(Cno,Cname,Cterm,Credits) values('C04','数据库技术',4,3);

insert into course(Cno,Cname,Cterm,Credits) values('C05','JAVA',1,2);

insert into course(Cno,Cname,Cterm,Credits) values('C06','操作系统',4,3);

insert into course(Cno,Cname,Cterm,Credits) values('C07','编译原理',4,3);

(4)在学生姓名列上创建一个索引，升序排列。

create index SnameIndex on students(Sname);

(5)查询姓“张”的同学的选修课信息。

create custom index sname\_mh\_idx ON students (Sname) USING

'org.apache.cassandra.index.sasi.SASIIndex' WITH OPTIONS = {'mode': 'CONTAINS',

'analyzer\_class': 'org.apache.cassandra.index.sasi.analyzer.StandardAnalyzer', 'case\_sensitive':

'false'};

(6)查询年龄超过20岁的学生信息及其选修课信息。

select \* from students where Sage>20 allow filtering;