图数据库

|  |
| --- |
| 划重点：  重点特性：高度**关联**的数据-复杂而动态的**联系**  主要领域：社交网络，主数据管理，地理空间，推荐系统。  图数据库是处理复杂的，**半结构化**的，紧密**关联**的数据的最好的技术。 |

问题：

|  |
| --- |
| 1. 什么是图论？ 2. 图论的经典模型是什么？（七桥问题） 3. 图不需要schema?那它用的是什么管理机制？ 4. Schema具体体现在哪些方面？ 5. 什么叫聚合存储？ |

《图数据库》笔记：

|  |
| --- |
| 第一二章：   1. 图领域主要划分为：图数据库和图计算引擎。 2. 主流的图模型主要是**属性图**（最主流），资源描述框架（RDF）三元图和超图。 3. 图数据库需要注意的两大特性：底层存储和处理引擎。    1. 图数据库运用的底层存储可能是**原生图存储**，或者是将图数据库序列化，保存到其他通用的数据存储中。    2. 使用**原生图处理**来代表使用**免索引链接**的图数据库。 4. 市场上有的一些图数据库：      1. 图数据库的明显优点：    1. **性能**提升，特别在全局查询中。因为哪怕在数据集增大的情况下，查询也总是和图的一部分相关，因此，每个查询的执行时间只和满足查询条件的那部分遍历的图的大小而不是整个图的大小成正比。    2. **灵活性**，可扩展性高。不必要一开始就把整个系统的结构设计出来，可以在开发的过程中低成本的增删。    3. **敏捷性**。图数据库不需要**schema**,所以它缺少以schema为导向的数据管理机制。他的管理通常作用于编程方式，利用测试来驱动数据模型和查询，以及依靠图来断言业务规则。试用与敏捷开发。 2. 只有图数据库在处理联系上鹤立鸡群    1. 关系型数据库缺少联系。（表连接，稀疏行，非空检查逻辑，复杂关联层数越多时空复杂度激增，需要处理schema…etc）    2. NoSQL缺少联系（这里主要指除了图数据库外的文件系统，列族，键值对）聚合存储模型中大量添加外键（内嵌映射结构），无反向指针。 |
| 第三章：使用图进行数据建模   1. 属性图，Cypher基础用法（start, match, return, where, create unique…etc） 2. 对比关系建模与图建模。    1. 关系建模（草图，er图，映射成表）但需求总是变化的，所以实际生活中常常使用的是**反规范化技术**（我个人觉得这部分的阐述是本书的精华之一）。总体的过程是“设计-规范化-反规范化”。   反规范化中会为了获得查询性能，人为制造重复数据。会造成大量**冗余数据**和**复杂结构**，也就是说，在实现关系型解决方案的过程中，为了适应关系模型，我们对白板上的模型强加了很多变化，这些变化使得概念模型和数据真实的物理布局之间产生了鸿沟，这种概念上的失调，使得将业务需求的变化转化为底层的稳固的关系结构变得非常复杂和繁琐。  有人认为这个过程虽然花时间不短，但建造一个高性能的关系模型所花费的经历相对开发整个项目的工作来说只是较小的一部分。但，实际生活中，需求的大量变更，使得**迁移**花销巨大和风险极高。  而图建模就去掉了使用复杂数据管理代码来规范化和反规范化数据的这一步骤。   * 1. 图的建模：高保真，易理解，对于跨域得心应手。但常见一个陷阱是错误的将本应是节点的数据用联系+联系的属性表示出来，反而丢失了节点本身的内容。简单的说，在建模时注意区别实体和联系，不要将实体误认为联系敷衍过去。一般来说，不要把实体建模成联系。应该用联系来传达实体之间是如何联系的，以及这些联系的质量。   2. 同样值得注意的一点是：不要小看图的天然可扩展性。图数据库的建模方式（一个个建立节点，建立联系）在关系型数据库看来可能是往db里猛灌大量详细数据，似乎是很粗糙的。但在图里，这样做是很自然的。 |
| 第四章：构建基于图数据库的应用 |
|  |
|  |
| 第七章：使用图论预分析：   1. 深度优先搜索（带有决策的）和广度优先搜索（Dijkstra算法，A\*算法） 2. 图论和预测建模（三元闭包，结构平衡的三元闭包，局部桥） |