

MYSQL_STUDY

零.数据库基础

- a. 数据库，保存有组织的数据的容器（通常是一个文件或者一组文件）
- b. DBMS（数据库管理系统），MySQL也是一种DBMS，即**MySQL是一种数据库软件**
- c. 选择数据库可以做到数据的可持久化

0.0 RDBMS与非RDBMS

0.1 关系型数据库

- **实质：**将复杂的数据结构归结于简单的二元关系，即二维表格的形式。

Tno	Tname	Tsex	Department	
T001	石云丹	女	计算机系	*****
T002	罗莉	女	计算机系	*****
T003	王国强	男	计算机系	*****
T004	吴栋	男	计算机系	*****
T005	高鸿轩	男	数学系	*****
T006	张怀良	男	数学系	*****
T007	刘晓伟	男	数学系	*****
T108	马莉莲	女	物电学院	*****
*****	*****	*****	*****	*****

- 现实生活中的各种实体以及实体之间的各种关系均用关系模型来表示。**关系型数据库**，就是建立在关系模型基础上的数据库。
- **优势：**
- 复杂查询：可以使用SQL语句在一个表以及多个表之间做非常复杂的数据查询
- 事务支持：使得对于安全性能很高的数据访问要求得以实现

0.2 非关系型数据库

- 关系型数据库的功能阉割版本，基于 键值对 存储数据，不需要经过SQL层的解析，性能非常高
- 有哪些**非关系型的数据库**：
- **键值型数据库**：通过Key-Value键值对的方式来存储数据，Redis是最流行的键值型数据库；
- **文档型数据库**：存放文档的数据库，JSON、XML；
- **搜索引擎数据库**：核心原理，倒排索引，Solr，Elasticsearch；
- 列式数据库；
- 图形数据库；

0.1 关系型数据库设计规则

0.1.1 表、记录、字段

- 实体联系模型：实体集、属性、联系集。

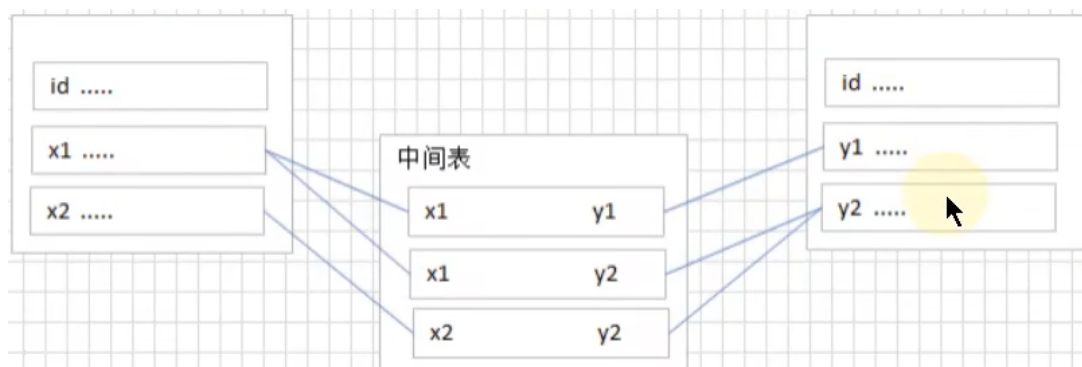
字段	列					属性
	学号	姓名	年龄	性别	专业	
记录	161228001	张三	20	男	JavaEE	实体、对象
	161228002	李四	19	女	H5	
	161228003	王五	21	男	Android	
	161228004	赵六	20	女	PHP	
	161228005	钱七	23	男	JavaEE	
	161228006	孙八	22	男	Android	
行						

- 1 数据库中的一个表 -----Java或者C++中的一个类
- 2
- 3 表中的一条数据-----类中的一个对象
- 4
- 5 表中的一个列-----类中的一个字段、属性

0.1.2 表的关联关系

一对一关系（一个表拆成两个表）、一对多关系（客户相对订单）、多对多关系、自我引用

多对多关系：两个一对多关系结合就成了一个多对多关系，利用中间的那个**联结表**；



比如：学生信息表-课程表-选课信息表

自我引用：



员工编号	姓名	部门编号	主管编号
101	吴小亮	30	NULL
103	刘京生	30	101
104	李向明	30	103
105	张哲夫	30	103
210	黄威	45	101
231	高大山	45	210

四. InnoDB记录存储结构

4.1 InnoDB页简介

InnoDB是一个将表中的数据存储在磁盘上的存储引擎。处理数据的过程是在内存中进行的，所以说我们需要先将数据加载到内存当中，这就涉及到了一个读取速率的问题。

如果我们存取数据的时候，是一条一条记录的存取，就会大大的拖慢读写的速度，所以说InnoDB采取的方式和操作系统处理方式差不多，将数据分成若干页，**以页作为磁盘和内存交互的基本单位。InnoDB中页的大小一般是16KB。**

4.2 InnoDB行格式

五. 索引和不索引

InnoDB存储引擎的索引是一棵B+树，完整的用户记录都存储在B+树的第0层的叶子节点，这些叶子节点，全部都是由数据所构成的页表所构成。（我们针对大量的数据页所制作出来的简易目录，到后面数据多的时候就形成了B+树，这个目录被我们称为索引）。所以说索引并不是像我们想象的那么简单。

InnoDB和MYISAM会自动为主键或者带有UNIOM属性的列建立索引。如果我们想为其他的列建立索引，就需要我们显式的说明。还有一点需要注意的是，虽然索引能够大大的节省我们的检索时间，**但是每建立一个索引都会建立一棵B+树，而且每增、删、改一条记录都要维护各个记录、数据页的排序关系，这是很费性能和存储空间的。**

InnoDB的索引分为两种：

- **聚簇索引**：（聚簇意思是所有的数据都堆叠在了一块，这一点和二级索引对比的时候可以很清楚的~~感觉到~~），以主键的大小作为页和记录的排序规则，在叶子节点处存储的记录包含了表中所有的列；
- **二级索引**：以索引列的大小作为页和记录的排序规则，在叶子节点处存储的记录内容是索引列+主键；

InnoDB存储引擎的B+树根节点自创建之日起就不再移动。

在二级索引的B+树内节点中，目录项记录由索引列的值、主键值和页号组成。

5.1 索引的创建

我们可以在创建表的时候，指定需要建立索引的单个列或者建立联合索引的多个列：

```
1 CREATE TABLE 表名 (  
2     各个列的信息，  
3     (KEY|INDEX) 索引名 (需要被索引的单个列和多个列)  
4 )
```

其中KEY和INDEX是同义词，我们随意的选择一个即可。

添加索引：

```
1 alter table 表名 ADD (INDEX|KEY) 索引名 (需要被索引的单个列或者多个列)；
```

删除索引：

```
1 alter table 表名 DROP (INDEX|KEY) 索引名；
```

命名规范：

一般我们对索引命名的时候，加上前缀 idx_。