



May it Be来自IparhanGeek00:0003:32

数据库系列剖析

数据库初探

我们平时除了开发和工作当中必然会用到数据库这个东西,数据库本身是一个**仓库的思维**,存储我们需要存起来以后要用的东西和资料,比如我们之前身边存点钱习惯在银行开户然后把我们积攒的钱财存进银行开户人员为我们开户的银行账户里面,这样我们的钱不用每天带在身边,就根据银行卡上面的**数字**就知道我们总共有多少钱,其实说白了咱们开发人员使用的数据库都是**银行存取钱的思维**。读到这里大家可以想象一下之前咱们到银行存钱取钱的过程:

- 1.到银行
- 2.开户人员开户
- 3.发放银行卡
- 4.银行卡里面存钱
- 5.ATM机器取现(需要用钱的时候)



看完上图应该能联想到那些没有用支付宝/微信之前的存钱和取现的情景了吧,然后再回到咱们开发人员每天会遇到的数据库这样也就能从另一个方面理解数据库了目前市场上用的数据库比较多(Mysql,Oracle,MariaDB,Nosql,图形数据库),我会此篇文章里面会把每个数据库的重要核心点剖析出来,因为只要掌握数据库架构设计的核心思路和算法那就我们也能试图设计出来一个数据库更适合自己的那种,下面我们就先把目前流行的几个数据库都剖析出来再来谈设计一个数据库了。



谈到数据库系列那我们首先能想到的就是Mysql了,因为这个数据库我们个人开发项目或者小型公司都会首选Mysql数据库为第一选择,为什么? 这个问题的答案等我们剖析完mysql 就自然而然的能知道答案了.

1

Mysql 历史

首先mysql是一个关系型数据库,是基于二维表的基础上设计出来的设计模式,后来是实体模型出来了也是我们现在关系型数据库的思维,这个思维是有一位华人科学家**陈品山**科学家提出来,由此这个思维在数据库思维里面广发使用。

后来一直到我们现在在使用的关系型数据库, 下面我们再来认识一下Mysql数据库从另一方面认识它。

1

Mysql 介绍

Mysql本是一个关系型数据库,之后是有这么几个优点可以回到我们上边问的问题。

- 1.Mysql是基于关系型数据库思维
- 2.支持多种语言,Java,C,C++,PHP,Python...等等
- 3.上手快,易于安装,使用和研究扩展开发。
- 4.支持多线程,适用于高并发场景足矣.
- 5.提供多种连接方式,JDBC ,ODBC,TCP/IP连接方式
- 6.支持多种存储引擎(Myisam,innnoDB,Memory,Merge,Archive)
- 7.....

上面的提到的7种是我们比较关心的特性,如果再分析还有很多,可以自行深入探讨,这章节我会主要举例几个Mysql重要的特性和实战。

2

Mysql 版本各个版本之间的区别

MySQL Community Server： 开源免费,社区版本

MySQL Enterprise Edition ： 企业版本,收费

MySQL Cluster ： Mysql集群,免费

MySQL Cluster CGE： 收费集群

Mysql自1996年1.0版本出来以来到现在8.0版本有很多新特性增加和减掉,比如：

1.Mysql 4.1版本(之前的版本就不说了):

- 添加UTF8数据格式
- 添加Group By语句

2.Mysql 5.0 版本:

- 提供了管理试图
- 提供存储过程

3.Mysql 5.5版本:

- innoDB作为默认存储引擎
- 提高I/O读取能力
- 提高数据恢复时间

4.Mysql 5.6:

- TIMESTAMP 时间格式可以设置多个TIMESTAMP
- InnoDB全文搜索支持

5.Mysql 5.7:

- 多主一从
- 原生JSON支持
- 性能速度比5.6提高3倍

6.Mysql 8.0:

- 模糊查询速度提升30倍
- ...

3 Mysql 存储引擎

谈完Mysql基本理论知识之后,我们再谈谈数据库比较重要的功能之一的存储引擎,这里面有我们常见的几个存储引擎比如:

- MyIsam
- InnoDB
- Memory
- Merge
- Archive

1. 事物隔离级别

说到存储引擎,说白了主要是负责Mysql的数据格式。

存储引擎还有重要的几个点下面我们来看看:

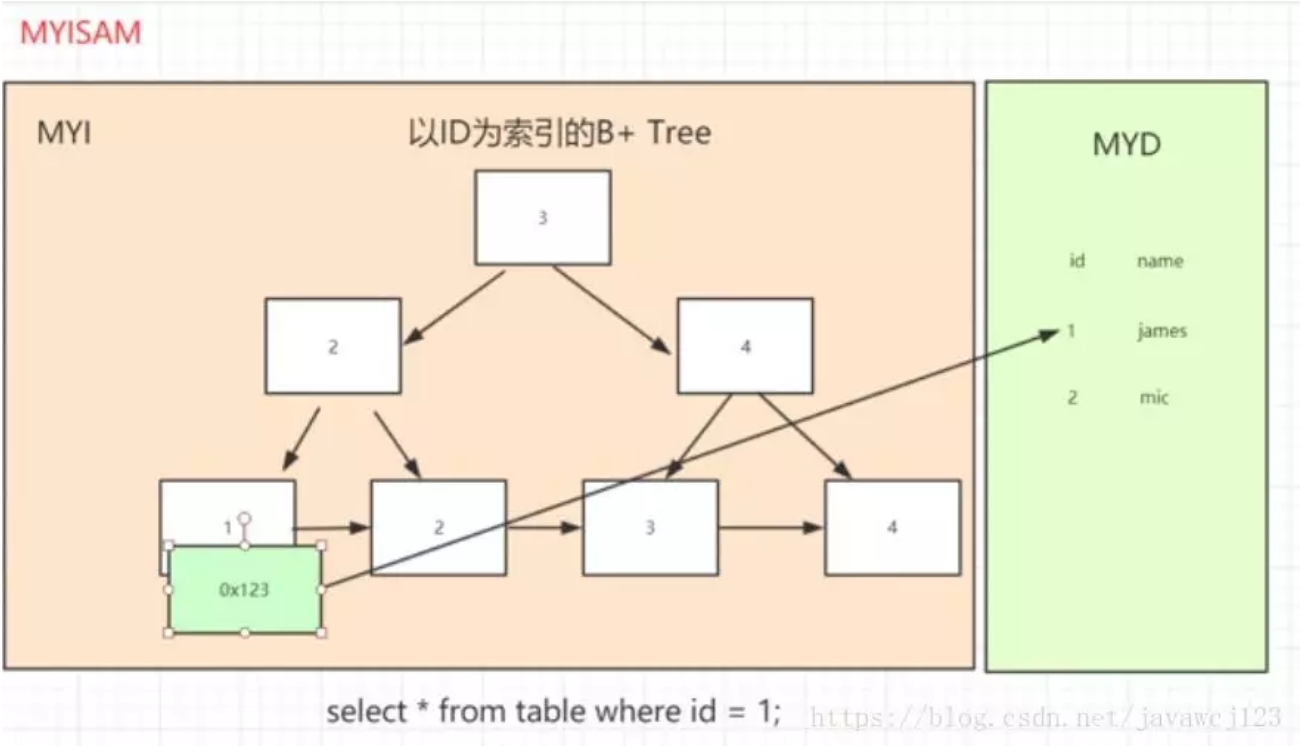
1.数据事物隔离

隔离级别	脏 读 （ Dirty Read)	不可重复读（NonRepeatable Read)	幻读（ Phantom Read)
------	-------------------	---------------------------	-------------------

未提交读（Read uncommitted）	可能	可能	可能
已提交读（Read committed）	不可能	可能	可能
可重复读（Repeatable read）	不可能	不可能	可能
可串行化（Serializable）	不可能	不可能	不可能

说到事物隔离级别就是指当我们多个服务器或者服务同时访问一个数据库的同一条数据的时候想想如果就在这时有一个线程把数据库的这个值Update了那这个时候别的线程会怎样,读到的数据能保证是完整的或者是无损的吗？**A线程正在改数据B线程过来读取了,读取的到的数据很可能是不完整的,这也叫脏读**,如果A线程改完数据并且保存了B线程再来读取这个是时候就不会发生任何错误问题了,这里面重要的知识点隔离级别就能有用了。

2.索引技术



说到索引这里我会重点解释Mysql的索引技术**索引本身也是一张表**,首先索引分有很多种(也可以自定义),这里举例常用的几种：

```
CREATE TABLE table_name[col_name data type][unique|fulltext][index|key][index_name](col_name[length])[asc|desc]
```

- 1.unique|fulltext为可选参数，分别表示唯一索引、全文索引
- 2.index和key为同义词，两者作用相同，用来指定创建索引
- 3.col_name为需要创建索引的字段列，该列必须从数据表中该定义的多个列中选择
- 4.index_name指定索引的名称，为可选参数，如果不指定，默认col_name为索引值

5.length为可选参数，表示索引的长度，只有字符串类型的字段才能指定索引长度

6.asc或desc指定升序或降序的索引值存储

普通索引：没有任何限制

创建索引表

```
CREATE INDEX index_name ON table(column(length))
```

修改普通索引

```
ALTER TABLE table_name ADD INDEX index_name ON (column(length))
```

创建数据表的时候就定义索引

```
CREATE TABLE `table` (  
  `id` int(11) NOT NULL AUTO_INCREMENT ,  
  `title` char(255) CHARACTER NOT NULL ,  
  `content` text CHARACTER NULL ,  
  `time` int(10) NULL DEFAULT NULL ,    PRIMARY KEY (`id`),    INDEX index_name (title(length))  
)
```

删除索引

```
DROP INDEX index_name ON table
```

唯一索引：

创建唯一索引

```
CREATE UNIQUE INDEX indexName ON table(column(length))
```

修改唯一索引表结构

```
ALTER TABLE table_name ADD UNIQUE indexName ON (column(length))
```

创建表的时候就定义唯一索引

```
CREATE TABLE `table` (  
  `id` int(11) NOT NULL AUTO_INCREMENT ,  
  `title` char(255) CHARACTER NOT NULL ,  
  `content` text CHARACTER NULL ,  
  `time` int(10) NULL DEFAULT NULL ,    UNIQUE indexName (title(length))  
)
```

主键索引：主键索引是Auto_crement 就是主键索引,不允许有空

创建索引

```
CREATE TABLE `table` (  
  `id` int(11) NOT NULL AUTO_INCREMENT ,  
  `title` char(255) NOT NULL ,    PRIMARY KEY (`id`)  
)
```

组合索引：指多个字段需要使用索引叫组合索引

创建组合索引：

```
ALTER TABLE `table` ADD INDEX name_city_age (name,city,age);
```

全文索引：

全文索引也叫全文检索,这里面的技术使用的是反向索引,也就是所谓非结构化搜索方式,这种方式有点先Java里面的LinkedList的链表模式,没有循序的查询, 这个技术在Mysql里面也使用了,所以理解这个时候可以参考全文检索技术就比好好理解了(Lucene,Solr,Elasticsearch)。

创建全文索引

```
CREATE TABLE `table` (  
  `id` int(11) NOT NULL AUTO_INCREMENT ,  
  `title` char(255) CHARACTER NOT NULL ,  
  `content` text CHARACTER NULL ,  
  `time` int(10) NULL DEFAULT NULL ,    PRIMARY KEY (`id`),  
  FULLTEXT (content)  
);
```

Mysql 锁技术

说到锁机制,这个在并发请求场景下常用,锁也有几种分类:

行锁,表锁,页面锁

行锁: InnoDB 使用innodb_lock 来实现行锁。MyIsam不支持行锁

```
lock in share mode      如:  select * from user where user_name='test' lock in share mode
```

读锁(排它锁)

```
for update如:  select * from user where user_name='wzzf' for update
```

实例-开启一个读锁:

```
BEGIN;  
  
SELECT * from tyg_consumer_chain_sell_order o  
where o.consumer_chain_order_number=26911523448454  
lock in SHARE MODEchain_order_number=26911523448454  
lock in SHARE MODE
```

表锁: InnoDB: Mysiam InnoDB 都支持

开启一个表锁

```
BEGIN;  
  
SELECT * from tyg_consumer_chain_sell_order o  
where o.consumer_chain_order_id=1 for update
```

END

IparhanGeek

海赛尔



My QR Code



Iparhan 

Haidian, Beijing



Scan the QR code to add me on WeChat

