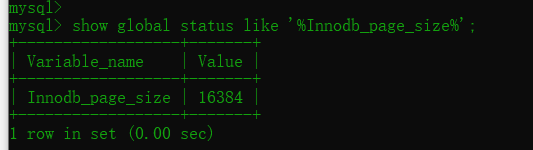
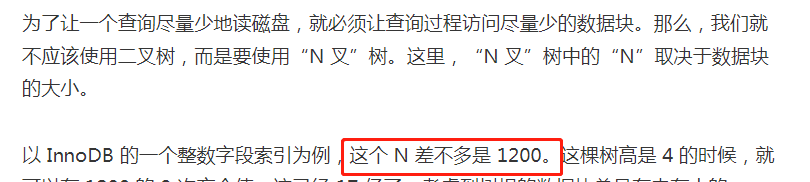
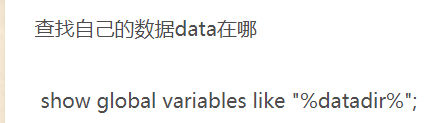


Mysql innodb页数据大小







2019年7月3日09:09:29

查看表索引所占数据块的大小情况

SELECT

sum(stat\_value) pages,

index\_name,

sum(stat\_value) \* @@innodb\_page\_size size

FROM

mysql.innodb\_index\_stats

WHERE

table\_name LIKE 't\_wx'

AND database\_name = 'lijin'

AND stat\_description LIKE 'Number of pages in the index'

GROUP BY

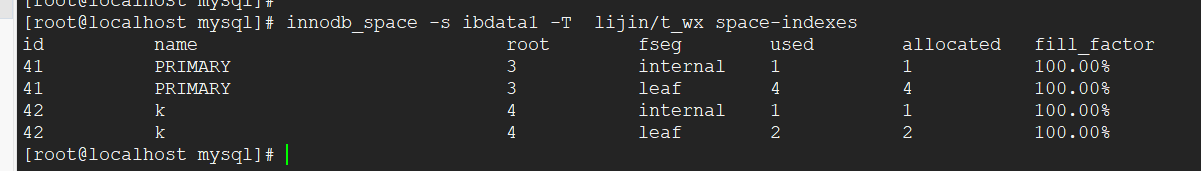
index\_name;

先安装ruby.然后gem install innodb\_ruby。Ruby版本要大于2.0  
innodb\_ruby innodb数据文件分析工具

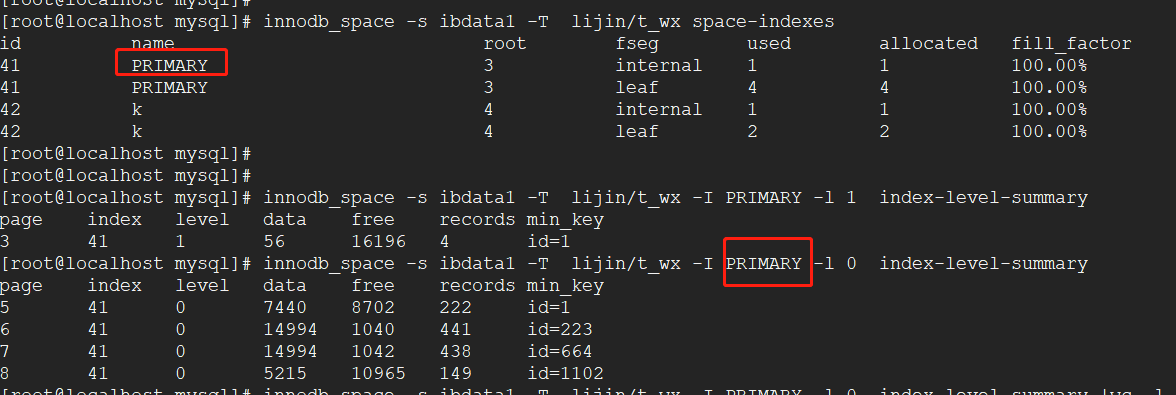
基本使用

innodb\_space -s ibdata1 -T lijin/t\_wx space-indexes

lijin/t\_wx 分别为数据库名和表名



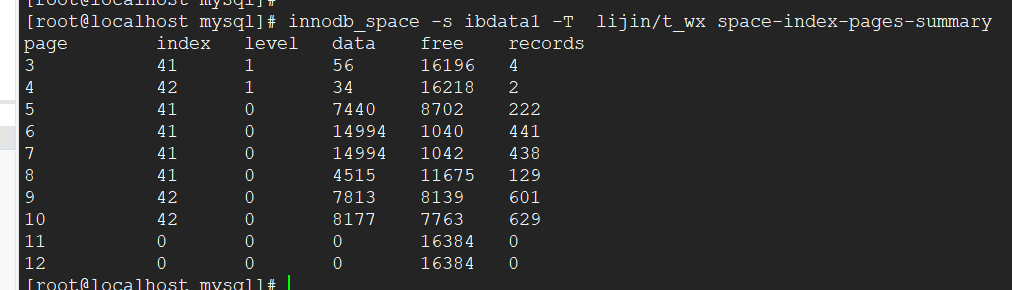
查看各索引情况。每页数据情况。



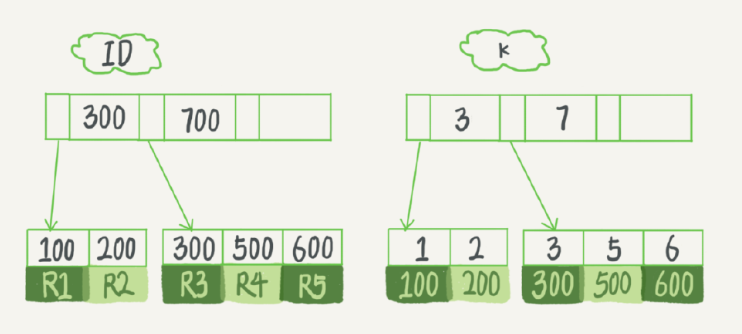
**22.2.1.6 Page Directory**

<https://dev.mysql.com/doc/internals/en/innodb-page-directory.html>

innodb\_space -s ibdata1 -T lijin/t\_wx space-index-pages-summary



流程：



主键索引(primary key)和非主键索引(secondary key)

为什么是B+树？为什么不是B树？

主键索引和非主键索引真的如上图所示吗，有什么证据。

Innodb数据按页page存储，内部结构怎样？

建表：

CREATE TABLE `t\_wx` (

`id` int(11) NOT NULL,

`k` int(11) NOT NULL,

`name` varchar(16) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `k` (`k`)

) ENGINE=InnoDB DEFAULT CHARSET=utf-8;

插入数据：

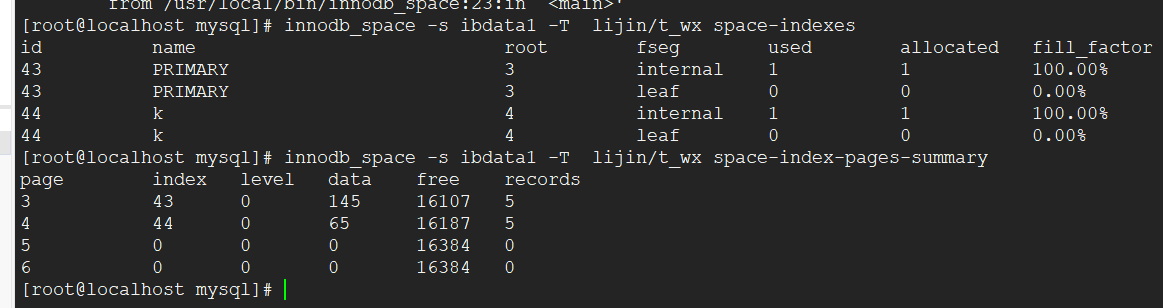
insert into t\_wx(id,k,name)values(1,10,'a');

insert into t\_wx(id,k,name)values(2,11,'a');

insert into t\_wx(id,k,name)values(3,12,'a');

insert into t\_wx(id,k,name)values(4,13,'a');

insert into t\_wx(id,k,name)values(5,13,'a');

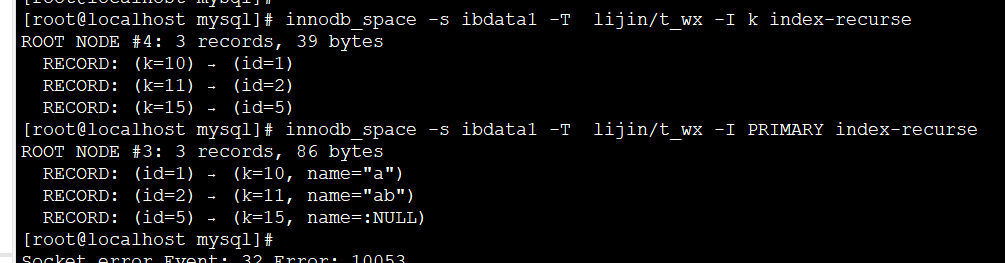


主键索引和非主键索引都只用到了一个数据页。当前的B+树就一个节点，没有子节点。

Page-3 数据大小为145 byte

Page-4 数据大小为 64 byte





数据库页文件内部结构

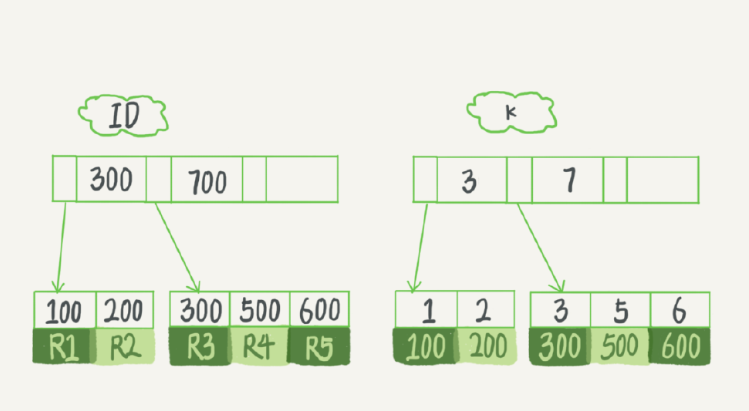
innodb\_space -s ibdata1 -T lijin/t\_wx -p 3 page-dump

优秀项目：

# [Meituan-Dianping](https://github.com/Meituan-Dianping)/[SQLAdvisor](https://github.com/Meituan-Dianping/SQLAdvisor)

输入SQL，输出索引优化建议

<https://github.com/Meituan-Dianping/SQLAdvisor>  
[git@github.com:Meituan-Dianping/SQLAdvisor.git](mailto:git@github.com:Meituan-Dianping/SQLAdvisor.git)



<https://mariadb.com/kb/en/library/building-the-best-index-for-a-given-select/>