数据库第五次上机

本次上机任务

本次的上机任务是熟悉SQL语言的基本操作：

- 索引

本次实验**请使用 MySQL 数据库完成**
数据库管理工具不限

关于作业提交

TASK1 提交

请在**PDF/WORD等任何方便助教阅读查看的文档**中按照各个作业要求提交相关内容，**记得标清题号**。

3. 若为PDF/WORD单文档文件直接提交即可，其他提交压缩包，命名为“学号_姓名_第*次实验”。

4. 提交网址：软件学院云平台**第五次上机**　　（按要求提交）

作业截止时间为**周日24:00之前，提交方式为提交到云平台。**

TASK 1 索引

1、请完成“准备工作”中的数据表建立

2、请完成Q1-Q4中的任务，并说明引入索引带来查询速度变化/不变的原因

提交要求：请在**PDF/WORD等任何方便助教阅读查看的文档中**完成以下提交内容：需要提交相关的**能够显示运行时间/速度的截图，以及相关现象出现的原因解释，请标好题号**。

TASK 1 索引

准备：生成包含3百万-4百万条记录的表（参考以下过程）

首先创建一个存放40万条记录的表（作为后续扩容的基础）：

- CREATE TABLE bdtable (bid int, v bigint, s smallint)
- Create procedure tp1()
- begin
- declare v1 integer;
- set v1=0;
- while v1<400000 do
- insert into bdtable values(v1, round(rand()*100000),round(rand()*10));
- set v1=v1+1;
- end while;
- End;
- Call tp1;

TASK 1 索引

使用以下方法扩容数据，生成我们需要的数百万记录的表：
（这样比直接生成百万记录要快）

- create table bdcopy like bdtable
- insert into bdcopy select * from bdtable
- update bdcopy set v=v+1
- update bdcopy set bid=bid+xxxx（xxx是bdtable表中bid最大编号）
- insert into bdtable select * from bdcopy

最后再复制我们生成好的表，生成三个对比实验用的百万记录表（一共有四个）：

- Create table bdcopy1 like bdtable
- Insert into bdcopy1 select * from bdtable;

再模仿以上过程生成bdcopy2、bdcopy3..

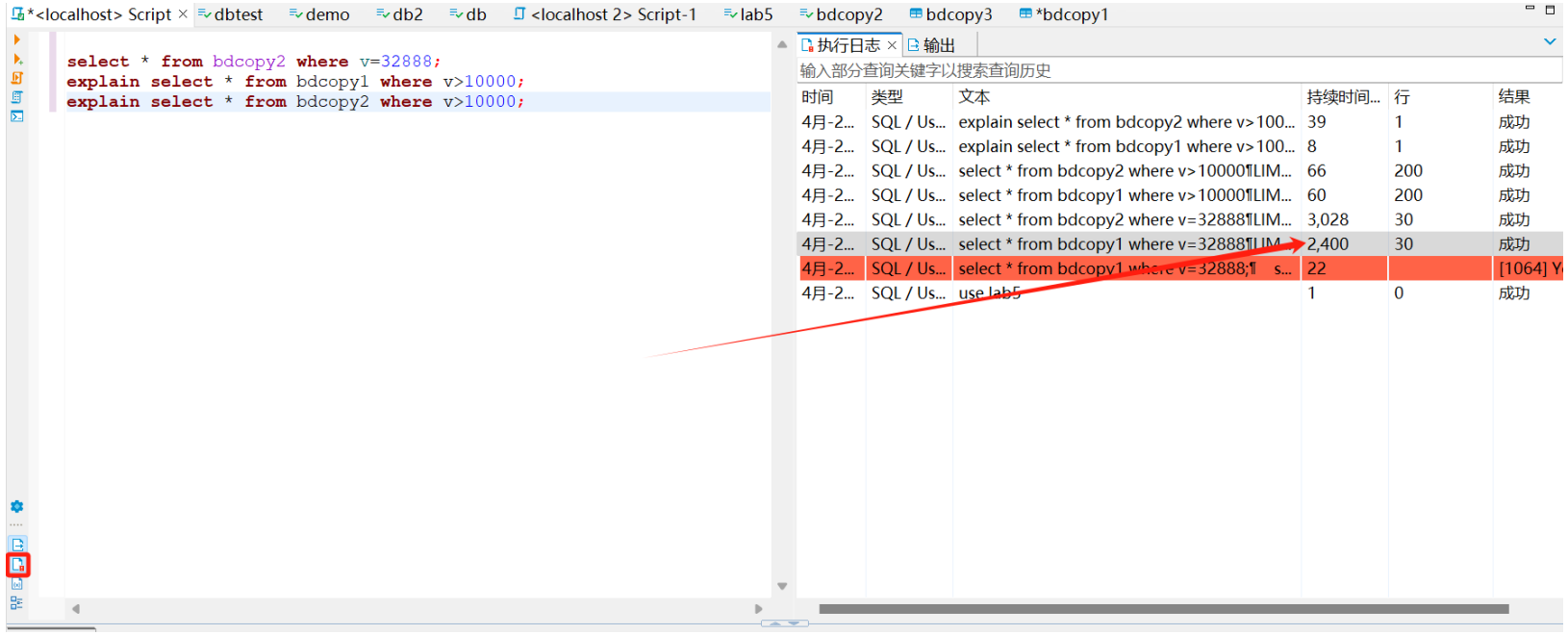
TASK 1 索引

关于查看时间：

Datagrip：直接控制台查看

```
completed in 22 ms
```

Dbeaver：通过点击左下角“显示执行日志”查看查询执行时间



TASK 1 索引

Q1：实验 B+树索引

- 在bdcopy1的 v字段上建立b+树索引， bdcopy2上没有建索引
- 对比两个表的 v 字段上**等值查询、范围查询**的速度差异
- **范围查询可以用如下语句对比实验**：

```
select * from bdcopy1/2 where v>10000
```

**可以用explain命令查看查询计划：

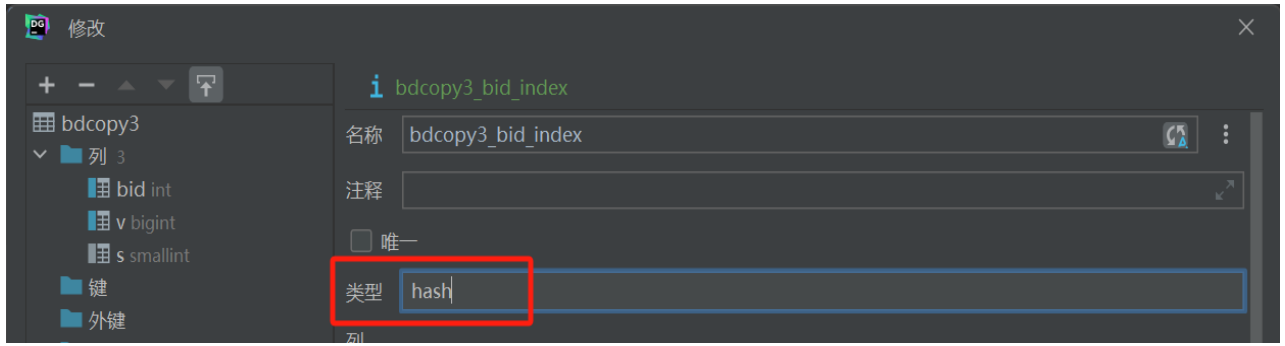
例如：explain select * from bdcopy1 where v>10000

提示：查看explain结果中的“key”词条以获知查询使用的索引

TASK 1 索引

Q2：实验散列索引

- 先通过可视化工具的管理界面在bdcopy3的v字段上建立散列索引。



Datagrip创建索引

dbeaver创建索引

Q2：实验散列索引-续

- 保存后，对比bdcopy1、3二张表在v字段上的等值查询速度
- Select * from bdcopy1/3 where v=3535353(**找一个v字段中实际存在的值**)

提示：关闭和重新打开bdcopy3的设计界面，检查在bdcopy3的v字段上建立的是什么索引

解释：MySql默认的InnoDB引擎不支持hash索引，此处实际建立了B+索引

Q2：实验散列索引-续

- 创建一个**memory引擎**的**带散列索引**的表：
- CREATE TABLE bdcopy4 (bid int, v bigint, s smallint, INDEX USING HASH (v)) engine = memory;

  （也可以创建表后在管理界面加上散列索引）
- 向表中插入数据：insert into bdcopy4 select * from bdtable

（此处只能插入几十万条记录：

因为Memory 存储引擎将表数据存储在计算机的内存中，而不是存储在磁盘上，由于内存的成本相对较高，通常只有有限的内存可供使用）

- 再生成一个表进行对比：create table bdcopy5 like bdtable; insert into bdcopy5 select * from bdcopy4.
- 对比实验查询速度。

Q3：实验聚簇索引

- 在bdcopy2的bid上建立主键**（mysql中的主键索引即聚簇索引）**。
  在bdcopy1的bid 字段上建立b+树索引
- 进行等值查询和范围查询速度对比：
- select * from bdcopy1/2 where bid between 10000 and 20000

Q4：实验联合索引

- 在bdcopy2上建立v、s字段的联合索引
- 分别实验 where v=353535(**找一个v字段上实际存在的数**） and s=3

where v=353535(**找一个v字段上实际存在的数**)

where s=3

的查询速度。

并用explain命令查看实际索引使用情况

例如：Explain select * from bdcopy2 where v>10000 and s=3

Q5：实验函数索引

- 对比分析两种查询
- Explain select * from bdcopy1 where bid/2=30000
- Explain select * from bdcopy1 where bid=30000*2

索引

索引：对数据库表中一列或多列的值进行排序的一种结构。使用索引可快速访问数据库表中的特定信息。

- 聚类索引：此类索引中键值的逻辑顺序决定了表中相应行的物理顺序。该索引可以包含多个列（组合索引）；一张表只能有一个聚类索引。
- 非聚类索引：此类索引中索引的逻辑顺序与磁盘上行的物理存储顺序不同；一张表可以有多个非聚类索引。（数据存储在一个地方，索引存储在另一个地方）

区别与联系：

- 聚类索引=>汉语字典正文（物理顺序）
- 非聚类索引=>部首检字表/拼音检字表/四角号码检字表（非物理顺序）

索引

推荐使用索引的情况：

- 数据量大
- 列值唯一性好
- 较少被修改
- 经常被访问、被join、被子查询
- 经常被order by或group by

不推荐使用索引的情况：

- 数据量小
- 列值唯一性差
- 经常被修改
- 很少被访问

使用哪种索引：

动作描述	使用聚集索引	使用非聚集索引
列经常被分组排序	应	应
返回某范围内的数据	应	不应
一个或极少不同值	不应	不应
小数目的不同值	应	不应
大数目的不同值	不应	应
频繁更新的列	不应	应
外键列	应	应
主键列	应	应
频繁修改索引列	不应	应

索引

MySQL聚簇索引:
在MySQL中，有一列值，专门被设定为聚簇索引，这列值就是主键，通常为数字类型的字段。如果数据表中没有主键，MySQL的解决办法是隐式地将一个唯一的非空的列定义为聚簇。如果这也没有呢？MySQL就自己创建一个聚簇索引，MySQL都会创建一个聚簇索引。

MySQL普通索引（非聚簇索引、二级索引）:
CREATE INDEX indexName ON table_name (column_name)
ALTER table tableName ADD INDEX indexName(columnName)

创建表时直接指定:

```
CREATE TABLE mytable(
 ID INT NOT NULL,
username VARCHAR(16) NOT NULL,
INDEX [indexName] (username [(length)] [ASC | DESC] )
);
```

http://www.aiuxian.com/article/p-onjvxeot-q.html

索引

MySQL多列索引：

多列索引，是指在创建索引时所关联的字段不是一个字段，而是多个字段，虽然可以通过所关联的字段进行查询，**但是只有查询条件中使用了所关联字段中的第一个字段，多列索引才会被使用**。

示例：

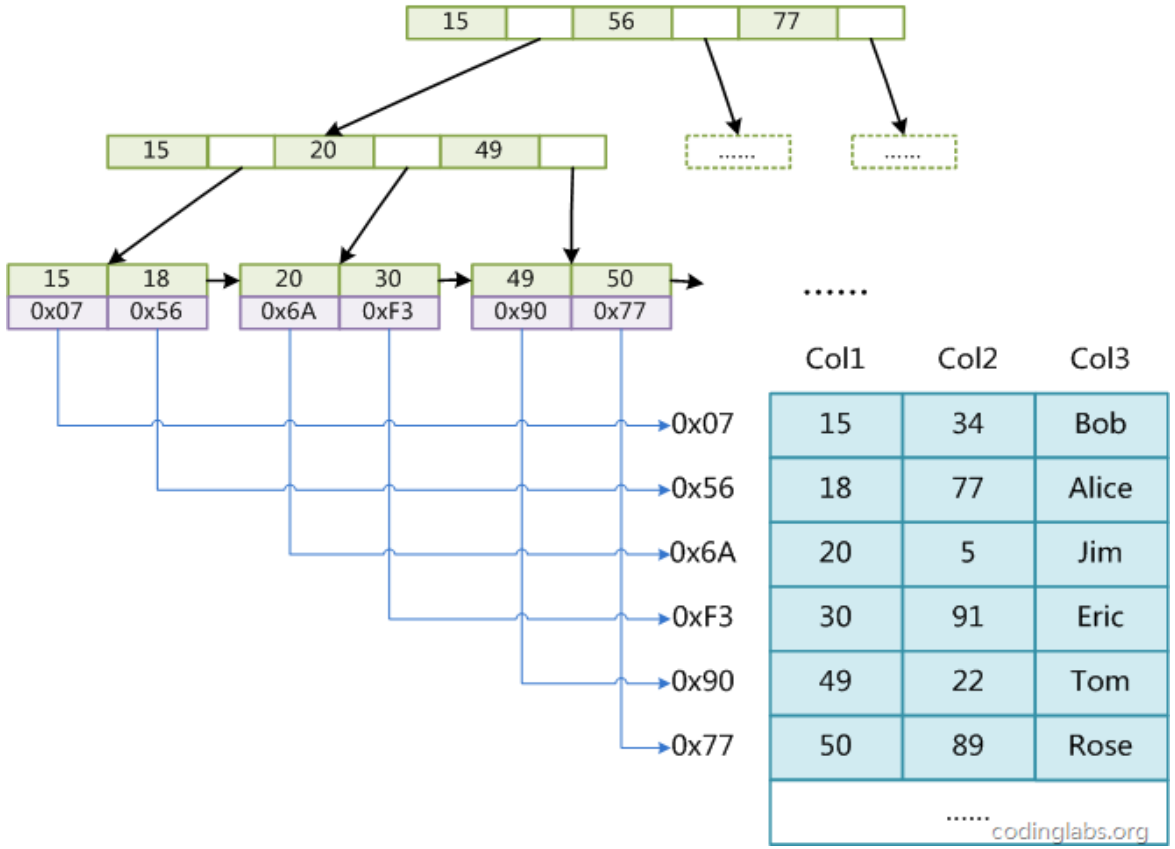
```
create table class(
    id int ,
    name varchar(128) unique ,
    teacher varchar(64),
    index index_mult_columns(id asc,
                             teacher)
);
```

更多可参考：https://blog.csdn.net/Linuxhus/article/details/118249810

索引

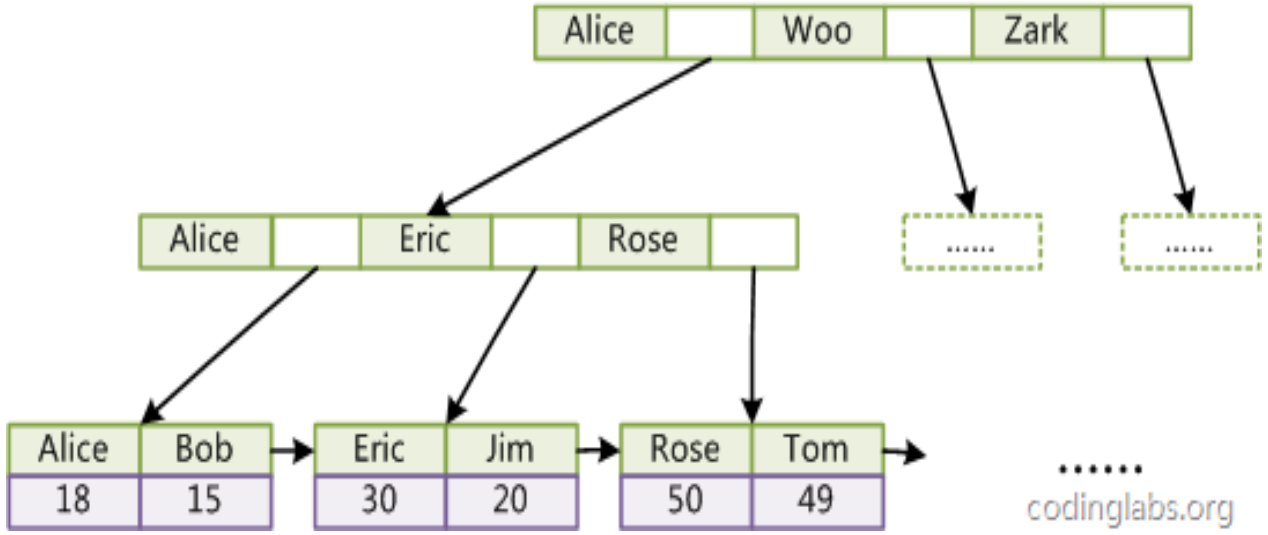
- 索引的原理：B Tree

非聚集索引：



索引

聚集索引:



参考链接: https://www.cnblogs.com/wuchanming/p/6886020.html

关于作业提交

TASK1 提交

请在**PDF/WORD等任何方便助教阅读查看的文档**中按照各个作业要求提交相关内容，**记得标清题号**。

3. 若为PDF/WORD单文档文件直接提交即可，其他提交压缩包，命名为“学号_姓名_第*次实验”。

4. 提交网址：软件学院云平台**第五次上机**　　（按要求提交）

作业截止时间为**周日24:00之前，提交方式为提交到云平台。**