



实 验 人： 伍建霖 学 号： 20337251 日 期： _____

院（系）： 计算机学院 专业（班级）： 网络空间安全

联系方式： QQ 773542531

实验题目： 1.6. 索引实验

掌握索引设计原则和技巧，能够创建合适的索引以提高数据库查询、统计分析效率。

针对给定的数据库模式和具体应用需求,创建唯一索引、函数索引、复合索引等;修改索引;删除索引。设计相应的 SQL 查询验证索引有效性。学习利用 EXPLAIN 命令分析 SQL 查询是否使用了所创建的索引,并能够分析其原因,执行 SQL 查询并估算索引提高查询效率的百分比。要求实验数据集达到 10 万条记录以上的数据量,以便验证索引效果。

实验重点：创建索引。

实验难点：设计 SQL 查询验证索引有效性。

MySQL、SQL Server、Navicat

1) 创建唯一索引

```
create unique index Idx_part_key on PART(PARTKEY);
```

截图如下：

字段	索引	外键	检查	触发器	选项	注释	SQL 预览			
名						字段		索引类型	索引方法	注释
idx part key						'partkey' ASC		UNIQUE	BTREE	

在零件表的零件名称字段上创建一个零件名称长度的函数索引。

1 / 5

字段上创建索引”

```
create index Idx_part_name on PART(name)
```

截图如下：

字段	索引	外键	检查	触发器	选项	注释	SQL 预览		
名								索引类型	索引方法
▶Idx_part_name								NORMAL	BTREE

3) 创建复合索引（对两个及两个以上的属性创建索引，称为复合索引）

在零件表的制造商和品牌两个字段上创建一个复合索引。

```
create index Idx_part_mfgr_brand on PART(MFGR, BRAND);
```

截图如下：

字段	索引	外键	检查	触发器	选项	注释	SQL 预览		
名								索引类型	索引方法
▶Idx_part_name								NORMAL	BTREE
Idx_part_mfgr_brand								NORMAL	BTREE

4) 创建聚簇索引

在零件表的 partkey 字段上创建一个聚簇索引。

```
create clustered index Idx_part_partkey on PART(PARTKEY);
```

注意：对于 MySQL、Sql server，由于主键的列自动存在聚簇索引，因此重复添加聚簇索引会失败！

5) 创建 Hash 索引

在零件表的名称字段上创建一个 Hash 索引。

```
create index Idx_part_name_hash on PART(NAME) USING HASH;
```

截图如下：

字段	索引	外键	检查	触发器	选项	注释	SQL 预览		
名								索引类型	索引方法
▶Idx_part_name								NORMAL	BTREE
Idx_part_mfgr_brand								NORMAL	BTREE
Idx_part_name_hash								NORMAL	BTREE

6) 修改索引名称

将零件表的索引 Idx_part_name_hash 修改为 Idx_part_name_hash_new。

```
ALTER TABLE Part RENAME INDEX Idx_part_name_hash TO Idx_part_name_hash_new
```

截图如下：

字段	索引	外键	检查	触发器	选项	注释	SQL 预览	
名								索引类型
Idx_part_name						`name` ASC		索引方法
Idx_part_mfgr_brand						`mfgr` ASC, `brand` ASC		
Idx_part_name_hash_new						`name` ASC		

7) 分析某个 SQL 查询语句执行时是否使用了索引

```
select * from part where name='压路机'
```

截图如下：

1 select * from part where name='压路机'

partkey	name	mfgr	brand	type	size	container	retailprice	comment
2084	压路机	盐城市高中庄	(Null)	YZ4	(Null)	(Null)	105000	(Null)
2578	压路机	锦州阀门总厂	(Null)	SD175D	(Null)	(Null)	1070000	(Null)
6240	压路机	瑞安市永久泵	(Null)	2YJ8-10	(Null)	(Null)	122000	(Null)
6241	压路机	瑞安市永久泵	(Null)	YZD4	(Null)	(Null)	122000	(Null)
6740	压路机	沈阳风动工具	(Null)	2YJ8/10	(Null)	(Null)	124000	(Null)
6745	压路机	徐州市泰力矿	(Null)	2Y8/10	(Null)	(Null)	124000	(Null)
6746	压路机	徐州市泰力矿	(Null)	2YJ8/10	(Null)	(Null)	124000	(Null)
6747	压路机	徐州市泰力矿	(Null)	2Y6/8	(Null)	(Null)	124000	(Null)

ect * from part where name='压路机'

运行时间: 0.057s

第 1 条记录 (共 1080 条)

id	select_type	table	partitions	type	possible_keys	key	key_len	ref
1	SIMPLE	part	(Null)	ref	Idx part name,Idx part n&Idx part name	Idx part name	223	const

可以看到查询时使用了名为 Idx_part_name 的索引

8) 验证索引效率

- 先创建如下函数：

```
set global log_bin_trust_function_creators=TRUE;

CREATE FUNCTION TestIndex(p_partname CHAR(55) ) RETURNS DOUBLE

BEGIN

    DECLARE begintime DATETIME(6);

    DECLARE endtime DATETIME(6);

    DECLARE durationtime DOUBLE;
```

```

DECLARE  name_count  INTEGER;

SELECT sysdate(6) INTO begintime; /*记录查询执行的开始时间*/

SELECT count(*) FROM Part WHERE name=p_partname into name_count;

SELECT sysdate(6) INTO endtime; /*记录查询执行的结束时间*/

SELECT TimeDiff(endtime,begintime) into durationtime; /*计算并返回查询执行
时间,时间单位为秒 s*/

return durationtime;

END;

```

- 然后，**无索引**查询情况：

/*查看当零件表 Part 数据规模比较大，但无索引时的执行时间*/

/*删除零件表 Part 中所有基于属性 name 建立的索引*/

```
SELECT TestIndex('压路机') ;
```

截图如下：



- 再，**有索引**查询情况：

先创建索引：

```
CREATE INDEX part_name ON Part(name); /*在零件表的零件名称字段上创建索引*/
```

/*查看零件表 Part 数据规模比较大，有索引时的执行时间*/

再查询：

```
SELECT TestIndex('压路机') ;
```

截图如下：

1 SELECT TestIndex('压路机') ;

信息

摘要

结果 1

剖析

状态

TestIndex('压路机')
0.010017

+ - ✓ ✕

ELECT TestIndex('压路机')	只读	运行时间: 0.026s	第 1 条记录 (共 1 条)
------------------------	----	--------------	-----------------

- 结论：
可以看出使用索引在查询时的效率大大提升。

六. 与实验结果相关的文件
无

七. 实验总结