



武汉生物工程学院

MySQL数据库

主讲教师：朱 华





3 索引

索引是一种特殊的文件（innodb数据表上的索引是表空间的一个组成部分），它们包含着对数据表里所有记录的引用指针。

索引在数据库中的作用与目录在书籍中的作用类似，都是用来**提高查找信息的速度**。





3.1 索引的运行机制

索引是一个表中所包含值的字段表，其中注明了表中包含各个值的行所在的存储位置，使用索引查找数据时，先从索引对象中获得相关字段的存储位置，然后再直接去其存储位置查找所需信息，这样就无须对整个表进行扫描，从而可以快速找到所需数据。（引申学科信息检索的概念）





3.1 索引的存储类型

根据索引的存储类型，可以将索引分为**B型树索引（BTREE）**和**哈希索引（HASH）**，一般INNODB和MYISAM支持B型索引，而MEMORY支持HASH类型，默认时支持B型索引。





3.2 索引的代价

索引不是万能的，索引可以加快数据检索操作，但会使数据修改操作变慢。每修改数据记录，索引就必须刷新一次。许多sql命令都有一个 `delay_key_write` 项来控制索引的刷新。

MySQL把同一个数据表里的索引总数限制为16个。





3.2 索引的代价

索引的代价主要有：

- 1、创建索引需要占用数据表以外的**物理存储空间**。
- 2、创建索引和维护索引要**花费一定的时间**。
- 3、当对表进行更新操作时，索引需要被重建，这样就**降低了数据的维护速度**。





3.2 索引的设计原则

索引实质是数据库表的字段值的复制，该字段值称为索引的关键字。设计索引的原则主要有：

- 1、主键字段一定要建立索引。
- 2、表的某个字段值离散度越高，该字段越适合选做索引的关键字。
- 3、占用空间少的字段更适合选做索引的关键字。





3.2 索引的设计原则

- 4、较频繁地作为where查询条件的字段应该创建索引，分组字段或者排序字段应该创建索引，两个表的连接字段（外键）应该创建索引。
- 5、更新频繁的字段不适合创建索引。
- 6、出现在select子句后的字段不应该创建索引。





3.3 索引分类

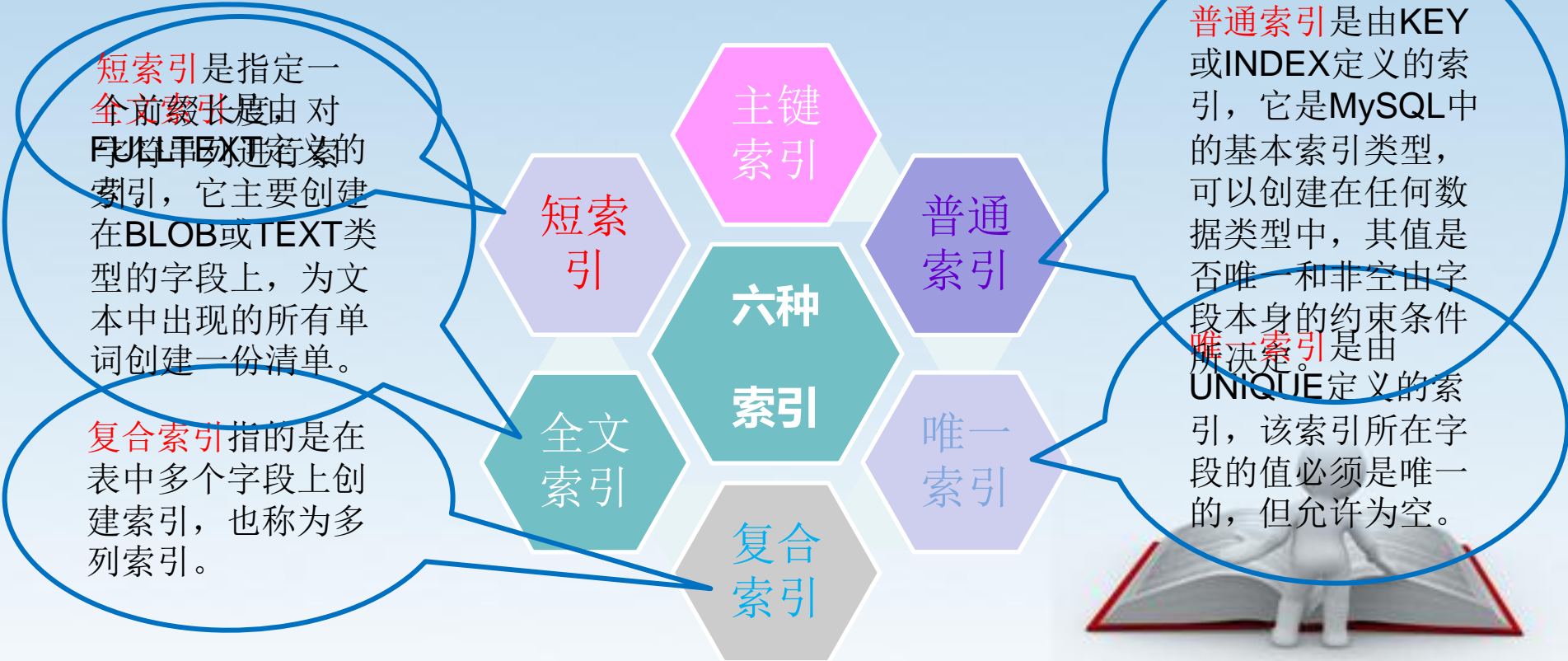
根据索引的存储结构不同将其分为两类，**聚簇索引**和**非聚簇索引**。MySQL数据库会自动将表中的所有记录主键值的“备份”和每条记录所在的起始页组成一张索引表，这种索引称为主键索引，也称为聚簇索引，其余索引称为非聚簇索引。

一张数据表只能创建一个聚簇索引，可以创建多个非聚簇索引。



3.3 索引的分类

MySQL支持6种索引，它们分别为**主键索引**、**普通索引**、**唯一索引**、**复合索引**、**全文索引**和**短索引**。





3.4 创建索引

创建索引是指在某个表的一列或多列上建立一个索引，以便提高对表的访问速度。

创建索引有两种方式，这两种方式分别是：

- 1、创建表的时候创建索引；
- 2、在已经存在的表上创建索引；



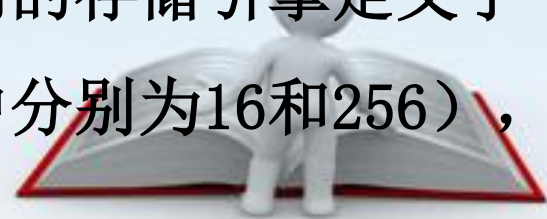


3.4 创建表的时候创建索引

```
CREATE TABLE 表名 ( 字段名 数据类型[完整性约束条件], 字段名 数据类型[完整性约束条件], ..... 字段名 数据类型[完整性约束条件],  
[UNIQUE|FULLTEXT] INDEX [索引名] ( 字段名1 [(长度)] [ASC|DESC] ) )  
[ENGINE=存储引擎];
```

说明：1、INDEX是创建索引的关键（动）词，其后应至少给出一个属性名（字段名）作为索引依据；

2、在属性名后可指定索引长度（不同的存储引擎定义了不同的最大索引数及最大索引长度，MYSQL中分别为16和256），亦可省略；





3.4 创建表的时候创建索引

说明：3、ASC | DESC指定索引时是采用升序（ASC）索引还是降序索引，**默认是升序索引**；

4、INDEX前使用UNIQUE指定创建唯一索引，使用FULLTEXT指定创建全文索引；

5、可在INDEX后第一个属性名前为创建的索引给出一个索引名，亦可省略；

6、作为索引依据的若有多个字段属多列索引，各列间应以“,”分隔；

7、不带任何限制条件（INDEX前无任何可选项）的单列索引为普通索引。





3.4 普通索引的创建

管理员: 命令提示符 - mysql -uroot -p

```
mysql> use company;  
Database changed  
mysql> CREATE TABLE t_dept(deptno int,dname varchar(20),loc varchar(40),INDEX index_deptno(deptno));  
Query OK, 0 rows affected (1.18 sec)
```

检查所创建的索引

管理员: 命令提示符 - mysql -uroot -p

```
mysql> show create table t_dept\G  
***** 1. row *****  
      Table: t_dept  
Create Table: CREATE TABLE `t_dept` (  
  `deptno` int(11) DEFAULT NULL,  
  `dname` varchar(20) DEFAULT NULL,  
  `loc` varchar(40) DEFAULT NULL,  
  KEY `index_deptno` (`deptno`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1  
1 row in set (0.10 sec)
```



3.4 普通索引的创建

检查索引是否被引用

```
C:\ 管理员: 命令提示符 - mysql -uroot -p

mysql> explain select * from t_dept where deptno=1\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: t_dept
         type: ref
possible_keys: index_deptno
          key: index_deptno
       key_len: 5
         ref: const
        rows: 1
     Extra: NULL
1 row in set (0.00 sec)

mysql>
```



3.4 在已存在的表上创建索引

使用CREATE INDEX 语句在已经存在的表上创建索引
语法格式如下：

```
CREATE [UNIQUE|FULLTEXT] INDEX |KEY 索引  
名 ON 表名 (字段名 [(长度)] [ASC|DESC] );
```

这种格式on后的表名应该是先前已经创建好了的无索引表。

其中“CREATE INDEX 索引名 ON 表名（属性名）”是这种语句的
最小构成，是最简单的为已存在的表创建索引（该索引为普通索
引）的语句；这种语句中 [UNIQUE|FULLTEXT] [ASC|DESC]等其他
参数的使用同前面介绍。





3.4 在已存在的表上创建索引

先创建无索引的表

```
mysql> CREATE TABLE t2_dept(deptno int,dname varchar(20),loc varchar(40));  
Query OK, 0 rows affected (0.27 sec)
```

再为该无索引表创建索引

```
mysql> CREATE INDEX index_deptno ON t2_dept (deptno);  
Query OK, 0 rows affected (0.52 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> show create table t2_dept\G  
***** 1. row *****  
      Table: t2_dept  
Create Table: CREATE TABLE `t2_dept` (  
  `deptno` int(11) DEFAULT NULL,  
  `dname` varchar(20) DEFAULT NULL,  
  `loc` varchar(40) DEFAULT NULL,  
  KEY `index_deptno` (`deptno`)  
> ENGINE=InnoDB DEFAULT CHARSET=latin1  
1 row in set (0.00 sec)
```



3.4 在已存在的表上创建索引

使用ALTER TABLE语句在已经存在表上创建索引
语法格式如下：

```
ALTER TABLE 表名 ADD [UNIQUE|FULLTEXT] INDEX  
索引名 (字段名 [(长度)] [ASC|DESC] )
```

这种格式中的表名应该是先前已经创建好了的无索引表。其中“ALTER TABLE 表名 INDEX 索引名（属性）”是这种语句的最小构成，是最简单的为已存在的表创建索引（普通索引）的语句；这种语句中 [UNIQUE|FULLTEXT] [ASC|DESC]等其他参数的使用也同前面介绍。





3.4 在已存在的表上创建索引

管理员: 命令提示符 - mysql -uroot -p

```
mysql> CREATE TABLE t9_dept(deptno int,dname varchar(20),loc varchar(40))ENGINE  
=MyISAM;
```

```
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> ALTER TABLE t9_dept ADD FULLTEXT INDEX index_loc(loc);
```

```
Query OK, 0 rows affected (0.08 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> Show create table t9_dept\G
```

```
***** 1. row *****
```

```
Table: t9_dept
```

```
Create Table: CREATE TABLE `t9_dept` (<
```

```
`deptno` int(11) DEFAULT NULL,
```

```
`dname` varchar(20) DEFAULT NULL,
```

```
`loc` varchar(40) DEFAULT NULL,
```

```
FULLTEXT KEY `index_loc` (`loc`)
```

```
> ENGINE=MyISAM DEFAULT CHARSET=latin1
```

```
1 row in set (0.00 sec)
```

```
mysql>
```



3.4 复合索引的创建示例

以第一种方法创建一带多列字段索引的表：

```
CREATE TABLE t10_dept(deptno int, dname  
varchar(20), loc varchar(40), INDEX  
index_dname_loc(dname, loc)) ;
```





3.4 复合索引的创建示例

```
mysql> CREATE TABLE t10_dept(deptno int,dname varchar(20),loc varchar(40), index index_dname_loc(dname,loc));
Query OK, 0 rows affected (0.38 sec)

mysql> Show create table t10_dept\G
***** 1. row *****
      Table: t10_dept
Create Table: CREATE TABLE `t10_dept` (
  `deptno` int(11) DEFAULT NULL,
  `dname` varchar(20) DEFAULT NULL,
  `loc` varchar(40) DEFAULT NULL,
  KEY `index_dname_loc` (`dname`,`loc`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
1 row in set (0.00 sec)

mysql> explain select * from t10_dept where dname='whsw'\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: t10_dept
         type: ref
possible_keys: index_dname_loc
          key: index_dname_loc
      key_len: 23
         ref: const
        rows: 1
   Extra: Using index condition
1 row in set (0.03 sec)

mysql>
```



3.5 删除索引



- 由于索引会占用一定的磁盘空间，因此，**为了避免影响数据库性能**，应该及时删除不再使用的索引。

1

使用**DROP INDEX**删除索引

语法格式如下所示：

```
DROP INDEX 索引名 ON 表名;
```

2

使用**ALTER TABLE**删除索引

语法格式如下所示：

```
ALTER TABLE 表名 DROP INDEX 索引名
```



3.6 索引对数据查询的影响

索引对**单张表**的影响是没有必要遍历所有的数据，能够快速查找到需要的数据，因为索引的结果会进行**排序**。

索引对**多个表查询**的价值更大，可以避免查找多个表中产生笛卡尔积的数量结果，通过多个表中的索引**快速查找到第一张表中的等值结果**。





3.6 创建索引时的注意事项

- 1、只有表或视图的**所有者才能创建索引**，并且可以随时创建。
- 2、对表中已依次排列的字段集合**只能定义一个索引**。
- 3、在创建聚集索引时，将会对表进行**复制**，对表中的数据进行**排序**，**然后删除原始的表**。
- 4、在使用create index语句创建索引时，**必须指定索引、表以及索引所应用的字段名称**。





3 小结

本节主要介绍关于索引的操作，主要包括**索引创建、索引查看与校验、索引删除**等操作。

通过对本部分内容的学习，我们应掌握索引的基本概念，而且还能熟练地对索引进行各种操作。

