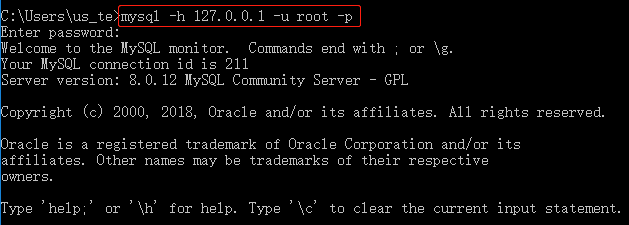
1. **基本术语**

* **数据库:** 数据库是一些关联表的集合。
* **数据表:** 表是数据的矩阵。在一个数据库中的表看起来像一个简单的电子表格。
* **列:** 一列(数据元素) 包含了相同的数据, 例如邮政编码的数据。
* **行：**一行（=元组，或记录）是一组相关的数据，例如一条用户订阅的数据。
* **冗余**：存储两倍数据，冗余降低了性能，但提高了数据的安全性。
* **主键**：主键是唯一的。一个数据表中只能包含一个主键。你可以使用主键来查询数据。
* **外键：**外键用于关联两个表。
* **复合键**：复合键（组合键）将多个列作为一个索引键，一般用于复合索引。
* **索引：**使用索引可快速访问数据库表中的特定信息。索引是对数据库表中一列或多列的值进行排序的一种结构。类似于书籍的目录。
* **参照完整性:** 参照的完整性要求关系中不允许引用不存在的实体。与实体完整性是关系模型必须满足的完整性约束条件，目的是保证数据的一致性。

1. **mysql数据库的连接**



C语言连接

/\*连接数据库\*/

std::string hostip = "localhost";

int port = 3306;

std::string user = "root";

std::string password = "";

std::string database = "metadata2";

std::string tablename = "ngcc\_metadata";

/\*初始化mysql\*/

MYSQL \* pMysql = mysql\_init(NULL); /\*在程序使用Mysql的最开始必须调用mysql\_int()初始化\*/

if (pMysql == NULL)

{

return -1;

}

/\*连接数据库服务器\*/

int arg = 1;

mysql\_options(pMysql, MYSQL\_OPT\_RECONNECT, &arg); /\*断开重连/

if (mysql\_real\_connect(pMysql, hostip.c\_str(),

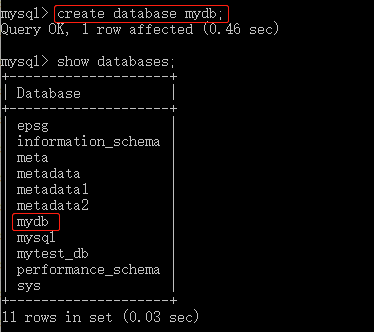
user.c\_str(), password.c\_str(), database.c\_str(), port, NULL, 0) == NULL)

{

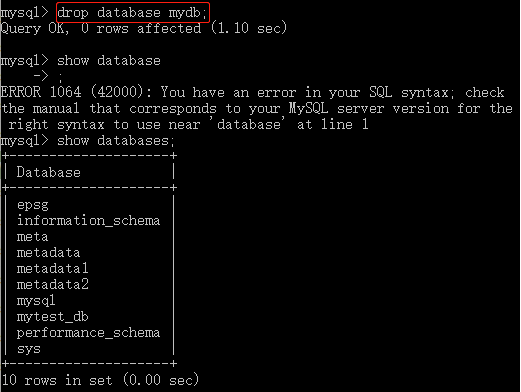
return -1;

}

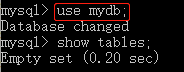
1. **创建数据库**



1. **删除数据库**



1. **选择数据库**



1. **创建数据表**

SET FOREIGN\_KEY\_CHECKS=0;

-- ----------------------------

-- Table structure for `ngcc\_metadata`

-- ----------------------------

DROP TABLE IF EXISTS `ngcc\_metadata`;

CREATE TABLE `ngcc\_metadata` (

`\_id` binary(16) NOT NULL,

`url` varchar(767) NOT NULL,

`status` int(11) NOT NULL,

`1` double DEFAULT NULL COMMENT '地理范围MaxX',

`2` geometry NOT NULL COMMENT '地理范围((西南，东南，东北，西北))',

`3` varchar(50) DEFAULT NULL COMMENT '元数据文件名称',

`4` int(11) DEFAULT NULL COMMENT '产品生产时间',

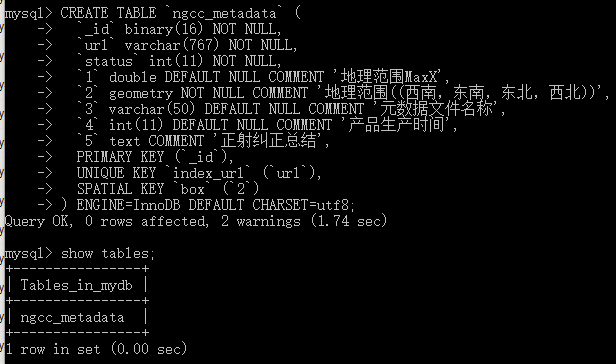
`5` text COMMENT '正射纠正总结',

PRIMARY KEY (`\_id`),

UNIQUE KEY `index\_url` (`url`),

SPATIAL KEY `box` (`2`)

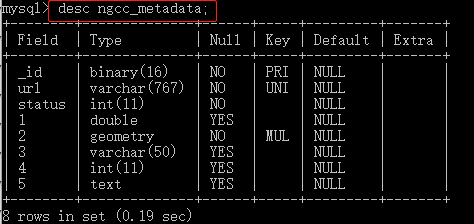
) ENGINE=InnoDB DEFAULT CHARSET=utf8;



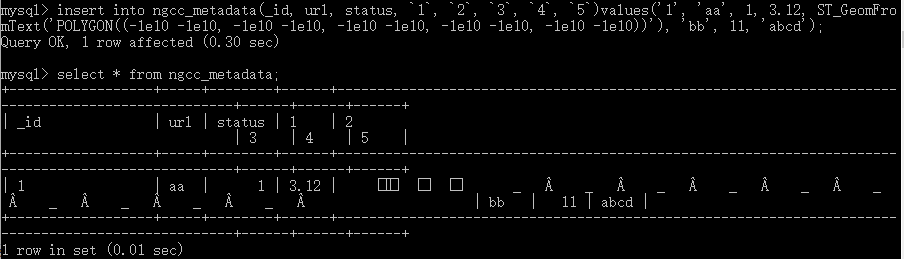
1. **删除数据表**



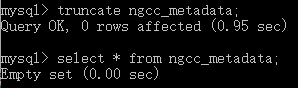
1. **查看表结构**



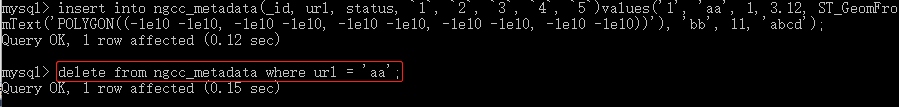
1. **插入数据**



1. **清空数据表**



1. **删除记录**



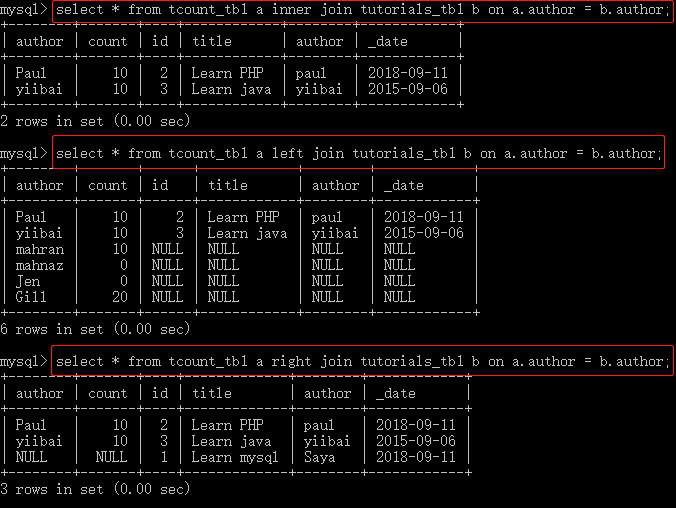
1. **更新记录**



1. **union查询的使用**

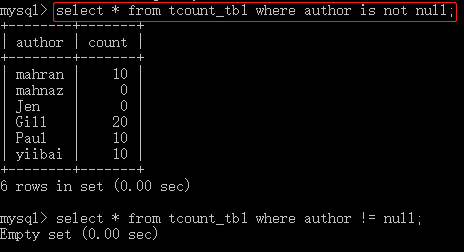


1. **连接**

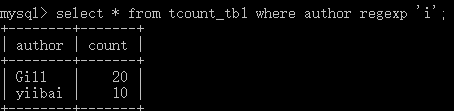


1. **NULL值的使用**

查找数据表中 列值是否为 NULL，必须使用 **IS NULL** 和 **IS NOT NULL**

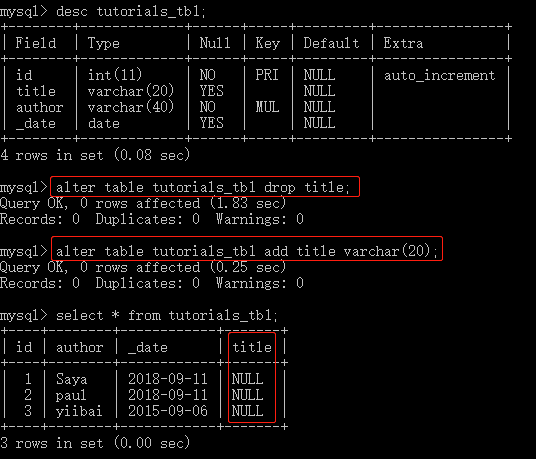


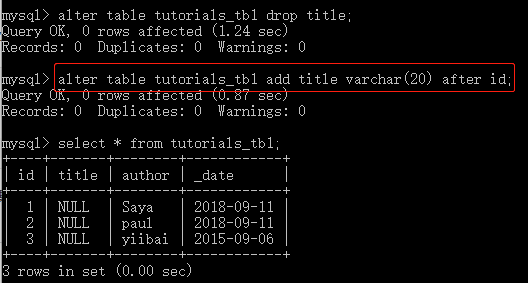
1. **正则表达式**



1. **alter命令**

**（1）删除，添加或者修改字段**



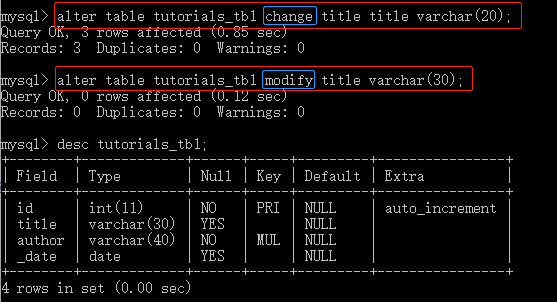


其他示例：

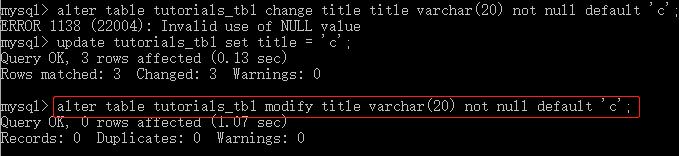
ALTER TABLE testalter\_tbl ADD i INT FIRST;

ALTER TABLE testalter\_tbl ADD i INT AFTER c;

**（2）修改字段类型及名称**

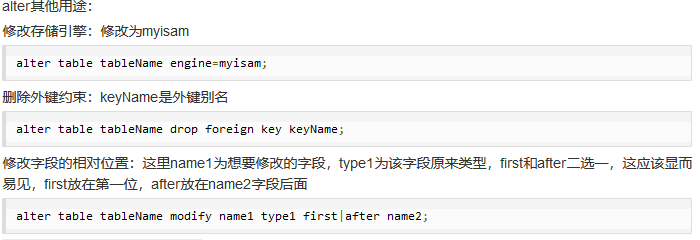


1. **修改字段的默认值**



1. **修改表名**





**Mysql数据库的查询操作：**

**参考：https://www.cnblogs.com/whgk/p/6149009.html**

**单表查询， 多表查询，使用正则表达式查询**

**单表查询：**

1. **查询所有字段：**

**select \* from fruits;**

1. **查询指定字段：**

**Select f\_name, f\_price from fruits;**

1. **查询指定记录：**

**Select \* from fruits where f\_name = ‘apple’;**

1. **带IN关键字的查询：**

**Select \* from fruits where f\_name in(‘apple’, ‘orange’);**

**Select \* from fruits where f\_name not in(‘apple’, ‘orange’);**

1. **带between and的范围查询：**

**Select \* from fruits f\_price between 5 and 15;**

**Select \* from fruits f\_price not between 5 and 15;**

1. **带like的字符匹配查询：**

**Select \* from fruits where f\_name like ‘B%’;**

**Select \* from fruits where f\_name like ‘\_\_\_y’;**

总结：'%'和'\_'可以在任意位置使用，只需要记住%能够表示任意个字符，\_只能表示一个任意字符

1. **查询空值：**

**Select \* from 表名 where 字段名 is (not) null;**

1. **带and的多条件查询：**

**Select \* from fruits where s\_id = 101 and f\_price > 5;**

1. **带or的多条件查询：**

**Select \* from fruits where s\_id = 101 or f\_price > 5;**

1. **关键字distinct（查询结果不重复）：**

**Select distinct s\_id from fruits;**

1. **对查询结果排序（order by）：**

**Select distinct s\_id from fruits order by s\_id; //默认时升序 ASC**

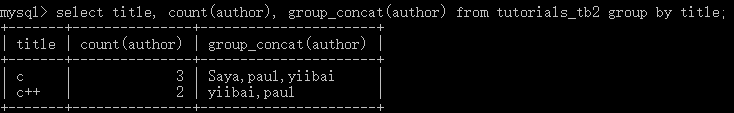
**Select distinct s\_id from fruits order by s\_id DESC; //降序**

1. **分组查询：**

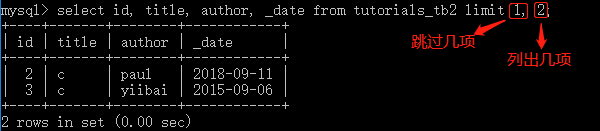
COUNT():这个是下面要讲解到的一个函数，作用就是计算有多少条记录，

GROUP\_CONCAT(): 将分组中的各个字段的值显示出来

SELECT s\_id, COUNT(f\_name), GROUP\_CONCAT(f\_name), GROUP\_CONCAT(f\_price) FROM fruits GROUP BY s\_id;



1. **使用limit限制查询结果的数量：**

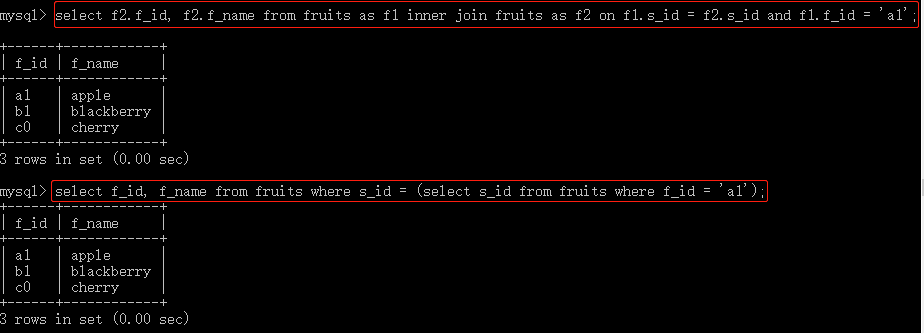


1. **集合函数查询：**

**Count() min() max() sum() avg()**

**多表查询：**

**（1）自连接查询：**



参考：<http://www.360doc.com/content/17/0612/22/43754018_662268826.shtml。>

1. **内连接查询：**



1. **外连接（左外和右外）**：

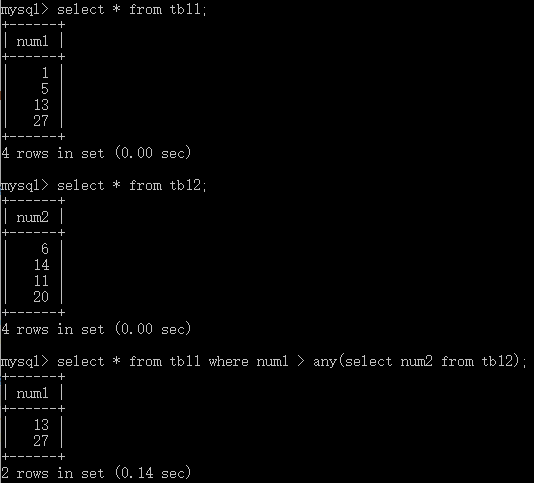


**子查询：**

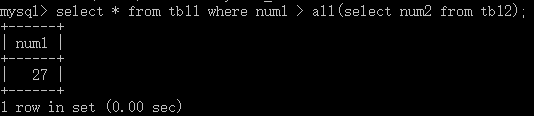
1. **带any ，some关键字的子查询：**

**Any关键字接在一个比较操作符的后面，表示若与子查询返回的任意一个值比较为true，则最终结果就为true。**

**Some跟any用法一样。**

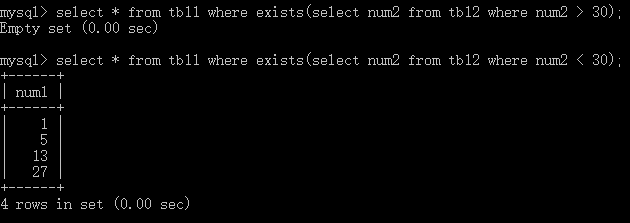


1. **带all关键字的子查询：**

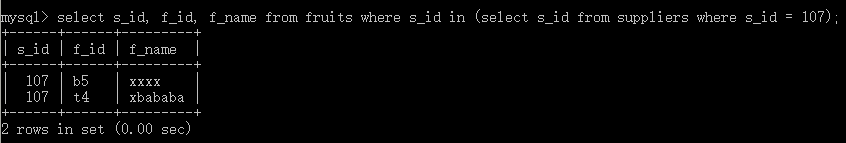


1. **带exists关键字的子查询：**

Exists关键字后面的参数是任意一个子查询，如果子查询有返回记录行，则为true，外层查询语句将会进行查询，如果子查询没有返回记录行，则为false，外层查询语句将不会进行查询。



1. **带in关键字的子查询：**



**Union查询：**



如果去掉all的话，会只去除重复的记录行

**正则查询**