

创建表

create语句即可,用法如下:

```
create table customers(  
    -- 如果未指定NULL属性,则默认可以为NULL值  
    cust_id int NOT NULL AUTO_INCREMENT,  
    cust_name char(50) NOT NULL,    -- 该列不可以是NULL值  
    ...,  
    PRIMARY KEY(cust_id)    -- 指定主键  
    -- 也可以指定其他的一些定义  
)ENGINE=InnoDB; -- 指定引擎
```

当然使用图形化管理工具,如navicat,为我们创建table时很方便,当然只是接口隐藏了sql语句的细节罢了。

如果创建的table名,和已有的重复了? 此时,将会报错。而不是覆盖原表。如果想覆盖原来的表,应先手动删除表,再重建它。

主键再介绍

可以使用多个字段作主键,

```
primary key(field1,field2,...)
```

使用AUTO_INCREMENT

当主键标识,除了唯一以外没有其他意思,例如,订单号可以任意,只要唯一即可。因此,最简单的就是让其递增即可。例如,

```
cust_id int NOT NULL AUTO_INCREMENT
```

其中的AUTO_INCREMENT指示MySQL,本列每次增加一条记录时自动增量,每次执行insert,MySQL自动对其增量,给该记录赋予下一个可用的值。这样每一条记录都是可用的唯一的cust_id,从而可以作为主键。

每个表只允许一个AUTO_INCREMENT字段,而且它必须被索引。

自动填充的值,是根据指定的初始值,然后类似填充脚本生成。

指定默认值

通过create table语句中的default关键字指定。

```
create table orderitems(  
    order_num int NOT NULL,
```

```
order_item int NOT NULL,  
prod_id char(10) NOT NULL,  
quantity int NOT NULL default 1, -- 指定默认值, 一般默认值只能为常量, 不支持函数。  
primary key (order_num, order_item)  
)ENGINE=InnoDB;
```

引擎类型

每个DBMS都有一个处理数据的引擎。例如create table时, 需要指定哪一个引擎负责创建。使用select时也是内部引擎负责处理请求。当然, 如果未指定, 将会使用默认的引擎。

不同的引擎有不同的功能和特性。

更新表 (更新表的定义, 结构, 如添加字段等)

一般来说, 表的定义一开始就要设计好, 不应该有较大改动。通过alter table更改表的结构, 定义等。

```
alter table vendors add vend_phone char(20);
```

这条语句给vendors表添加了一个字段。