

## 实验四 实验 电子商务网站的数据库设计

## 一、实验目的

- (1) 理解并掌握数据库设计过程
- (2) 熟悉数据库相关操作的命令

## 二、实验内容

- 1、根据教材 P95-P109, 完成一个电子商务网站的数据库建设。
- 2、根据教材 P109-P1113, 商品购物流程数据库的设计与实现。
- 3、完成相关数据库以及表的创建。

## 三、相关操作

参看实验材料内容完成。

## 完成一个电子商务网站的数据库建设

**步骤 1:** 要设计电子商务网站的数据库设计, 首先需要创建商品分类表。创建商品分类表的语句如下图 1 所示。

```
mysql> CREATE TABLE sh_goods_category (
  -> id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT COMMENT '分类id',
  -> parent_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '上级分类id',
  -> name VARCHAR(100) NOT NULL DEFAULT '' COMMENT '名称',
  -> sort INT NOT NULL DEFAULT 0 COMMENT '排序',
  -> is_show TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否显示',
  -> create_time DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  -> update_time DATETIME DEFAULT NULL COMMENT '更新时间'
  -> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.05 sec)
```

图 1 创建商品分类表

**步骤 2:** 创建商品分类表后, 紧接着要做的就是添加测试数据。添加测试数据的语句和结果如下图 2 所示。

```
mysql> INSERT INTO sh_goods_category (id, parent_id, name) VALUES
  -> (1, 0, '办公'), (2, 1, '耗材'), (3, 2, '文具'),
  -> (4, 0, '电子产品'), (5, 4, '通讯'), (6, 5, '手机'), (7, 4, '影音'), (8, 7, '音箱'), (9, 7, '耳机'),
  -> (10, 4, '电脑'), (11, 10, '台式电脑'), (12, 10, '笔记本'),
  -> (13, 0, '服装'), (14, 13, '女装'), (15, 14, '风衣'), (16, 14, '毛衣');
Query OK, 16 rows affected (0.04 sec)
Records: 16 Duplicates: 0 Warnings: 0
```

图 2 添加测试数据

**步骤 3:** 添加数据之后, 紧接着就是创建商品表。创建商品表的语句和结果如下图 3 所示。

```
mysql> CREATE TABLE sh_goods (
  -> id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT COMMENT '商品 id',
  -> category_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '分类id',
  -> spu_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT 'SPU id',
  -> sn VARCHAR(20) NOT NULL DEFAULT '' COMMENT '编号',
  -> name VARCHAR(120) NOT NULL DEFAULT '' COMMENT '名称',
  -> keyword VARCHAR(255) NOT NULL DEFAULT '' COMMENT '关键词',
  -> picture VARCHAR(255) NOT NULL DEFAULT '' COMMENT '图片',
  -> tips VARCHAR(255) NOT NULL DEFAULT '' COMMENT '提示',
  -> description VARCHAR(255) NOT NULL DEFAULT '' COMMENT '描述',
  -> content TEXT NOT NULL COMMENT '详情',
  -> price DECIMAL(10, 2) UNSIGNED NOT NULL DEFAULT 0 COMMENT '价格',
  -> stock INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '库存',
  -> score DECIMAL(3, 2) UNSIGNED NOT NULL DEFAULT 0 COMMENT '评分',
  -> is_on_sale TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否上架',
  -> is_del TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否删除',
  -> is_free_shipping TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否包邮',
  -> sell_count INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '销量计数',
  -> comment_count INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '评论计数',
  -> on_sale_time DATETIME DEFAULT NULL COMMENT '上架时间',
  -> create_time DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  -> update_time DATETIME DEFAULT NULL COMMENT '更新时间'
  -> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.07 sec)
```

图 3 创建商品表

**步骤 4:** 创建商品表之后, 紧接着就是插入测试数据。插入测试数据的语句和结果如下图 4 所示。

```
mysql> INSERT INTO sh_goods(id, category_id, name, keyword, content, price, stock, score, comment_count) VALUES
-> (1, 3, '2B铅笔', '文具', '考试专用', 0.5, 5001, 4.9, 40000),
-> (2, 3, '钢笔', '文具', '练字必不可少', 15.3, 300, 3.9, 500),
-> (3, 3, '碳素笔', '文具', '平时使用', 1.5, 500, 5.98000),
-> (4, 12, '超薄笔记本', '电子产品', '轻小便携', 5999, 0, 2.5, 200),
-> (5, 6, '智能手机', '电子产品', '人人必备', 1999, 0, 5.98000),
-> (6, 8, '面音箱', '电子产品', '扩音装备', 69.75, 4.5, 1000),
-> (7, 9, '头戴耳机', '电子产品', '独享个人世界', 109, 0, 3.9, 500),
-> (8, 10, '办公电脑', '电子产品', '适合办公', 2000, 0, 4.8, 6000),
-> (9, 15, '收腰风衣', '服装', '春节潮流单品', 299, 0, 4.9, 40000),
-> (10, 16, '薄毛衣', '服装', '居家旅行必备', 48, 0, 4.8, 98000);
Query OK, 10 rows affected (0.04 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

图 4 插入测试数据

**步骤 5:** 插入测试数据之后, 紧接着就是创建含有商品属性多维度的表。创建含有商品属性多维度的表的语句和结果如下图 5-8 所示。

```
mysql> #商品SPU表
mysql> CREATE TABLE sh_goods_spu(
-> id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT COMMENT 'SPU id',
-> name VARCHAR(80) NOT NULL DEFAULT '' COMMENT 'SPU名称'
-> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.06 sec)
```

图 5 创建商品 SPU 表

```
mysql> CREATE TABLE sh_goods_spec(
-> id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT COMMENT '规格 id',
-> name VARCHAR(80) NOT NULL DEFAULT '' COMMENT '规格名称'
-> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.05 sec)
```

图 6 创建商品规格表

```
mysql> #商品规格项表
mysql> CREATE TABLE sh_goods_spec_item(
-> id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT COMMENT '规格项id',
-> spec_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '规格id',
-> name VARCHAR(80) NOT NULL DEFAULT '' COMMENT '名称',
-> description VARCHAR(255) NOT NULL DEFAULT '' COMMENT '描述',
-> picture VARCHAR(255) NOT NULL DEFAULT '' COMMENT '可选图'
-> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.05 sec)
```

图 7 创建商品规格表

```
mysql> #商品规格组合表
mysql> CREATE TABLE sh_goods_spec_set(
-> goods_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT 'SKU id',
-> spec_item_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '规格 id'
-> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.02 sec)
```

图 8 创建商品规格组合表

**步骤 6:** 创建商品规格组合表之后, 紧接着就是向上述四张表添加测试数据。添加测试数据的语句和结果如下图 9 所示。



```

mysql> INSERT INTO sh_goods_spu(id,name) VALUES
-> (1,'新款智能手机');
Query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO sh_goods_spec (id, name) VALUES
-> (1,'网络'), (2,'颜色'), (3,'内存');
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> INSERT INTO sh_goods_spec_item (id, spec id, name) VALUES
-> (1,1,'移动'), (2,1,'电信'), (3,2,'白色'),
-> (4,2,'黑色'), (5,3,'4GB'), (6,3,'6GB');
ERROR 1064 (42000): You have an error in your SQL syntax; check the m
(1,1,'移动'), (2,1,'电信'), (3,2,'白色'),
(4,2,'黑色') at line 1
mysql> INSERT INTO sh_goods_spec_set(goods_id, spec_item_id) VALUES
-> (5,2), (5,3), (5,5);
Query OK, 3 rows affected (0.03 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> INSERT INTO sh_goods_spec_item (id, spec_id, name) VALUES
-> (1,1,'移动'), (2,1,'电信'), (3,2,'白色'),
-> (4,2,'黑色'), (5,3,'4GB'), (6,3,'6GB');
Query OK, 6 rows affected (0.04 sec)
Records: 6 Duplicates: 0 Warnings: 0

```

图 9 向商品规格组合表添加测试数据

**步骤 7:** 向商品规格组合表添加测试数据之后, 紧接着就是要考虑不同分类的商品应该具有不同的属性模板。创建含有商品不同属性模板的语句和结果如下图 10-13 所示。

```

mysql> CREATE TABLE sh_goods_attr (
-> id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT COMMENT '属性 id',
-> parent_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '上级属性id',
-> category_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '商品分类 id',
-> name VARCHAR(50) NOT NULL DEFAULT '' COMMENT '名称',
-> sort INT NOT NULL DEFAULT 0 COMMENT '排序',
-> )ENGINE=INNODB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.01 sec)

```

图 10 创建商品属性表

```

mysql> CREATE TABLE sh_goods_attr_value (
-> id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT COMMENT '属性值 id',
-> goods_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '商品 id',
-> attr_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '属性 id',
-> attr_value VARCHAR(80) NOT NULL DEFAULT '' COMMENT '属性值',
-> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.02 sec)

```

图 11 创建商品属性值表

```

mysql> CREATE TABLE sh_goods_selector (
-> id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT COMMENT '筛选 id',
-> parent_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '上级筛选 id',
-> category_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '商品分类id',
-> name VARCHAR(80) NOT NULL DEFAULT '' COMMENT '名称',
-> sort INT NOT NULL DEFAULT 0 COMMENT '排序',
-> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.02 sec)

```

图 12 创建商品筛选表

```

mysql> CREATE TABLE sh_goods_selector_value(
-> id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT COMMENT '筛选值id',
-> goods_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '商品id',
-> selector_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '筛选id',
-> selector_value VARCHAR(80) NOT NULL DEFAULT '' COMMENT '筛选值',
-> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.02 sec)

```

图 13 创建商品筛选值表

**步骤 8:** 创建含有商品不同属性模板表之后, 紧接着就是向上述四张表添加测试数据。添加测试数据的语句和结果如下图 14 所示。

```
mysql> INSERT INTO sh_goods_attr VALUES
-> (1,0,6,'基本信息',0),(2,1,6,'机身颜色',0),(3,1,6,'输入方式',1),(4,1,6,'操作系统',2),
-> (5,0,6,'屏幕',1),(6,5,6,'屏幕尺寸',0),(7,5,6,'屏幕材质',1),(8,5,6,'分辨率',2),(9,0,6,'摄像头',2),
-> (10,9,6,'前置摄像头',0),(11,9,6,'后置摄像头',1),(12,0,6,'电池信息',3),(13,12,6,'电池容量',0),(14,12,6,'是否可拆卸',1);
Query OK, 14 rows affected (0.01 sec)
Records: 14 Duplicates: 0 Warnings: 0

mysql> INSERT INTO sh_goods_attr_value VALUES
-> (1,5,2,'黑色'),(2,5,3,'触摸屏'),(3,5,4,'Android'),(4,5,6,'5.5 寸'),(5,5,7,'IPS'),
-> (6,5,8,'1920*1080'),(7,5,10,'1600万'),(8,5,11,'800万'),(9,5,13,'3500mAh'),(10,5,14,'否');
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

图 14 向含有商品不同属性模板表添加测试数据

**步骤 9:** 向含有商品不同属性模板表添加测试数据之后, 紧接着就是创建用户表。创建用户表的语句和结果如下图 15 所示。

```
mysql> CREATE TABLE sh_user(
-> id INT UNSIGNED PRIMARY KEY AUTO INCREMENT COMMENT '用户id',
-> name VARCHAR(100) NOT NULL UNIQUE DEFAULT '' COMMENT '用户名',
-> password VARCHAR(255) NOT NULL DEFAULT '' COMMENT '密码',
-> salt CHAR(32) NOT NULL DEFAULT '' COMMENT '密码盐',
-> email VARCHAR(128) NOT NULL DEFAULT '' COMMENT '邮箱',
-> mobile CHAR(11) NOT NULL DEFAULT '' COMMENT '手机号',
-> level TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '用户级别',
-> money DECIMAL(10, 2) UNSIGNED NOT NULL DEFAULT 0 COMMENT '金额',
-> gender TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '性别',
-> qq VARCHAR(20) NOT NULL DEFAULT '' COMMENT 'QQ',
-> is_active TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否激活',
-> reg_time DATETIME DEFAULT NULL COMMENT '注册时间',
-> create_time DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
-> update_time DATETIME DEFAULT NULL COMMENT '更新时间'
-> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.02 sec)
```

图 15 创建用户表

**步骤 10:** 创建用户表之后, 考虑到 Web 项目的安全性, 应对用户的密码进行加密, 不能明文存储, 否则一旦泄露会造成很大的损失。用 MD5 加盐算法加密方式的语句和结果如下图 16 所示。

```
mysql> SELECT MD5(CONCAT(MD5('password'),'salt'));
+-----+
| MD5(CONCAT(MD5('password'),'salt')) |
+-----+
| d514dee5e76bbb718084294c835f312c |
+-----+
1 row in set (0.00 sec)
```

图 16 利用 MD5 加盐算法加密

**步骤 11:** 用户表创建完并利用 MD5 加盐算法加密之后, 紧接着就是为用户表添加测试数据, 添加数据的语句和结果如下图 17 所示。

```
mysql> INSERT INTO sh_user(id,name,password,salt,money,is_active) VALUES
-> (1,'Alex',MD5(CONCAT(MD5('123'),'salt1')),'salt1',1000,1),
-> (2,'Bill',MD5(CONCAT(MD5('123'),'salt2')),'salt2',1000,1);
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

图 17 为用户表添加测试数据



**步骤 12:** 可以通过 SELECT 命令来查看 sh\_user 表中的记录, 观察密码加密的结果。其使用的方式如下图 18 所示。

```
mysql> SELECT id,name,password,salt FROM sh_user;
```

id	name	password	salt
1	Alex	a3a5b6c3c00b37a4c53f82dlce8bb86c	salt1
2	Bill	3c2b3fef1af992bc0313247b089b954d	salt2

2 rows in set (0.00 sec)

图 18 查看密码

**步骤 13:** 创建用户表之后, 还需要创建商品评论表。创建商品评论表的 SQL 语句和结果如下图 19 所示。

```
mysql> CREATE TABLE sh_goods_comment (
  -> id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT COMMENT '评论 id',
  -> parent_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '上级评论 id',
  -> user_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '用户 id',
  -> goods_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '商品 id',
  -> content TEXT NOT NULL COMMENT '评论内容',
  -> is_staff TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否为工作人员',
  -> is_show TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否显示',
  -> is_del TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否删除',
  -> create_time DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  -> update_time DATETIME DEFAULT NULL COMMENT '更新时间'
  -> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.01 sec)
```

图 19 创建商品评论表

**步骤 14:** 创建商品评论表之后, 还需要为商品添加测试数据。添加测试数据的 SQL 语句和结果如下图 20 所示。

```
mysql> INSERT INTO sh_goods_comment(id, user_id, goods_id, content, is_show, create_time)
VALUES
  -> (1,1,8,'好',0,'2017-11-08 00:00:00'),
  -> (2,2,10,'不错',1,'2017-12-03 00:00:00'),
  -> (3,3,9,'满意',1,'2017-12-30 00:00:00'),
  -> (4,4,4,'携带方便',1,'2018-01-19 00:00:00'),
  -> (5,4,7,'中低音效果特别棒',1,'2018-01-19 00:00:00'),
  -> (6,5,8,'卡机',1,'2018-01-22 00:00:00'),
  -> (7,6,5,'黑夜拍照也清晰',1,'2018-02-15 00:00:00'),
  -> (8,7,9,'掉色、有线头',0,'2018-03-03 00:00:00'),
  -> (9,4,9,'还行',1,'2018-04-05 00:00:00'),
  -> (10,8,9,'特别彰显气质',1,'2018-04-16 00:00:00');
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

图 20 添加测试数据

### 商品购物流程数据库的设计与实现

**步骤 1:** 在商品购物过程中, 用户可以将想购买的商品添加到购物车, 从而方便一次购买多件商品, 此外, 将商品添加到购物车后不会影响商品的库存。因此给出如下图 21 所示的 SQL 语句和结果。

```
mysql> CREATE TABLE sh_user_shopcart (
  -> id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT COMMENT '购物车 id',
  -> user_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '用户 id',
  -> goods_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '商品 id',
  -> goods_price DECIMAL(10, 2) UNSIGNED NOT NULL DEFAULT 0 COMMENT '单价',
  -> goods_num INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '购买件数',
  -> is_select TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否选中',
  -> create_time DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  -> update_time DATETIME DEFAULT NULL COMMENT '更新时间'
  -> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.05 sec)
```

图 21 创建购物车表

**步骤 2:** 创建购物车表之后, 还需要创建存储收货地址的表。创建存储收货地址的表的 SQL 语句和结果如下图 22 所示。

```
mysql> CREATE TABLE sh_user_address (
  -> id INT UNSIGNED PRIMARY KEY AUTO INCREMENT COMMENT '地址 id',
  -> user_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '用户 id',
  -> is_default TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否默认',
  -> province VARCHAR(20) NOT NULL DEFAULT '' COMMENT '省',
  -> city VARCHAR(20) NOT NULL DEFAULT '' COMMENT '市',
  -> district VARCHAR(20) NOT NULL DEFAULT '' COMMENT '区',
  -> address VARCHAR(255) NOT NULL DEFAULT '' COMMENT '具体地址',
  -> zip VARCHAR(20) NOT NULL DEFAULT '' COMMENT '邮编',
  -> consignee VARCHAR(20) NOT NULL DEFAULT '' COMMENT '收件人',
  -> phone VARCHAR(20) NOT NULL DEFAULT '' COMMENT '联系电话',
  -> create_time DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  -> update_time DATETIME DEFAULT NULL COMMENT '更新时间'
  -> )ENGINE=INNODB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.02 sec)
```

图 22 创建收货地址表

**步骤 3:** 创建存储收货地址表之后, 还需要创建存订单表和订单商品表。创建存订单表和订单商品表的 SQL 语句和结果如下图 23、24 所示。

```
mysql> #订单表
mysql> CREATE TABLE sh_order(
  -> id INT UNSIGNED PRIMARY KEY AUTO INCREMENT COMMENT '订单 Id',
  -> user_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '用户 id',
  -> total_price DECIMAL(10, 2) UNSIGNED NOT NULL DEFAULT 0 COMMENT '订单总价',
  -> order_price DECIMAL(10, 2) UNSIGNED NOT NULL DEFAULT 0 COMMENT '应付金额',
  -> province VARCHAR(20) NOT NULL DEFAULT '' COMMENT '省',
  -> city VARCHAR(20) NOT NULL DEFAULT '' COMMENT '市',
  -> district VARCHAR(20) NOT NULL DEFAULT '' COMMENT '区',
  -> address VARCHAR(255) NOT NULL DEFAULT '' COMMENT '具体地址',
  -> zip VARCHAR(20) NOT NULL DEFAULT '' COMMENT '邮编',
  -> consignee VARCHAR(20) NOT NULL DEFAULT '' COMMENT '收件人',
  -> phone VARCHAR(20) NOT NULL DEFAULT '' COMMENT '联系电话',
  -> is_valid TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否有效',
  -> is_cancel TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否取消',
  -> is_pay TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否付款',
  -> status TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '物流状态',
  -> is_del TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否删除',
  -> create_time DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  -> update_time DATETIME DEFAULT NULL COMMENT '更新时间'
  -> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.06 sec)
```

图 23 创建订单表

```
mysql> #订单商品表
mysql> CREATE TABLE sh_order_goods(
  -> id INT UNSIGNED PRIMARY KEY AUTO INCREMENT COMMENT 'id',
  -> order_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '订单id',
  -> goods_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '商品id',
  -> goods_name VARCHAR(120) NOT NULL DEFAULT '' COMMENT '商品名称',
  -> goods_num INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '购买数量',
  -> goods_price DECIMAL(10, 2) UNSIGNED NOT NULL DEFAULT 0 COMMENT '单价',
  -> user_note VARCHAR(255) NOT NULL DEFAULT '' COMMENT '用户备注',
  -> staff_note VARCHAR(255) NOT NULL DEFAULT '' COMMENT '卖家备注'
  -> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.07 sec)
```

图 24 创建订单商品表

**步骤 4:** 创建存订单表和订单商品表之后, 还需要创建存商品评分表。创建存商品评分表的 SQL 语句和结果如下图 25 所示。



```

)ENGINE=InnoDB DEFAULT CHARSET=utf8' at line 9
mysql> CREATE TABLE sh_goods_score (
-> id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT COMMENT '评分 id',
-> user_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '用户 id',
-> goods_id INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '商品 id',
-> goods_score TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '商品评分',
-> service_score TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '服务评分',
-> express_score TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '物流评分',
-> is_invalid TINYINT UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否无效',
-> create_time DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '评分时间'
-> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.06 sec)

```

图 25 创建存商品评分表

#### 四、问题思考与总结

##### 1、数据库设计的过程有哪些步骤？

数据库设计一般要经过需求分析、概要设计、逻辑设计、代码编写、软件测试、安装部署六步。需求分析是数据库设计的第一步，是最困难、最耗费时间的一步，也是整个设计过程的基础。需求分析是否做的充分和准确，直接决定了在其上构建数据库大厦的速度与质量。需求分析做的不好，会导致整个数据库设计返工重做。概要设计概要设计是数据库设计的关键，通过综合、归纳与抽象用户需求，形成一个具体 DBMS 的概念模型，也就是绘制数据库的 E-R 图；逻辑结构设计

将 E-R 图转换为多张表，进行逻辑设计，确认各表的主外键，并应用数据库设计的三大范式进行审核，对其优化；物理设计阶段经项目组开会讨论确定 E-R 图后，根据项目的技术实现，团队开发能力及项目的成本预算，选择具体的数据库（如 MySQL 或 Oracle 等）进行物理实现。数据库实施阶段运用 DBMS 提供的数据库语言（例如 SQL）、工具及宿主语言，根据逻辑设计和物理设计的结果建立数据库，编制与调试应用程序，组织数据入库，并进行试运行。数据库运行和维护阶段

##### 2、确定数据类型需要注意哪些问题？

###### (1) INT 数据类型可以用来保存那些不包含小数点的数字。INT 代表整数。其中：

TINYINT——这个类型最多可容纳三位数 8bit

SMALLINT——最多可容纳五位数 16bit

MEDIUMINT——最多可容纳八位数 24bit

INT——可以容纳十位数 32bit

BIGINT——最多可容纳二十位数 64bit

###### (2) 整数类型有可选的 unsigned（无符号）属性，它表示不允许为负数，并大致把正上限提高了一倍，例如：tinyint unsigned 保存的翻围为 0 到 255，而不是-127 到 128。

###### (3) 实数有分数部分，然而，它们并不仅仅是分数。可以使用 decimal 保存比出 bigint 还大的整数

###### (4) 比起 decimal 类型，浮点类型保存同样大小的值使用的空间通常更小，float 类型占用 4 个字节，double 占用 8 个字节，而且精度更大，范围更广。只有在需要对小数进行精确的时候才使用 decimal

###### (5) varchar 保存了可变长度的字符串，是使用得最多的字符串类型，它能比固定类型占用更少的存储空间，因为它只占用了自己需要的空间（也就是说较短的值占用的空间更小）。它使用额外的 1-2 个字节来存储值的长度，在存取值时候，MySQL 不会去掉字符串末尾的空格。

###### (6) char 固定长度，char 存取值时候，MySQL 会去掉末尾的空格。Char 在存储很短的字符串或长度近似相同的字符的时候很有用，MySQL 使用\0（0 字节）填充 binary 值，而不是空格，并且不会在获取数据的时候把填充的值截掉。

- (7) TEXT 保存的是字符数据，有字符集和排序规则
- (8) DATETIME: 能够保存大范围的值，从 1001 年到 9999 年，精度为秒，它把日期和时间封装到一个格式为 yyyy:MM:dd:HH:mm:ss 的整数当中，与时区无关。它使用了 8 个字节存储空间。
- (9) TIMESTAMP 显示的值依赖于时区，MySQL 服务器、操作系统及客户端连接都有时区设置。

### 3、其他自己遇到的问题整理总计

由于本次实验是照着书上代码写的，因此除了粗心打错字母之外，没有遇到棘手的问题。唯一遇到的非粗心问题见下述：

**问题 1：**当我登陆成功后，不管运行任何命令，总是提示如下面图 26 所示的错误。

```
mysql> create database shop;
ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.
```

图 26 报错 1 截图

遇到这个问题的原因是从 MySQL 版本 5.6.6 版本起，添加了 password\_expired 功能，它允许设置用户的过期时间。这个特性已经添加到 mysql.user 数据表，但是它的默认值是“N”，可以使用 SET PASSWORD 语句来修改这个值。，解决成功截图如下图 27 所示。

```
mysql> SET PASSWORD = PASSWORD('20180112');
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> create database shop;
Query OK, 1 row affected (0.00 sec)
```

图 27 报错 1 解决截图

### 4、本章重要命令的总结。

创建数据表：CREATE TABLE+数据表名

插入数据：INSERT INTO + 数据表名 + 属性 + values

查看数据：SELECT +属性 + from +表名

非空约束：name VARCHAR(20) NOT NULL

主键约束：alter table UserId add constraint PK\_UserId primary key (UserId)

唯一约束：alter table UserInfo add constraint UQ\_IDNumber unique(IdentityCardNumber)

默认约束：alter table UserInfo add constraint DF\_UserAddress default ( '地址不详' ) for UserAddress

检查约束：alter table UserInfo add constraint CK\_UserAge check (UserAge between 20 and 40)

外键约束：alter table UserOrder add constraint FK\_UserId\_UserId foreign key(UserId)references UserInfo(UserId)