数据库设计

表1. 用户表 user

```
CREATE TABLE user (
uid INT PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '用户ID',
username VARCHAR(50) NOT NULL COMMENT '用户名',
password VARCHAR(255) NOT NULL COMMENT '密码',
email VARCHAR(100) COMMENT '邮箱',
phone VARCHAR(20) COMMENT '电话',
status TINYINT DEFAULT 1 COMMENT '状态：1正常，0禁用',
avatar VARCHAR(255) COMMENT '头像',
role VARCHAR(10) COMMENT '角色：admin/user',
create_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
update_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间'
) COMMENT '用户表';
```

表2. 商品表 product

```
CREATE TABLE product (
pid INT PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '商品ID',
name VARCHAR(100) NOT NULL COMMENT '商品名称',
price DECIMAL(10, 2) NOT NULL COMMENT '价格',
description TEXT COMMENT '描述',
CPU VARCHAR(50) COMMENT '处理器',
GPU VARCHAR(50) COMMENT '显卡',
storage VARCHAR(50) COMMENT '内存',
size VARCHAR(20) COMMENT '屏幕尺寸',
type VARCHAR(50) COMMENT '产品类型',
picture VARCHAR(255) COMMENT '默认图片',
stock INT NOT NULL DEFAULT 0 COMMENT '库存',
rating INT DEFAULT 0 COMMENT '评分（推荐度）'
) COMMENT '商品表';
```

表3. 商家表 business

```
CREATE TABLE business (
bid INT PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '商家ID',
bname VARCHAR(100) NOT NULL COMMENT '商家名称',
password VARCHAR(255) NOT NULL COMMENT '密码'
) COMMENT '商家表';
```

表4. 商家-商品关联表 business_product

```
CREATE TABLE business_product (
bpid INT PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '关联ID',
bid INT NOT NULL COMMENT '商家ID',
pid INT NOT NULL COMMENT '商品ID',
FOREIGN KEY (bid) REFERENCES business(bid),
FOREIGN KEY (pid) REFERENCES product(pid)
) COMMENT '商家-商品关联表';
```

表5. 身份表 identity

```
CREATE TABLE identity (
iid INT PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '身份ID',
level INT DEFAULT 0 COMMENT '等级：0~5',
student TINYINT DEFAULT 0 COMMENT '是否学生：1学生，0普通',
points INT DEFAULT 0 COMMENT '积分：1元1分，5000分一段',
uid INT COMMENT '用户ID',
FOREIGN KEY (uid) REFERENCES user(uid)
) COMMENT '身份表';
```

表6. 购物车表 cart

```
CREATE TABLE cart (
cid INT PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '购物车ID',
uid INT COMMENT '用户ID',
pid INT COMMENT '商品ID',
FOREIGN KEY (uid) REFERENCES user(uid),
FOREIGN KEY (pid) REFERENCES product(pid)
) COMMENT '购物车表';
```

表7. 评论表 feedback

```
CREATE TABLE feedback (
fid INT PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '评论ID',
uid INT COMMENT '用户ID',
pid INT COMMENT '商品ID',
feed_time VARCHAR(50) COMMENT '评论时间',
star INT DEFAULT 5 COMMENT '商品评级',
comment TEXT COMMENT '评论内容',
FOREIGN KEY (uid) REFERENCES user(uid),
FOREIGN KEY (pid) REFERENCES product(pid)
) COMMENT '评论表';
```

表8.订单表 (ordered)

```
CREATE TABLE ordered (
oid INT PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '订单ID',
status INT DEFAULT 0 COMMENT '状态',
order_time DATETIME COMMENT '订单时间',
amount DECIMAL(10, 2) DEFAULT 0.00 COMMENT '实付款',
uid INT COMMENT '用户ID',
receiver_address VARCHAR(255) COMMENT '联系地址',
receiver_name VARCHAR(50) COMMENT '联系人',
receiver_phone VARCHAR(20) COMMENT '联系电话',
remark TEXT COMMENT '备注',
FOREIGN KEY (uid) REFERENCES user(uid) ON DELETE RESTRICT ON UPDATE CASCADE
) COMMENT '订单表';
```

表9.订单详情表 (orderItem)

```
CREATE TABLE orderItem (
id INT PRIMARY KEY NOT NULL AUTO_INCREMENT COMMENT '主键',
oid INT NOT NULL COMMENT '订单ID',
pid INT NOT NULL COMMENT '商品编号',
name VARCHAR(100) NOT NULL COMMENT '商品名称',
price DECIMAL(10, 2) NOT NULL COMMENT '商品单价',
quantity INT NOT NULL DEFAULT 1 COMMENT '购买数量',
bid INT NOT NULL COMMENT '商家ID',
FOREIGN KEY (oid) REFERENCES ordered(oid) ON DELETE CASCADE,
FOREIGN KEY (pid) REFERENCES product(pid) ON DELETE RESTRICT,
FOREIGN KEY (bid) REFERENCES business(bid) ON DELETE RESTRICT
) COMMENT '订单详情表';
```

其他

```
-- 1.在订单表中增加省份
ALTER TABLE ordered
ADD COLUMN province VARCHAR(50) DEFAULT NULL
COMMENT '省份';

-- 2.在订单详情表中增加小计金额
ALTER TABLE orderItem
ADD COLUMN subtotal DECIMAL(12, 2) GENERATED ALWAYS AS (price * quantity) STORED
COMMENT '小计金额';

-- 3. 修改评论表（feedback）
-- 修改feed_time字段类型为DATETIME，并设置默认值
ALTER TABLE feedback
MODIFY COLUMN feed_time DATETIME DEFAULT CURRENT_TIMESTAMP COMMENT '评论时间';

-- 4. 改进身份表
ALTER TABLE identity
MODIFY COLUMN student BOOLEAN DEFAULT FALSE COMMENT '是否学生：true学生，false普通';

-- 5. 修改订单表（ordered）
-- 将order_time字段设置为默认当前时间
```

```
ALTER TABLE ordered
MODIFY COLUMN order_time DATETIME DEFAULT CURRENT_TIMESTAMP COMMENT '订单时间';

-- 6. 完善购物车表，添加必要字段
-- 6.1 先清空cart表中的数据
TRUNCATE TABLE cart;

-- 6.2. 然后再执行ALTER TABLE添加字段和外键
ALTER TABLE cart
ADD COLUMN quantity INT NOT NULL DEFAULT 1 COMMENT '购买数量',
ADD COLUMN bid INT NOT NULL COMMENT '商家ID',
ADD FOREIGN KEY (bid) REFERENCES business(bid);

--7.确保购物车表有唯一约束，避免同一用户重复添加同一商品
ALTER TABLE cart ADD UNIQUE INDEX idx_uid_pid (uid, pid);
```

表10 新增短信验证码的表，用于存储和验证短信验证码

CREATE TABLE `sms_verification` (
`id` int NOT NULL AUTO_INCREMENT COMMENT '主键ID',
`phone` varchar(20) NOT NULL COMMENT '手机号',
`code` varchar(6) NOT NULL COMMENT '验证码',
`send_time` datetime NOT NULL COMMENT '发送时间',
`expire_time` datetime NOT NULL COMMENT '过期时间',
`used` tinyint(1) NOT NULL DEFAULT '0' COMMENT '是否使用（0未使用，1已使用）',
`verify_time` datetime DEFAULT NULL COMMENT '验证时间',
`purpose` varchar(20) DEFAULT 'register' COMMENT '用途：register登录register/找回密码forgot/修改手机update',
`ip_address` varchar(50) DEFAULT NULL COMMENT '请求IP地址',
`created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
PRIMARY KEY (`id`),
KEY `idx_phone` (`phone`),
KEY `idx_phone_used` (`phone`, `used`),
KEY `idx_expire_time` (`expire_time`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='短信验证码表';

表11 创建订单物流信息表（不含订单状态和备注字段）

CREATE TABLE order_logistics (
id INT PRIMARY KEY AUTO_INCREMENT COMMENT '物流信息ID',
order_id INT NOT NULL COMMENT '订单ID（外键，关联ordered表的oid）',
logistics_company VARCHAR(100) NOT NULL COMMENT '物流公司',
tracking_number VARCHAR(50) NOT NULL COMMENT '运单号',
created_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
updated_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',

```
-- 添加外键约束，关联ordered表的oid列
CONSTRAINT fk_order_logistics_order_id
    FOREIGN KEY (order_id)
```

```
      REFERENCES ordered(oid)
      ON DELETE CASCADE
      ON UPDATE CASCADE,

  -- 添加唯一约束，确保一个订单只有一个物流记录
  CONSTRAINT uk_order_id UNIQUE (order_id),

  -- 添加索引
  INDEX idx_tracking_number (tracking_number),
  INDEX idx_logistics_company (logistics_company),
  INDEX idx_created_time (created_time)
```

) COMMENT '订单物流信息表';

表12 学生认证

CREATE TABLE `student_verification` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`uid` int(11) NOT NULL,
`school` varchar(255) DEFAULT NULL,
`student_id` varchar(100) DEFAULT NULL,
`card_image` LONGTEXT DEFAULT NULL,
`status` int(11) DEFAULT '0' COMMENT '0: Pending, 1: Approved, 2: Rejected',
`create_time` datetime DEFAULT NULL,
`update_time` datetime DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

触发器

-- 首先检查是否已存在同名触发器，先删除
DROP TRIGGER IF EXISTS after_user_insert;

-- 创建触发器：用户注册后自动创建identity记录
DELIMITER $$

CREATE TRIGGER after_user_insert
AFTER INSERT ON user
FOR EACH ROW
BEGIN
-- 为新用户创建默认identity记录
INSERT INTO identity (uid, level, student, points)
VALUES (NEW.uid, 1, FALSE, 0); -- level=1, student=0(false), points=0
END$$

DELIMITER ;

-- 测试触发器
-- 插入一个新用户
INSERT INTO user (username, password, email, phone, status, role, avatar)
VALUES ('trigger_test', 'test123456', 'test@example.com', '13800138111', 1, 'user', 'default_avatar.jpg');

```
-- 检查是否自动创建了identity记录
SELECT
u.uid,
u.username,
u.create_time,
i.iid,
i.level,
i.student,
i.points,
i.uid as identity_uid
FROM user u
LEFT JOIN identity i ON u.uid = i.uid
WHERE u.username = 'trigger_test'
ORDER BY u.create_time DESC
LIMIT 5;

-- 也可以查看最新的用户和他们的identity
SELECT
u.uid,
u.username,
u.email,
i.level,
CASE i.student
WHEN TRUE THEN '是'
ELSE '否'
END as 是否学生,
i.points as 积分
FROM user u
LEFT JOIN identity i ON u.uid = i.uid
ORDER BY u.uid DESC
LIMIT 10;
```