目录

目录

文档作者

文档修改历史

1.引言

1.1 编制目的

1.2 词汇表

1.3 参考资料

2.产品概述

3.逻辑视角

4.组合视角

4.1 开发包图

4.2 运行时进程

4.3 物理部署

5.**接口视角**

5.1 模块的职责

5.2 用户界面层的分解

5.2.1 职责

5.2.2 接口规范

5.3 业务逻辑层的分解

5.3.1 职责

5.3.2 接口规范

5.4 数据层的分解

5.4.1 职责

5.4.2 接口规范

6.信息视角

6.1 描述数据持久化对象(PO)

6.2 数据库表

文档作者

主要编写者: 李祥郡

其他编写者: 戚俊涵, 张铭恺, 易飞翔

文档修改历史

修改人员	日期	修改原因	版本号
李祥郡	4.20	初步创建体系架构	1.0.0
李祥郡	4.27	更新自选需求部分的信息	1.1.0
李祥郡	5.12	更新更多自选需求部分的信息	1.2.0
李祥郡	5.30	最终更新	2.0.0

1.引言

1.1 编制目的

本报告详细完成对 **SBEAM系统** 的概要设计,达到指导详细设计和开发的目的,同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写,是了解系统的导航。

1.2 词汇表

词汇名称	词汇含义	备注
РО	持久化对象(Persistent Object)	数据层对象名
VO	可视化对象(View Object)	数据层对象名
DAO	数据访问对象(Data Access Object)	

1.3 参考资料

软件架构的 "4+1" 视图 (附视图案例) 4+1视图模型案例-CSDN博客

什么是Dao层、Entity层、Service层、Servlet层、Utils层? -CSDN博客

计算机系统: 进程图解析, 全面详细! -CSDN博客

16. 面向对象——UML包图 (Package Diagram) - 知平

UML部署图详解-CSDN博客

2.产品概述

├ 1. 游戏浏览与详情

│ ├─ 分类检索、关键词搜索 │ ├─ 游戏封面、截图、视频预览

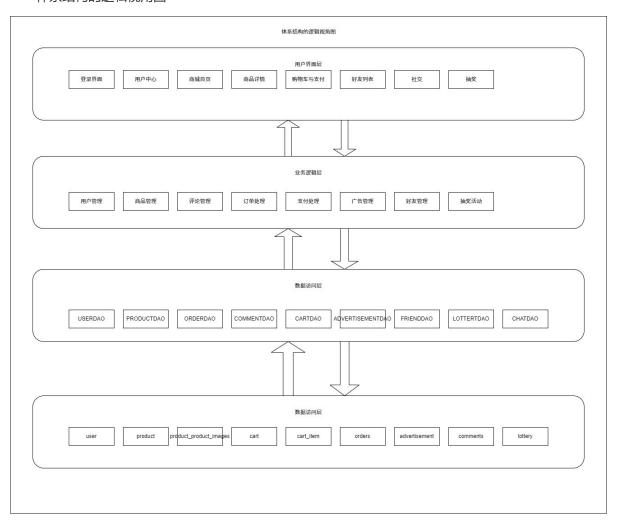
SBEAM 游戏商城 — 一、项目背景与目标 ├─ 1. 市场背景: │ └─ 社交化、互动化的游戏购买体验成为差异化竞争点 └ 2. 项目目标: |-- 打造集游戏浏览、购买、评论、抽奖、社交为一体的一站式平台 └─ 提升用户活跃度与粘性,实现平台与厂商多方共赢 —二、系统定位与价值 ├─ 定位:面向广大电竞和休闲玩家的综合数字游戏商城 └─ 核心价值: |-- 丰富的游戏内容与详细的评测资料 ├─ 安全便捷的支付与购买流程 ├─ 多样化的互动机制(评论、抽奖、社交) └─ 实时聊天增强用户社区氛围 — 三、核心功能概览

│ └ 评分与用户评论展示 ├ 2. 评论系统 ├─ 发布、删除与回复评论 │ └─ 评论点赞与举报 |-- 3. 抽奖活动 ├─ 多种抽奖机制 (抽取余额) |-- 活动页面与奖品池管理 └─ 抽奖结果通知与领奖流程 ├─ 4. 购买与支付 ├─ 购物车与订单管理 ├─ 多渠道支付(支付宝) └─ 订单状态跟踪与售后支持 └ 5. 好友社交 |-- 好友申请与管理 ├─ 私信与群聊 └ 在线状态与消息推送 └─ 四、主要用户角色 ├─ 游客: 浏览游戏、查看信息 |-- 注册用户: 评论、抽奖、购买、添加好友、聊天

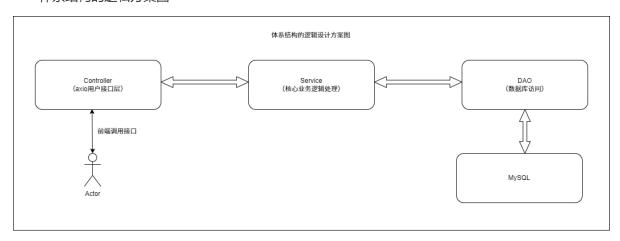
└─ 管理员: 活动及奖品管理、用户与内容审核

3.逻辑视角

体系结构的逻辑视角图



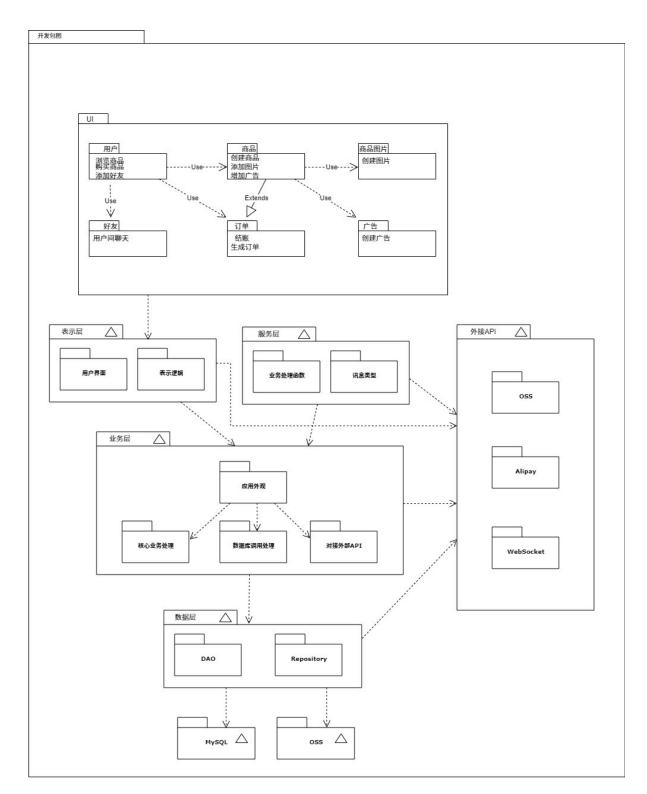
体系结构的逻辑方案图



4.组合视角

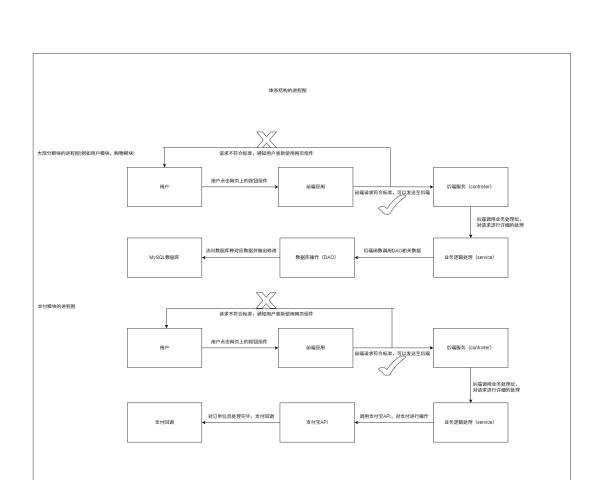
4.1 开发包图

体系结构的开发包图



4.2 运行时进程

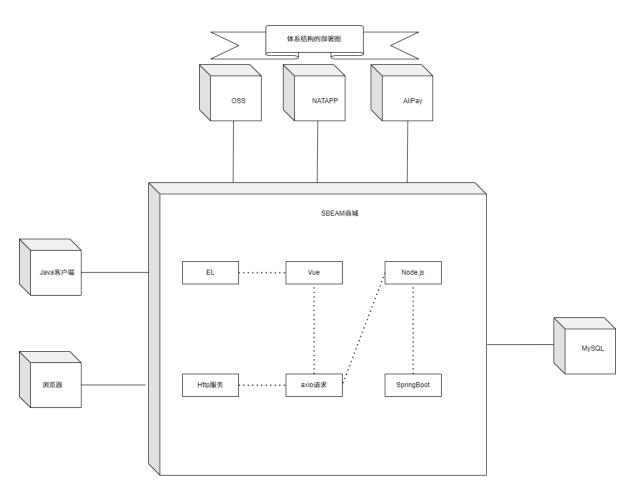
体系结构的进程图



4.3 物理部署

体系结构的部署图

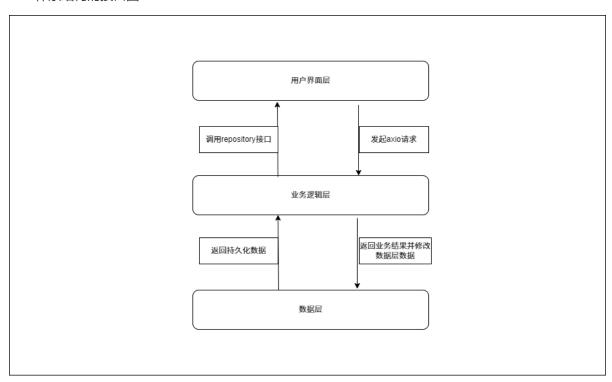
]



5.接口视角

5.1 模块的职责

体系结构的接口图



5.2 用户界面层的分解

5.2.1 职责

- 1. 提供前端组件用于展示商品,订单,支付,抽奖,好友等等。
- 2. 发起http请求,与后端进行交互
- 3. 输入数据校验
- 4. 提供操作反馈信息

5.2.2 接口规范

機口名称	方法	新 校	18 F F M	建学 体	100
MPEM	POST	/api/users/register	8	(drinteger, name: String, phone: String, password: String, storeid integer, address: String, role RoleEnum, createTime: Date; balance: Double, austar: String)	("code"000), "message" "mail", "data" () d integer, name String, phone String, password String, stored dinteger, address String, role Role Enum, create/Time:Data, balance:Double, avatar:String)
用户型装	POST	/api/users/login	phone?,password?	R	("code":000,"musuage":null,"data":)token:Soring()
研究性中国教	GET	/api/users	20	R .	("code":000,"message":"null","data")/dxtraeger,nume:String,phone:String,password:String,atoreid:integer,address:String/role:RoleEnum,create/Eme:Data/plaince:Double,avatar:String)
見知用户信息	POST	/api/users/(userid)	20	R	(*code*1003,*message*1*nulf*,*data*1*nuef*)
198AS	POST	/api/product/createProduct	8	$\{productid integer, movel d integer, product Name String, product Name$	("code"000,"message"muli,"data":"mue")
BHAS	27330	/api/product/deleteProduct/(productid)	R	R	("code*1000,"message*muli "data***mue")
GREEN	GET	/api/product	R	R	["code":500,"message":mail,"data": iproductid integer starekt integer productName-String product Amount integer, productPrice Double productDescription String productImages; List productLogs String productDescription Double)
税常用品	GET	/api/product/search/[productid)	n	克	("code":003," message" mult, "data": - dproducted treeger, product bit reseger, product Name String, product Amount integer, product Price Double, product Description String, product images List, product Logo String, product Discount Double (i)
PERSON	PUT	/api/product/productDetail/(productid)	20	$\{ product d integer, moreld integer, product Name String, product Amount integer, product Product Double product Description String, product Images, List, product Logo, String, product Double \}$	("code":000,"message":null("data":"true")
补充高品库存	PATCH	/api/product/productDetail/(productid)	productAmount?	R	("code":000,"message":null("data":"mve")
門際和物本型是 四有某件用品	POST	/api/product/productDetall/(productid)/(cartid)	я	π	("code":000,"message":null("data":"brus")
見が取得を取役 再級(包括報 前)	PUT	/api/product/productDetall/(productid)/cart	productAmount?	(sarridinteger, world integer, cardemics (air)	("code":001,"message":nul("desa":"nue")
所有用的 所有用品	GET	/api/user/juseridj/cart	R	R	("code" 000, "message" mult, "data" (continential reager product duringer product Quantity integer product Price Double))
987988	PUT	/api/user/juseridy/cart	8	(cartidinteger,useridinteger,cartternsClat)	("code":000,"message":null, "data": Jorderid integer, userid integer, ptalPrice Double, psyMethod:String, orderStatus; OrderStatus; createTime:Date, putTradeNo; String())
391	GET	/api/alipay/pay	orderid?	R	void
351894	POST	/api/alipay/return	8	R	String
研究は単位者	GET	/api/ordens/jordenskl)	8	R	("code"000,"message"null,"data"-jorderStatucorderStatus)}
上和銀件	POST	/api/tools/mages	file?	R	(*code*000),*mercuge*znull,*data*/(String))
19875	POST	/api/advertisement	8	(advertisement) distager, advertisement) tile String, advertisement Content String, advertisementimagel/H String, product/distager)	("code*1000)"message*nul("data**fmue")
MRFTS	DELETE	/api/advertisement	advertisementid?	R	("code":000,"message":nul("data":"mue")
REMAILS	GET	/api/advertisement	8	R	("code"000,"message"null, "data"/advertisementid:integer,advertisementTeleString,advertisementConsectString,advertisementImageUntString.productid:integer()
見終广告	PUT	/api/advertisement/(advertisementid)	20	(advertisement) dinteger, advertisement Citie String, advertisement Content String, advertisement image Lint String, product id integer)	("code"000,"message"muli "data":"mue")

5.3 业务逻辑层的分解

5.3.1 职责

业务逻辑层负责处理核心业务流程、验证、权限校验、事务管理、调用数据层接口并返回结果。

- 实现应用核心流程,如注册、登录、下单、支付。
- 用户权限校验(基于 JWT 或会话)。
- 事务管理(下单与扣库存需保证原子性)。
- 调用数据层接口并组合业务结果。
- 异常处理与日志记录。

5.3.2 接口规范

```
public interface UserService {
    Boolean register(UservO uservO);
    String login(String phone,String password);
    UservO getInformation();
    Boolean updateInformation(int userId,UservO uservO);
    void depositBalance(Integer userId, double amount);
}

public interface ProductService {
    Boolean createProduct(ProductvO productvO);
    List<ProductvO> getAllProductsByStoreId(Integer storeId);
    List<ProductvO> getAllProducts();
    ProductvO getProductByProductId(Integer productId);
    Boolean deleteProduct(Integer productId);
```

```
Boolean updateProduct(Integer productId, ProductVO productVO);
    Boolean updateProductAmount(Integer productId,Integer amount);
    Boolean updateProductDiscount(Integer productId,Double productDiscount);
}
public interface CartService {
    Boolean createCart(CartVO cartVO);
    CartVO getCart(int userId);
    Boolean judgeCartItemExist(Integer productId,Integer cartId);
    Boolean updateCart(int productId,int numberOfProduct,CartVO cartVO);
    OrdersVO settleAccount(int userId, CartVO cartVO);
    List<CartItemVO> showAllCartItems(int userId);
    int getCartItemCount(int cartId);
public interface OrdersService {
    Boolean createOrders(OrdersVO ordersVO);
    List<OrdersVO> getAllOrders();
    OrdersVO getOrdersById(int orderId);
    Boolean deleteOrdersByOrderId(int orderId);
    OrderStatus getOrderStatus(int ordersId);
}
public interface ImageService {
    String upload(MultipartFile file);
public interface AdvertisementService {
    public Boolean createAdvertisement(AdvertisementVO advertisementVO);
    public Boolean deleteAdvertisement(Integer advertisementId);
    public Boolean updateAdvertisement(int advertisementId,AdvertisementVO
advertisementVO);
    public List<AdvertisementVO> showAllAdvertisements();
    public AdvertisementVO getAdvertisementByAdvertisementId(Integer
advertisementId);
}
```

5.4 数据层的分解

5.4.1 职责

与数据库交互,实现 CRUD 操作并提供数据持久化服务.

- 定义实体模型(使用 TypeORM 或 Sequelize)。
- 提供 CRUD 接口与数据库交互。
- 支持分页查询与条件筛选。
- 保证数据一致性与性能优化。

5.4.2 接口规范

```
public interface UserRepository extends JpaRepository<User, Integer> {
    User findByPhone(String phone);
    User findByPhoneAndPassword(String phone, String password);
    Optional<User> findById(Integer Id);
}
public interface ProductRepository extends JpaRepository<Product, Integer> {
    Product findByProductId(Integer productId);
```

```
List<Product> findAllByStoreId(Integer storeId);
    Product findByProductName(String productName);
}

public interface CartRepository extends JpaRepository<Cart, Integer> {
    Cart findByCartId(Integer cartId);
    Cart findByUserId(Integer userId);
}

public interface OrdersRepository extends JpaRepository<Orders, Integer> {
    List<Orders> findAll();
    Orders findByOrdersId(Integer OrderId);
    Orders findByOutTradeNo(String outTradeNo);
}

public interface AdvertisementRepository extends JpaRepository<Advertisement,
Integer> {
    Advertisement findByAdvertisementId(int id);
}
```

6.信息视角

6.1 描述数据持久化对象(PO)

```
@Getter
@Setter
@NoArgsConstructor
@Entity
public class User {
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private Integer id;
    @Basic
    @Column(name = "name")
    private String name;
    @Basic
    @Column(name = "phone")
    private String phone;
    @Basic
    @Column(name = "password")
    private String password;
    @Basic
    @Column(name = "create_time")
    private Date createTime;
    @Basic
    @Column(name = "store_id")
    private Integer storeId;
    @Column(name = "address")
```

```
private String address;
    @Basic
    @Column(name = "balance")
    private Double balance= (double) 0;
    @Basic
    @Column(name = "role")
    @Enumerated(EnumType.STRING)
    private RoleEnum role;
    @Basic
    @Column(name="avatar")
    private String avatar;
}
@Getter
@Setter
@NoArgsConstructor
@Entity
@Table(name = "product")
public class Product {
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "product_id")
    private Integer productId;
    @Basic
    @Column(name = "store_id")
    private Integer storeId;
    @Basic
    @Column(name = "product_name")
    private String productName;
    @Basic
    @Column(name = "product_amount")
    private Integer productAmount;
    @Basic
    @Column(name = "product_price")
    private Double productPrice;
    @Basic
    @Column(name = "product_discount")
    private Double productDiscount;
    @Basic
    @Column(name = "product_description", columnDefinition = "TEXT")
    private String productDescription;
    @Basic
    @Column(name = "product_logo")
    private String productLogo;
    @ElementCollection
```

```
@Column(name = "product_images")
    private List<String> productImages;
}
@Getter
@Setter
@NoArgsConstructor
@Entity
@Table(name = "cart")
public class Cart {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer cartId;
    @Column(nullable = false)
    private Integer userId;
    @OneToMany(mappedBy = "cart", cascade = CascadeType.ALL, orphanRemoval =
true)
    @Column(name="cart_item")
    private List<CartItem> cartItems;
}
@Getter
@Setter
@NoArgsConstructor
@Entity
@Table(name = "cart_item")
public class CartItem {
   @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer cartItemId;
    @ManyToOne
    @JoinColumn(name = "cart_id", nullable = false)
    @JsonIgnore
    private Cart cart;
    @Column(name = "product_id", nullable = false)
    private Integer productId;
    @Column(name = "product_quantity", nullable = false)
    private Integer productQuantity;
    @Column(name = "product_price", nullable = false)
    private Double productPrice;
}
@Getter
@Setter
@NoArgsConstructor
@Entity
@Table(name = "orders")
public class Orders {
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Id
    @Column(name = "orders_id")
    private Integer ordersId;
```

```
@Basic
    @Column(name = "user_id")
    private Integer userId;
    @Basic
    @Column(name = "total_price")
    private Double totalPrice;
    @Basic
    @Column(name = "pay_method")
    private String payMethod;
    @Basic
    @Column(name = "order_status")
    @Enumerated(EnumType.STRING)
    private OrderStatus orderStatus;
    @Basic
    @Column(name = "create_time")
    private Date createTime;
    @Basic
    @Column(name = "out_trade_no")
    private String outTradeNo;
}
@Entity
@Table(name = "advertisements")
@Setter
@Getter
public class Advertisement {
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "advertisement_id", nullable = false, columnDefinition = "INT
COMMENT '广告id'")
    private Integer advertisementId;
    @Column(name = "advertisement_title", nullable = false, length = 50,
columnDefinition = "VARCHAR(50) COMMENT '广告标题,不允许为空'")
    private String advertisementTitle;
    @Column(name = "advertisement_content", nullable = false, length = 500,
columnDefinition = "VARCHAR(500) COMMENT '广告内容'")
    private String advertisementContent;
    @Column(name = "advertisement_image_url", nullable = false, length = 500,
columnDefinition = "VARCHAR(500) COMMENT '广告图片url'")
    private String advertisementImageUrl;
    @Column(name="product_id", nullable = false)
    private Integer productId;
}
@Data
@Getter
@Setter
public class AliPay {
   private String traceNo;
```

```
private double totalAmount;
private String subject;
private String alipayTraceNo;
}
```

6.2 数据库表

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
user	id	int	NO	PRI	NULL	auto_increment
user	address	varchar(255)	YES		NULL	
user	avatar	varchar(255)	YES		NULL	
user	balance	double	YES		NULL	
user	create_time	datetime(6)	YES		NULL	
user	lottery_chances	int	YES		NULL	
user	name	varchar(255)	YES		NULL	
user	password	varchar(255)	YES		NULL	
user	phone	varchar(255)	YES		NULL	
user	role	varchar(255)	YES		NULL	
user	store_id	int	YES		NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外 信息
user_already_friends	user_id	int	NO	MUL	NULL	
user_already_friends	already_friend_id	int	NO	MUL	NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外 信息
user_prepared_friends	user_id	int	NO	MUL	NULL	
user_prepared_friends	prepared_friend_id	int	NO	MUL	NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
product	product_id	int	NO	PRI	NULL	auto_increment
product	product_amount	int	YES		NULL	
product	product_description	text	YES		NULL	
product	product_discount	double	YES		NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
product	product_logo	varchar(255)	YES		NULL	
product	product_name	varchar(255)	YES		NULL	
product	product_price	double	YES		NULL	
product	store_id	int	YES		NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外 信息
product_product_images	product_product_id	int	NO	MUL	NULL	
product_product_images	product_images	varchar(255)	YES		NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
orders	orders_id	int	NO	PRI	NULL	auto_increment
orders	create_time	datetime(6)	YES		NULL	
orders	order_status	varchar(255)	YES		NULL	
orders	out_trade_no	varchar(255)	YES		NULL	
orders	pay_method	varchar(255)	YES		NULL	
orders	total_price	double	YES		NULL	
orders	user_id	int	YES		NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
comment	comment_id	int	NO	PRI	NULL	auto_increment
comment	comment_score	bit(1)	YES		NULL	
comment	comment_send_time	datetime(6)	YES		NULL	
comment	comment_text	text	YES		NULL	
comment	user_id	int	YES		NULL	
comment	product_id	int	NO	MUL	NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
inventory	cart_item_id	int	NO	PRI	NULL	auto_increment
inventory	product_id	int	NO		NULL	
inventory	product_price	double	NO		NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
inventory	product_quantity	int	NO		NULL	
inventory	user_id	int	NO	MUL	NULL	

表名	字段名	类型	可为 NULL	键	默认值	额外信息
cart	cart_id	int	NO	PRI	NULL	auto_increment
cart	user_id	int	NO		NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
cart_item	cart_item_id	int	NO	PRI	NULL	auto_increment
cart_item	product_id	int	NO		NULL	
cart_item	product_price	double	NO		NULL	
cart_item	product_quantity	int	NO		NULL	
cart_item	cart_id	int	NO	MUL	NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
advertisements	advertisement_id	int	NO	PRI	NULL	auto_increment
advertisements	advertisement_content	varchar(500)	NO		NULL	
advertisements	advertisement_image_url	varchar(500)	NO		NULL	
advertisements	advertisement_title	varchar(50)	NO		NULL	
advertisements	product_id	int	NO	MUL	NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
lottery	lottery_id	int	NO	PRI	NULL	auto_increment
lottery	lottery_name	varchar(255)	YES		NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
lottery_item	lottery_item_id	int	NO	PRI	NULL	auto_increment
lottery_item	lottery_item_probability	double	NO		NULL	
lottery_item	product_id	int	NO		NULL	
lottery_item	product_quantity	int	NO		NULL	
lottery_item	product_value	double	YES		NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
lottery_item	lottery_id	int	NO	MUL	NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
lottery_orders	lottery_order_id	int	NO	PRI	NULL	auto_increment
lottery_orders	create_time	datetime(6)	YES		NULL	
lottery_orders	lottery_chances_count	int	YES		NULL	
lottery_orders	order_status	varchar(255)	YES		NULL	
lottery_orders	out_trade_no	varchar(255)	YES		NULL	
lottery_orders	pay_method	varchar(255)	YES		NULL	
lottery_orders	total_price	double	YES		NULL	
lottery_orders	user_id	int	YES		NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
messages	message_id	int	NO	PRI	NULL	auto_increment
messages	message_content	text	NO		NULL	
messages	message_sent_time	datetime(6)	NO		NULL	
messages	receiver_id	int	NO	MUL	NULL	
messages	sender_id	int	NO	MUL	NULL	

表名	字段名	类型	可为 NULL	键	默认 值	额外信息
chat_history	id	bigint	NO	PRI	NULL	auto_increment
chat_history	ai_response	text	NO		NULL	
chat_history	timestamp	datetime(6)	NO		NULL	
chat_history	user_input	text	NO		NULL	