

课程设计说明书

课程名称: 数据库系统课程设计

学 院: 计算机科学与工程学院

专 业: 计算机科学与技术 班级: 2020 级 4 班

姓 名: 杨昊明 学号: 202001051126

姓 名: 王乾润 学号: 202001030924

姓 名: 何超 学号: 202001021107

山东科技大学

2022 年 12 月 20 日

指导教师对课程设计的评语

指导教师签字：_____

2022 年 月 日

课程设计报告

任务发出日期	2022-10-07	课程设计完成日期	2022-12-23
课程设计成绩			
<p>一. 课程设计目的</p> <p>数据库系统课程设计目的在于加深对关系数据库理论知识的理解，培养数据库管理系统的数据操纵能力，熟练掌握数据库前端开发工具，培养运用数据库技术解决实际问题的能力。</p> <p>二. 课程设计内容</p> <p>2.1 课程设计题目：<u>网络商城购物系统</u></p> <p>2.2.完成数据库设计的需求分析、概念结构设计、逻辑结构设计、物理结构设计等。</p> <p>2.3.根据所设计数据库，开发简单的数据库应用系统。</p> <p>三. 设计过程及结果</p> <p>3.1.需求分析</p> <p>数据库需求分析就是分析人员在调查现存系统基础上，对用户的数据需求进行分析和确认。</p> <p>3.1.1 应用背景分析</p> <p>随着互联网的普及和计算机技术的发展，电子商务得到了广泛的应用。网上购物作为电子商务的重要组成部分，已经被越来越多的人所青睐。网上购物以节省时间和精力、无纸化的科学管理而闻名，它是基于浏览器的交互式信息传递管理系统。为了消费者和商家的利益，各大购物中心已经建立了自己的网站，并开放了网上购物平台，开发面向购物中心的网上购物平台，使消费者能够到购物中心购买或网上购买，这有利于商品的普及和市场的开拓。</p> <p>在日常生产生活中网络超市购物日常有大量的数据需要进行处理。包括供应商的管理、消费者会员的管理、销售情况以及商品信息等等。面对如此大的信息量，就需要有相应的计算机管理系统来提高工作的效率和系统管理的安全性。通过这样的系统，网络超市购物就可以提高工作效率，减少人工操作的工作量，使数据安全可靠的记录在数据库中。</p> <p>现在我们小组准备建立网络超市购物系统。其目的是要把销售商的商品、会员的信息、会员购买商品的记录等基本信息进行管理。确保有足够的安全性。以便网上购物系统可以进行推</p>			

广，让更多的销售商和购物者认为网上购物系统非常实惠便利的。

3.1.2 数据结构分析

(1) 在注册会员信息管理部分。要求：

- a 可以查看注册会员信息。
- b. 可以对注册会员信息进行添加及删除的操作。

(2) 在商品信息管理部分。要求：

- a 可以浏览商品信息。
- b. 可以对商品信息进行维护, 包括添加及删除的操作。

(3) 在订单信息管理部分，要求：

- a 可以浏览订单信息。
- b. 可以对订单信息进行维护，包括添加及删除的操作。

(4) 在供应商信息管理部分。要求：

- a 可以浏览供应商信息
- b. 对供应商信息进行维护，包括添加及删除的操作。

由以上需求可以制订如下关系模式：

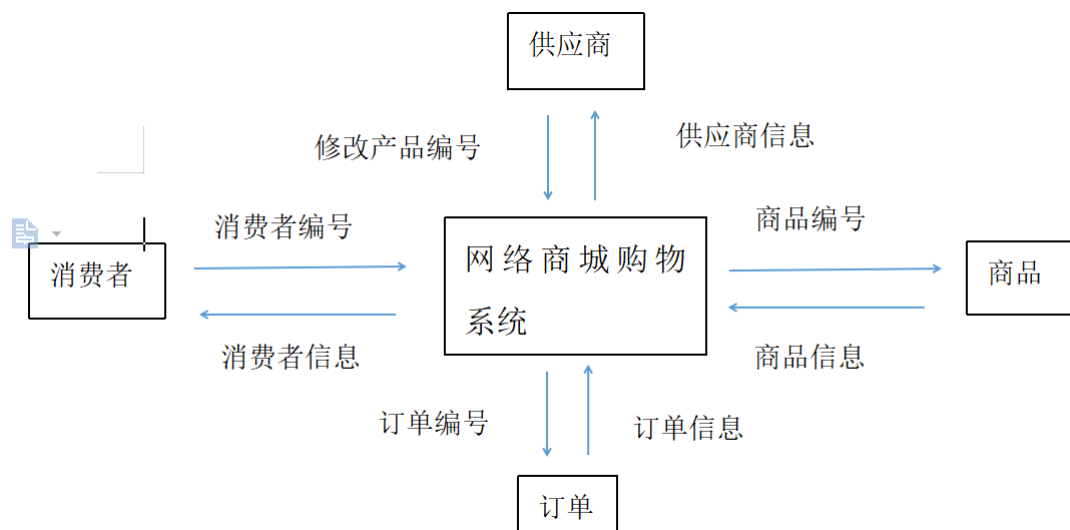
消费者 {消费者编号, 姓名, 性别, 出生日期, 昵称, 账号, 密码, 手机号, 消费额}

供应商 {供应商编号, 用户类型, 名称, 地址, 手机号, 评价}

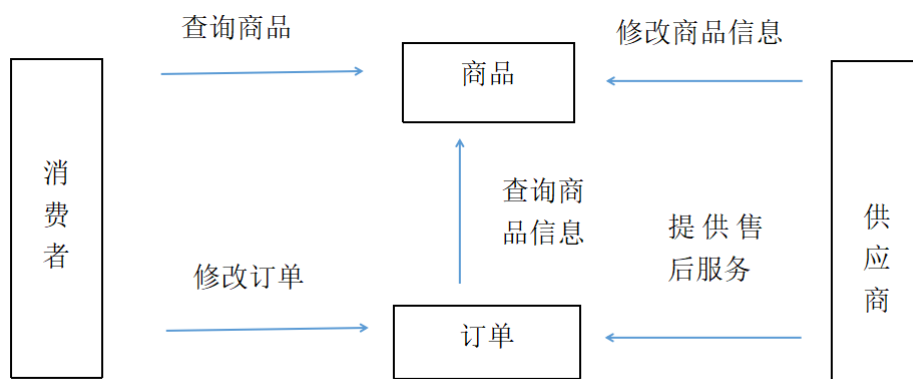
商品 {商品编号, 商品名称, 描述, 数量, 价格, 供应商编号, 上架时间, 评价}、

订单 {订单编号, 商品编号, 订单产生时间, 订单总价, 送货地址, 订单状态}

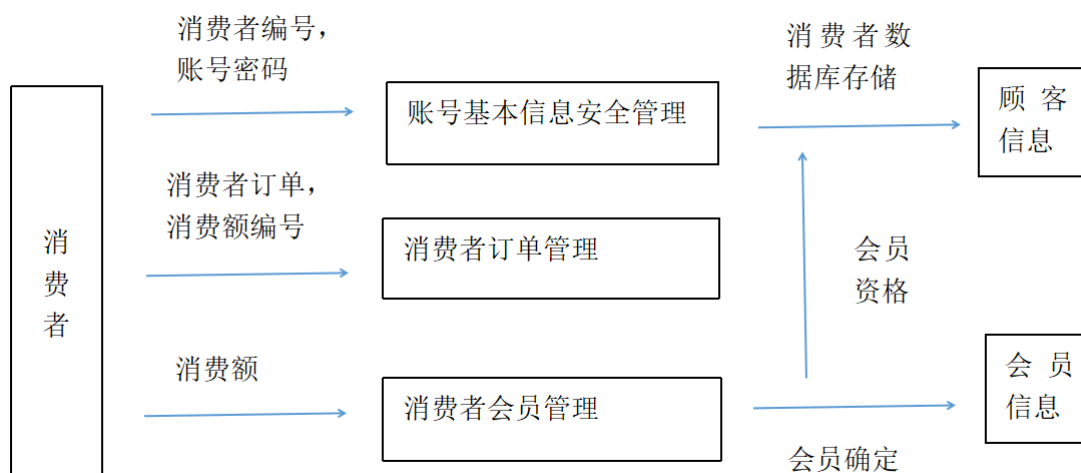
(5) 数据流图



(顶层数据流图)



(中层数据流图)



(消费者底层数据流图)

3.1.3 数据处理分析

在本系统中：消费者编号为消费者的主码；供应商编号为供应商的主码；商品编号为商品的主码；订单编号为订单的主码；在本系统中可以增加或者删除消费者编号，供应商编号，商品编号，订单编号，通过增加或者删除不同关系模式的主码，可以实现对不同关系模式数量的改动，例如消费者编号加 1，即代表消费者数量加一，增加了一个消费者的实体。而在本系统中进行查询时，也可以通过查找不同关系模式的主码来实现，在本系统中每个关系模式可以建立一个数据表，在不同的数据表中查找该表中的主码即可查找出其所对应的主体。主码是系统默认生成实现的，其他的实体属性如果需要修改，那么就可以操作数据库修改特定值，例如要在消费者的表中修改编号为 2 的消费者手机号和消费额，都可以进行修改，需要修改的属性即是每个实体的相应属性。表的列级即是即是不同关系模式的属性，一行代表记录一个主体。操作的频次要依照系统用户的使用量决定。

创建联动的触发器：1. 在订单完成时，修改消费者消费总额，商品数量；每一个订单完成时，消费者可能会增加或者减少商品，消费总额也可能会发生变化，就需要在变化的消费者实体上修改消费金额。2. 在删除消费者时，删除其所有订单；由于消费者实体被取消，消费者的消费金额也会为 0，需要删除所有订单。3、在消费者退款时，修改消费者消费总额；消费者进行退款，消费总额应该恢复到产品消费前的数值。4、在消费者消费总额达到某个值时修改消费者会员状态；商家可能会有某些促销活动，消费者消费达到某个数值的时候获得商家会员获取更多优惠，在消费者满足消费总额条件时及时进行会员状态修改。5、在商品数量为 0 时，无法下单。消费者的消费总额为 0，无法下单。

3.1.4 数据分析

a. 创建数据库时需要用户输入账号和密码，每个用户都必须要有唯一的账号和其相对于的密码，以便于用户的安全登录。

b. 用户角色可以操作的权限有订单表的增加与删除，消费者属性的更改，用户可以修改自己的姓名，性别，昵称，密码，手机号等消费者属性，但用户没有权限修改消费者权限和消费金额。用户也不可以修改订单属性，但可以增加或者删除订单，也可以查看订单属性。用户可以对商品和供应商进行评价，但无法对商品，供应商的其他属性进行修改。

c. 可以使用的数据加密存储技术：前置代理加密，在数据库之前增加一道安全代理服务，所有访问数据库的行为都必须经过该安全代理服务，在此服务中实现如数据加解密、存取控制等安全策略，安全代理服务通过数据库的访问接口实现数据存储。。应用端加密，该技术是应用系统通过加密 API 对敏感数据进行加密。文件系统加密，对数据存储的载体从操作系统或文

件系统层面进行加解密。

3.1.5 数据完整性分析

在网络购物平台的交易中，可能存在数据冗余的情况，也就是指数据库中可能存在一些重复的数据，导致数据库的数据完整性遭到了一定程度的破坏，而数据完整性代表了数据库中的数据能够正确反应实际情况。本网络购物平台项目数据库完整性通过数据库的多种完整性约束来保证。为了能够满足数据完整性，我们在设计数据库管理系统时根据网络购物系统的用户，商家，商品特性编写完整性约束条件以实现目的要求。在该网络购物系统中，存在许多的完整性细节要求，例如交易订单编号应能代表唯一的一笔订单，而不能对应多个。而商品类别，用户地区等数据则并不唯一，可作为多个用户或订单的共有数据，但应能被查询，例如向数据库请求某一类别商品的类别信息，应给出含有这一类别信息的所有商品类别。此外，还需要排除用户在使用数据库时向数据库中传入的一些不合语义的数据，或是间接产生的非法数据。例如当用户更换了用户账户的手机号码，而若此前手机号码作为主键，此时应主动处理这一情况，将手机号码数据库原号码对应数据与用户解除关联，以避免非法情况产生。或是当用户因网络原因提交了一些非法参数，比如-1 的数量，除男或女以外的性别信息等，数据库应不接受该数据以免发生程序错误。我们可以使用基于数据库管理系统的完整性控制机制来实现此类业务规则，易于定义且容易理解，并能降低应用程序的复杂性，提高应用程序的运行效率。同时，基于 DBMS 的完整性控制机制是集中管理的，这比应用程序更容易实现数据库的完整性。

3.1.6 性能需求分析

网络购物平台经常面临一些高并发的使用场景，例如商家放出的某些抢购，秒杀，等活动，当活动面向用户定时开放时，服务器会在极短的时间内收到大量用户的访问请求，此时将对数据库的各种资源在很短的时间开始进行高强度占用。而一旦服务器性能优化欠缺，就会造成服务异常，给商户带来损失，在用户使用时为商家和平台带来不良的印象。而若并发控制出现问题，则容易出现资源访问抢占，用户实拍数大于库存的情况，这就为商户带来了损失或不良的购物使用体验。且此类问题易成为网络购物平台被攻击的入口，存在安全隐患。数据库应避免出现此类问题，满足一定的性能要求，首先从逻辑层面上，应优化数据库数据查询逻辑，避免在基础操作上出现不必要的性能浪费，其次我们可以增加服务器数量，线性扩充系统性能。而此种扩充操作对系统架构设计有一定要求的。最主要的优化包括减少 IO 次数，降低 CPU 计算，减少 JOIN, OR, 类型转换和排序操作的使用。我们可以通过增加缓存数据库，增加数据库

索引，页面静态化，MySQL 主从读写分离，分表分库，以及负载均衡集群等操作对数据库的结构设计进行优化。除此之外，也可以按照数据范围，对数据库进行数据哈希的方式来进行水平扩展；

3.2.概念结构设计

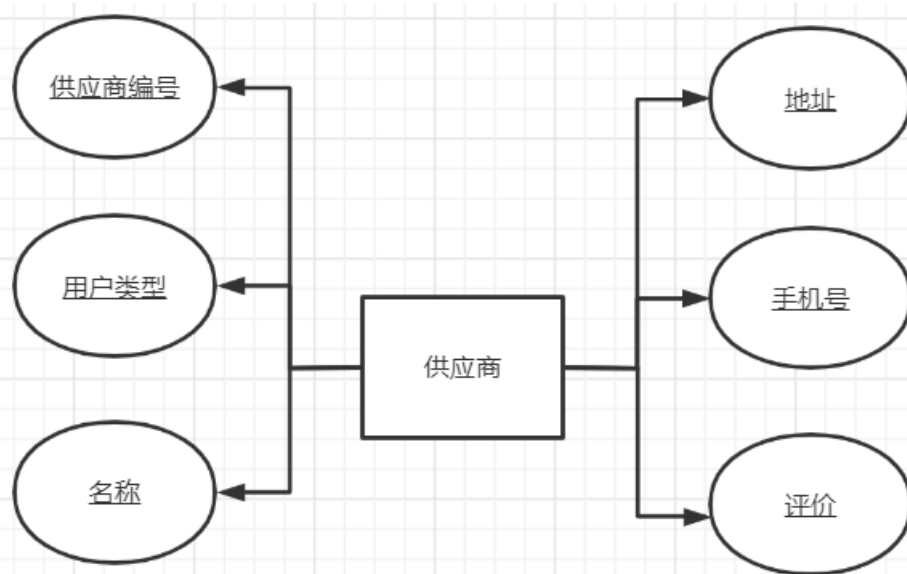
概念结构设计是将需求分析得到的用户需求抽象成信息世界的概念结构模型的过程。概念结构是各种数据模型的基础，它比数据模型更独立于机器，更加抽象更加稳定。概念结构设计是数据库设计的关键。它能够真实充分的反应现实世界，可扩充，易更改，易理解，并且易于向其他数据模型转换。

我们采用自底向上的概念结构设计方法，根据子需求获得部分概念视图，再逐步集成，成为一个全局概念视图。我们采集到关于供应商，消费者，职工，订单的相关信息，绘制出分 ER 图，值得注意的是供应商编号，消费者编号，职工编号，订单编号分别是这四个实体的主码，绘制出的分 ER 图如下所示。最终的 ER 图不仅要考虑到分 ER 图实体的属性，还要考虑他们彼此之间的相对关系。分 ER 图实体的属性都必须要在最终 ER 图中体现，而彼此之间的关系我们进行了如下分析：

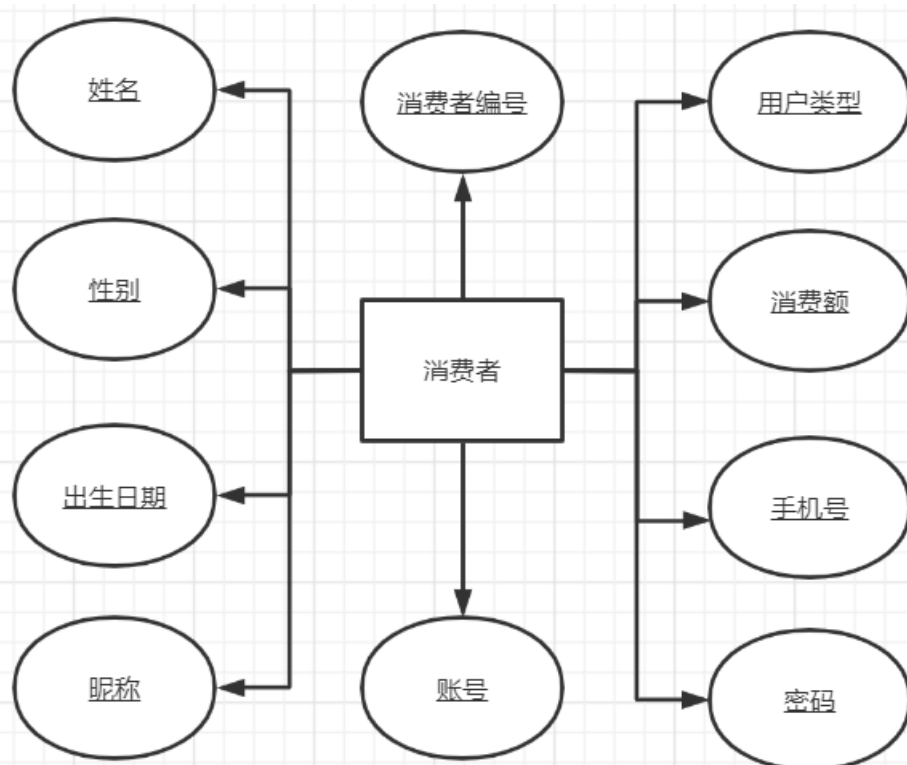
1. 用户存在用户类型属性，用户可能是 VIP 用户或者普通用户，那么就将有 VIP 用户编号或者普通用户编号。2. 用户账号对于用户昵称，应该都是唯一的，并且用户账号和密码对于用户的安全登录问题。3. 用户可以购买商品，并且用户评价构成商品的属性。4. 商品由供应商供应，并且用户也可以对供应商进行评价。5. 用户购买商品后就会产生订单，订单上面会包含商品，用户，供应商的部分信息，同时也会有类似送货地址，订单状态，订单产生时间等订单专属信息。

（分 ER 图和最终 ER 图如下所示）

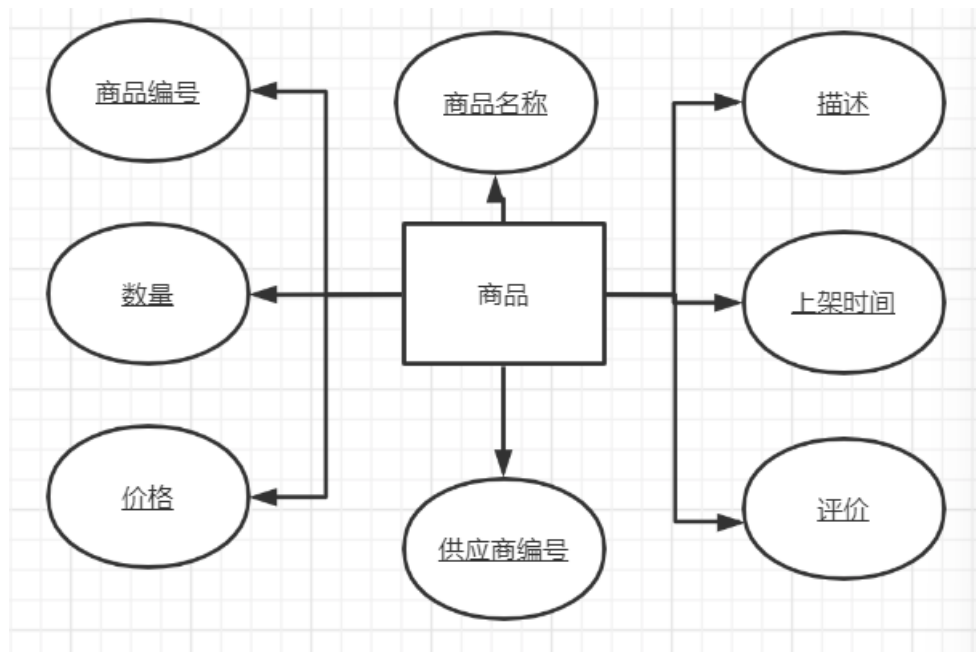
1.供应商实体图



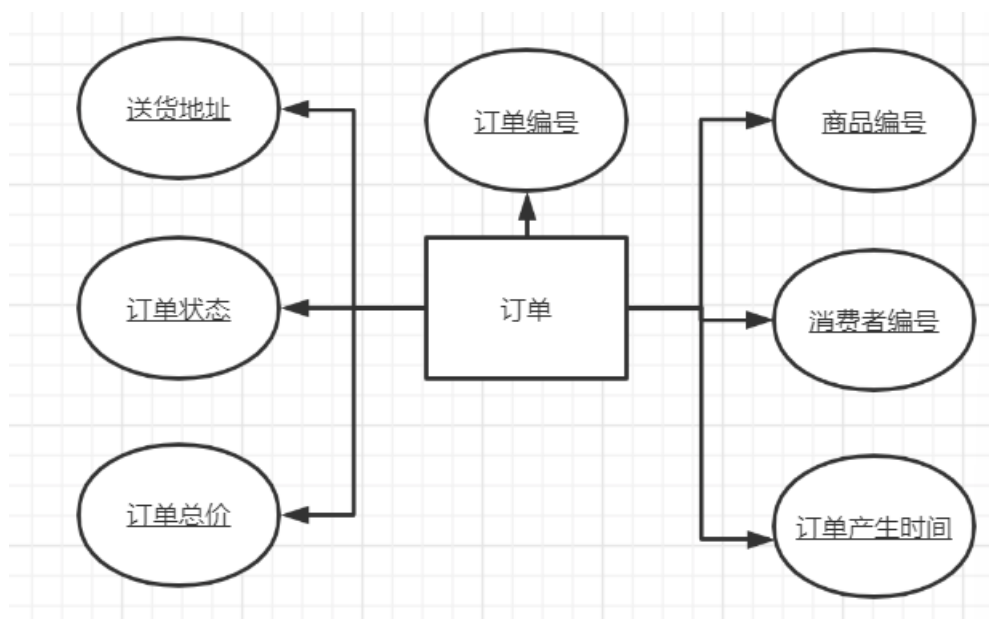
2.消费者实体图



3.职工实体图

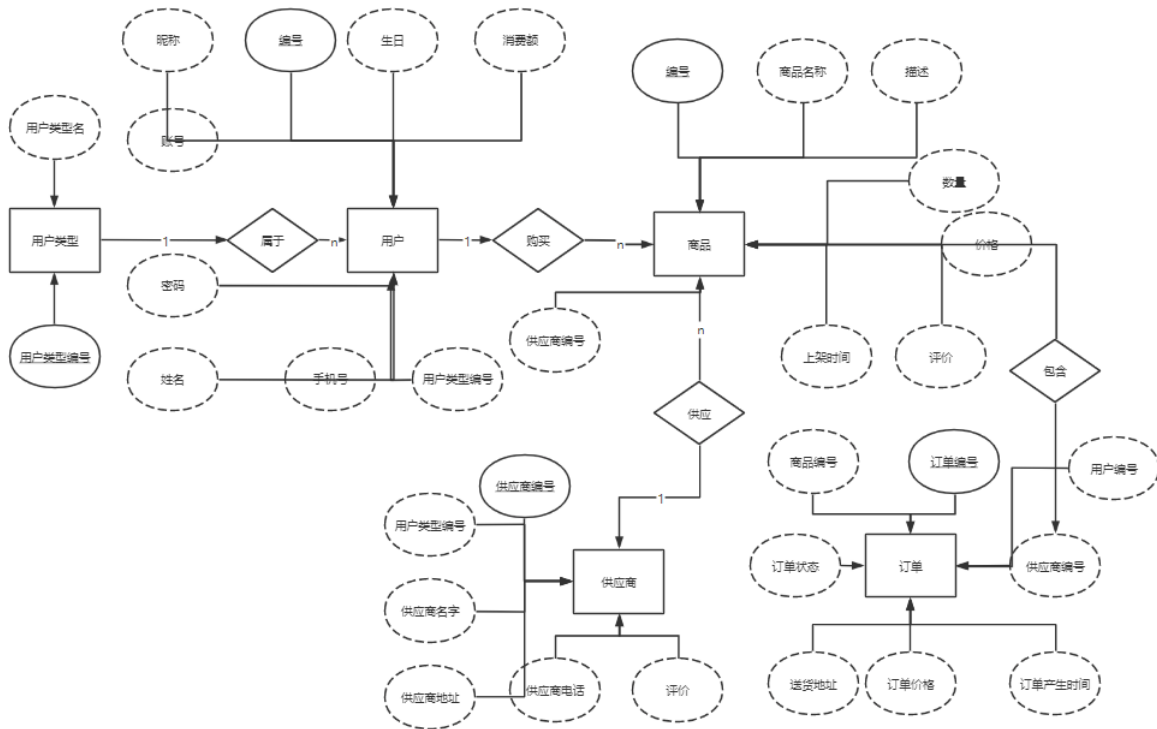


4. 订单实体图



5. 全局 E-R 图

山东科技大学学生课程设计



在如下表格中，我们生成了一些模拟随机数据，记录了供应商，消费者，职工，订单的相关信息，其中包括了大量的实体数据，在商品表格中，有着供应商编号，同时也有用户对商品的评价；同理，订单和供应商表格也有上文说明的各种关系。

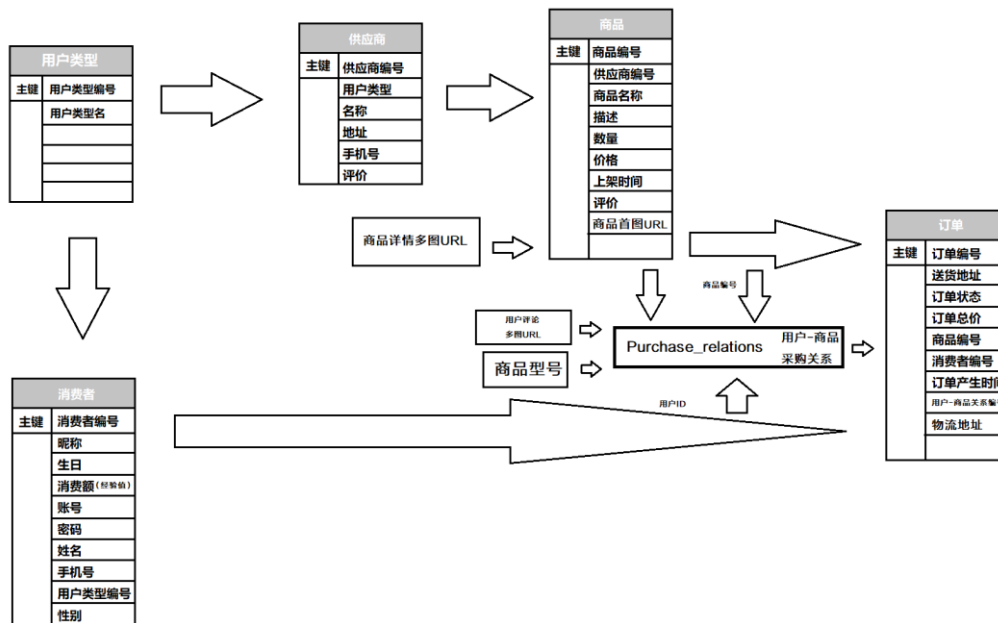
商品编号	商品名称	描述	数量	价格	供应商编号	上架时间	评价	订单编号	商品编号	订单产生时间	订单总价	送货地址	订单状态
85122968	IDO	GYHI	2581	834	19018	2020/1/27	SNLH	19494126	35196769	2020/1/1	7574	ESXS	1
54917060	IPX	RGBR	7975	188	39254	2020/1/10	RBEX	92706681	70571464	2020/1/8	3666	KYRW	1
95577357	UWR	IJHX	5905	972	91254	2020/1/8	XAGN	98810223	25483839	2020/1/18	8802	LAOY	1
9762232	TXG	KQPZ	7139	820	78616	2020/1/13	BMK	56843721	38116899	2020/1/29	7418	KYZH	1
70385729	TBV	HDAK	345	323	28401	2020/1/12	NPUW	52399727	67967426	2020/1/5	3888	KGLK	0
40535365	GMM	NUJM	19	495	39822	2020/1/16	TJPY	67719552	75415153	2020/1/30	7260	ONFK	1
28549742	ZEG	RSWC	7070	454	56576	2020/1/7	NRZJ	42788719	59224262	2020/1/30	2509	QLGH	0
94042286	MBS	LRIL	618	120	89023	2020/1/19	XEMB	61299003	42747491	2020/1/15	8931	CSNY	0
42563634	XRU	ITKR	8160	385	88821	2020/1/27	JXUX	87119973	30466110	2020/1/18	7422	JFXG	1
32214075	JZC	CAGK	3023	143	94070	2020/1/24	JTHC	44866800	47653272	2020/1/23	8618	CQNE	0
40553677	EYC	AQDG	8854	2	5807	2020/1/13	FXZO	90966474	73104270	2020/1/25	6490	KQQF	0
4005422	JSD	KHUY	5785	381	53757	2020/1/10	JZXD	94722250	14838451	2020/1/11	8305	BBKK	1
2058602	ISV	EJG	7756	409	58239	2020/1/19	KAFI	78670942	69833320	2020/1/6	4685	WHIJ	0
43861567	CJI	IVLZ	9585	974	70849	2020/1/20	XNNC	33348125	71071329	2020/1/4	7660	AERY	1
56155987	JFC	FVSQ	7391	772	72052	2020/1/9	MFAU	50879548	93802533	2020/1/5	9992	GXNQ	1
80233739	WAA	VJCJ	6076	422	22862	2020/1/19	NYWJ	92051240	25342105	2020/1/22	1305	PCKU	1
44152368	BTF	HKDE	5946	389	43549	2020/1/29	AMAZ	92063196	82783807	2020/1/27	8283	QWQO	1
89119162	KQZ	QJZG	2965	671	30350	2020/1/24	OGJH	77190276	96188564	2020/1/5	4939	ZCNF	1
84545942	PVS	GBRA	884	196	37570	2020/1/23	FDLX	21465597	64657718	2020/1/30	4214	KAMP	0
12998335	LWS	RMRF	6285	796	69323	2020/1/3	BIZT	79442109	53265769	2020/1/30	5173	SUDM	1
95439174	XNY	KIAY	2521	564	85779	2020/1/17	AUSJ	89625079	48243371	2020/1/1	1159	YEZI	1
73235514	JTE	MKQI	1475	129	52340	2020/1/20	QBBO	48286273	29484191	2020/1/17	4606	ZDOY	0
49786622	INN	PFKM	152	773	3561	2020/1/18	PWUD	99410951	47590616	2020/1/30	4693	EOXQ	1
69088072	BRE	RJW	6113	539	34630	2020/1/4	MHJM	60166026	67788596	2020/1/10	1575	VZAR	1
85936451	WLY	XWCK	9634	20	7752	2020/1/8	FQPP	21986424	41992250	2020/1/27	5401	YUEY	0
83809383	XFA	TGPK	9481	665	68147	2020/1/15	QPYP	51479832	87188066	2020/1/20	1657	VLJY	0
8677461	VMC	OKIX	6302	696	99153	2020/1/14	IRID	57299309	5331655	2020/1/28	8742	DXNF	1

山东科技大学学生课程设计

订单编号	商品编号	订单产生时间	订单总价	送货地址	订单状态	供应商编号	用户类型	名称	地址	手机号	评价
70003012	20577879	2020/1/2	637	PYZE	1	235773682	1	TR	WJM	13877350415	ASZTWW
99476050	27416665	2020/1/9	315	LNFL	1	235715109	9	BH	NRR	13952182670	WNHIRY
44362560	47647656	2020/1/6	1608	OCFA	1	235772926	2	JZ	LGR	13961938375	PFSBNJ
90168054	67807218	2020/1/21	5597	ARVH	0	235751653	9	MW	ZLU	13895705676	CWPJZN
69374608	3590639	2020/1/23	8394	CLXN	0	235709343	2	IE	VGY	13951015391	GHIKNM
86234507	97076093	2020/1/12	7532	INWR	0	235709983	2	GG	JKX	13928878926	ESEUSF
41746960	57123630	2020/1/29	1496	FUTM	1	235730854	1	QH	JII	13977764278	MTAGHE
67064447	20509743	2020/1/5	6663	CDRF	0	235796341	4	TQ	LCB	13999750913	SGZOHQ
79600342	77313306	2020/1/23	8788	PAPZ	1	235749287	9	GK	BGO	13970940734	YFQYDR
17260625	45642731	2020/1/6	9531	CYOE	1	235711971	0	VP	QWR	13925317739	NSCKJK
49942942	58054624	2020/1/8	3213	TNMX	0	235735488	9	ZR	RMM	13828471158	HORZSB
37812542	98682366	2020/1/14	4301	XAHH	1	235757188	6	AL	CWL	13939614577	QXMGOJ
32044530	34994563	2020/1/17	936	SBKO	1						
88735592	72295928	2020/1/5	6384	YJLB	1						
22159613	88635670	2020/1/26	817	FIFC	1						
23956123	55789178	2020/1/9	6334	XDFF	1						
13983892	86646145	2020/1/15	9751	QAOR	1						
88139027	56754415	2020/1/12	7963	PNSR	1						
45194380	11473153	2020/1/16	5173	JUKA	1						
99529131	83852659	2020/1/22	5756	WMFA	1						
48983974	63487210	2020/1/14	748	NTWO	1						
30457610	48265286	2020/1/5	3192	FGRN	1						
24075462	30485649	2020/1/28	3053	WHHC	0						
69714041	82211071	2020/1/29	727	ZQZC	0						
44401222	49796783	2020/1/22	3969	NEGY	0						
53162668	24183505	2020/1/25	5779	OQUY	1						
77286964	79613183	2020/1/16	6112	MYHK	0						
45001959	17154924	2020/1/21	7181	MEDY	1						

3.3 逻辑结构设计

逻辑结构设计



山东科技大学学生课程设计

将已绘制出来的 E-R 图转换成关系图 E-R 图向关系模型的转换要解决的问题是如何将实体型和实体间的联系转换为关系模式，如何确定这些关系模式的属性和码。

关系模型的逻辑结构是一组关系模式的集合。E-R 图则是有实体型、实体的属性和实体型之间的联系 3 个要素组成的。该数据库的关系模式主要有以下几个方面：

关系模型的逻辑结构是一组关系模式的集合。E-R 图则是有实体型、实体的属性和实体型之间的联系 3 个要素组成的。该数据库的关系模式主要有以下几个方面：

带下划线的属性为各关系的模式的主码：

消费者 {消费者编号, 姓名, 性别, 出生日期, 昵称, 账号, 密码, 手机号, 消费额}

供应商 {供应商编号, 用户类型, 名称, 地址, 手机号, 评价}

商品 {商品编号, 商品名称, 描述, 数量, 价格, 供应商编号, 上架时间, 评价}、

订单 {订单编号, 商品编号, 订单产生时间, 订单总价, 送货地址, 订单状态}

在以上关系中，实体中的每个属性都是简单属性，都是不可再分的，实体遵循第一范式。

对于消费者关系模式中存在的函数依赖有所有非主属性均依赖于消费者编号。

对于供应商关系模式中存在的函数依赖有所有非主属性均依赖于供应商编号。

对于商品关系模式中存在的函数依赖有所有非主属性均依赖于商品编号。

对于订单关系模式中存在的函数依赖有所有非主属性均依赖于商品编号。

上述关系模式都满足 1NF、2NF、3NF、BCNF。

3.4 物理结构设计

(以 MySQL 作为 DBMS，详细描述数据库设计、数据表创建、索引等的脚本和参数，使用有意义的数据表名称和字段名称，需实现主键、外键、以及完整性约束。)

1. 数据库设计：利用 mysql 软件创建数据库，建立新的数据库名称为 network_supermarket。在建立的数据库中创建数据表。并编写相应的功能函数。



2. 创建数据表，根据之前的 E-R 图，创建了上述的数据表，其中最重要的是 users, goods, order, purchase_relations, vendor 表，其主键分别为 constumer_id, goods_id, order_id, commodity_list_id, vendor_id。

(1) 顾客表：记录用户的基本信息，完整性约束条件先是对主键的设置，其次设置 consumer_id, consumer_name, gender, birthday, account, password 不能为空值。

(2) 货物表：对商品信息进行描述，其完整性约束包括主键设置，vendor_id 的外键设置 goods_id, goods_name, quantity, price, vendor_id, good_status 不能为空值。

(3) 订单表：描述产生的订单信息，其完整性约束包括主键设置，constumer_id, constumer_nickname 的外键设置，order_id, constumer_id, commodity_list_id, order_generation_time, total_order_price, order_status 设置为非空。

(4) 购买关系表：描述产品订单和相应产品之间的关系，需要设置所有属性为非空。

(5) 供应商表：记录供应商信息，vendor_name 需要设置为非空，没有外键。

山东科技大学学生课程设计

calhost_3306

JOYE

数据库

database1

information_schema

mysql

network_supermarket

表

cart

comments_media_url

goods

goods_media_url

goods_type

order

purchase_relations

users

vendor

视图

函数

f()

f()

p()

p()

p()

p()

对象

netw...

order @network_s...

purchase_relations ...

users @network_su...

vendor @network

新建

保存

另存为

添加栏位

插入栏位

删除栏位

主键

上移

下移

栏位

索引

外键

触发器

选项

注释

SQL 预览

名	类型	长度	小数点	不是 null	
consumer_id	bigint	0	0	<input checked="" type="checkbox"/>	2
consumer_name	varchar	40	0	<input checked="" type="checkbox"/>	
phone_number	bigint	0	0	<input checked="" type="checkbox"/>	1
gender	int	0	0	<input checked="" type="checkbox"/>	
birthday	date	0	0	<input checked="" type="checkbox"/>	
consumer_nickname	varchar	255	0	<input checked="" type="checkbox"/>	
password	varchar	255	0	<input checked="" type="checkbox"/>	
user_type	int	0	0	<input checked="" type="checkbox"/>	
costumer_icon_url	varchar	255	0	<input checked="" type="checkbox"/>	
exp	int	0	0	<input checked="" type="checkbox"/>	

(顾客表)

calhost_3306

JOYE

数据库

database1

information_schema

mysql

network_supermarket

表

cart

comments_media_url

goods

goods_media_url

goods_type

order

purchase_relations

users

vendor

视图

函数

f()

f()

p()

p()

p()

p()

对象

url ...

goods_type @netw...

order @network_s...

purchase_relations ...

users @network_su...

新建

保存

另存为

添加栏位

插入栏位

删除栏位

主键

上移

下移

栏位

索引

外键

触发器

选项

注释

SQL 预览

名	类型	长度	小数点	不是 null	
goods_id	bigint	0	0	<input checked="" type="checkbox"/>	1
goods_name	varchar	255	0	<input checked="" type="checkbox"/>	
goods_describe	varchar	255	0	<input type="checkbox"/>	
amount	decimal	10	2	<input checked="" type="checkbox"/>	
price	decimal	10	2	<input checked="" type="checkbox"/>	
vendor_id	bigint	0	0	<input type="checkbox"/>	
listing_time	timestamp	0	0	<input checked="" type="checkbox"/>	
goods_score	varchar	20	0	<input type="checkbox"/>	
goods_status	int	0	0	<input checked="" type="checkbox"/>	
unit	varchar	255	0	<input checked="" type="checkbox"/>	
head_photo	varchar	255	0	<input type="checkbox"/>	

(商品表)

calhost_3306

JOYE

机

database1

information_schema

mysql

network_supermarket

表

cart

comments_media_url

goods

goods_media_url

goods_type

order

purchase_relations

users

vendor

视图

函数

f()

BOOL检查商家账号密码 (

f()

BOOL检查用户账号密码 (

p()

cart-删除购物车的某个商品

p()

cart-使用购物车创建订单

p()

cart-修改购物车某商品的数

p()

cart-直接删除某个用户购

对象

rk_s...

purchase_relations ...

users @network_su...

vendor @network_...

cart @network_sup...

新建

保存

另存为

添加栏位

插入栏位

删除栏位

主键

上移

下移

栏位

索引

外键

触发器

选项

注释

SQL 预览

名	类型	长度	小数点	不是 null	
order_id	bigint	0	0	<input checked="" type="checkbox"/>	1
consumer_id	bigint	0	0	<input checked="" type="checkbox"/>	
order_generation_time	timestamp	0	0	<input checked="" type="checkbox"/>	
total_order_price	decimal	10	2	<input checked="" type="checkbox"/>	
shipping_address	varchar	255	0	<input checked="" type="checkbox"/>	
order_status	int	0	0	<input checked="" type="checkbox"/>	
shipping_phone_number	bigint	0	0	<input checked="" type="checkbox"/>	

(订单表)

calhost_3306

JOYE

机

database1

information_schema

mysql

network_supermarket

表

cart

comments_media_url

goods

goods_media_url

goods_type

order

purchase_relations

users

vendor

视图

函数

f()

BOOL检查商家账号密码 (

f()

BOOL检查用户账号密码 (

p()

cart-删除购物车的某个商品

p()

cart-使用购物车创建订单

p()

cart-修改购物车某商品的数

p()

cart-直接删除某个用户购

对象

ions ...

users @network_su...

vendor @network_...

cart @network_sup...

goods @netwo

新建

保存

另存为

添加栏位

插入栏位

删除栏位

主键

上移

下移

栏位

索引

外键

触发器

选项

注释

SQL 预览

名	类型	长度	小数点	不是 null	
purchase_id	bigint	0	0	<input checked="" type="checkbox"/>	1
goods_id	bigint	0	0	<input checked="" type="checkbox"/>	
goods_type_id	bigint	0	0	<input type="checkbox"/>	
order_id	bigint	0	0	<input checked="" type="checkbox"/>	
goods_number	decimal	10	2	<input checked="" type="checkbox"/>	
user_goods_status	int	0	0	<input checked="" type="checkbox"/>	
actually_paid_money	decimal	10	2	<input checked="" type="checkbox"/>	
comments	varchar	255	0	<input type="checkbox"/>	
score	decimal	10	2	<input type="checkbox"/>	
express_number	varchar	255	0	<input type="checkbox"/>	
return_express_number	varchar	255	0	<input type="checkbox"/>	

(购买关系表)

山东科技大学学生课程设计

localhost_3306

JOYE

数据库

information_schema

mysql

network_supermarket

表

- cart
- comments_media_url
- goods
- goods_media_url
- goods_type
- order
- purchase_relations
- users
- vendor

视图

函数

- f() BOOL检查商家账号密码 (
- f() BOOL检查用户账号密码 (
- p() cart-删除购物车的某个商品
- p() cart-使用购物车创建订单
- p() cart-修改购物车某商品的数
- p() cart-直接删除某个用户购

对象
rk_su...
vendor @network...
cart @network_sup...
goods @network_s...
users @netwo

新建
保存
另存为
添加栏位
插入栏位
删除栏位
主键
上移
下移

栏位
索引
外键
触发器
选项
注释
SQL 预览

名	类型	长度	小数点	不是 null	
vendor_id	bigint	0	0	<input checked="" type="checkbox"/>	1
vendor_password	varchar	255	0	<input checked="" type="checkbox"/>	
vendor_name	varchar	255	0	<input checked="" type="checkbox"/>	
vendor_addr	varchar	255	0	<input type="checkbox"/>	
vendor_phone	bigint	0	0	<input checked="" type="checkbox"/>	2
vendor_icon_url	varchar	255	0	<input type="checkbox"/>	

(供货商表)

以构建商品表的脚本语句为例：

```
CREATE TABLE goods (
goods_id bigint NOT NULL COMMENT '商品 ID',
goods_name varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL
COMMENT '商品名称',
goods_describe varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NULL
DEFAULT NULL COMMENT '商品介绍',
amount decimal(10, 2) NOT NULL DEFAULT 9999.00 COMMENT '剩余量，默认 9999.00，需要
触发器 A，当 purchase_relations 产生新记录时自动扣除余量',
price decimal(10, 2) NOT NULL COMMENT '商品单价',
vendor_id bigint NULL DEFAULT NULL COMMENT '外键，商家 ID 号，服从于 vendor',
listing_time date NOT NULL DEFAULT '2000-01-01' COMMENT '上架时间，默认 2000-01-01',
goods_score varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NULL DEFAULT
NULL COMMENT '总评分，需要触发器 B，当 purchase_relations 改变时自动计算其总评分',
goods_status int NOT NULL DEFAULT 0 COMMENT '商品状态，默认 0，0 正在出售，1 暂时下
架，2 缺货（需要触发器）',
goods_media_group_id bigint NULL DEFAULT NULL COMMENT '商品详情媒体文件组 ID，控制
goods_media_url',
```

```
unit varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL DEFAULT  
'个' COMMENT '商品单位，默认\“个”',  
PRIMARY KEY (goods_id) USING BTREE,  
INDEX vid(vendor_id) USING BTREE,  
INDEX mid(goods_media_group_id) USING BTREE,  
CONSTRAINT vid FOREIGN KEY (vendor_id) REFERENCES vendor (vendor_id) ON DELETE SET  
NULL ON UPDATE CASCADE  
) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE = utf8mb4_0900_ai_ci ROW_FORMAT =  
Dynamic;
```

3. 建立索引

(1) 建立主键索引:

```
ALTER TABLE constumer ADD PRIMARY KEY (constumer_id)
```

```
ALTER TABLE goods ADD PRIMARY KEY (goods_id)
```

```
ALTER TABLE order ADD PRIMARY KEY (order_id)
```

```
ALTER TABLE purchase_relations ADD PRIMARY KEY (commodity_list_id)
```

```
ALTER TABLE vendor ADD PRIMARY KEY (vendor_id)
```

(2) 建立部分普通索引: 选择部分重要属性建立索引, 增加表的连接性和查找速度

```
CREATE INDEX acctont ON table_name (constumer)
```

```
CREATE INDEX price ON table_name (goods)
```

```
CREATE INDEX good_status ON table_name (goods)
```

```
CREATE INDEX total_order_price ON table_name (order)
```

```
CREATE INDEX vendor_name ON table_name (vendor)
```

3.5 数据库应用系统设计

3.5.1 数据初始化脚本:

将所有表内插入对应数据, 呈现结果如下图所示

山东科技大学学生课程设计

localhost_3306

ZUOYE

本机

database1

information_schema

mysql

network_supermarket

cart

comments_media

goods

goods_media_url

goods_type

order

purchase_relation

users

vendor

视图

函数

事件

查询

报表

备份

performance_schema

sys

webshopping

zuoye

对象

users @network_supermark...

开始事务

备注

筛选

排序

导入

导出

consumer_id	consumer_name	phone_number	gender	birthday	consumer_nickname	password	user_type	costumer_icon_url	exp
96267586	NonameUser	1848030850		0 2020-01-01	2f70b8a8-8688-11ed-a5ff-ds			0 http://gg.gg/12uucq	0
2	李小璐	13803022474		0 2020-01-22	爱吃西瓜子	673943	1	'http://gg.gg/12uucq'	0
4	韩先欢	13805524805		0 2020-01-26	ddddddddd	883077	3	'http://gg.gg/12uucq'	0
14	张翠芬	13810276502		0 2020-01-28	种草莓的小行家	623219	3	'http://gg.gg/12uucq'	0
15	梁哲宇	13817673620	1	2020-01-18	KUQA_16564684	316990	2	'http://gg.gg/12uucq'	0
12	林国瑞	13818918021	1	2020-01-14	WAKP_546848	231947	8	'http://gg.gg/12uucq'	0
18	吕浩然	13827004146	1	2020-01-13	WordExp_id1115	447533	4	'http://gg.gg/12uucq'	0
1	王小明	13834262909	1	2020-01-18	初夏的忧伤	aa330134	2	'http://gg.gg/12uucq'	6
13	刘珊珊	13839950701	0	2020-01-16	CETF_3055541	368628	4	'http://gg.gg/12uucq'	0
5	方晓彤	13859090908	0	2020-01-02	眼前心上人	553937	2	'http://gg.gg/12uucq'	0
11	许新翔	13859620967	1	2020-01-06	仿佛没有时间	140384	1	'http://gg.gg/12uucq'	0
16	李杰文	13869538941	1	2020-01-04	PFLUX_1656161	84401	3	'http://gg.gg/12uucq'	0
10	陈海洲	13871014729	1	2020-01-16	Dlplane	161077	8	'http://gg.gg/12uucq'	0
9	夏海琦	13893174123	0	2020-01-21	虚拟	15697	2	'http://gg.gg/12uucq'	0
8	钱欢孟	13895086451	0	2020-01-13	所谓对象	966801	4	'http://gg.gg/12uucq'	0
7	郑吉万	13927121280	1	2020-01-13	Browser	137374	5	'http://gg.gg/12uucq'	0
3	王青	13927915881	1	2020-01-04	NYIZ_626164	759809	4	'http://gg.gg/12uucq'	0
17	史路英	13944055767	0	2020-01-24	养猪的老婆婆	580987	5	'http://gg.gg/12uucq'	0
6	范久然	13992309231	0	2020-01-05	温柔善良了吗	213434	3	'http://gg.gg/12uucq'	0
67080256	NonameUser	17790788445	0	2020-01-01	87689ce4-8646-11ed-a5ff-Hc766443			0 http://gg.gg/12uucq	0

localhost_3306

ZUOYE

本机

database1

information_schema

mysql

network_supermarket

cart

comments_media

goods

goods_media_url

goods_type

order

purchase_relation

users

vendor

视图

函数

事件

查询

报表

备份

performance_schema

sys

webshopping

zuoye

对象

users @network_supermark...

order @network_supermar...

vendor @network_superma...

开始事务

备注

筛选

排序

导入

导出

vendor_id	vendor_password	vendor_name	vendor_addr	vendor_phone	vendor_icon_url
235704600	330134	善财	贵州省百色市上甘岭区鞍山	13843352331	
235709643	673943	楚元	贵州省安顺市伊春区白水	13909535916	
235726740	759809	冠诚	福建省安阳市金山屯区安顺	13861841029	
235727469	883077	凯杰	海南省白银市新市区安福路	13922257659	
235740687	553937	浩雄	湖北省安顺市南岗区安汾路	13919310081	
235742290	213434	兴简	江苏省巴中市乌马河区安远	13835463408	
235750163	137374	圣鸿	云南省白山市友好区白城南	13951483166	
235751798	966801	圣立	贵州省百色市上甘岭区宝通	13820493026	
235766924	15697	豪百	江西省安庆市带岭区北艾路	13990650885	
235777501	161077	心海	云南省安顺市伊春区宝源路	13891285775	
235777788	140384	财华	海南省巴罗维尔市美溪区安	13953644999	
235781881	231947	瀚新	吉林省保山市榆次区安化路	13847513742	

MyConnect

information_schema

mysql

network_supermarket

costumer

goods

order

purchase_relations

vendor

视图

函数

查询

报表

备份

performance_schema

sakila

sys

world

对象

costumer @network_supermarket (...)

costumer @network_supermarket (...)

goods @network_supermarket (My...

order @network_supermar...

开始事务

文本

筛选

排序

导入

导出

order_id	consumer_id	consumer_nickname	commodity_list_id	order_generation_time	total_order_price	shipping_address	order_status
15125094	1	初夏的忧伤	33647585	2020-01-08	3577.00	安徽省白银市新青区白	1
18381405	2	爱吃西瓜子	59722099	2020-01-26	2705.00	青海省巴中市乌马河区	1
20638443	3	NYIZ_626164	59269617	2020-01-29	6442.00	吉林省包头市红山区西	1
22186516	4	ddddddddd	75117886	2020-01-12	9631.00	甘肃省安顺市南岗区安	0
30313514	5	眼前心上人	509327	2020-01-23	6131.00	云南省白山市友好区保	1
31026789	6	温柔善良了吗	38938204	2020-01-10	679.00	云南省包头市红山区白	0
32410106	7	Browser	70652958	2020-01-30	1107.00	山东省白山市友好区宝	0
37073146	8	所谓对象	64555955	2020-01-11	2299.00	山东省白银市新青区宝	1
38816738	9	虚拟	38640722	2020-01-17	2470.00	河南省包头市红山区西	1
39707074	10	Dlplane	58650034	2020-01-16	8601.00	广东省鞍山市西林区安	1
41903657	11	仿佛没有时间	18610262	2020-01-06	8384.00	江苏省宝源市源旺河区	0
42396757	12	WAKP_546848	23487284	2020-01-06	6394.00	黑龙江省保山市榆次区	1
45694499	13	CETF_3055541	93579713	2020-01-20	7918.00	吉林省安阳市带岭区宝	1
46988070	14	种草莓的小行家	81465746	2020-01-21	9064.00	福建省安顺市南岗区西	1
47540076	15	KUQA_16564684	99593905	2020-01-21	7076.00	云南省鞍山市西林区区	1
49279940	16	PFLUX_1656161	97317480	2020-01-11	41.00	湖北省安顺市伊春区安	1
56446157	17	养猪的老婆婆	810289	2020-01-16	4347.00	安徽省安顺市伊春区都	1
58608254	18	WordExp_id1115	58363399	2020-01-22	794.00	云南省安顺市南岗区白	0
60465251	7	Browser	50049067	2020-01-23	4019.00	甘肃省鞍山市西林区安	0
71898731	8	所谓对象	64988131	2020-01-23	4185.00	河北省包头市红山区安	0
74989728	9	虚拟	65829481	2020-01-03	3478.00	河南省保山市榆次区宝	1
76546575	10	Dlplane	10371318	2020-01-25	4423.00	湖北省鞍山市西林区白	1
79412857	11	仿佛没有时间	2402424	2020-01-26	4318.00	贵州省白银市新青区白	1
85861760	12	WAKP_546848	10415602	2020-01-27	5304.00	福建省鞍山市西林区安	0
89188421	13	CETF_3055541	24910237	2020-01-22	279.00	台湾省白山市红山区宝	1
91713787	14	种草莓的小行家	57257140	2020-01-23	2287.00	山西省包头市红山区安	1
98458244	15	KUQA_16564684	50172452	2020-01-06	2003.00	甘肃省安阳市带岭区安	0
98910210	16	PFLUX_1656161	81562544	2020-01-24	5381.00	湖南省安阳市带岭区巴	1

山东科技大学学生课程设计

MyConnect

information_schema

mysql

network_supermarket

costumer

goods

order

purchase_relations

vendor

视图

函数

查询

备份

performance_schema

sakila

sys

world

costumer @network_supermar...

costumer @network_supermar...

goods @network_supermarke...

order @network_super

开始事务

文本

筛选

排序

导入

导出

commodity_list_id	goods_id	goods_number
509327	14474266	1
509327	88352114	1
509327	76102266	1
509327	77941370	1
509327	30197556	1
810289	34132264	1
810289	61694478	1
810289	13732011	1
810289	29087391	1
810289	65934330	1
2402424	6291540	1
2402424	68115411	1
2402424	38869730	1
2402424	29796372	1
2402424	38762704	1
10371318	92788221	1
10371318	2848984	1
10371318	97166641	1
10371318	61716169	1
10371318	37135991	1
10415602	65934330	1
10415602	75748589	1
10415602	92788221	1
10415602	2848984	1
10415602	97166641	1
18610262	61694478	1
18610262	13732011	1
18610262	29087391	1
18610262	65934330	1
18610262	75748589	1

MyConnect

information_schema

mysql

network_supermarket

costumer

goods

order

purchase_relations

vendor

视图

函数

查询

备份

performance_schema

sakila

sys

world

costumer @network_su...

costumer @network_su...

goods @network_super...

order @network_super...

purchase_relations @net...

vendor @netv

开始事务

文本

筛选

排序

导入

导出

vendor_id	vendor_name	vendor_addr	vendor_phone	vendor_comment
235704600	贵州	贵州省百色市上甘	13843352331	
235709643	贵州	贵州省安顺市伊春	13909535916	
235726740	福建	福建省安阳市金山	13861841029	
235727469	海南	海南省白银市新青	13922257659	
235740687	湖北	湖北省安阳市南农	13919310081	
235742290	江苏	江苏省巴中市乌马	13835463408	
235750163	云南	云南省白山市友好	13951483166	
235751798	贵州	贵州省百色市上甘	13820493026	
235766924	江西	江西省安阳市南农	13990650885	
235777501	云南	云南省安顺市伊春	13891285775	
235777788	海南	海南省巴罗维尔市	13953644999	
235781881	吉林	吉林省保山市施次	13847513742	

);

INSERT INTO `goods` VALUES (14474266, '章鱼小丸子', '来自日本的章鱼烧更名而成, 名字可爱, 丸子更可爱, 风中摇曳的买皮酥, 仿佛在热情的向你挥动双手。', 5073.00, 0.33, 235750163, '2020-01-13', '9.2', 1, NULL, '个');

INSERT INTO `goods` VALUES (16857887, '手表', '现在很多摆摊卖手表的小贩, 虽然卖的人多, 但不得不说卖手表真的是一个很好的选择。', 3508.00, 3.58, 235751798, '2020-01-25', '3.1', 1, NULL, '个');

INSERT INTO `goods` VALUES (29087391, '烤红薯', '健康食品不会消失在街头巷尾, 捧着热烘烘的新鲜红薯, 温暖整个人。', 9766.00, 2.27, 235781881, '2020-01-17', '0.9', 1, NULL, '个');

INSERT INTO `goods` VALUES (29796372, '手机壳', '现代青年每天与手机形影不离, 生活中比什么都要爱护它, 所以很多人都喜欢把自己的手机装饰得更好看, 而这种经常会换的物品, 摆地摊再合适不过了。', 7501.00, 7.35, 235766924, '2020-01-28', '0.2', 1, NULL, '个');

INSERT INTO `goods` VALUES (30197556, '毛绒玩具', '摆地摊选择的一定是人流比较密集的地方, 而毛绒玩具主要用来吸引女顾客和小孩子。', 9670.00, 6.42, 235740687, '2020-01-06', '7.1', 1, NULL, '个');

INSERT INTO `goods` VALUES (34132264, '雨伞', '雨伞可谓是男女老少皆宜, 款式风格多样, 价格也没有固定的, 卖雨伞的利润可想而知。', 1635.00, 4.43, 235704600, '2020-01-03', '3.6', 1, NULL, '个');

INSERT INTO `goods` VALUES (37135991, '帆布鞋', '现在的青少年, 谁的手机没有几双帆布鞋, 这种价格不贵, 又富有青春气息的物品一直就很讨人喜欢。', 8727.00, 7.72, 235750163, '2020-01-05', '2.6', 1, NULL, '个');

INSERT INTO `goods` VALUES (37936671, '水杯', '水杯有各种各样, 而且是个实用的物品, 用这个商品摆地摊, 也能吸引不少顾客呢。', 4744.00, 0.48, 235777501, '2020-01-14', '9.1', 1, NULL, '个');

INSERT INTO `goods` VALUES (38762704, 'T 恤', '夏季还没有来临之前, 就有一些小商贩们开始摆地摊卖T 恤, 纯白色的印有英文和卡通人物, 按照尺寸大小, 卖价到块不等。这类商品在前几年出现的时候遭到顾客的疯抢。不要看它价格低, 其实利润还是不错的。摆这类商品的地摊最好选择在居民区夜市或者学校旁边。', 2481.00, 0.29, 235781881, '2020-01-12', '3.3', 1, NULL, '个');

山东科技大学学生课程设计

`INSERT INTO `goods` VALUES (38869730, '手抓饼', '起源于台湾, 千层百叠, 层如薄纸, 用手抓之, 面丝千连, 香酥可口。', 1964.00, 0.56, 235742290, '2020-01-19', '5.1', 1, NULL, '个');`

`INSERT INTO `goods` VALUES (47201884, '小风扇', '这是夏季的标配, 很多学校没有安装空调, 只能靠小风扇过夜, 但学校点之后会断电, 那种带 USB 插口可以靠着充电宝持续供电的小风扇就得到了他们青睐, 所以它的销量持续走俏, 基本上是人手一个。坐办公室的白领有些也需要这种小风扇。这种功能小风扇在地摊上可以卖到二十三十, 但成本只有几块。', 9250.00, 5.74, 235727469, '2020-01-15', '8.9', 1, NULL, '个');`

`INSERT INTO `goods` VALUES (58298810, '煎饼果子', '从小吃到大, 现在几乎走到哪个城市, 都能在地摊上见到它, 热乎乎的煎饼果子, 营养又美味, 老少皆宜。', 7834.00, 2.39, 235709643, '2020-01-30', '7.0', 1, NULL, '个');`

`INSERT INTO `goods` VALUES (61694478, '炸臭豆腐', '对于有些人来说是避之不及的食物, 但是对于有些人来说却是人间的美味, 它闻着臭吃着香, 有一种特殊的神奇魅力。', 8925.00, 8.21, 235742290, '2020-01-17', '5.0', 1, NULL, '个');`

`INSERT INTO `goods` VALUES (61716169, '烤冷面', '风靡东三省, 逐渐向南延伸, 已成为街边小吃的心头爱, 甜甜辣辣的味道, 劲道的面皮, 小小的一道小吃, 满足了人们对面的各种幻想。', 5644.00, 6.83, 235777501, '2020-01-17', '0.6', 1, NULL, '个');`

`INSERT INTO `goods` VALUES (65934330, '铁板鱿鱼', '皮酥肉香是鱿鱼的特色, 不仅是国人的美味, 连外国友人都啧啧称奇。', 761.00, 7.46, 235726740, '2020-01-16', '6.9', 1, NULL, '个');`

3.5.2 业务查询及查询脚本

以函数进行业务查询的实现, 编写的函数语句及其功能如下图所示:

p() 按ID查询商品
p() 按ID查询用户
p() 按名称查询用户
p() 按商品名称查询商品库里商品的描述与首张图
p() 查询某订单的详细信息
p() 查询某个商家所有商品的描述与首张图
p() 查询某评论的图片视频
p() 查询某商家特定状态的交易列表
p() 查询某商品的所有文字评论
p() 查询某商品的所有文字信息
p() 查询某商品的图片视频
p() 查询某商品所有型号详细信息
p() 查询某用户的购物车内容
p() 查询某用户的信息
p() 查询某用户特定状态的订单
p() 查询所有的商品
p() 查询所有的用户

(1) 按 ID 查询商品:

BEGIN

SELECT goods.* FROM goods WHERE goods.goods_id=gid;

END

(2) 按 ID 查询用户:

BEGIN

SELECT users.* FROM users WHERE users.consumer_id=uid;

END

(3) 按名称查询用户:

BEGIN

SELECT users.* FROM users

WHERE users.consumer_name like CONCAT('%',ipt,'%') or users.consumer_nickname like
CONCAT('%',ipt,'%');

END

(4) 按商品名称查询商品库里商品的描述与首张图:

BEGIN

IF paixv_condition='SCORE' THEN

IF paixv_method='DESC' THEN

SELECT goods.* FROM goods

```
WHERE goods.goods_describe like CONCAT('%',ipt,'%') or goods.goods_name like  
CONCAT('%',ipt,'%') ORDER BY goods.goods_score DESC;
```

```
ELSE
```

```
SELECT goods.* FROM goods
```

```
WHERE goods.goods_describe like CONCAT('%',ipt,'%') or goods.goods_name like  
CONCAT('%',ipt,'%') ORDER BY goods.goods_score ASC;
```

```
END IF;
```

```
ELSE
```

```
IF paixv_method='DESC' THEN
```

```
SELECT goods.* FROM goods
```

```
WHERE goods.goods_describe like CONCAT('%',ipt,'%') or goods.goods_name like  
CONCAT('%',ipt,'%') ORDER BY goods.price DESC;
```

```
ELSE
```

```
SELECT goods.* FROM goods
```

```
WHERE goods.goods_describe like CONCAT('%',ipt,'%') or goods.goods_name like  
CONCAT('%',ipt,'%') ORDER BY goods.price ASC;
```

```
END IF;
```

```
END IF;
```

```
END
```

（5）查询某订单的详细信息：

```
BEGIN
```

```
SELECT
```

```
`order`.*,
```

```
purchase_relations.purchase_id,
```

```
purchase_relations.goods_id,
```

```
purchase_relations.goods_type_id,
```

```
purchase_relations.goods_number,
```

```
purchase_relations.user_goods_status,
```

```
purchase_relations.actually_paid_money,
```



```
purchase_relations.comments,  
purchase_relations.score,  
purchase_relations.express_number,  
purchase_relations.return_express_number  
FROM  
`order`  
LEFT JOIN  
purchase_relations  
ON  
`order`.order_id = purchase_relations.order_id  
WHERE  
`order`.order_id=oid;  
END
```

（6）查询某个商家所有商品的描述与首张图：

```
BEGIN  
IF paixv_condition='SCORE' THEN  
IF paixv_method='DESC' THEN  
SELECT goods.* FROM goods  
WHERE goods.vendor_id=vid ORDER BY goods.goods_score DESC;  
ELSE  
SELECT goods.* FROM goods  
WHERE goods.vendor_id=vid ORDER BY goods.goods_score ASC;  
END IF;  
ELSE  
IF paixv_method='DESC' THEN  
SELECT goods.* FROM goods  
WHERE goods.vendor_id=vid ORDER BY goods.price DESC;  
ELSE  
SELECT goods.* FROM goods
```

```
WHERE goods.vendor_id=vid ORDER BY goods.price ASC;
```

```
END IF;
```

```
END IF;
```

```
END
```

（7）查询某评论的图片视频：

```
BEGIN
```

```
SELECT
```

```
comments_media_url.media_id,
```

```
comments_media_url.url
```

```
FROM
```

```
comments_media_url
```

```
WHERE
```

```
comments_media_url.purchase_id=pid
```

```
ORDER BY comments_media_url.in_group_id ASC;
```

```
END
```

（8）查询某商家特定状态的交易列表：

```
BEGIN
```

```
IF stat=-1 THEN
```

```
SELECT
```

```
purchase_relations.*
```

```
FROM
```

```
goods
```

```
INNER JOIN
```

```
purchase_relations
```

```
ON
```

```
goods.goods_id = purchase_relations.goods_id
```

```
WHERE
```

```
goods.vendor_id = vid;
```

```
ELSE
```

```
SELECT
purchase_relations.*
FROM
goods
INNER JOIN
purchase_relations
ON
goods.goods_id = purchase_relations.goods_id
WHERE
goods.vendor_id = vid AND purchase_relations.user_goods_status=stat;
END IF;
END
```

（9）查询某商品的所有文字评论：

```
BEGIN
SELECT purchase_relations.comments
FROM purchase_relations WHERE purchase_relations.goods_id=gid;
END
```

（10）查询某商品的所有文字信息：

```
BEGIN
SELECT goods.* FROM goods WHERE goods.goods_id=gid;
END
```

（11）查询某商品的图片视频：

```
BEGIN
SELECT
goods_media_url.media_id,
goods_media_url.url
FROM
goods_media_url
WHERE
```

```
goods_media_url.goods_id=gid
```

```
ORDER BY goods_media_url.in_group_id ASC;
```

```
END
```

（11）查询某商品所有型号详细信息：

```
BEGIN
```

```
SELECT goods_type.goods_type_id, goods_type.goods_type_name,
```

```
goods_type.type_media_url
```

```
FROM goods_type
```

```
WHERE goods_type.goods_id=gid ORDER BY goods_type.in_group_id ASC;
```

```
END
```

（12）查询某用户的购物车内容：

```
BEGIN
```

```
SELECT
```

```
*
```

```
FROM
```

```
cart
```

```
INNER JOIN
```

```
goods
```

```
ON
```

```
cart.goods_id = goods.goods_id
```

```
WHERE cart.user_id=uid;
```

```
END
```

（13）查询某用户的信息：

```
BEGIN
```

```
SELECT users.* FROM users WHERE users.consumer_id=uid;
```

```
END
```

（14）查询某用户特定状态的订单：

```
BEGIN
```

```
IF stat=-1 THEN
```

```
SELECT `order`.* FROM `order` WHERE `order`.consumer_id=uid;
```

```
ELSE
```

```
SELECT `order`.* FROM `order` WHERE `order`.consumer_id=uid AND
```

```
`order`.order_status=stat;
```

```
END IF;
```

```
END
```

（15）查询所有的商品：

```
BEGIN
```

```
SELECT goods.* FROM goods;
```

```
END
```

（16）查询所有的用户：

```
BEGIN
```

```
SELECT users.* FROM users;
```

```
END
```

3.5.3 数据更新及脚本

f() BOOL检查商家账号密码（可用手机号做账号）

f() BOOL检查用户账号密码（可用手机号做账号）

p() cart-删除购物车的某个商品

p() cart-使用购物车创建订单

p() cart-修改购物车某商品的数量

p() cart-直接删除某个用户购物车中所有商品

p() comments_media_url_用户向评论中添加一个图片或视频的url

p() goods&goods_type&cart-向购物车中添加不含型号商品

p() goods&goods_type&cart-向购物车中添加含型号商品

p() goods&goods_type-商家为某商品添加属于某商品的型号

p() goods_media_url-商家向某个商品中添加图片视频URL

p() goods-商家删除某个商品

p() goods-商家上架一款商品

p() goods-商家修改商品信息

f() order-MAKE_AN_EMPTY_ORDER
p() order-商家填写物流单号
p() order-商家于用户商品状态操作
p() order-使用一个含型号商品直接创建某用户的订单
p() order-使用一个无型号商品直接创建某用户的订单
p() order-用户填写退货单号
p() order-用户修改订单电话号码
p() order-用户修改订单收货地址
p() order-用户于订单状态操作
p() purchase-用户对一条某商品评分
p() purchase-用户上传一条某商品评论
p() purchase-用户于单商品状态操作
p() users-创建一个新用户
p() users-检查用户账号密码 (可用手机号做账号)
p() users-修改一个用户的信息
p() users-注销一个用户
p() vendor-创建商家
p() vendor-检查商家账号密码 (可用手机号做账号)
p() vendor-修改商家信息
p() vendor-注销商家

(1) cart-删除购物车的某个商品:

BEGIN

DELETE FROM cart WHERE cart.goods_id=gid AND cart.user_id=uid;

END

(2) cart-修改购物车某商品的数量:

BEGIN

#Routine body goes here...

UPDATE cart SET cart.goods_number=num

WHERE cart.user_id=uid AND cart.goods_id=gid;

END

(3) cart-直接删除某个用户购物车中所有商品:

BEGIN

#Routine body goes here...

DELETE FROM cart WHERE cart.user_id=uid;

END

(4) goods-商家删除某个商品:

```
BEGIN
```

```
DELETE FROM goods WHERE goods.goods_id=gid;
```

```
END
```

（5）goods-商家修改商品信息：

```
BEGIN
```

```
UPDATE goods
```

```
SET goods.amount=amt,goods.goods_describe=descri,
```

```
goods.unit=danwei,goods.goods_name=nam,goods.price=pr;
```

```
END
```

（6）order-用户修改订单电话号码：

```
BEGIN
```

```
UPDATE `order` SET `order`.shipping_phone_number=phone
```

```
WHERE `order`.order_id=orderid;
```

```
END
```

（7）order-用户修改订单收货地址：

```
BEGIN
```

```
UPDATE `order` SET `order`.shipping_address=addr
```

```
WHERE `order`.order_id=orderid;
```

```
END
```

（8）vendor-修改商家信息：

```
BEGIN
```

```
UPDATE vendor
```

```
SET vendor.vendor_password=pwd, vendor.vendor_name=nam,
```

```
vendor.vendor_addr=addr, vendor.vendor_phone=phone,
```

```
vendor.vendor_icon_url=icon WHERE vendor.vendor_id=id;
```

```
END
```

（9）order-商家填写物流单号：

```
BEGIN
```

```
UPDATE purchase_relations SET purchase_relations.express_number=express
```

```
WHERE purchase_relations.purchase_id=pid;
```

```
END
```

（10）order-商家于用户商品状态操作：

```
BEGIN
```

```
UPDATE purchase_relations SET purchase_relations.user_goods_status=goalstatus
```

```
WHERE purchase_relations.purchase_id=pid;
```

```
END
```

3.5.4 视图创建脚本

视图可以对建立的数据表进行预览，如下建立了用户名视图

(1)用户名视图：

```
CREATE VIEW username AS
```

```
SELECT
```

```
consumer_id,consumer_name
```

```
FROM
```

```
users
```

```
SELECT
```

```
*
```

```
FROM
```

```
username;
```

（2）商品价格视图：

```
CREATE VIEW goodsprice AS
```

```
SELECT
```

```
goods_id,goods_name,goods_price
```

```
FROM
```

```
goods
```

```
SELECT
```

```
*
```

```
FROM
```

```
goodsprice;
```


(3)购买信息视图:

```
CREATE VIEW purchase_relations_infor AS
```

```
SELECT
```

```
purchase_id,goods_id,order_id
```

```
FROM
```

```
purchase_relations
```

```
SELECT
```

```
*
```

```
FROM
```

```
purchase_relations_info;
```

(4)购货商姓名视图:

```
CREATE VIEW vendor_name AS
```

```
SELECT
```

```
vendor_id,vendor_name
```

```
FROM
```

```
vendor
```

```
SELECT
```

```
*
```

```
FROM
```

```
vendor_name;
```

(5)订单生成时间视图:

```
CREATE VIEW ordertime AS
```

```
SELECT
```

```
order_id, order_generation_time
```

```
FROM
```

```
order
```

```
SELECT
```

```
*
```

```
FROM
```

ordertime;

3.5.5 索引创建脚本:

名	栏位	索引类型	索引方法
vendor_id	vendor_id	Unique	BTREE

排序规则: A
基数: 12
已包:
注释:

名	栏位	索引类型	索引方法
consumer_id	consumer_id	Unique	BTREE
phone	phone_number	Unique	BTREE
consumer_name	consumer_name	Normal	BTREE

排序规则: A
基数: 19
已包:
注释:

The screenshot shows the MySQL Workbench 'Table Structure' window for the 'goods' table in the 'network_supermarket' database. The table has two columns: 'gid' (Unique, BTREE) and 'vid' (Normal, BTREE). The '排序规则' (Collation) is 'A' and the '基数' (Cardinality) is '24'. The '已包' (Included) and '注释' (Comments) fields are empty.

(在 navicat 中提供了建立索引的可视化操作，以下取主键索引的建立为例)

```
ALTER TABLE users ADD PRIMARY KEY (constumer id)
```

```
ALTER TABLE goods ADD PRIMARY KEY (goods_id)
```

```
ALTER TABLE order ADD PRIMARY KEY (order_id)
```

```
ALTER TABLE purchase_relations ADD PRIMARY KEY (commodity_list_id)
```

```
ALTER TABLE vendor ADD PRIMARY KEY( vendor_id )
```

3.5.6 存储过程创建脚本

(1) cart-使用购物车创建订单:

```
BEGIN

set @order_id=`order-MAKE_AN_EMPTY_ORDER`(uid, addr, PHONE);

INSERT INTO purchase_relations (purchase_relations.order_id, purchase_relations.goods_id,
purchase_relations.goods_number,purchase_relations.goods_type_id,
purchase_relations.actually_paid_money)
SELECT @order_id, cart.goods_id, cart.goods_number, cart.goods_type_id, cart.actually_money
FROM cart WHERE cart.user_id=uid;
```

END

(2) comments_media_url-用户向评论中添加一个图片或视频的 url:

```
BEGIN
```

```
#Routine body goes here...
```

```
INSERT INTO comments_media_url SET comments_media_url.purchase_id=purchase_id,
comments_media_url.url=url;
```

END

(3) goods&goods_type&cart-向购物车中添加不含型号商品:

```
BEGIN
```

```
#Routine body goes here...
```

```
INSERT INTO cart
```

```
SET cart.discount=disc, cart.goods_id=gid, cart.goods_number=nums, cart.user_id=uid;
```

END

(4) order-使用一个无型号商品直接创建某用户的订单:

```
BEGIN
```

```
#Routine body goes here...
```

```
set @order_id=`order-MAKE_AN_EMPTY_ORDER`(uid, addr, PHONE);
```

```
set @pc=(SELECT goods.price FROM goods
```

```
WHERE goods.goods_id=GOODSID)*discount;
```

```
INSERT INTO purchase_relations
```

```
SET purchase_relations.goods_id=GOODSID,
```

```
purchase_relations.order_id=@order_id, purchase_relations.actually_paid_money=@pc;
```

END

(5) users-创建一个新用户:

BEGIN

INSERT INTO users SET users.phone_number=phone, users.`password`=pwd;

END

3.5.7 触发器创建脚本



The screenshot shows a MySQL database management tool interface. On the left, a tree view shows the database structure, with the 'goods' table selected under the 'network_supermarket' database. On the right, the '定义' (Definition) tab is active, displaying the SQL script for creating a trigger named 'goods_id_trigger'. The script is as follows:

```
BEGIN
DECLARE rnd BIGINT UNSIGNED;
DECLARE msg VARCHAR(255);
IF new.amount<1 THEN
    set msg = "数量非法";
    SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;
END IF;
r: REPEAT
SET rnd = ROUND(10000000+(rand()*90000000));
UNTIL NOT EXISTS( SELECT 1 FROM `goods` WHERE goods.goods_id = rnd ) END REPEAT r;
SET new.goods_id = rnd;
END
```

(1) 商品表的触发器:

BEGIN

DECLARE rnd BIGINT UNSIGNED;

DECLARE msg VARCHAR(255);

IF new.amount<1 THEN

set msg = "数量非法";

SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;

END IF;

r: REPEAT

SET rnd = ROUND(10000000+(rand()*90000000));

UNTIL NOT EXISTS(SELECT 1 FROM `goods` WHERE goods.goods_id = rnd)

END REPEAT r;

SET new.goods_id = rnd;

END

(2) 商品表的非法数据触发器:

```
BEGIN

DECLARE msg VARCHAR(255);

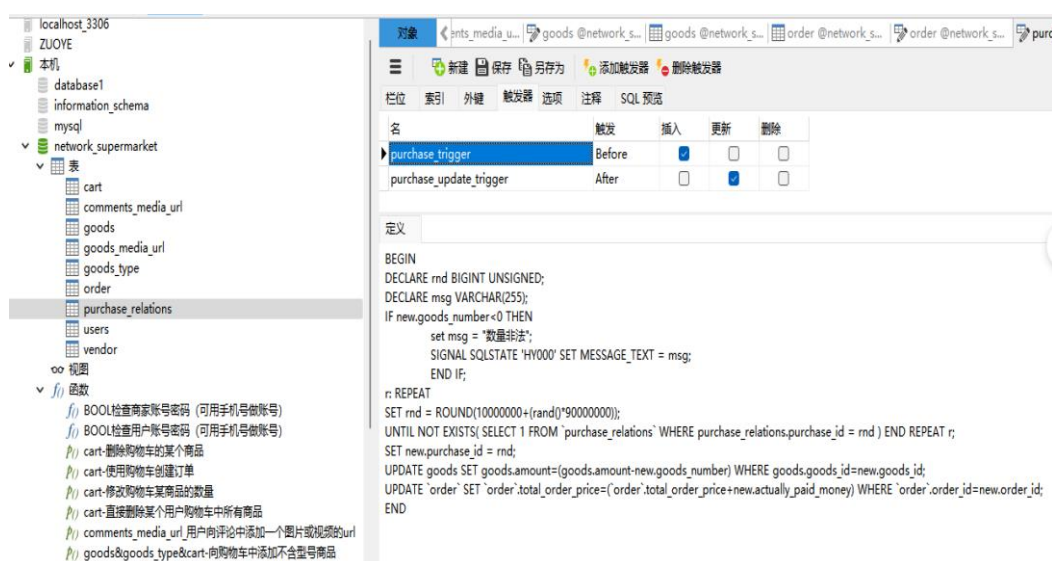
IF new.amount<0 THEN

set msg = "剩余量不足";

SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;

END IF;

END
```



(3) 购买关系表的触发器:

```
BEGIN

DECLARE rmd BIGINT UNSIGNED;

DECLARE msg VARCHAR(255);

IF new.goods_number<0 THEN

set msg = "数量非法";

SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;

END IF;

r: REPEAT

SET rmd = ROUND(100000000+(rand()*900000000));

UNTIL NOT EXISTS( SELECT 1 FROM `purchase_relations`

WHERE purchase_relations.purchase_id = rmd ) END REPEAT r;
```

```
SET new.purchase_id = rnd;

UPDATE goods SET goods.amount=(goods.amount-new.goods_number)

WHERE goods.goods_id=new.goods_id;

UPDATE `order`

SET `order`.total_order_price=(`order`.total_order_price+new.actually_paid_money)

WHERE `order`.order_id=new.order_id;

END
```

（4）购买关系表的更新异常触发器：

```
BEGIN

DECLARE tmp_score DECIMAL(10,2);

DECLARE userid INT;

DECLARE msg VARCHAR(255);

IF new.goods_number<0 THEN

set msg = "数量非法";

SIGNAL SQLSTATE 'HY000' SET MESSAGE_TEXT = msg;

END IF;

SET tmp_score=(SELECT AVG(purchase_relations.score) FROM purchase_relations

WHERE new.goods_id=purchase_relations.goods_id);

SET userid=(SELECT `order`.consumer_id FROM `order`

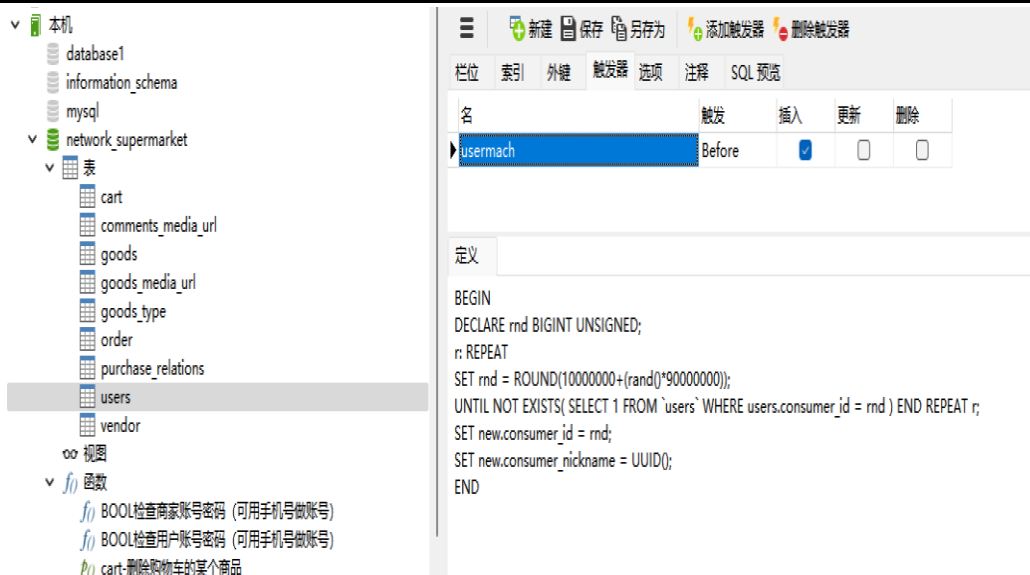
WHERE `order`.order_id=new.order_id);

UPDATE users SET users.exp=(users.exp+1) WHERE users.consumer_id=userid;

UPDATE goods SET goods.goods_score=tmp_score

WHERE goods.goods_id=new.goods_id;

END
```



Left pane (Database Explorer):

- 本机
 - database1
 - information_schema
 - mysql
 - network_supermarket
 - 表
 - cart
 - comments_media_url
 - goods
 - goods_media_url
 - goods_type
 - order
 - purchase_relations
 - users
 - vendor
 - 视图
 - 函数
 - f() BOOL检查商家账号密码 (可用手机号做账号)
 - f() BOOL检查用户账号密码 (可用手机号做账号)
 - cart-删除购物车的某个商品

Right pane (Trigger Definition):

名	触发	插入	更新	删除
usermach	Before	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

定义:

```

BEGIN
DECLARE rnd BIGINT UNSIGNED;
r: REPEAT
SET rnd = ROUND(10000000+(rand()*90000000));
UNTIL NOT EXISTS( SELECT 1 FROM `users` WHERE users.consumer_id = rnd ) END REPEAT r;
SET new.consumer_id = rnd;
SET new.consumer_nickname = UUID();
END
    
```

(5) 顾客触发器:

BEGIN

DECLARE rnd BIGINT UNSIGNED;

r: REPEAT

SET rnd = ROUND(10000000+(rand()*90000000));

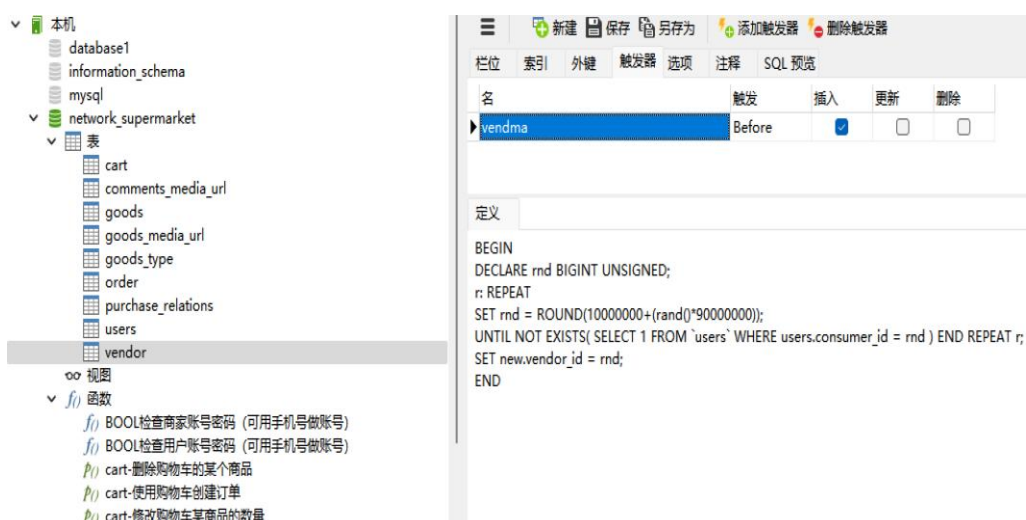
UNTIL NOT EXISTS(SELECT 1 FROM `users` WHERE users.consumer_id = rnd)

END REPEAT r;

SET new.consumer_id = rnd;

SET new.consumer_nickname = UUID();

END



Left pane (Database Explorer):

- 本机
 - database1
 - information_schema
 - mysql
 - network_supermarket
 - 表
 - cart
 - comments_media_url
 - goods
 - goods_media_url
 - goods_type
 - order
 - purchase_relations
 - users
 - vendor
 - 视图
 - 函数
 - f() BOOL检查商家账号密码 (可用手机号做账号)
 - f() BOOL检查用户账号密码 (可用手机号做账号)
 - cart-删除购物车的某个商品
 - cart-使用购物车创建订单
 - cart-修改购物车某商品的数量

Right pane (Trigger Definition):

名	触发	插入	更新	删除
vendma	Before	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

定义:

```

BEGIN
DECLARE rnd BIGINT UNSIGNED;
r: REPEAT
SET rnd = ROUND(10000000+(rand()*90000000));
UNTIL NOT EXISTS( SELECT 1 FROM `users` WHERE users.consumer_id = rnd ) END REPEAT r;
SET new.vendor_id = rnd;
END
    
```

(6) 供货商触发器:


```
BEGIN

DECLARE rnd BIGINT UNSIGNED;

r: REPEAT

SET rnd = ROUND(100000000+(rand()*900000000));

UNTIL NOT EXISTS( SELECT 1 FROM `users` WHERE users.consumer_id = rnd ) END REPEAT r;

SET new.vendor_id = rnd;

END
```

3.5.8 安全方案设计及实现脚本

安全方案设计主要实现对账号和密码的安全性检测，如果账号和密码之间不匹配，那么就没有通过安全性检测，就不能对数据进行正常操作。同时在真实的网络购物系统中，为方便客户熟记账号，常常用手机号来代替账号，因此在设计安全方案的时候也增加了这一功能。

(1) **BOOL 检查商家账号密码（可用手机号做账号）:**

```
BEGIN

SET @A = ((SELECT vendor.vendor_password
FROM vendor WHERE vendor.vendor_id = id) = pwd);

SET @B = (IF(@A = 1, 1, 0));

SET @C = ((SELECT vendor.vendor_password FROM vendor
WHERE vendor.vendor_phone = id) = pwd);

SET @D = (IF(@C = 1, 1, 0));

SET @E = (IF(@B = 1 OR @D = 1, 1, 0));

RETURN @E;

END
```

(2) **BOOL 检查用户账号密码（可用手机号做账号）:**

```
BEGIN

SET @A = ((SELECT users.`password` FROM users
WHERE users.consumer_id = id) = pwd);

SET @B = (IF(@A = 1, 1, 0));

SET @C = ((SELECT users.`password` FROM users
WHERE users.phone_number = id) = pwd);
```

```
SET @D = (IF(@C = 1, 1, 0));

SET @E = (IF(@B = 1 OR @D = 1, 1, 0));

RETURN @E;

END
```

(3) vendor-检查商家账号密码 (可用手机号做账号):

```
BEGIN

SELECT vendor.vendor_id FROM vendor

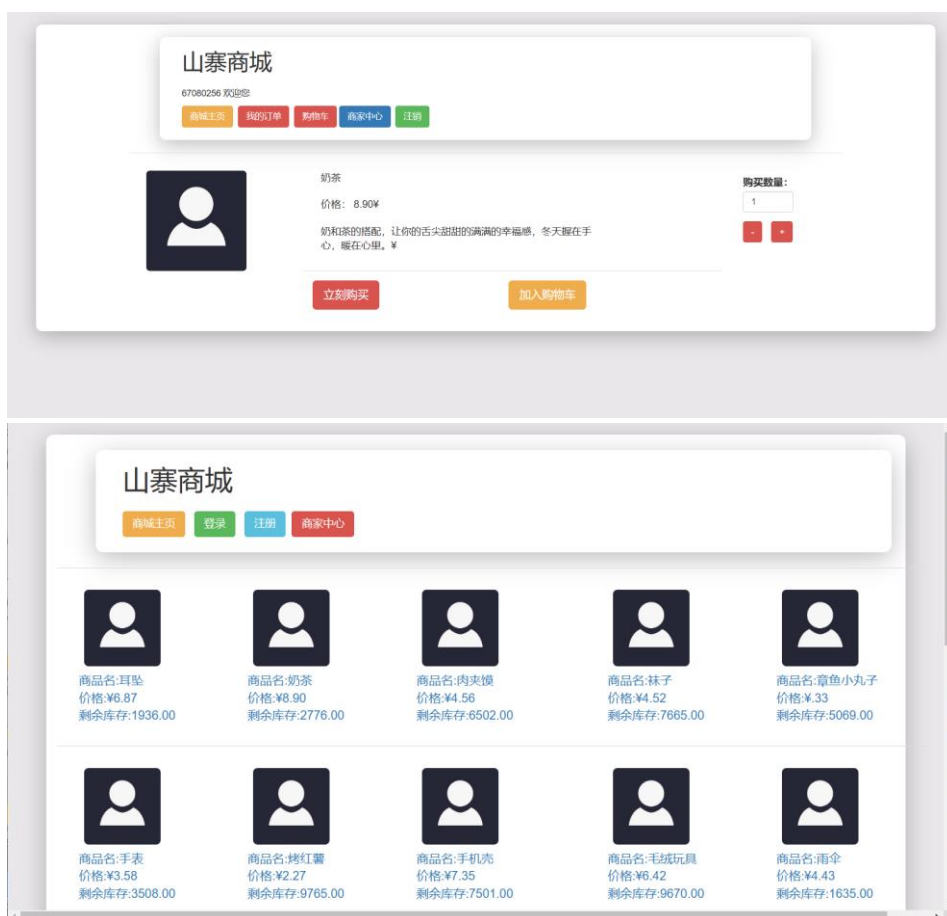
WHERE (vendor.vendor_id=uid AND vendor.vendor_password=pwd)

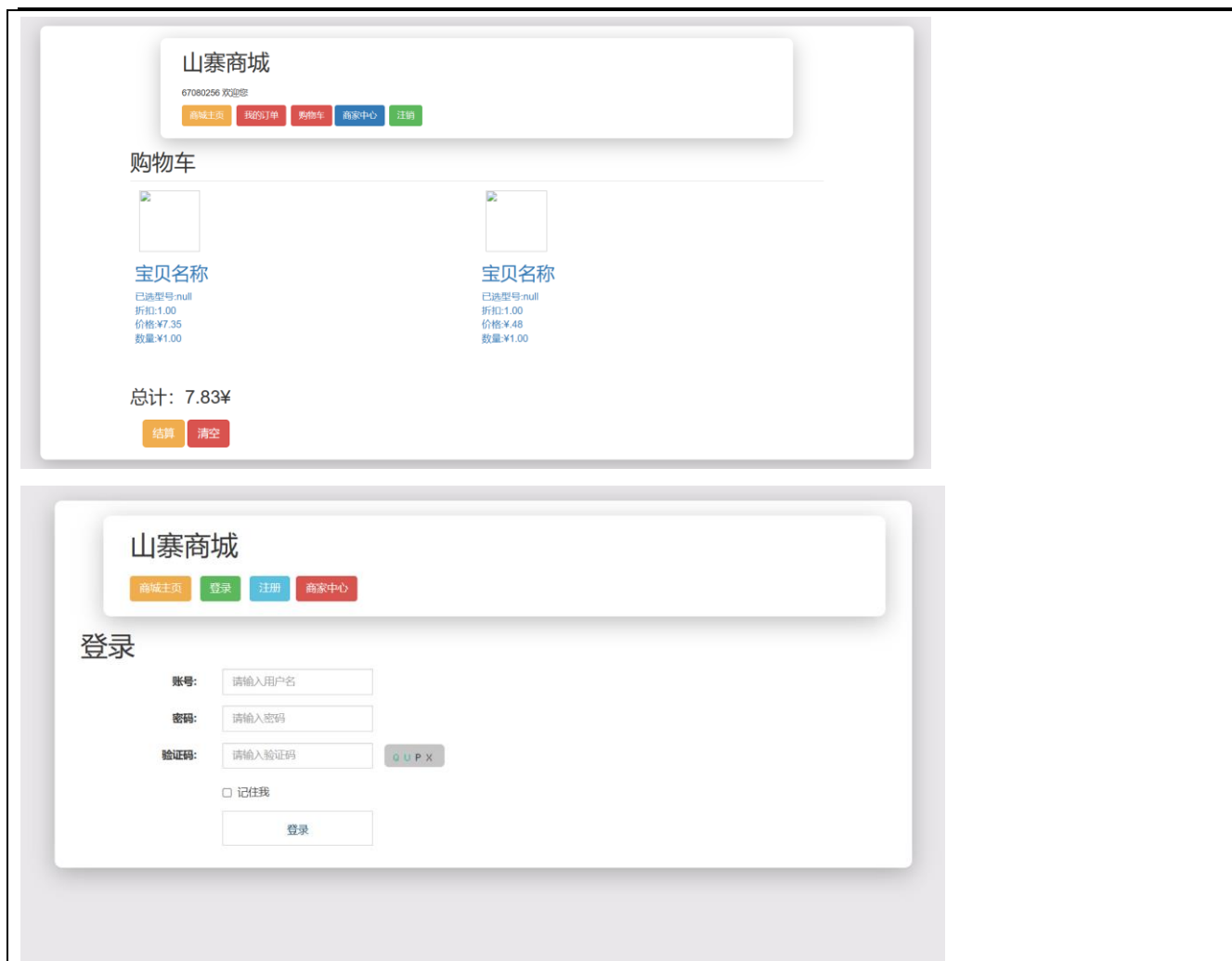
OR (vendor.vendor_phone=uid AND vendor.vendor_password=pwd);

END
```

3.5.9 应用系统设计

在最后的系统数据库实现过程中，我们使用了 java 作为数据库的呈现高级语言，实现的效果呈现如下图所示：





四. 遇到的主要问题及解决的办法

- 1.在开始时我们对如何建立一个网上购物商城进行了讨论，该建立几个表，表与表之间的逻辑关系是怎样等问题进行了探索，最后完成了上述的概念逻辑结构设计，为实际建立数据库留下了清晰的指导方向。
- 2.对表属性的内容存在争议，为了能够更好的贴合网上购物商城的实际，我们对表的属性定义进行了斟酌与思考，最后删除了一些不合适的属性，也增加了一些更贴合实际的属性元素。
- 3.对 navicat 的使用不熟悉，例如出现了如何创建数据库，如何导入数据库等问题；我们队伍也是通过相互帮助，网上搜集各种资料来熟悉了 navicat 的使用。
- 4.在 web 的实现上也有许多问题，例如 tomcat 服务器配置，数据库与 web 的连接等问题。我们也是通过上网搜寻相应资料最后实现了 web 展示。

五. 思考与感想

1.数据库设计是指对于一个给定的应用环境，构造优化的数据库逻辑模式和物理结构，并据此建立数据库及其应用系统，使之能够有效地存储和管理数据，满足各种用户的应用需求，包括信息管理要求和数据操作要求。我们通过真实的数据库设计经历，对其有了更加深刻的了解。明确需求，构造数据库逻辑模式是数据库结构设计重中之重，需求不明确会导致产品最后的反复修改甚至无法交工，这将会大大增加数据库设计的工作量。其次物理结构的实现是必不可少的，否则再完美的逻辑结构也无法投入商业使用，这就必须要求我们掌握物理结构设计的相关能力，能够做出相应产品并实现用户需求。

2.经过数据库课程的学习，我们掌握了许多数据库的基础知识，例如使用基于 E-R 模型的数据库设计方法；在进行物理结构设计之前必须要完成需求分析阶段，概念结构设计阶段，逻辑结构设计阶段这三个重要阶段；SQL 语句基础等等数据库内容。这些重要的数据库基础知识不仅成功的指导了我们的数据库课程设计，也为我们日后的数据库实际工作打下了坚实的基础。数据库作为重要的计算机专业知识，我们之后也会不断地进行数据库学习，从而全面的提高自身数据库能力，为社会创造更大的价值。