



天津大学
Tianjin University

淘宝数据库实践答辩展示

答辩人：周昊忆、黄涛、马宏伟

目录 CONTENT

01

需求分析与架构确立

Requirements Analysis and Architecture Establishment

02

E/R图设计

E/R diagram design

03

数据库实现

Database implementation

04

功能测试与结果展示

Functional testing and presentation of results

05

工作量统计

Workload statistics



天津大学
Tianjin University

PART·ONE

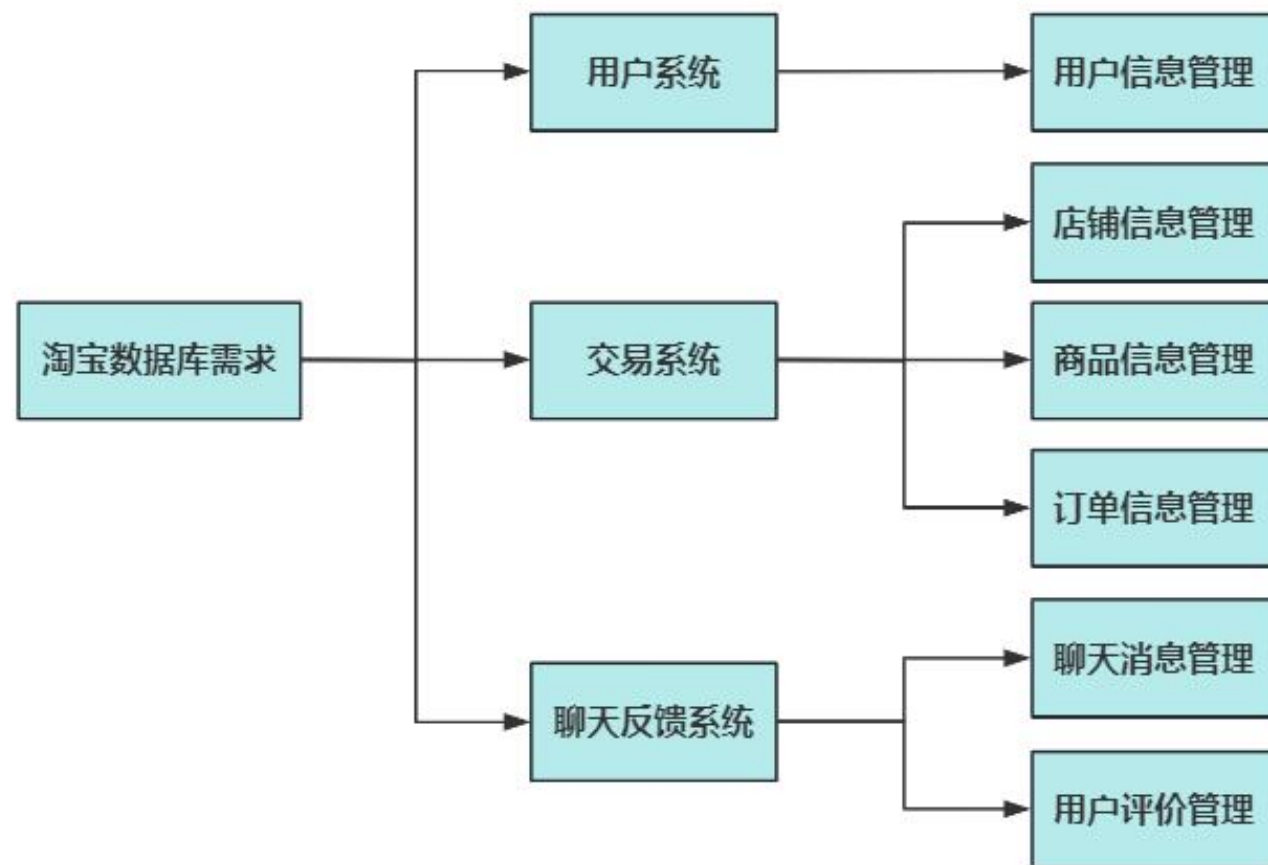
需求分析与架构确立

Requirements Analysis and Architecture Establishment

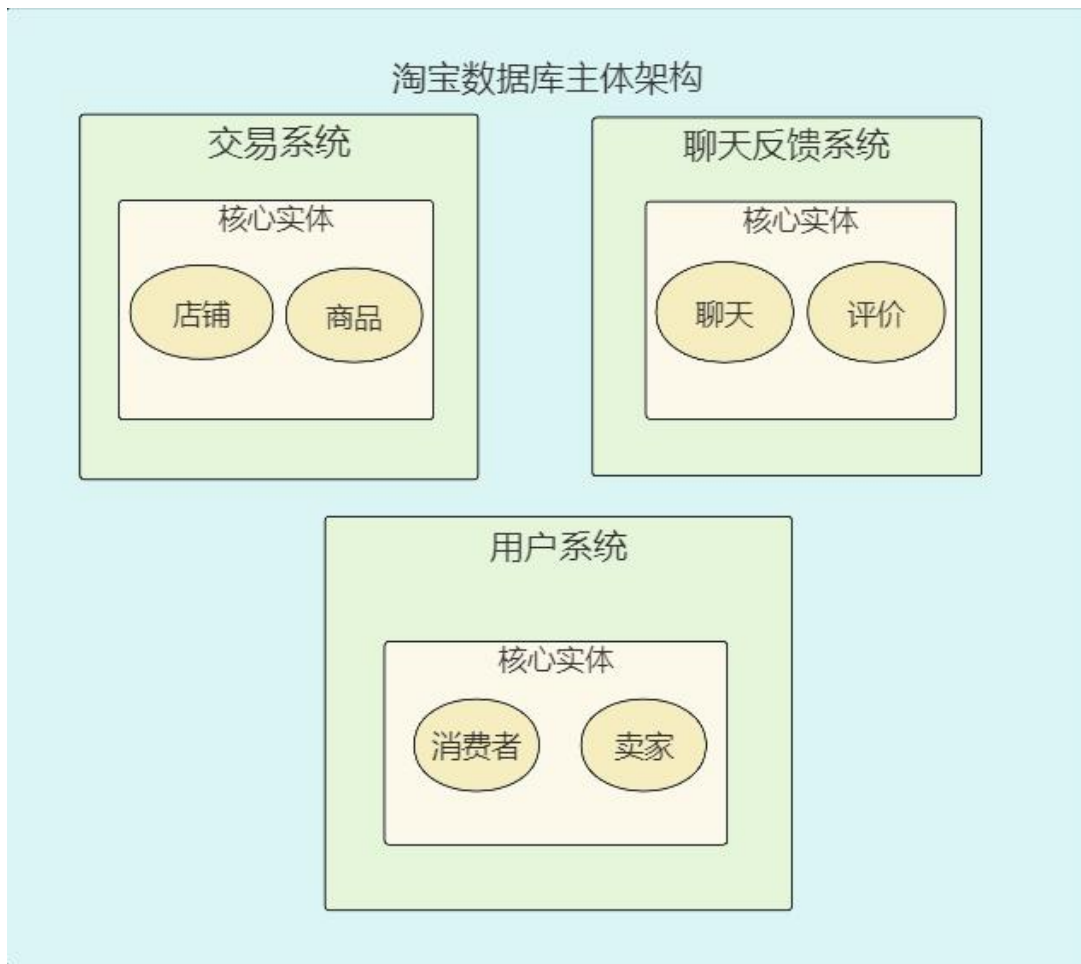
需求分析

核心功能需求分析

- 消费者：能够浏览、购买商品，将商品加入购物车等
- 卖家：能够管理店铺，上架、下架商品，管理订单，跟踪物流情况等
- 消费者和卖家的联系：消费者能够通过客服聊天等进行反馈，能够对购买的商品进行评价



主体架构确立



- **用户系统**包括用户、消费者、卖家、红包、优惠券、淘宝历程、收货地址、档案等实体
- **交易系统**包括店铺、商品、商品子类、活动、店铺评分、商品物流等实体
- **聊天反馈系统**包括官方客服、用户评价、用户评论、问题、群组、对话消息、群组消息、请求等实体

部分细节分析展示

用户系统

- (1) 消费者可以[浏览商品](#)，淘宝数据库需要记录[浏览的商品信息](#)与[浏览时间](#)，并且能够根据[浏览时间](#)进行查询。浏览记录最多[保留30天](#)
- (2) 消费者可以将商品[加入购物车](#)，并能实时查看购物车的商品信息。购物车最多添加[120个商品](#)。
- (3) 消费者可以[收藏商品](#)，并在收藏商品后能按[收藏时间和分类](#)进行查询。
- (4) 消费者购买商品后能实时[检测订单状态](#)，查看物流信息。
- (5) 消费者也可以[关注店铺](#)，实时获得店铺最新消息，并能浏览店铺中所有商品。
- (6) 消费者可以有多个[收货地址](#)，在购买商品时可以自行选择收货地址。若未选择则使用[默认地址](#)。
- (7) 消费者能获得[红包、优惠券](#)，购买商品后可以[积累淘金币](#)，在购买商品时可以使用优惠券、淘金币进行优惠
- (8) 卖家需要区分[主账号子账号](#)，并且主账号能够管理子账号

交易系统

- (1) 用户可以[创建店铺](#)，同一个身份证仅能创建一家店铺。
- (2) 店铺可以[添加商品](#)，在添加商品时，可以指定商品的类目。
- (3) 用户可以给[店铺打分](#)。所有用户打分的平均值决定了该店铺的总体评分。
- (4) 用户可以对[商品给出评分](#)。
- (5) 图片以链接也就是字符串形式存储在数据库中。淘宝有自己的文件系统TFS来保存图片视频等，因此数据库里采用链接的形式存储图片视频，以节省数据库空间，提升查询效率。
- (6) 卖家可以为商品分配特定的[物流](#)，买家可以更换商品的物流公司。
- (7) [订单](#)有几个状态：待付款、待发货、待收货、待评价、待退款，订单完成后不会被删除，因为日后用户可能会查询之前买过的商品。待付款超出24小时后自动删除订单。

聊天反馈系统

- (1) 只有当一个消费者购买某样商品并且[确认收货](#)，才能对该商品进行[评价](#)，确认收货之前不能评价。
- (2) 消费者可以在交易完成以后的[15天内](#)进行评价。
- (3) 消费者在进行中评或者差评的评价以后，是可以[修改评价](#)为好评的，但只能在做出评价的[30天内](#)进行操作。若消费者给出的是中差评，其评价被删除或者修改以后，其追加的评论同时也会被删除。
- (4) [商品评价共享](#)：消费者可以在商品中看到其它消费者的评价，可以对消费者的评价进行评论。一个消费者可以对一条评价进行多条评论，一条评价可以被多个消费者评论，消费者的评论可以被消费者点赞。
- (5) [商品提问](#)：消费者可以以商品的质量、样式等内容向其它消费者进行提问，其它消费者可以在该商品中看到问题，而且可以回答问题。一个消费者可以提出多个问题，一个问题只能属于一个消费者。



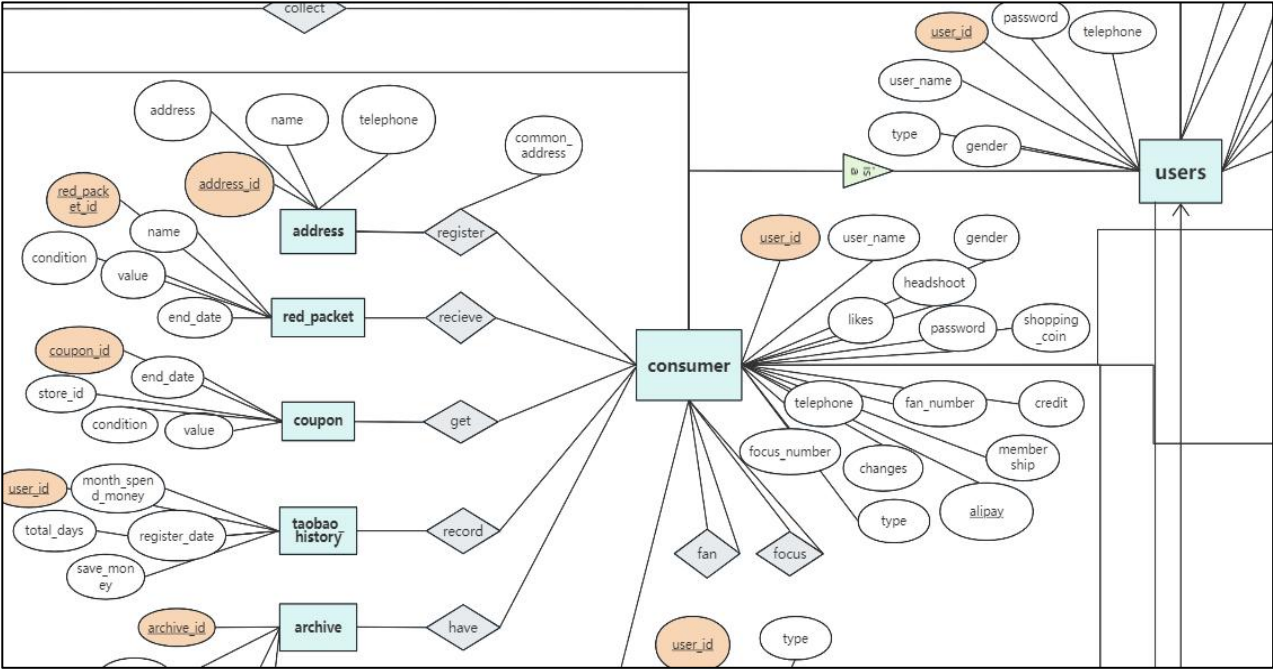
天津大学
Tianjin University

PART·TWO

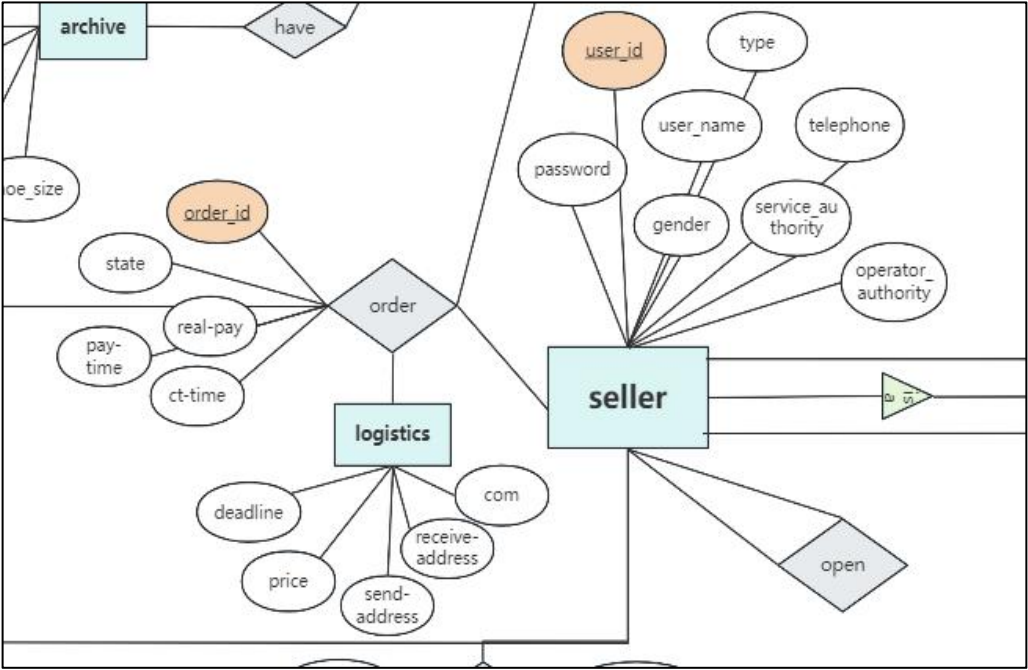
E/R图设计

E/R diagram design

ER图设计——用户系统

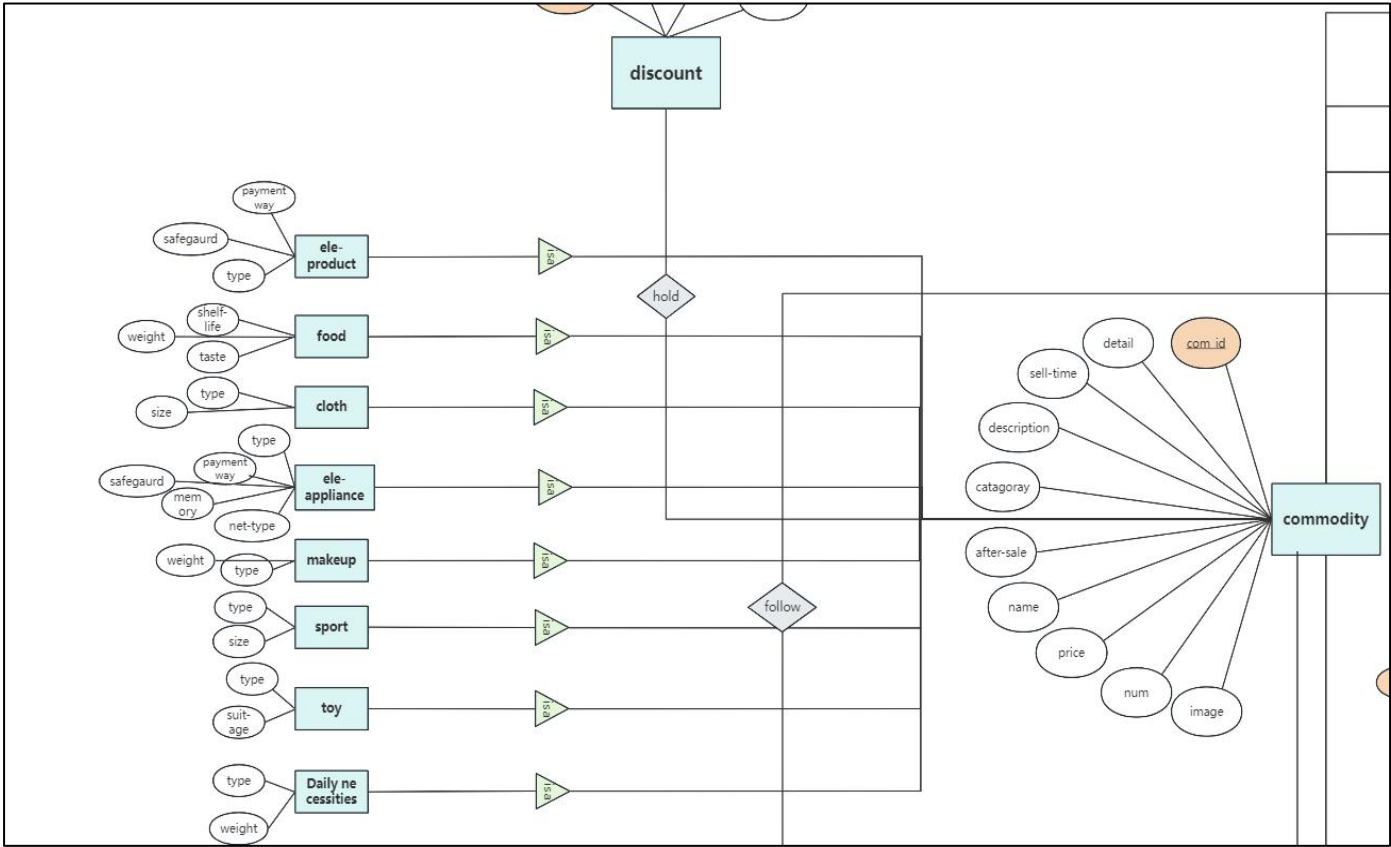


消费者

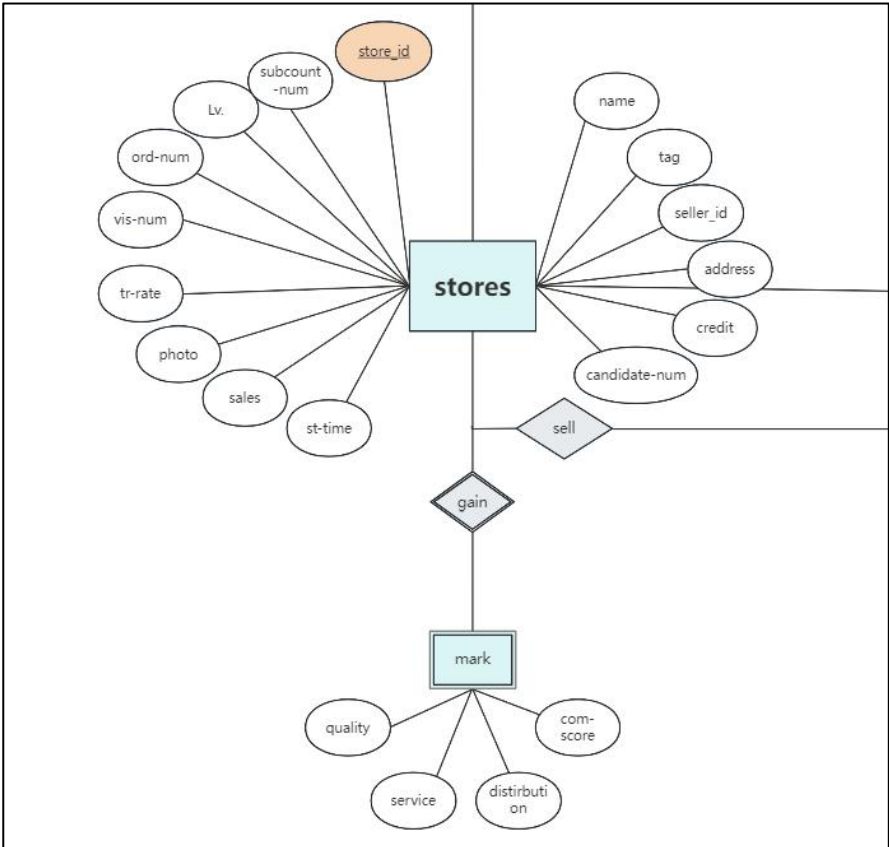


卖家

ER图设计——交易系统

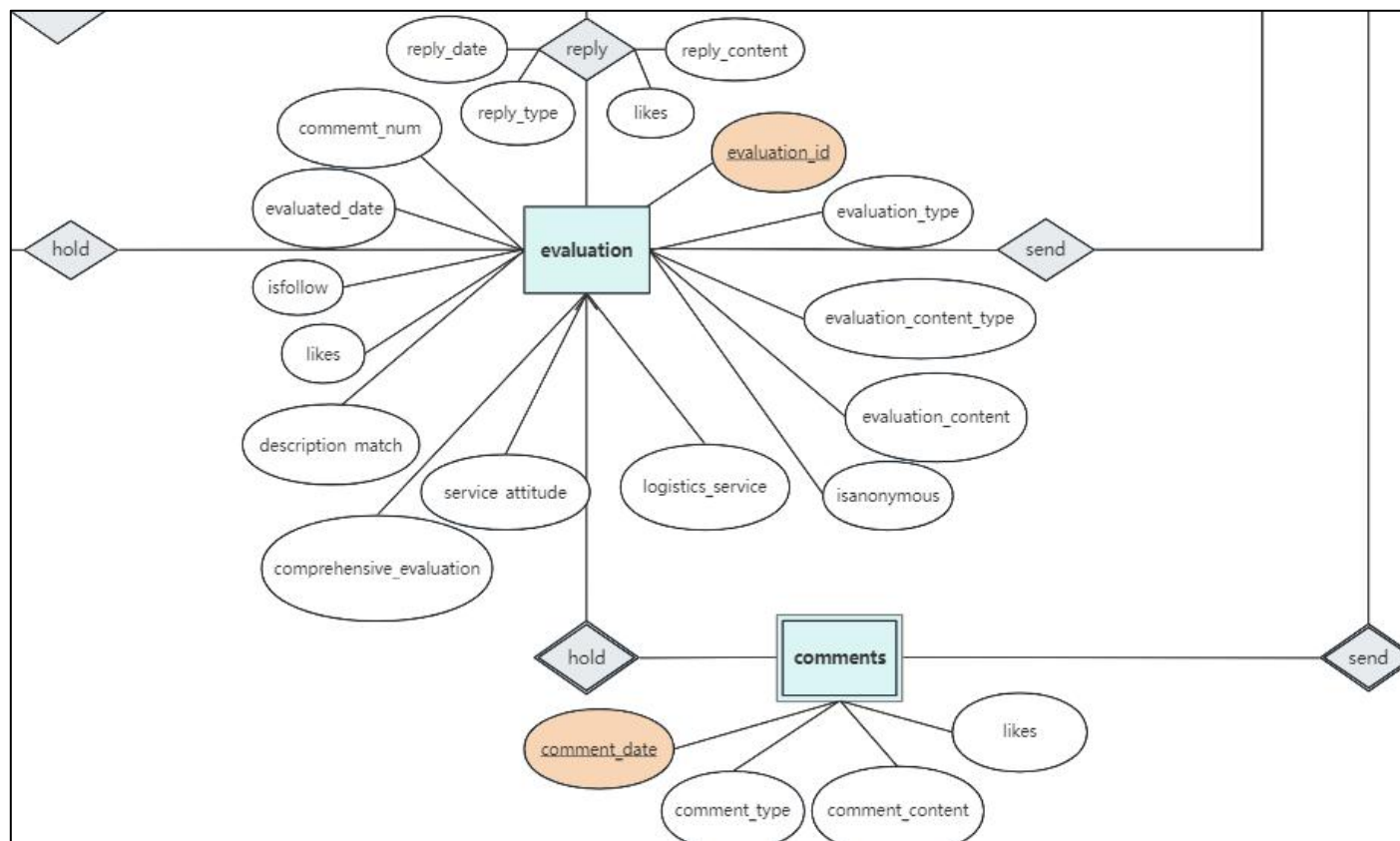


商品

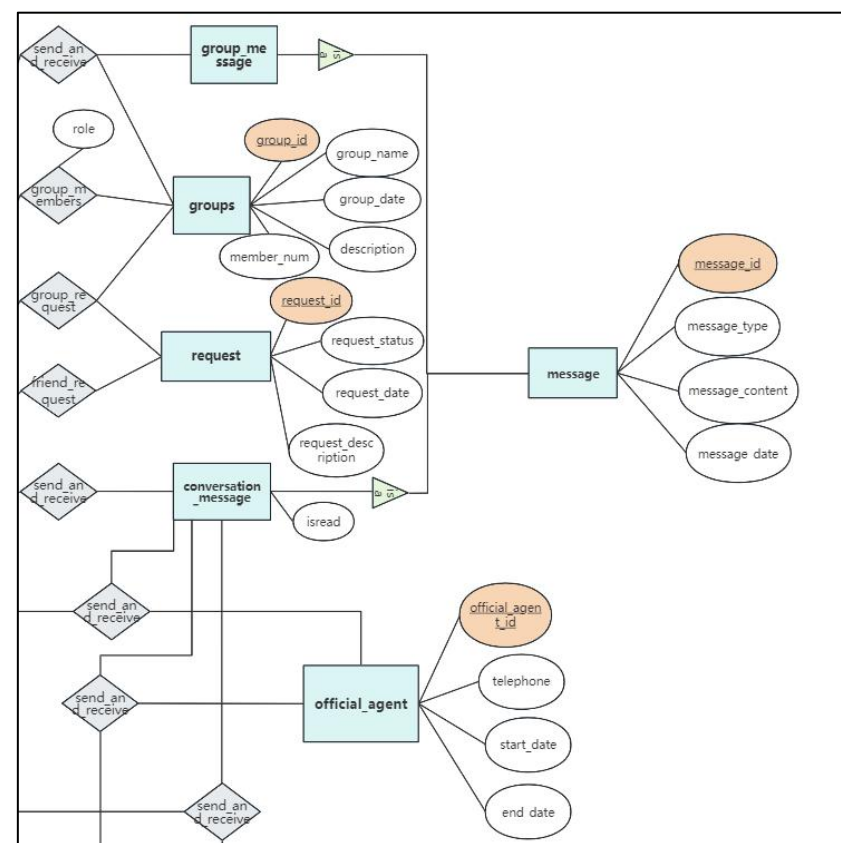


店铺

ER图设计——聊天反馈系统



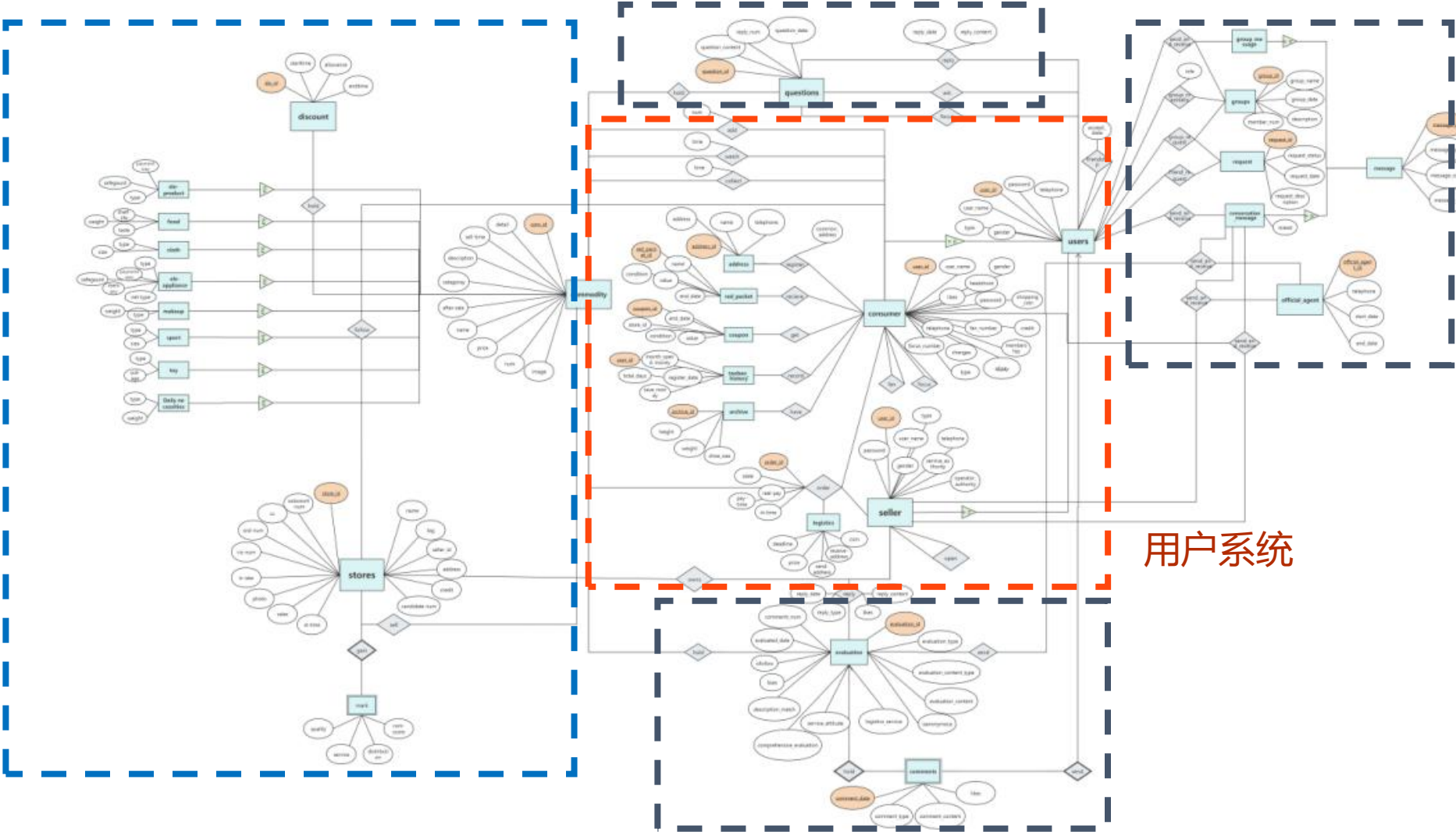
评价



聊天

ER图设计——总体设计

交易系统



聊天反馈系统

用户系统



天津大学
Tianjin University

PART·THREE

数据库实现

Database implementation

细化表结构

消费者表

列名	说明	数据类型	约束
user_id	用户账号	VARCHAR(20)	NOT NULL, PRIMARY KEY
user_name	用户名	VARCHAR(30)	NULL DEFAULT NULL
password_	密码	VARCHAR(20)	NOT NULL, 密码位数不少于 8 位
gender	性别	CHAR(1)	NOT NULL, 取值必须为"M"或"F"
telephone	电话	CHAR(11)	NOT NULL
membership	会员	BOOLEAN	NOT NULL
credit	信誉	INT	NOT NULL, 正数
headshot	头像	VARCHAR(500)	NULL DEFAULT NULL
fan_number	粉丝数目	INT	NOT NULL, 正数
focus_number	关注人数	INT	NOT NULL, 正数
likes	获赞数目	INT	NOT NULL, 正数
change	零钱	DOUBLE PRECISION	NOT NULL, 正数
shopping_coin	淘金币	INT	NOT NULL, 正数
alipay	支付宝号	VARCHAR(20)	NULL DEFAULT NUL
type_	类别	VARCHAR(10)	NOT NULL, 类型为"comsumer"
shopping_cart	购物车	VARCHAR(3000)	NULL DEFAULT NULL

数据存储形式：图片以链接（字符串）形式存储于数据库中，以此节省存储空间并提升查询效率。

优惠券关系表

列名	说明	数据类型	约束
coupon_id	优惠券标号	SERIAL	NOT NULL, PRIMARY KEY
store_id	店铺标号	VARCHAR(20)	NOT NULL, FOREIGN KEY
value	价值	DOUBLE PRECISION	NOT NULL
condition	使用条件	VARCHAR(20)	NOT NULL
end_date	截至日期	DATE	NOT NULL

规定主键外键，为每个属性设计约束

表项命名：

- 实体表以实体命名，关系表以关系名称命名。
- 对于联系两张表的中间关系表，名字以下划线间隔（表一_两者关系_表二）命名。
- 字段命名依据实际名称命名，字段长度结合实际应用设置。

SQL语句-建表

```
335 -- 创建消费者表
336 CREATE TABLE consumer (
337     user_id VARCHAR(20) NOT NULL PRIMARY KEY,
338     user_name VARCHAR(30) NULL DEFAULT NULL,
339     password_ VARCHAR(20) NOT NULL CHECK (LENGTH(password_) >= 8),
340     gender CHAR(1) NOT NULL CHECK (gender IN ('M', 'F')),
341     telephone CHAR(11) NOT NULL,
342     membership BOOLEAN NOT NULL,
343     credit INT NOT NULL CHECK (credit >= 0),
344     headshot VARCHAR(500) NULL DEFAULT NULL, -- 存储用户头像的地址
345     fan_number INT NOT NULL CHECK (fan_number >= 0),
346     focus_number INT NOT NULL CHECK (focus_number >= 0),
347     likes INT NOT NULL CHECK (likes >= 0),
348     change DOUBLE PRECISION NOT NULL CHECK (change >= 0), -- 零钱
349     shopping_coin INT NOT NULL CHECK (shopping_coin >= 0),
350     alipay VARCHAR(20) NULL DEFAULT NULL,
351     type_ VARCHAR(10) NOT NULL, CHECK (type_ = 'consumer'),
352     shopping_cart VARCHAR(3000) NULL DEFAULT NULL
353 );
```

创建消费者表

```
375 -- 创建优惠券表
376 CREATE TABLE coupon (
377     coupon_id SERIAL NOT NULL PRIMARY KEY,
378     store_id VARCHAR(20) NOT NULL,
379     value_ DOUBLE PRECISION NOT NULL,
380     condition_ VARCHAR(20) NOT NULL,
381     end_date DATE NOT NULL,
382     FOREIGN KEY (store_id) REFERENCES store (store_id)
383 );
```

创建优惠券关系表

SQL语句-视图

商品子表视图

```
733 --视图
734 --商品各种类别的视图
735 CREATE VIEW com_food AS SELECT commodity.com_name, commodity.price, commodity.dis_price, commodity.description, commodity.detail, commodity.com_image, shelf_life, weight
736 FROM commodity join food on commodity.com_id = food.com_id
737 where commodity.category = 'food';
738 DROP VIEW com_food;
739 CREATE VIEW com_cloth AS SELECT commodity.com_name, commodity.price, commodity.dis_price, commodity.description, commodity.detail, commodity.com_image, type, size
740 FROM commodity join cloth on commodity.com_id = cloth.com_id
741 where commodity.category = 'cloth';
742 DROP VIEW com_cloth;
743 CREATE VIEW com_ele_appliment AS SELECT commodity.com_name, commodity.price, commodity.dis_price, commodity.description, commodity.detail, commodity.com_image, paymentway
744 FROM commodity join ele_appliment on commodity.com_id = ele_appliment.com_id
745 where commodity.category = 'ele_appliment';
746 DROP VIEW com_ele_appliment;
747 CREATE VIEW com_ele_product AS SELECT commodity.com_name, commodity.price, commodity.dis_price, commodity.description, commodity.detail, commodity.com_image, paymentway
748 FROM commodity join ele_product on commodity.com_id = ele_product.com_id
749 where commodity.category = 'ele_product';
750 DROP VIEW com_ele_product;
751 CREATE VIEW com_makeup AS SELECT commodity.com_name, commodity.price, commodity.dis_price, commodity.description, commodity.detail, commodity.com_image, type, weight
752 FROM commodity join makeup on commodity.com_id = makeup.com_id
753 where commodity.category = 'makeup';
754 DROP VIEW com_makeup;
755 CREATE VIEW com_sport AS SELECT commodity.com_name, commodity.price, commodity.dis_price, commodity.description, commodity.detail, commodity.com_image, type, size
756 FROM commodity join sport on commodity.com_id = sport.com_id
757 where commodity.category = 'sport';
758 DROP VIEW com_sport;
759 CREATE VIEW com_toy AS SELECT commodity.com_name, commodity.price, commodity.dis_price, commodity.description, commodity.detail, commodity.com_image, type, suit_age
760 FROM commodity join toy on commodity.com_id = toy.com_id
761 where commodity.category = 'toy';
762 DROP VIEW com_toy;
763 CREATE VIEW com_daily_necessarity AS SELECT commodity.com_name, commodity.price, commodity.dis_price, commodity.description, commodity.detail, commodity.com_image, type
764 FROM commodity join daily_necessarity on commodity.com_id = daily_necessarity.com_id
765 where commodity.category = 'daily_necessarity';
766 DROP VIEW com_daily_necessarity;
767 --店铺所拥有的商品创建视图(改store_id)
768 CREATE VIEW com_of_store AS SELECT com_name, category, num, price, dis_price, description, detail, com_image, after_sale, up_time
769 FROM commodity join sell on commodity.com_id = sell.com_id
770 where store_id = '012345678';
771 DROP VIEW com_of_store;
```

视图面向数据库中最核心的属性

- 简化数据访问
- 提高操作效率,
- 提供更加定制化的数据展示
- 提供数据访问的权限控制, 保护基础数据表的安全

SQL语句-函数

购买商品

购买商品时先计算实际支付的价格，再向订单表插入数据

```
1060 -- 直接购买商品
1061 DROP FUNCTION buy_commodity_directly(IN product_id VARCHAR(20),IN user_id VARCHAR(20),IN order_id VARCHAR(20),IN buy_num INT);
1062 CREATE OR REPLACE FUNCTION buy_commodity_directly(
1063     IN product_id VARCHAR(20),
1064     IN user_id VARCHAR(20),
1065     IN order_id VARCHAR(20),
1066     IN buy_num INT
1067 ) RETURNS VOID AS $$
1068 DECLARE
1069     total_payment DOUBLE PRECISION;
1070 BEGIN
1071     UPDATE commodity SET num = num - buy_num WHERE com_id = product_id;
1072     SELECT calculate_price(product_id,user_id,total_payment) INTO total_payment;
1073     INSERT INTO order_ VALUES (order_id,product_id,user_id,total_payment,CURRENT_TIMESTAMP,CURRENT_TIMESTAMP,'WAITDELIVER') ;
1074     RAISE INFO 'Calculated Total Payment: %', total_payment;
1075     RAISE INFO 'Insert order';
1076     RAISE INFO 'update Commodity num';
1077 END;
1078 $$ LANGUAGE plpgsql;
1079
1080 INSERT INTO red_packet
1081 VALUES(1,'无门槛红包',2,'无门槛','2023-12-25');
1082
1083 INSERT INTO consumer_recieve_redpacket
1084 VALUES ('tb12345',1);
1085 DELETE FROM order_ WHERE order_id = 'com12347tb12345';
1086 DELETE FROM logistics WHERE com_id = 'com12347';
1087 UPDATE commodity SET num = 1000 WHERE com_id = 'com12347';
1088
1089 SELECT buy_commodity_directly('com12347', 'tb12345', 'com12347tb12345', 1);
1090 SELECT * FROM order_;
1091
1092 SELECT * FROM commodity WHERE com_id = 'com12347';
1093 SELECT * FROM food WHERE com_id = 'com12347';
1094 SELECT * FROM logistics;
1095 SELECT * FROM red_packet;
1096 SELECT * FROM consumer_recieve_redpacket;
```


SQL语句-函数

计算商品价格

根据商品的折扣，用户拥有的红包，计算最终价格，同时删除红包表中已被使用的红包信息

```
1028 -- 函数
1029 -- 计算商品价格
1030 DROP FUNCTION calculate_price( IN product_id VARCHAR(20),IN user_id VARCHAR(20),INOUT total_payment DOUBLE PRECISION);
1031 CREATE OR REPLACE FUNCTION calculate_price(
1032     IN product_id VARCHAR(20),
1033     IN user_id2 VARCHAR(20),
1034     INOUT total_payment DOUBLE PRECISION
1035 ) AS $$
1036 DECLARE
1037     product_price DOUBLE PRECISION;
1038     discount_price DOUBLE PRECISION;
1039     red_packet_discount DOUBLE PRECISION;
1040     red_packet_id2 INT;
1041 BEGIN
1042     SELECT price INTO product_price FROM commodity WHERE com_id = product_id;
1043     SELECT dis_price INTO discount_price FROM commodity WHERE com_id = product_id;
1044     SELECT red_packet_id INTO red_packet_id2 FROM consumer_recieve_redpacket WHERE user_id = user_id2;
1045     SELECT value_ INTO red_packet_discount FROM red_packet WHERE red_packet_id = red_packet_id2;
1046     RAISE INFO 'origin price: %', product_price;
1047     RAISE INFO 'discount_price: %', discount_price;
1048     RAISE INFO 'red_packet_discount: %', red_packet_discount;
1049     total_payment := product_price - red_packet_discount - discount_price;
1050     -- 更新红包表
1051     DELETE FROM consumer_recieve_redpacket WHERE user_id = user_id;
1052     DELETE FROM red_packet WHERE red_packet_id = red_packet_id;
1053
1054     RETURN;
1055 END;
1056 $$ LANGUAGE plpgsql;
```

SQL语句-触发器

订单表物流表触发器

订单表物流表触发器 (order_insert_trigger):

- 与订单表(order_)相关联。
- 当在订单表中插入新记录（即创建新订单）后，触发器order_insert_trigger()会被执行。
- 当新订单的商品ID(com_id)插入到物流表(logistics)中，表示该商品需要物流处理。

```
810 --订单表物流表触发器
811
812 CREATE OR REPLACE FUNCTION order_insert_trigger()
813 RETURNS TRIGGER AS $$
814 BEGIN
815     INSERT INTO logistics (com_id)
816     VALUES (NEW.com_id);
817     RETURN NEW;
818 END;
819 $$ LANGUAGE plpgsql;
820
821 CREATE TRIGGER insert_order_trigger
822 AFTER INSERT ON order_
823 FOR EACH ROW
824 EXECUTE FUNCTION order_insert_trigger();
```

商品大表与小表触发器

- 与商品主表（假设表名为commodity）相关联。
- 当在商品主表中插入新记录后，触发器insert_comsubtable_function()会被执行。
- 触发器函数根据新商品的类别(NEW.category)，将商品信息插入到相应的商品子表中，以实现数据的分类存储。

```
847 --商品大表与小表触发器
848 CREATE OR REPLACE FUNCTION insert_comsubtable_function()
849 RETURNS TRIGGER AS $$
850 BEGIN
851     -- 判断商品类别
852     IF NEW.category = 'food' THEN
853         -- 更新food
854         INSERT INTO food(com_id,com_name,category,price,dis_price,num,detail,after_sale,up_time,com_image)
855         VALUES (NEW.com_id,NEW.com_name,NEW.category,NEW.price,NEW.dis_price,NEW.num,NEW.description,NEW.detail,NEW.after_sale,NEW.up_time,NEW.com_image);
856     ELSIF NEW.category = 'cloth' THEN
857         -- 更新cloth
858         INSERT INTO cloth(com_id,com_name,category,price,dis_price,num,detail,after_sale,up_time,com_image)
859         VALUES (NEW.com_id,NEW.com_name,NEW.category,NEW.price,NEW.dis_price,NEW.num,NEW.description,NEW.detail,NEW.after_sale,NEW.up_time,NEW.com_image);
860     ELSIF NEW.category = 'ele_appliment' THEN
861         -- 更新ele_appliment
862         INSERT INTO ele_appliment(com_id,com_name,category,price,dis_price,num,detail,after_sale,up_time,com_image)
863         VALUES (NEW.com_id,NEW.com_name,NEW.category,NEW.price,NEW.dis_price,NEW.num,NEW.description,NEW.detail,NEW.after_sale,NEW.up_time,NEW.com_image);
864     ELSIF NEW.category = 'ele_product' THEN
865         -- 更新ele_product
866         INSERT INTO ele_product(com_id,com_name,category,price,dis_price,num,detail,after_sale,up_time,com_image)
867         VALUES (NEW.com_id,NEW.com_name,NEW.category,NEW.price,NEW.dis_price,NEW.num,NEW.description,NEW.detail,NEW.after_sale,NEW.up_time,NEW.com_image);
868     ELSIF NEW.category = 'makeup' THEN
869         -- 更新makeup
870         INSERT INTO makeup(com_id,com_name,category,price,dis_price,num,detail,after_sale,up_time,com_image)
871         VALUES (NEW.com_id,NEW.com_name,NEW.category,NEW.price,NEW.dis_price,NEW.num,NEW.description,NEW.detail,NEW.after_sale,NEW.up_time,NEW.com_image);
872     ELSIF NEW.category = 'sport' THEN
873         -- 更新sport
874         INSERT INTO sport(com_id,com_name,category,price,dis_price,num,detail,after_sale,up_time,com_image)
875         VALUES (NEW.com_id,NEW.com_name,NEW.category,NEW.price,NEW.dis_price,NEW.num,NEW.description,NEW.detail,NEW.after_sale,NEW.up_time,NEW.com_image);
876     ELSIF NEW.category = 'toy' THEN
877         -- 更新toy
878         INSERT INTO toy(com_id,com_name,category,price,dis_price,num,detail,after_sale,up_time,com_image)
879         VALUES (NEW.com_id,NEW.com_name,NEW.category,NEW.price,NEW.dis_price,NEW.num,NEW.description,NEW.detail,NEW.after_sale,NEW.up_time,NEW.com_image);
880     ELSIF NEW.category = 'daily_necessarity' THEN
881         -- 更新daily_necessarity
882         INSERT INTO daily_necessarity(com_id,com_name,category,price,dis_price,num,detail,after_sale,up_time,com_image)
883         VALUES (NEW.com_id,NEW.com_name,NEW.category,NEW.price,NEW.dis_price,NEW.num,NEW.description,NEW.detail,NEW.after_sale,NEW.up_time,NEW.com_image);
884     ELSE
```



天津大学
Tianjin University

PART·FOUR

功能测试与结果展示

Functional testing and presentation of results

实体表查询展示

评价表

	evaluation_id [PK] character varying	evaluation_cc character var	evaluation_content character varying (500)	isanonymous boolean	evaluation_ty character (1)	evaluated_date timestamp without tim	isfollow boolean	likes integer	comment_nu integer	comprehensi integer	description_n integer	logistics_serv integer	service_attitu integer
1	ev00000000000001	文本	这件衣服的材质很好,穿起来...	false	好	2023-02-22 14:00:00	false	0	0	5	5	5	5
2	ev00000000000002	文本	这个充电宝充电慢, 还会漏电	false	差	2023-03-12 15:30:53	false	0	0	1	1	3	3
3	ev00000000000003	文本	这个薯片很好吃, 值得一买	true	好	2023-04-11 08:12:22	false	0	0	5	5	4	5
4	ev00000000000004	文本	这件衣服真的很好看, 还便宜	false	好	2023-02-23 10:10:55	true	25	30	5	5	5	5
5	ev00000000000005	图片	D:\image\1.jpg	false	好	2023-03-23 11:25:35	false	10	3	5	4	4	5
6	ev00000000000006	视频	D:\video\1.mp4	false	好	2023-01-25 16:55:45	false	0	0	5	5	5	5

消息表

对象 message @taobao_database.public (...)

表配置文件 开始事务 单元格编辑器 筛选 & 排序 列 数据分析 工具

message_id varchar(20)	message_type varchar(10)	message_content varchar(500)	message_date timestamp(6)
ms00000000000001	文本	你吃饭了吗?	2024-12-10 12:00:00
ms00000000000002	文本	还没有	2024-12-10 12:00:30
ms00000000000003	文本	今天布置了什么作业了?	2024-12-12 18:00:00
ms00000000000004	文本	数值计算的编程作业	2024-12-12 18:30:53
ms00000000000005	文本	edge detection	2024-12-12 18:44:24
ms00000000000006	文本	一会去哪吃?	2024-12-10 12:00:47
ms00000000000007	文本	兰园二楼	2024-12-10 12:01:20
ms00000000000008	图片	D:\image\2.jpg	2024-12-12 18:50:55
ms00000000000009	视频	D:\video\2.mp4	2024-12-12 18:51:56

约束效果展示



天津大学
Tianjin University

```
1068 ALTER TABLE consumer
1069 ALTER COLUMN headshot TYPE VARCHAR(500);
1070 INSERT INTO consumer
1071 VALUES ('tb12345', 'Daming', '123456', 'F', '13981367234', 'T', 5, 'http://img.alicdn.com/sns_logo/i2/2212109088345/01CN01rsPVu02BW3rihzm8e_!!2212109088
1072
1073
```

Data Output Messages Notifications

ERROR: 错误: 关系 "consumer" 的新列违反了检查约束 "consumer_password__check"

DETAIL: 失败, 行包含(tb12345, Daming, 123456, F, 13981367234, t, 5, http://img.alicdn.com/sns_logo/i2/2212109088345/01CN01rsPVu02BW3..., 0, 0, 5, 10.5, 500, 13981367234, consumer).

约束条件: 账号密码长度必须大于8

实现效果: 当插入数据违反约束条件时无法插入

实体表查询展示

```
773 SELECT * FROM com_food;
774 SELECT * FROM com_cloth;
775 SELECT * FROM com_ele_appliment;
```

Data Output 消息 通知

	com_name character varying (30)	price double precis	dis_price double precision	description character varying (50)	detail character varying (2000)	com_image character varying (500)	shelf_life integer	weight character varying (20)	taste character varying (20)
1	cookie	5.99	1	very delicious	Deliciously crisp an...	https://img.alicdn.co...	180	1kg	Original taste
2	River snail rice noodle	10.99	0	smelly but delicio...	A popular dish from ...	https://img.alicdn.co...	90	225g	Original taste
3	spicy strip	0.5	0	Taste of childhood	Spicy and addictive,...	https://img.alicdn.co...	360	50g	Original taste
4	Braised beef noodles	3.5	0	filling	Spicy and addictive,...	https://g-search3.alic...	90	225g	Braised beef
5	jelly	20.99	5.99	chewy and delici...	Delightfully wobbly ...	https://img.alicdn.co...	360	100g	orange

查询商品表上的食品视图，展示数据库中最核心的信息
例如：商品名称，商品价格，折扣价格，商品描述等

函数执行效果展示

	red_packet_id [PK] integer	name_ character varying (30)	value_ double precision	condition_ character varying (20)	end_date date
1	1	无门槛红包	10	无门槛	2023-12-25

	red_packet_id [PK] integer	name_ character varying (30)	value_ double precision	condition_ character varying (20)	end_date date

```
信息: origin price: 5.99
信息: discount_price: 1
信息: red_packet_discount: 2
信息: Calculated Total Payment: 2.99
信息: Insert order
信息: update Commodity num








Successfully run. Total query runtime: 47 msec.
1 rows affected.
```

执行“购买商品”函数后，观察到红包表数据被删除（使用红包），订单表和物流表均被插入数据，商品表中商品剩余数目更新








触发器效果展示

在更新商品大表的同时观察到对应的商品小表（例如食物表，订单表，物流表）被更新









商品大表

	com_id [PK] character varying (20) 	com_name character varying (30) 	category character varying (20) 	price double precision 	dis_price double precision 	num integer 	description character varying (500) 	detail character varying (2000)
1	com12347	cookie	food	5.99	1	999	very delicious	Deliciously crisp and golden, these cookies are the perfect blend


食物表

	com_id [PK] character varying (20) 	com_name character varying (30) 	category character varying (20) 	price double precision 	dis_price double precision 	num integer 	description character varying (500) 	detail character varying (2000)
1	com12347	cookie	food	5.99	1	999	very delicious	Deliciously crisp and golden, these cookies are the perfect blend

订单表

	order_id [PK] character varying (20) 	com_id character varying (20) 	user_id character varying (20) 	real_pay double precision 	ct_time timestamp without time zone 	pay_time timestamp without time zone 	state character varying (20) 	pickup_code character varying (20) 
1	com12347tb12345	com12347	tb12345	2.99	2023-12-14 21:44:53.67256	2023-12-14 21:44:53.67256	WAITDELIVER	[null]

物流表

	com_id [PK] character varying (20) 	deadline timestamp without time zone 	price double precision 	comp_name character varying (100) 	send_addr character varying (50) 	receive_addr character varying (50) 
1	com12347	[null]	[null]	[null]	[null]	[null]



天津大学
Tianjin University

PART·FIVE

工作量统计

Workload statistics

报告整体框架与工作量统计

目录	
一、分组情况	4
二、报告摘要	5
1. 用户系统	5
2. 交易系统	6
3. 聊天反馈系统	6
三、需求分析	6
1. 数据分析	6
1.1 用户系统	6
1.2 交易系统	7
1.3 聊天反馈系统	9
2. 基础功能分析	10
2.1 用户系统	10
2.2 交易系统	11
2.3 聊天反馈系统	11
四、方案设计	12
1 数据库设计规则	12
2 概念设计	13
2.1 三大系统	13
2.2 数据库中各个实体的表示以及 ER 图	16
2.3 系统整体 ER 图	27
2 逻辑结构设计	27
2.1 逻辑转换	27
2.2 细化表结构	30
4 创建视图	40
5 创建触发器	40
6 创建函数	41
五、结果展示	42
1. 用户系统部分实体表展示	42
1.1 用户表	42
1.2 消费者表	42
1.3 卖家表	43
2. 交易系统部分实体表、关系表展示	43
2.1 商品表	43
2.2 店铺表	43
2.3 食品表	44
2.4 店铺商品关系表	44
3. 聊天反馈系统部分实体表、关系表展示	44
3.1 消息总表	44
3.2 对话消息总表	45
3.3 群组消息总表	45
3.4 消费者评价总表	45
3.5 好友聊天消息索引	46
4. 约束	46
5. 视图	46

6. 函数	47
6.1 函数调用前	47
6.2 函数执行后	48
7. 触发器	49
六、总结	50
1. 工作量	50
2. 复杂度	50
3. 新颖性	50
4. 难度	51
5. 实践中用到的知识	51
6. 实践中遇到的困难和解决方法	52
7. 体会与建议	52
附录	52
1. 交易系统 sql 建表	53
2. 用户系统 sql 建表	61
3. 聊天反馈系统 sql 建表	67
4. 部分视图 sql 实现	73
5. 部分触发器 sql 实现	73
6. 部分函数 sql 实现	75

- 我们共创建了**57个表**，**32个实体表**，**23个关系表**，共计**上百个主键外键**，设计**12个视图**、**14个触发器**、**9个函数**
- 迭代了**数十版ER图**，后端贡献代码**数千行**
- 共撰写**76页**，共**17452字**的实验报告，细化表结构设计时列出数百行excel表



天津大学
Tianjin University

感谢您的观看

T H A N K Y O U F O R W A T C H I N G

答辩人：周昊忆、黄涛、马宏伟