

쇼핑몰 프로젝트

서종태



목차

1. DB 설계목적
2. 요구 사항 분석 및 엔티티 도출
3. ERD(개념적모델링)
4. 테이블 기술서
5. 논리적 모델링
6. 물리적 모델링
7. SQL문 작성



DB 설계 목적

인터넷 쇼핑몰을 창업하려는 사람들의 요구사항 분석서를 조사해서
DB를 설계



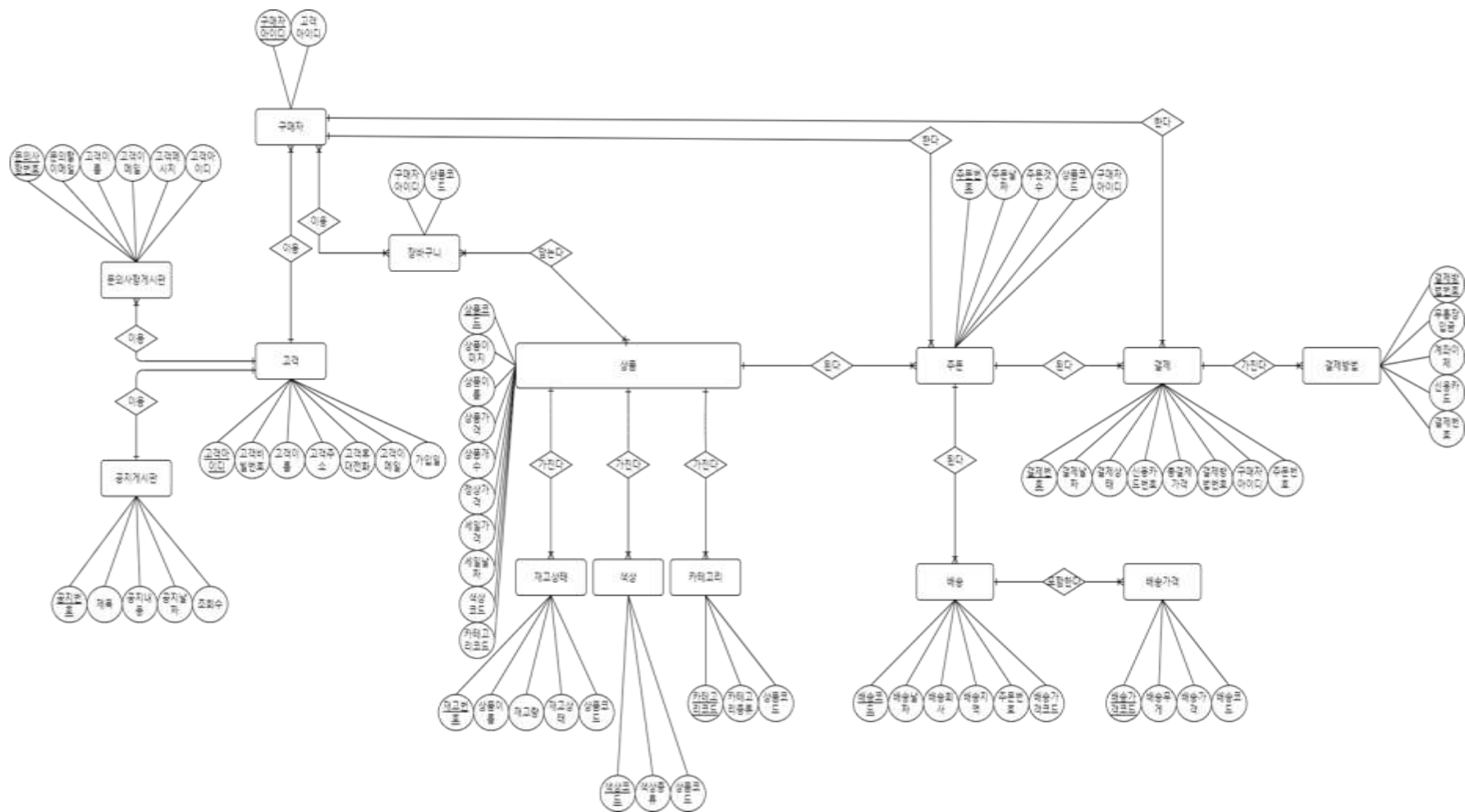
요구 사항 분석서

1. **고객**은 회원가입을 할수있다.
2. 고객은 **공지게시판**의 공지글을 확인할수 있고 **문의사항게시판**에 문의를 할수 있다.
3. 고객은 **상품**을 구매하고자 할때는 **구매자** 아이디가 있어야 한다.
4. 고객은 원하는 상품을 **장바구니**에 담을수 있다.
5. 상품별 **카테고리**(품목분류),**색상**(상품옵션),**재고상태**를 파악할수 있다.
6. 고객은 구매자아이디로 상품 주문 및 장바구니에 담은 상품을 **결제**할수 있다.
7. 고객은 **주문**에 대한 **배송업체** 및 **배송가격**을 알수 있다.
8. 고객이 주문한 상품을 결제하는데 여러가지 **결제방법**이 있다.
9. 고객은 주문에 대한 배송정보를 알수 있다.

빨간색 표시 : 엔티티



ERD 개념적 모델링





테이블 기술서

PDF 파일 별도 첨부



논리적 모델링

문의사항 게시판	
PK	문의사항번호
	문의할 이메일 고객이름 고객이메일 고객메시지 고객아이디(FK)

구매자	
PK	구매자아이디
	고객아이디(FK)

고객	
PK	고객아이디
	고객비밀번호 고객이름 고객주소 고객휴대전화 고객이메일 가입일

공지 게시판	
PK	공지번호
	제목 공지내용 공지날짜 조회수

장바구니	
	구매자아이디(FK) 상품코드(FK)

상품	
PK	상품코드
	상품이미지 상품명 상품가격 상품개수 정상가격 세일가격 세일날짜 색상코드(FK) 카테고리코드(FK)

주문	
PK	주문번호
	주문날짜 주문개수 상품코드(FK) 구매자아이디(FK)

결제	
PK	결제번호
	결제날짜 결제상태 신용카드번호 총결제가격 결제방법번호(FK) 구매자아이디(FK) 주문번호(FK)

결제방법	
PK	결제방법번호
	무통장입금 계좌이체 신용카드 결제번호(FK)

재고상태	
PK	재고번호
	상품명 재고량 재고상태 상품코드(FK)

색상	
PK	색상코드
	색상종류 상품코드(FK)

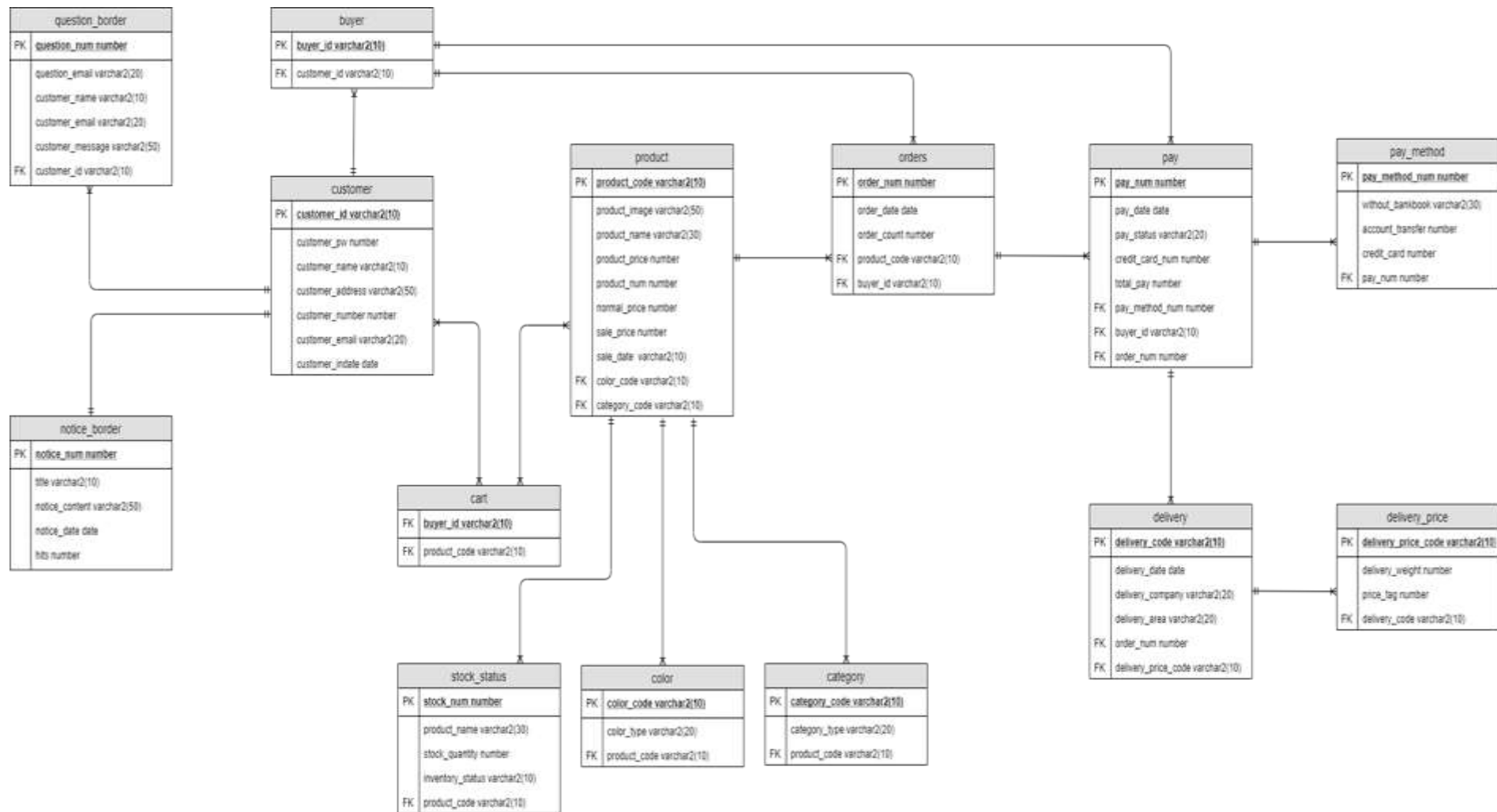
카테고리	
PK	카테고리코드
	카테고리종류 상품코드(FK)

배송	
PK	배송코드
	배송날짜 배송회사 배송지역 주문번호(FK) 배송가격코드(FK)

배송가격	
PK	배송가격코드
	배송루트 배송가격 배송코드(FK)



물리적 모델링





SQL문 작성

테이블 생성

문의사항 게시판

```
create table question_border(  
question_num number primary key,  
question_email varchar2(20) default 'www.egoing@naver.com',  
customer_name varchar2(10) not null,  
customer_email varchar2(20) not null,  
customer_message varchar2(50) not null,  
customer_id varchar2(10) constraint question_border_customer_id_fk references customer(customer_id));
```

고객

```
create table customer(  
customer_id varchar2(10) primary key,  
customer_pw number not null,  
customer_name varchar2(10) not null,  
customer_address varchar2(50) not null,  
customer_number number not null,  
customer_email varchar2(20) not null,  
customer_indate date default sysdate);
```

공지게시판

```
create table notice_border(  
notice_num number primary key,  
title varchar2(10) not null,  
notice_content varchar2(50) not null,  
notice_date date not null,  
hits number not null);
```

구매자

```
create table buyer(  
buyer_id varchar2(10) primary key,  
customer_id varchar2(10) constraint  
buyer_customer_id_fk references  
customer(customer_id));
```

장바구니

```
create table cart(  
buyer_id varchar2(10) constraint  
cart_buyer_id_fk references buyer(buyer_id),  
product_code varchar2(10) constraint  
cart_product_code_fk references product(product_code));
```

상품

```
create table product(  
product_code varchar2(10) primary key,  
product_image varchar2(50),  
product_name varchar2(30) not null,  
product_price number not null,  
product_num number not null,  
normal_price number not null,  
sale_price number not null,  
sale_date date not null);
```

```
alter table product  
add color_code varchar2(10);
```

```
alter table product  
add category_code varchar2(10);
```

```
alter table product  
add constraint product_color_code_fk foreign key(color_code) references color(color_code);
```

```
alter table product  
add constraint product_category_code_fk foreign key(category_code) references category(category_code);
```

```
alter table product  
modify sale_date varchar2(10);
```

```
alter table product  
rename column prodcut_price to product_price;
```

```
alter table product  
rename column prodcut_num to product_num;
```

카테고리

```
create table category(  
category_code varchar2(10) primary key,  
category_type varchar2(20) not null,  
product_code varchar2(10) constraint  
category_product_code_fk references product(product_code));
```


재고상태

```
create table stock_status(  
stock_num number primary key,  
product_name varchar2(30) not null,  
stock_quantity number not null,  
inventory_status varchar2(10) not null,  
product_code varchar2(10) constraint  
stock_status_product_code_fk references  
product(product_code));
```

색깔

```
create table color(  
color_code varchar2(10) primary key,  
color_type varchar2(10) not null,  
product_code varchar2(10) constraint  
color_product_code_fk references product(product_code));
```

```
alter table color  
modify color_type varchar2(20);
```

주문

```
create table orders(  
order_num number primary key,  
order_date date not null,  
order_count number not null,  
product_code varchar2(10) constraint  
order_product_code_fk references product(product_code),  
buyer_id varchar2(10) constraint  
order_buyer_id_fk references buyer(buyer_id));
```

배송

```
create table delivery(  
delivery_code varchar2(10) primary key,  
delivery_date date not null,  
delivery_company varchar2(20) not null,  
delivery_area varchar2(20) not null,  
order_num number constraint  
delivery_order_num_fk references orders(order_num));
```

```
alter table delivery  
add delivery_price_code varchar2(10);
```

```
alter table delivery
add constraint product_delivery_price_code_fk foreign key(delivery_price_code) references
delivery_price(delivery_price_code);
```

배송가격

delivery_price

```
create table delivery_price(
delivery_price_code varchar2(10) primary key,
delivery_weight number not null,
price_tag number not null,
delivery_code varchar2(10) constraint
delivery_p_delivery_code_fk references
delivery(delivery_code));
```

결제

```
create table pay(  
pay_num number primary key,  
pay_date date not null,  
pay_status varchar2(20) not null,  
credit_card_num number ,  
total_pay number not null,  
pay_method_num number,  
buyer_id varchar2(10) constraint  
pay_buyer_id_fk references buyer(buyer_id),  
order_num number constraint  
pay_order_num_fk references orders(order_num));
```

```
alter table pay  
add constraint pay_pay_method_num_fk foreign key(pay_method_num) references  
pay_method(pay_method_num);
```

결제방법

```
create table pay_method(  
pay_method_num number primary key,  
without_bankbook varchar2(30) ,  
account_transfer number ,  
credit_card number ,  
pay_num number constraint pay_method_pay_num_fk references pay(pay_num));
```

인서트

고객테이블에 인서트

insert into

```
customer(customer_id,customer_pw,customer_name,customer_address,customer_number,customer_email)
values ('c1',1111,'강준기','서울시 강동구',01011112222,'www.c1@naver.com');
```

insert into

```
customer(customer_id,customer_pw,customer_name,customer_address,customer_number,customer_email)
values ('c2',2222,'강지영','서울시 서초구',01012342222,'www.c2@naver.com');
```

insert into

```
customer(customer_id,customer_pw,customer_name,customer_address,customer_number,customer_email)
values ('c3',3333,'나인권','서울시 강남구',01023452312,'www.c3@naver.com');
```

insert into customer

```
values ('c4',4444,'정다솜','서울시 노원구',01012523312,'www.c4@naver.com','19/01/23');
```

insert into customer

```
values ('c5',5555,'신재득','서울시 관악구',01098763312,'www.c5@naver.com','19/02/23');
```

commit;

문의사항 게시판에 인서트

```
insert into question_border(question_num,customer_name,customer_email,customer_message,customer_id)
values (question_num_seq.nextval,'강준기','www.c1@naver.com','재고없나여?','c1');
```

```
insert into question_border(question_num,customer_name,customer_email,customer_message,customer_id)
values (question_num_seq.nextval,'강지영','www.c2@naver.com','결제가됐나여?','c2');
```

```
insert into question_border(question_num,customer_name,customer_email,customer_message,customer_id)
values (question_num_seq.nextval,'나인권','www.c3@naver.com','배송이진행중인가요?','c3');
```

```
insert into question_border(question_num,customer_name,customer_email,customer_message,customer_id)
values (question_num_seq.nextval,'정다솜','www.c4@naver.com','상품바꿀수있나여?','c4');
```

```
insert into question_border(question_num,customer_name,customer_email,customer_message,customer_id)
values (question_num_seq.nextval,'신재득','www.c5@naver.com','환불어떻게하져?','c5');
```

```
commit;
```

공지게시판에 인서트

```
insert into notice_border values (notice_seq.nextval,'공지','신상품 목록 입니다',sysdate,10);
insert into notice_border values (notice_seq.nextval,'공지','세일기간 입니다',sysdate,20);
insert into notice_border values (notice_seq.nextval,'공지','할인품목 입니다',sysdate,3);
insert into notice_border values (notice_seq.nextval,'공지','1+1 행사입니다.',sysdate,100);
insert into notice_border values (notice_seq.nextval,'공지','이벤트목록입니다.',sysdate,40);
```

구매자에 인서트

```
insert into buyer values('bc1','c1');
insert into buyer values('bc2','c2');
insert into buyer values('bc3','c3');
insert into buyer values('bc4','c4');
insert into buyer values('bc5','c5');
```


카테고리에 insert

```
insert into category
```

```
values('cloth','의류','p01');
```

```
insert into category
```

```
values('cloth1','의류','p02');
```

```
insert into category
```

```
values('book','책','p03');
```

```
insert into category
```

```
values('book1','책','p04');
```

```
insert into category
```

```
values('shoes','신발','p05');
```

```
insert into category
```

```
values('shoes1','신발','p06');
```

```
insert into category
```

```
values('health','건강','p07');
```

```
insert into category
```

```
values('health1','건강','p08');
```

```
insert into category
```

```
values('game','게임','p09');
```

```
insert into category
```

```
values('game1','게임','p10');
```

색깔(옵션)

```
insert into color values('color1','빨간색,검정색','p01');
insert into color values('color2','검정색,파란색','p01');
insert into color values('color3','파란색,흰색','p01');
insert into color values('color4','흰색,보라색','p01');
insert into color values('color5','보라색,검정색','p01');
insert into color values('color6','빨간색,검정색','p02');
insert into color values('color7','검정색,파란색','p02');
insert into color values('color8','파란색,흰색','p02');
insert into color values('color9','흰색,보라색','p02');
insert into color values('color10','보라색,검정색','p02');
```

상품

```
insert into product values('p01','cloth.jpg','티셔츠',10000,100,10000,5000,'19/05','color1','cloth');
```

```
insert into product values('p02','cloth2.jpg','바지',20000,10,20000,10000,'19/05','color5','cloth');
```

```
insert into
```

```
product(product_code,product_image,product_name,product_price,product_num,normal_price,sale_price,sale_date,category_code)
```

```
values('p03','book.jpg','연금술사',7000,5,7000,5000,'19/06','book');
```

```
insert into
```

```
product(product_code,product_image,product_name,product_price,product_num,normal_price,sale_price,sale_date,category_code)
```

```
values('p04','book1.jpg','파리5구역',8000,3,8000,6000,'19/06','book');
```

```
insert into
```

```
product(product_code,product_image,product_name,product_price,product_num,normal_price,sale_price,sale_date,category_code)
```

```
values('p05','shoes.jpg','스니커즈',30000,10,30000,15000,'19/07','shoes');
```

```
insert into
```

```
product(product_code,product_image,product_name,product_price,product_num,normal_price,sale_price,sale_date,category_code)
```

```
values('p06','shoes1.jpg','등산화',15000,20,15000,10000,'19/07','shoes');
```

insert into

```
product(product_code,product_image,product_name,product_price,product_num,normal_price,sale_price,sale_date,category_code)
```

```
values('p07','health.jpg','녹용',100000,5,100000,50000,'19/08','health');
```

insert into

```
product(product_code,product_image,product_name,product_price,product_num,normal_price,sale_price,sale_date,category_code)
```

```
values('p08','health1.jpg','비타민',20000,10,20000,150000,'19/08','health');
```

insert into

```
product(product_code,product_image,product_name,product_price,product_num,normal_price,sale_price,sale_date,category_code)
```

```
values('p09','game.jpg','디아블로4',70000,30,70000,50000,'19/09','game');
```

insert into

```
product(product_code,product_image,product_name,product_price,product_num,normal_price,sale_price,sale_date,category_code)
```

```
values('p10','game1.jpg','오버워치2',80000,30,80000,60000,'19/09','game');
```

재고상태

```
insert into stock_status values(stock_num_seq.nextval,'티셔츠',100,'유','p01');
insert into stock_status values(stock_num_seq.nextval,'바지',10,'유','p02');
insert into stock_status values(stock_num_seq.nextval,'연금술사',5,'유','p03');
insert into stock_status values(stock_num_seq.nextval,'파리5구역',3,'유','p04');
insert into stock_status values(stock_num_seq.nextval,'스니커즈',10,'유','p05');
insert into stock_status values(stock_num_seq.nextval,'등산화',20,'유','p06');
insert into stock_status values(stock_num_seq.nextval,'녹용',5,'유','p07');
insert into stock_status values(stock_num_seq.nextval,'비타민',10,'유','p08');
insert into stock_status values(stock_num_seq.nextval,'디아블로4',30,'유','p09');
insert into stock_status values(stock_num_seq.nextval,'오버워치2',30,'유','p10');
```

주문

```
insert into orders values(order_num_seq.nextval,sysdate,2,'p01','bc1');
insert into orders values(order_num_seq.currval,sysdate,1,'p02','bc1');
insert into orders values(order_num_seq.currval,'19/06/02',3,'p03','bc2');
insert into orders values(order_num_seq.currval,'19/06/02',1,'p04','bc2');
insert into orders values(order_num_seq.currval,sysdate,2,'p05','bc3');
insert into orders values(order_num_seq.currval,'19/07/05',5,'p06','bc3');
insert into orders values(order_num_seq.currval,sysdate,2,'p07','bc4');
insert into orders values(order_num_seq.currval,sysdate,1,'p08','bc4');
insert into orders values(order_num_seq.currval,'19/09/23',1,'p09','bc5');
insert into orders values(order_num_seq.currval,'19/09/24',1,'p10','bc5');
```

배송

```
insert into delivery values(delivery_num_seq.nextval,sysdate,'한진','서울',2,'d01');
insert into delivery values(delivery_num_seq.nextval,sysdate,'한진','서울',3,'d01');
insert into delivery values(delivery_num_seq.nextval,'19/06/02','한진','서울',4,'d01');
insert into delivery values(delivery_num_seq.nextval,'19/06/02','한진','서울',5,'d02');
insert into delivery values(delivery_num_seq.nextval,sysdate,'한진','서울',6,'d02');
insert into delivery values(delivery_num_seq.nextval,'19/07/05','한진','서울',7,'d02');
insert into delivery values(delivery_num_seq.nextval,sysdate,'한진','서울',8,'d02');
insert into delivery values(delivery_num_seq.nextval,sysdate,'한진','서울',9,'d03');
insert into delivery values(delivery_num_seq.nextval,'19/09/23','한진','서울',10,'d03');
insert into delivery values(delivery_num_seq.nextval,'19/09/24','한진','서울',11,'d03');
```

배송가격

```
insert into delivery_price values('d01',10,3000,1);
insert into delivery_price values('d02',20,4000,2);
insert into delivery_price values('d03',30,5000,3);
```

결제방법

```
insert into pay_method(pay_method_num,account_transfer,pay_num)
values(pay_num_method_seq.nextval,1102413334,1);
insert into pay_method(pay_method_num,account_transfer,pay_num)
values(pay_num_method_seq.nextval,1102413334,2);
insert into pay_method(pay_method_num,without_bankbook,pay_num) values(pay_num_method_seq.nextval,'
무통장입금',3);
insert into pay_method(pay_method_num,without_bankbook,pay_num) values(pay_num_method_seq.nextval,'
무통장입금',4);
insert into pay_method(pay_method_num,credit_card,pay_num)
values(pay_num_method_seq.nextval,102411234,5);
insert into pay_method(pay_method_num,credit_card,pay_num)
values(pay_num_method_seq.nextval,102312334,6);
insert into pay_method(pay_method_num,account_transfer,pay_num)
values(pay_num_method_seq.nextval,1102413334,7);
insert into pay_method(pay_method_num,without_bankbook,pay_num) values(pay_num_method_seq.nextval,'
무통장입금',8);
insert into pay_method(pay_method_num,credit_card,pay_num)
values(pay_num_method_seq.nextval,232122345,9);
insert into pay_method(pay_method_num,credit_card,pay_num)
values(pay_num_method_seq.nextval,432312345,10);
```

결제

```
insert into pay(pay_num,pay_date,pay_status,total_pay,pay_method_num,buyer_id,order_num)
values(pay_num_seq.nextval,sysdate,'유',23000,1,'bc1',2);
insert into pay(pay_num,pay_date,pay_status,total_pay,pay_method_num,buyer_id,order_num)
values(pay_num_seq.nextval,sysdate,'유',23000,2,'bc1',3);
insert into pay(pay_num,pay_date,pay_status,total_pay,pay_method_num,buyer_id,order_num)
values(pay_num_seq.nextval,'19/06/02','유',18000,3,'bc2',4);
insert into pay(pay_num,pay_date,pay_status,total_pay,pay_method_num,buyer_id,order_num)
values(pay_num_seq.nextval,'19/06/02','유',10000,4,'bc2',5);
insert into
pay(pay_num,pay_date,pay_status,credit_card_num,total_pay,pay_method_num,buyer_id,order_num)
values(pay_num_seq.nextval,sysdate,'유',102411234,64000,5,'bc3',6);
insert into
pay(pay_num,pay_date,pay_status,credit_card_num,total_pay,pay_method_num,buyer_id,order_num)
values(pay_num_seq.nextval,'19/07/05','유',102312334,54000,6,'bc3',7);
insert into pay(pay_num,pay_date,pay_status,total_pay,pay_method_num,buyer_id,order_num)
values(pay_num_seq.nextval,sysdate,'유',34000,7,'bc4',8);
insert into pay(pay_num,pay_date,pay_status,total_pay,pay_method_num,buyer_id,order_num)
values(pay_num_seq.nextval,sysdate,'유',25000,8,'bc4',9);
insert into
pay(pay_num,pay_date,pay_status,credit_card_num,total_pay,pay_method_num,buyer_id,order_num)
values(pay_num_seq.nextval,'19/09/23','유',232122345,55000,9,'bc5',10);
```



```
insert into pay(pay_num,pay_date,pay_status,credit_card_num,total_pay,pay_method_num,buyer_id,order_num)
values(pay_num_seq.nextval,'19/09/24','유',432312345,65000,10,'bc5',11);
```

장바구니

```
insert into cart values('bc1','p01');
insert into cart values('bc1','p02');
insert into cart values('bc2','p03');
insert into cart values('bc2','p04');
insert into cart values('bc3','p05');
insert into cart values('bc3','p06');
insert into cart values('bc4','p07');
insert into cart values('bc4','p08');
insert into cart values('bc5','p09');
insert into cart values('bc5','p10');
```

시퀀스 생성

1. 문의사항 번호 시퀀스 생성

```
create sequence question_num_seq  
start with 1  
increment by 1  
maxvalue 300;
```

2. 공지번호 시퀀스 생성

```
create sequence notice_seq  
start with 1  
increment by 1  
maxvalue 200;
```

3. 재고번호 시퀀스 생성

```
create sequence stock_num_seq  
start with 1  
increment by 1  
maxvalue 1000;
```

4. 주문번호 시퀀스 생성

```
create sequence order_num_seq  
start with 1  
increment by 1  
maxvalue 10000;
```

5.배송번호 시퀀스 생성

```
create sequence delivery_num_seq  
start with 1  
increment by 1  
maxvalue 10000;
```

6.결제번호 시퀀스 생성

```
create sequence pay_num_seq  
start with 1  
increment by 1  
maxvalue 10000;
```

7.결제방법번호 시퀀스 생성

```
create sequence pay_num_method_seq  
start with 1  
increment by 1  
maxvalue 10000;
```

트리거 생성

주문 테이블에 주문이 들어오면 주문번호 생성

```
create or replace trigger orders_trigger
before insert
on orders
for each row
begin
select order_num_seq.nextval into :new.order_num from dual;
end;
/
```

주문이 취소되면서 결제 자동취소

```
create or replace trigger cancel_trigger
after delete on orders
for each row
begin
delete from pay
where order_num = :old.order_num;
end;
/
```

조인

1. 구매자의 buyer id 와 상품명과 주문갯수 알아보기

```
select c.customer_id,b.buyer_id,p.product_name,p.product_price,o.order_count from customer c, buyer b ,product p ,orders o
where c.customer_id = b.customer_id and p.product_code = o.product_code
and b.buyer_id =o.buyer_id;
```

2,배송업체의 배송가격 알아보기

```
select * from delivery d left outer join delivery_price dp
on d.delivery_price_code = dp.delivery_price_code;
```

3. 총결제해야하는 금액에 대한 정보 알아보기

```
select product_name,product_price,sale_price,sale_date,order_count,delivery_company,price_tag
from product p,orders o,delivery d,delivery_price dp
where p.product_code = o.product_code and o.order_num = d.order_num and dp.delivery_price_code = d.delivery_price_code ;
```

인덱스 생성

```
create unique index idx_cusomer_customer_id  
on customer(customer_id);
```

```
create unique index idx_orders_order_num  
on orders(order_num);
```

```
create unique index idx_buyer_buyer_id  
on buyer(buyer_id);
```

```
create index idx_product_com  
on product(product_name,product_price,normal_price,sale_price,sale_date);
```

```
set timing on;
```

```
select product_name,product_price,normal_price,sale_price from product;
```

뷰 생성

회원 비밀번호 검색

```
create or replace view pass_view_chk  
as  
select customer_pw from customer  
where customer_id ='c1' with check option;  
  
select * from pass_view_chk;
```

회원 아이디 검색

```
create or replace view id_view_chk  
as  
select customer_id from customer  
where customer_number = 01011112222 with check option;  
  
select * from id_view_chk;
```

상품의 가격이 큰순서대로 5개만 보여주기

```
create or replace view product_view  
as  
select * from (select rownum list,product_name,normal_price,sale_price,sale_date from product order by normal_price desc)  
where list<=5;
```

PL/SQL

1. 상품명 검색해서 할인월에 들어가는지 확인하고 할인된 가격알아보기

```
create or replace procedure product_sale
(vproduct_name in product.product_name%type)
is
vnormal_price product.normal_price%type;
vsale_price product.sale_price%type;
vsale_date product.sale_date%type;
begin
select normal_price,sale_price,sale_date into vnormal_price,vsale_price,vsale_date from product
where product_name =vproduct_name;

if substr(vsale_date,4,2) in ('06','07','08','09') then
dbms_output.put_line('지금 할인기간이어서'||vsale_price || '입니다. ');
else dbms_output.put_line('할인기간이 아니어서 정사가'||vnormal_price||'입니다' );
end if;
end;
/

set serveroutput on;
execute product_sale('디아블로4');
execute product_sale('바지');
```


2. 구매자 id와 주문번호를 이용해서 주문가격 구하기

create or replace procedure search_buyer

(vbuyer_id in buyer.buyer_id%type,

vorder_num in orders.order_num%type)

is

vcustomer_id customer.customer_id%type;

vproduct_name product.product_name%type;

vproduct_price product.product_price%type;

vorder_count orders.order_count%type;

p_calculator number;

begin

select product_name,product_price,order_count into vproduct_name,vproduct_price,vorder_count from product p,orders o,buyer b

where p.product_code=o.product_code and b.buyer_id=vbuyer_id and b.buyer_id=o.buyer_id and o.order_num=vorder_num;

p_calculator := vproduct_price * vorder_count;

dbms_output.put_line(vbuyer_id||'의 배송비를제외한 구매상품의 가격은'||p_calculator||'입니다');

end;

/

set serveroutput on

execute search_buyer('bc2',5);

execute search_buyer('bc1',3);

3. 구매자가 담은 CART 상품 보는 프로시저

create or replace procedure cart_wish

is

cursor c1 is

select buyer_id,product_code from cart;

begin

for v_cart in c1 loop

dbms_output.put_line('구매자아이디:||v_cart.buyer_id);

dbms_output.put_line('상품코드:||v_cart.product_code);

end loop;

end;

/

set serveroutput on

execute cart_wish;

4. 고객회원정보 update

```
create or replace procedure update_customer_info(  
vcustomer_id in customer.customer_id%type,  
vcustomer_pw in customer.customer_pw%type,  
vcustomer_name in customer.customer_name%type,  
vcustomer_address in customer.customer_address%type,  
vcustomer_email in customer.customer_email%type,  
vcustomer_num in customer.customer_number%type)  
is  
begin  
update customer  
set customer_pw=vcustomer_pw,customer_name=vcustomer_name,customer_address=vcustomer_address,  
customer_email=vcustomer_email,customer_number=vcustomer_num  
where customer_id=vcustomer_id;  
end;  
  
/  
  
set serveroutput on  
execute update_customer_info('c1',1111,'강준기','서울시 강동구','www.c1@naver.com',01011112222);
```

5. 상품의 이름을 검색해서 카테고리 and 재고상태 알아보기

```
create or replace procedure cart_pro_status(  
vproduct_name in product.product_name%type,  
vcategory_type out category.category_type%type,  
vstock_quantity out stock_status.stock_quantity%type)  
is  
begin  
select category_type,stock_quantity into vcategory_type,vstock_quantity  
from product p,category c,stock_status s  
where p.product_code = c.product_code and p.product_code = s.product_code  
and p.product_name=vproduct_name;  
end;  
/
```

```
variable vcategory_type varchar2(30);  
variable vstock_quantity number;  
execute cart_pro_status('디아블로4',:vcategory_type,:vstock_quantity);  
print vcategory_type;  
print vstock_quantity;
```

6. 상품 가격별 순위와 할인상품 가격별 순위를 보여줌

```
create or replace procedure rank_procedure
```

```
is
```

```
cursor c1 is
```

```
select product_name,product_price,sale_price,
```

```
rank()over(order by product_price desc) product_rank,
```

```
rank()over(order by sale_price desc) sale_product_rank from product;
```

```
begin
```

```
for rank_product in c1 loop
```

```
exit when c1 %notfound;
```

```
dbms_output.put_line(rank_product.product_name||'      '||rank_product.product_price||' ' ||rank_product.sale_price
```

```
||'      '||rank_product.product_rank||'      '||rank_product.sale_product_rank);
```

```
end loop;
```

```
end;
```

```
/
```

```
execute rank_procedure;
```

7. 상품별 최대,최소 가격 과 카테고리별 최대,최소 가격보여줌

```
create or replace procedure mmc_product
```

```
is
```

```
cursor c1 is
```

```
select product_name,category_type,max(product_price) maxp,min(product_price) minp
```

```
from product p,category c
```

```
where p.product_code=c.product_code
```

```
group by grouping sets(product_name,category_type);
```

```
begin
```

```
for search_price in c1 loop
```

```
exit when c1 %notfound;
```

```
dbms_output.put_line(search_price.product_name||'   '||search_price.category_type||'   '||
```

```
search_price.maxp||'   '||search_price.minp);
```

```
end loop;
```

```
end;
```

```
/
```

```
execute mmc_product;
```

8. 카드번호를 넣으면 결제금액 출력

```
create or replace function fn_pay(  
vcredit_card_num in pay.credit_card_num%type)  
return number  
is  
vtotal_pay pay.total_pay%type;  
begin  
select total_pay into vtotal_pay from pay  
where credit_card_num = vcredit_card_num;  
return vtotal_pay;  
end;  
/
```

```
variable vtotal_pay number;  
execute :vtotal_pay := fn_pay(102411234);  
print vtotal_pay;
```

9.고객이름과 고객주소를 각각의 테이블에 배열로 저장

create or replace procedure save_table

is

type customer_name_table is table of customer.customer_name%type

index by binary_integer;

type customer_address_table is table of customer.customer_address%type

index by binary_integer;

customer_name_tab customer_name_table;

customer_address_tab customer_address_table;

i binary_integer := 0;

begin

for customer_list in (select customer_name,customer_address from customer) loop

i := i+1;

customer_name_tab(i) := customer_list.customer_name;

customer_address_tab(i) := customer_list.customer_address;

end loop;

for cnt in 1..i loop

dbms_output.put_line('고객이름:||customer_name_tab(cnt));

dbms_output.put_line('고객주소:||customer_address_tab(cnt));

end loop;

end;

/

set serveroutput on;

execute save_table;

패키지(모든상품정보,모든주문정보,주문번호를넣었을때 결제정보,주문번호를넣었을때 배송정보)

```
create or replace package all_info as
procedure all_product_info;
procedure all_orders_info;
procedure order_pay_info(vorder_num in number);
procedure order_delivery_info(vorder_num in number);
end all_info;
/

create or replace package body all_info as
procedure all_product_info
is
cursor product_cursor is
select product_code,product_name,product_price,sale_price,sale_date from product;

begin
for aa in product_cursor loop
dbms_output.put_line('상품코드:|| aa.product_code);
dbms_output.put_line('상품명:|| aa.product_name);
dbms_output.put_line('상품가격:|| aa.product_price);
dbms_output.put_line('세일가격:|| aa.sale_price);
dbms_output.put_line('세일기간:|| aa.sale_date);
end loop;

exception
when others then
dbms_output.put_line(sqlerrm||'에러발생');

end all_product_info;
```

```
procedure all_orders_info
is
cursor orders_cursor is
select order_date,order_count,order_num from orders;

begin
for aa in orders_cursor loop
dbms_output.put_line('주문날짜:|| aa.order_date);
dbms_output.put_line('주문갯수:|| aa.order_count);
dbms_output.put_line('주문번호:|| aa.order_num);
end loop;

exception
when others then
dbms_output.put_line(sqlerrm||'에러발생');

end all_orders_info;
```

```
procedure order_pay_info(vorder_num in number)
is
cursor pay_cursor is
select buyer_id,pay_date,total_pay from pay
where order_num = vorder_num;

begin

for aa in pay_cursor loop
dbms_output.put_line('구매자아이디:|| aa.buyer_id);
dbms_output.put_line('결제일:|| aa.pay_date);
dbms_output.put_line('총결제금액:|| aa.total_pay);
end loop;

exception
when others then
dbms_output.put_line(sqlerrm ||'에러발생');

end order_pay_info;
```

```
procedure order_delivery_info(vorder_num in number)
is
cursor delivery_cursor is
select delivery_company,price_tag from delivery d,delivery_price dp
where order_num=vorder_num and d.delivery_price_code = dp.delivery_price_code;

begin
for aa in delivery_cursor loop
dbms_output.put_line('배송회사:|| aa.delivery_company);
dbms_output.put_line('배송가격:|| aa.price_tag);
end loop;

exception
when others then
dbms_output.put_line(sqlerrm||'에러발생');

end order_delivery_info;

end all_info;
/

set serveroutput on
exec all_info.all_product_info;
exec all_info.all_orders_info;
exec all_info.order_pay_info(7);
exec all_info.order_delivery_info(8);
```

감사합니다

