[Deal Plus] DB설계 프로젝트

3조 정수현, 이기순, 최준현

1 주제

공동구매 기반 오픈마켓

2 개발 배경

- 1. SNS, 홈쇼핑, 라이브 커머스 등 일부 채널에서만 이루어지는 대량판매 할인 서비스는 광고 비용부터 입점 비용, 쇼 호스트 섭외까지 상품 판매를 하기 위해 부담해야 하는 비용이 상당하여 일반 업체 또는 자영업자들이 이용하기에는 어려움이 많음
- 2. 일반 판매자들도 폭 넓게 이용할 수 있을 만큼 대량판매 시스템이 잘 갖추어진 매체가 현저하게 없음.

3 설계 목적

- 1. 영세한 자영업자를 비롯하여 전자상거래를 하는 업체들에게 큰 비용 부담 없이 누구나 대량판매를 기반으로 규모의 경제를 실현시킬 수 있는 편리한 판매 시스템을 만들고자 함
- 2. 인원 별, 할인율, 마감일 기준에 따라 상품 입고 컨디션과 판매전략을 달리하여 판매 리스크를 줄이고자 함
- 3. 판매되었던 데이터들을 베이스로 판매자에게 꼭 필요한 매출 관련 리포트를 제공하여 양질의 효과를 보고자 함

4 기대효과

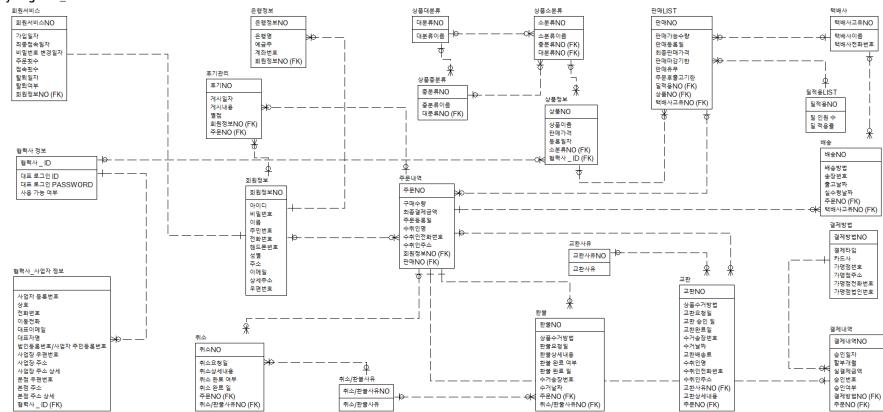
- 1. 인풋 비용을 제거하여 좀 더 쉽고 편리하게 많은 판매자들의 입점이 가능
- 2. 대량 판매를 통해 총 판매이윤이 올라가면 양질의 다양한 상품들이 거래가 될 것으로 예상
- 3. 판매자는 재고 리스크 예측이 가능하며 판매 기간에 따라 입고 및 판매를 조절하여 손실을 최소화할 수 있음
- 4. 소비자는 시장 거래 금액 보다 더욱 합리적인 가격으로 다양한 상품을 구매할 수 있어 수요 상승
- 5. 개인이 아닌 불특정 다수의 구매자들이 모여 거래가 이루어짐으로써 소비 부담을 줄여 거래량 증가

5 주요기능

- 1. 구매인원 대비 할인율 자동 변동
 - 판매기간 동안 모인 판매 수량에 따라 판매자가 설정해 놓은 할인율이 마감일에 맞추어 상이하게 적용 됨
- 2. 선 결제 후 자동 취소 및 재 결제 시스템
 - 구매 일 기준 금액으로 결제가 진행 되지만 마감일 기준 할인율이 다시 적용되어 결제 취소 후 최종 할인율 적용 받아 재 결제
 - 소비자의 불편함과 불이익(시점에 따른 금액 차이 피해) 을 없앨 수 있음
- 3. 업체들에게 꼭 필요한 매출 관련 리포트를 제공
 - 분기별, 카테고리별, 상품별, 고객별, 결제수단별 등등 매출 자료 및 예상 자료 분석 기능

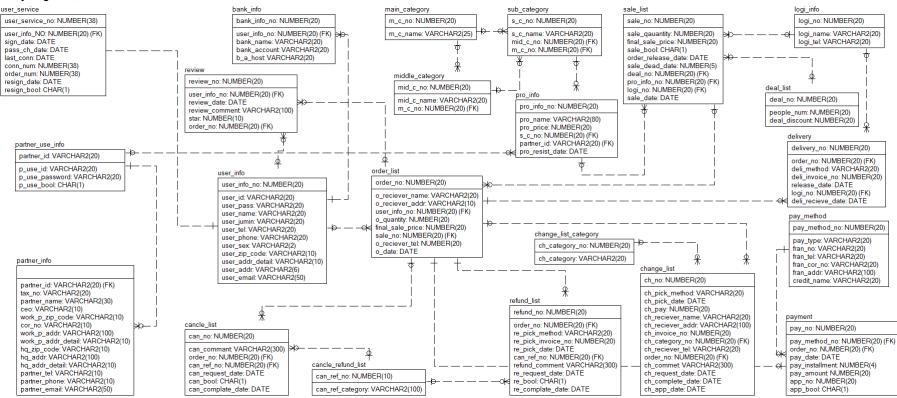
■ DATA_TABLE_LOGICAL

Anything Deal_data table



■ DATA TABLE PHYSICAL

Anything Deal_data table



■ 마케팅

	리포트 항목	SQL 쿼리	
1	요일 별 상품 구매 현황	select to_char(o_date, 'day') "요일", count(*) "판매갯수"from order_list group by to_char(o_date, 'day'), to_char(o_date - 1, 'D') order by to_char(o_date - 1, 'D'); 요일	
2	기간 별 객 단가 변동 현황	select trunc((sum(final_sale_price)/count(user_info_no)),-2) "인당평균구매금액" from order_list;	
3	특정 업체의 특정 기간 동안 판매된 상품들의 남녀 비율 현황	select p.partner_id "업체번호", u.user_sex "성별", count(u.user_sex) "인원수" from order_list o, sale_list s , user_info u, pro_info p where o.sale_no=s.sale_no and s.pro_info_no=p.pro_info_no and o.user_info_no = u.user_info_no and o.o_date between '2021-01-01' and '2021-01-31' and p.partner_id = 19 group by p.partner_id, u.user_sex;	

	리포트 항목	SQL 쿼리
		select p.partner_id "업체번호", nvl((f_age(user_jumin)),0) "연령대", count(*) "판매갯수"
		from order_list o, sale_list s , user_info u, pro_info p
		where o.sale_no=s.sale_no and s.pro_info_no=p.pro_info_no and o.user_info_no = u.user_info_no and
		o.o_date between '2021-01-01' and '2021-01-31' and p.partner_id = 19
1	특정 업체의 특정 기간 동안 판매된 상품들의 연령대 비율 현	group by p.partner_id, nvl((f_age(user_jumin)),0)
4	에진 성품들의 전당대 미필 연 황	having nvl((f_age(user_jumin)),0) >= 20 and nvl((f_age(user_jumin)),0) <= 70
	ਰ	order by 2;
		업체번호 연형대 판매갯수
		19 20 5 19 30 5 19 40 4 19 60 3
		select f_age(user_jumin) "연령대",
		sum(decode(p.s_c_no, 1,1)) 소분류_1,
		sum(decode(p.s_c_no, 2,1)) 소분류_2,
		sum(decode(p.s_c_no, 3,1)) 소분류_3
		from user_info u, order_list o, sale_list s, pro_info p
		where u.user_info_no = o.user_info_no and o.sale_no = s.sale_no and s.pro_info_no = p.pro_info_no group
		by f_age(user_jumin) order by 1 asc;
5	연령별, 카테고리별 주문 현황	연령대 소분류_1 소분류_3
		0 229 187 172 10 237 192 168
		0 229 187 172 10 237 192 168 20 87 75 69 30 68 44 29 40 60 46 27 50 47 44 47 60 56 47 37 70 52 46 45 80 56 36 38 90 59 39 48 100 68 34 48
		40 60 46 27 50 47 44 47 60 56 47 37
		70 52 46 45 80 56 36 38
		100 68 34 48
		연령대 소분류_1 소분류_3
		110 62 58 47 120 19 12 5

	리포트 항목	SQL 쿼리
	지역별, 연령별 회원 분포 현황	select f_age(user_jumin) "연령대", sum(decode(user_addr,'서울',1)) 서울, sum(decode(user_addr,'인천',1)) 인천, sum(decode(user_addr,'광주',1)) 광주, sum(decode(user_addr,'세종',1)) 세종, sum(decode(user_addr,'대구',1)) 대구, sum(decode(user_addr,'부산',1)) 부산, sum(decode(user_addr,'제주',1)) 제주 from user_info group by f_age(user_jumin) order by 1;
6		변형대 서울 인천 광주 세종 대구 부산 제주 0 381 370 374 352 395 408 391 10 386 384 365 399 412 432 404 20 148 127 146 155 122 140 139 30 72 87 93 84 82 93 88 40 80 79 80 82 95 86 94 50 100 96 80 71 73 79 81 60 71 76 80 90 116 97 94 70 88 81 96 87 80 82 75 80 74 102 78 90 96 79 81 90 91 91 90 89 103 72 88 100 78 85 87 78 103 89 87 연형대 서울 인천 광주 세종 대구 부산 제주 110 82 90 89 78 95 93 88 120 20 17 19 18 25 19 18
7	주문 목록 중 카드사별 이용 현황	select pm.pay_method_no "결제방법번호", pm.credit_name "카드사", count(pm.pay_method_no) "이용횟수" from pay_method pm, payment pa, order_list o, sale_list s where pm.pay_method_no = pa.pay_method_no and pa.order_no = o.order_no and o.sale_no = s.sale_no having pm.pay_method_no <= 8 group by pm.pay_method_no, pm.credit_name order by 1; ②제방법번호 카드사 이용횟수 1 BC라드 1003 2 KB국민카드 982 3 삼성카드 983 4 신한카드 981 5 우리카드 984 6 하나카드 984 6 하나카드 985 7 루데카드 985 8 현대카드 985

	리포트 항목	SQL 쿼리
8	특정 일 하루 동안 발생한 주문 목록 중 상품별 결제타입	select o.order_no "주문번호", sum(decode(pm.pay_type,'카드',1)) 카드, sum(decode(pm.pay_type,'계좌이체',1)) 계좌이체 from pay_method pm, payment pa, order_list o where pm.pay_method_no = pa.pay_method_no and pa.order_no = o.order_no and o.o_date = '2021-01- 01' group by o.order_no; 18988
9	연령대 별 결제 수단 현황	select f_age(user_jumin) "연령대", sum(decode(pm.pay_type,'카드',1)) 카드, sum(decode(pm.pay_type,'계좌이체',1)) 계좌이체 from pay_method pm, payment pa, order_list o, user_info u where pm.pay_method_no = pa.pay_method_no and pa.order_no = o.order_no and o.user_info_no = u.user_info_no group by f_age(user_jumin) having f_age(user_jumin) >= 20 and f_age(user_jumin) <= 70 order by 1 asc; - 연령대 - 카드 - 계좌이체 - 기도 - 계좌이체 - 기도 - 기조

	리포트 항목	SQL 쿼리
10	디포드 영국 연령대 별 선호하는 구매 가격대	SQL 거디 select f_age(user_jumin) "연령대", count(case when final_sale_price < 10000 then 1 end) "1만원미만", count(case when final_sale_price >=10000 or final_sale_price < 30000 then 1 end) "3만원미만", count(case when final_sale_price >=30000 or final_sale_price < 50000 then 1 end) "5만원미만", count(case when final_sale_price >=50000 or final_sale_price < 100000 then 1 end) "10만원미만", count(case when final_sale_price >=100000 then 1 end) "10만원이상" from user_info u, order_list o where u.user_info_no = o.user_info_no group by f_age(user_jumin) having f_age(user_jumin) >= 20 and f_age(user_jumin) <= 70 order by 1 asc; 20 122 1362 1362 1362 246 30 70 847 847 847 143 40 83 825 825 825 153 50 68 813 813 813 151 60 90 861 861 861 861 150 77 818 818 818 128

	리포트 항목	SQL 쿼리
11	카테고리 별 선호하는 구매 가격대	select s_c_no "소분류", count(case when o.final_sale_price < 10000 then 1 end) "1만원미만", count(case when o.final_sale_price >=10000 or o.final_sale_price < 30000 then 1 end) "3만원미만", count(case when o.final_sale_price >=30000 or o.final_sale_price < 50000 then 1 end) "5만원미만", count(case when o.final_sale_price >=50000 or o.final_sale_price < 100000 then 1 end) "10만원미만", count(case when o.final_sale_price >=100000 then 1 end) "10만원이상" from order_list o, sale_list s, pro_info p where o.sale_no = s.sale_no and s.pro_info_no = p.pro_info_no group by s_c_no order by 1 asc; 32 25 450 450 450 450 0

■ 회원

	리포트 항목	SQL 쿼리
1	기간 별 신규 및 재구매 현황	select user_info_no 고객번호,s.pro_info_no 상품코드,(select pro_name from pro_info p where p.pro_info_no=s.pro_info_no) 상품명,count(O_QUANTITY) 수량 from order_list o, sale_list s where o.sale_no=s.sale_no and o.o_date between '2021-01-01' and '2021-03-31' group by o.user_info_no,s.pro_info_no having count(O_QUANTITY) >=2; 고객번호 상품코드 상품명 수량 수량 8099 384 [미쟝센]미쟝센 스타일케어 화이트스프레이 300ML x6개 2 3187 552 (포) 비트엠 Newsync B2482H ZERO HDR 무결점
2	연령대 별 구매목록의 만족도 현황	select f_age(u.user_jumin) 연령대,lpad('*',avg(c.star),'*') avg_star 별점 from order_list o, user_info u, review c where o.user_info_no=u.user_info_no and o.user_info_no=c.user_info_no and o.order_no=c.order_no and o.o_date between '2020-01-01' and '2021-08-10' group by u.user_jumin order by avg_star;
3	최우수 고객 5명 (최대 구매 고객)	select user_info_no "최대구매고객" from user_info where user_info_no in (select user_info_no from (select *from order_list order by final_sale_price desc) where rownum < 6); 최대구매고객 11053 2083 3585 1916
4	시간대 별 구매자 접속 현황	select to_char(o_date,'HH') "구매시간" ,sum(final_sale_price) "시간대별 구매 총액" from order_list where o_date between '2021-01-01' and '2021-01-15' group by to_char(o_date,'HH');

■ 매출

	리포트 항목	SQL 쿼리
	특정기간, 업체별 매출 현황	select s.pro_info_no 상품코드,(select p.pro_name from pro_info p where p.pro_info_no=s.pro_info_no)상품 명, sum(o.o_quantity) 총수량,sum(o.final_sale_price) 총매출 from order_list o, sale_list s where o.sale_no = s.sale_no and o.o_date between '2021-01-01' and '2021-01-16' group by s.pro_info_no order by
		s.pro_info_no;
1		585 [웨스턴디지털]WD BLUE (WD10EZEX) 3.5 SATA HDD (1TB) 5 360000 594 [로마네]브런치브라더 11 13 15 17 파우치 탭 패드 노트북 8D 5 125000 597 [갤럭시]jj 삼성전자 갤럭시워치 액티브2 알루미늄 40mm 10 2668100 598 [다이아큐브](3매)4H T-Hybrid 고경도 유리하드코팅 액정보호필름 5 90000 600 [사성전자]1+1 삼성전용 C타일 마이크로5핀C toC고속충전 케이블 5 44500 602 [디제이아이]DJI MINI 2 플라이 모어 콤보 초경량 미니드론 (4K동영상 오큐 5 3820450 603 소니 ZV-1 디지털카메라 정품 새상품 5 4164700 612 [인스탁스]미니11/몰라로이드/즉석카메라 핑크/필름+가방외 5 637500 614 캐논 정품 EF 50mm F1.8 STM 당 일 발 송 아워스 5 1250000 620 [엑스박스]XBOX SERIES X 엑스박스 무선 컨트롤러 4세대 화이트 5 309000 624 [닌텐도]닌텐도 스위치 AC 어댑터 5 160000
		select m.partner_id 협력사_ID, s.pro_info_no 상품코드,(select p.pro_name from pro_info p where
		p.pro_info_no=s.pro_info_no)상품명, sum(o.final_sale_price)총매출 from order_list o, sale_list s, partner_info
		m where o.sale_no = s.sale_no and m.partner_id=15 group by s.pro_info_no, m.partner_id order by 2 desc;
		15 3 [올리비아로렌]올리비아로렌 여름 마감임박 블라우스/티셔츠외 24% 189000
2	특정 협력사 별 상품 별 매출 현황	협력사_ID 상품코드 상품명
		15 1 [페이지플린]22%쿠폰+무배~썸머 베스트아이템 원피스/블라우스 外 495000
		623 개의 행이 선택되었습니다.

	리포트 항목	SQL 쿼리	
3	카테고리 별, 상품 별 매풀 현황	select s1.s_c_no 카테고리, p.pro_info_no 상품코드,(select pro_name from pro_info_p where pp.pro_info_no=p.pro_info_no),sum(o.o_quantity) 총수량, sum(o.final_sale_p 총매출 from pro_info p, sale_list s, order_list o, sub_category s1 where p.pro_info_no = s.pro_info_no and p.s_c_no = s1.s_c_no and s.sale_no = o.sale_no group by s1.s_c_p.pro_info_no order by p.pro_info_no desc; 42	rice)
4	카테고리 별 일일 판매 현황	select s1.s_c_no 카테고리, p.pro_info_no 상품코드, (select pro_name from pro_info where pp.pro_info_no=p.pro_info_no) 상품명,sum(o.o_quantity) 총수량, sum(o.final_sale_price) 총매출 from pro_info p ,sale_list s, sub_category s1,order_list o where s.pro_info_no = p.pro_info_no and s1.s_c_no = p.s_c_no and s.sale_no = o.sale_no and s.sale_date between '2021-01-01' and '2021-01-01' group by s1.s_c_no ,p.pro_info_no order by sum(s.final_sale_price) desc;	рр

	리포트 항목	SQL 쿼리	
4	카테고리 별 일일 매출 현황	카테고리 상품코드 상품명 총수량	
5	특정 일 판매된 상품 별 평균 매출	select p.pro_info_no 상품코드, (select pro_name from pro_info pp where pp.pro_info_no=p.pro_info_no) 상품명 ,avg(o.final_sale_price)평균 from pro_info p, sale_list s, order_list o where p.pro_info_no = s.pro_info_no and s.sale_no = o.sale_no and s.sale_date between '2021-01-01' and '2021-01-01' group by p.pro_info_no order by avg(o.final_sale_price) desc; 상품코드 상품명 평균	

	리포트 항목	SQL 쿼리
6 카E	디포트 양폭 전 년도 대비 테고리별, 분기별 출 증감율 현황	with A as (select p.s_c_no as 올해 , to_char (o.o_date,'Q') as 분기 , sum(o.final_sale_price) 매출 from sale_list s, pro_info p, order_list o where s.pro_info_no = p.pro_info_no and o.sale_no = s.sale_no and o.o_date between '2020-01-01' and '2020-12-31' group by p.s_c_no, to_char(o.o_date,'Q')), B as (select p.s_c_no as 작년 , to_char (o.o_date-365,'Q') as 분기, sum(o.final_sale_price) 매출 from sale_list s, pro_info p, order_list o where s.pro_info_no = p.pro_info_no and o.sale_no = s.sale_no and o.o_date between '2021-01-01' and '2021-12-31' group by p.s_c_no, to_char(o.o_date-365,'Q')) select A.올해,(select s_c_name from sub_category s where s.s_c_no=A.올해) 분류명 ,A.분 기,A.매출,B.작년, B.분기,B.매출 from A , B where A.올해 = B.작년;

6	직전 년도 대비 카테고리별, 분기별 매출 증감율 현황	용해 분류명 분기 매출 작년 분기 매출 41 카메라 4 62750800 41 1 69851000 41 카메라 3 11707900 41 1 69851000 41 카메라 4 62750800 41 3 5557450 41 카메라 3 11707900 41 3 5557450 42 게임 4 3993100 42 1 3508250 42 게임 3 581050 42 1 3508250 42 게임 4 3993100 42 3 3528350 42 게임 3 581050 42 3 3906250 42 게임 3 581050 42 2 3906250
7	매출 기준 우수 협력사 순위	select p1.partner_ID 협력사,(select partner_name from partner_info pp where pp.partner_id=p1.partner_id) 파트너명 , sum(o.final_sale_price) 총매출 from partner_use_info p1, pro_info p, sale_list s ,order_list o where p1.partner_ID = p.partner_ID and p.pro_info_no = s.pro_info_no and s.sale_no = o.sale_no group by p1.partner_ID order by sum(s.final_sale_price) desc; 점역사

	리포트 항목	SQL 쿼리	
8	할인률 별 상품 판매 현황	select d.DEAL_no 딜타입,count(*) 발생건수 ,(select deal_discount from deal_list dd where dd.deal_no=d.deal_no) 할인률 from deal_list d, order_list o, sale_list s where d.deal_no = s.deal_no and s.sale_no = o.sale_no group by d.DEAL_no;	

```
■ Function
                                                                            else
                                                                               v_age := trunc((v_year - (2000 + substr(user_jumin, 1, 2))),-1);
연령대 구하는 함수
                                                                               return v_age;
                                                                            end if;
create or replace function f_age(user_jumin varchar2)
                                                                         end;
   return number
is
                                                                         기간 중 토일을 제외한 배송일 산출
            number(1);
   v temp
            number(4);
   v_year
           number(3);
                                                                         create or replace function f_get_weekdays(v_sale_date date,v_day_count
   v_age
                                                                         number)
begin
                                                                             return date
   v_temp := substr(user_jumin, 8, 1);
                                                                         is
   v_year := to_char(sysdate,'yyyy');
                                                                             v_weekend number(3);
                                                                             v_return date;
   if v_{temp} = 1 or v_{temp} = 2 then
                                                                         begin
      v_age := trunc((v_year - (1900 + substr(user_jumin, 1, 2))), -1);
                                                                             SELECT count(d_day) into v_weekend FROM (
      return v_age;
```

```
SELECT (v_sale_date) + LEVEL as d_date,to_char(((v_sale_date) +
LEVEL), 'd') AS d_day FROM DUAL
       CONNECT BY LEVEL <= ((v_sale_date + v_day_count) -
v_sale_date)
                                                                         Procedure
   ) B WHERE B.d_day in (1,7);
   v_return:= v_sale_date + (v_day_count + v_weekend);
                                                                  주문 정보 추가 / 업데이트
   return (v_return);
end;
                                                                  CREATE OR REPLACE Procedure P Order List In Up(
                                                                      V_ORDER_NO
                                                                      V_O_RECIEVER_NAME
                                                                      V_O_RECIEVER_ADDR
    ■ Sequence
                                                                      V_USER_INFO_NO
                                                                      V O QUANTITY
주문 정보 시퀀스
                                                                      V_FINAL_SALE_PRICE
                                                                      V_SALE_NO
create Sequence SEQ_Order_List increment by 1 start with 1 maxvalue
                                                                      V_O_RECIEVER_TEL
1000000;
                                                                      V_O_DATE
```

ORDER_LIST.ORDER_NO%type,

ORDER_LIST.O_RECIEVER_NAME%type,

ORDER_LIST.O_RECIEVER_ADDR%type,

ORDER_LIST.USER_INFO_NO%type,

ORDER_LIST.FINAL_SALE_PRICE%type,

ORDER_LIST.O_RECIEVER_TEL%type,

ORDER_LIST.O_QUANTITY%type,

ORDER_LIST.SALE_NO%type,

ORDER_LIST.O_DATE%type

```
V_O_RECIEVER_ADDR,
                                                                                             V_USER_INFO_NO,
                                                                                                                    V_O_QUANTITY,
                                                                   V_FINAL_SALE_PRICE, V_SALE_NO, V_O_RECIEVER_TEL, V_O_DATE);
is
                                                                       END IF;
begin
                                                                   END;
                                              O RECIEVER NAME
   UPDATE
               ORDER LIST
                              SET
=V_O_RECIEVER_NAME,
                          O RECIEVER ADDR=V O RECIEVER ADDR,
                                                                       ■ Trigger
                          USER INFO NO=V USER INFO NO,
                          O_QUANTITY=V_O_QUANTITY,
                                                                   테이블 백업
                          FINAL_SALE_PRICE=V_FINAL_SALE_PRICE,
                                                                   create or replace trigger tri_order_list_b
                          SALE_NO=V_SALE_NO,
                                                                   after insert or delete or update on order list
                          O RECIEVER TEL=V O RECIEVER TEL,
                                                                   for each row
                          O_DATE=V_O_DATE
   WHERE ORDER_NO=V_ORDER_NO;
                                                                   begin
                                                                       if deleting then
   IF sql%rowcount=0 THEN
               INTO ORDER_LIST(ORDER_NO, O_RECIEVER_NAME,
                                                                          delete order_list_b where order_no=:old.order_no;
       INSERT
O_RECIEVER_ADDR, USER_INFO_NO, O_QUANTITY, FINAL_SALE_PRICE,
                                                                       else
SALE_NO, O_RECIEVER_TEL, O_DATE)
                                                                                                   order_list_b
                                                                          update
                                                                                                                                set
       VALUES(SEQ_Order_List.nextval,
                                           V_O_RECIEVER_NAME,
```

```
O_RECIEVER_NAME=:new.O_RECIEVER_NAME,
                                                                    w.O_RECIEVER_TEL,:new.O_DATE);
                                                                            end if;
O_RECIEVER_ADDR=:new.O_RECIEVER_ADDR,
                                                                        end if;
                              USER_INFO_NO=:new.USER_INFO_NO,
                                                                    end;
                               O_QUANTITY=:new.O_QUANTITY,
FINAL_SALE_PRICE=:new.FINAL_SALE_PRICE,
                              SALE_NO=:new.SALE_NO,
                                                                    협력사 업무 중지 시 해당되는 판매 상품에 대한 판매 중지 변경
O RECIEVER TEL=:new.O RECIEVER TEL,
                              O_DATE=:new.O_DATE
                                                                    create or replace trigger tri_partner_use_info_p_use
       where order no=:new.order no;
                                                                        after insert or delete or Update of P_USE_BOOL on partner_use_info
       if sql%rowcount=0 then
                                                                        for each row
            insert
                                                            into
order_list_b(order_no,O_RECIEVER_NAME,O_RECIEVER_ADDR,USER_INFO_
                                                                    declare
NO,O_QUANTITY,FINAL_SALE_PRICE,SALE_NO,O_RECIEVER_TEL,O_DATE)
                                                                    begin
           values
                                                                           if :new.P_USE_BOOL = '0' then
(:new.order_no,:new.O_RECIEVER_NAME,:new.O_RECIEVER_ADDR,:new.USE
R_INFO_NO,:new.O_QUANTITY,:new.FINAL_SALE_PRICE,:new.SALE_NO,:ne
                                                                               update sale_list set sale_bool='0' where pro_info_no in
```

(select pro_info_no from pro_info where partner_id=:new.partner_id)	for each row
end if;	declare
end;	begin
	INSERT INTO USER_Service(User_Service_No, Sign_date, Pass_Ch_Date, Last_Conn, Conn_num, Order_Num, User_Info_No,Resign_date,Resign_bool) VALUES
판매 시작 직후 판매 마감일자 자동 업데이트	(SEQ_UserService.nextval,sysdate,sysdate,:new.last_conn,0,0,:new.user_info _no,sysdate,1); end;
select to_char(to_date('2021-01-01') + level - 1,'YYYY-MM-DD'),to_char(to_date('2021-01-31') + level-1, 'D') D from dual	
 회원의 접속 정보 백업	
create or replace trigger tri_user_info_last_conn	적용 딜 % 정보 변경 / 이전 주문의 딜 적용가 변경
after insert or delete or Update of last_conn on user_info	

create or replace trigger tri_order_info_deal_count
after insert or delete or Update of O_QUANTITY on order_list
for each row

declare

v_people deal_list.people_num%type;

v_sale_bool sale_list.sale_bool%type;

v_deal_discount deal_list.deal_discount%type;

begin

select sale_bool into v_sale_bool from sale_list where sale_no=:old.sale_no;

if v_sale_bool='1' then

select count(*) into v_people from order_list where
sale_no=:old.sale_no;

UPDATE SALE_LIST SET deal_no=(SELECT min(deal_no)
FROM deal_list where people_num >= v_people)

WHERE sale_no=:old.sale_no;

select deal_discount into v_deal_discount from deal_list where deal_no= (SELECT min(deal_no) FROM deal_list where people_num