/\*

1. 제품 테이블은 제품\_ID 컬럼이 \_\_\_기본키(Primary Key)\_\_\_ 컬럼으로 그 행을 다른 행과 식별할 때 사용된다.

2. 제품 테이블의 제조사 컬럼은 Not Null(NN) 인 것으로 봐서 \_\_\_Null\_\_\_ 인 상태일 수가 없다.

3. 고객 테이블에서 다른행과 식별할 때 사용하는 컬럼은 \_\_\_고객\_ID(cust\_id)\_\_\_\_ 이다.

4. 고객 테이블의 전화번호 컬럼의 데이터 타입은 \_\_\_\_\_VARCHAR2\_\_\_ 으로 \_\_\_\_문자열\_\_\_형태의 값 \_\_\_15\_\_\_\_바이트 저장할 수 있으며 NULL 값을 \_\_\_가질 수 있다\_\_\_.

5. 고객 테이블의 가입일 컬럼에 대해 4번 처럼 설명해 보시오.

기입일 컬럼의 데이터 타입은 Date으로 날짜 형태의 값을 저장할 수 있으며 NULL을 가질 수 없다.

6. 주문 테이블은 총 5개 컬럼이 있다. 정수 타입이 \_\_3\_\_개이고 문자열 타입이 \_\_1\_\_개 이고 날짜 타입이 \_\_1\_\_개이다.

7. 고객 테이블과 주문테이블은 서로 관계가 있는 테이블입니다.

부모테이블은 \_\_고객테이블\_\_\_ 이고 자식 테이블은 \_\_\_\_주문테이블\_\_\_\_이다.

부모테이블의 \_\_고객\_ID\_\_\_컬럼을 자식테이블의 \_\_\_고객\_ID\_\_\_\_컬럼이 참조하고 있다.

고객테이블의 한행의 데이터는 주문테이블의 \_\_\_0~n\_\_\_ 행과 관계가 있을 수 있다.

주문테이블의 한행은 고객테이블의 \_\_\_\_1\_\_\_\_\_행과 관계가 있을 수 있다.

8. 주문 테이블과 주문\_제품 테이블은 서로 관계가 있는 테이블입니다.

부모 테이블은 \_\_\_주문테이블\_\_\_\_\_ 이고 자식 테이블은 \_\_\_주문\_제품 테이블\_\_\_\_이다.

부모 테이블의 \_\_\_\_주문\_ID\_\_\_\_컬럼을 자식 테이블의 \_\_\_\_\_주문\_ID\_\_\_\_\_\_컬럼이 참조하고 있다.

주문 테이블의 한행의 데이터는 주문\_제품 테이블의 \_\_\_\_0~n\_\_\_\_ 행과 관계가 있을 수 있다.

주문\_제품 테이블의 한행은 주문 테이블의 \_\_\_\_\_1\_\_\_\_\_행과 관계가 있을 수 있다.

9. 제품과 주문\_제품은 서로 관계가 있는 테이블입니다.

부모 테이블은 \_\_\_\_제품테이블\_\_\_\_\_\_\_ 이고 자식 테이블은 \_\_\_주문\_제품 테이블\_\_\_\_이다.

부모 테이블의 \_\_\_\_제품\_ID\_\_\_컬럼을 자식 테이블의 \_\_제품\_ID\_\_\_\_\_\_\_컬럼이 참조하고 있다.

제품 테이블의 한행의 데이터는 주문\_제품 테이블의 \_\_\_0~n\_\_\_\_\_ 행과 관계가 있을 수 있다.

주문\_제품 테이블의 한행은 제품 테이블의 \_\_\_1\_\_\_\_행과 관계가 있을 수 있다.

\*/

-- TODO: 4개의 테이블에 어떤 값들이 있는지 확인.

select \* from products;

select \* from order\_items;

select \* from orders;

select \* from customers;

-- TODO: 주문 번호가 1인 주문의 주문자 이름, 주소, 우편번호, 전화번호 조회

select c.cust\_name, c.address, c.postal\_code, c.phone\_number

from customers c join orders o on c.cust\_id=o.cust\_id

where o.order\_id=1;

-- TODO : 주문 번호가 2인 주문의 주문일, 주문상태, 총금액, 주문고객 이름, 주문고객 이메일 주소 조회

select o.order\_date, o.order\_status, o.order\_total, c.cust\_name, c.cust\_email

from customers c join orders o on c.cust\_id=o.cust\_id

where o.order\_id = 2;

-- TODO : 고객 ID가 120인 고객의 이름, 성별, 가입일과 지금까지 주문한 주문정보중 주문\_ID, 주문일,

--총금액을 조회

select c.cust\_name, c.gender, c.join\_date, o.order\_id, o.order\_date, o.order\_total

from customers c join orders o on c.cust\_id = o.cust\_id

where c.cust\_id = 120;

-- TODO : 고객 ID가 110인 고객의 이름, 주소, 전화번호, 그가 지금까지 주문한 주문정보중 주문\_ID,

--주문일, 주문상태 조회

select c.cust\_name, c.address, c.phone\_number, o.order\_id, o.order\_date, o.order\_status

from customers c join orders o on c.cust\_id = o.cust\_id

where c.cust\_id = 110;

-- TODO : 고객 ID가 120인 고객의 정보와 지금까지 주문한 주문정보를 모두 조회.

select c.\*, o.\*

from customers c join orders o on c.cust\_id = o.cust\_id

where c.cust\_id = 120;

-- TODO : '2017/11/13'(주문날짜) 에 주문된 주문의 주문고객의 고객\_ID, 이름, 주문상태, 총금액을 조회

select c.cust\_id, c.cust\_name, o.order\_status, o.order\_total

from customers c join orders o on c.cust\_id = o.cust\_id

where o.order\_date = '2017/11/13';

-- TODO : 주문상세 ID가 xxxx인 주문제품의 제품이름, 판매가격, 제품가격을 조회.

select p.product\_name, o.sell\_price, p.price

from order\_items o join products p on o.product\_id=p.product\_id

where o.ordere\_item\_id= 1;

-- TODO : 주문 ID가 4인 주문의 주문 고객의 이름, 주소, 우편번호, 주문일, 주문상태, 총금액,

--주문 제품이름, 제조사, 제품가격, 판매가격, 제품수량을 조회.

select c.cust\_name, c.address, c.postal\_code, o.order\_date, o.order\_status, o.order\_total,

p.product\_name, p.maker, p.price, oi.sell\_price, oi.quantity

from order\_items oi join orders o on oi.order\_id = o.order\_id

join products p on oi.product\_id = p.product\_id

join customers c on o.cust\_id = c.cust\_id

where o.order\_id = 4;

-- TODO : 제품 ID가 200인 제품이 2017년에 몇개 주문되었는지 조회.

select count(\*) "2017년 판매수량"

from order\_items oi join orders o on oi.order\_id = o.order\_id

where oi.product\_id = 200

and to\_char(o.order\_date ,'yyyy')=2017;

--ORACLE문법

select count(\*)

from order\_items oi, orders o

where oi.order\_id = o.order\_id

and oi.product\_id = 200

and to\_char(o.order\_date,'yyyy')=2017;

-- TODO : 제품분류별 총 주문량을 조회

select p.category , count(\*) 주문량

from products p join order\_items oi on p.product\_id = oi.product\_id

group by p.category;

--outer join하기

select p.category , nvl(sum(oi.quantity),0) 주문량

from products p left join order\_items oi on p.product\_id = oi.product\_id

group by p.category;

--ORACLE문법

select decode(grouping\_id(p.category),1,'총주문량',p.category)category , nvl(sum(oi.quantity),0) 주문량

from products p, order\_items oi

where p.product\_id = oi.product\_id(+)

group by rollup(p.category);

insert into products values(600,'새제품','아이스크림','메이커1',3000);

insert into products values(610,'새제품','아이스크림','메이커1',3000);

commit;

select \*

from customers natural join orders;

select \*

from customers join orders using(cust\_id);

select \* from customers cross join orders;

select \* from customers, orders;