

[사용테이블]● **PRODUCT Table(제품)**

설명	컬럼	타입	제약조건	참조테이블/컬럼
제품번호	PDNO	NUMBER	PK	
제품카테고리	PDNAME	VARCHAR2(10)		
제품명	PDSUBNAME	VARCHAR2(10)		
공장번호	FACTNO	VARCHAR2(5)	FK	FACTORY/ FACTNO
제품생산일	PDDATE	DATE		
제품원가	PDCOST	NUMBER		제품의 생산 원가
제품가격	PDPRICE	NUMBER		판매점에 판매되는 가격
재고수량	PDAMOUNT	NUMBER		공장 내 재고 수량

● **FACTORY Table(공장)**

설명	컬럼	타입	제약조건	참조테이블/컬럼
공장번호	FACTNO	VARCHAR2(5)	PK	
공장명	FACNAME	VARCHAR2(14)		
공장위치	FACLOC	VARCHAR2(13)		

● **STORE Table(판매점)**

설명	컬럼	타입	제약조건	참조테이블/컬럼
판매점번호	STONO	VARCHAR2(5)	PK	
판매점명	STONAME	VARCHAR2(14)		
제품번호	PDNO	NUMBER	FK	PRODUCT/ PDNO
판매점재고수량	STAMOUNT	NUMBER		판매점 내 재고 수량
판매수량	STSALES	NUMBER		
판매가격	STPRICE	NUMBER		소비자에게 판매되는 가격

* 전자제품을 생산하는 회사로서 전국에 생산 공장과 판매점을 두고 있는 회사 입니다.

* 판매점에서는 모든 제품들의 생산 공장번호와 제품번호를 관리 합니다.

* 공장과 판매점에서 제품에 대한 재고를 각각 관리 한다.

[문제 1] 아래 요구사항에 맞는 프로그램을 작성 하시오.

1. 요구사항

제품이 생산된 공장위치가 "SEOUL"인 제품 중 판매점에 재고가 없는 상품을 출력한다.

조건:

1) 재고가 없는 조건은 재고 수량이 0이거나 null을 의미한다.

2) null인 경우 "0" 표시 한다.

2. 구현 클래스

Package명	클래스명	메소드	설명
workshop08	JDBCTest1	<u>+main(String args[]):void</u>	main 함수 안에서 모든 코드 작업 진행

3. 실행 결과 예

제품카테고리	제품명	공장명	판매점명	판매점재고수량
-----+	-----+	-----+	-----+	-----
TV	INFINIA-01	SEOUL FACTORY	TESCO	0
TV	INFINIA-01	SEOUL FACTORY	TESCO	0
WASHER	PA-01	SEOUL FACTORY	TESCO	0

[문제 2] 아래 요구사항에 맞는 프로그램을 작성 하시오.

1. 요구사항

제품카테고리가 "TV"인 제품 중 가장 싼 것보다 비싼 모든 제품과,
제품카테고리가 "CELLPHONE"인 제품 중 가장 비싼 제품보다 싼 모든 제품을 출력한다.

조건:

- 1) UNION을 사용하지 않고 하나의 쿼리 문장으로 작성 한다.
- 2) 제품원가를 기준으로 한다.

2. 구현 클래스

Package명	클래스명	메소드	설명
workshop08	JDBCTest2	<u>+main(String args[]):void</u>	main 함수 안에서 모든 코드 작업 진행

3. 실행 결과 예

제품명	제품원가	제품가격
-----+-----+-----		
XCANVAS-01	800000	1200000
INFINIA-01	830000	970000
OPTIMUS-01	830000	890000
OPTIMUS-02	940000	990000

[문제 3,4,5] 문제에서 요구하는 쿼리문장 및 프로그램을 작성 하시오.

공장 위치가 'CHANGWON'에서 생산된 제품들에 결함이 발견되어 생산된 모든 제품을 폐기 하고자 한다.

[문제 3] 아래 요구사항에 맞는 테이블을 생성하는 쿼리문장을 작성 하시오.

1. 요구사항

신규 테이블을 만들어 폐기 되는 모든 데이터를 관리하고자 한다.

테이블 정의

TABLE NAME: **DISCARDED_PRODUCT**

설명	컬럼	타입	제약조건	참조테이블/컬럼
제품번호	PDNO	NUMBER	PK	
제품카테고리	PDNAME	VARCHAR2(10)		
제품명	PDSUBNAME	VARCHAR2(10)		
공장번호	FACTNO	VARCHAR2(5)	FK	FACTORY/ FACTNO
제품생산일	PDDATE	DATE		
제품원가	PDCOST	NUMBER		제품의 생산 원가
제품가격	PDPRICE	NUMBER		판매점에 판매되는 가격
재고수량	PDAMOUNT	NUMBER		공장 내 재고 수량
폐기날짜	DISCARDED_DATE	DATE		

[문제 4] 아래 요구사항에 맞는 프로그램을 작성 하시오.**1. 요구사항**

PRODUCT 테이블에서 폐기 되는 제품정보들을 모두 조회 하여 DISCARDED_PRODUCT 테이블로 INSERT 한다. 단, 트랜잭션 처리를 반드시 한다.

조건:

- 1) 폐기 날짜는 현재 시스템 날짜로 한다.

2. 구현 클래스

Package명	클래스명	메소드	설명
workshop08	JDBCTest3	<u>+main(String args[]):void</u>	main 함수 안에서 모든 코드 작업 진행

3. 실행 결과 예

2개의 데이터가 정상적으로 INSERT 되었습니다.

[문제 5] 아래 요구사항에 맞는 프로그램을 작성 하시오.

1. 요구사항

[문제 4]에서 폐기된 제품을 PRODUCT 테이블에서 모두 삭제 한다. 단, 트랜잭션 처리를 반드시 한다.

2. 구현 클래스

Package명	클래스명	메소드	설명
workshop08	JDBCTest4	<u>+main(String args[]):void</u>	main 함수 안에서 모든 코드 작업 진행

3. 실행 결과

2개의 데이터가 정상적으로 DELETE 되었습니다.

[문제 6] 아래 요구사항에 맞는 프로그램을 작성하시오.**1. 요구사항**

DTO 패턴과 DAO 패턴을 사용하여 FACTORY 테이블의 모든 데이터를 조회한다. 데이터 출력은 JDBCTest5 클래스의 main 메서드에서 작업한다.

2. 클래스 다이어그램

FactoryDAO
+ getFactoryAll(con:Connection): ArrayList<FactoryDTO>

JDBCTest5
+ main(args:String[]):void

FactoryBiz
- driver:String - url:String - user:String - pass:String
+ FactoryBiz() + FactoryBiz(driver:String, url:String, user:String, pass:String) + getFactoryAll():ArrayList<FactoryDTO>

FactoryDTO
- facNo:String - facName:String - facLoc:String
+ FactoryDTO() + FactoryDTO(facNo:String, facName:String, facLoc:String) + getXXX() + setXXX()

3. 구현 클래스

Package명	클래스명	메소드	설명
workshop08.factory	JDBCTest5	<u>+main(String args[]):void</u>	main 함수 안에서 모든 코드 작업 진행
	FactoryDAO	+getFactoryAll(con:Connection) :ArrayList<FactoryDTO>	모든 공장정보를 조회하고 ArrayList로 반환한다.
	FactoryBiz	+FactoryBiz()	기본 생성자
		+FactoryBiz(driver:String, url:String, user:String, pass:String)	4개의 클래스 변수를 받는 생성자
		+getFactoryAll() :ArrayList<FactoryDTO>	조회에 필요한 Connection 객체를 만들고, FactoryDAO 클래스의 getFactoryAll()를 호출한다.

- Getter/Setter는 필요 시 직접 구현한다.

4. 클래스 구조

```
public class JDBCTest5 {
    public static void main(String args[]) {
        // FactoryBiz 객체 생성
        // getFactoryAll() 함수를 이용하여 내용을 출력 한다.
    }
}
```

4. 실행 결과 예

공장번호	공장명	공장위치
98710	SEOUL FACTORY	SEOUL
98711	PUSAN FACTORY	PUSAN
98712	CHANGWON FACTORY	CHANGWON
98713	OSAN FACTORY	OSAN

workshop08.factory.FactoryDTO
<ul style="list-style-type: none"> factNo: String facName: String facLoc: String
<ul style="list-style-type: none"> FactoryDTO(factNo: String, facName: String, facLoc: String) FactoryDTO() getFactNo(): String setFactNo(factNo: String): void getFacName(): String setFacName(facName: String): void getFacLoc(): String setFacLoc(facLoc: String): void toString(): String

workshop08.factory.FactoryBiz
<ul style="list-style-type: none"> driver: String url: String user: String pass: String
<ul style="list-style-type: none"> FactoryBiz() FactoryBiz(driver: String, url: String, user: String, pass: String) getFactoryAll(): ArrayList<FactoryDTO>

workshop08.factory.FactoryDAO
<ul style="list-style-type: none"> getFactoryAll(con: Connection): ArrayList<FactoryDTO>