**학번 : 202084676**

**성명 : 박상열**

데이터베이스

설계 및 구축

프로젝트 보고서

재고 관리 시스템

목차

[1. 프로젝트 개요 2](#_Toc56128763)

[2. 요구사항 분석 3](#_Toc56128764)

[4 물리적 데이터 모델 5](#_Toc56128765)

[5 용어 사전 정의 6](#_Toc56128766)

[6. 테이블술서 7](#_Toc56128767)

[7. SQL문 사용하기 12](#_Toc56128768)

[**1) 기본 SQL (select, where, 그룹함수, group by, having, order by)-기본 5개** 12](#_Toc56128769)

[**2) JOIN – 기본 2개** 12](#_Toc56128770)

[**3) Sub Query – 기본 3개** 12](#_Toc56128771)

8. 별첨(프로젝트 후기, Databace 구축 자료) 13

# 1. 프로젝트 개요

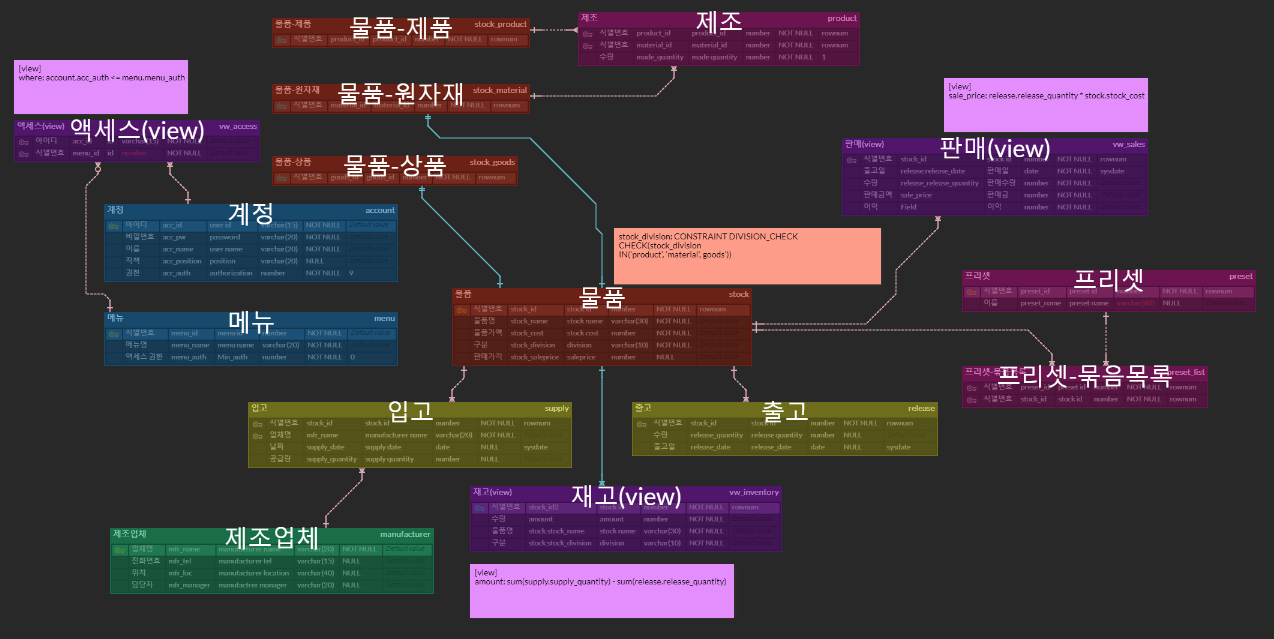
|  |  |
| --- | --- |
| 프로젝트 개요 | |
| **프로젝트 명** | 재고 관리 시스템 |
| 프로젝트 명에서 명시한 것처럼 식당, 카페, 의류샵 외 기타 점포 등의 저자본 소상공인들을 대상으로 제작하는 프리웨어 재고 관리 프로그램입니다. 기본적인 입고, 출고 기능과 매출 관리 기능을 포함합니다.  프로그램의 기능 중 물품의 구분과 물품 묶음 기능이 있습니다. 모든 물품은 상품, 제품, 원자재로 구분됩니다. 상품이란 공급 업체로부터 사들여 2차 가공 없이 판매하는 물품을 의미하며(의류, 음료수 캔 등의 완제품 식품) 제품은 원자재를 공급받아 2차 가공 후 판매하는 물품을 의미합니다. (김과 김밥 속재료(원재료)를 공급받아 제작한 김밥(제품))  그리고 해당 프로그램은 기본적으로 당일의 영업 종료후 정산시에 판매 데이터를 입력하는 것을 전제로 만들어지며 입력의 용이성을 위해 프리셋을 제공합니다. 프리셋이란 사용자의 편의에 따라 상품이나 제품을 하나의 항목으로 묶은 것을 의미합니다. (빵, 패티, 상추를 이용하여 만든 햄버거와 완제품 감자튀김, 캔 음료수가 함께 판매되는 A세트를 프리셋으로 구성 가능함. 혹은 옷가게에서 행사로 1+1으로 판매하는 옷 2개를 묶은 프리셋으로 구성 가능함.)    전체적인 메뉴 흐름은 위와 같습니다. | |

# 2. 요구사항 분석

|  |  |
| --- | --- |
| 요구사항 분석 | |
| **프로젝트 명** | 재고 관리 시스템 |
| **1. 요구사항**   1. 재고 관리 프로그램을 이용하려면 아이디, 비밀번호, 이름, 직책, 권한이 포함된 계정이 필요하다. 2. 회원 가입 폼은 별도로 사용하지 않으며, 최초 설정된 관리자 계정을 이용한다. 3. 각각의 관리 메뉴에 액세스하기 위해서는 일정 수준 이상의 권한이 필요하다. 4. 각각의 관리 메뉴는 액세스 권한이 별도로 주어진다. 5. 관리자 계정은 모든 메뉴 액세스 권한을 갖고 있으며, 다른 계정들의 권한도 설정할 수 있다. 6. 각 계정은 아이디로 식별한다. 7. 모든 물품에 대한 물품명, 원가, 물품 구분 정보를 유지해야한다. 8. 입고되는 물품은 상품과 원자재로 구분된다. 9. 각 상품 및 원자재는 한 제조업체가 공급하고, 제조업체 하나는 여러 상품을 공급할 수 있다. 10. 제조업체에 대한 제조업체명, 전화번호, 위치, 담당자 정보를 유지해야한다. 11. 점주는 입고된 원자재를 사용한 제품을 등록할 수 있다. 12. 제품에 대한 사용 원자재 종류 및 수량 정보를 유지해야한다. 13. 판매 등록시 제품과 상품, 프리셋의 수량 감소가 이루어진다. 제품의 경우 사용된 원자재의 감소, 프리셋의 경우 묶음처리된 제품 혹은 상품 단위로 출고처리한다. 14. 모든 입고와 출고에 대한 물품명, 입고일, 출고일, 수량 정보를 유지해야 한다. 15. 판매 등록에서 등록한 판매에 대한 물품명, 판매일, 수량, 판매 금액 정보, 이익 정보를 유지해야한다.   **2. 제공 기능**   * 입고 물품의 구분 기능(상품, 원자재) * 판매시 입고품의 묶음 출고처리 기능(제품, 프리셋)(하단 용어사전 참조) * 입출고 내역 조회 * 공급 업체의 정보 저장 * 등록된 상품 및 제품 조회 * 잔여 재고 및 소비 내역 조회 * 이익 데이터에 기반한 일, 주, 월간 매출 정보 저장 및 분석 * 출고 데이터에 기반한 재고 소비량 통계 제공 * 메뉴에 엑세스할 수 있는 권한 조정과 그에 따른 다중 계정 기능 | |

3. 논리적 데이터 모델

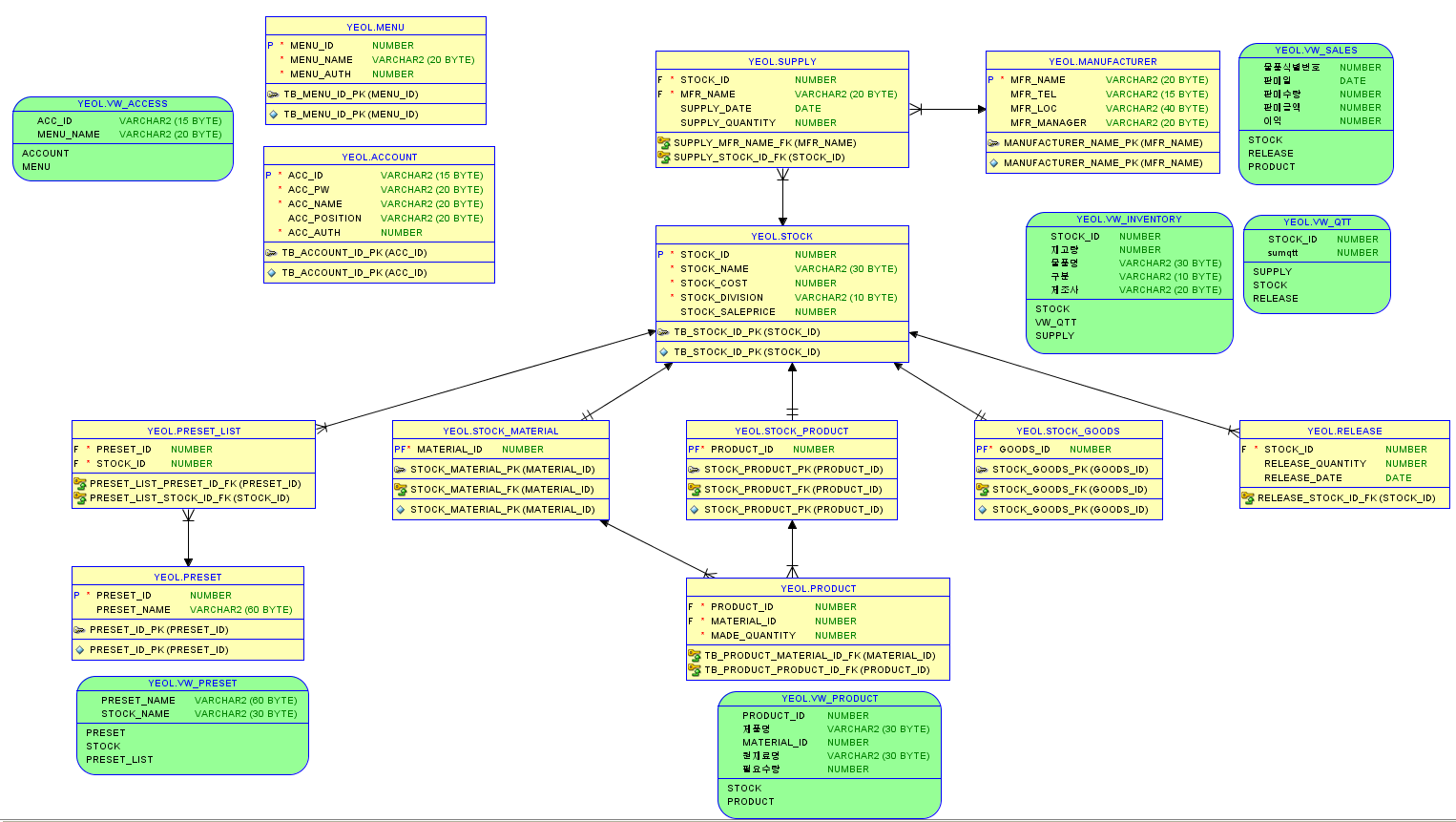
|  |  |
| --- | --- |
| 논리적 데이터 모델(ERD) | |
| **프로젝트 명** | 재고 관리 시스템 |



# 4 물리적 데이터 모델

|  |  |
| --- | --- |
| 물리적 데이터 모델(관계형 데이터 모델) | |
| **프로젝트 명** | 재고 관리 시스템 |





# 5 용어 사전 정의

|  |  |  |  |
| --- | --- | --- | --- |
| 용어 사전 | | | |
| **프로젝트 명** | 재고 관리 시스템 | | |
| **논리명** | **물리명** | **약어** | **설 명** |
| 계정 테이블 | account |  | 사용자 계정이 저장되는 테이블 |
| 계정 아이디 | acc\_id |  | 사용자 아이디 |
| 계정 비밀번호 | acc\_pw |  | 유저 패스워드 |
| 사용자명 | acc\_username |  | 계정 사용자 이름 |
| 직책 | acc\_position |  | 계정 사용자 직책 |
| 권한 | acc\_auth |  | 사용자가 시스템 내에서 메뉴에 액세스할 수 있는 권한 |
| 메뉴 테이블 | menu |  | 메뉴 이름과 권한이 저장되는 테이블 |
| 식별번호 | menu\_id |  | 해당 메뉴 고유 식별번호 |
| 메뉴명 | menu\_name |  | 해당 메뉴 이름 |
| 액세스 권한 | menu\_auth |  | 해당 메뉴에 액세스하기위한 필요 권한 |
| 액세스 view | vw\_access |  | 계정, 메뉴 테이블에서 데이터를 조회해 해당 계정이 액세스할 수 있는 메뉴를 표시 |
| 물품 테이블 | stock |  | 등록된 모든 물품이 등록되는 테이블 |
| 물품명 | stock\_name |  |  |
| 물품가액 | stock\_cost |  | 입고가격 |
| 판매가격 | stock\_saleprice |  | 판매가격 |
| 구분 | stock\_division |  | 해당 물품의 분류를 나타냄. product, material, goods |
| 제조업체 테이블 | manufacturer |  | 물품 공급을 담당하는 제조업체를 기록하는 테이블 |
| 업체명 | mfr\_name |  |  |
| 전화번호 | mfr\_tel |  |  |
| 위치 | mfr\_loc | location |  |
| 담당자 | mfr\_manager |  |  |
| 입고 테이블 | supply |  | 입고 기록을 남기는 테이블 |
| 날짜 | supply\_date |  |  |
| 공급량 | supply\_quantity |  |  |
| 출고 테이블 | release |  | 제품 출고 기록을 남기는 테이블 |
| 수량 | release\_quantity |  |  |
| 출고일 | release\_date |  |  |
| 재고view | vw\_inventory |  | 입고와 출고 테이블을 조회하여 제품의 식별번호, 재고량, 물품명, 구분, 제조사를 표시하는 뷰 |
| 상품 | goods |  | 공급받은 그대로 판매할 수 있는 상품. 이미 완성되어 있음 |
| 원자재 | material |  | 공급받은 후 후가공이 필요한 모든 원자재. |
| 제품 | product |  | 공급받은 원자재를 가공하며 완성된 물품. |
| 물품-상품 테이블 | stock\_goods |  | stock테이블에서 division=goods만 가져와 저장하는 테이블 |
| 물품-원자재 테이블 | stock\_material |  | stock테이블에서 division=material만 가져와 저장하는 테이블 |
| 물품-제품 테이블 | stock\_product |  | stock테이블에서 division=product만 가져와 저장하는 테이블 |
| 제조 테이블 | product |  | stock\_product, stock\_material 테이블을 조회해 제품 제조에 필요한 원자재의 식별번호를 기록하는 테이블 |
| 수량 | made\_quantity |  | 해당 제품을 제조하는 데에 필요한 해당 원자재의 수량 |
| 제조view | vw\_product |  | 제조, 물품 테이블을 조회하여 제품별로 필요한 원자재의 제품명을 보여주는 view |
| 프리셋 테이블 | preset |  | 프리셋의 식별번호와 이름을 기록하는 테이블 |
| 프리셋 | preset |  | 상품이나 제품 등 판매 가능한 물품을 묶어놓은 상태 개별 판매도 상관없이 가능함 |
| 프리셋-묶음목록 테이블 | preset\_list |  | 프리셋, 물품 테이블에 조회하여 해당 프리셋에 묶음처리된 물품의 식별번호를 저장하는 테이블 |
| 판매view | vw\_sales |  | stock, release, product테이블을 조회해 판매된 물품의 식별번호와 출고일, 판매 수량, 판매 금액과 입고 금액 대비 판매 이익을 산출해 보여주는 뷰 제품 판매의 경우 제품에 사용된 원자재를 계산함 |

# 6. 테이블기술서

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 테이블 기술서 | | | | | | |
| **테이블 명** | | ACCOUNT | | | | |
| **테이블 설명** | | 계정 정보를 저장하는 테이블 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | ACC\_ID | VARCHAR2(15) | Y | PK |  |  |
| 2 | ACC\_PW | VARCHAR2(20) | Y |  |  |  |
| 3 | ACC\_NAME | VARCHAR2(20) | Y |  |  |  |
| 4 | ACC\_POSITION | VARCHAR2(20) | N |  |  |  |
| 5 | ACC\_AUTH | NUMBER | Y |  | 9 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **비 고** | | | | | | |
| 권한은 9~0까지 존재하고 9가 가장 낮으며 0이 가장 높은 관리자 권한 | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | MENU | | | | |
| **테이블 설명** | | 프로그램에 사용될 메뉴와 해당 메뉴에 액세스하기 위한 권한을 저장 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | MENU\_ID | NUMBER | Y | PK |  |  |
| 2 | MENU\_NAME | VARCHAR2(20) | Y |  |  |  |
| 3 | MENU\_AUTH | NUMBER | Y |  | 0 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **비 고** | | | | | | |
|  | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | STOCK | | | | |
| **테이블 설명** | | 데이터를 유지하는 모든 물품을 기록 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | STOCK\_ID | NUMBER | Y | PK |  |  |
| 2 | STOCK\_NAME | VARCHAR2(30) | Y |  |  |  |
| 3 | STOCK\_COST | NUMBER | Y |  |  | 입고가격 |
| 4 | STOCK\_DIVISION | VARCHAR2(10) | Y |  |  | CHECK(PRODUCT, GOODS, MATERIAL) |
| 5 | STOCK\_SALEPRICE | NUMBER | N |  |  | 판매가격 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **비 고** | | | | | | |
|  | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | STOCK\_GOODS | | | | |
| **테이블 설명** | | STOCK 테이블에서 DIVISION=GOODS인 상품들 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | GOODS\_ID | NUMBER | Y | PK  FK |  | REFERENCES STOCK(STOCK\_ID) |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **비 고** | | | | | | |
| STOCK\_DIV\_CLASSIFICATION 트리거로 작동함. | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | STOCK\_MATERIAL | | | | |
| **테이블 설명** | | STOCK 테이블에서 DIVISION=MATERIAL인 상품들 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | MATERIAL\_ID | NUMBER | Y | PK  FK |  | REFERENCES STOCK(STOCK\_ID) |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **비 고** | | | | | | |
| STOCK\_DIV\_CLASSIFICATION 트리거로 작동함. | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | STOCK\_PRODUCT | | | | |
| **테이블 설명** | | STOCK 테이블에서 DIVISION=PRODUCT인 상품들 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | PRODUCT\_ID | NUMBER | Y | PK  FK |  | REFERENCES STOCK(STOCK\_ID) |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **비 고** | | | | | | |
| STOCK\_DIV\_CLASSIFICATION 트리거로 작동함. | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | PRODUCT | | | | |
| **테이블 설명** | | STOCK\_PRODUCT와 STOCK\_MATERIAL테이블을 참조해 제품 제조에 필요한 원자재를 기록하는 테이블 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | PRODUCT\_ID | NUMBER | Y | FK |  | REFERENCES STOCK\_PRODUCT(PRODUCT\_ID) |
| 2 | MATERIAL\_ID | NUMBER | Y | FK |  | REFERENCES STOCK\_MATERIAL(MATERIAL\_ID) |
| 3 | MADE\_QUANTITY | NUMBER | Y |  | 1 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **비 고** | | | | | | |
|  | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | MANUFACTURER | | | | |
| **테이블 설명** | | 물품을 공급하는 제조업체를 기록하는 테이블 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | MFR\_NAME | VARCHAR2(20) | Y | PK |  |  |
| 2 | MFR\_TEL | VARCHAR2(15) | N |  |  |  |
| 3 | MFR\_LOC | VARCHAR2(40) | N |  |  |  |
| 4 | MFR\_MANAGER | VARCHAR2(20) | N |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **비 고** | | | | | | |
|  | | | | | | |

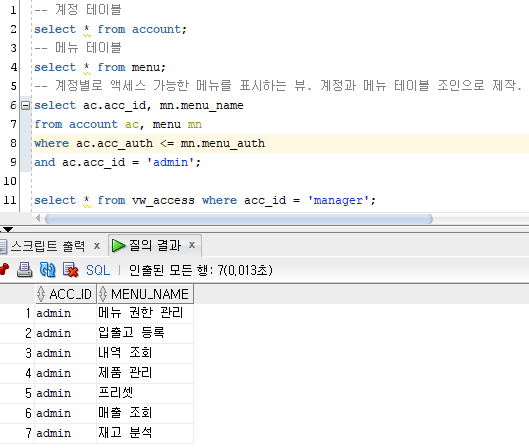
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | SUPPLY | | | | |
| **테이블 설명** | | 입고현황을 기록하는 테이블 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | STOCK\_ID | NUMBER | Y | FK |  | REFERENCES STOCK(STOCK\_ID) |
| 2 | MFR\_NAME | VARCHAR2(20) | Y | FK |  | REFERENCES MANUFACTURER(MFR\_NAME) |
| 3 | SUPPLY\_DATE | DATE | N |  | SYSDATE |  |
| 4 | SUPPLY\_QUANTITY | NUMBER | N |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **비 고** | | | | | | |
|  | | | | | | |

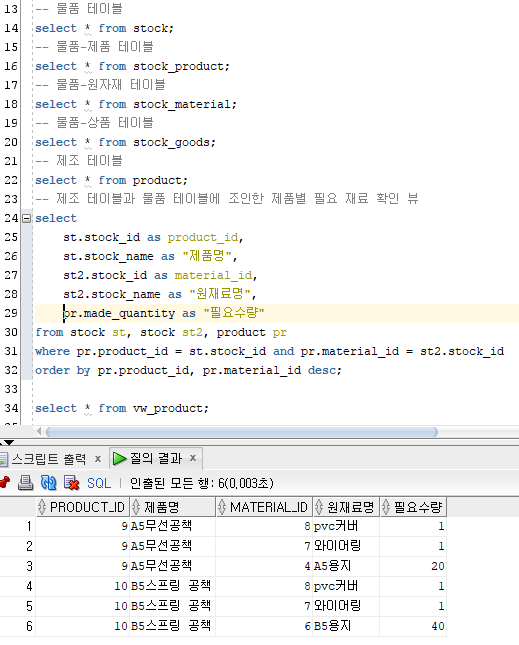
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | RELEASE | | | | |
| **테이블 설명** | | 출고현황을 기록하는 테이블 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | STOCK\_ID | NUMBER | Y | FK |  | REFERENCES STOCK(STOCK\_ID) |
| 2 | RELEASE\_QUANTITY | NUMBER | N |  |  |  |
| 3 | RELEASE\_DATE | DATE | N |  | SYSDATE |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **비 고** | | | | | | |
|  | | | | | | |

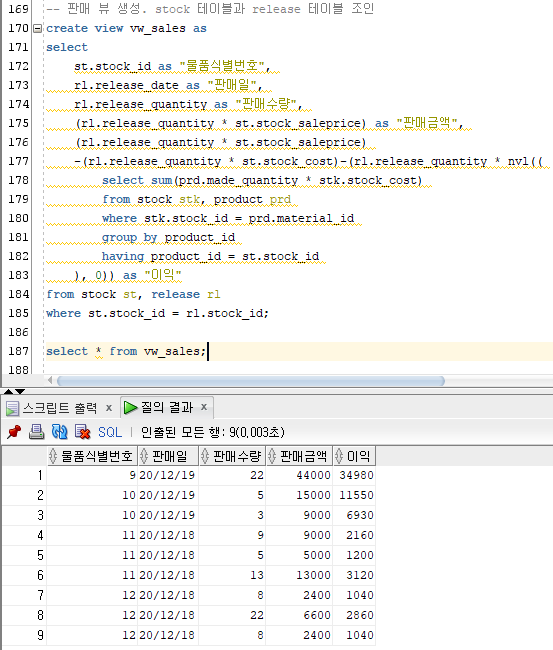
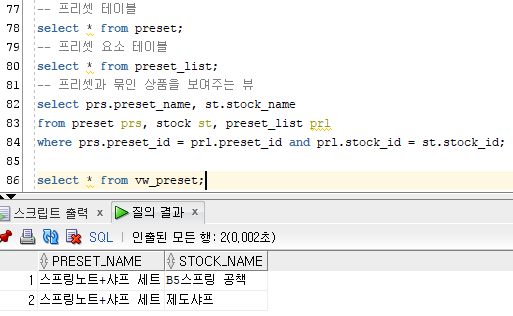
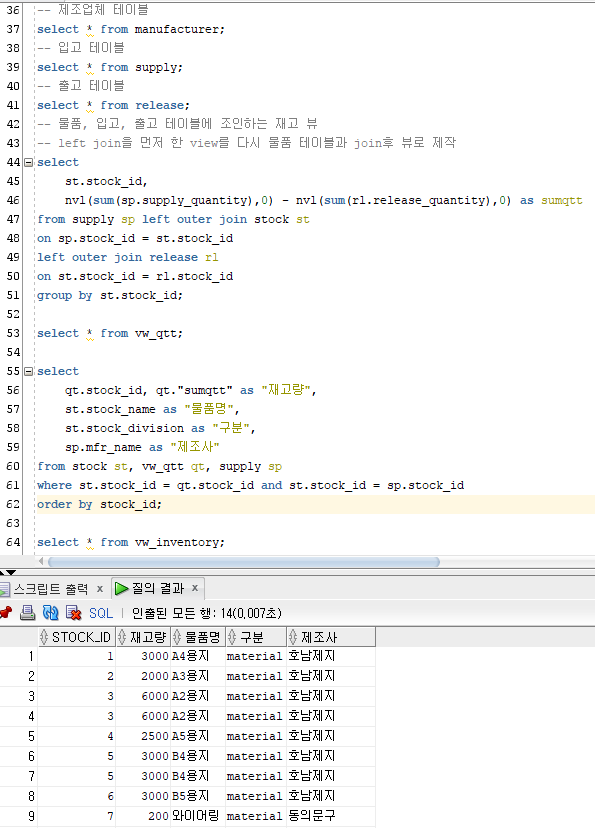
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | PRESET | | | | |
| **테이블 설명** | | 프리셋 이름을 기록하는 테이블 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | PRESET\_ID | NUMBER | Y | PK |  |  |
| 2 | PRESET\_NAME | VARCHAR(60) | N |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **비 고** | | | | | | |
|  | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | PRESET\_LIST | | | | |
| **테이블 설명** | | PRESET테이블에 기록된 프리셋의 내용을 기록하는 테이블 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | PRESET\_ID | NUMBER | Y | FK |  | REFERENCES PRESET(PRESET\_ID) |
| 2 | STOCK\_ID | NUMBER | Y | FK |  | REFERENCES STOCK(STOCK\_ID) |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **비 고** | | | | | | |
|  | | | | | | |

# 7. SQL문 사용하기







stock테이블 inserting, updating, deleting 시에 stock\_product, stock\_material, stock\_goods에 데이터를 추가, 삭제, 업데이트하는 트리거 전문입니다.

***-- 트리거 생성 stock에 제품이 insert, delete, update될 때 자동으로 stock\_product, stock\_material, stock\_goods에 stock\_id를 수정***

create or replace trigger stock\_div\_classification

after insert or delete or update on stock ***-- 생성, 삭제, 업데이트될때***

for each row

begin

if inserting then ***-- 추가할때***

if :new.stock\_division = 'product' then

insert into stock\_product

values(:new.stock\_id);

end if;

if :new.stock\_division = 'material' then

insert into stock\_material

values(:new.stock\_id);

end if;

if :new.stock\_division = 'goods' then

insert into stock\_goods

values(:new.stock\_id);

end if;

end if;

if updating then ***-- 업데이트할때***

if :new.stock\_division = 'product' then

if :old.stock\_division = 'material' then

delete from stock\_material where material\_id = :new.stock\_id;

insert into stock\_product values(:new.stock\_id);

elsif :old.stock\_division = 'goods' then

delete from stock\_goods where goods\_id = :new.stock\_id;

insert into stock\_product values(:new.stock\_id);

end if;

end if;

if :new.stock\_division = 'material' then

if :old.stock\_division = 'product' then

delete from stock\_product where product\_id = :new.stock\_id;

insert into stock\_material values(:new.stock\_id);

elsif :old.stock\_division = 'goods' then

delete from stock\_goods where goods\_id = :new.stock\_id;

insert into stock\_material values(:new.stock\_id);

end if;

end if;

if :new.stock\_division = 'goods' then

if :old.stock\_division = 'product' then

delete from stock\_product where product\_id = :new.stock\_id;

insert into stock\_goods values(:new.stock\_id);

elsif :old.stock\_division = 'material' then

delete from stock\_material where material\_id = :new.stock\_id;

insert into stock\_goods values(:new.stock\_id);

end if;

end if;

end if;

if deleting then ***-- 삭제할때***

if :old.stock\_division = 'product' then

delete from stock\_product where product\_id = :old.stock\_id;

elsif :old.stock\_division = 'material' then

delete from stock\_material where material\_id = :old.stock\_id;

elsif :old.stock\_division = 'goods' then

delete from stock\_goods where goods\_id = :old.stock\_id;

end if;

end if;

end;

/

***-- 트리거 끝!***

|  |
| --- |
| 별 첨 |

**별첨 1. 프로젝트 후기**

**별첨 2. Database 구축 자료**

**1. 프로젝트 후기**

프로젝트를 진행할수록 수정할 일이 점점 많아지는 걸 보고 초안 설계의 중요성을 다시한번 깨달았습니다. 논리명, 물리명도 정하지 않고 만들어가다 보니, 일관적이지 않은 물리명도 많이 생겨나서 나중에 다시 수정하느라 애를 먹었습니다. 다음부턴 초안을 보다 신중하고 세밀하게 작성해야겠다고 생각했습니다.

그와 별개로 프로젝트를 진행하면서 자신의 스킬이 많이 늘었다는 생각이 들었습니다. 생각했던 기능을 어떻게든 구현하고자 수없이 검색하고 시행착오를 겪으며 타협하고 넘어간 부분도 있지만 제대로 생각대로 작동하게 만들었을 때는 뿌듯함을 느꼈습니다.

다음 문단은 제가 힘들었던 부분을 토로하는 부분으로 굳이 읽지 않으셔도 상관없습니다.

개인적으로 가장 힘들었던 부분은 제조(product)테이블을 생성할 때였습니다. 제품 하나당 여러 개의 원자재가 들어가는 다중 값 속성이라, 제품 열에는 제품만 들어가게 해야 하고, 원자재 열에는 원자재만 들어가게 해야 하는데 처음엔 단순히 check로 가능하리라 생각했지만 check에 서브 쿼리를 넣을 수가 없었습니다. 그렇다고 외래키를 주려고 하니 이미 물품 테이블은 하나로 뭉쳐진 채로 미리 작업한 다른 테이블과 묶여 있어 손을 댈 수도 없었고, view를 이용하려고 하니 view를 외래키로 참조할 수 없다는 사실도 알게 되었습니다. 사실 굳이 제품인지 원자재인지 체크를 안 하고 넘어가는 방법도 있었지만 왠지 그냥 넘어가면 두고두고 더 귀찮아질 것 같았기에 힘껏 매달려서 해결했습니다. 해결할 방법은 여럿 있겠지만 결국 제가 해결한 방법은 trigger였습니다. 처음 사용해보아서 엄청난 오류와 싸워야 했지만 제대로 작동하게 만들고 나니 재미있었습니다.

**2. Database 구축 자료**

**<계정 생성>**

create user yeol identified ditdb20;

**<권한 부여>**

grant connect, resource to yeol;

**<테이블 생성>**

-- 계정 테이블 생성

create table account(

acc\_id varchar2(15) constraint tb\_account\_id\_pk primary key not null,

acc\_pw varchar2(20) not null,

acc\_name varchar2(20) not null,

acc\_position varchar2(20),

acc\_auth number default 9 not null

);

-- 메뉴 테이블 생성

create table menu(

menu\_id number constraint tb\_menu\_id\_pk primary key not null,

menu\_name varchar2(20) not null,

menu\_auth number default 0 not null

);

-- 물품 테이블 생성, 구분 제약조건 product, material, goods

create table stock(

stock\_id number constraint tb\_stock\_id\_pk primary key not null,

stock\_name varchar2(30) not null,

stock\_cost number default 0 not null,

stock\_division varchar2(10) not null,

stock\_saleprice number,

constraint division\_check check(stock\_division in('product', 'material', 'goods'))

);

-- division = product인 stock\_id만 저장되는 테이블

create table stock\_product(

product\_id number not null,

constraint stock\_product\_pk primary key(product\_id),

constraint stock\_product\_fk foreign key(product\_id) references stock(stock\_id)

);

-- division = material인 stock\_id만 저장되는 테이블

create table stock\_material(

material\_id number not null,

constraint stock\_material\_pk primary key(material\_id),

constraint stock\_material\_fk foreign key(material\_id) references stock(stock\_id)

);

-- division = goods인 stock\_id만 저장되는 테이블

create table stock\_goods(

goods\_id number not null,

constraint stock\_goods\_pk primary key(goods\_id),

constraint stock\_goods\_fk foreign key(goods\_id) references stock(stock\_id)

);

-- stock\_id 설정용 시퀀스

create sequence stock\_id\_seq

start with 1 increment by 1;

-- 시퀀스 초기화

drop sequence stock\_id\_seq;

-- 트리거 생성 stock에 제품이 insert, delete, update될 때 자동으로 stock\_product, stock\_material, stock\_goods에 stock\_id를 수정

create or replace trigger stock\_div\_classification

after insert or delete or update on stock -- 생성, 삭제, 업데이트될때

for each row

begin

if inserting then -- 추가할때

if :new.stock\_division = 'product' then

insert into stock\_product

values(:new.stock\_id);

end if;

if :new.stock\_division = 'material' then

insert into stock\_material

values(:new.stock\_id);

end if;

if :new.stock\_division = 'goods' then

insert into stock\_goods

values(:new.stock\_id);

end if;

end if;

if updating then -- 업데이트할때

if :new.stock\_division = 'product' then

if :old.stock\_division = 'material' then

delete from stock\_material where material\_id = :new.stock\_id;

insert into stock\_product values(:new.stock\_id);

elsif :old.stock\_division = 'goods' then

delete from stock\_goods where goods\_id = :new.stock\_id;

insert into stock\_product values(:new.stock\_id);

end if;

end if;

if :new.stock\_division = 'material' then

if :old.stock\_division = 'product' then

delete from stock\_product where product\_id = :new.stock\_id;

insert into stock\_material values(:new.stock\_id);

elsif :old.stock\_division = 'goods' then

delete from stock\_goods where goods\_id = :new.stock\_id;

insert into stock\_material values(:new.stock\_id);

end if;

end if;

if :new.stock\_division = 'goods' then

if :old.stock\_division = 'product' then

delete from stock\_product where product\_id = :new.stock\_id;

insert into stock\_goods values(:new.stock\_id);

elsif :old.stock\_division = 'material' then

delete from stock\_material where material\_id = :new.stock\_id;

insert into stock\_goods values(:new.stock\_id);

end if;

end if;

end if;

if deleting then -- 삭제할때

if :old.stock\_division = 'product' then

delete from stock\_product where product\_id = :old.stock\_id;

elsif :old.stock\_division = 'material' then

delete from stock\_material where material\_id = :old.stock\_id;

elsif :old.stock\_division = 'goods' then

delete from stock\_goods where goods\_id = :old.stock\_id;

end if;

end if;

end;

/

-- 트리거 끝!

create table product( -- 제조 테이블 생성, product테이블과 material 테이블에서 각각 참조

product\_id number not null,

material\_id number not null,

made\_quantity number default 1 not null,

constraint tb\_product\_product\_id\_fk

foreign key (product\_id) references stock\_product(product\_id),

constraint tb\_product\_material\_id\_fk

foreign key (material\_id) references stock\_material(material\_id)

);

create table supply( -- 입고 테이블 생성, stock 테이블과 manufacturer 테이블에서 각각 참조

stock\_id number not null,

mfr\_name varchar2(20) not null,

supply\_date date default sysdate,

quantity number not null,

constraint supply\_stock\_id\_fk

foreign key(stock\_id) references stock(stock\_id),

constraint supply\_mfr\_name\_fk

foreign key(mfr\_name) references manufacturer(mfr\_name)

);

create table manufacturer( -- 제조사 테이블 생성

mfr\_name varchar2(20) constraint manufacturer\_name\_pk primary key not null,

mfr\_tel varchar2(15),

mfr\_loc varchar2(40),

mfr\_manager varchar2(20)

);

create table release( -- 출고 테이블 생성, stock 테이블 참조

stock\_id number not null,

release\_quantity number not null,

release\_date date default sysdate not null,

constraint release\_stock\_id\_fk

foreign key(stock\_id) references stock(stock\_id)

);

-- preset\_id 설정용 시퀀스

create sequence preset\_id\_seq

start with 1 increment by 1;

-- 시퀀스 초기화

drop sequence preset\_id\_seq;

create table preset( -- 프리셋 테이블 생성

preset\_id number constraint preset\_id\_pk primary key not null,

preset\_name varchar2(20)

);

create table preset\_list( -- 프리셋 목록 테이블 생성, preset테이블과 stock 테이블 참조

preset\_id number not null,

stock\_id number not null,

constraint preset\_list\_preset\_id\_fk

foreign key(preset\_id) references preset(preset\_id),

constraint preset\_list\_stock\_id\_fk

foreign key(stock\_id) references stock(stock\_id)

);

-- 판매 뷰 생성. stock 테이블과 release 테이블 조인

create view vw\_sales as

select

st.stock\_id as "물품식별번호",

rl.release\_date as "판매일",

rl.release\_quantity as "판매수량",

(rl.release\_quantity \* st.stock\_saleprice) as "판매금액",

(rl.release\_quantity \* st.stock\_saleprice)

-(rl.release\_quantity \* st.stock\_cost)-(rl.release\_quantity \* nvl((

select sum(prd.made\_quantity \* stk.stock\_cost)

from stock stk, product prd

where stk.stock\_id = prd.material\_id

group by product\_id

having product\_id = st.stock\_id

), 0)) as "이익"

from stock st, release rl

where st.stock\_id = rl.stock\_id;

select \* from vw\_sales;

-- 재고 뷰 생성, stock, release, supply 테이블 조인 후 재고 view 생성, 이후 재고 view에 물품명과 구분을 붙여주는 view 추가

create view vw\_qtt as

select st.stock\_id, nvl(sum(sp.supply\_quantity),0) - nvl(sum(rl.release\_quantity),0) as "sumqtt"

from supply sp left outer join stock st

on sp.stock\_id = st.stock\_id

left outer join release rl

on st.stock\_id = rl.stock\_id

group by st.stock\_id;

create view vw\_inventory as

select qt.stock\_id, qt."sumqtt" as "재고량", st.stock\_name as "물품명", st.stock\_division as "구분", sp.mfr\_name as "제조사"

from stock st, vw\_qtt qt, supply sp

where st.stock\_id = qt.stock\_id and st.stock\_id = sp.stock\_id

order by stock\_id;

-- 액세스 뷰 생성, account, menu 테이블 조인

create view vw\_access as

select ac.acc\_id, mn.menu\_name

from account ac, menu mn

where ac.acc\_auth <= mn.menu\_auth;

-- 제조 뷰 생성

create view vw\_product as

select st.stock\_id as product\_id, st.stock\_name as "제품명", st2.stock\_id as material\_id, st2.stock\_name as "원재료명", pr.made\_quantity as "필요수량"

from stock st, stock st2, product pr

where pr.product\_id = st.stock\_id and pr.material\_id = st2.stock\_id

order by pr.product\_id, pr.material\_id desc;

-- 프리셋 뷰 생성

create view vw\_preset as

select prs.preset\_name, st.stock\_name

from preset prs, stock st, preset\_list prl

where prs.preset\_id = prl.preset\_id and prl.stock\_id = st.stock\_id;

**<데이터 입력>**

insert into stock

values(stock\_id\_seq.nextval, 'A4용지', 10, 'material', null);

insert into stock

values(stock\_id\_seq.nextval, 'A3용지', 12, 'material', null);

insert into stock

values(stock\_id\_seq.nextval, 'A2용지', 15, 'material', null);

insert into stock

values(stock\_id\_seq.nextval, 'A5용지', 8, 'material', null);

insert into stock

values(stock\_id\_seq.nextval, 'B4용지', 14, 'material', null);

insert into stock

values(stock\_id\_seq.nextval, 'B5용지', 11, 'material', null);

insert into stock

values(stock\_id\_seq.nextval, '와이어링', 100, 'material', null);

insert into stock

values(stock\_id\_seq.nextval, 'pvc커버', 150, 'material', null);

insert into stock

values(stock\_id\_seq.nextval, 'A5무선공책', 0, 'product', 2000);

insert into stock

values(stock\_id\_seq.nextval, 'B5스프링 공책', 0, 'product', 3000);

insert into stock

values(stock\_id\_seq.nextval, '제도샤프', 760, 'goods', 1000);

insert into stock

values(stock\_id\_seq.nextval, '지우개', 170, 'goods', 300);

insert into manufacturer

values('호남제지', '010-8788-6569', '부산 강서구', 'Cave Johnson');

insert into manufacturer

values('동의문구', '010-1234-1234', '부산 강서구', '김동의');

insert into manufacturer

values('ERD Chemical', '010-3322-2134', '울산 북구', '한지우');

insert into supply(stock\_id, mfr\_name, quantity)

values(1, '호남제지', 3000);

insert into supply(stock\_id, mfr\_name, quantity)

values(2, '호남제지', 2000);

insert into supply(stock\_id, mfr\_name, quantity)

values(3, '호남제지', 3000);

insert into supply(stock\_id, mfr\_name, quantity)

values(4, '호남제지', 2500);

insert into supply(stock\_id, mfr\_name, quantity)

values(5, '호남제지', 1500);

insert into supply(stock\_id, mfr\_name, quantity)

values(6, '호남제지', 3000);

insert into supply(stock\_id, mfr\_name, quantity)

values(7, '동의문구', 200);

insert into supply(stock\_id, mfr\_name, quantity)

values(8, '동의문구', 200);

insert into supply(stock\_id, mfr\_name, quantity)

values(11, '동의문구', 500);

insert into supply(stock\_id, mfr\_name, quantity)

values(12, 'ERD Chemical', 200);

insert into release(stock\_id, release\_quantity)

values(11, 13);

insert into release(stock\_id, release\_quantity)

values(11, 5);

insert into release(stock\_id, release\_quantity)

values(11, 9);

insert into release(stock\_id, release\_quantity)

values(12, 8);

insert into release(stock\_id, release\_quantity)

values(12, 22);

insert into release(stock\_id, release\_quantity)

values(12, 8);

insert into release(stock\_id, release\_quantity)

values(10, 3);

insert into release(stock\_id, release\_quantity)

values(10, 5);

insert into release(stock\_id, release\_quantity)

values(9, 22);

insert into product

values(9, 4, 20);

insert into product

values(9, 7, 1);

insert into product

values(9, 8, 1);

insert into product

values(10, 6, 40);

insert into product

values(10, 7, 1);

insert into product

values(10, 8, 1);

insert into preset

values(preset\_id\_seq.nextval, '스프링노트+샤프 세트');

insert into preset\_list

values(2, 10);

insert into preset\_list

values(2, 11);

insert into account

values('admin', '0000', '관리자', '관리자', 0);

insert into account

values('manager', '0000', '홍길동', '매니저', 4);

insert into account

values('testid2', '1234', 'testname', '테스터', 8);

insert into menu

values(1, '사용자 권한 설정', 0);

insert into menu

values(2, '메뉴 권한 관리', 9);

insert into menu

values(3, '입출고 등록', 9);

insert into menu

values(4, '내역 조회', 4);

insert into menu

values(5, '제품 관리', 9);

insert into menu

values(6, '프리셋', 4);

insert into menu

values(7, '매출 조회', 9);

insert into menu

values(8, '재고 분석', 4);