

[2023-02학기 데이터베이스과제보고서]

제목: (예시) 조선대 컴퓨터공학과 수강시스템

\* 이 주제는 안됨

과목:데이터베이스

분반: 0x분반

담당교수: 정현숙

제출자: xxxxxxxx

학번:xxxxxxxxxxxx

제출일자:20XX.XX.XX

학과:xxxxxxxxxxxx

## 순서

1. 개발 동기
2. 개발 설계 목적
3. 개발환경
  - 개발환경, 개발일정
4. 요구사항 분석
5. 개념적설계 -데이터모델링 (7단계)- ERD 도출
6. 논리적 설계
  - ERWin - ERD작성
7. 물리적 설계
  - 1) 스키마
  - 2) 데이터베이스 상태
  - 3) 정규화
    - 제1정규화
    - 제2정규화
    - 제3정규화
  - 4) 데이터베이스 테이블 작성
8. 구현 및 데모
  - 자바로 구현
9. 소감 및 느낀점
10. 참고문헌
  - 사이트
  - 책

\*보고서는 ERWIN소스, 정규화(엑셀소스), 보고서, source(oracledb , java source,\*.sql, \*.txt(sql설명,실행한 것) , 본인사진등) 작성해서 제출해 주시기 바랍니다. 단, 주어진 ERD에 맞도록 테이블이 도출되어야 하며, 각 테이블에 레코드는 최소 의미있는 값으로 10개이상 입력해 주셔야 하며, 데모에서 삽입, 삭제, 추가, 조회 등이 가능하도록 준비해 주시면 됩니다.

### 1. 개발 동기

// 간단히 조선대 IT융합대학 수강신청시스템 구축에 관한 내용 , 아래 ERD 보고 작성

### 2. 개발 설계 목적

// 간단히 조선대 IT융합대학 수강신청시스템 구축에 관한 내용 , 아래 ERD 보고 작성

### 3. 개발환경

- 개발환경

(예시 참고)

---

#### 시스템 프로그램 환경

장치 이름	:	DESKTOP-RKRUMQE (데스크탑)
프로세서	:	AMD Ryzen 7 5800X 8-Core Processor 3.80 GHz
설치된 RAM	:	24.0GB
윈도우 에디션	:	Windows 11 Home
윈도우 버전	:	22H2
설치 날짜	:	2023-07-15
OS 빌드	:	22621.2134 (Windows Feature Experience Pack 1000.22659.1000.0)
시스템 종류	:	64비트 운영 체제, x64 기반 프로세서
펜 및 터치	:	이 디스플레이에 사용할 수 있는 펜 또는 터치식 입력이 없습니다.
과제 수행	:	집

#### 응용 프로그램 환경

1) Oracle Database	:	11g Express Edition Release 11.2.0.2.0 - 64bit Production
2) Oracle JDBC driver	:	ojdbc6
3) Eclipse IDE	:	Eclipse Version: 3.19.200.v20230903-I000
4) Java JDK version	:	21

---

- 개발일정

#### 4. 요구사항 분석

// 간단히 작성

COLLEGE -DEAN-INSTRUCOR: 단과대학- 학장이관리한다-교수

INSTRUCOR- TEACHES- SECTION:교수- 가르킨다-섹션

COURSE-SECS- SECTION: 코스 -섹션과 연관되어 있다.-섹션

COURSE- OFFERS-DEPT: 코스-제공한다-학과

DEPT- ADMIN-COLLEGE:학과-관리한다-단과대학

DEPT-CHAIR-INSTUCTOR : 학과 - 강좌 개설한다 -교수

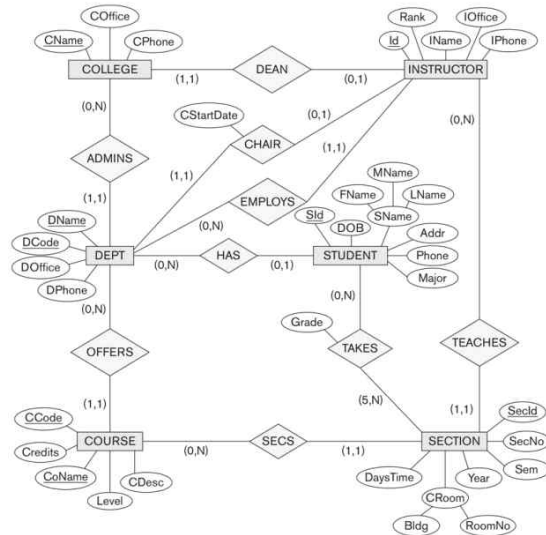
DEPT-EMPLOYS- INSTUCTOR: 학과 - 소속해서일한다. - 교수

DEPT-HAS-STUDENT: 학과 -가지고 있다- 학생

STUDENT-TAKES-SECTION: 학생 -가진다 -섹션

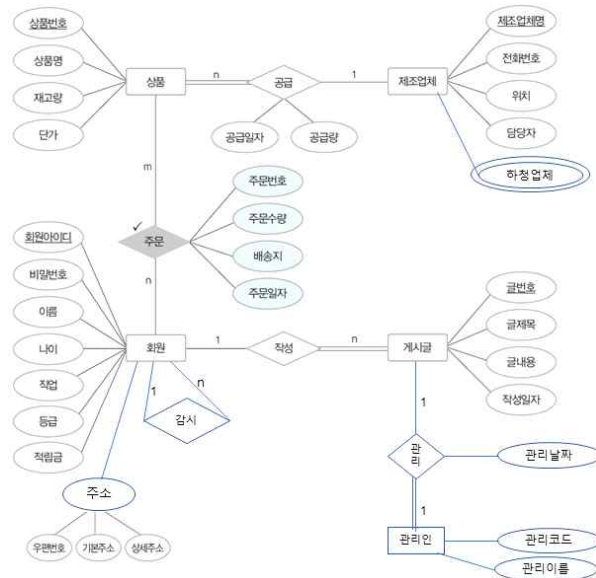
## 5. 개념적설계 - ERD 도출

다음 E-R 다이어그램을 보고 릴레이션 스키마 변환 규칙을 이용해서 논리적 설계와 물리적설계를 완성 하세요.



COLLEGE -DEAN-INSTRUCOR: 단과대학- 학장이관리한다-교수  
 INSTRUCOR- TEACHES- SECTION:교수- 가르킨다-섹션  
 COURSE-SECS- SECTION: 코스 -섹션과 연관되어 있다.-섹션  
 COURSE- OFFERS-DEPT: 코스-제공한다-학과  
 DEPT- ADMIN-COLLEGE:학과-관리한다-단과대학  
 DEPT-CHAIR-INSTUCTOR : 학과 - 강좌 개설한다 -교수  
 DEPT-EMPLOY- INSTUCTOR: 학과 - 소속해서일한다. - 교수  
 DEPT-HAS-STUDENT: 학과 -가지고 있다- 학생  
 STUDENT-TAKES-SECTION: 학생 -가진다 -섹션

(과제3참고 -예시) ---다음 E-R 다이어그램을 보고 릴레이션 스키마 변환 규칙을 이용해서 논리적 설계와 물리적설계를 완성하세요.



1-1. 모든 개체는 릴레이션으로 변환된다.

설명 :

현재 그림에서 보이는 개체는 상품, 제조업체, 회원, 게시글, 관리인이 있다 이 개체들을 릴레이션으로 변환한다.

그림에서는 관리인의 기본 키가 없는 것처럼 나온다.

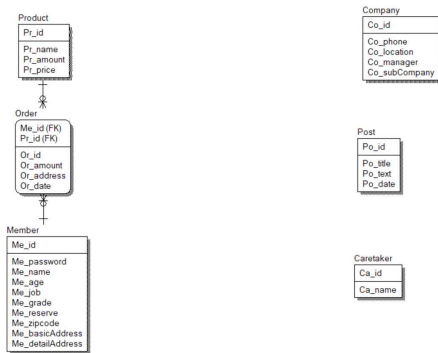
하지만, 이것은 튜플의 유일성을 해치는 것이기에 관리 코드를 기본 키로 지정하여 릴레이션을 형성한다.



1-2. m : n 관계(다대다 관계)는 관계 릴레이션으로 변환한다.

설명 :

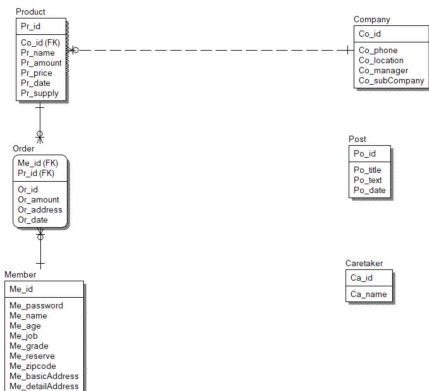
그림에서 m : n 관계는 주문이 있다. 이것을 관계 릴레이션으로 변환하여 상품번호와 회원아이디를 참조하는 외래키를 형성한 후 기본 키를 지정하고, 나머지 관계 속성을 추가해준다. 그런 후, 관계 식별 여부를 설정하는데, 현재 회원과 상품 사이에 주문이 생기려면, 회원이 존재하며, 상품이 존재해야 발생한다. 그렇기에 식별 상태로 설정하고, 회원은 무조건 상품을 주문할 필요가 없고, 상품도 무조건 주문할 필요가 없으므로, 그림에서 나온 것과 같이 관계 참여를 선택적 참여로 설정한다.



### I-3. 일대다 관계는 외래키로 표현한다.

설명 :

현재 그림에서 보이는 일대다 관계는 공급, 작성이 있다. 공급은 1 : n 관계인데, 상품은 필수적 참여로 상품은 무조건 제조업체를 가져야 한다는 의미이다. 또한, 하나의 제조업체는 상품을 여러 개 가질 수 있으며, 선택적 참여로 상품이 없는 제조회사가 존재할 수 있다. 그러므로, Company의 기본 키를 참조하는 외래키에 Not null 제약 사항을 추가하고, 참조 무결성 제약 조건을 추가하여준다. 또한, 관계 생성 시 Company의 공급 관계에 존재하는 속성들을 Product의 속성으로 넘겨준다 (n 쪽에 줘야 관계마다 관계 속성들을 가질 수 있다).



설명 :

작성도 1 : n 관계인데, 회원은 선택적 참여이며 게시글을 여러 개 작성할 수도 있고 하나도 작성을 안 할 수 있다. 게시글은 무조건 회원이 작성해야 하며, 스스로 작성이 안 된다(필수적 참여). 그러므로 위에 공급과 비슷하게 Member를 참조하는 외래키를 Post 속성에 넣어주고, 이 속성을 not null 제약 사항을 추가하며 참조 무결성 제약 조건도 추가해준다(비식별 관계로 설정했기 때문).

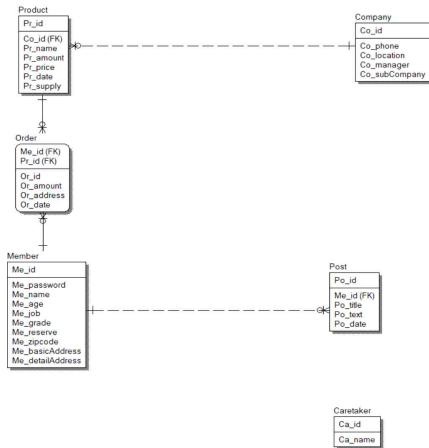
### I-4. 일대일 관계는 외래키로 변환한다.

설명 :

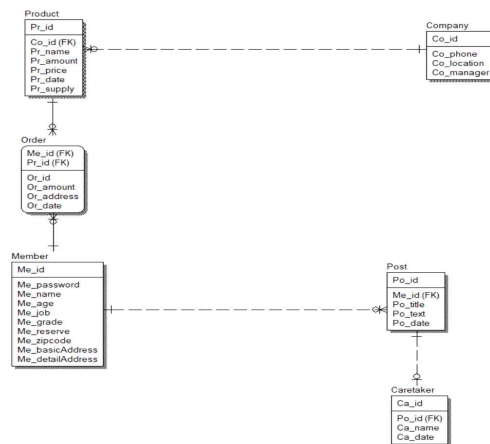
현재 그림에서 1 : 1 관계는 관리 관계가 있다. 게시글은 관리인이 하나씩 담당하며 게시글을 관리하는 관리인이 없을 수도 있다.

관리인은 게시글을 무조건 하나씩 담당한다(필수적 참여). 그러므로 게시글의 기본 키를 참조하는 외래키

를 관리인의 속성에 넣어주며, 관리 관계가 맺고 있는 속성은 관리인에게 넘겨준다. 이것도 비식별 관계



로 설정했으므로 외래키의 not null 제약을 추가해주고 참조 무결성 제약 조건을 추가해준다.



#### I-5. 다중값 속성은 릴레이션으로 변환한다.

설명 :

현재 그림에서 다중 값 속성은 하청 업체(Co\_subCompany)가 있다. 속성의 원자성을 위배하므로 릴레이션을 생성하여 Company의 기본 키를 참조하는 외래키를 기본 키로 설정하고, 하청 업체도 기본키 속성으로 설정해준다. 이러한 하청 업체는 한 회사가 여러 개 가질 수 있고, 하나도 안가질 수도 있으므로, 1 : N 관계로 설정해준다.

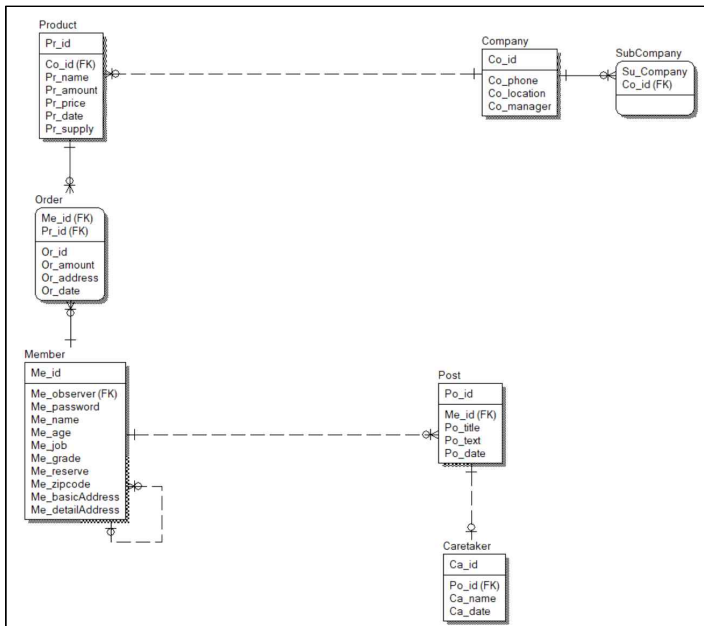
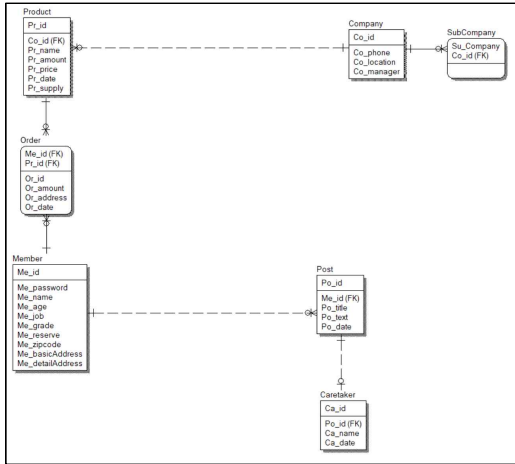
#### I-6. 기타 고려사항

설명 :

감시는 순환 관계이며 1 : n 관계를 맺고 있다. 예를 들어 하나의 상사가 여러 명의 부하 직원을 감시할 수 있는 것이다.

상사는 감시를 여러 명 할 수도 있고 아예 안 할 수 있다. 또한, 부하 직원은 감시를 당할 수도 있으며 감시를 안 당하는 상태일 수도 있다. (비식별 관계) 그러므로, Me\_id를 참조하는 Me\_observer 외래 키 속성을 추가해준다.





최종 릴레이션 스키마의 테이블 명세서 (이해하기 쉽게 표로 표현했습니다. - 색칠된 것은 외래키를 나타내고 있습니다.)- 7개

SubCompany	
<u>Su_company</u>	Co_id

Company			
<u>Co_id</u>	Co_phone	Co_location	Co_manager

Product						
<u>Pr_id</u>	Co_id	Pr_name	Pr_amount	Pr_price	Pr_date	Pr_supplier

Orders					
<u>Me_id</u>	<u>Pr_id</u>	Or_id	Or_amount	Or_address	Or_date

Member										
<u>Me_id</u>	<u>Me_observer</u>	Me_password	Me_name	Me_age	Me_job	Me_grade	Me_reserve	Me_zipcode	Me_basicAddress	Me_contact

Post				
<u>Po_id</u>	<u>Me_id</u>	Po_title	Po_text	Po_date

Caretaker			
<u>Ca_id</u>	<u>Po_id</u>	Ca_name	Ca_date

#### 물리적 설계

테이블 이름		SubCompany				
속성이름	데이터타입	널 허용 여부	기본값	기본키	외래키	제약조건
Su_company	VARCHAR2(20)	N		PK		
Co_id	VARCHAR2(10)	N		PK	FK	

테이블 이름		Company				
속성이름	데이터타입	널 허용 여부	기본값	기본키	외래키	제약조건
Co_id	VARCHAR2(10)	N		PK		
Co_phone	VARCHAR2(20)	Y				
Co_location	VARCHAR2(30)	Y				
Co_manager	VARCHAR2(20)	Y				

테이블 이름		Order				
속성이름	데이터타입	널 허용 여부	기본값	기본키	외래키	제약조건
Me_id	VARCHAR2(10)	N		PK	FK	
Pr_id	VARCHAR2(10)	N		PK	FK	
Or_id	VARCHAR2(10)	N				
Or_amount	INT	N				Or_amount > 0
Or_address	VARCHAR2(30)	N				
Or_date	DATE	Y				

테이블 이름		Product				
속성이름	데이터타입	널 허용 여부	기본값	기본키	외래키	제약조건
Pr_id	VARCHAR2(10)	N		PK		
Co_id	VARCHAR2(10)	N			FK	
Pr_name	VARCHAR2(20)	Y				
Pr_amount	INT	Y	0			Pr_amount >= 0
Pr_price	INT	Y				Pr_price >= 0
Pr_date	DATE	Y				
Pr_supply	INT	Y	0			Pr_supply >= 0

테이블 이름		Member				
속성이름	데이터타입	널 허용 여부	기본값	기본키	외래키	제약조건
Me_id	VARCHAR2(10)	N		PK		
Me_observer	VARCHAR2(10)	Y			FK	
Me_password	VARCHAR2(20)	N				
Me_name	VARCHAR2(20)	Y				
Me_age	INT	Y				Me_age >= 0
Me_job	VARCHAR2(20)	Y				
Me_grade	VARCHAR2(20)	Y				
Me_reserve	INT	Y	0			Me_reserve >= 0
Me_zipcode	INT	Y				
Me_basicAddress	VARCHAR2(30)	Y				
Me_detailAddress	VARCHAR2(30)	Y				

테이블 이름		Post				
속성이름	데이터타입	널 허용 여부	기본값	기본키	외래키	제약조건
Po_id	VARCHAR2(10)	N		PK		
Me_id	VARCHAR2(10)	N			FK	
Po_title	VARCHAR2(20)	Y				
Po_text	VARCHAR2(30)	Y				

Po_date	DATE	Y				
---------	------	---	--	--	--	--

테이블 이름		Caretaker				
속성이름	테이터타입	널 허용 여부	기본값	기본키	외래키	제약조건
Ca_id	VARCHAR2(10)	N		PK		
Po_id	VARCHAR2(10)	N			FK	
Ca_name	VARCHAR2(20)	Y				
Ca_date	DATE	Y				

#### SubCompany ( 하청업체 )

Su\_Company = 하청업체  
Co\_id = 제조업체명

#### Company ( 제조업체 )

Co\_id = 제조업체명  
Co\_phone = 전화번호  
Co\_location = 위치  
Co\_manager = 담당자

#### Product ( 상품 )

Pr\_id = 상품번호  
Co\_id = 제조업체명  
Pr\_name = 상품명  
Pr\_amount = 재고량  
Pr\_price = 단가  
Pr\_date = 공급일자  
Pr\_supply = 공급량

#### Order ( 주문 )

Me\_id = 회원아이디  
Pr\_id = 상품번호  
Or\_id = 주문번호  
Or\_amount = 주문수량  
Or\_address = 배송지  
Or\_date = 주문일자

#### Member ( 회원 )

Me_id	=	회원아이디
Me_observer	=	상사아이디
Me_password	=	비밀번호
Me_name	=	이름
Me_age	=	나이
Me_job	=	직업
Me_grade	=	등급
Me_reserve	=	적립금
Me_zipcode	=	우편번호
Me_basicAddress	=	기본주소
Me_detailAddress	=	상세주소

#### Post ( 게시글 )

Po_id	=	글번호
Me_id	=	회원아이디
Po_title	=	글제목
Po_text	=	글내용
Po_date	=	작성일자

#### Caretaker ( 관리인 )

Ca_id	=	관리코드
Po_id	=	글번호
Ca_name	=	관리이름
Ca_date	=	관리날짜

#### 테이블 작성 단계

##### 제조업체

```
CREATE TABLE 제조업체 (
    제조업체명 VARCHAR2(10) NOT NULL,
    전화번호 VARCHAR2(20),
    위치 VARCHAR2(30),
    담당자 VARCHAR2(20),
    PRIMARY KEY(제조업체명)
);
```

```
CREATE TABLE Company (
    Co_id VARCHAR2(10) NOT NULL,
    Co_phone VARCHAR2(20),
    Co_location VARCHAR2(30),
    Co_manager VARCHAR2(20),
    PRIMARY KEY(Co_id)
```

);

#### 회원

```
CREATE TABLE 회원 (  
    회원아이디 VARCHAR2(10) NOT NULL,  
    상사아이디 VARCHAR2(10),  
    비밀번호 VARCHAR2(20) NOT NULL,  
    이름 VARCHAR2(20),  
    나이 INT,  
    직업 VARCHAR2(20),  
    등급 VARCHAR2(20),  
    적립금 INT DEFAULT 0,  
    우편번호 INT,  
    기본주소 VARCHAR2(30),  
    상세주소 VARCHAR2(30),  
    PRIMARY KEY(회원아이디),  
    FOREIGN KEY(상사아이디) REFERENCES 회원(회원아이디),  
    CHECK (나이 >= 0),  
    CHECK (적립금 >= 0)  
);
```

```
CREATE TABLE Member (  
    Me_id VARCHAR2(10) NOT NULL,  
    Me_observer VARCHAR2(10),  
    Me_password VARCHAR2(20) NOT NULL,  
    Me_name VARCHAR2(20),  
    Me_age INT,  
    Me_job VARCHAR2(20),  
    Me_grade VARCHAR2(20),  
    Me_reserve INT DEFAULT 0,  
    Me_zipcode INT,  
    Me_basicAddress VARCHAR2(30),  
    Me_detailAddress VARCHAR2(30),  
    PRIMARY KEY(Me_id),  
    FOREIGN KEY(Me_observer) REFERENCES Member(Me_id),  
    CHECK (Me_age >= 0),  
    CHECK (Me_reserve >= 0)  
);
```

#### 하청업체

```
CREATE TABLE 하청업체 (  
    하청업체 VARCHAR2(20) NOT NULL,  
    제조업체명 VARCHAR2(10) NOT NULL,  
    PRIMARY KEY(하청업체, 제조업체명),  
    FOREIGN KEY(제조업체명) REFERENCES 제조업체(제조업체명)  
);
```

```
CREATE TABLE SubCompany (  
    Su_Company VARCHAR2(20) NOT NULL,  
    Co_id VARCHAR2(10) NOT NULL,  
    PRIMARY KEY(Su_Company, Co_id),  
    FOREIGN KEY(Co_id) REFERENCES Company(Co_id)  
);
```

#### 상품

```
CREATE TABLE 상품 (  
    상품번호 VARCHAR2(10) NOT NULL,  
    제조업체명 VARCHAR2(10) NOT NULL,  
    상품명 VARCHAR2(20),  
    재고량 INT DEFAULT 0,  
    단가 INT,  
    공급일자 DATE,  
    공급량 INT DEFAULT 0,  
    PRIMARY KEY(상품번호),  
    FOREIGN KEY(제조업체명) REFERENCES 제조업체(제조업체명),  
    CHECK (재고량 >= 0),  
    CHECK (단가 >= 0),  
    CHECK (공급량 >= 0)  
);
```

```
CREATE TABLE Product (  
    Pr_id VARCHAR2(10) NOT NULL,  
    Co_id VARCHAR2(10) NOT NULL,  
    Pr_name VARCHAR2(20),  
    Pr_amount INT DEFAULT 0,  
    Pr_price INT,  
    Pr_date DATE,  
    Pr_supply INT DEFAULT 0,  
    PRIMARY KEY(Pr_id),  
    FOREIGN KEY(Co_id) REFERENCES Company(Co_id),  
    CHECK (Pr_amount >= 0),
```

```

CHECK (Pr_price >= 0),
CHECK (Pr_supply >= 0)
);

```

#### 주문

```

CREATE TABLE 주문 (
    회원아이디 VARCHAR2(10) NOT NULL,
    상품번호 VARCHAR2(10) NOT NULL,
    주문번호 VARCHAR2(10) NOT NULL,
    주문수량 INT NOT NULL,
    배송지 VARCHAR2(30) NOT NULL,
    주문일자 DATE,
    PRIMARY KEY(회원아이디, 상품번호),
    FOREIGN KEY(회원아이디) REFERENCES 회원(회원아이디),
    FOREIGN KEY(상품번호) REFERENCES 상품(상품번호),
    CHECK (주문수량 > 0)
);

```

[ SQL에서 Order 이름으로 테이블 생성이 되지 않아 Orders 로 수정했습니다. ]

```

CREATE TABLE Orders (
    Me_id VARCHAR2(10) NOT NULL,
    Pr_id VARCHAR2(10) NOT NULL,
    Or_id VARCHAR2(10) NOT NULL,
    Or_amount INT NOT NULL,
    Or_address VARCHAR2(30) NOT NULL,
    Or_date DATE,
    PRIMARY KEY(Me_id, Pr_id),
    FOREIGN KEY(Me_id) REFERENCES Member(Me_id),
    FOREIGN KEY(Pr_id) REFERENCES Product(Pr_id),
    CHECK (Or_amount > 0)
);

```



);

#### 게시글

```
CREATE TABLE 게시글 (  
    글번호 VARCHAR2(10) NOT NULL,  
    회원아이디 VARCHAR2(10) NOT NULL,  
    글제목 VARCHAR2(20),  
    글내용 VARCHAR2(30),  
    작성일자 DATE,  
    PRIMARY KEY(글번호),  
    FOREIGN KEY(회원아이디) REFERENCES 회원(회원아이디)  
);
```

```
CREATE TABLE Post (  
    Po_id VARCHAR2(10) NOT NULL,  
    Me_id VARCHAR2(10) NOT NULL,  
    Po_title VARCHAR2(20),  
    Po_text VARCHAR2(30),  
    Po_date DATE,  
    PRIMARY KEY(Po_id),  
    FOREIGN KEY(Me_id) REFERENCES Member(Me_id)  
);
```

#### 관리인

```
CREATE TABLE 관리인 (  
    관리코드 VARCHAR2(10) NOT NULL,  
    글번호 VARCHAR2(10) NOT NULL,  
    관리이름 VARCHAR2(20),  
    관리날짜 DATE,  
    PRIMARY KEY(관리코드),  
    FOREIGN KEY(글번호) REFERENCES 게시글(글번호)  
);
```

```
CREATE TABLE Caretaker (  
    Ca_id VARCHAR2(10) NOT NULL,  
    Po_id VARCHAR2(10) NOT NULL,  
    Ca_name VARCHAR2(20),  
    Ca_date DATE,  
    PRIMARY KEY(Ca_id),  
    FOREIGN KEY(Po_id) REFERENCES Post(Po_id)
```

);

## 문제2. 오라클DBMS + JAVA GUI 구현 (JDBC 연결)

### I. 자바 설치

1) oracle 자바 설치 사이트 접속

<https://www.oracle.com/kr/java/technologies/downloads/#jdk21-windows>

JDK 21 JDK 17 GraalVM for JDK 21 GraalVM for JDK 17

#### JDK Development Kit 21.0.1 downloads

JDK 21 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions](#) (NFTC).

JDK 21 will receive updates under the NFTC, until September 2026, a year after the release of the next LTS. Subsequent JDK 21 updates will be licensed under the [Java SE OTN License](#) (OTN) and production use beyond the [limited free grants](#) of the OTN license will [require a fee](#).

Linux macOS Windows

Product/file description	File size	Download
x64 Compressed Archive	185.39 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip</a> (sha256)
x64 Installer	163.82 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe</a> (sha256)
x64 MSI Installer	162.60 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi</a> (sha256)

### 2) 프로그램 실행 후 설치 진행

jdk-21_windows-x64_bin.exe	2023-11-05 오후 2:51	응용 프로그램	167,753KB
----------------------------	--------------------	---------	-----------

### 3) 설치 경로를 입력하여 설치 확인

내 PC > 로컬 디스크 (C:) > Program Files > Java > jdk-21 >			
이름	수정된 날짜	유형	크기
bin	2023-11-05 오후 2:51	파일 폴더	
conf	2023-11-05 오후 2:51	파일 폴더	
include	2023-11-05 오후 2:51	파일 폴더	
jmods	2023-11-05 오후 2:51	파일 폴더	
legal	2023-11-05 오후 2:51	파일 폴더	
lib	2023-11-05 오후 2:51	파일 폴더	
LICENSE	2023-11-05 오후 2:51	파일	7KB
README	2023-11-05 오후 2:51	파일	1KB
release	2023-11-05 오후 2:51	파일	2KB

## 2. DBMS 설치

1) DBMS: Oracle Database 11g Express Edition

이미 설치 되어 있으므로, 패스한다.

## 3. JDBC 드라이버

1) C:\oracle\app\oracle\product\11.2.0\server\jdbc\lib에서 ojdbc6 파일 복사



2) C:\Program Files\Java\jdk-21\lib 에 붙여넣기

( 자바 JDK 21 버전은 jre 폴더가 일반적으로 설치되지 않음, 그러므로 lib 파일에 저장한 다음 eclipse에서 추가해줌. )



#### 4. 오라클 DBMS

( 문제 1번의 회원, 제조업체, 상품, 게시글 테이블 4개와 각 테이블에 데이터 5개 삽입 )

[ 사용자 생성, 권한 부여 및 접속 ]

create user assignment3 identified by 1234

default tablespace users

temporary tablespace temp

quota unlimited on users;

grant connect, resource, create view to assignment3;

connect assignment3/1234;

[ 회원, 제조업체, 상품, 게시글 테이블 생성 ]

[ 회원 ]

CREATE TABLE 회원 (

회원아이디 VARCHAR2(10) NOT NULL,

상사아이디 VARCHAR2(10),

비밀번호 VARCHAR2(20) NOT NULL,

```

이름 VARCHAR2(20),
나이 INT,
직업 VARCHAR2(20),
등급 VARCHAR2(20),
적립금 INT DEFAULT 0,
우편번호 INT,
기본주소 VARCHAR2(30),
상세주소 VARCHAR2(30),
PRIMARY KEY(회원아이디),
FOREIGN KEY(상사아이디) REFERENCES 회원(회원아이디),
CHECK (나이 >= 0),
CHECK (적립금 >= 0)
);

```

```

CREATE TABLE Member (
    Me_id VARCHAR2(10) NOT NULL,
    Me_observer VARCHAR2(10),
    Me_password VARCHAR2(20) NOT NULL,
    Me_name VARCHAR2(20),
    Me_age INT,
    Me_job VARCHAR2(20),
    Me_grade VARCHAR2(20),
    Me_reserve INT DEFAULT 0,
    Me_zipcode INT,
    Me_basicAddress VARCHAR2(30),
    Me_detailAddress VARCHAR2(30),
    PRIMARY KEY(Me_id),
    FOREIGN KEY(Me_observer) REFERENCES Member(Me_id),
    CHECK (Me_age >= 0),
    CHECK (Me_reserve >= 0)
);

```

#### [ 제조업체 ]

```

CREATE TABLE 제조업체 (
    제조업체명 VARCHAR2(10) NOT NULL,
    전화번호 VARCHAR2(20),
    위치 VARCHAR2(30),
    담당자 VARCHAR2(20),
    PRIMARY KEY(제조업체명)
);

```

);

```
CREATE TABLE Company (  
    Co_id VARCHAR2(10) NOT NULL,  
    Co_phone VARCHAR2(20),  
    Co_location VARCHAR2(30),  
    Co_manager VARCHAR2(20),  
    PRIMARY KEY(Co_id)  
);
```

### [ 상품 ]

```
CREATE TABLE 상품 (  
    상품번호 VARCHAR2(10) NOT NULL,  
    제조업체명 VARCHAR2(10) NOT NULL,  
    상품명 VARCHAR2(20),  
    재고량 INT DEFAULT 0,  
    단가 INT,  
    공급일자 DATE,  
    공급량 INT DEFAULT 0,  
    PRIMARY KEY(상품번호),  
    FOREIGN KEY(제조업체명) REFERENCES 제조업체(제조업체명),  
    CHECK (재고량 >= 0),  
    CHECK (단가 >= 0),  
    CHECK (공급량 >= 0)  
);
```

```
CREATE TABLE Product (  
    Pr_id VARCHAR2(10) NOT NULL,  
    Co_id VARCHAR2(10) NOT NULL,  
    Pr_name VARCHAR2(20),  
    Pr_amount INT DEFAULT 0,  
    Pr_price INT,  
    Pr_date DATE,  
    Pr_supply INT DEFAULT 0,  
    PRIMARY KEY(Pr_id),  
    FOREIGN KEY(Co_id) REFERENCES Company(Co_id),  
    CHECK (Pr_amount >= 0),  
    CHECK (Pr_price >= 0),  
    CHECK (Pr_supply >= 0)  
);
```

### [ 게시글 ]

```

CREATE TABLE 게시글 (
    글번호 VARCHAR2(10) NOT NULL,
    회원아이디 VARCHAR2(10) NOT NULL,
    글제목 VARCHAR2(20),
    글내용 VARCHAR2(30),
    작성일자 DATE,
    PRIMARY KEY(글번호),
    FOREIGN KEY(회원아이디) REFERENCES 회원(회원아이디)
);

```

```

CREATE TABLE Post (
    Po_id VARCHAR2(10) NOT NULL,
    Me_id VARCHAR2(10) NOT NULL,
    Po_title VARCHAR2(20),
    Po_text VARCHAR2(30),
    Po_date DATE,
    PRIMARY KEY(Po_id),
    FOREIGN KEY(Me_id) REFERENCES Member(Me_id)
);

```

#### [ 회원 릴레이션 데이터 삽입 ]

```

INSERT INTO 회원 VALUES ('M001', null, 'I234', '배준영', 24, '연구생', 'gold', 0, I234, '광주광역시', '동구');

```

```

INSERT INTO 회원 VALUES ('M002', null, 'I234', '장지엽', 24, '알바생', 'vip', 0, I234, '광주광역시', '광산구');

```

```

INSERT INTO 회원 VALUES ('M003', null, 'I234', '서대철', 24, '학생', 'gold', 0, I234, '광주광역시', '북구');

```

```

INSERT INTO 회원 VALUES ('M004', null, 'I234', '윤경호', 24, '연구생', 'silver', 0, I234, '광주광역시', '서구');

```

```

INSERT INTO 회원 VALUES ('M005', null, 'I234', '방형기', 24, '학생', 'gold', 0, I234, '광주광역시', '서구');

```

```

INSERT INTO Member VALUES ('M001', null, 'I234', '배준영', 24, '연구생', 'gold', 0, I234, '

```

광주광역시', '동구');

INSERT INTO Member VALUES ('M002', null, 'I234', '장지엽', 24, '알바생', 'vip', 0, I234, '광주광역시', '광산구');

INSERT INTO Member VALUES ('M003', null, 'I234', '서대철', 24, '학생', 'gold', 0, I234, '광주광역시', '북구');

INSERT INTO Member VALUES ('M004', null, 'I234', '윤경호', 24, '연구생', 'silver', 0, I234, '광주광역시', '서구');

INSERT INTO Member VALUES ('M005', null, 'I234', '방형기', 24, '학생', 'gold', 0, I234, '광주광역시', '서구');

#### [ 제조업체 릴레이션 데이터 삽입 ]

INSERT INTO 제조업체 VALUES ('농심', '062-380-7900', '서울 동작구', '박준');

INSERT INTO 제조업체 VALUES ('빙그레', '080-022-0056', '서울 종로구', '전창원');

INSERT INTO 제조업체 VALUES ('삼양', '02-740-7114', '서울 성북구', '최낙현');

INSERT INTO 제조업체 VALUES ('롯데', 'I577-000I', '서울 송파구', '신동빈');

INSERT INTO 제조업체 VALUES ('오뚜기', '02-2010-0114', '서울 강남구', '황성만');

INSERT INTO Company VALUES ('농심', '062-380-7900', '서울 동작구', '박준');

INSERT INTO Company VALUES ('빙그레', '080-022-0056', '서울 종로구', '전창원');

INSERT INTO Company VALUES ('삼양', '02-740-7114', '서울 성북구', '최낙현');

INSERT INTO Company VALUES ('롯데', 'I577-000I', '서울 송파구', '신동빈');

INSERT INTO Company VALUES ('오뚜기', '02-2010-0114', '서울 강남구', '황성만');

#### [ 상품 릴레이션 데이터 삽입 ]

INSERT INTO 상품 VALUES ('P001', '농심', '신라면', 10000, 950, to\_date('23/10/10', 'yy/mm/dd'), 1000);

INSERT INTO 상품 VALUES ('P002', '빙그레', '바나나우유', 5000, 1700, to\_date('23/09/11', 'yy/mm/dd'), 800);

INSERT INTO 상품 VALUES ('P003', '삼양', '불닭볶음면', 30000, 1050, to\_date('23/11/01', 'yy/mm/dd'), 2000);

INSERT INTO 상품 VALUES ('P004', '롯데', '칸쵸', 2000, 950, to\_date('23/07/20', 'yy/mm/dd'), 500);

INSERT INTO 상품 VALUES ('P005', '오뚜기', '오뚜기밥', 8000, 840, to\_date('23/04/15', 'yy/mm/dd'), 1500);

INSERT INTO Product VALUES ('P001', '농심', '신라면', 10000, 950, to\_date('23/10/10', 'yy/mm/dd'), 1000);

INSERT INTO Product VALUES ('P002', '빙그레', '바나나우유', 5000, 1700, to\_date('23/09/11', 'yy/mm/dd'), 800);

INSERT INTO Product VALUES ('P003', '삼양', '불닭볶음면', 30000, 1050, to\_date('23/11/01', 'yy/mm/dd'), 2000);

```
INSERT INTO Product VALUES ('P004', '롯데', '칸초', 2000, 950, to_date('23/07/20',
'yy/mm/dd'), 500);
```

```
INSERT INTO Product VALUES ('P005', '오투기', '오투기밥', 8000, 840, to_date('23/04/15',
'yy/mm/dd'), 1500);
```

[ 게시글 릴레이션 데이터 삽입 ]

```
INSERT INTO 게시글 VALUES ('001', 'M001', '안녕하세요', '하이', to_date('23/11/10',
'yy/mm/dd'));
```

```
INSERT INTO 게시글 VALUES ('002', 'M002', 'sql', '어려워요', to_date('23/11/05',
'yy/mm/dd'));
```

```
INSERT INTO 게시글 VALUES ('003', 'M003', '과제제출', 'ch9_sql', to_date('23/10/20',
'yy/mm/dd'));
```

```
INSERT INTO 게시글 VALUES ('004', 'M004', '스페인어', 'Hola', to_date('23/09/10',
'yy/mm/dd'));
```

```
INSERT INTO 게시글 VALUES ('005', 'M005', '중국어', 'Ni hao', to_date('23/08/15',
'yy/mm/dd'));
```

```
INSERT INTO Post VALUES ('001', 'M001', '안녕하세요', '하이', to_date('23/11/10',
'yy/mm/dd'));
```

```
INSERT INTO Post VALUES ('002', 'M002', 'sql', '어려워요', to_date('23/11/05',
'yy/mm/dd'));
```

```
INSERT INTO Post VALUES ('003', 'M003', '과제제출', 'ch9_sql', to_date('23/10/20',
'yy/mm/dd'));
```

```
INSERT INTO Post VALUES ('004', 'M004', '스페인어', 'Hola', to_date('23/09/10',
'yy/mm/dd'));
```

```
INSERT INTO Post VALUES ('005', 'M005', '중국어', 'Ni hao', to_date('23/08/15',
'yy/mm/dd'));
```



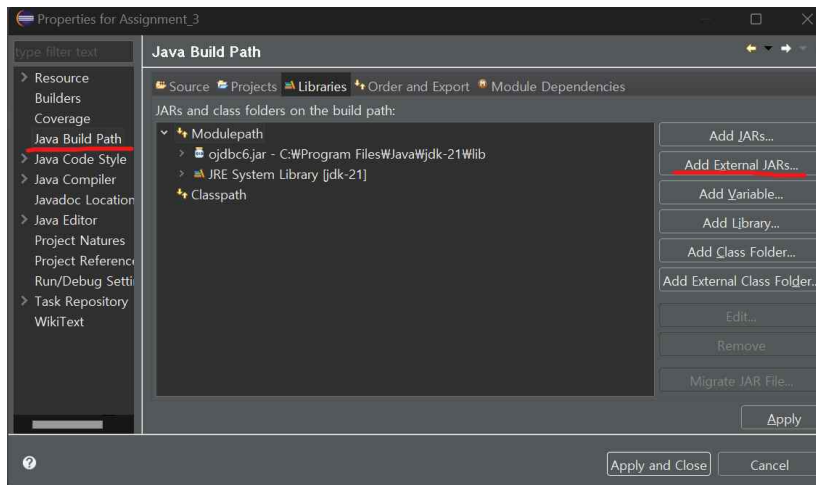
5. 이클립스 - 자바 프로젝트 생성 - 클래스 작성 - 연동 부분, 화면 설계 부분  
(여기는 데이터 예시이고 본인이 설계한 영화 또는 은행 테이블과 데이터가 삽입되어야 함 )

주의사항)

JDK 21를 설치했을 경우, 프로젝트 생성 후 JDBC를 따로 추가 해줘야 한다.

프로젝트 오른쪽 클릭 후 Properties 클릭 후 Java Build Path 클릭한다음 Add External JARs 클릭

그런다음, C:\Program Files\Java\jdk-21\lib 에 있는 ojdbc6.jar 추가



## I. 나머지 릴레이션 추가 및 데이터 삽입

### [ 하청업체 릴레이션 ]

```
CREATE TABLE 하청업체 (
    하청업체 VARCHAR2(20) NOT NULL,
    제조업체명 VARCHAR2(10) NOT NULL,
    PRIMARY KEY(하청업체, 제조업체명),
    FOREIGN KEY(제조업체명) REFERENCES 제조업체(제조업체명)
);
```

```
CREATE TABLE SubCompany (
    Su_Company VARCHAR2(20) NOT NULL,
    Co_id VARCHAR2(10) NOT NULL,
    PRIMARY KEY(Su_Company, Co_id),
    FOREIGN KEY(Co_id) REFERENCES Company(Co_id)
);
```

### [ 주문 릴레이션 ]

#### 주문

```
CREATE TABLE 주문 (
    회원아이디 VARCHAR2(10) NOT NULL,
    상품번호 VARCHAR2(10) NOT NULL,
    주문번호 VARCHAR2(10) NOT NULL,
    주문수량 INT NOT NULL,
    배송지 VARCHAR2(30) NOT NULL,
    주문일자 DATE,
    PRIMARY KEY(회원아이디, 상품번호),
    FOREIGN KEY(회원아이디) REFERENCES 회원(회원아이디),
    FOREIGN KEY(상품번호) REFERENCES 상품(상품번호),
);
```

```

        CHECK (주문수량 > 0)
    );

CREATE TABLE Orders (
    Me_id VARCHAR2(10) NOT NULL,
    Pr_id VARCHAR2(10) NOT NULL,
    Or_id VARCHAR2(10) NOT NULL,
    Or_amount INT NOT NULL,
    Or_address VARCHAR2(30) NOT NULL,
    Or_date DATE,
    PRIMARY KEY(Me_id, Pr_id),
    FOREIGN KEY(Me_id) REFERENCES Member(Me_id),
    FOREIGN KEY(Pr_id) REFERENCES Product(Pr_id),
    CHECK (Or_amount > 0)
);

```

#### [ 관리인 릴레이션 ]

##### 관리인

```

CREATE TABLE 관리인 (
    관리코드 VARCHAR2(10) NOT NULL,
    글번호 VARCHAR2(10) NOT NULL,
    관리이름 VARCHAR2(20),
    관리날짜 DATE,
    PRIMARY KEY(관리코드),
    FOREIGN KEY(글번호) REFERENCES 게시글(글번호)
);

```

```

CREATE TABLE Caretaker (
    Ca_id VARCHAR2(10) NOT NULL,
    Po_id VARCHAR2(10) NOT NULL,
    Ca_name VARCHAR2(20),
    Ca_date DATE,
    PRIMARY KEY(Ca_id),
    FOREIGN KEY(Po_id) REFERENCES Post(Po_id)
);

```

#### [ 하청업체 릴레이션 데이터 삽입 ]

```

INSERT INTO 하청업체 VALUES ('하청업체1', '농심');
INSERT INTO 하청업체 VALUES ('하청업체2', '빙그레');
INSERT INTO 하청업체 VALUES ('하청업체3', '삼양');
INSERT INTO 하청업체 VALUES ('하청업체4', '롯데');
INSERT INTO 하청업체 VALUES ('하청업체5', '오뚜기');

```

```

INSERT INTO SubCompany VALUES ('하청업체1', '농심');
INSERT INTO SubCompany VALUES ('하청업체2', '빙그레');
INSERT INTO SubCompany VALUES ('하청업체3', '삼양');
INSERT INTO SubCompany VALUES ('하청업체4', '롯데');
INSERT INTO SubCompany VALUES ('하청업체5', '오뚜기');

```

[ 주문 릴레이션 데이터 삽입 ]

```

INSERT INTO 주문 VALUES ('M001', 'P001', 'O001', 3, '서울시 강남구', TO_DATE('23/11/07',
'yy/mm/dd'));
INSERT INTO 주문 VALUES ('M002', 'P002', 'O002', 2, '서울시 송파구', TO_DATE('23/11/09',
'yy/mm/dd'));
INSERT INTO 주문 VALUES ('M003', 'P003', 'O003', 1, '서울시 종로구', TO_DATE('23/11/02',
'yy/mm/dd'));
INSERT INTO 주문 VALUES ('M004', 'P004', 'O004', 4, '서울시 마포구', TO_DATE('23/10/17',
'yy/mm/dd'));
INSERT INTO 주문 VALUES ('M005', 'P005', 'O005', 2, '서울시 강동구', TO_DATE('23/06/02',
'yy/mm/dd'));

```

```

INSERT INTO Orders VALUES ('M001', 'P001', 'O001', 3, '서울시 강남구', TO_DATE('23/11/07',
'yy/mm/dd'));
INSERT INTO Orders VALUES ('M002', 'P002', 'O002', 2, '서울시 송파구',
TO_DATE('23/11/09', 'yy/mm/dd'));
INSERT INTO Orders VALUES ('M003', 'P003', 'O003', 1, '서울시 종로구',
TO_DATE('23/11/02', 'yy/mm/dd'));
INSERT INTO Orders VALUES ('M004', 'P004', 'O004', 4, '서울시 마포구',
TO_DATE('23/10/17', 'yy/mm/dd'));
INSERT INTO Orders VALUES ('M005', 'P005', 'O005', 2, '서울시 강동구',
TO_DATE('23/06/02', 'yy/mm/dd'));

```

[ 관리인 릴레이션 데이터 삽입 ]

```

INSERT INTO 관리인 VALUES ('A001', '001', '관리인1', TO_DATE('23/11/07', 'yy/mm/dd'));
INSERT INTO 관리인 VALUES ('A002', '002', '관리인2', TO_DATE('23/10/10', 'yy/mm/dd'));
INSERT INTO 관리인 VALUES ('A003', '003', '관리인3', TO_DATE('23/11/11', 'yy/mm/dd'));
INSERT INTO 관리인 VALUES ('A004', '004', '관리인4', TO_DATE('23/04/02', 'yy/mm/dd'));
INSERT INTO 관리인 VALUES ('A005', '005', '관리인5', TO_DATE('23/06/25', 'yy/mm/dd'));

```

```

INSERT INTO Caretaker VALUES ('A001', '001', '관리인1', TO_DATE('23/11/07', 'yy/mm/dd'));
INSERT INTO Caretaker VALUES ('A002', '002', '관리인2', TO_DATE('23/10/10',
'yy/mm/dd'));
INSERT INTO Caretaker VALUES ('A003', '003', '관리인3', TO_DATE('23/11/11',

```

```
'yy/mm/dd'));
INSERT INTO Caretaker VALUES ('A004', '004', '관리인4', TO_DATE('23/04/02',
'yy/mm/dd'));
INSERT INTO Caretaker VALUES ('A005', '005', '관리인5', TO_DATE('23/06/25',
'yy/mm/dd'));
```

#### [ 자바 연결 소스 - demol.java ]

```
import java.io.*;
import java.lang.*;
import java.sql.*;

public class demol {

    private Connection conn = null;

    public void connect() {
        try {
            Class.forName( "oracle.jdbc.driver.OracleDriver");
            // DriverManager 클래스에 등록된 드라이버를 사용하여 데이터베이스
에 연결
            conn =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "assignment3",
"1234");

            // DB 접속 정보 : 호스트 주소, 포트 번호, SID, 계정, 비밀번호
            //DB 접속, 계정 : assignment3 비밀번호 : 1234
            System.out.println( "성공적으로로딩되었음\n");
            // 성공적으로 접속되면 이 문구 출력

            Statement stmt = conn.createStatement ();
            // SQL 문을 실행하기 위한 Statement 객체를 생성
            ResultSet rset = stmt.executeQuery ("SELECT * FROM
Product");

            // Product 테이블에서 모든 데이터를 조회하는 쿼리를 실행
```



P003	삼양	불닭볶음면	30000	1050	2023-11-01 00:00:00	2000
P004	롯데	칸쵸	2000	950	2023-07-20 00:00:00	500
P005	오뚜기	오뚜기밥	8000	840	2023-04-15 00:00:00	1500

# [ 자바 GUI 구현 - demo2.java ]

```
// 5-2 demo2 package test;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

public class demo2 extends JFrame implements ActionListener {
    // 연결을 위한 변수 선언
    private Connection conn = null;
    // 맨 위 text 설정 (JLabel)
    private JLabel state;
    // Table Header Name
    private String columnNames[];
    // 테이블 데이터 (Header data(테이블 속성 리스트), adding row (행추가))
    private DefaultTableModel model = new DefaultTableModel(columnNames,
0;

    // Create Table
    private JTable table = new JTable(model);
    // 하나의 튜플에 대한 속성 값을 넣기 위한 변수
    private String row[];

    // 이미지 아이콘
    ImageIcon imageicon = new ImageIcon("picture.jpg");
    // JLabel 에 이미지 아이콘 추가
    JLabel label = new JLabel(imageicon);

    // 회원, 제조업체, 하청업체, 상품, 주문, 게시글, 관리인 버튼 생성
    JButton btn[] = new JButton[7];
    // 영어, 한글 둘다 작성 하였습니다. 영어로 할 시 주석처리 후 주석 풀어주시면 됩니다.
    private String table_name[] = { "회원", "제조업체", "하청업체", "상품", "주문",
"게시글", "관리인" };
    //private String table_name[] = { "Member", "Company", "SubCompany",
"Product", "Orders", "Post", "Caretaker" };

    // SQL 문을 실행하기 위한 Statement 객체
    private Statement stmt;
    // SQL 문 결과 저장 객체
    private ResultSet rset;

    // 버튼 JPanel에 테이블 조회를 위한 버튼 추가
    public void button_add(JPanel Btn_panel, JButton btn[], String
table_name[]) {
        for (int i = 0; i < btn.length; i++) {
            // 버튼 객체 생성
            btn[i] = new JButton(table_name[i]);
            // 버튼 Btn_panel 에 추가
            Btn_panel.add(btn[i]);
        }
    }

    // 버튼 리스너 추가
    public void button_listener(JButton btn[]) {
        for (int i = 0; i < btn.length; i++) {
            // 버튼마다 ActionListener 추가
            btn[i].addActionListener(this);
        }
    }

    // 버튼 리스너
    @Override
    public void actionPerformed(ActionEvent e) {
        // 누른 버튼 값 가져오기
    }
}
```



```

JButton target = JButton e.getSource();

try {
    // SQL 문을 실행하기 위한 Statement 객체 생성
    stmt = conn.createStatement();
    for (int i = 0; i < btn.length; i++) {
        if (btn[i] == target) {
            // 버튼의 해당되는 릴레이션 select 문 실행
            rset = stmt.executeQuery("select * from "
+ table_name[i]);

            // ResultSet (rset) 의 메타데이터 가져오기
            ResultSetMetaData metaData = rset.getMetaData();
            // 속성 이름을 저장하기 위한 배열 생성
            columnNames = new
String[metaData.getColumnCount()];
            // 속성 이름을 배열에 저장
            for (int j = 1; j <= metaData.getColumnCount(); j++) {
                columnNames[j - 1] = metaData.getColumnName(j);
            }

            // 기존 데이터 제거 후 속성이름 다시 배열
            model.setRowCount(0);
            model.setColumnIdentifiers(columnNames);
            // 튜플 데이터 저장을 위한 배열 생성
            row = new
String[rset.getMetaData().getColumnCount()];

            while (rset.next()) {
                // row 값읽기
                for (int k = 1; k <=
rset.getMetaData().getColumnCount(); k++) {
                    row[k - 1] =
rset.getString(k);
                }

                // JTable에출력
                model.addRow(row);
            }

            // 상태명 수정
            state.setText("DBProject_배준영 : " +
table_name[i] + "릴레이션");

            // 테이블 다시 그리기
            table.repaint();
            return;
        }
    }

} catch (SQLException el) {
    // 기존 model 테이블 데이터 제거
    model.setRowCount(0);
    // 속성이름 초기화
    String temp[] = {};
    model.setColumnIdentifiers(temp);
    el.printStackTrace();
    // 오류 문구 state text로 설정
    state.setText("DB 읽기 실패 : " + el.toString());
}

// DB 연결 함수
public void connect() {
    try {
        // DriverManager 클래스에 등록된 드라이버를 사용하여
데이터베이스에 연결

```

```

        Class.forName("oracle.jdbc.driver.OracleDriver");
        // DB 접속 정보 : 호스트 주소, 포트 번호, SID, 계정, 비밀번호
        // DB 접속, 계정 : assignment3 비밀번호 : 1234
        conn =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "assignment3",
"1234");
    }
    catch (ClassNotFoundException e) {
        // JDBC 드라이버를 찾을 수 없는 경우 예외 처리
        e.printStackTrace();
        // state 문구 설정
        state.setText("해당 드라이버를 찾을 수 없습니다 : " +
e.toString());
    }
    catch (SQLException e) {
        // 데이터베이스 연결 또는 쿼리 실행 중 오류 발생 시 예외 처리
        e.printStackTrace();
        // state 문구 설정
        state.setText("DB 연결 실패 : " + e.toString());
    }
}

// JFrame 초기 설정 함수
public void init() {
    // GUI 타이틀 설정
    setTitle("DataBase");
    // 닫히는 버튼 눌렀을때 동작
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    // BorderLayout을 활용해 동,서,남,북,중앙으로 컴포넌트 위치
    setLayout(new BorderLayout());
    // 하단부 연결 및 출력 버튼 컴포넌트용 패널
    JPanel Btn_panel = new JPanel();
    JPanel panel = new JPanel();
    // 버튼 패널을 FlowLayout으로 설정하여 왼쪽에서 오른쪽으로 컴포넌트 추가
    Btn_panel.setLayout(new FlowLayout());
    // 패널을 FlowLayout으로 설정하여 왼쪽에서 오른쪽으로 컴포넌트 추가
    panel.setLayout(new FlowLayout());
    // 아까 label (이미지) 를 panel (JPanel) 에 붙임
    panel.add(label);
    //DB 상태출력용라벨
    state = new JLabel();
    state.setText("Oracle DB 연동테스트");
    //JFrame에 배치관리자를 활용한 각자의 위치에 부착
    add(state, BorderLayout.NORTH);
    //JScrollPane : 화면에넘어갈경우스크롤바가생김
    add(new JScrollPane(table), BorderLayout.CENTER);
    add(Btn_panel, BorderLayout.EAST);
    add(panel, BorderLayout.WEST);

    // 테이블 조회 버튼 추가
    button_add(Btn_panel, btn, table_name);
    // 테이블 조회 버튼 이벤트 추가
    button_listener(btn);
    // 화면 크기 1500x500 으로 설정
    setSize(1500, 500);
    // 보여지도록 설정
    setVisible(true);
}

// demo2 생성자
public demo2() {
    // 자바 JFrame 초기 설정 함수 실행
    init();
    // DB 연결 함수 실행
    connect();
}

```

```

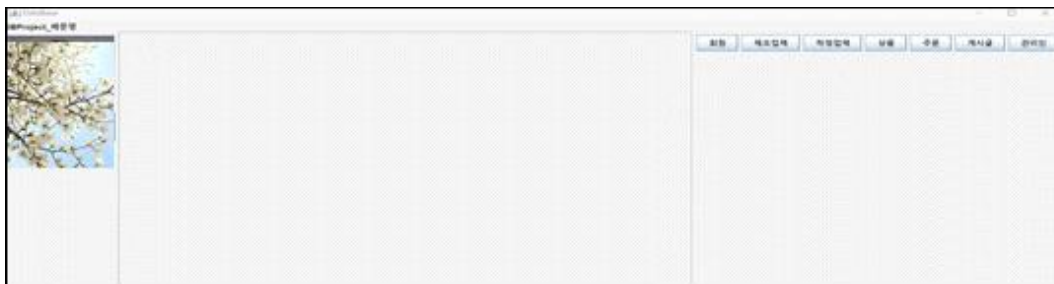
        // JLabel 의 text 설정
        state.setText("DBProject_홍길동");
    }

    public static void main(String[] args) {
        new demo2();
    }
}

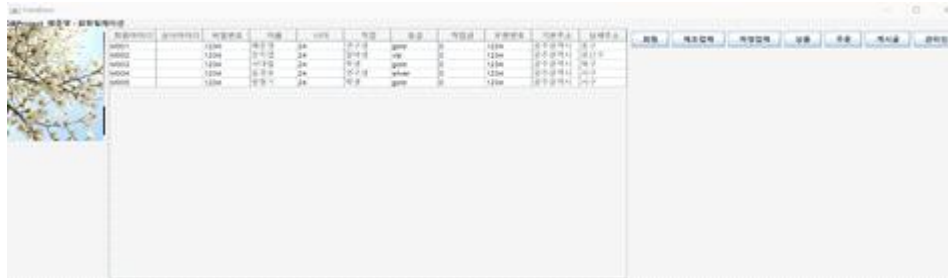
```

## [ 최종결과 ]

### I. 처음 실행 시



## 2. 회원 버튼 클릭



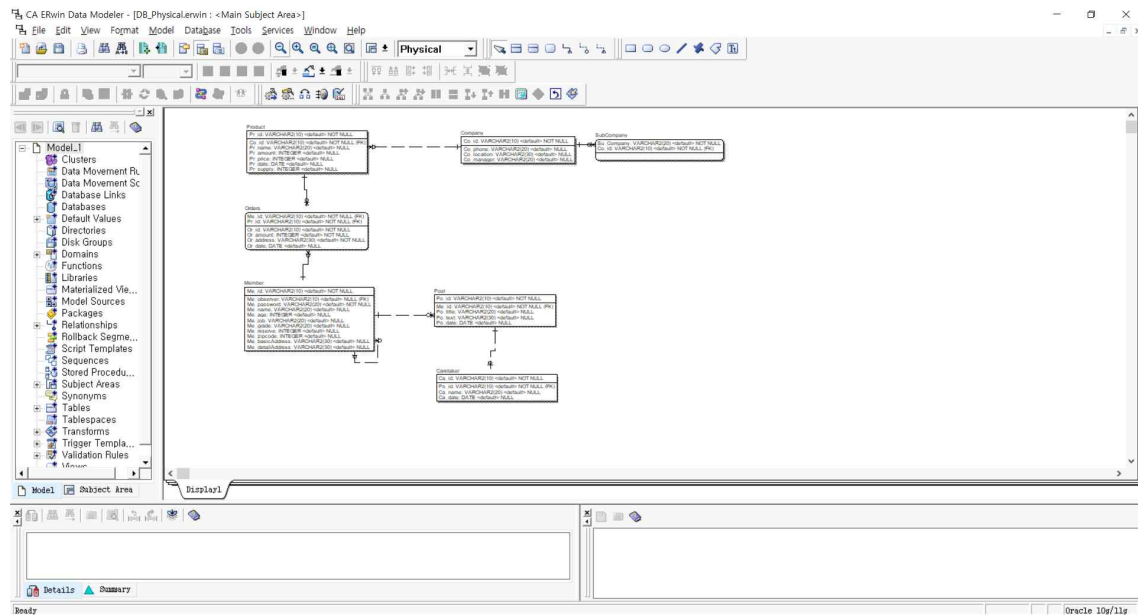
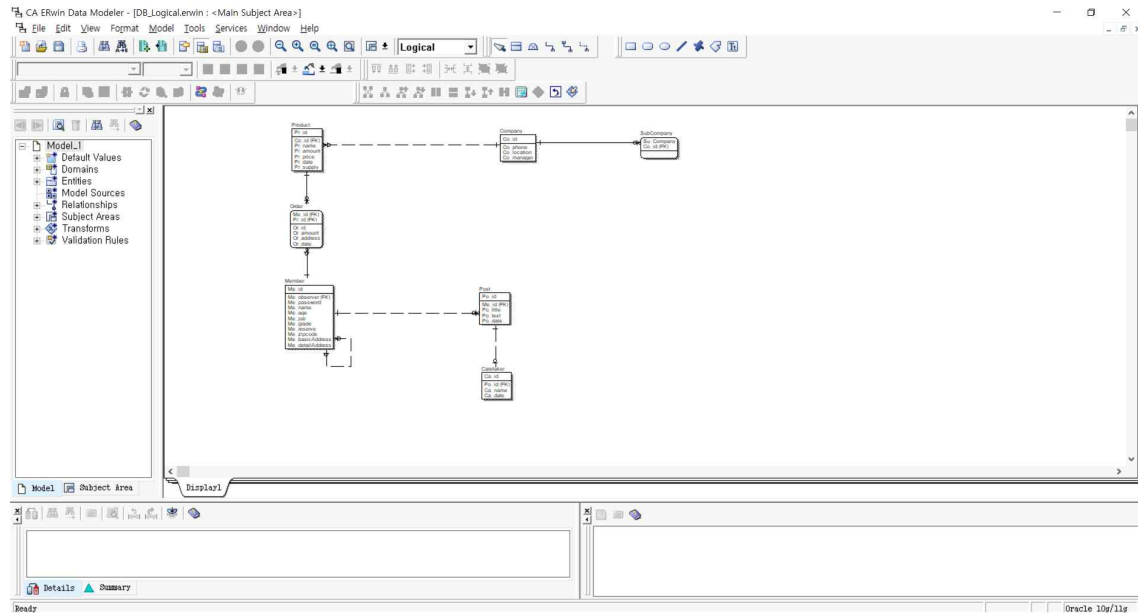
회원번호	회원이름	비밀번호	이름	성명	성명	성명	성명	성명	성명
00001	김민준	1234	김민준	김민준	김민준	김민준	김민준	김민준	김민준
00002	이민준	1234	이민준	이민준	이민준	이민준	이민준	이민준	이민준
00003	박민준	1234	박민준	박민준	박민준	박민준	박민준	박민준	박민준
00004	정민준	1234	정민준	정민준	정민준	정민준	정민준	정민준	정민준
00005	최민준	1234	최민준	최민준	최민준	최민준	최민준	최민준	최민준

// 7개 각각 클릭하면 되도록 해야 한다.

## 6. 논리적 설계

### - ERWin - ERD작성

### ERWin 모델 - 소스 첨부하기



## 7. 물리적 설계

- 1) 데이터베이스 스키마
- 2) 데이터베이스 상태
- 3) 정규화
  - 제1정규화
  - 제2정규화
  - 제3정규화

(정규화 예시- 엑셀이용)

주문번호	주문일	회원이름	회원번호	주소	전화	도서번호	도서명	수량	단가
20060303001	2006-03-03	장다연	aurty	경기도 일산	031-964-1004	8982413820	기름하는 리다가 되라	10	9500
						898241455x	Being Java	1	22000
20060408943	2006-04-08	옥주현	yoga	서울시 강남구	02-365-9999	8982413820	기름하는 리다가 되라	5	9500
20060510519	2006-05-10	장다연	aurty	경기도 일산	031-964-1004	8982410090	알기쉽게 해설한 C	2	20000
제 1 정규화									
주문번호	주문일	회원이름	회원번호	주소	전화	도서번호	도서명	수량	단가
20060303001	2006-03-03	장다연	aurty	경기도 일산	031-964-1004	8982413820	기름하는 리다가 되라	10	9500
20060303001	2006-03-03	장다연	aurty	경기도 일산	031-964-1004	898241455x	Being Java	1	22000
20060408943	2006-04-08	옥주현	yoga	서울시 강남구	02-365-9999	8982413820	기름하는 리다가 되라	5	9500
20060510519	2006-05-10	장다연	aurty	경기도 일산	031-964-1004	8982410090	알기쉽게 해설한 C	2	20000
제 2 정규화									
주문번호	주문일	회원이름	회원번호	주소	전화	주문번호	도서번호	수량	
20060303001	2006-03-03	장다연	aurty	경기도 일산	031-964-1004				
20060408943	2006-04-08	옥주현	yoga	서울시 강남구	02-365-9999				
20060510519	2006-05-10	장다연	aurty	경기도 일산	031-964-1004				
						20060303001	8982413820	10	
						20060303001	898241455x	1	
						20060408943	8982413820	5	
						20060510519	8982410090	2	
제 3 정규화									
주문번호	주문일	회원이름	회원번호	주소	전화	주문번호	도서번호	수량	단가
20060303001	2006-03-03	장다연	aurty	경기도 일산	031-964-1004	20060303001	8982413820	10	9500
20060408943	2006-04-08	옥주현	yoga	서울시 강남구	02-365-9999	20060303001	898241455x	1	22000
20060510519	2006-05-10	장다연	aurty	경기도 일산	031-964-1004	20060408943	8982413820	5	9500
						20060510519	8982410090	2	20000

## 4) 데이터베이스 테이블 작성

//DDL 테이블 작성 예시

// 학생 테이블 - student

create table student (

sid number(10),

sname varchar2(10),

age number(4),

address varchar2(20),

primary key(sid)

);

// DML 학생 테이블 - student 에 대해 투플을 10개 정도 삽입한다.

insert into student values(20100001,'용선정', 22, '광주 서구');

insert into student values(20100002,'최오성', 23, '광주 북구');

insert into student values(20100003,'김창민', 24, '광주 남구');

.....

[ 주문 릴레이션 데이터 삽입 ]

INSERT INTO 주문 VALUES ('M001', 'P001', 'O001', 3, '서울시 강남구', TO\_DATE('23/11/07', 'yy/mm/dd'));

.....

// DML 학생 테이블 확인한다.

select \* from student;

## 8. 구현 및 데모(과제 3참고)

사용자 이름 : sugang 비밀번호 : 1234를 입력한다.

[ 자바 연결 소스 - demol.java ]

[ 자바 GUI 구현 - demo2.java ]

eclipse에서 해당 프로젝트를 import 시킨 후 실행한다.

// 본인얼굴사진 작게 넣기, 삽입, 조회, 삭제, 수정이 되어야 한다.

사용자 이름 : sugang 비밀번호 : 1234를 입력한다.

※ java 어플리케이션에서 접속 할 오라클 계정과 비밀번호 임으로 같아야 한다.

SQL -> SQL 명령 창으로 들어가 응용 프로그램에 사용될 테이블을 삽입한다.

- 삽입할 테이블 sql문은 해당 CD의 Create Table.txt 파일에 기록되어 있다.
- Create 명령어 1개씩 복사하여 SQL 명령 창에 복사 후 실행한다.
- 테이블 생성 순서는 외래 키(Foreign key)가 없는 테이블 우선 생성 후 외래 키를 생성하여야 한다.

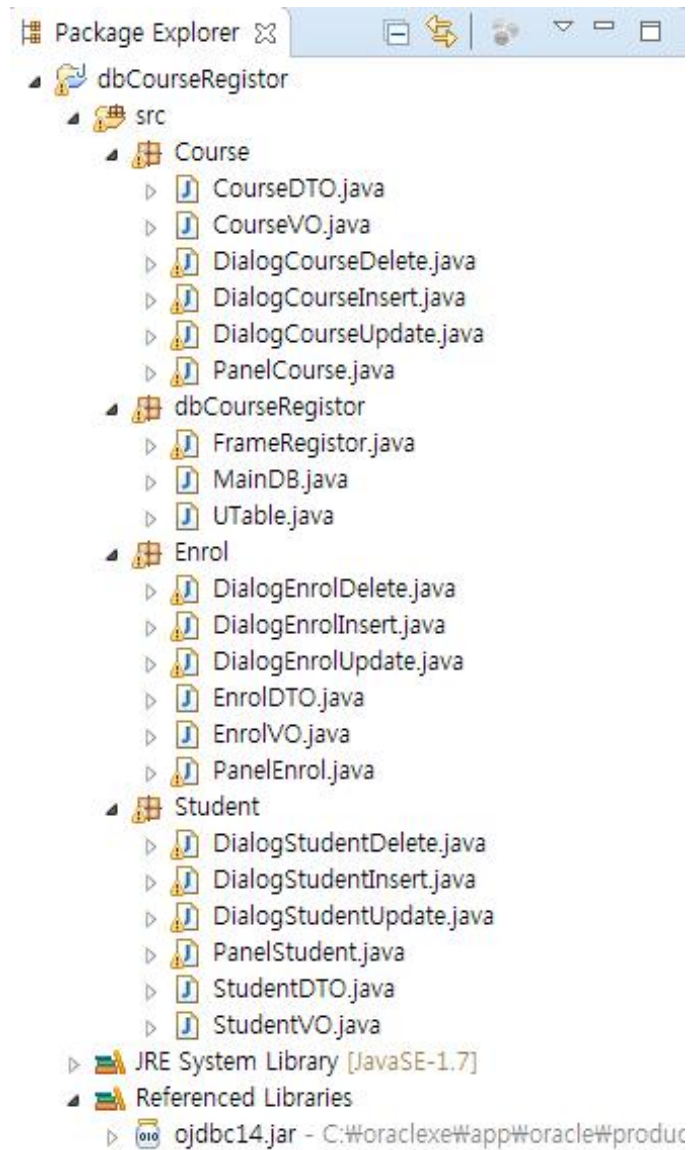
레코드는 삽입할 양이 많을 경우, csv 파일로 작성해 두고,오라클 홈으로 이동 후, 유틸리티 -> 데이터 로드/

업로드 -> 로드 -> 스프레드시트 데이터 로드 -> 기존테이블, 파일 업로드 -> 테이블에 맞는 .csv 파일 선택, 문자집합

선택 -> 날짜데이터 경우 데이터 형식 기입 (YYYY/MM/DD HH24:MI:SS) -> 데이터 삽입 완료

※ 외래 키(Foreign key)가 아닌 테이블 부터 삽입하여야 한다.

자바 프로젝트 과제 리스트



시작 첫 화면(MainDB.java)



수강신청 창

### 학생 정보

아이디	이름	나이	주소

### 수강 정보

학생 아이디	과목 아이디	점수

### 과목 정보

아이디	제목	교수	학년

각각 검색 눌렀을 때 화면

수강신청 창

### 학생 정보

아이디	이름	나이	주소
20100002	최오성	23	광주 북구
20100003	김창민	24	광주 남구
20100001	윤선정	22	광주 서구

### 수강 정보

학생 아이디	과목 아이디	점수
20100001	600001	85
20100002	600002	90
20100003	600001	75

### 과목 정보

아이디	제목	교수	학년
600002	유닉스시스템	이길동	3
600001	자료구조	홍길동	2

학생 정보에서 추가 버튼 누르고 데이터 입력한 창(DialogStudentInsert.java)

학생 정보 입력창

아이디

이름

나이

주소

학생 정보에서 삭제 버튼 누르고 데이터 입력한 창(DialogStudentDelete.java)



학생 정보에서 변경 버튼 누르고 데이터 입력한 창(DialogStudentUpdate.java)



9. 소감 및 느낀점

10. 참고문헌

- 사이트
- 책

11. 구글 드라이브 사이트