

# 데이터베이스 설계

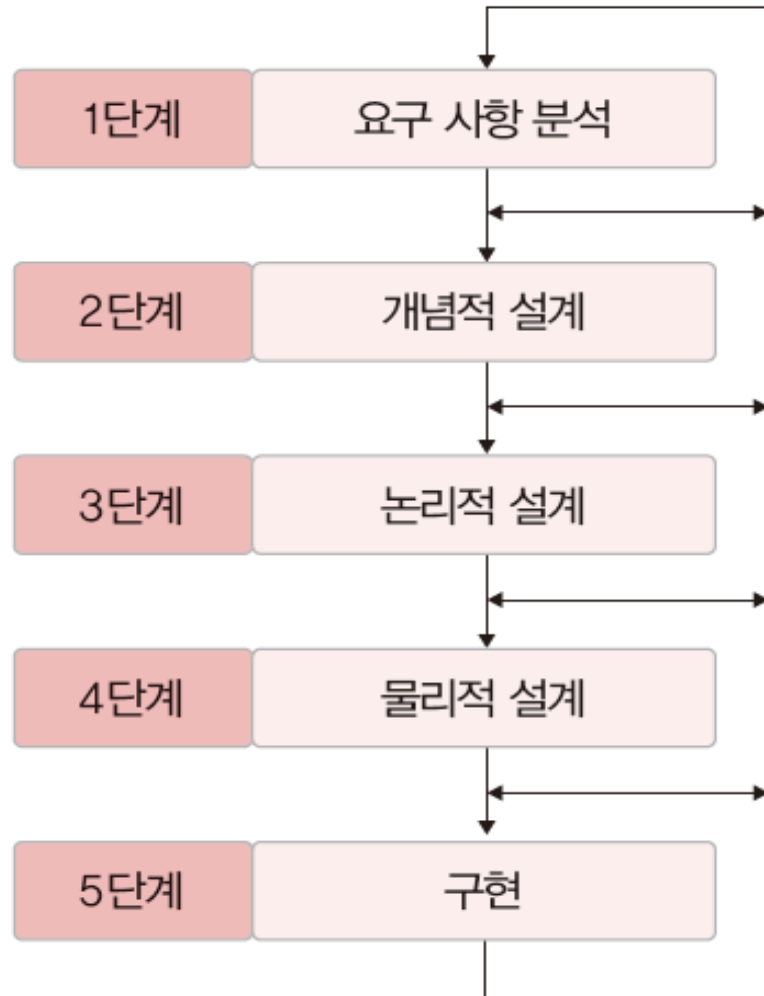
# 데이터베이스 설계의 개념

---

- 현실 세계의 업무적인 프로세서를 물리적으로 데이터베이스화하기 위한 과정.
- 실체와 관계를 중심으로 체계적으로 표현하고 문서화하여, 현실 세계를 컴퓨터가 처리할 수 있는 데이터로 변환.

# 데이터베이스 설계

## ➤ 데이터베이스 구축을 위한 절차



설계 과정 중에 오류가 발견되어 변경  
이 필요하면 이전 단계로  
되돌아가 설계 내용을 변경 가능

데이터베이스 설계의 과정

# 1단계. 요구 사항 분석 / 2단계. 개념적 설계

---

## ➤ 1. 요구 사항 분석

- 사용자나 관리자들의 면담이나 주요 문서들을 통해서 시스템 개발 목표가 무엇이고 요청자의 의도를 파악하여 요구되는 정보가 무엇인지 정확하게 분석해야 함.
- 사용자가 원하는 데이터베이스의 용도를 파악하는 단계.

## ➤ 2. 개념적 설계

- 사용자의 요구사항을 이해하기 쉬운 형식으로 간단히 기술하는 단계.

## 3단계. 논리적 데이터 모델링 단계

---

- 오라클은 관계형 데이터베이스 중의 하나이기 때문에 관계형 모델을 근거로 엔티티 관계 모델링(E-R, Entity-Relationship Modeling)을 합니다. 엔티티 관계 모델링은 시스템 내에 존재하는 엔티티(개체)와 이들 간의 관계로 표현되며 주요 구성 요소는 엔티티, 속성, 관계임.
- 사용자나 관리자들의 면담이나 주요 문서들을 통해서 시스템 개발 목표가 무엇이고 요청자의 의도를 파악하여 요구되는 정보가 무엇인지 정확하게 분석해야 함.

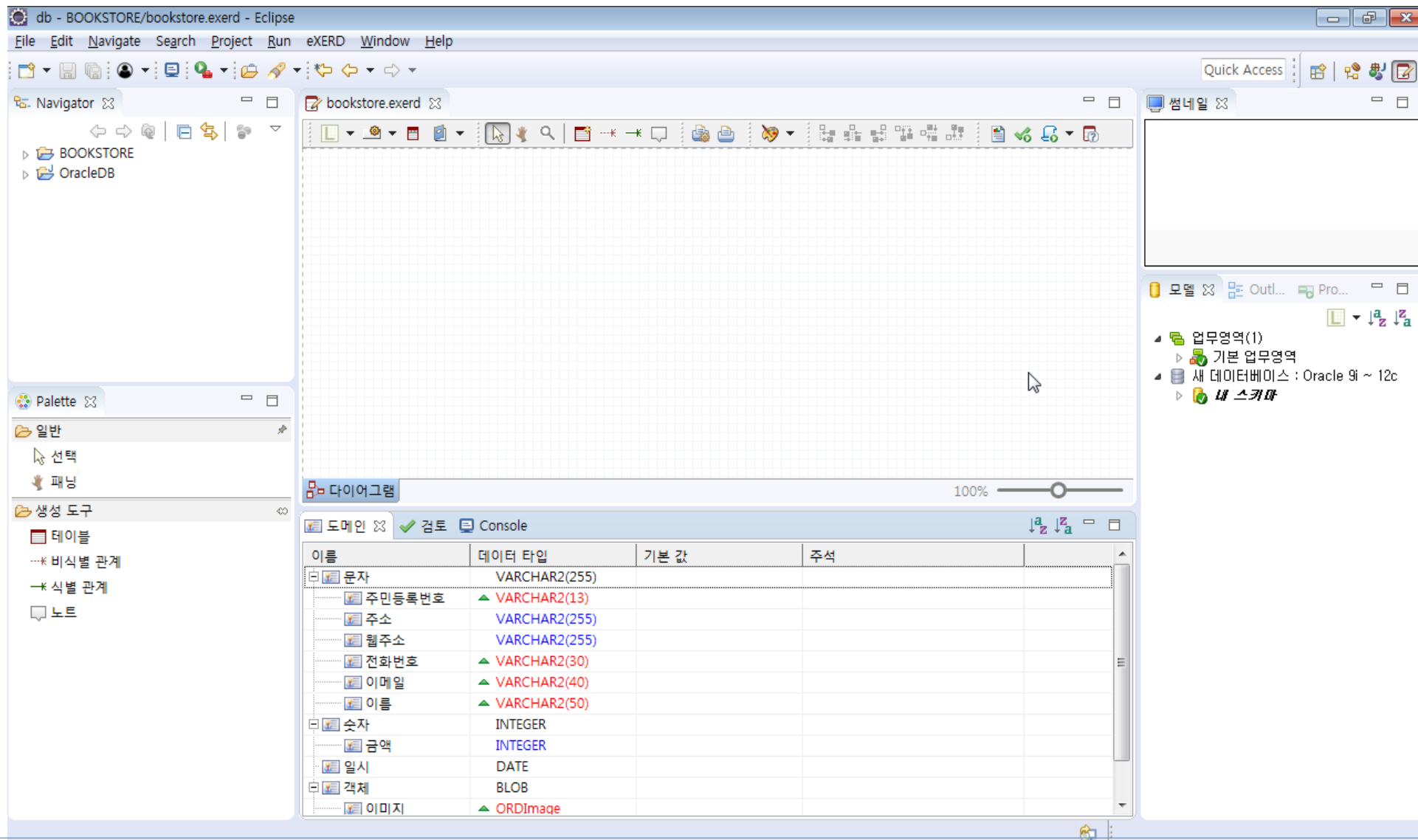
# 4단계. 물리적 데이터 모델링 단계

➤ 물리적 데이터 모델링은 논리적 설계의 산출물인 엔티티, 관계, 속성, 식별자들을 테이블, 칼럼, 키 등으로 변환하는 과정.

테이블명	tb_zipcode	Table 기술서		작성일	2019/10/20		Page 7
System	nonageShop			작성자	홍길동		
테이블 설명		가입한 회원 정보를 관리합니다					
No	Attribute	Data Type	NN	Ky	Default	Description	
1	zipcode	varchar(7)	Y	PK		우편번호	
2	sido	varchar(30)				시도	
3	gugum	varchar(30)				구군	
4	dong	varchar(30)				동	
5	bunji	varchar(30)				번지	

# exERD로 논리적 / 물리적 모델링

➤ exERD는 모델링을 위한 자동화 도구



# exERD로 논리적 / 물리적 모델링

The screenshot shows the exERD software interface with two tables defined: DEPT\_EXERD and EMP\_EXERD. The interface includes a toolbar with various modeling tools and a status bar at the bottom.

**DEPT\_EXERD**

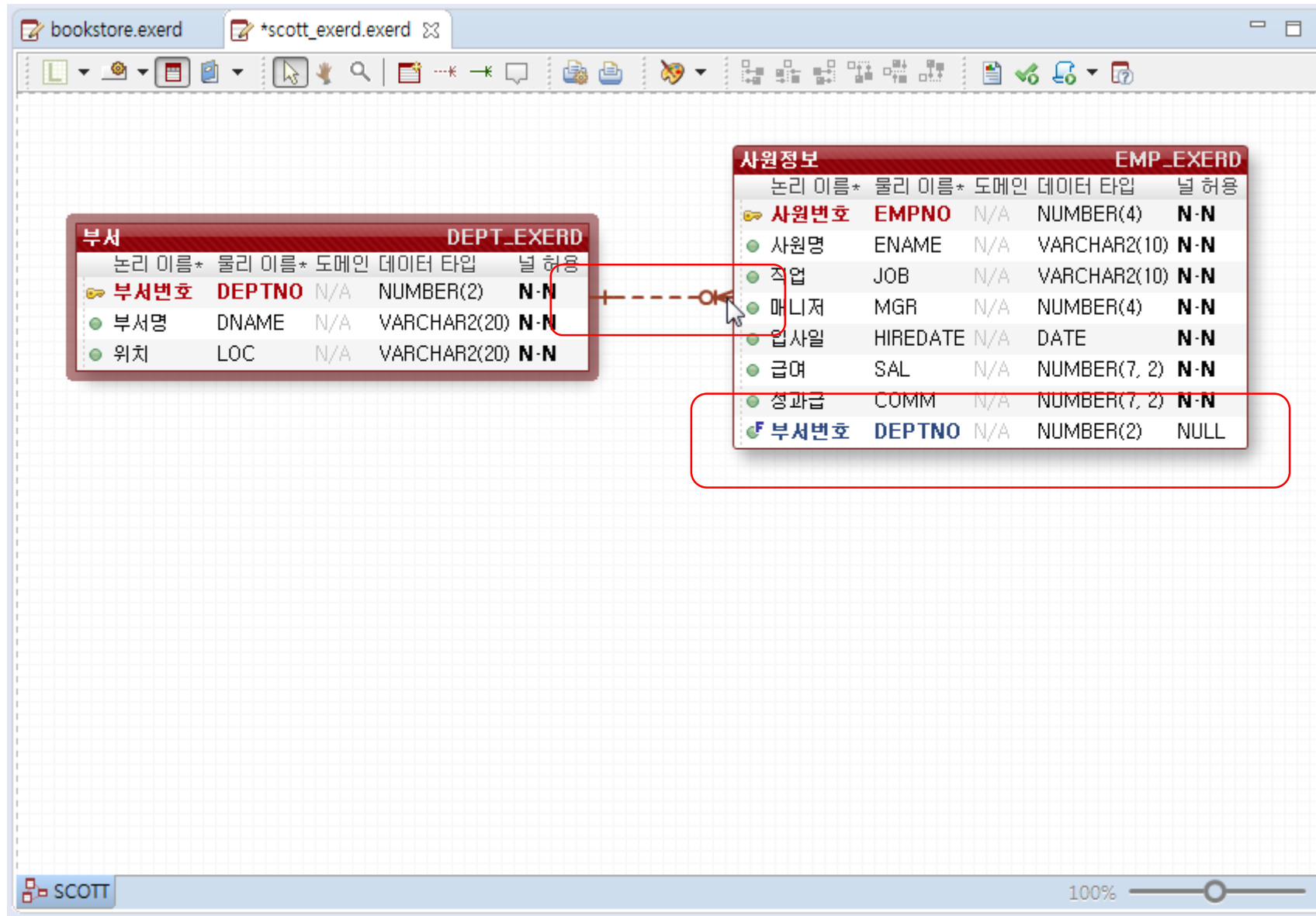
논리 이름*	물리 이름*	도메인	데이터 타입	널 허용
부서번호	DEPTNO	N/A	NUMBER(2)	N·N
부서명	DNAME	N/A	VARCHAR2(20)	N·N
위치	LOC	N/A	VARCHAR2(20)	N·N

**EMP\_EXERD**

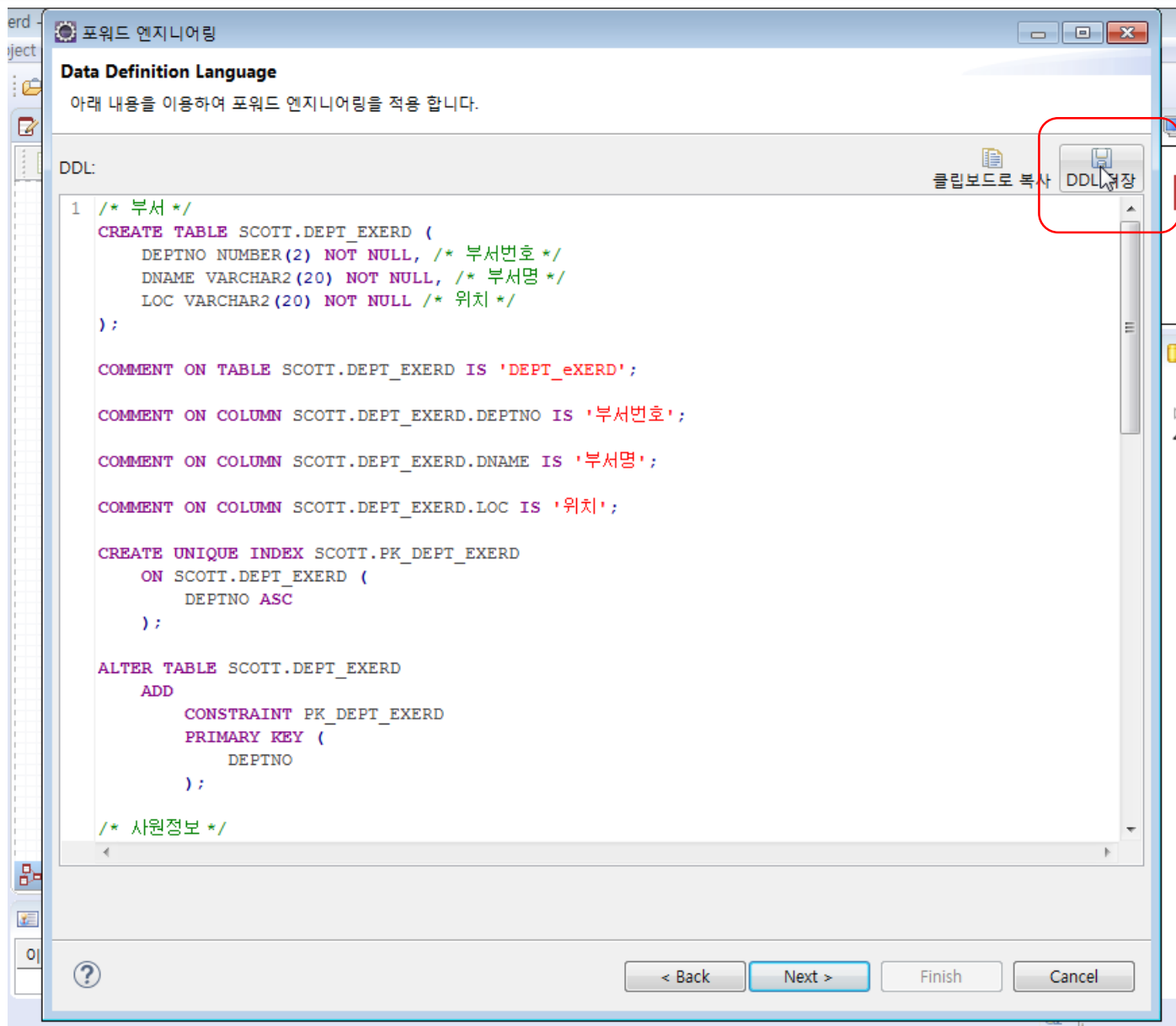
논리 이름*	물리 이름*	도메인	데이터 타입	널 허용
사원번호	EMPNO	N/A	NUMBER(4)	N·N
사원명	ENAME	N/A	VARCHAR2(10)	N·N
직업	JOB	N/A	VARCHAR2(10)	N·N
매니저	MGR	N/A	NUMBER(4)	N·N
입사일	HIREDATE	N/A	DATE	N·N
급여	SAL	N/A	NUMBER(7, 2)	N·N
성과금	COMM	N/A	NUMBER(7, 2)	N·N



# exERD로 논리적 / 물리적 모델링



## 5단계. 데이터베이스 구축



# SQL 문 사용하기

---

- 회원 정보 추가하기

```
insert into member values  
( 'one', '1111', '김일동', '152-761',  
'서울시 구로구 구로1동 주공아파트104-1004호', '010-111-1111');  
insert into member values  
( 'two', '2222', '홍길동', '143-200',  
'서울시 광진구 구의동 우성 아파트 302-1403호', '010-222-2222');
```

- 가입된 회원의 아이디로 패스워드 검색하기

```
select pwd  
from member  
where id='one';
```

# SQL 문 사용하기

---

- 시퀀스 생성하기

```
CREATE SEQUENCE ORDER_SEQ  
START WITH 1  
INCREMENT BY 1  
MAXVALUE 100000 ;
```

- 트리거 생성하기

```
create or replace trigger orders_tr01  
before insert  
on orders  
for each row  
begin  
select ORDER_SEQ.nextval into :new.o_seq from dual;  
end;  
/
```

# SQL 문 사용하기

---

- 조인하기

```
select *  
from orders o, member m, products p  
where o.id = m.id  
and o.product_code = p.product_code;
```

- 뷰 생성하기

```
create view o_m_p_view  
as  
select *  
from orders o, member m, products p  
where o.id = m.id  
and o.product_code = p.product_code;
```