웹서버프로그래밍

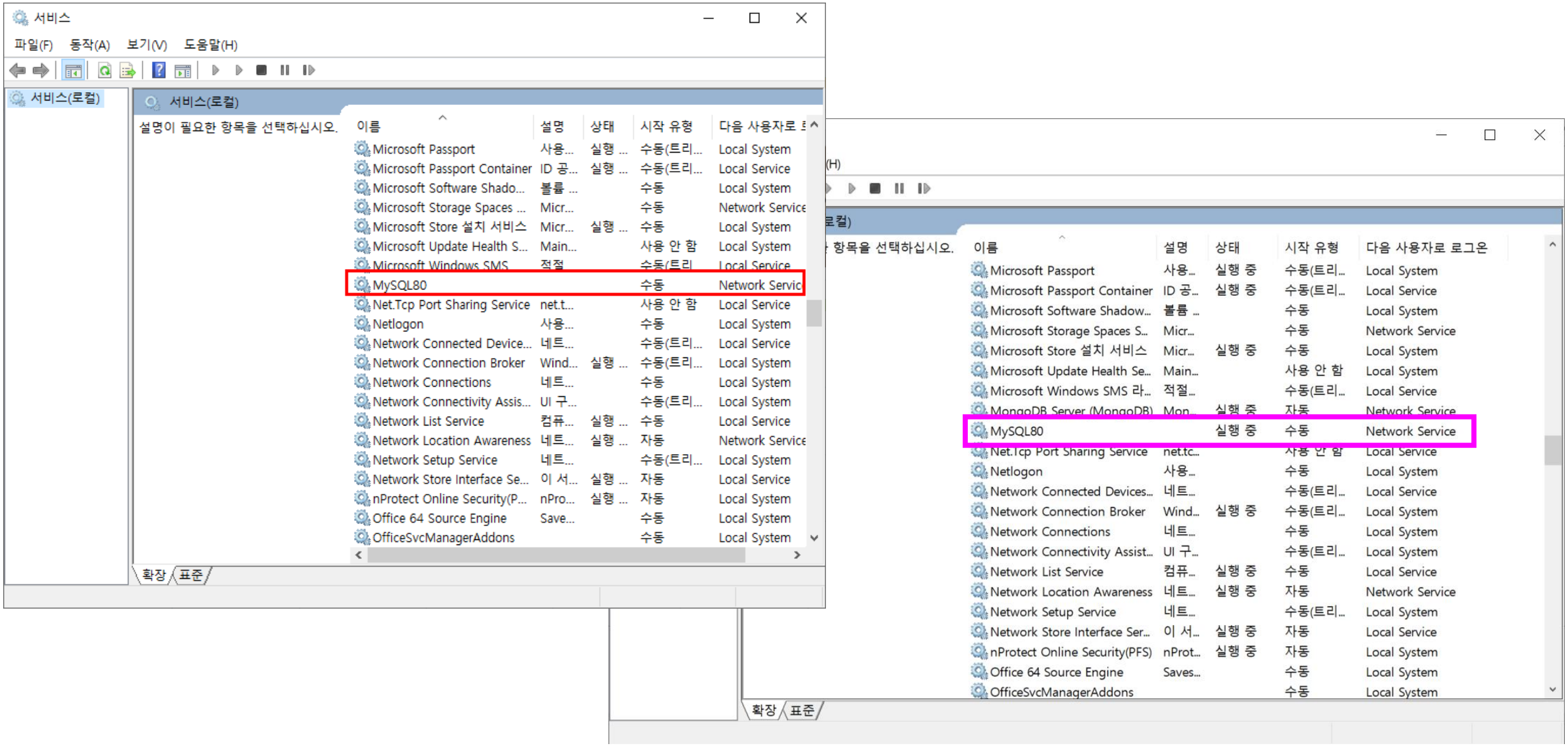
07. 데이터베이스와 SQL, JSP 연동 2

한성대학교 컴퓨터공학부 신 성

데이터베이스 만들기와 SQL 2

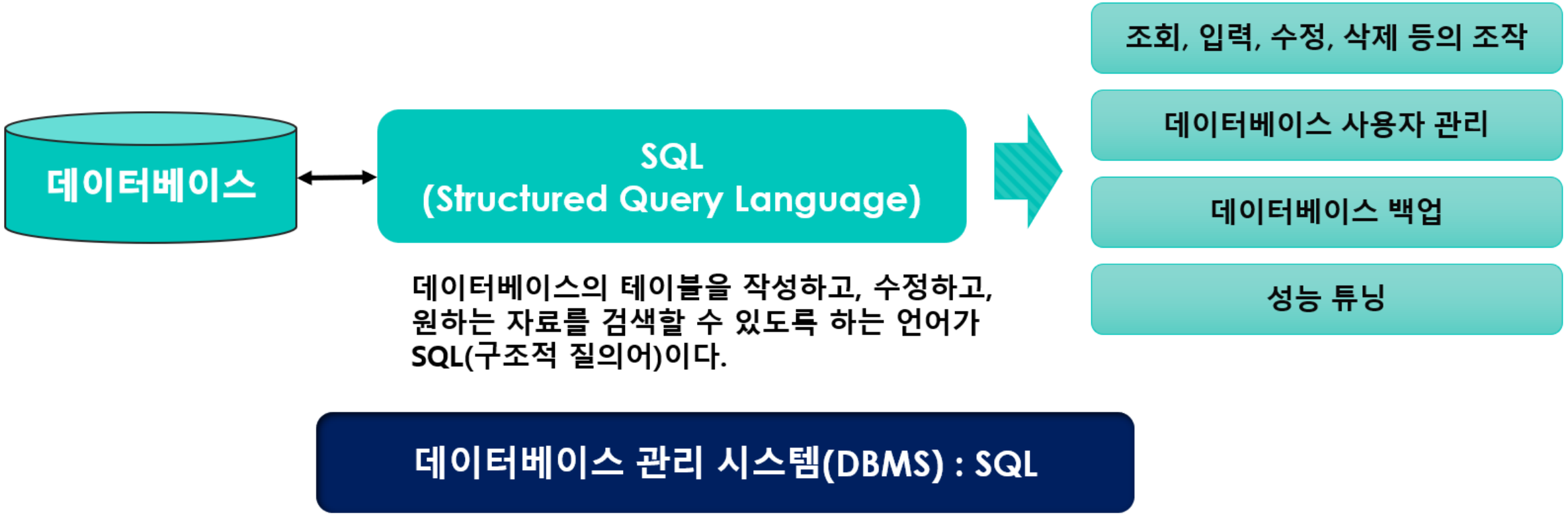
인사관리 데이터베이스 만들기

[참고] 실습 전에 다음 MySQL 서버 실행 여부 확인



SQL : Structured Query Language

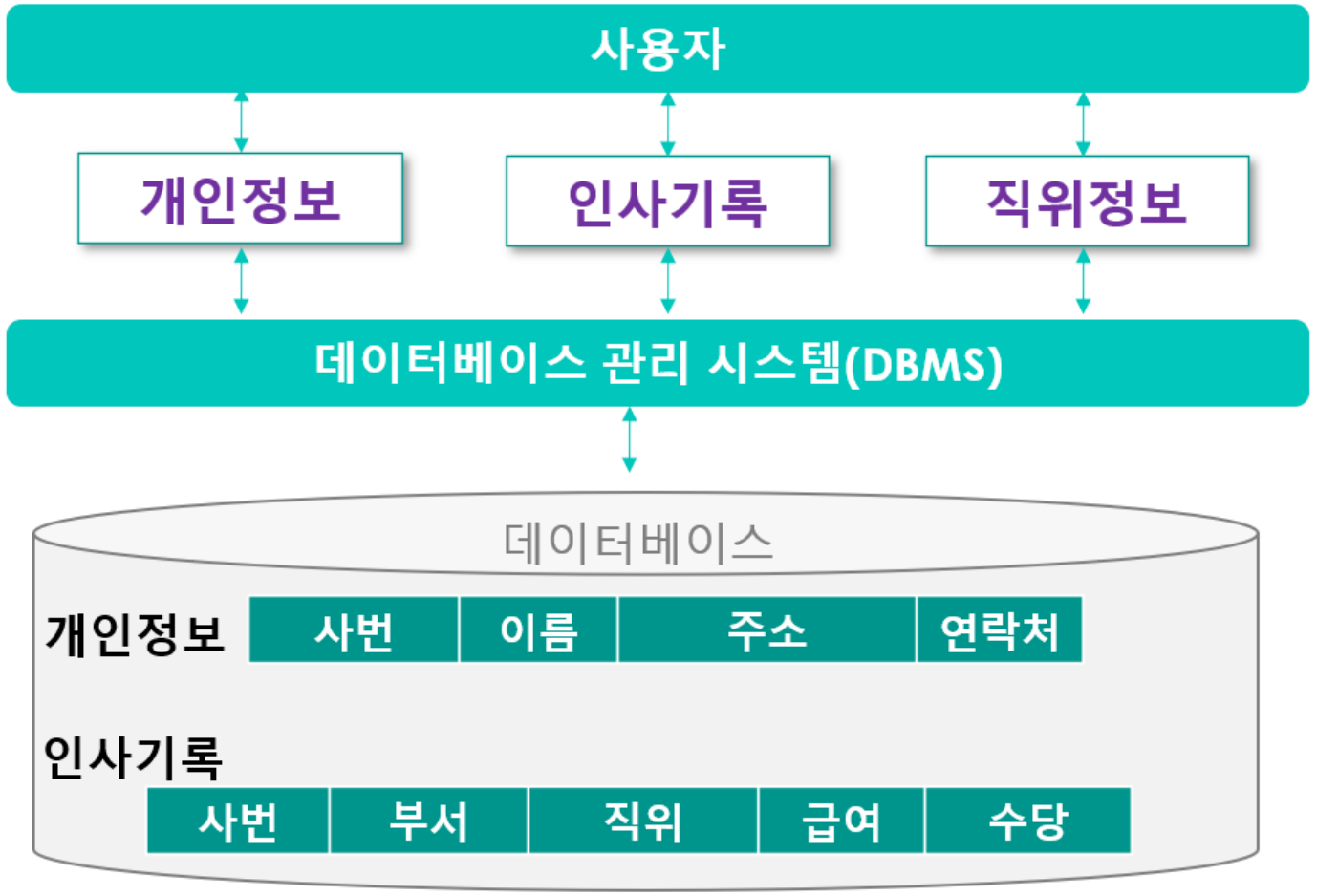
- SQL은 사용자와 데이터베이스 시스템 간에 의사소통을 하기 위한 언어.



데이터베이스 만들기

<인사관리 데이터베이스 만들기>

- 회사의 인사관리는
  개인정보, 인사기록, 직위정보 등이 있다.
- 개인정보와 인사기록에는 어떤 속성이 필요할까?



- 인사관리 데이터베이스와
  개인정보, 인사기록 테이블을
  만들어보자.

인사관리 PM(Personnel Management) 데이터베이스 만들기

CREATE DATABASE PM;

USE PM;

※ 실습은 온라인 강의 참고

- 사용 방법(지난 시간 수업)

```
mysql> CREATE DATABASE 데이터베이스명;
```

```
mysql> DROP DATABASE 데이터베이스명;
```

```
mysql> USE 데이터베이스명;
```

테이블 만들기

1. 인사관리(PM) 데이터베이스 만들기
 - 앞의 실습과 같이 인사관리 데이터베이스 생성

속성 만들기

2. 인사관리 데이터베이스 안에 [개인정보] 테이블 만들기

개인정보

사번	이름	주소	연락처

```
CREATE TABLE 개인정보 (
   사번      INTEGER   NOT   NULL,
   이름      VARCHAR(20),
   주소      VARCHAR(20),
   연락처    VARCHAR(20),
   PRIMARY KEY(사번)
);
```

테이블명

열이름 저장공간 값의 허용(옵션) 구분값

열이름 저장공간 구분값

열이름 저장공간 구분값

열이름 저장공간 구분값

기본키

INSERT

2.1 [개인정보] 테이블에 데이터 입력하기

개인정보(사번, 이름, 학과, 주소, 연락처)

개인정보

사번	이름	주소	연락처
20190001	김한성	서울시 성북	010-1234-5678
20190002	홍길동	서울시 구로	010-2345-6789
20190003	박나래	충남 아산	010-2356-7890
20190004	유재석	강원도 춘천	010-4567-8901
20190005	강호동	경기도 평택	010-5678-9012

다음과 같이 여러 데이터 입력 가능

```
INSERT INTO 개인정보  VALUES (20190001, '김한성', '서울시 성북', '010-1234-5678');
INSERT INTO 개인정보  VALUES (20190002, '홍길동', '서울시 구로', '010-2345-6789');
INSERT INTO 개인정보  VALUES (20190003, '박나래', '충남 아산', '010-3456-7890');
INSERT INTO 개인정보  VALUES (20190004, '유재석', '강원도 춘천', '010-4567-8901');
INSERT INTO 개인정보  VALUES (20190005, '강호동', '경기도 평택', '010-5678-9012');
```

INSERT

3. 인사관리 데이터베이스 안에 [인사기록] 테이블 만들기

인사기록

사번	부서	직위	급여	수당	점수

```
CREATE TABLE  인사기록 (
    사번    INTEGER   NOT NULL,
    부서    VARCHAR(20),
    직위    VARCHAR(20),
    급여    INTEGER,
    수당    INTEGER,
    점수    INTEGER,
    PRIMARY KEY(사번)
);
```

테이블명

열이름 저장공간 기본키 지정

열이름 저장공간

열이름 저장공간

열이름 저장공간

열이름 저장공간

열이름 저장공간

INSERT

3.1. [인사기록] 테이블에 데이터 입력하기

인사기록(사번, 부서, 직위, 급여, 수당, 점수)

인사기록

사번	부서	직위	급여	수당	점수
20190001	전산팀	대리	3000000	500000	95
20190002	인사팀	부장	5000000	300000	83
20190003	경리팀	대리	3000000	200000	65
20190004	마케팅	과장	4000000	250000	72
20190005	전산팀	대리	3000000	150000	60

```
INSERT INTO 인사기록 VALUES (20190001, '전산팀', '대리', 3000000, 500000, 95);
INSERT INTO 인사기록 VALUES (20190002, '인사팀', '부장', 5000000, 300000, 83);
INSERT INTO 인사기록 VALUES (20190003, '경리팀', '대리', 3000000, 200000, 65);
INSERT INTO 인사기록 VALUES (20190004, '마케팅', '과장', 4000000, 250000, 72);
INSERT INTO 인사기록 VALUES (20190005, '전산팀', '대리', 3000000, 150000, 60);
```

생성된 테이블

<인사관리(PM)>

개인정보

사번	이름	주소	연락처
20190001	김한성	서울시 성북	010-1234-5678
20190002	홍길동	서울시 구로	010-2345-6789
20190003	박나래	충남 아산	010-2356-7890
20190004	유재석	강원도 춘천	010-4567-8901
20190005	강호동	경기도 평택	010-5678-9012

인사기록

사번	부서	직위	급여	수당	점수
20190001	전산팀	대리	3000000	500000	95
20190002	인사팀	부장	5000000	300000	83
20190003	경리팀	대리	3000000	200000	65
20190004	마케팅	과장	4000000	250000	72
20190005	전산팀	대리	3000000	150000	60

※ 실습은 온라인 강의 참고

SELECT, UPDATE, DELETE

INSERT, SELECT, UPDATE, DELETE

- ☞ 데이터 입력하기 INSERT
- ☞ 데이터 검색하기 SELECT
- ☞ 데이터 수정하기 UPDATE
- ☞ 데이터 삭제하기 DELETE

인사기록

사번	부서	직위	급여	수당	점수
20190001	전산팀	대리	3000000	500000	95
20190002	인사팀	부장	5000000	300000	83
20190003	경리팀	대리	3000000	200000	65
20190004	마케팅	과장	4000000	250000	72
20190005	전산팀	대리	3000000	150000	60

데이터 검색하기

4. 테이블 전체 검색하기

SELECT * FROM 인사기록;

*는 테이블의 전체(all)를 검색할 때 사용

SELECT는 데이터베이스 테이블에서 필요한 자료를 검색할 때 사용하는 명령어

형식 : SELECT * FROM 테이블이름;

인사기록

사번	부서	직위	급여	수당	점수
20190001	전산팀	대리	3000000	500000	95
20190002	인사팀	부장	5000000	300000	83
20190003	경리팀	대리	3000000	200000	65
20190004	마케팅	과장	4000000	250000	72
20190005	전산팀	대리	3000000	150000	60

전체(all)

예)

SELECT * FROM 개인정보;

데이터 검색하기

4.1 테이블에서 특정 속성 검색하기

테이블의 전체(all)가 아닌

특정 속성들인

부서와 직위 속성만 검색하려면?

인사기록

사번	부서	직위	급여	수당	점수
20190001	전산팀	대리	3000000	500000	95
20190002	인사팀	부장	5000000	300000	83
20190003	경리팀	대리	3000000	200000	65
20190004	마케팅	과장	4000000	250000	72
20190005	전산팀	대리	3000000	150000	60

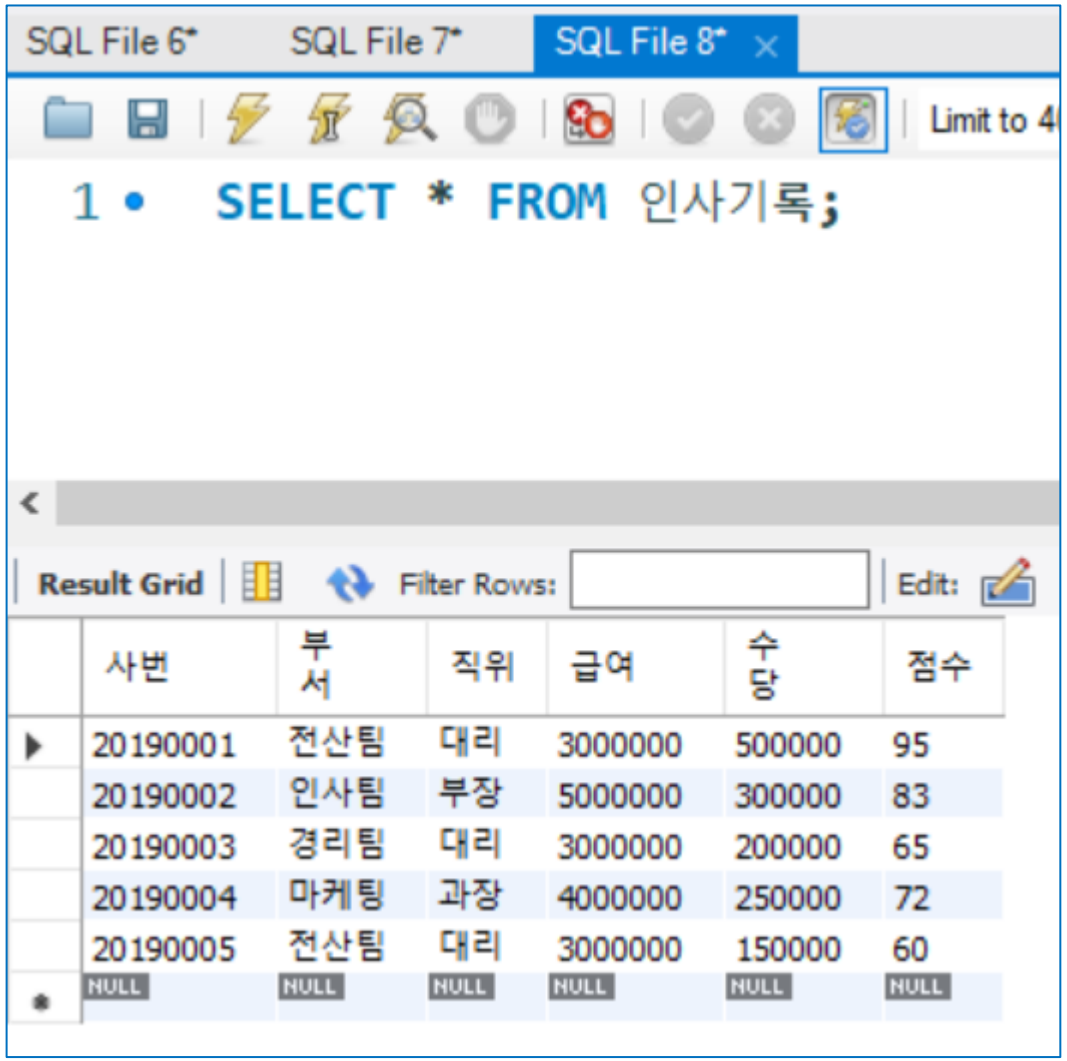
```
SELECT 부서, 직위 FROM 인사기록;
```

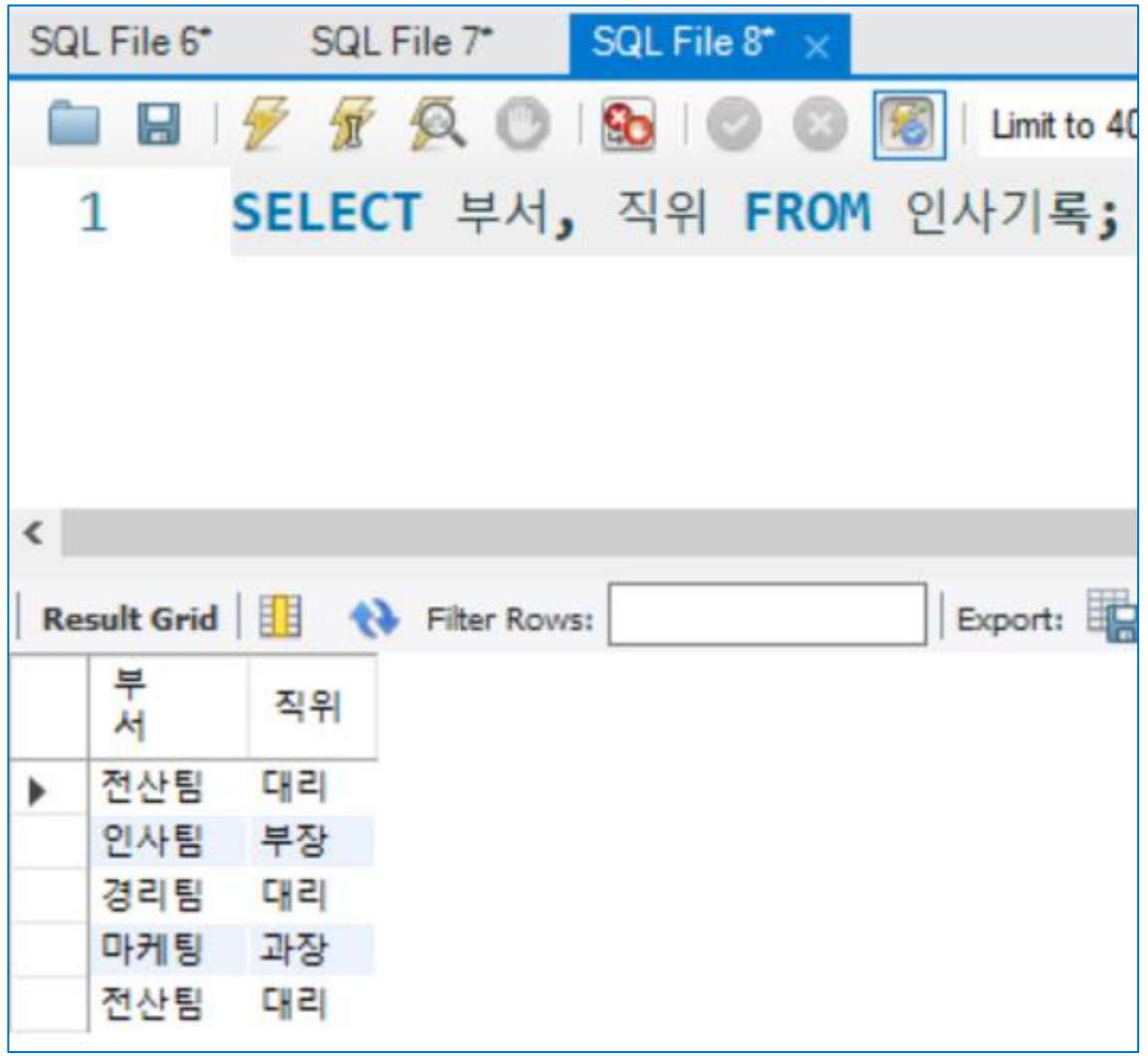
형식 : SELECT 속성이름 FROM 테이블이름;

인사기록

부서	직위
인사팀	과장
인사팀	부장
경리팀	대리
마케팅	과장
전산팀	대리

[실습] 다음과 같이 직접 실습해서 확인해 보시기 바랍니다.





데이터 검색하기

4.2 테이블에서 조건을 이용하여 특정 값만 검색하기

테이블의 전체(all) 내용 중

특정 속성의 값인

직위가 대리인 데이터만 검색하려면?

인사기록

사번	부서	직위	급여	수당	점수
20190001	전산팀	대리	3000000	500000	95
20190002	인사팀	부장	5000000	300000	83
20190003	경리팀	대리	3000000	200000	65
20190004	마케팅	과장	4000000	250000	72
20190005	전산팀	대리	3000000	150000	60

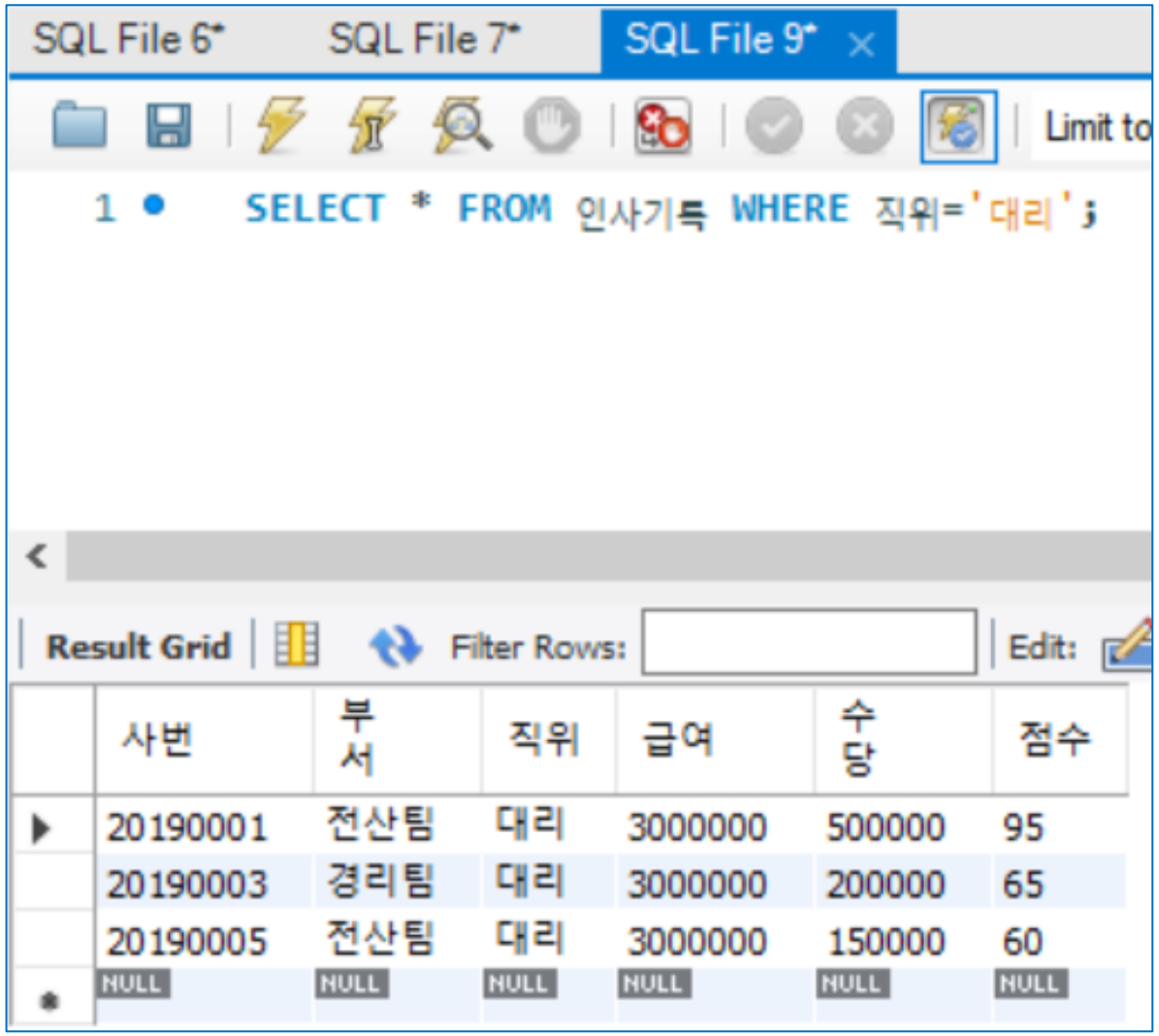
```
SELECT * FROM 인사기록 WHERE 직위='대리';
```

형식 : SELECT 속성이름 FROM 테이블이름 WHERE 조건;

인사기록

사번	부서	직위	급여	수당	점수
20190001	전산팀	대리	3000000	500000	95
20190003	경리팀	대리	3000000	200000	65
20190005	전산팀	대리	3000000	150000	60

[실습] 다음과 같이 직접 실습해서 확인해 보시기 바랍니다.



사번	부서	직위	급여	수당	점수
20190001	전산팀	대리	3000000	500000	95
20190003	경리팀	대리	3000000	200000	65
20190005	전산팀	대리	3000000	150000	60
NULL	NULL	NULL	NULL	NULL	NULL

데이터 검색하기

4.3 테이블에서 수식을 이용하여 검색하기

수식을 이용하여

급여가 4000000원 이상인 데이터만

검색하려면?

인사기록

사번	부서	직위	급여	수당	점수
20190001	전산팀	대리	3000000	500000	95
20190002	인사팀	부장	5000000	300000	83
20190003	경리팀	대리	3000000	200000	65
20190004	마케팅	과장	4000000	250000	72
20190005	전산팀	대리	3000000	150000	60

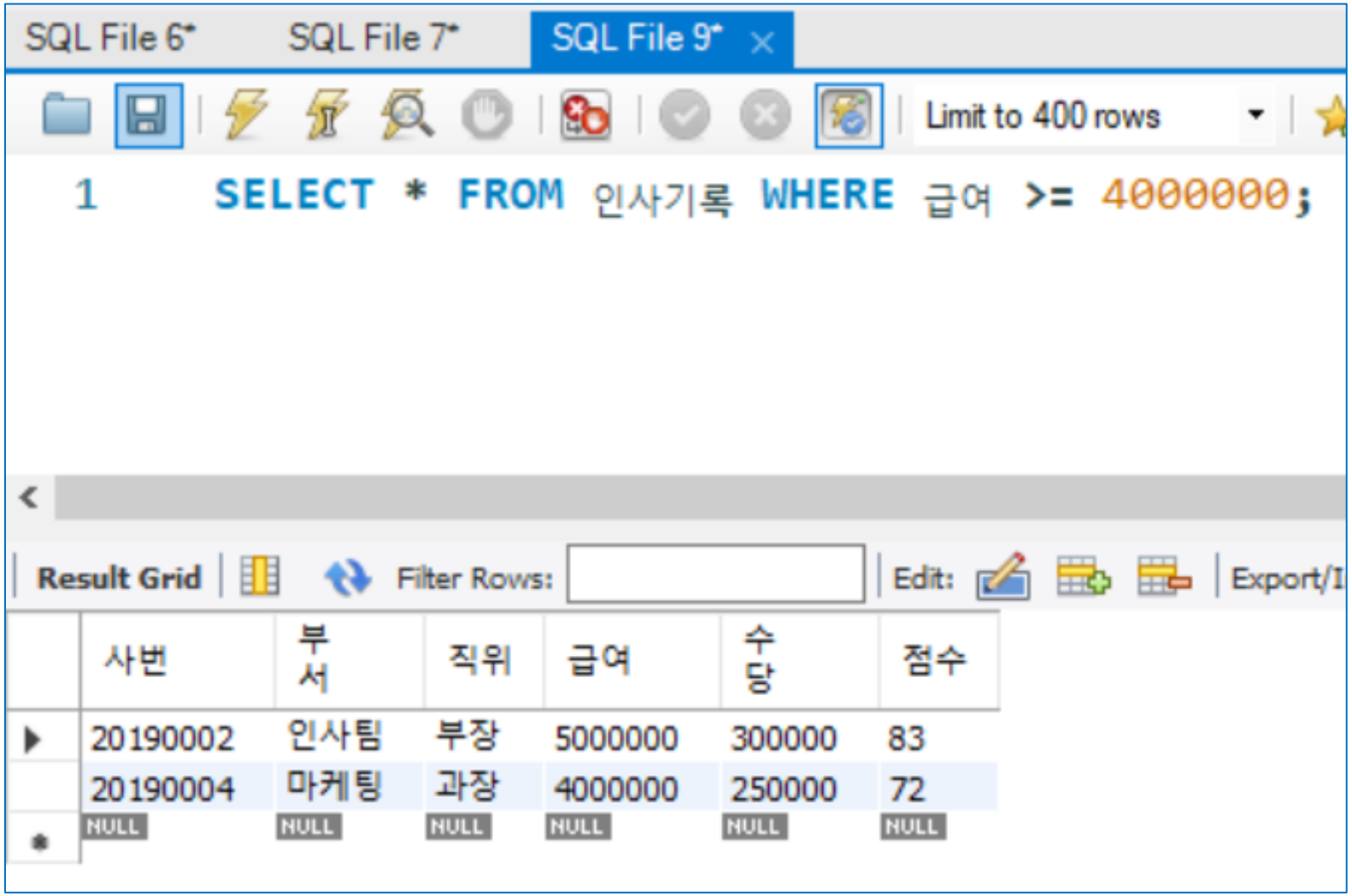
```
SELECT * FROM 인사기록 WHERE 급여 >= 4000000;
```

형식 : SELECT 속성이름 FROM 테이블이름 WHERE 수식;

인사기록

사번	부서	직위	급여	수당	점수
20190002	인사팀	부장	5000000	300000	83
20190004	마케팅	과장	4000000	250000	72

[실습] 다음과 같이 직접 실습해서 확인해 보시기 바랍니다.



사번	부서	직위	급여	수당	점수
20190002	인사팀	부장	5000000	300000	83
20190004	마케팅	과장	4000000	250000	72
NULL	NULL	NULL	NULL	NULL	NULL

데이터 수정하기

4. 인사기록 테이블에 데이터 수정하기

사번이 20190001인 자료를 찾아
부서를 인사팀으로, 직위를 과장으로
데이터를 수정하려면?

인사기록

사번	부서	직위	급여	수당	점수
20190001	전산팀	대리	3000000	500000	95
20190002	인사팀	부장	5000000	300000	83
20190003	경리팀	대리	3000000	200000	65
20190004	마케팅	과장	4000000	250000	72
20190005	전산팀	대리	3000000	150000	60

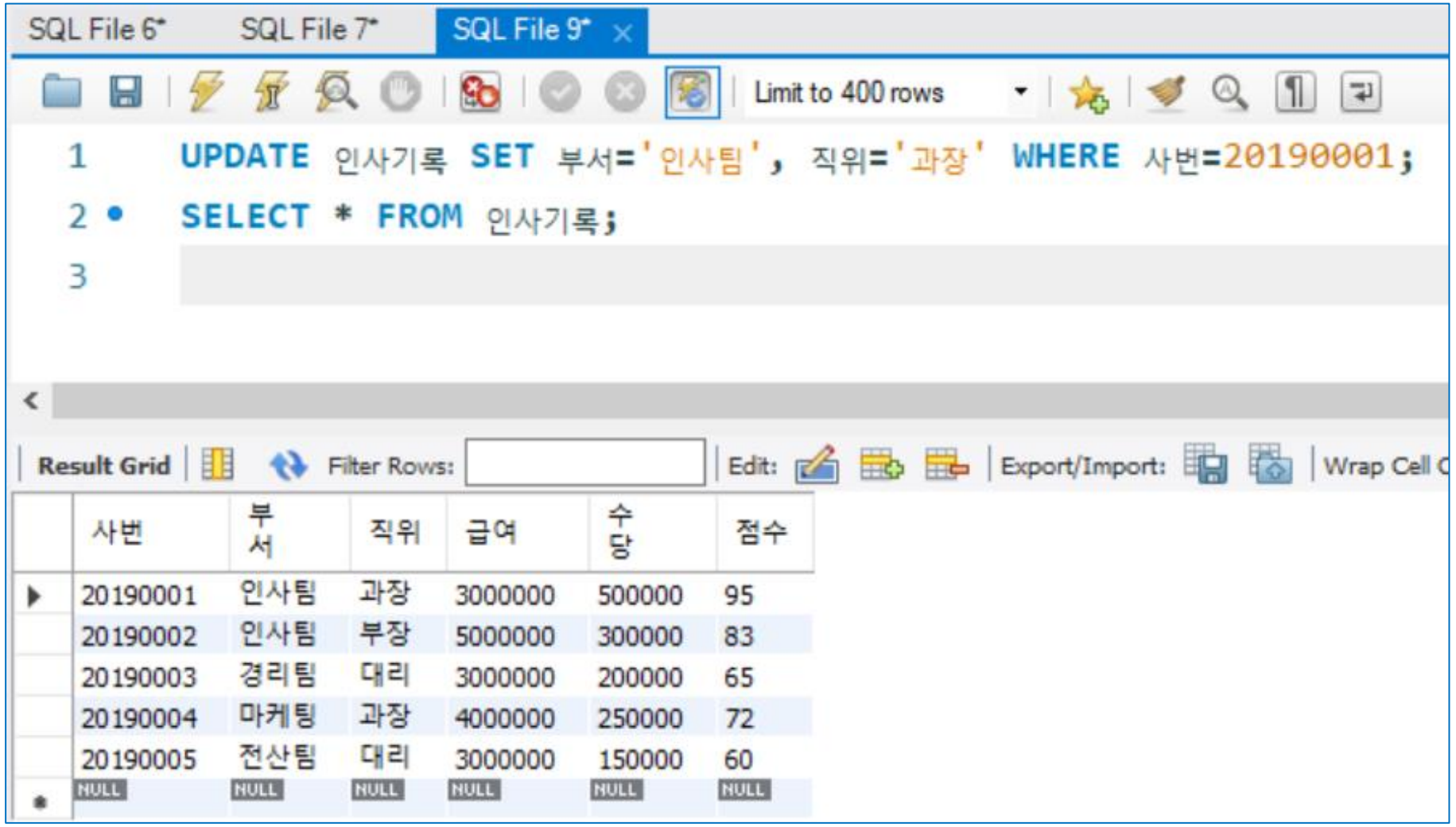
```
UPDATE 인사기록 SET 부서='인사팀', 직위='과장' WHERE 사번=20190001;
```

형식 : UPDATE 테이블이름 SET 속성명=변경할 내용 WHERE 조건;

인사기록

사번	부서	직위	급여	수당	점수
20190001	인사팀	과장	3000000	500000	95

[실습] 다음과 같이 직접 실습해서 확인해 보시기 바랍니다.



사번	부서	직위	급여	수당	점수
20190001	인사팀	과장	3000000	500000	95
20190002	인사팀	부장	5000000	300000	83
20190003	경리팀	대리	3000000	200000	65
20190004	마케팅	과장	4000000	250000	72
20190005	전산팀	대리	3000000	150000	60
NULL	NULL	NULL	NULL	NULL	NULL

데이터 삭제하기

5. 인사기록 테이블에 데이터 삭제하기

사번이 20190001인 자료를 찾아

데이터를 삭제하려면?

인사기록

사번	부서	직위	급여	수당	점수
20190001	전산팀	대리	3000000	500000	95
20190002	인사팀	부장	5000000	300000	83
20190003	경리팀	대리	3000000	200000	65
20190004	마케팅	과장	4000000	250000	72
20190005	전산팀	대리	3000000	150000	60

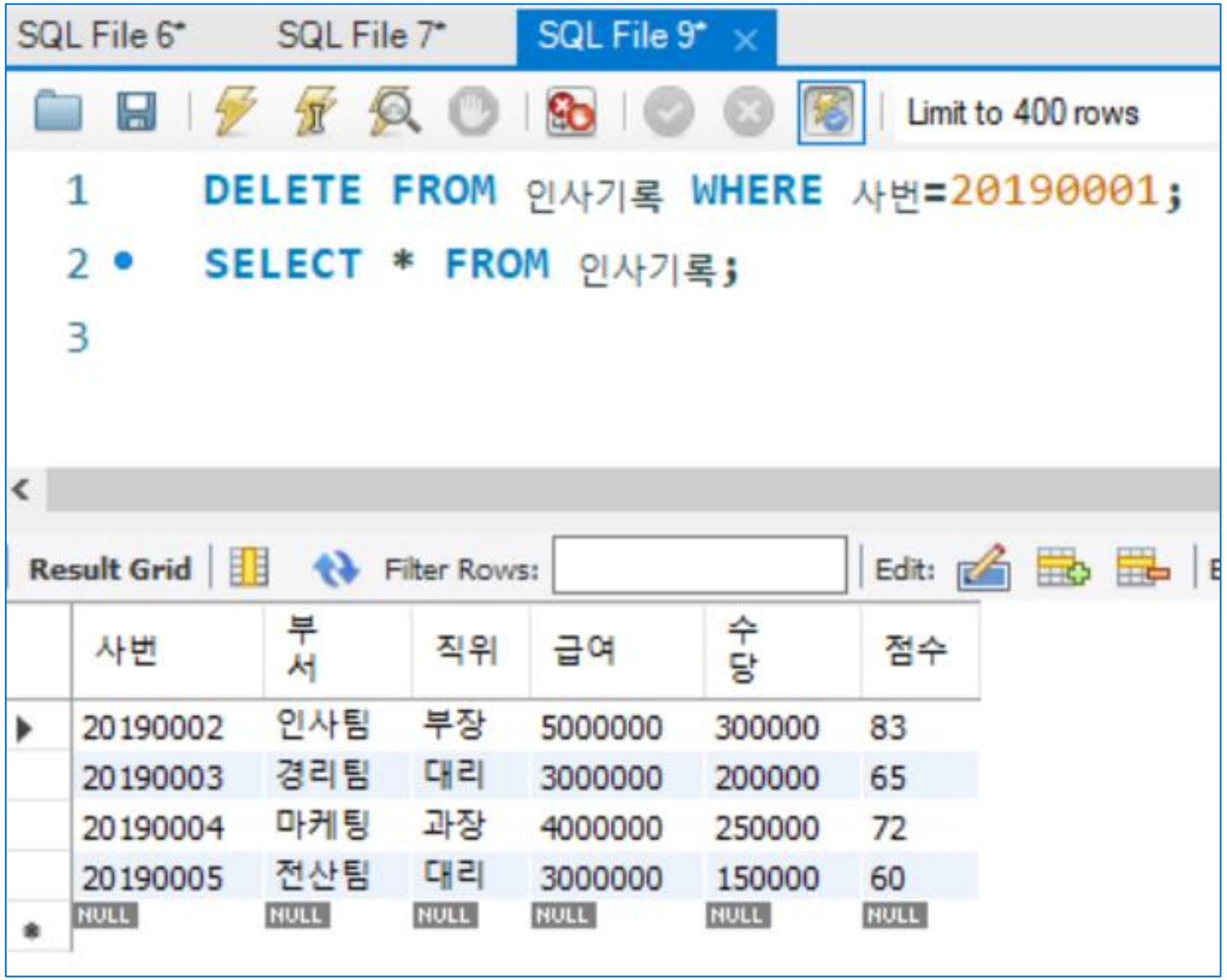
```
DELETE FROM 인사기록 WHERE 사번=20190001;
```

형식 : DELETE FROM 테이블이름 WHERE 조건;

인사기록

사번	부서	직위	급여	수당	점수
20190002	인사팀	부장	5000000	300000	83
20190003	경리팀	대리	3000000	200000	65
20190004	마케팅	과장	4000000	250000	72
20190005	전산팀	대리	3000000	150000	60

[실습] 다음과 같이 직접 실습해서 확인해 보시기 바랍니다.



사번	부서	직위	급여	수당	점수
20190002	인사팀	부장	5000000	300000	83
20190003	경리팀	대리	3000000	200000	65
20190004	마케팅	과장	4000000	250000	72
20190005	전산팀	대리	3000000	150000	60
NULL	NULL	NULL	NULL	NULL	NULL

예제) SQL 연습

1) 고객 테이블에서 등급을 검색하시오.

```
SELECT 등급 FROM 고객;
```

고객

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

결과

등급
gold
vip
gold
silver

예제) SQL 연습

2) 고객 테이블에서 등급이 gold인 투플(행)을 검색하시오.

```
SELECT * FROM 고객 WHERE 등급='gold' ;
```

고객

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

결과

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
carrot	원유선	28	gold	교사	4500

예제) SQL 연습

문제3) 고객 테이블에서 등급이 gold이고, 적립금이 2000원 이상인 투플(행)을 검색하시오.

```
SELECT * FROM 고객 WHERE 등급='gold' and 적립금>=2000 ;
```

고객

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

결과

고객아이디	고객이름	나이	등급	직업	적립금
carrot	원유선	28	gold	교사	4500

```
SELECT * FROM 고객 WHERE 등급='gold' or 적립금>=2000 ;
```

예제) SQL 연습

문제4) 고객 테이블에서 고객이름, 등급, 적립금을 검색하시오.

```
SELECT 고객이름, 등급, 적립금 FROM 고객;
```

고객

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

결과

고객이름	등급	적립금
정소화	gold	1000
김선우	vip	2500
고명석	gold	4500
김용욱	silver	0

JOIN

조인(join)

- **두 테이블**로부터 **연관된 투플(행)들을 결합하는 연산자**
- 관계 데이터베이스에서 **두 개 이상의 테이블 관계**를 **다루는 연산자**

○ 환자정보

이름	주민번호
홍길동	970405-1201234
박나래	990101-1245667
김철수	981225-2412234

○ 진료정보

주민번호	진단
970405-1201234	식중독
981225-2412234	장염
990101-1245667	감기

이름	진단
홍길동	식중독
박나래	감기
김철수	장염

조인(join)

○ 환자정보

이름	주민번호
홍길동	970405-1201234
박나래	990101-1245667
김철수	981225-2412234

○ 진료정보

주민번호	진단
970405-1201234	식중독
981225-2412234	장염
990101-1245667	감기

Equi Join (동등 조인)

```
SELECT * FROM 환자정보, 진료정보 WHERE 환자정보.주민번호=진료정보.주민번호;
```

이름	주민번호	주민번호	진단
홍길동	970405-1201234	970405-1201234	식중독
박나래	990101-1245667	990101-1245667	감기
김철수	981225-2412234	981225-2412234	장염

조인(join)

○ 환자정보

이름	주민번호
홍길동	970405-1201234
박나래	990101-1245667
김철수	981225-2412234
김한성	960202-1536512

○ 진료정보

주민번호	진단
970405-1201234	식중독
981225-2412234	장염
990101-1245667	감기

```
SELECT * FROM 환자정보, 진료정보 WHERE 환자정보.주민번호=진료정보.주민번호;
```

이름	주민번호	주민번호	진단
홍길동	970405-1201234	970405-1201234	식중독
박나래	990101-1245667	990101-1245667	감기
김철수	981225-2412234	981225-2412234	장염

조인(join)

○ 환자정보

이름	주민번호
홍길동	970405-1201234
박나래	990101-1245667
김철수	981225-2412234

○ 진료정보

주민번호	진단
970405-1201234	식중독
981225-2412234	장염
990101-1245667	감기

```
SELECT 환자정보.이름, 환자정보.주민번호, 진료정보.진단 FROM 환자정보, 진료정보
WHERE 환자정보.주민번호=진료정보.주민번호;
```

Natural Join (자연 조인)

```
SELECT 이름, 환자정보.주민번호, 진단 FROM 환자정보, 진료정보
WHERE 환자정보.주민번호=진료정보.주민번호;
```

이름	주민번호	진단
홍길동	970405-1201234	식중독
박나래	990101-1245667	감기
김철수	981225-2412234	장염

※ 이 부분은 그냥 참고로 한번 읽어만 보고 넘어가시면 됩니다.

- Natural Join (자연 조인)

SELECT 이름, 환자정보.주민번호, 진단 FROM 환자정보, 진료정보 WHERE 환자정보.주민번호 = 진료정보.주민번호;

※ 참고로 위의 쿼리는 다음과 같이 작성하는 것도 가능 (참고로만 알아두세요.)

SELECT * FROM 환자정보 natural join 진료정보;

조인(join)

○ 환자정보

이름	주민번호
홍길동	970405-1201234
박나래	990101-1245667
김철수	981225-2412234

○ 진료정보

주민번호	진단
970405-1201234	식중독
981225-2412234	장염
990101-1245667	감기

```
SELECT 이름, 진단 FROM 환자정보, 진료정보 WHERE 환자정보.주민번호=진료정보.주민번호;
```

이름	진단
홍길동	식중독
박나래	감기
김철수	장염

```
SELECT 이름, 진단 FROM 환자정보, 진료정보
    WHERE 환자정보.주민번호=진료정보.주민번호 and 이름='박나래';
```

이름	진단
박나래	감기

조인(join)

○ 환자정보

이름	환자주민번호
홍길동	970405-1201234
박나래	990101-1245667
김철수	981225-2412234

○ 진료정보

주민번호	진단
970405-1201234	식중독
981225-2412234	장염
990101-1245667	감기

```
SELECT 이름, 진료정보.주민번호, 진단 FROM 환자정보, 진료정보 WHERE 환자정보.환자주민번호=진료정보.주민번호;
```

```
SELECT 이름, 주민번호, 진단 FROM 환자정보, 진료정보 WHERE 환자주민번호=주민번호;
```

이름	주민번호	진단
홍길동	970405-1201234	식중독
박나래	990101-1245667	감기
김철수	981225-2412234	장염

비교 연산자 및 논리 연산자 정리

- 비교 연산자 : =, <>, >, <, >=, <=

- 논리 연산자 : and, or, not

[조인 실습]

올려드린 'SQL 실습' 파일을 참고해서

MySQL에서 직접 실습해 보세요~ ^^

T h a n k y o u

TECHNOLOGY

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Velit ex
ullicabo ipsum, labore sed tempora ratione asperiores des
cipserat bore sed tempora rati jgert one bore sed tem!