2019-07-02 SQL 실습

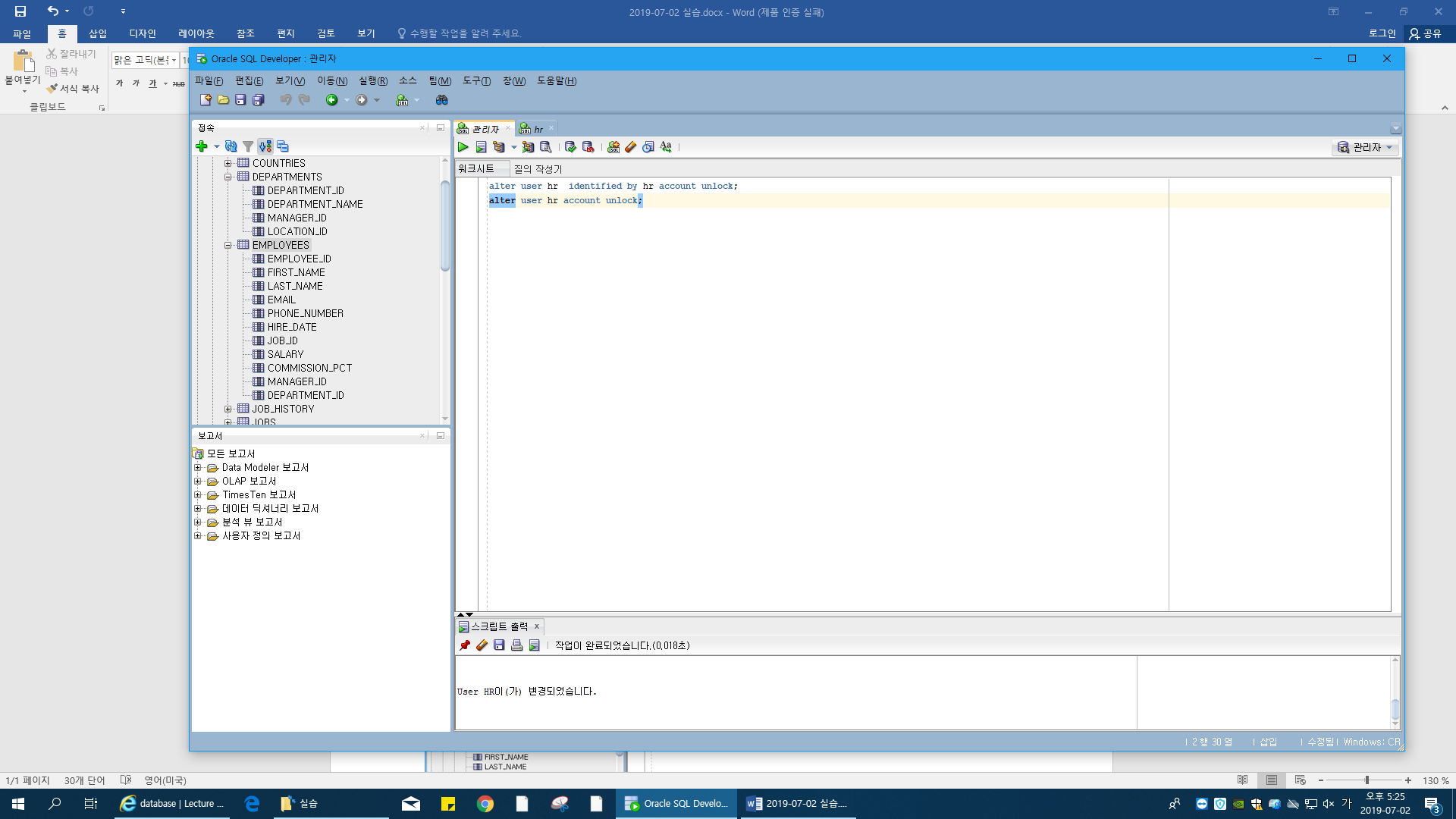
Cmd 실행

HR = id pw = hr

1.

alter user hr identified by hr account unlock;

alter user hr account unlock;



USER LOCK AND UNLOCK

2.

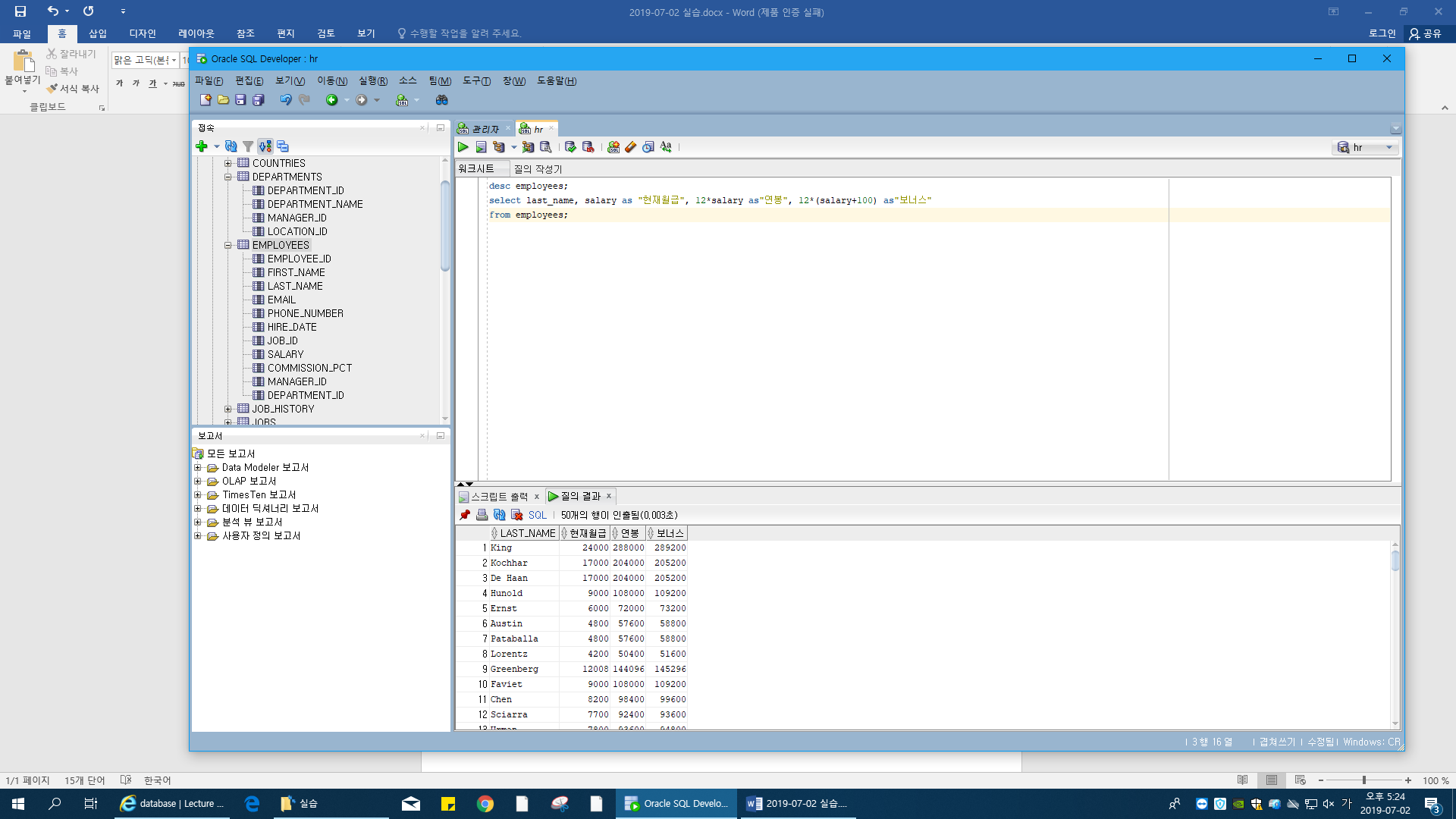
desc employees; ->EMPLOYEES LIST를 불러와라

select last\_name, salary as "현재월급", 12\*salary as"연봉", 12\*(salary+100) as"보너스"

->성, 현재월급, 연봉, 보너스만 보여줘라

from employees;

->EMPLOYEES 에서.



2019-07-03 SQL 실습

Select 구문을이용해서 Hr 계정이 가지고 있는 테이블로부터 작업을 시작함

교재에서는 테이블을 먼저 만들고 시작함

1.

도움될만한 사이트

<https://wikidocs.net/3920> ->예제를 들어서 실습할 수 있게 도와줌 많이 참고해라

<https://www.data.go.kr/>

<https://kbig.kr/portal/;jsessionid=0C7BDF04E8B33C8A800CD4B4E61AC448>

<http://www.dbguide.net/index.db>

<https://swcodingschool.github.io/database/03-%EA%B8%B0%EC%B4%88SELECT%EB%AC%B8%EC%9E%91%EC%84%B1.html>

<https://limkydev.tistory.com/> 이해하기 쉽게 포스팅한블로그

2.

SQL 실습

계정 HR, 스키마가 어떻게 이루어져있는지 확인 해당 테이블의 각각 항목은 스키마

1. Countries : 국가, 지역 등의 정보. Region값을 참조한다. (포린 키 참조)
2. Departments :
3. Employees : 아이디, 성, 이름, 이메일, 전화번호, 입사일, 해당부서 아이디 등…
4. Job History : 직원아이디, 시작일, 퇴사일, 직급, 부서
5. Jobs : 직종id(부서), 직급, 최소월급, 최대월급
6. Locations : 여러 국가에 위치한…주소
7. Regions : 대륙

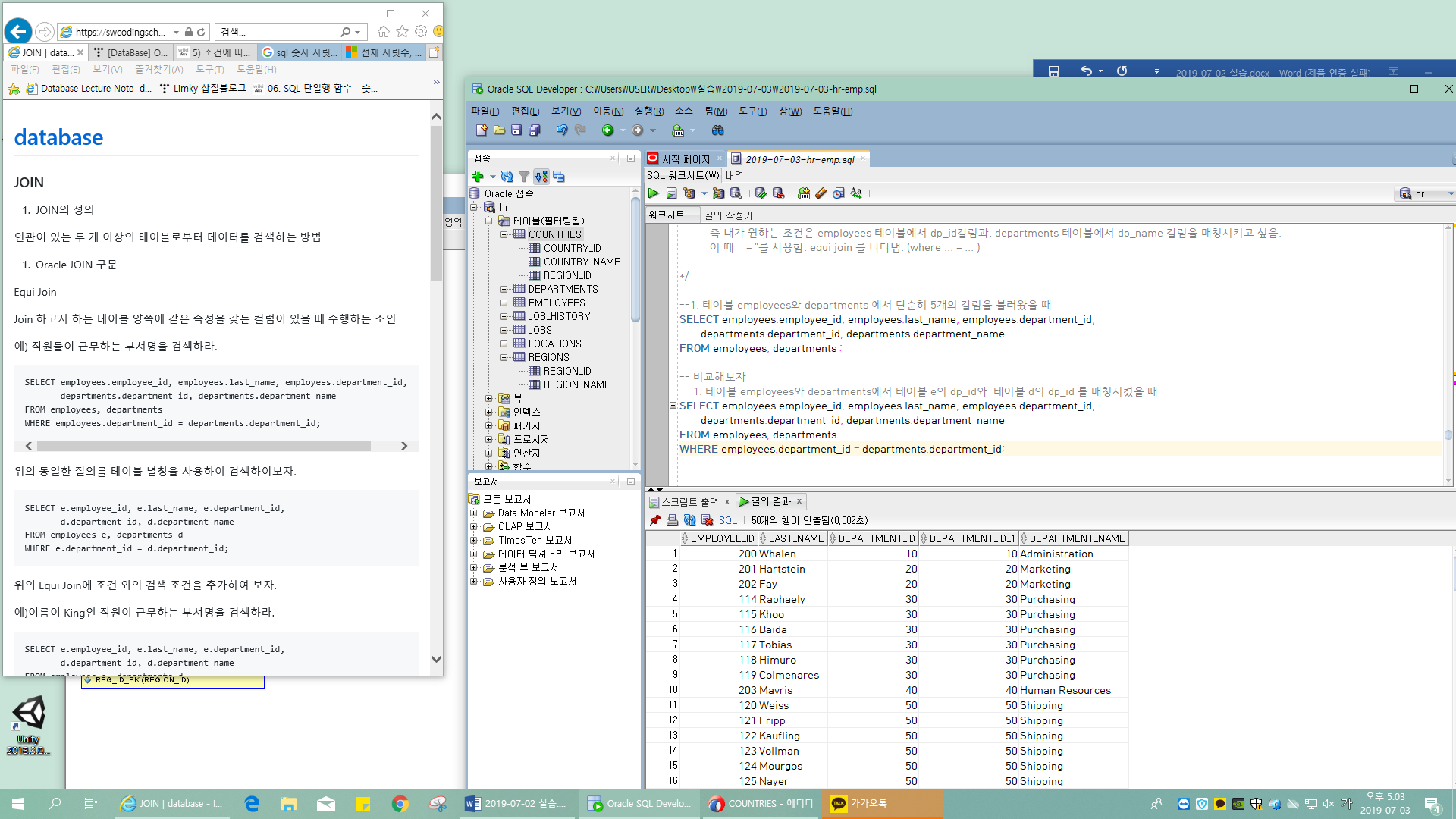
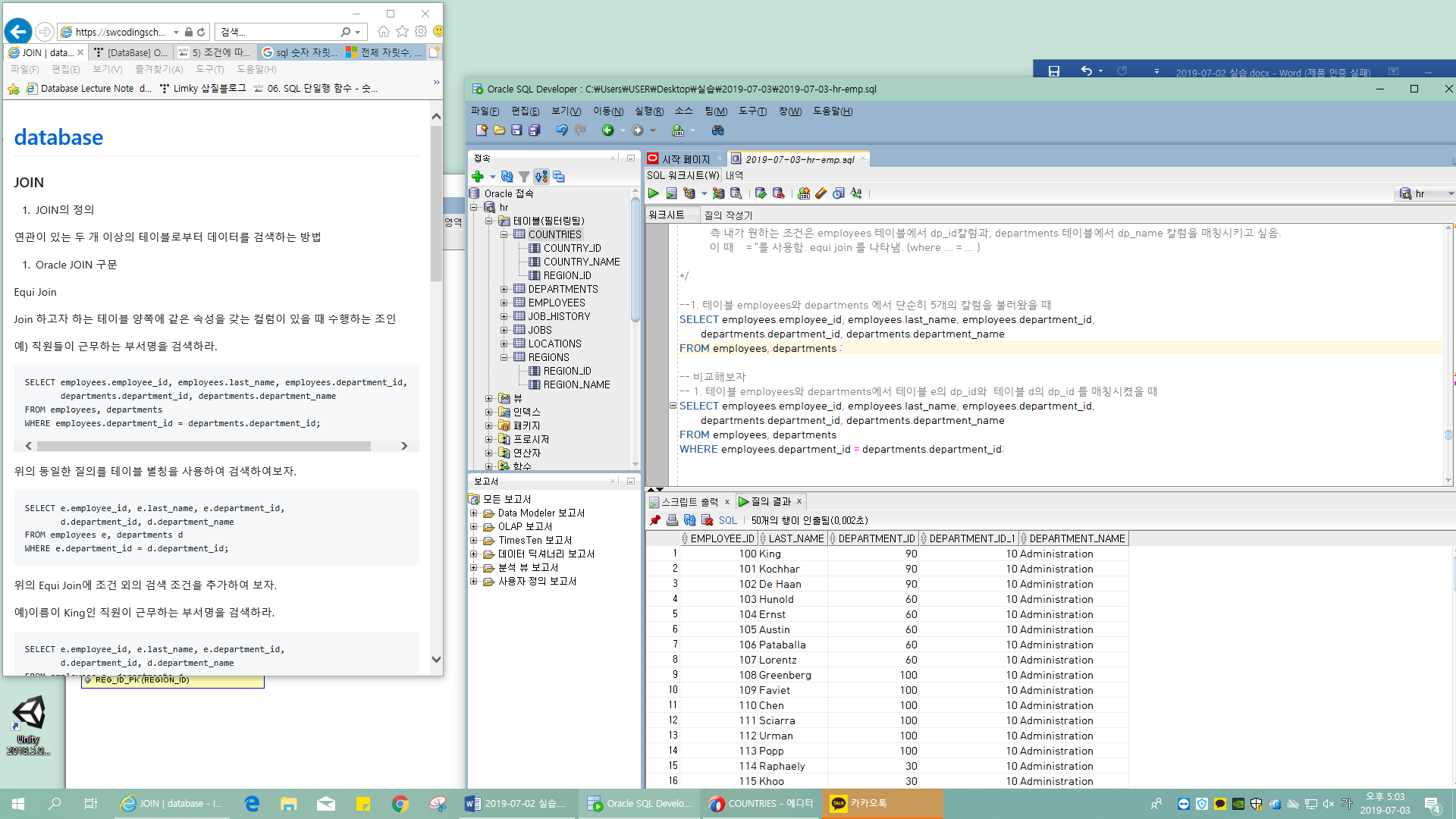
3. 주석달기

워크시트 안에서

빼기2개 연속으로 쓰면 주석처리가 되고, 출력에는 반영이 안됨.

한 줄 사용시 --,

두 줄 이상 사용시 /\* 로 시작해 \*/로 끝내주면 됨.



--주석을 한 줄로만 달 때에는 -- 를 먼저 쓰고, 주석내용을 입력하면 됨. 따로 세미콜론이나 --로 마무리할 필요 없음.

select \*

from employees;

/\*

주석을 두 줄 이상으로 쓰고 싶을 때에는 "/\*"로 먼저 시작하고

내용을 입력한 뒤에 메모(주석)를 끝낼 때 반대로 닫아주면 됨--> \*/

--1. employee\_id, first\_name, salary 값을 불러오고 싶다.

select employee\_id, first\_name, salary

from employees;

--2.

select first\_name, salary, salary+100

from employees;

--3. 급여가 많은 순으로, 이름과 급여를, 조회하고 싶을 때

select first\_name, salary, salary+100

from employees

order by salary, salary+100;

--3-1. 연습

select last\_name, salary

from employees

order by salary;

--4. 실습 : 이름, 연봉+100, 급여+100 을 했을 때 연봉계산

select last\_name, 12\*salary+100, 12\*(salary+100)

from employees;

--5. 산술연산에서 null 값이 있을 때. 커미션 퍼센트가 null값도 있고 0.4인 사람도 있고...

select last\_name, commission\_pct

from employees;

--5-1. 산술연산에서 하나라도 null값이 있다면 null값으로 나온다.

-- 즉, 커미션퍼센트가 없는(해당안되는) 사람들은 null 값을 가지고 있을 때, 그 결과는 무조건 null.

select last\_name, salary, commission\_pct, 12\*salary\*commission\_pct

from employees;

--6. 컬럼의 별칭을 사용해보자. 원래 테이블 안에서 스키마(컬럼)가 lastname 이라면, 출력에 name 으로 표기하게 하는 방법

select last\_name as name

from employees;

--7. 연결연산자

-- (1) 성하고 이름을 붙여서 띄어쓰기 없이 나타내고 싶어요

select last\_name || first\_name

from employees;

-- (2) 성하고 이름을 띄어서 한 번에 나타내고 싶어요

select last\_name || ' ' || first\_name

from employees;

--(3) 성하고 이름을 띄어서 한 번에 나타내고 싶어요, 그런데 컬럼네임은 깔끔하게 name 으로 나타내고 싶어요.

select last\_name || ' ' || first\_name as name

from employees;

--(4) 김사탄님은 부서99에서 근무합니다.

select last\_name || '님은 부서' || department\_id || '에서 근무합니다.' as "근무부서확인"

from employees;

--8. 중복제거 distinct : 중복된 값이 많아요. 중복된 값을 제거하고 싶어요.

--(5) 근무부서만 보이고 싶어요

select department\_id

from employees;

-- (이 경우 그럼 부서가 개별부서 한 개씩만 나타남)

select distinct department\_id

from employees;

-- 9. 조건을 달고 싶어요. where

-- (1) 부서 90에 누가있는지 알고싶어요. 부서 90만 나타나게 할 때.

select employee\_id, last\_name, department\_id

from employees

where department\_id = 90 ;

-- (2) 대 소문자가 구별이 되어야한다. King 인 사람인데 소문자 king으로 검색하면 안 나옴.

select employee\_id, job\_id

from employees

where last\_name = 'king';

--(3) 대소문자 구별, 대문자로 검색했을 때

select employee\_id, job\_id

from employees

where last\_name = 'King';

--10. Like

-- lastname 중 두번째 글자가 모두 a였으면 좋겠어

select employee\_id, last\_name

from employees

where last\_name like '\_a%';

--11. 비교연산자

-- 월급 2500이상 3000이하를 부등호로 표현해 나타낼 때

select employee\_id, last\_name, salary

from employees

where salary >= 2500 and salary <= 3000 ;

-- 부등호를 대체할 수 있는 연산자 : between

select employee\_id, last\_name, salary

from employees

where salary between 2500 and 3000 ;

-- 연산자 in : 이에 해당하는 조건만 검색하고 싶다. 딱 지정해줘서 나타내게 할 수 있음

select employee\_id, last\_name, salary

from employees

where salary in (2400, 2500, 3000);

-- 연산자 or : a이거나 b거나 c 일때

select employee\_id, last\_name, salary

from employees

where salary=2400 or salary=2500 or salary=3000;

-- 연산자 is null : 값이 null인 값만 불러오기

select employee\_id, last\_name, salary, commission\_pct

from employees

where commission\_pct is null;

--(2) 논리연산자 사용의 예

----AND

-- 샐러리가 8천보다 크고

--이름에 k가 들어간 사람을 검색해라

SELECT employee\_id, last\_name, salary

FROM employees

WHERE salary > 8000

AND last\_name LIKE '%K%';

--(3) 내림차순 정렬

SELECT employee\_id, last\_name, salary

FROM employees

ORDER BY salary DESC;

-- 컬럼의 별칭을 통한 정렬, as를 뺄 수 있음.

SELECT employee\_id, last\_name name

FROM employees

ORDER BY name;

-- ---------> order by name에서 last\_name이 아니라 "name"이냐면, select에서 last\_name name으로 별칭을 붙여줬기 때문

-- 부서아이디, 이름, 사원id가 제시되었을 때, department id 기준으로, 월급순으로 표기할 때

-- 실생활 적용 : 주문량이 많거나 판매인기순대로 정렬할 때 (인기많은상품순으로 정렬 )

select employee\_id, last\_name, department\_id, salary

from employees

order by department\_id, salary desc, last\_name;

--함수 : 대문자를 소문자로 나타내는 함수 : lower

select lower(last\_name), last\_name

from employees;

--문자함수

--concat(컬럼1,컬럼2)

select concat(first\_name, last\_name), first\_name, last\_name

from employees;

--lpad -> 총 10자리 숫자 중 부족한 단위수는 \*로 표시한다.

select last\_name, lpad(salary, 10, '\*')

from employees;

--trim -> 지정된 문자를 삭제

--입사일 날짜.시간 함수

--기본 (yy-mm-dd)

select last\_name, hire\_date

from employees;

--03/06/17로 표현하는 날짜를 20030617로 만들고 싶어요.

select last\_name, hire\_date, to\_char(hire\_date, 'yyyymmdd')

from employees;

-- 홈페이지게시판을 만들 때 날짜를 표현하게 하기 위해서 쓰여요.

--변환함수

--날짜

--03/06/17 로 표기된 날짜를 2003 6월 화 로 표기할 때

select last\_name, hire\_date, to\_char(hire\_date, 'yyyy mon dy ')

from employees;

--03/06/17 로 표기된 날짜를 2003 6월 17 화요일 로 표기할 때

select last\_name, hire\_date, to\_char(hire\_date, 'yyyy mon dd day ')

from employees;

--03/06/17 로 표기된 날짜를 tow thousand three 6월 17 화요일 로 표기할 때

select last\_name, hire\_date, to\_char(hire\_date, 'year mon dd day ')

from employees;

--일반 함수

--nvl 함수 : null값을 다른값으로 대치해연산하거나 다른 값으로 출력하는 함수

--예시 : manager\_id가 숫자값이었으니, 대체값도 숫자값으로 입력해야함. 문자값으로 넣으면 출력오류뜸.

select last\_name, nvl(manager\_id, '0000')

from employees;

--- nvl2 는 왜 문자처리가 가능하지??????왜????

SELECT employee\_id, last\_name, NVL2(manager\_id, '부서있음', '부서없음') salary

FROM employees;

--

--nullif a와 b의 값이 같으면 null, 다르면 a값 출력

-- 이 경우에는 ellen이 5글자라서 lenghth(first or last\_name) (글자개수세기) 에서 글자개수가 같지 않아 first name인 첫번째 조건 출력.

SELECT first\_name, last\_name, NULLIF(LENGTH(first\_name), LENGTH(last\_name))name

FROM employees ;

--자료의 일부분만 확인해 처리할 때

--첫번째페이지에 보여주는 레코드 개수 제한할 때: 글목록 정리에서 10개만 보이게 할 때, 15개만 보이게 할 때... 등

select last\_name, first\_name

from employees

where rownum <=10;

--★★★ 아 근데 테이블에 있는 내용말고 그냥 숫자나 문자 코딩 연습해보고 싶은데 그럴 땐???? from dual; 을 쓰면 된당

select 5\*4 from dual;

--그룹함수 : 통계와 관련된 함수. 평균 행의 수, 최대 최소값 표준편차 합계 분산 등을 계산해줌

--

--

select avg(salary), sum(salary), max(salary), min(salary)

from employees;

--최근 입사날짜, 최대 오래된 입사날짜

select max(hire\_date), min(hire\_date)

from employees;

select count(commission\_pct)

from employees

where department\_id = 60;

--그룹함수의 null값이 포함되지 않은 경우

select avg(commission\_pct)

from employees;

--null값을 포함시키는 경우

select avg (nvl (commission\_pct, 0))

from employees;

--group by : 직종별 평균급여

select job\_id, avg(salary)

from employees

group by job\_id ;

-- group by : 부서별 급여 : 근데 이러면 소숫점까지 나옴

select department\_id, avg(salary)

from employees

group by department\_id;

-- 문자 처리해서

select department\_id, to\_char(avg(salary), 9999.99)

from employees

group by department\_id;

-- 숫자로 소수점 3번째자리에서 반올림 처리한다면

-- 함수 안에 함수처리를 하면 됨.

select department\_id, round(avg(salary), 3)

from employees

group by department\_id;

-- 함수 안에 함수를 넣어서 코딩하는 방법밖에없음

-- 숫자를 천의 자리 단위로 정리해서 소수점 3번째 자리에서 반올림

select department\_id, to\_char( round (avg(salary), 3),

'999,999.999'

)

from employees

group by department\_id;

--

--잘못된 group by절 사용

select job\_id, avg(salary)

from employees;

-- ---> 이 경우 오류가 나옴. 왜? 샐러리를 평균냈는데 평균낼 표본이 없음.(그룹이 없으니까)

-- (1) 샐러리 평균만 보거나

select avg(salary)

from employees;

--(2) group by job\_id;를 넣거나

select job\_id, avg(salary)

from employees

group by job\_id;

-- group 에 대한 제한을 where절에 제시한 경우

select job\_id, avg(salary)

from employees

where avg(salary) > 4000

group by job id;

-- >.>>>>오류 : 왜? 조건을 쓸 때, group by는 having을 뒤에 쓸 수 잇음. group by는 where은 안쓰임★

--순서 : (1) select (2) from (3) group by (4) having... ;

SELECT job\_id, AVG(salary)

FROM employees

GROUP BY job\_id

HAVING AVG(salary) > 4000 ;

--> 출력됨.

/\*

select + all | distinct | 칼럼 + 집계함수(avg, sum, min, max..)

from

where

order by

group by

having 순으로...

\*/

--Join : 두개의 테이블을 결합하는 것 자료의 불일치를 막기 위해 테이블을 구분해서 사용

/\* eg. 1. contries, 2. region 라는 테이블이 있을 때

1. contries 는 (1) Country id, (2) name (3) region id 으로 구분되어있음

(1) C id 에는 ar, au, be, br...zw 까지 총 25개가 존재함 C id는 C name의 앞 두글자만 따서 만든 것임

(2) C name에는 argentina, Australia, Belgium... 등 25개가 있음. C id는 C name의 앞 두글자만 따서 만든 것임

(3) R id 는 1, 2, 3, 4로 분류되어 있음. 대륙으로 분류한듯함

eg: america 의 region id는 2임 그런데 똑같은 america임에도 불구하고 America로 대문자로 시작하는 경우

불필요하게 2개가 뜨고 분류가 어려워짐. 이 경우 region id 를 사용해 코드를 부여함. : 이 작업을 join 말함 .

즉 같은 속성이고 같은 값을 가질 때 결합을 시켜준다 .

서로 다른 두 개의 테이블 안에서 공통값을 찾아내 묶어주는 것

eg : 1. 동등조인(equi join) = 김사탄과 호랑이는 같은 A 수업을 듣는다. 이들을 'A과목 수강생들' 이라고 부를 수 있겠다.

2. cross join =

eg2 : 2.america, 3.asia 일 때 ?

2. regions 에는 (1)id (2)name 으로 구분되어있음 즉

region name : 1=europe 2=americas 3=asia 4=middle east and africa 로 구분되어 있음

내부적으로 두 개의 테이블을 결합시킬 때 한 행과 타 테이블 한 행과 한 번씩 matching 시키면서 레코드를 만들어낸다

그 방법을 cross 방법이라고 함. 경우의 수 를 생각하면 됨.

3. 그럼 여기서 매칭될 수 있는 것은 공통부분인 R id 인데, 테이블 1인 Country에도 R id 값이 4개,

테이블 2인 Region 에도 R id 값이 4개임.

크게 4개의 대분류인 Region Id로 매칭을 시키면 1에 해당하는 나라, 2에 해당하는 나라, 3에 해당하는 나라, 4에 해당하는 나라를 찾을 수 있음.

예를 들면 R 에서 1 = asia 인 경우

C 에서 1 = Japan, Korea, China 인경우

JOIN = 1 = ASIA = KOREA, CHINA, JAPAN 라는 공통점이 존재함.

그럼 1 ASIA KOREA

1 ASIA CHINA

1 ASIA JAPAN 이라는 새로운 JOIN TABLE 이 만들어짐~

\*/

--

select \*

from countries, regions;

--> 이 경우 모든 자료가조회됨

-- 우리는 여기서 속성이 같은 데이터를 찾아 지정해주는 작업이 필요함 . : 즉 우리가 조건을 부여해서 분류하는 작업이 필요함 .

-- 이렇게 하면 모든 경우의 수 즉 조건을 부여하기 전에 매칭된 모든 값이 나옴, 구럼 25\*4 니까 100개 ~~

select c.country\_name, r.region\_name

from countries c, regions r;

-- 두 테이블 중 같은 region id 만 취급을 하겠다-> 컨트리테이블의 region id와 리전테이블의 region id의 값이 일치할 때"만"의 결과를 보고싶다.

select c.country\_name, r.region\_name

from countries c, regions r

where c.region\_id = r.region\_id;

-- --->결과 : KOREA ASIA

JAPAN ASIA

INDIA ASIA....

/\* 두 개의 레코드를 가지고 결합 후 join 한다는 것은??

전제조건 1. A테이블과 B테이블이 존재해야한다.

2. 내가 원하는 조건을 선택을 해야한다

1) Equi Join A 테이블과 B 테이블 값에서 JOIN시 공통인 것들만 찾는다 .

2) Left Join A 테이블과 B 테이블 값에서 JOIN시 A테이블의 모두의 것과, A테이블과 B테이블의 공통인 것만

3) Right Join A테이블과 B 테이블 값에서 JOIN시 B테이블의 모두의 것과, A테이블과 B테이블의 공통인 것만

\*/

-- JOIN 실습

/\*

전제조건 1. 테이블 : 2개이다.(혹은 3개 이상일 수도 있으나 다른 join방법 사용) eg (1) employees (2)departments

2. 조건 : 내가 원하는 조건이 무엇인가?

3. select - from - where 순으로 처리한다.

\*/

/\* Equi Join

1. 테이블이 2개여야함 : (1) employees (2)departmens

2. 필요한 칼럼 id를 select 한다. : employees 에서 선택할 칼럼 (1) e.id, (2) e.last\_name, (3) e.dp\_id

departments에서 선택할 칼럼 (1) dp.dp\_id, (2)dp.name

총 칼럼 5개

3. 불러올 테이블 지정 : from employees, departments

4. 내가 필요한 조건을 where 문을 이용해 지정한다 : where e.dp\_id = dp.dp\_name

즉 내가 원하는 조건은 employees 테이블에서 dp\_id칼럼과, departments 테이블에서 dp\_name 칼럼을 매칭시키고 싶음.

이 때 = "를 사용함. equi join 를 나타냄. (where ... = ... )

\*/

--1. 테이블 employees와 departments 에서 단순히 5개의 칼럼을 불러왔을 때

SELECT employees.employee\_id, employees.last\_name, employees.department\_id,

departments.department\_id, departments.department\_name

FROM employees, departments ;

-- 비교해보자

-- 1. 테이블 employees와 departments에서 테이블 e의 dp\_id와 테이블 d의 dp\_id 를 매칭시켰을 때

SELECT employees.employee\_id, employees.last\_name, employees.department\_id,

departments.department\_id, departments.department\_name

FROM employees, departments

WHERE employees.department\_id = departments.department\_id;

/\* 위의 코드를 출력해보면 e.dp\_id 값이 10일 때 d.dp\_id 값도 10이다. 20은 20, 30은 30으로 같은 값만 매칭되는 결과를 볼 수 있음.

술게임에서 얼굴에 점 있는사람 손 접어, 바지 입은 사람 손 접어 하는 거랑 똑같음)

/\*

실습 문제 : 위의 코드에서(즉 e.dp\_id = d.dp\_id 조건에서) 이름이 King 인 직원이 근무하는 부서명은 무엇일까?

KING 이라는 이름을 가진 직원이 근무하는 부서명이라는 소리는 적어도 1. 이름 2. 부서명 이라는 칼럼이 필요하다는 소리

1. 테이블이 2개여야함 : (1) employees (2)departmens

2. 필요한 칼럼 id를 select 한다. : employees 에서 선택할 칼럼 (1) e.id, (2) e.last\_name, (3) e.dp\_id

departments에서 선택할 칼럼 (1) dp.dp\_id, (2)dp.name

총 칼럼 5개 (사실 여기선 그렇게 많이 필요하진 않지만)

3. 불러올 테이블 지정 : from employees, departments 간편하게 e, d로 지정.

4. 내가 필요한 조건을 where 문을 이용해 지정한다 : where e.dp\_id = dp.dp\_name

즉 내가 원하는 조건은 employees 테이블에서 dp\_id칼럼과, departments 테이블에서 dp\_name 칼럼을 매칭시키고 싶음.

이 때 = "를 사용함. equi join 를 나타냄. (where ... = ... )

5. 마지막 조건인 KING 을 검색해야함.

이미 4에서 조건 2개를 썻으니, 조건이 3개일 때는 where a = b and b = c로 검색해야함~

\*/

SELECT e.employee\_id, e.last\_name, e.department\_id,

d.department\_id, d.department\_name

FROM employees e, departments d

WHERE e.department\_id = d.department\_id

AND last\_name = 'King';

/\* ★★★3개 이상의 N개의 테이블을 E JOIN 시켜야할 때.

이 때에는 최소 n-1 개의 조건이 필요하다는 소리. 그니까 적어도 2개 이상의 조건이필요함.

참고로 조건이 3개일 때는 where a = b and b = c로 검색해야함~

예) 직원이 근무하는 부서명과 근무하는 도시를 검색하라.

(select)칼럼: (1) employee\_id (2) last\_name (3)department\_id (4) department\_name (5) location\_id (6)city

(from) 테이블 : (1) employees , (2) departments (3) locatins 총 3개( 약어 e , d , l )

(where) e = d and d = i

\*/

SELECT e.employee\_id, e.last\_name, e.department\_id,

d.department\_name, d.location\_id, l.city

FROM employees e, departments d, locations l

WHERE e.department\_id = d.department\_id

AND d.location\_id = l.location\_id ;

-- 표준 join

-- cross join --->테이블 employees와 departments 에 둘 다 있는 애들만 출력

select last\_name, department\_name

from employees CROSS JOIN departments;

-- natural join

-- using

-- 공통된 속성이 두 개 이상의 속성이 들어가는 경우도 있으니 그 중에 내가 지정하는 값만 골라라(내가 기준을 지정해야한다)

SELECT last\_name, department\_name

FROM employees JOIN departments

USING (department\_id);

--on 절

-- 특성자체는 같지만 컬럼의 이름이 다를 때 WHERE절의 JOIN조건과 같은 기능이다 생각하면 됨 (ON조건절은 WHERE조건절의 역할을 함) but 가독성 떨어짐

/\*

실습 문제 : 위의 코드에서(즉 e.dp\_id = d.dp\_id 조건에서)

1. 테이블이 2개여야함 : (1) employees (2)departmens

2. 필요한 칼럼 id를 select 한다. : employees 에서 선택할 칼럼 (1) last\_name

departments에서 선택할 칼럼 (1) department\_name

총 칼럼 2개

3. 불러올 테이블 지정 : from employees, departments 간편하게 e, d로 지정.

4. 내가 필요한 조건을 on문으로 불러온다 e테이블과 d테이블 id가 같을 때만 출력

on(e.department\_id = d.department\_id);

\*/

SELECT e.last\_name, d.department\_name

FROM employees e JOIN departments d

ON (e.department\_id = d.department\_id );

/\*

1. 테이블이 2개여야함 : (1) employees (2)departmens

2. 필요한 칼럼 id를 select 한다. : employees 에서 선택할 칼럼 (1) last\_name

departments에서 선택할 칼럼 (1) department\_name

총 칼럼 2개

3. 불러올 테이블 지정 : from employees, departments 간편하게 e, d로 지정.

4. 내가 필요한 조건을 on문으로 불러온다 e테이블과 d테이블 id가 같을 때만 출력

on(e.department\_id = d.department\_id);

\*/

SELECT employee\_id, department\_name, city

FROM employees e

JOIN departments d

ON (e.department\_id = d.department\_id )

JOIN locations l

ON (d.location\_id = l.location\_id );

/\* outer join

즉 A, B 테이블을 JOIN 할 경우, 조건에 맞지 않는 데이터도 표시하고 싶을 때 OUTER JOIN을 사용합니다.

\*/

-- (1) Left join

SELECT e.last\_name, e.department\_id, d.department\_name

FROM employees e

LEFT OUTER JOIN departments d

ON (e.department\_id = d.department\_id );

-- (2) Right Join

SELECT e.last\_name, e.department\_id, d.department\_name

FROM employees e

RIGHT OUTER JOIN departments d

ON (e.department\_id = d.department\_id );

--(3) full outer join : left right 두 개를 다 가지고 오는 경우

SELECT e.last\_name, e.department\_id, d.department\_name

FROM employees e

FULL OUTER JOIN departments d

ON (e.department\_id = d.department\_id );

2019-07-04 목요일

테이블 스페이스 만들기 구문

1. create tablespace : 테이블 스페이스 이름
2. datafile ‘데이터파일경로’
3. size 초기사이즈
4. auotoextend on next 자동증가사이즈
5. maxsize 최대사이즈;

create tablespace madang

datafile ‘

size 50M

autoextend on next 10M

maxsize unlimited;

관리자 비번 바꾸는법

C:\Users\USER>sqlplus / as sysdba

SQL> alter user system identified by 1234;

마당만들기

사용자 계정 만들기

오류체계를 번호로 알려줌

뭘까???그코드가????? 구글링하라고함 ㅎㅎㅎ

권한부여

Grant 권한리스트 to 아이디;

Connect 로그인권한

Resource 자원사용권한 dba: db관리자 권한

Grant connect, resource, dba to madang

---테이블 생성하기

---create tanle systax

Create tanle table\_name (

Column\_1 data\_type column\_constraint,

Column\_2 data\_type column\_constraint,

… ->>>> 컬럼에 제약조건을 걸겠다는 뜻

Table\_constraint

);

Create table table\_name (

컬럼의 개수만큼 계속…

Column\_1 data\_type(길이)

Column\_2 data\_type(길이),

컬럼의 개수만큼 계속…

…

);

테이블 스키마표현하는 방법

Book(bookid, bookname, publisher, price)

가 있어야함

Create table book(

Bookid number,

1. 조건: null값 불허용 ->그럼 null값을 뭐로 지정할거인지 constraint로 지정해줄 수 있음
2. 공백(space)불가

Bookname varchar2,

Publisher varchar2,

Price number

);

실습

Create table book(

Bookid number not null,

Bookname varchar2(50) not null,

Publisher varchar2(10) not null,

Price number

Primary key(bookid) 프라이머리키는 북아이디로 쓰겠다는 뜻  
보통 프라이머리키는 맨 아랫단에 적는 경우가 많음 변경되는 경우가 많아서

Foreign key(deptid)

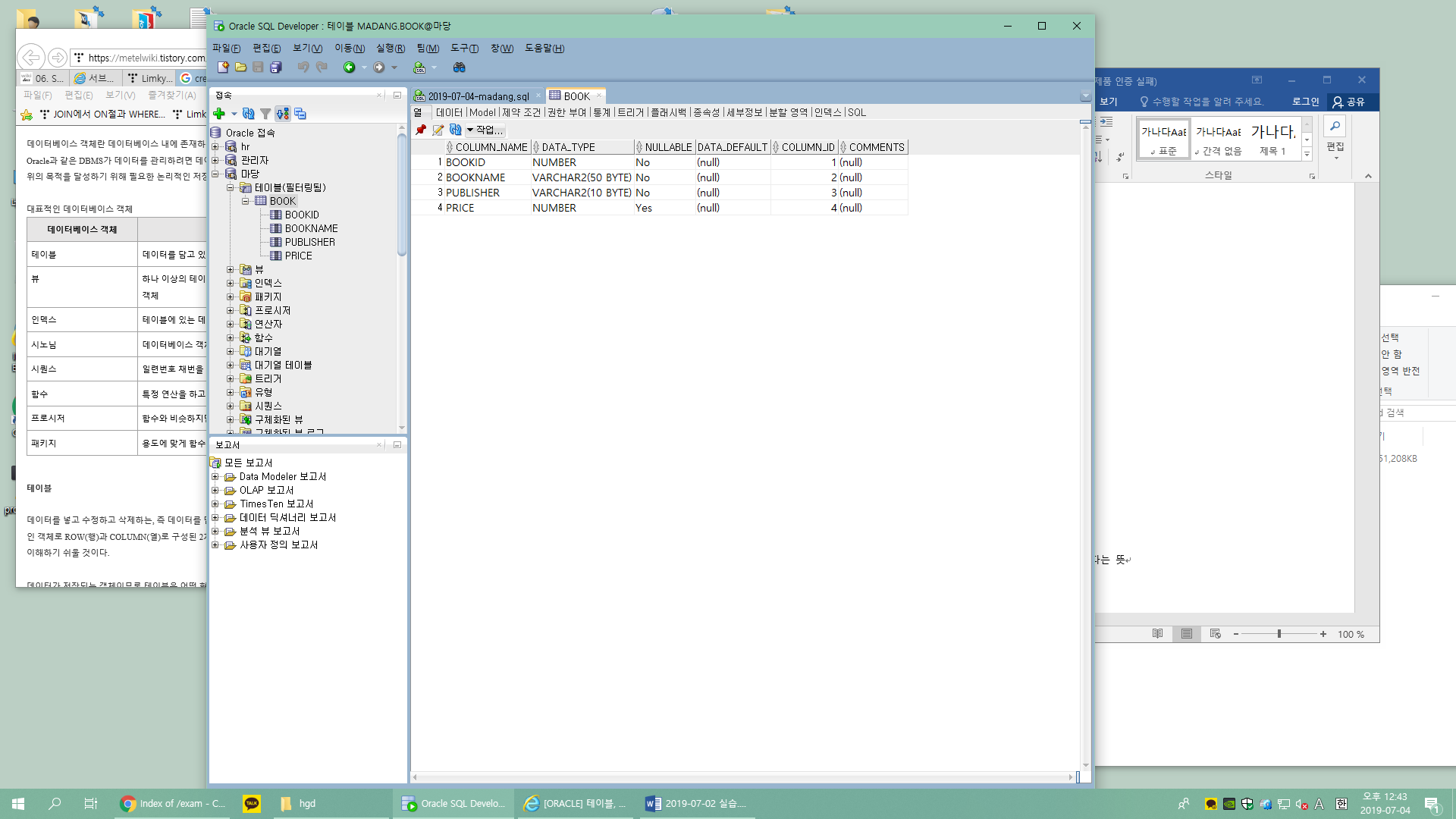
);

131쪽 178쪽 참고

insert into tablename

values( ~ 값리스트) (185쪽)

**insert** into book **values**(1, "레미제라블", "대한출판사", 14000);



Create table book(

Bookid number(3),

Bookname varchar2(50) not null,

Publisher varchar2(10) not null,

Price number(8),

primary key(bookid)

);

insert into book values(1, '레미제라블', '대한출', 14000)

insert into book(bookid, bookname, publisher)

values(2, '장발장', '한빛');

---1. 테이블 생성하기

---create table systax

Create table table\_name (

Column\_1 data\_type,

Column\_2 data\_type,

);

---2. 테이블 스키마 표현하는 방법

---Book(bookid, bookname, publisher, price) 총4개 칼럼이 잇어야함

Create table book(

Bookid number(3),

Bookname varchar2(50) not null,

Publisher varchar2(10) not null,

Price number(8),

primary key(bookid)

);

특정 레코드 삭제

Delete from book

Where bookid=2

* Bookid 2번인 책 만 삭제

insert into book(bookid, bookname, publisher)

Create table book(

Bookid number(3),

Bookname varchar2(50) not null,

Publisher varchar2(10) not null,

Price number(8),

primary key(bookid)

);

insert into book values(1, '레미제라블', '대한출', 14000);

-🡪 이거 여기서 세미콜론 안찍으면 ora-00933 sql command not properly eneded 오류뜸

insert into book(bookid, bookname, publisher)

values(2, '장발장', '한빛');

/\*

테이블 업데이트 하는 방법

update book

set price = 20000 (새로운 대체값)

이 때 where절을 안 쓰면 모든 값이 바뀌어버려~~그니까 where절로 조건을 걸어

select \*

라고 한다면, 모든 값이 20000으로 바뀌어버리니까 조건을 걸어야함. where

\*/

delete from book;

여기도 where로 조건걸 수 있음 그냥 단순히 delete from book 하면 모든 테이블 사라지니  
조건을 걸어서 사라지게 할 것만 ..

rollback;

select \* from book;

update book

set price = 20000

where bookid = 2;

select \*

from book;

---테이블삭제는 drop table

drop table book;

select \*

from book;

--🡪 이렇게 하면 테이블을 삭제했는데, 검색을 all 해버리니 오류가 남. 데이터가 없으니 검색하니까 오류가나오지~~ㄴ

Create table book(

Bookid number(3),

Bookname varchar2(50) not null,

Publisher varchar2(10) not null,

Price number(8),

primary key(bookid)

);

insert into book values(1, '레미제라블', '대한출', 14000);

insert into book(bookid, bookname, publisher)

values(2, '장발장', '한빛');

/\*

update book

set price = 20000

select \*

라고 한다면, 모든 값이 20000으로 바뀌어버리니까 조건을 걸어야함. where

\*/

---삭제하는것

delete from book;

rollback;

select \* from book;

----수정하는것

update book

set price = 20000

where bookid = 2;

select \*

from book;

book(bookid, bookname, publisher, price)

customer(cust\_id, name, address, phone)

orders(order\_id, cust\_id, book\_id, sal

create tanle

주키

외래키

Primary key(주키속성명)

Foreign key(외래키속성명) References테이블이름(외래키속성명) 다른 테이블에 있는 주키

* 어떤 테이블을 참조할지? References 테이블이름
* Customer 테이블에서 custid 를 참조할건데 여기서도 custid라고 부를거야

Set auto commit

\

--set auto on 이된 상태에서는 rollback이안된다. (undo 같은 느낌)

set auto off;

insert into customer

values(6, '홍길동', '대한민국 대전', NULL);

select\*from customer;

commit;

rollback;

select \* from customer;

delete from customer where custid=6;

select \* from customer;

rollback;

2019-7-5