

9. 테이블에 내용을 추가 · 수정 · 삭제하기 위한 DML

테이블에 새로운 행을 추가하는 INSERT 문을 학습한다.
 테이블의 내용을 수정하기 위한 UPDATE문을 학습한다.
 테이블에 불필요한 행을 삭제하기 위한 DELETE문을 학습한다.

1) 테이블에 새로운 행을 추가하는 INSERT 문

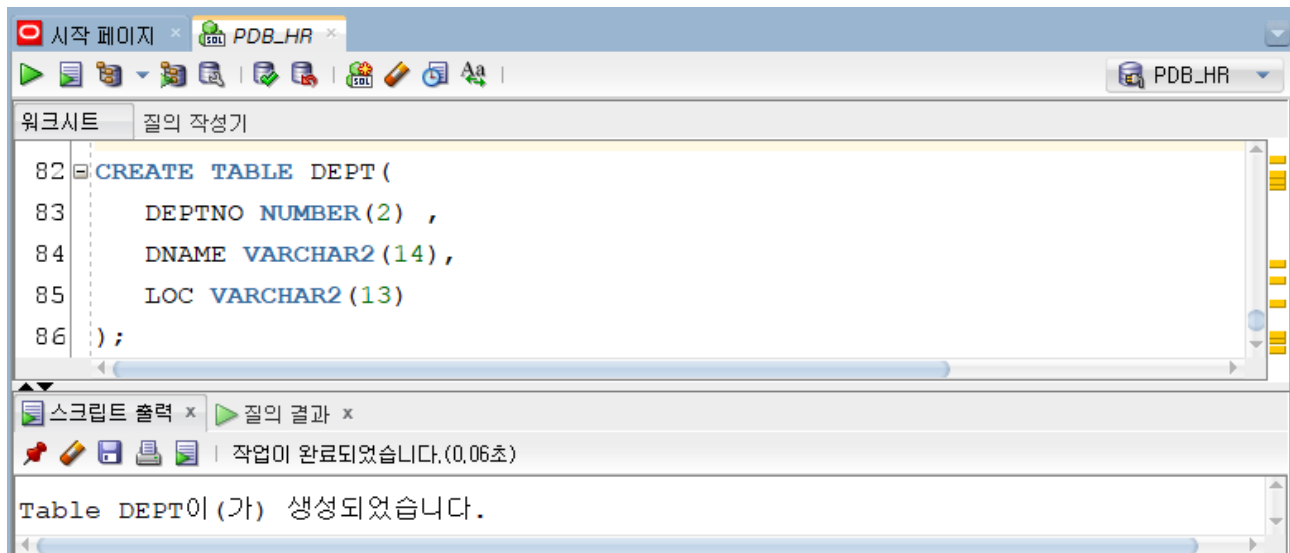
특정한 컬럼에만 DATA를 입력하는 경우	모든 컬럼에 DATA를 입력하는 경우
INSERT INTO table_name (column_value, ...) VALUES(column_value, ...)	INSERT INTO table_name VALUES(column_value, ...)

새로운 행을 추가하기 위해 INSERT문을 사용하면 한 번에 하나의 행만 삽입된다. 기술한 컬럼 목록 순서대로 VALUES에 지정된 값이 삽입된다.

만약 컬럼 목록을 기술하지 않으면 테이블에 있는 컬럼의 디폴트 순서대로 VALUES이하의 값이 삽입되며 문자와 날짜 값은 단일 따옴표(' ')를 사용해야 한다.

먼저 부서 테이블 DEPT을 생성한다.

```
CREATE TABLE DEPT(
    DEPTNO NUMBER(2) ,
    DNAME VARCHAR2(14),
    LOC VARCHAR2(13)
);
```

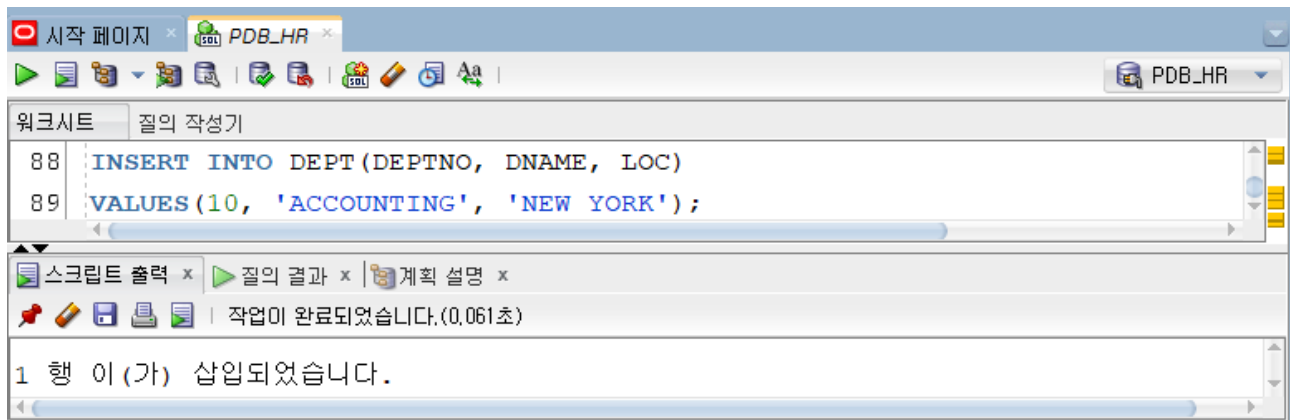


```
DESC DEPT;

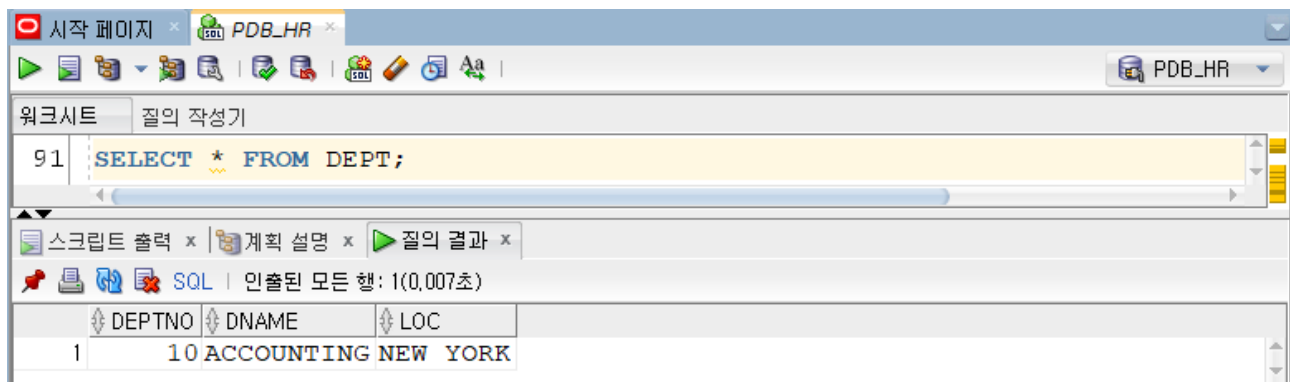
SELECT * FROM DEPT;
```

칼럼 DEPTNO에 10번 부서, DNAME에는 'ACCOUNTING'을, LOC에는 'NEW YORK'을 추가하자

```
INSERT INTO DEPT(DEPTNO, DNAME, LOC)
VALUES(10, 'ACCOUNTING', 'NEW YORK');
```



```
SELECT * FROM DEPT;
```



① INSERT 구문에서 오류 발생의 예

- 칼럼명에 기술된 목록의 수보다 VALUES 다음에 나오는 괄호 안에 기술한 값의 개수가 적으면 에러가 발생한다.

```
INSERT INTO DEPT(DEPTNO, DNAME, LOC)
VALUES (10, 'ACCOUNTING');
```

- 칼럼명에 기술된 목록의 수보다 VALUES 다음에 나오는 괄호에 기술한 값의 개수가 많으면 에러가 발생한다.

```
INSERT INTO DEPT(DEPTNO, DNAME)
VALUES(10, 'ACCOUNTING', 'NEW YORK', 20);
```

- 칼럼명이 잘못 입력되었을 때에도 에러가 발생한다.

```
INSERT INTO DEPT(NUM, DNAME, LOC)
VALUES(10, 'ACCOUNTING', 'NEW YORK');
```

- 칼럼과 입력할 값의 데이터 타입이 서로 맞지 않을 경우에도 에러가 발생한다.

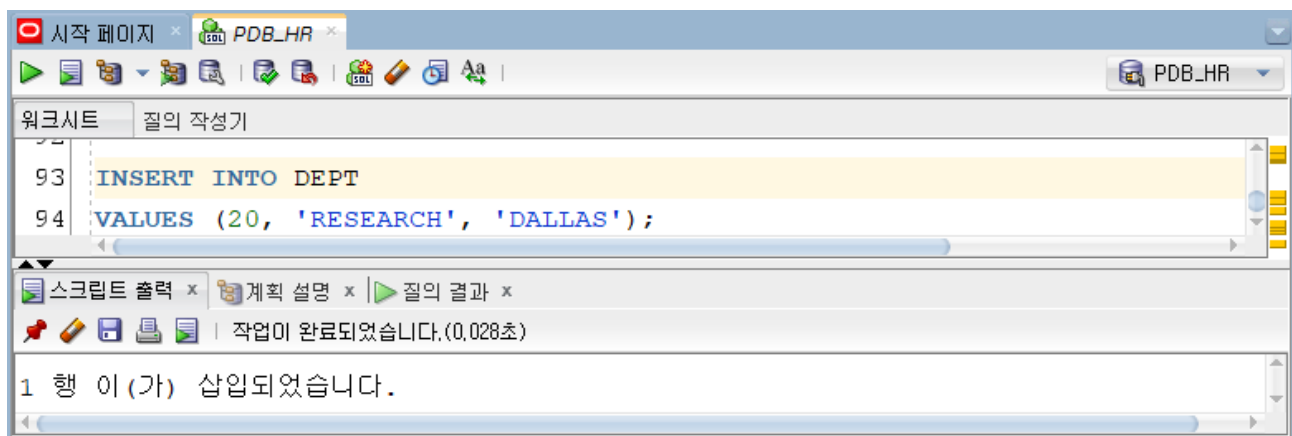
```
INSERT INTO DEPT(DEPTNO, DNAME, LOC)
VALUES(10, ACCOUNTING, 'NEW YORK');
```

② 칼럼명을 생략한 INSERT 구문

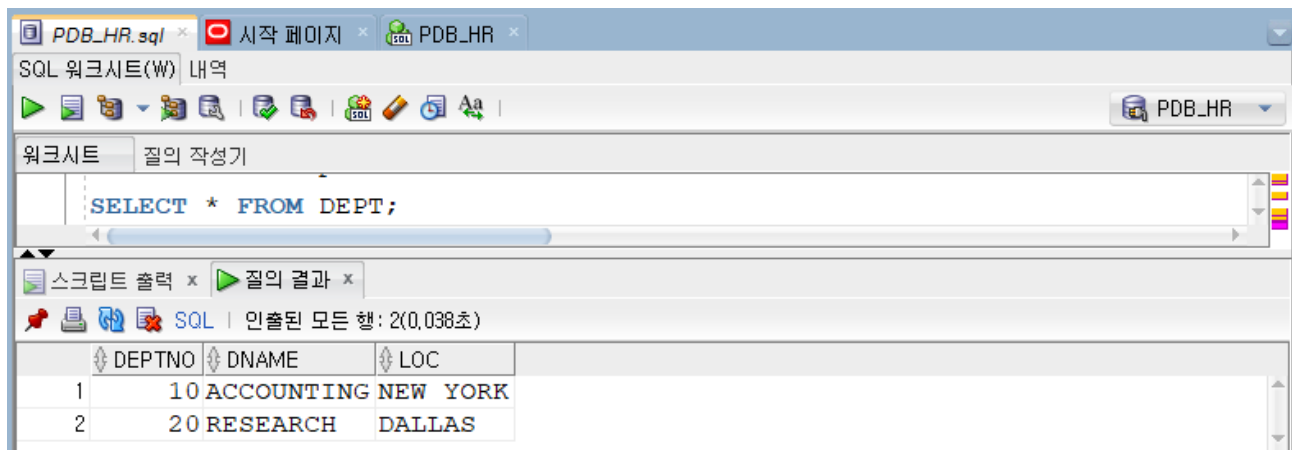
테이블에 로우를 추가할 때 몇몇 특정 칼럼이 아닌 모든 칼럼에 자료를 입력하는 경우에는 굳이 칼럼 목록을 기술하지 않아도 된다. 칼럼 목록이 생략되면 VALUES절 다음의 값들이 테이블의 기본 칼럼 순서대로 입력된다.

칼럼명을 생략한 채 테이블이 갖는 모든 칼럼에 데이터를 추가한다.

```
INSERT INTO DEPT
VALUES (20, 'RESEARCH', 'DALLAS');
```



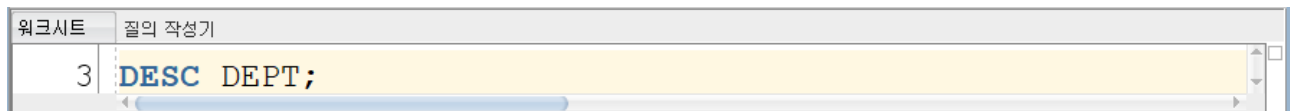
```
SELECT * FROM DEPT;
```



③ NULL 값의 삽입

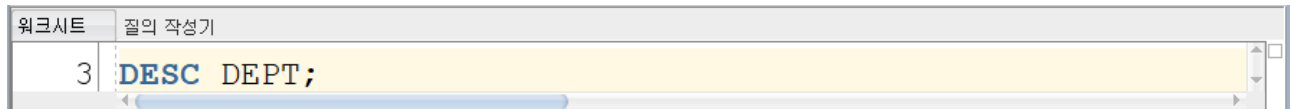
데이터를 입력하는 시점에서 해당 칼럼 값을 모르거나 확정되지 않았을 경우에는 NULL값을 입력해야 한다. NULL 값 삽입에는 암시적인 방법과 명시적인 방법이 있다.

부서 테이블에 칼럼이 NULL값 허용하는지 살펴보면 NULL값을 입력하지 못하는 칼럼에 대해서는 NOT NULL이라고 표시된다.



이름 널 유형

```
-----
DEPTNO    NUMBER (2)
DNAME     VARCHAR2 (14)
LOC       VARCHAR2 (13)
```



DEPT테이블을 생성할 때 제약 조건을 지정하지 않았기에 모든 칼럼에 NULL값을 저장할 수 있도록 허용하고 있다.

이름 널? 유형

```
-----
DEPTNO NOT NULL NUMBER (2)
DNAME   NOT NULL VARCHAR2 (14)
LOC                   VARCHAR2 (13)
```

④ 암시적으로 NULL 값 삽입

암시적인 방법은 칼럼명 리스트에 칼럼을 생략하는 것이다. 즉, 다른 칼럼은 값을 입력하지만 이렇게 생략한 칼럼에는 암시적으로 NULL값이 할당된다.

지역명이 결정되지 않은 30부서에 부서명만 입력

```
INSERT INTO DEPT(DEPTNO, DNAME)
VALUES (30, 'SALES');
```

⑤ 명시적으로 NULL 값의 삽입

명시적 방법은 VALUES에 NULL이라고 직접 기술하여 입력하는 방법이다.

지역명이 결정되지 않아서 다음과 같이 2개의 칼럼만 입력하면 오류가 발생한다.

```
INSERT INTO DEPT
VALUES (40, 'OPERATIONS');
```

쿼리 수정

```
INSERT INTO DEPT
VALUES (40, 'OPERATIONS', NULL);
```

```
SELECT * FROM DEPT;
```

DEPTNO	DNAME	LOC
1	10 ACCOUNTING	NEW YORK
2	20 RESEARCH	DALLAS
3	30 SALES	(null)
4	40 OPERATIONS	(null)

NULL 값을 갖는 컬럼을 추가하기 위해서 NULL 대신 '' 를 사용할 수 있다.

```
INSERT INTO DEPT
VALUES (50, '', 'CHICAGO');
```

```
SELECT * FROM DEPT;
```

기존 데이터를 삭제하고 다시 입력 작업을 하자.

```
DELETE FROM DEPT;
```

기존 테이블에 존재하는 데이터를 다른 테이블에 입력할 때 다음과 같은 쿼리문을 작성하면 된다.

```
INSERT INTO table_name(column1, column2,...)
SELECT column1, column2,... FROM table_name WHERE 조건;
```

DEPT 테이블 생성시 컬럼들의 자릿수와 DEPARTMENTS 테이블의 컬럼의 자릿수를 일치하도록 테이블을 수정하자.

```
ALTER TABLE DEPT MODIFY(DEPTNO NUMBER(4), DNAME VARCHAR2(30));
```

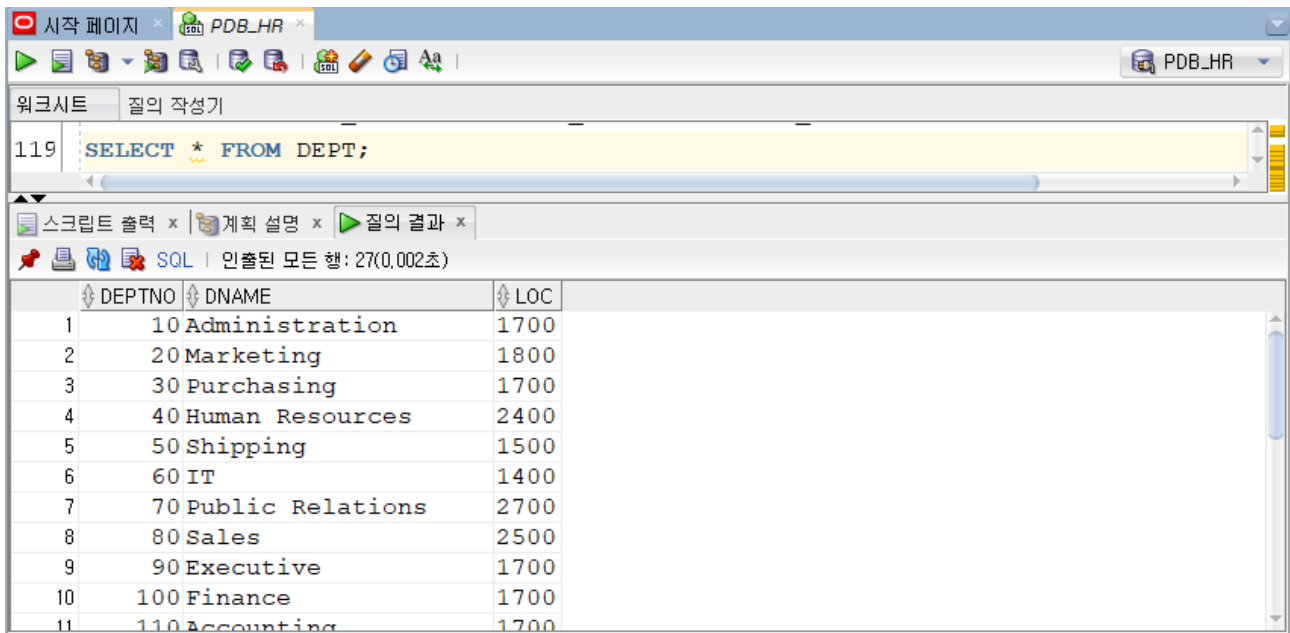
```
INSERT INTO DEPT
SELECT DEPARTMENT_ID, DEPARTMENT_NAME, LOCATION_ID FROM DEPARTMENTS;
```

```
117 INSERT INTO DEPT
118 SELECT DEPARTMENT_ID, DEPARTMENT_NAME, LOCATION_ID FROM DEPARTMENTS;
```

작업이 완료되었습니다. (0.069초)

27개 행 이 (가) 삽입되었습니다.

```
SELECT * FROM DEPT;
```



<문제> TB_CUSTOMER 테이블에 아래 데이터를 추가하라

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2017042	강원진	M	19810603
2017053	나경숙	W	19891225
2017108	박승대	M	19711430

PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
010-8202-8790	wjgang@navi.com	280300	20170123132432
010-4509-0043	ksna#boram.co.kr	4500	20170210180930
NULL	sdpark@haso.com	23450	20170508203450

	CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
1	2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2	2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
3	2017108	박승대	M	19711430	(null)	sdpark@haso.com	23450	20170508203450
4	2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
5	2018254	이혜숙	W	19839012	003 1287 9734	hylee@hansoft.com	570	20180619230805
6	2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518
7	2019069	김한길	M	1992 315	(null)	hkkim@ssoya.com	15320	20190217110704
8	2019167	한잔희	M	19711106	002=1202=5563	chhan@ecom.co.kr	6800	20190508155600
9	2019281	이아름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308

2) 테이블의 내용을 수정하기 위한 UPDATE 문

UPDATE table_name

SET column_name1 = value1, column_name2 = value2, ...

WHERE conditions;

UPDATE문은 기존의 행을 수정하는 것이다. 따라서 어떤 행의 데이터를 수정하는지 WHERE절을 이용하여 조건을 지정해야 한다. WHERE절을 사용하지 않을 경우에는 테이블에 있는 모든 행이 수정된다. 테이블의 전체 행을 수정하려고 했던 것이 아니라면 큰 문제가 발생하므로 WHERE절의 사용 유무를 신중하게 판단해야 한다.

- ① 테이블의 모든 행 변경: WHERE절을 추가하지 않으면 테이블의 모든 행이 변경된다.
EMP 테이블을 생성한다.

```
CREATE TABLE EMP
AS
SELECT * FROM EMPLOYEES;
```

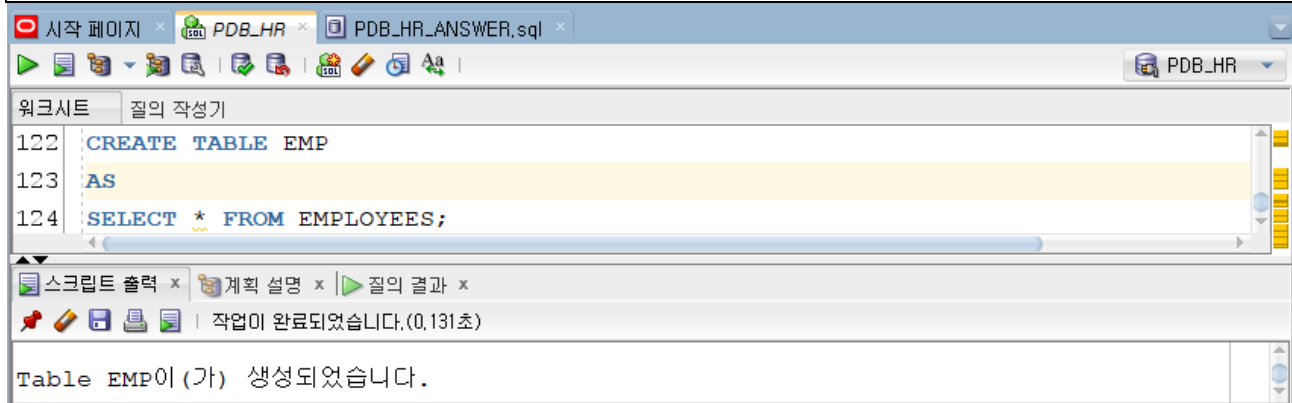
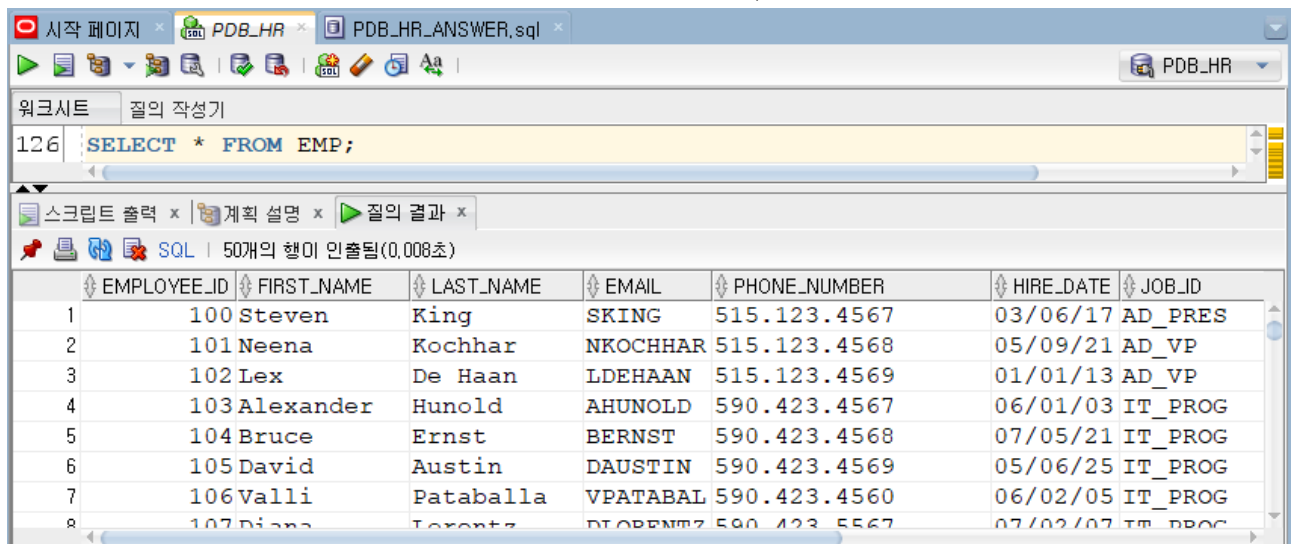


Table EMP이 (가) 생성되었습니다.

```
SELECT * FROM EMP;
```

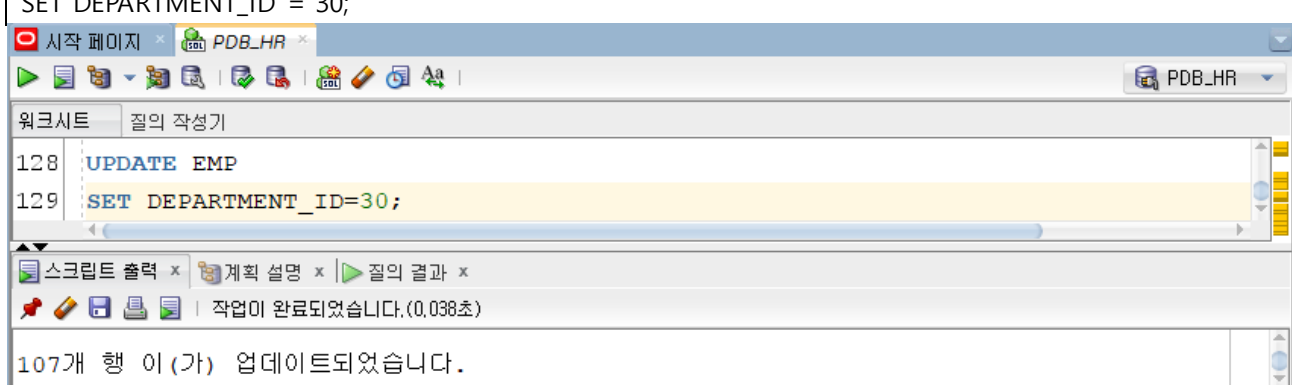
테이블이 이미 존재하여 생성되지 않을 때 DROP TABLE EMP; 로 제거한 후에 다시 생성한다.



	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
1	100	Steven	King	SKING	515.123.4567	03/06/17	AD_PRES
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP
3	102	Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	06/01/03	IT_PROG
5	104	Bruce	Ernst	BERNST	590.423.4568	07/05/21	IT_PROG
6	105	David	Austin	DAUSTIN	590.423.4569	05/06/25	IT_PROG
7	106	Valli	Pataballa	VPATABAL	590.423.4560	06/02/05	IT_PROG
8	107	Diana	Florencia	DFLORENTZ	590.423.5567	07/02/07	IT_PROG

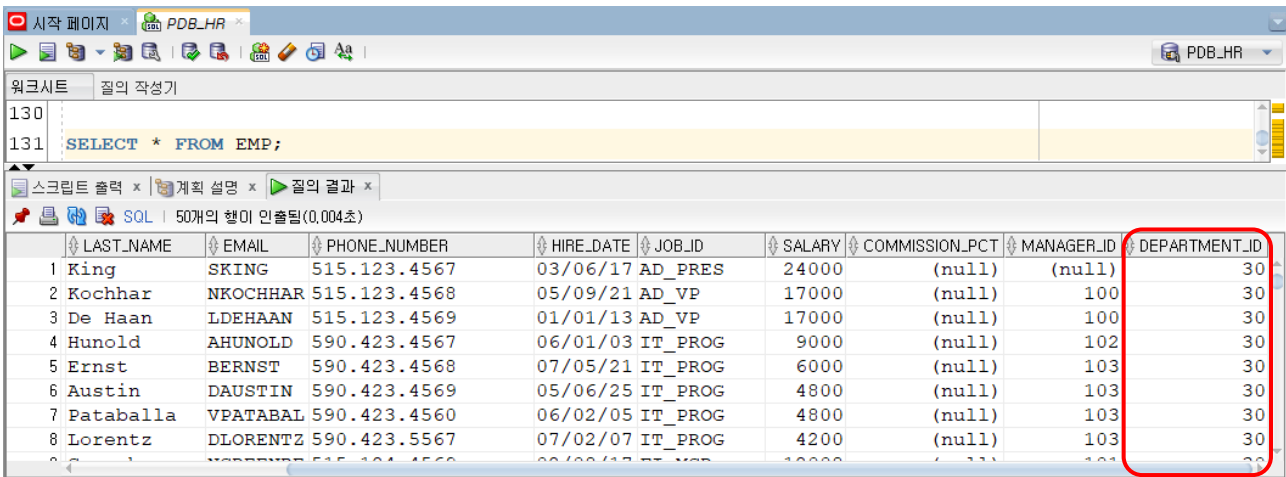
모든 사원의 부서번호를 30번으로 수정하자.

```
UPDATE EMP
SET DEPARTMENT_ID = 30;
```



107개 행 이 (가) 업데이트되었습니다.

```
SELECT * FROM EMP;
```

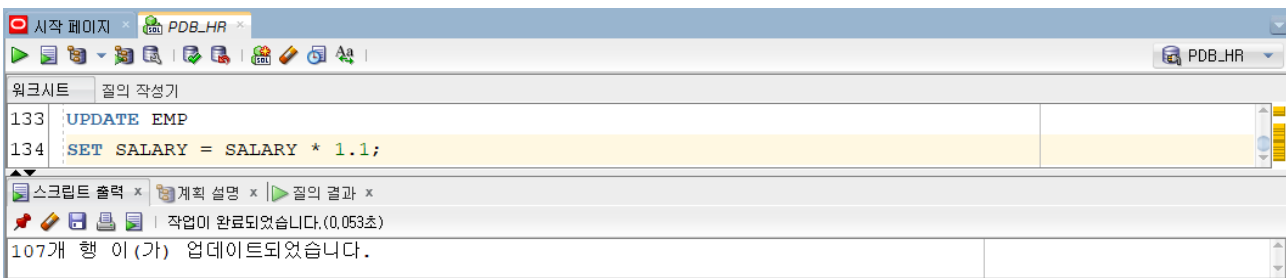


SQL Developer interface showing the results of the query `SELECT * FROM EMP;`. The table has 8 rows and 10 columns. The `DEPARTMENT_ID` column is highlighted with a red box.

	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	King	SKING	515.123.4567	03/06/17	AD_PRES	24000	(null)	(null)	30
2	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP	17000	(null)	100	30
3	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP	17000	(null)	100	30
4	Hunold	AHUNOLD	590.423.4567	06/01/03	IT_PROG	9000	(null)	102	30
5	Ernst	BERNST	590.423.4568	07/05/21	IT_PROG	6000	(null)	103	30
6	Austin	DAUSTIN	590.423.4569	05/06/25	IT_PROG	4800	(null)	103	30
7	Pataballa	VPATABAL	590.423.4560	06/02/05	IT_PROG	4800	(null)	103	30
8	Lorentz	DLORENTZ	590.423.5567	07/02/07	IT_PROG	4200	(null)	103	30

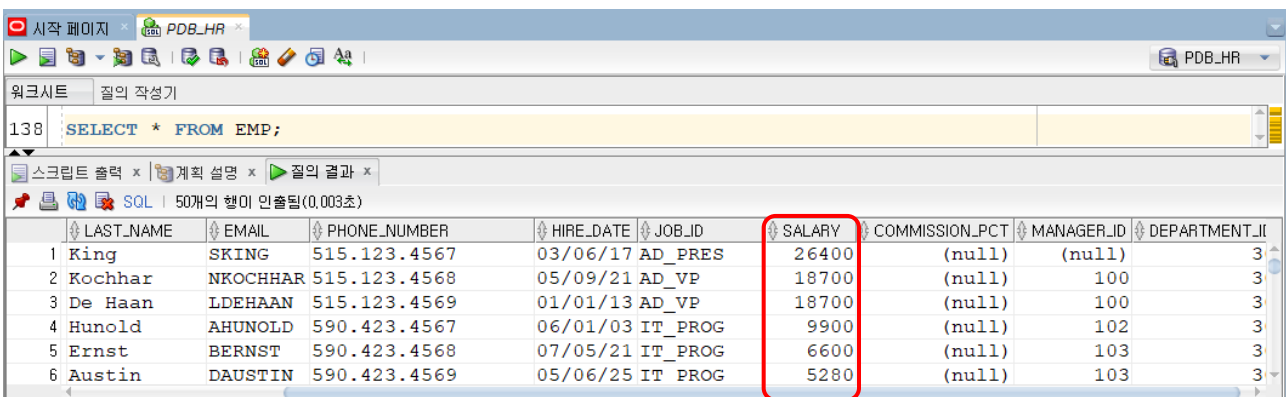
모든 사원의 급여를 10% 인상한다.

```
UPDATE EMP  
SET SALARY = SALARY * 1.1;
```



SQL Developer interface showing the execution of the query `UPDATE EMP SET SALARY = SALARY * 1.1;`. The message "107개 행 이 (가) 업데이트되었습니다." is displayed.

```
SELECT * FROM EMP;
```

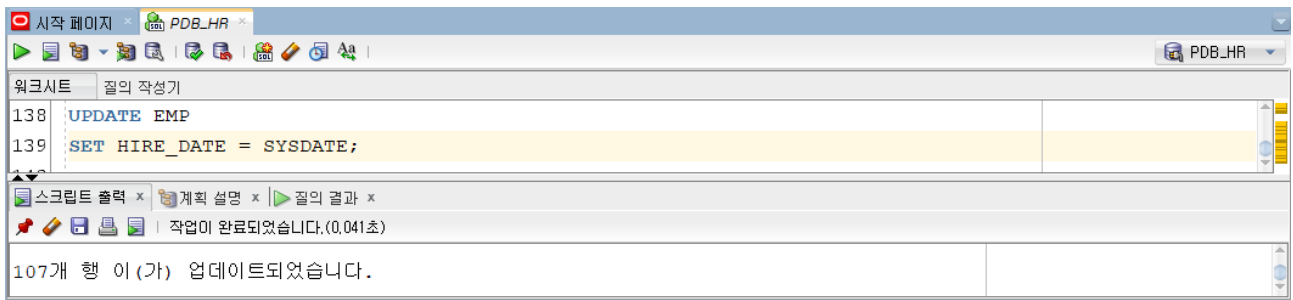


SQL Developer interface showing the results of the query `SELECT * FROM EMP;`. The table has 8 rows and 10 columns. The `SALARY` column is highlighted with a red box.

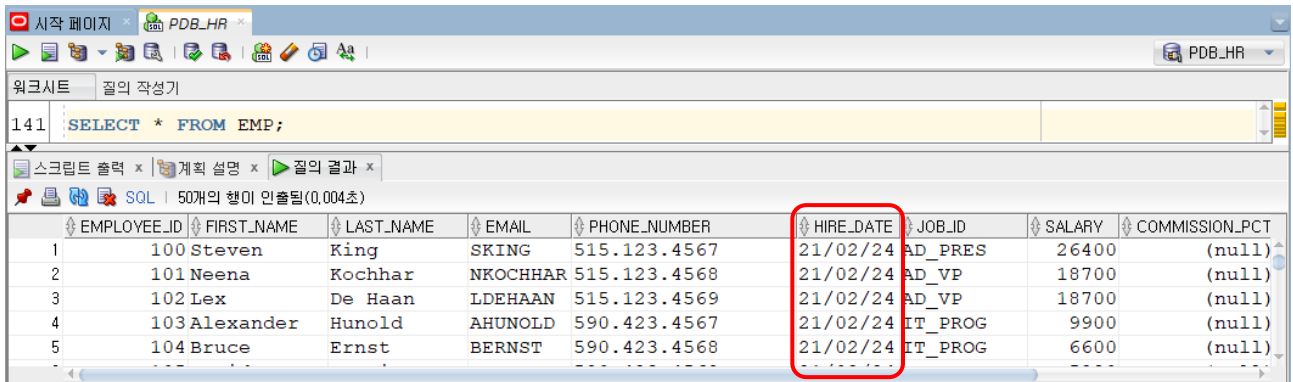
	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	King	SKING	515.123.4567	03/06/17	AD_PRES	26400	(null)	(null)	30
2	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP	18700	(null)	100	30
3	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP	18700	(null)	100	30
4	Hunold	AHUNOLD	590.423.4567	06/01/03	IT_PROG	9900	(null)	102	30
5	Ernst	BERNST	590.423.4568	07/05/21	IT_PROG	6600	(null)	103	30
6	Austin	DAUSTIN	590.423.4569	05/06/25	IT_PROG	5280	(null)	103	30

입사일을 오늘로 수정한다.

```
UPDATE EMP  
SET HIRE_DATE = SYSDATE;
```

```
SELECT * FROM EMP01;
```

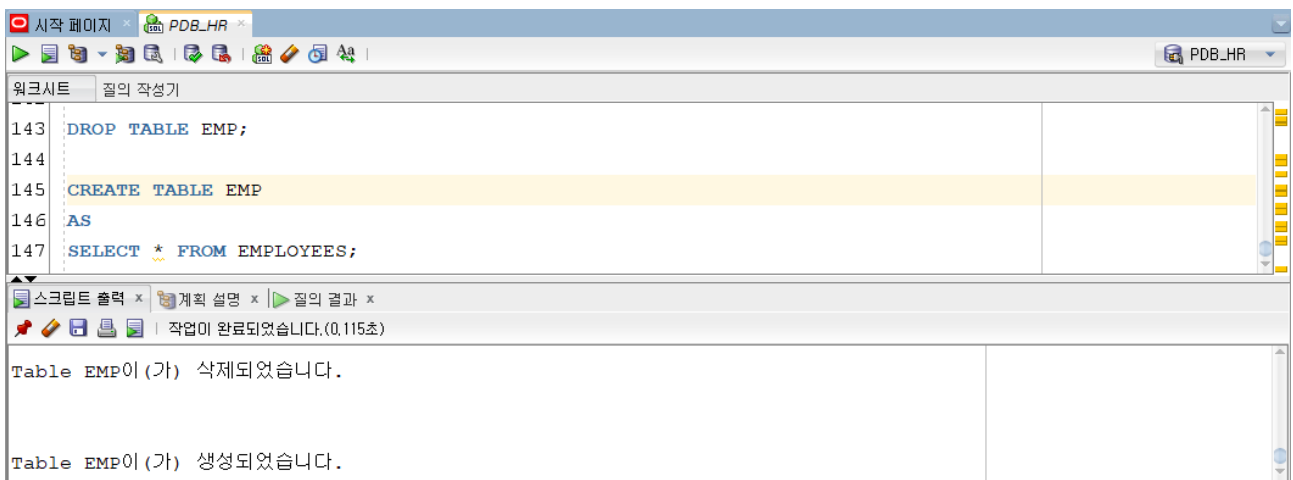


② 테이블의 특정 행만 변경

UPDATE문에 WHERE절을 추가하면 조건을 만족하는 테이블의 특정 행만 변경할 수 있다.

새롭게 사원 테이블을 생성한다.

```
DROP TABLE EMP;
CREATE TABLE EMP
AS
SELECT * FROM EMPLOYEES;
```



```
SELECT * FROM EMP;
```

시작 페이지 PDB_HR

워크시트 질의 작성기

148

149 `SELECT * FROM EMP;`

스크립트 출력 x 계획 설명 x 질의 결과 x

SQL | 50개의 행이 인출됨(0.004초)

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT
1	100	Steven	King	SKING	515.123.4567	03/06/17	AD_PRES	24000	(null)
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP	17000	(null)
3	102	Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP	17000	(null)
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	06/01/03	IT_PROG	9000	(null)
5	104	Bruce	Ernst	BERNST	590.423.4568	07/05/21	IT_PROG	6000	(null)
6	105	David	Austin	DAUSTIN	590.423.4569	05/06/25	IT_PROG	4800	(null)

부서번호가 10번인 사원의 부서번호를 30번으로 수정

```
UPDATE EMP
SET DEPARTMENT_ID =30
WHERE DEPARTMENT_ID =10;
```

시작 페이지 PDB_HR

워크시트 질의 작성기

150

151 `UPDATE EMP`

152 `SET DEPARTMENT_ID =30`

153 `WHERE DEPARTMENT_ID =10;`

스크립트 출력 x 계획 설명 x 질의 결과 x

작업이 완료되었습니다.(0.046초)

1 행 이 (가) 업데이트되었습니다.

```
SELECT * FROM EMP;
```

시작 페이지 PDB_HR

워크시트 질의 작성기

154

155 `SELECT * FROM EMP`

156 `ORDER BY DEPARTMENT_ID;`

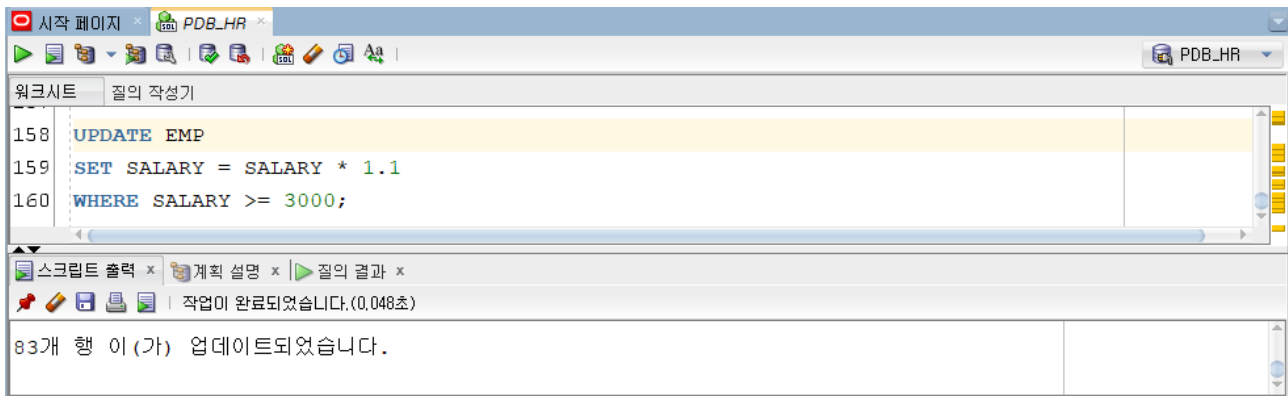
스크립트 출력 x 계획 설명 x 질의 결과 x

SQL | 50개의 행이 인출됨(0.005초)

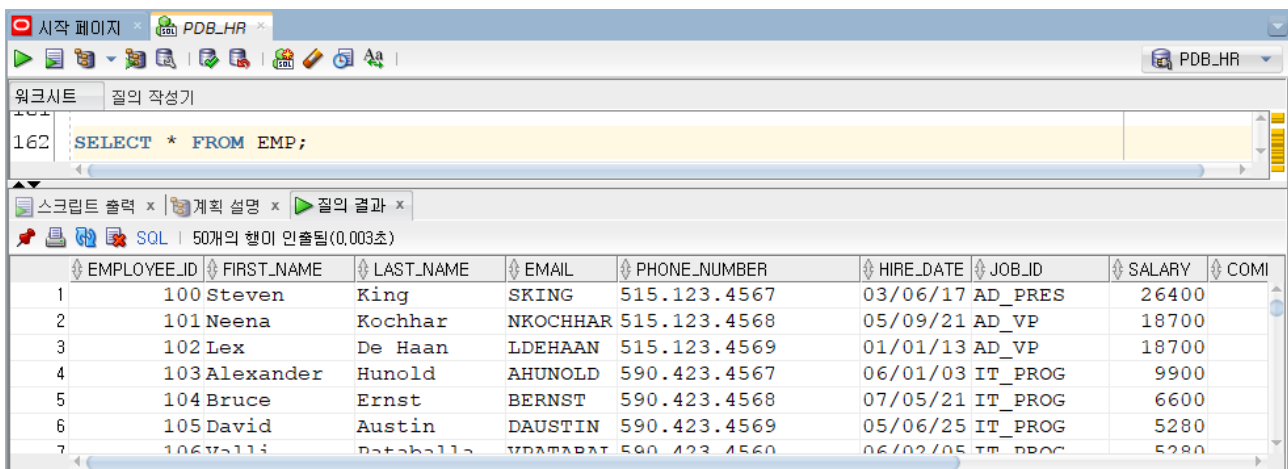
	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	Hartstein	MHARTSTE	515.123.5555	04/02/17	MK_MAN	13000	(null)	100	20
2	Fay	PFAY	603.123.6666	05/08/17	MK_REP	6000	(null)	201	20
3	Raphaely	DRAPHEAL	515.127.4561	02/12/07	PU_MAN	11000	(null)	100	30
4	Khoo	AKHOO	515.127.4562	03/05/18	PU_CLERK	3100	(null)	114	30

급여가 3000 이상인 사원만 급여를 10% 인상

```
UPDATE EMP
SET SALARY = SALARY * 1.1
WHERE SALARY >= 3000;
```

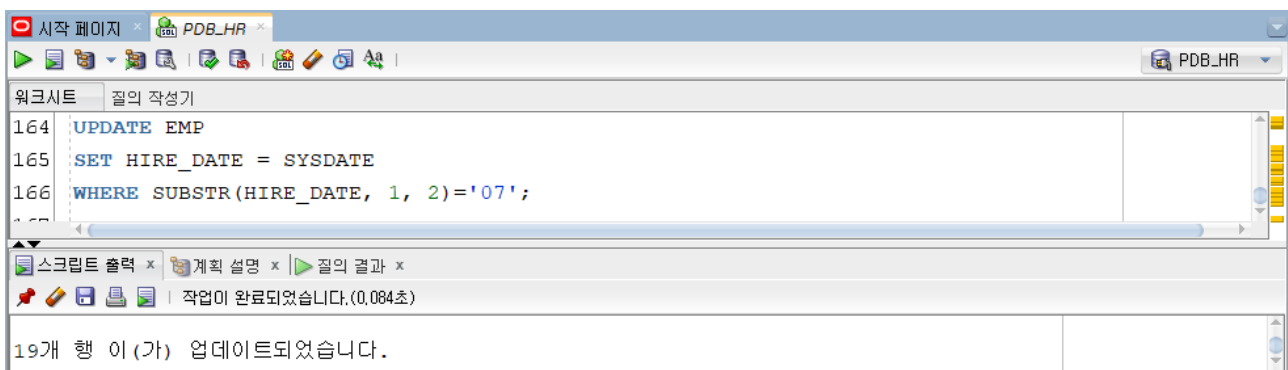


```
SELECT * FROM EMP;
```



2007년에 입사한 사원의 입사일이 오늘로 수정한다

```
UPDATE EMP
SET HIRE_DATE = SYSDATE
WHERE SUBSTR(HIRE_DATE, 1, 2)='07';
```



```
SELECT * FROM EMP;
```

시작 페이지 PDB_HR

워크시트 질의 작성기

168 SELECT * FROM EMP;

스크립트 출력 x 계획 설명 x 질의 결과 x

SQL | 50개의 행이 인출됨(0.003초)

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMI
1	100	Steven	King	SKING	515.123.4567	03/06/17	AD_PRES	26400	
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP	18700	
3	102	Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP	18700	
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	06/01/03	IT_PROG	9900	
5	104	Bruce	Ernst	BERNST	590.423.4568	21/02/24	IT_PROG	6600	
6	105	David	Austin	DAUSTIN	590.423.4569	05/06/25	IT_PROG	5280	
7	106	Valli	Pataballa	VPATABAL	590.423.4560	06/02/05	IT_PROG	5280	
8	107	Diana	Lorentz	DLORENTZ	590.423.5567	21/02/24	IT_PROG	4620	

③ 테이블에서 2개 이상의 칼럼 값 변경

테이블에서 하나의 컬럼이 아닌 복수 개의 컬럼 값을 변경하려면 기존 SET절에 콤마를 추가하고 컬럼=값을 추가 기술하면 된다.

테이블에 2개 이상의 칼럼 값을 수정하기 위해 새로운 테이블 생성

```
DROP TABLE EMP;
CREATE TABLE EMP
AS
SELECT * FROM EMPLOYEES;
```

시작 페이지 PDB_HR

워크시트 질의 작성기

143 DROP TABLE EMP;

144

145 CREATE TABLE EMP

146 AS

147 SELECT * FROM EMPLOYEES;

스크립트 출력 x 계획 설명 x 질의 결과 x

작업이 완료되었습니다.(0.115초)

Table EMP이 (가) 삭제되었습니다.

Table EMP이 (가) 생성되었습니다.

```
SELECT * FROM EMP;
```

시작 페이지 PDB_HR

워크시트 | 질의 작성기

148

149 `SELECT * FROM EMP;`

스크립트 출력 | 계획 설명 | 질의 결과

SQL | 50개의 행이 인출됨(0.004초)

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT
1	100	Steven	King	SKING	515.123.4567	03/06/17	AD_PRES	24000	(null)
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP	17000	(null)
3	102	Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP	17000	(null)
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	06/01/03	IT_PROG	9000	(null)
5	104	Bruce	Ernst	BERNST	590.423.4568	07/05/21	IT_PROG	6000	(null)
6	105	David	Austin	DAUSTIN	590.423.4569	05/06/25	IT_PROG	4800	(null)

`SELECT * FROM EMP WHERE FIRST_NAME = 'Susan';`

시작 페이지 PDB_HR

워크시트 | 질의 작성기

168

168 `SELECT * FROM EMP WHERE FIRST_NAME = 'Susan';`

스크립트 출력 | 계획 설명 | 질의 결과

SQL | 인출된 모든 행: 1(0.004초)

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	203	Susan	Mavris	SMAVRIS	515.123.7777	02/06/07	HR_REP	7150	(null)	101	40

Susan의 부서번호는 20번으로, 직급은 FI_MGR

`UPDATE EMP`
`SET DEPARTMENT_ID = 20, JOB_ID = 'FI_MGR'`
`WHERE FIRST_NAME = 'Susan';`

시작 페이지 PDB_HR

워크시트 | 질의 작성기

170

171 `UPDATE EMP`

172 `SET DEPARTMENT_ID = 20, JOB_ID = 'FI_MGR'`

173 `WHERE FIRST_NAME = 'Susan';`

스크립트 출력 | 계획 설명 | 질의 결과

SQL | 작업이 완료되었습니다.(0.059초)

1 행 이 (가) 업데이트되었습니다.

`SELECT * FROM EMP WHERE FIRST_NAME = 'Susan';`

시작 페이지 PDB_HR

워크시트 | 질의 작성기

173

174 `SELECT * FROM EMP WHERE FIRST_NAME = 'Susan';`

스크립트 출력 | 계획 설명 | 질의 결과

SQL | 인출된 모든 행: 1(0.003초)

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	203	Susan	Mavris	SMAVRIS	515.123.7777	02/06/07	FI_MGR	7150	(null)	101	20

`SELECT * FROM EMP WHERE LAST_NAME = 'Russell';`

177: SELECT * FROM EMP WHERE LAST_NAME = 'Russell';

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	145 John	Russell	JRUSSEL	011.44.1344.429268	04/10/01	SA_MAN	15400	0.4	100	80

LAST_NAME이 Russell인 사원의 급여를 17000로, 커미션 비율이 0.45로 인상된다.

```
UPDATE EMP
SET SALARY = 17000, COMMISSION_PCT = 0.45
WHERE LAST_NAME = 'Russell';
```

179: UPDATE EMP
180: SET SALARY = 17000, COMMISSION_PCT = 0.45
181: WHERE LAST_NAME = 'Russell';

작업이 완료되었습니다. (0.061초)

1 행 이 (가) 업데이트되었습니다.

```
SELECT * FROM EMP WHERE LAST_NAME = 'Russell';
```

183: SELECT * FROM EMP WHERE LAST_NAME = 'Russell';

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	145 John	Russell	JRUSSEL	011.44.1344.429268	04/10/01	SA_MAN	17000	0.45	100	80

3) 테이블에 불필요한 행(레코드)을 삭제하기 위한 DELETE 문

테이블에 특정 로우(행)의 데이터를 삭제한다.

```
DELETE FROM table_name
WHERE conditions;
```

```
SELECT * FROM DEPT;
```

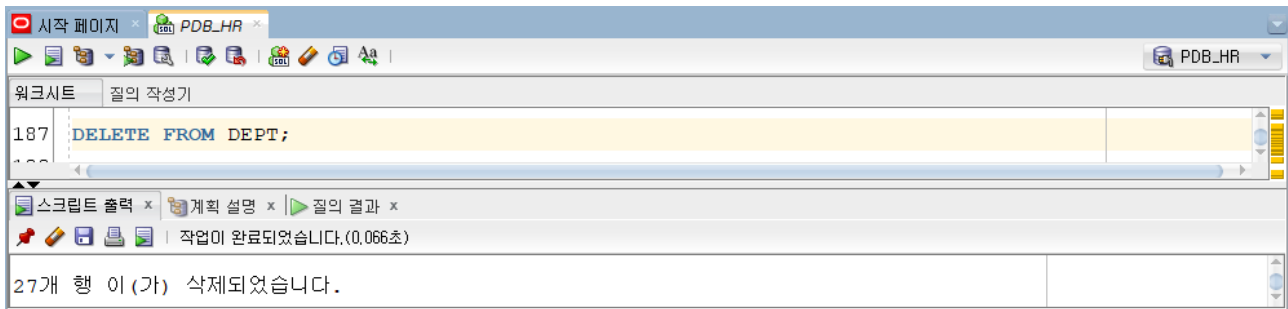
질의 결과

SQL | 인출된 모든 행: 27(0.007초)

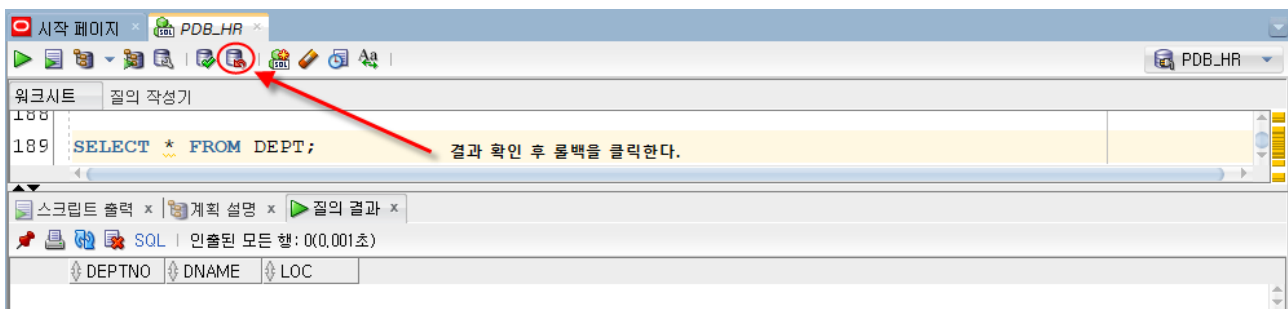
DEPTNO	DNAME	LOC
1	10 Administration	1700
2	20 Marketing	1800
3	30 Purchasing	1700
4	40 Human Resources	2400
5	50 Shipping	1500

부서 테이블의 데이터를 모두 삭제한다.

```
DELETE FROM DEPT;
```

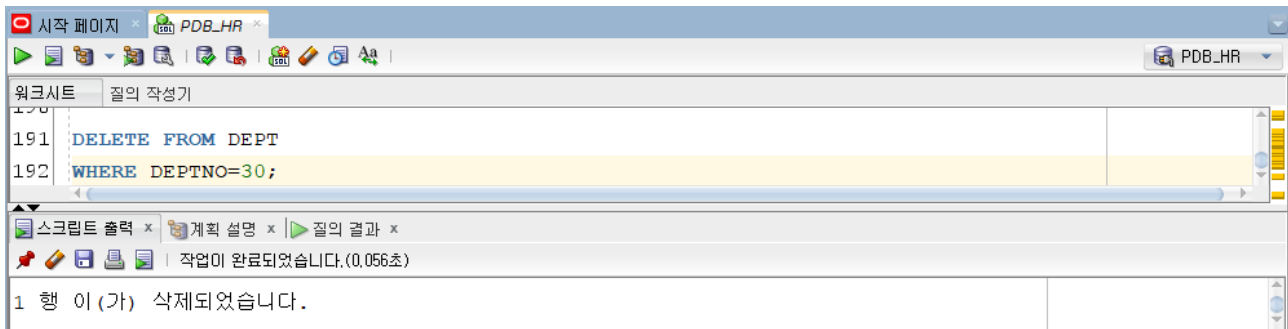


```
SELECT * FROM DEPT;
```

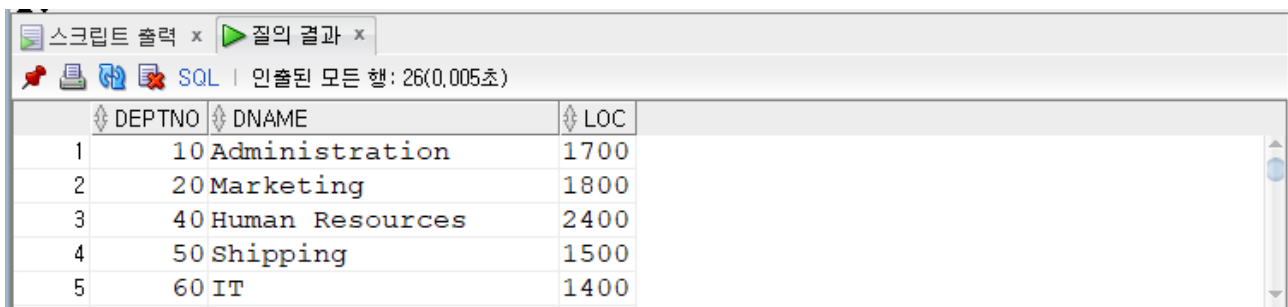


새로운 테이블 생성한 후 30번 부서를 삭제

```
DELETE FROM DEPT WHERE DEPTNO=30;
```



```
SELECT * FROM DEPT;
```



※ DELETE와 TRUNCATE의 차이점

- DELETE 명령어

DELETE 명령어를 사용하여 TABLE의 행을 삭제할 경우에 행이 많으면 행이 삭제될 때마다 많은 자원이 소모된다. 왜냐하면 DELETE 명령어는 삭제 이전 상태로 원상 복귀할 경우를 생각해서 ROLLBACK 정보를 저장하고 있어야 하기 때문이다.

- TRUNCATE 명령어

TRUNCATE 명령어를 사용하여 TABLE의 모든 행을 삭제할 경우 TRUNCATE 명령어는 DDL 명령문으로 ROLLBACK 될 수 없다. 어떤 ROLLBACK 정보도 만들지 않고 즉시 COMMIT 하기에 빠르고 효율적인 방법이다. (TRUNCATE TABLE TABLE_NAME)

4) MERGE 문

MERGE문은 조건을 비교해서 테이블에 해당 조건에 맞는 데이터가 없으면 INSERT문, 있으면 UPDATE를 수행하는 문장이다.

```

MERGE INTO table_name
  USING ( update나 insert 될 데이터 원천 )
  ON ( update될 조건 )
WHEN MATCHED THEN
  UPDATE SET 컬럼1 = 값1, 컬럼2 = 값2, ...
WHEN NOT MATCHED THEN
  INSERT ( 컬럼1, 컬럼2, ...) VALUES (값1, 값2,...)
    
```

MERGE INTO	update나 insert 할 대상 테이블을 명시
USING	원하는 결과를 추출하기 위한 SELECT문을 명시한다. 이 SELECT 문에서 나온 결과를 INSERT 또는 UPDATE 할 예정이다.
ON	SELECT 한 결과와 입력하고 싶은 테이블의 UNIQUE 한 값을 매칭하는 연결고리이다.
WHEN MATCHED THEN ~	SELECT 의 결과가 INSERT 할 테이블에 값이 이미 존재하는 경우 UPDATE 를 실행한다.
WHEN NOT MATCHED THEN ~	ON절에 명시한 조건에 일치하지 않을 경우, INSERT 할 컬럼과 값 명시

실습을 위해 테이블을 생성하여보자.

```

CREATE TABLE TB_ADD_CUSTOMER (
  CUSTOMER_CD      CHAR(7)          PRIMARY KEY,      -- 고객코드
  CUSTOMER_NM      VARCHAR2(10)    NOT NULL,          -- 고객명
  MW_FLG           CHAR(1)          NOT NULL,          -- 성별구분
  BIRTH_DAY        CHAR(8)          NOT NULL,          -- 생일
  PHONE_NUMBER     VARCHAR2(16)
);
    
```



```
INSERT INTO TB_ADD_CUSTOMER (customer_cd, customer_nm, mw_flg, birth_day, phone_number)
VALUES ('2017108','박승대','M','19711430','010-2580-9919');
INSERT INTO TB_ADD_CUSTOMER (customer_cd, customer_nm, mw_flg, birth_day, phone_number)
VALUES ('2019302','전미래','W','19740812','010-8864-0232');
```

```
SELECT * FROM TB_ADD_CUSTOMER;
```

	CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER
1	2017108	박승대	M	19711430	010-2580-9919
2	2019302	전미래	W	19740812	010-8864-0232

```
SELECT * FROM TB_CUSTOMER;
```

	CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
1	2017042	강원진	M	19810603	010-8202-8790	wjgang@navi.com	280300	20170123132432
2	2017053	나경숙	W	19891225	010-4509-0043	ksna#boram.co.kr	4500	20170210180930
3	2017108	박승대	M	19711430	(null)	sdpark@haso.com	23450	20170508203450
4	2018087	서유리	W	19810925	010-1265-8372	urseo@epnt.co.kr	18700	20180204160903
5	2018254	이혜옥	W	19839012	010_1287_9734	hylee@hansoft.com	570	20180619230805
6	2019001	김진숙	W	20010426	010-9842-0074	jskim\$dreami.org	12820	20190101080518
7	2019069	김한길	M	1992 315	(null)	hkkim@ssoya.com	15320	20190217110704
8	2019095	남궁소망	M	19620728	010-6273-8539	(null)	890	20190312124558
9	2019167	한찬희	M	19711106	010=1202=5563	chhan@ecom.co.kr	6800	20190508155600
10	2019281	이아름	W	19940513	010-2620-0723	aulee@hoki.com	35600	20190709201308

TB_ADD_CUSTOMER 테이블의 내용을 TB_CUSTOMER 테이블과 비교하여 데이터가 있을 경우 업데이트 하고 데이터가 없을 경우 입력되도록 쿼리문을 작성한다.

```

MERGE INTO TB_CUSTOMER CU
  USING TB_ADD_CUSTOMER NC
  ON   (CU.CUSTOMER_CD = NC.CUSTOMER_CD)
  WHEN MATCHED THEN
    UPDATE SET CU.CUSTOMER_NM = NC.CUSTOMER_NM,
              CU.MW_FLG = NC.MW_FLG,
              CU.BIRTH_DAY = NC.BIRTH_DAY,
              CU.PHONE_NUMBER = NC.PHONE_NUMBER
  WHEN NOT MATCHED THEN
    INSERT (CU.CUSTOMER_CD, CU.CUSTOMER_NM, CU.MW_FLG, CU.BIRTH_DAY,
          CU.PHONE_NUMBER, CU.EMAIL, CU.TOTAL_POINT, CU.REG_DTTM)
    VALUES (NC.CUSTOMER_CD, NC.CUSTOMER_NM, NC.MW_FLG, NC.BIRTH_DAY,
          NC.PHONE_NUMBER, "", 0, TO_CHAR(SYSDATE,'YYYYMMDDHHMISS'));

```



PDB_HR.sql x 시작 페이지 x PDB_HR x PDB_HR_TABLE.sql x

워크시트 | 질의 작성기

12

13 `SELECT * FROM TB_CUSTOMER;`

질의 결과 x

SQL | 인출된 모든 행: 11(0.002초)

	CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
1	2017042	강원진	M	19810603	010-8202-8790	wjgang@navi.com	280300	20170123132432
2	2017053	나경숙	W	19891225	010-4509-0043	ksna#boram.co.kr	4500	20170210180930
3	2017108	박승대	M	19711430	010-2580-9919	sdpark@haso.com	23450	20170508203450
4	2018087	서유리	W	19810925	010-1265-8372	urseo@epnt.co.kr	18700	20180204160903
5	2018254	이혜옥	W	19839012	010_1287_9734	hylee@hansoft.com	570	20180619230805
6	2019001	김진숙	W	20010426	010-9842-0074	jskim\$dreami.org	12820	20190101080518
7	2019069	김한길	M	1992 315	(null)	hkkim@ssoya.com	15320	20190217110704
8	2019095	남궁소망	M	19620728	010-6273-8539	(null)	890	20190312124558
9	2019167	한찬희	M	19711106	010=1202=5563	chhan@ecom.co.kr	6800	20190508155600
10	2019281	이아름	W	19940513	010-2620-0723	aulee@hoki.com	35600	20190709201308
11	2019302	전미래	W	19740812	010-8864-0232	(null)	0	20210225055103

```

INSERT INTO TB_CUSTOMER VALUES ('2017042','강원진','M','19810603','002-8202-8790','wjgang@navi.com',280300, '20170123132432');
INSERT INTO TB_CUSTOMER VALUES ('2017053','나경숙','W','19891225','002-4509-0043','ksna#boram.co.kr',4500, '20170210180930');
INSERT INTO TB_CUSTOMER VALUES ('2017108','박승대','M','19711430','','sdpark@haso.com',23450, '20170508203450');
INSERT INTO TB_CUSTOMER VALUES ('2018087','서유리','W','19810925','003-1265-8372','urseo@epnt.co.kr',18700, '20180204160903');
INSERT INTO TB_CUSTOMER VALUES ('2018254','이혜옥','W','19839012','003_1287_9734','hylee@hansoft.com',570, '20180619230805');
INSERT INTO TB_CUSTOMER VALUES ('2019001','김진숙','W','20010426','002-9842-0074','jskim$dreami.org',12820, '20190101080518');
INSERT INTO TB_CUSTOMER VALUES ('2019069','김한길','M','1992 315','','hkkim@ssoya.com',15320, '20190217110704');
INSERT INTO TB_CUSTOMER VALUES ('2019095','남궁소망','M','19620728','003-6273-8539','',890,'20190312124558');
INSERT INTO TB_CUSTOMER VALUES ('2019167','한찬희','M','19711106','002=1202=5563','chhan@ecom.co.kr',6800, '20190508155600');
INSERT INTO TB_CUSTOMER VALUES ('2019281','이아름','W','19940513','003-2620-0723','aulee@hoki.com',35600, '20190709201308');

commit;

```

[예제]

[문제] EMP01 테이블을 삭제한 후 다음과 같은 구조로 EMP01테이블을 생성하라.

컬럼명	NULL 허용여부	유형
EMPNO	N	NUMBER(4)
ENAME	N	VARCHAR2(10)
JOB	Y	VARCHAR2(9)
MGR	Y	NUMBER(4)
HIREDATE	N	DATE
SAL	N	NUMBER(7,2)
COMM	Y	NUMBER(7,2)
DEPTNO	N	NUMBER(2)

[문제] EMP01 테이블에 다음과 같이 데이터를 추가하라

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLEAK	7836	80/12/17	800		20
7499	ALLEN	SALESMAN	7369	87/12/20	1600	300	30
7839	KING	PRESIDENT		81/02/08	5000		10

[문제] 회원 정보를 저장하는 테이블을 MEMBERS란 이름으로 생성한다. MEMBERS 테이블은 다음과 같은 컬럼으로 구성된다. (pk는 primary key를 의미한다.)

NO	컬럼명	자료형	크기	NULL허용	키	비고
1	ID	VARCHAR2	20	N	PK	회원ID
2	NAME	VARCHAR2	20	N		이름
3	REGNO	VARCHAR2	13	N		주민등록번호
4	HP	VARCHAR2	13	N		핸드폰번호
5	ADDRESS	VARCHAR2	100	N		주소

[문제] 도서 정보를 저장하는 테이블을 BOOKS이란 이름으로 생성한다. 테이블은 다음과 같은 컬럼으로 구성된다. (pk는 primary key를 의미한다.)

NO	컬럼명	자료형	크기	NULL허용	키	비고
1	CODE	NUMBER	4	N	PK	제품코드
2	TITLE	VARCHAR2	50	N		도서명

3	COUNT	NUMBER	6	N		수량
4	PRICE	NUMBER	10	N		정가
5	PUBLISH	VARCHAR2	50	N		출판사

각자 위 두개의 테이블에 임의의 데이터를 입력해 주세요.