7-1. 데이터 입력/수정/삭제 - INSERT, UPDATE, DELETE 문

홍형경
chariehong@gmail.com
2020.06

1. 데이터 입력, 수정, 삭제

· SQL의 DML 문은 SELECT, **INSERT, UPDATE, DELETE, MERGE**

· INSERT : 테이블에 데이터를 신규로 입력

· UPDATE : 이미 저장된 데이터를 수정

· DELETE : 저장된 데이터를 삭제

· MERGE : 조건에 따라 INSERT와 UPDATE 수행

2. INSERT 문

· 테이블에 데이터를 신규로 입력, 즉 새로운 ROW를 입력하는 문장

· 기본적으로 하나의 INSERT 문장은 한 개의 ROW 입력

· INSERT 구문 종류에 따라 한 INSERT 문장으로 여러 개의 ROW를 동시에 입력 가능

2. INSERT 문

· 구문1

- **INSERT INTO** 테이블명 (컬럼1, 컬럼2, …)

  VALUES (값1, 값2, ….);

· 한 번 실행 시 한 개의 ROW 입력

· 컬럼1, 컬럼2, …와 값1, 값2, … 는 개수, 순서, 데이터 형이 맞아야 함

· 테이블 명 다음 (컬럼1, 컬럼2, …) 부분은 생략 가능, 생략 시 모든 컬럼 값 입력

· NOT NULL 속성인 컬럼은 반드시 입력해야 함

2. INSERT 문 – 구문1 실습

· 실습용 EMP 테이블 생성

```
CREATE TABLE  EMP (
      emp_no      VARCHAR2(30)  NOT NULL,
      emp_name    VARCHAR2(80)  NOT NULL,
      salary      NUMBER          NULL,
      hire_date   DATE            NULL
);

-- 기본 키 추가
ALTER TABLE EMP
ADD CONSTRAINTS EMP_PK PRIMARY KEY (emp_no);
```


2. INSERT 문 – 구문1 실습

```
INSERT INTO EMP ( emp_no, emp_name, salary, hire_date)
VALUES (1, '홍길동', 1000, '2020-06-01');
```

```
SELECT *
  FROM emp;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE
1	1	홍길동	1000	2020-06-01 00:00:00

※ hire_date는 date 형이지만 묵시적 형변환이 적용되어
문자형 값인 '2019-01-01'이 날짜로 자동 변환됨.

2. INSERT 문 – 구문1 실습

```
INSERT INTO EMP ( emp_no, emp_name)
VALUES (2, '김유신');
```

```
SELECT *
  FROM emp;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE
1	1	홍길동	1000	2020-06-01 00:00:00
2	2	김유신	(null)	(null)

※ 테이블의 일부 컬럼만 선정해 입력 가능

2. INSERT 문 – 구문1 실습

```
INSERT INTO EMP ( emp_name, emp_no )
VALUES ('강감찬', 3);
```

```
SELECT *
  FROM emp;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE
1	1	홍길동	1000	2020-06-01 00:00:00
2	2	김유신	(null)	(null)
3	3	강감찬	(null)	(null)

※ 테이블 생성 시 컬럼 순서대로 입력할 필요는 없음,
입력하려는 컬럼과 입력된 값의 순서만 맞추면 정상 입력됨

2. INSERT 문 – 구문1 실습

```
INSERT INTO EMP
VALUES (4, '세종대왕', 1000, SYSDATE);
```

```
SELECT *
  FROM emp;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE
1	1	홍길동	1000	2020-06-01 00:00:00
2	2	김유신	(null)	(null)
3	3	강감찬	(null)	(null)
4	4	세종대왕	1000	2020-06-29 19:45:32

※ 컬럼명 생략 시, VALUES 절에는 테이블의 모든 컬럼에 입력될 값을 명시해야 함. 입력 순서는 테이블 생성 시 기술한 컬럼 순서

2. INSERT 문 – 구문1 실습

```
INSERT INTO EMP ( emp_no,  salary, hire_date)
VALUES (5,  1000, SYSDATE);
```

```
명령의 1 행에서 시작하는 중 오류 발생 -
INSERT INTO EMP ( emp_no,  salary, hire_date)
VALUES (5,  1000, SYSDATE)
오류 보고 -
ORA-01400: NULL을 ("HR"."EMP"."EMP_NAME") 안에 삽입할 수 없습니다
```

※ emp_name 컬럼은 Not Null 컬럼, 따라서 반드시 입력해야 하는데, 누락해서 오류 발생

2. INSERT 문 – 구문1 실습

```
INSERT INTO EMP
VALUES (4, '신사임당', 1000, SYSDATE);
```

```
SELECT *
  FROM emp;
```

```
명령의 1 행에서 시작하는 중 오류 발생 -
INSERT INTO EMP
VALUES (4, '신사임당', 1000, SYSDATE)
오류 보고 -
ORA-00001: 무결성 제약 조건(HR.EMP_PK)에 위배됩니다
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE
1	1	홍길동	1000	2020-06-01 00:00:00
2	2	김유신	(null)	(null)
3	3	강감찬	(null)	(null)
4	4	세종대왕	1000	2020-06-29 19:45:32

※ emp_no에 4를 입력했으나, 이미 입력되어 있음.
기본 키 컬럼은 중복 값을 허용하지 않음

2. INSERT 문 – 구문1 실습

```
INSERT INTO EMP
VALUES (5, '신사임당', 1000, TO_DATE('2020-06-29 19:54:30', 'YYYY-MM-DD HH24:MI:SS'));
```

```
SELECT *
  FROM emp;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE
1	1	홍길동	1000	2020-06-01 00:00:00
2	2	김유신	(null)	(null)
3	3	강감찬	(null)	(null)
4	4	세종대왕	1000	2020-06-29 19:45:32
5	5	신사임당	1000	2020-06-29 19:54:30

※ hire_date 입력 시, TO_DATE 함수를 사용해 정확한 날짜 형식을 주고 입력

2. INSERT 문

· 구문2

- INSERT INTO 테이블명 (컬럼1, 컬럼2, …)

  SELECT exp1, exp2, ..

  FROM …

· 한 번 실행 시 여러 개의 ROW 입력 가능 ➔ SELECT 문이 반환하는 데이터에 따라 좌우됨

· 컬럼1, 컬럼2, …와 exp1, exp2, … 는 개수, 순서, 데이터 형이 맞아야 함

· 테이블명 다음 (컬럼1, 컬럼2, …) 부분은 생략 가능, 생략 시 모든 컬럼 값 입력

· NOT NULL 속성인 컬럼은 반드시 입력해야 함

2. INSERT 문 – 구문2 실습

```
INSERT INTO EMP
SELECT emp_no + 10, emp_name, salary, hire_date
  FROM EMP;


SELECT *
  FROM emp;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE
1	1	홍길동	1000	2020-06-01 00:00:00
2	2	김유신	(null)	(null)
3	3	강감찬	(null)	(null)
4	4	세종대왕	1000	2020-06-29 19:45:32
5	5	신사임당	1000	2020-06-29 19:54:30
6	11	홍길동	1000	2020-06-01 00:00:00
7	12	김유신	(null)	(null)
8	13	강감찬	(null)	(null)
9	14	세종대왕	1000	2020-06-29 19:45:32
10	15	신사임당	1000	2020-06-29 19:54:30

※ 기존에 입력된 5건을 select 해 다시 입력. 단, emp_no 기본 키 컬럼 중복 값 입력 방지를 위해 기존 값에 + 10 해서 입력

2. INSERT 문 – 구문2 실습

```
TRUNCATE TABLE EMP;


INSERT INTO EMP
SELECT employee_id, first_name || ' ' || last_name, salary, hire_date
FROM EMPLOYEES
WHERE department_id = 90;


SELECT *
  FROM emp;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE
1	100	Steven King	24000	2003-06-17 00:00:00
2	101	Neena Kochhar	17000	2005-09-21 00:00:00
3	102	Lex De Haan	17000	2001-01-13 00:00:00

※ employees 테이블에서 부서번호가 90번인 사원의 데이터를 조회해 emp 테이블에 입력

2. INSERT 문 – 구문2 실습

```
INSERT INTO EMP
SELECT employee_id, first_name || ' ' || last_name, salary, hire_date
FROM employees;
```

```
명령의 1 행에서 시작하는 중 오류 발생 -
INSERT INTO EMP
SELECT employee_id, first_name || ' ' || last_name, salary, hire_date
FROM employees
오류 보고 -
ORA-00001: 무결성 제약 조건(HR.EMP_PK)에 위배됩니다
```

※ employees 테이블에서 90번 부서 사원 이미 입력. 다시 전체 사원 입력을 시도하니 기본키인 emp_no 중복 값 오류 발생

2. INSERT 문 – 구문2 실습

· 실습용 EMP_INFO1 테이블 생성

```
CREATE TABLE EMP_INFO1 (
   emp_no          VARCHAR2(30)  NOT NULL,
   emp_name        VARCHAR2(80)  NOT NULL,
   salary          NUMBER           NULL,
   hire_date       DATE             NULL,
   department_name VARCHAR2(80)     NULL,
   country_name    VARCHAR2(80)     NULL
);
```


2. INSERT 문 – 구문2 실습

```
INSERT INTO EMP_INFO1
SELECT a.employee_id,
       a.first_name || ' ' || a.last_name,
       a.salary, a.hire_date,   b.department_name,
       d.country_name
  FROM employees a,  departments b,
       locations c,  countries d
 WHERE a.department_id = b.department_id
   AND b.location_id   = c.location_id
   AND c.country_id    = d.country_id;
```

```
SELECT *
FROM EMP_INFO1;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	DEPARTMENT_NAME	COUNTRY_NAME
1	100	Steven King	24000	2003-06-17 00:00:00	Executive	United States of America
2	101	Neena Kochhar	17000	2005-09-21 00:00:00	Executive	United States of America
3	102	Lex De Haan	17000	2001-01-13 00:00:00	Executive	United States of America
4	103	Alexander Hunold	9000	2006-01-03 00:00:00	IT	United States of America
5	104	Bruce Ernst	6000	2007-05-21 00:00:00	IT	United States of America
6	105	David Austin	4800	2005-06-25 00:00:00	IT	United States of America
7	106	Valli Pataballa	4800	2006-02-05 00:00:00	IT	United States of America
8	107	Diana Lorentz	4200	2007-02-07 00:00:00	IT	United States of America
9	108	Nancy Greenberg	12008	2002-08-17 00:00:00	Finance	United States of America
10	109	Daniel Faviet	9000	2002-08-16 00:00:00	Finance	United States of America
11	110	John Chen	8200	2005-09-28 00:00:00	Finance	United States of America
12	111	Ismael Sciarra	7700	2005-09-30 00:00:00	Finance	United States of America
13	112	Jose Manuel Urman	7800	2006-03-07 00:00:00	Finance	United States of America
14	113	Luis Popp	6900	2007-12-07 00:00:00	Finance	United States of America

※ 조인을 사용한 쿼리를 사용해 그 결과를 emp_info1 테이블에 입력

2. INSERT 문

· 구문3 (Unconditional Multitable Insert)

- **INSERT ALL**
  INTO 테이블명1 (컬럼1, 컬럼2, …)
  VALUES (값1, 값2, …)
  INTO 테이블명2 (컬럼1, 컬럼2, …)
  VALUES (값1, 값2, …)
  ….
  SELECT exp1, exp2, ..
  FROM …

· 한 번 실행 시 여러 테이블에 동시 INSERT

· 컬럼과 값의 쌍 개수, 순서 데이터 형이 맞아야 함

· 입력하고자 하는 컬럼은 조정 가능

· 실제 사용하는 경우는 별로 없음

2. INSERT 문 – 구문3 실습

```
CREATE TABLE  EMP1 (
      emp_no      VARCHAR2(30)  NOT NULL,
      emp_name    VARCHAR2(80)  NOT NULL,
      salary      NUMBER          NULL,
      hire_date   DATE            NULL,
      dept_id     NUMBER          NULL
);
ALTER TABLE EMP1
ADD CONSTRAINTS EMP1_PK PRIMARY KEY (emp_no);

CREATE TABLE  EMP2 (
      emp_no      VARCHAR2(30)  NOT NULL,
      emp_name    VARCHAR2(80)  NOT NULL,
      salary      NUMBER          NULL,
      hire_date   DATE            NULL,
      dept_id     NUMBER          NULL
);
ALTER TABLE EMP2
ADD CONSTRAINTS EMP2_PK PRIMARY KEY (emp_no);
```

```
CREATE TABLE  EMP3 (
      emp_no      VARCHAR2(30)  NOT NULL,
      emp_name    VARCHAR2(80)  NOT NULL,
      salary      NUMBER          NULL,
      hire_date   DATE            NULL,
      dept_id     NUMBER          NULL
);

ALTER TABLE EMP3
ADD CONSTRAINTS EMP3_PK PRIMARY KEY (emp_no);
```


2. INSERT 문 – 구문3 실습

```
INSERT ALL
   INTO EMP1 (emp_no, emp_name, salary, hire_date)
      VALUES  (emp_no, emp_name, salary, hire_date)
   INTO EMP2 (emp_no, emp_name, salary, hire_date)
      VALUES  (emp_no, emp_name, salary, hire_date)
SELECT employee_id emp_no,
          first_name || ' ' || last_name emp_name, salary, hire_date
  FROM employees;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	DEPT_ID
1	100	Steven King	24000	2003-06-17 00:00:00	(null)
2	101	Neena Kochhar	17000	2005-09-21 00:00:00	(null)
3	102	Lex De Haan	17000	2001-01-13 00:00:00	(null)
4	103	Alexander Hunold	9000	2006-01-03 00:00:00	(null)
5	104	Bruce Ernst	6000	2007-05-21 00:00:00	(null)
6	105	David Austin	4800	2005-06-25 00:00:00	(null)
7	106	Valli Pataballa	4800	2006-02-05 00:00:00	(null)
8	107	Diana Lorentz	4200	2007-02-07 00:00:00	(null)
9	108	Nancy Greenberg	12008	2002-08-17 00:00:00	(null)
10	109	Daniel Faviet	9000	2002-08-16 00:00:00	(null)
11	110	John Chen	8200	2005-09-28 00:00:00	(null)
12	111	Ismael Sciarra	7700	2005-09-30 00:00:00	(null)
13	112	Jose Manuel Urman	7800	2006-03-07 00:00:00	(null)
14	113	Luis Popp	6900	2007-12-07 00:00:00	(null)
15	114	Den Raphaely	11000	2002-12-07 00:00:00	(null)
16	115	Alexander Khoo	3100	2003-05-18 00:00:00	(null)
17	116	Shelli Baida	2900	2005-12-24 00:00:00	(null)
18	117	Sigal Tobias	2800	2005-07-24 00:00:00	(null)
19	118	Guy Himuro	2600	2006-11-15 00:00:00	(null)
20	119	Karen Colmenares	2500	2007-08-10 00:00:00	(null)
21	120	Matthew Weiss	8000	2004-07-18 00:00:00	(null)
22	121	Adam Fripp	8200	2005-04-10 00:00:00	(null)
23	122	Payam Kaufling	7900	2003-05-01 00:00:00	(null)

```
SELECT *
FROM EMP1;

SELECT *
FROM EMP2;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	DEPT_ID
1	100	Steven King	24000	2003-06-17 00:00:00	(null)
2	101	Neena Kochhar	17000	2005-09-21 00:00:00	(null)
3	102	Lex De Haan	17000	2001-01-13 00:00:00	(null)
4	103	Alexander Hunold	9000	2006-01-03 00:00:00	(null)
5	104	Bruce Ernst	6000	2007-05-21 00:00:00	(null)
6	105	David Austin	4800	2005-06-25 00:00:00	(null)
7	106	Valli Pataballa	4800	2006-02-05 00:00:00	(null)
8	107	Diana Lorentz	4200	2007-02-07 00:00:00	(null)
9	108	Nancy Greenberg	12008	2002-08-17 00:00:00	(null)
10	109	Daniel Faviet	9000	2002-08-16 00:00:00	(null)
11	110	John Chen	8200	2005-09-28 00:00:00	(null)
12	111	Ismael Sciarra	7700	2005-09-30 00:00:00	(null)
13	112	Jose Manuel Urman	7800	2006-03-07 00:00:00	(null)
14	113	Luis Popp	6900	2007-12-07 00:00:00	(null)
15	114	Den Raphaely	11000	2002-12-07 00:00:00	(null)
16	115	Alexander Khoo	3100	2003-05-18 00:00:00	(null)
17	116	Shelli Baida	2900	2005-12-24 00:00:00	(null)
18	117	Sigal Tobias	2800	2005-07-24 00:00:00	(null)
19	118	Guy Himuro	2600	2006-11-15 00:00:00	(null)
20	119	Karen Colmenares	2500	2007-08-10 00:00:00	(null)
21	120	Matthew Weiss	8000	2004-07-18 00:00:00	(null)
22	121	Adam Fripp	8200	2005-04-10 00:00:00	(null)

2. INSERT 문 – 구문3 실습

```
TRUNCATE TABLE emp1;
TRUNCATE TABLE emp2;

INSERT ALL
   INTO EMP1 (emp_no, emp_name, salary, hire_date)
      VALUES  (emp_no, emp_name, salary, hire_date)
   INTO EMP2 (emp_no, emp_name, salary, hire_date)
      VALUES  (emp_no, emp_name, salary, hire_date)
   INTO EMP3 (emp_no, emp_name)
       VALUES  (emp_no, emp_name)
SELECT employee_id emp_no,
          first_name || ' ' || last_name emp_name, salary, hire_date
  FROM employees;
```

```
SELECT *
FROM EMP3;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	DEPT_ID
1	100	Steven King	(null)	(null)	(null)
2	101	Neena Kochhar	(null)	(null)	(null)
3	102	Lex De Haan	(null)	(null)	(null)
4	103	Alexander Hunold	(null)	(null)	(null)
5	104	Bruce Ernst	(null)	(null)	(null)
6	105	David Austin	(null)	(null)	(null)
7	106	Valli Pataballa	(null)	(null)	(null)
8	107	Diana Lorentz	(null)	(null)	(null)
9	108	Nancy Greenberg	(null)	(null)	(null)
10	109	Daniel Faviet	(null)	(null)	(null)
11	110	John Chen	(null)	(null)	(null)
12	111	Ismael Sciarra	(null)	(null)	(null)
13	112	Jose Manuel Urman	(null)	(null)	(null)
14	113	Luis Popp	(null)	(null)	(null)
15	114	Den Raphaely	(null)	(null)	(null)
16	115	Alexander Khoo	(null)	(null)	(null)
17	116	Shelli Baida	(null)	(null)	(null)
18	117	Sigal Tobias	(null)	(null)	(null)

2. INSERT 문

· 구문4-1 (Conditional Multitable Insert)

```
- INSERT ALL
      WHEN 조건1 THEN
         INTO 테이블명1 (컬럼1, 컬럼2, …)
               VALUES ( 값1, 값2, …)
      WHEN 조건2 THEN
         INTO 테이블명2 (컬럼1, 컬럼2, …)
               VALUES ( 값1, 값2, …)
      ELSE INTO …
      ….
   SELECT  exp1, exp2, ..
      FROM …
```

· WHEN 조건을 체크해 조건이 맞으면 INSERT

· WHEN 조건과 INTO 절을 여러 개 명시할 수 있음

· 한 번 실행 시 여러 테이블에 동시 INSERT

· ELSE 절 추가 가능

2. INSERT 문

· 구문4-2 (Conditional Multitable Insert)

```
 - INSERT FIRST
      WHEN 조건1 THEN
        INTO 테이블명1 (컬럼1, 컬럼2, …)
             VALUES ( 값1, 값2, …)
      WHEN 조건2 THEN
        INTO 테이블명2 (컬럼1, 컬럼2, …)
             VALUES ( 값1, 값2, …)
      ….
      ELSE
        INTO …
  SELECT  exp1, exp2, ..
      FROM …
```

· ALL 대신 FIRST 사용

· 각 ROW 데이터 기준으로 **첫 번째** WHEN 조건을 만족하면 이후 INTO 절 수행

· 첫 번째 조건을 만족한 데이터(ROW)가 두 번째 조건을 만족하더라도 **두 번째 테이블에는 INSERT 되지 않음**, 이후 조건도 동일하게 처리됨

· CASE 표현식과 동작 방식 흡사

2. INSERT 문 – 구문4-1 실습

```
TRUNCATE TABLE emp1;
TRUNCATE TABLE emp2;
TRUNCATE TABLE emp3;

INSERT ALL
  WHEN dept_id = 20 THEN
    INTO EMP1 (emp_no, emp_name, salary, hire_date, dept_id)
       VALUES (emp_no, emp_name, salary, hire_date, dept_id)
  WHEN dept_id BETWEEN 30 AND 50 THEN
    INTO EMP2 (emp_no, emp_name, salary, hire_date, dept_id)
       VALUES (emp_no, emp_name, salary, hire_date, dept_id)
  WHEN dept_id > 50 THEN
    INTO EMP3 (emp_no, emp_name, salary, hire_date, dept_id)
       VALUES (emp_no, emp_name, salary, hire_date, dept_id)
SELECT employee_id emp_no,
       first_name || ' ' || last_name emp_name,
       salary, hire_date, department_id dept_id
  FROM employees;
```

```
SELECT *
FROM EMP1;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	DEPT_ID
1	201	Michael Hartstein	13000	2004-02-17 00:00:00	20
2	202	Pat Fay	6000	2005-08-17 00:00:00	20

```
SELECT *
FROM EMP2;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	DEPT_ID
1	114	Den Raphaely	11000	2002-12-07 00:00:00	30
2	115	Alexander Khoo	3100	2003-05-18 00:00:00	30
3	116	Shelli Baida	2900	2005-12-24 00:00:00	30
4	117	Sigal Tobias	2800	2005-07-24 00:00:00	30
5	118	Guy Himuro	2600	2006-11-15 00:00:00	30
6	119	Karen Colmenares	2500	2007-08-10 00:00:00	30
7	120	Matthew Weiss	8000	2004-07-18 00:00:00	50
8	121	Adam Fripp	8200	2005-04-10 00:00:00	50
9	122	Payam Kaufling	7900	2003-05-01 00:00:00	50
10	123	Shanta Vollman	6500	2005-10-10 00:00:00	50
11	124	Kevin Mourgos	5800	2007-11-16 00:00:00	50

2. INSERT 문 – 구문4-1 실습

```
TRUNCATE TABLE emp1;
TRUNCATE TABLE emp2;
TRUNCATE TABLE emp3;

INSERT ALL
  WHEN dept_id = 20 THEN
    INTO EMP1 (emp_no, emp_name, salary, hire_date, dept_id)
       VALUES (emp_no, emp_name, salary, hire_date, dept_id)
  WHEN dept_id BETWEEN 30 AND 50 THEN
    INTO EMP2 (emp_no, emp_name, salary, hire_date, dept_id)
       VALUES (emp_no, emp_name, salary, hire_date, dept_id)
  WHEN dept_id > 50 THEN
    INTO EMP3 (emp_no, emp_name, salary, hire_date, dept_id)
       VALUES (emp_no, emp_name, salary, hire_date, dept_id)
SELECT employee_id emp_no,
       first_name || ' ' || last_name emp_name,
       salary, hire_date, department_id dept_id
  FROM employees;
```

```
SELECT *
FROM EMP3;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	DEPT_ID
1	100	Steven King	24000	2003-06-17 00:00:00	90
2	101	Neena Kochhar	17000	2005-09-21 00:00:00	90
3	102	Lex De Haan	17000	2001-01-13 00:00:00	90
4	103	Alexander Hunold	9000	2006-01-03 00:00:00	60
5	104	Bruce Ernst	6000	2007-05-21 00:00:00	60
6	105	David Austin	4800	2005-06-25 00:00:00	60
7	106	Valli Pataballa	4800	2006-02-05 00:00:00	60
8	107	Diana Lorentz	4200	2007-02-07 00:00:00	60
9	108	Nancy Greenberg	12008	2002-08-17 00:00:00	100
10	109	Daniel Faviet	9000	2002-08-16 00:00:00	100
11	110	John Chen	8200	2005-09-28 00:00:00	100
12	111	Ismael Sciarra	7700	2005-09-30 00:00:00	100
13	112	Jose Manuel Urman	7800	2006-03-07 00:00:00	100
14	113	Luis Popp	6900	2007-12-07 00:00:00	100
15	145	John Russell	14000	2004-10-01 00:00:00	80
16	146	Karen Partners	13500	2005-01-05 00:00:00	80
17	147	Alberto Errazuriz	12000	2005-03-10 00:00:00	80
18	148	Gerald Cambrault	11000	2007-10-15 00:00:00	80
19	149	Eleni Zlotkey	10500	2008-01-29 00:00:00	80
20	150	Peter Tucker	10000	2005-01-30 00:00:00	80
21	151	David Bernstein	9500	2005-03-24 00:00:00	80

2. INSERT 문 – 구문4-1 실습

```
TRUNCATE TABLE emp1;
TRUNCATE TABLE emp2;
TRUNCATE TABLE emp3;

INSERT ALL
  WHEN salary >= 2500 THEN
    INTO EMP1 (emp_no, emp_name, salary, hire_date, dept_id)
       VALUES (emp_no, emp_name, salary, hire_date, dept_id)
  WHEN salary >= 5000 THEN
    INTO EMP2 (emp_no, emp_name, salary, hire_date, dept_id)
       VALUES (emp_no, emp_name, salary, hire_date, dept_id)
  WHEN salary >= 7000 THEN
    INTO EMP3 (emp_no, emp_name, salary, hire_date, dept_id)
       VALUES (emp_no, emp_name, salary, hire_date, dept_id)
SELECT employee_id emp_no,
       first_name || ' ' || last_name emp_name,
       salary, hire_date, department_id dept_id
  FROM employees;
```

```
SELECT MIN(salary), MAX(salary)
FROM EMP1;
SELECT MIN(salary), MAX(salary)
FROM EMP2;
SELECT MIN(salary), MAX(salary)
FROM EMP3;
```

	MIN(SALARY)	MAX(SALARY)
1	2500	24000

	MIN(SALARY)	MAX(SALARY)
1	5800	24000

	MIN(SALARY)	MAX(SALARY)
1	7000	24000

2. INSERT 문 – 구문4-2 실습

```
TRUNCATE TABLE emp1;
TRUNCATE TABLE emp2;
TRUNCATE TABLE emp3;

INSERT FIRST
  WHEN salary >= 2500 THEN
    INTO EMP1 (emp_no, emp_name, salary, hire_date, dept_id)
      VALUES (emp_no, emp_name, salary, hire_date, dept_id)
  WHEN salary >= 5000 THEN
    INTO EMP2 (emp_no, emp_name, salary, hire_date, dept_id)
      VALUES (emp_no, emp_name, salary, hire_date, dept_id)
  WHEN salary >= 7000 THEN
    INTO EMP3 (emp_no, emp_name, salary, hire_date, dept_id)
      VALUES (emp_no, emp_name, salary, hire_date, dept_id)
SELECT employee_id emp_no,
      first_name || ' ' || last_name emp_name,
      salary, hire_date, department_id dept_id
  FROM employees;
```

```
SELECT MIN(salary), MAX(salary)
FROM EMP1;
SELECT MIN(salary), MAX(salary)
FROM EMP2;
SELECT MIN(salary), MAX(salary)
FROM EMP3;
```

※ employees 테이블의 데이터가 첫 번째 조건을 만족하므로 EMP1 테이블에만 데이터 INSERT

	MIN(SALARY)	MAX(SALARY)
1	2500	24000

	MIN(SAL...	MAX(SALARY)
1	(null)	(null)

	MIN(SAL...	MAX(SALARY)
1	(null)	(null)

3. UPDATE 문

· 테이블에 저장된 데이터를 **수정**하는 문장

· 컬럼 값을 수정, 조건에 따라 여러 개의 ROW 처리 가능

· 한 번 실행으로 여러 개의 컬럼 값, 여러 개의 ROW 처리 가능

· 어떤 ROW를 수정할 것인지는 WHERE 절에서 처리

3. UPDATE 문

· 구문

- UPDATE 테이블명
  SET 컬럼1 = 변경값1,
  컬럼2 = 변경값2,
  …
  WHERE 조건

· 변경 하려는 컬럼과 변경값은 데이터 형이 맞아야 함

· 변경값 항목에는 표현식, 서브쿼리도 사용 가능

· WHERE 조건을 만족하는 ROW 만 처리됨, WHERE 절 생략 시 전체 ROW에 대해 컬럼 값 변경

3. UPDATE 문 – 실습

```
SELECT *
  FROM EMP;
```

	EM...	EMP_NAME	SALARY	HIRE_DATE
1	100	Steven King	24000	2003-06-17 00:00:00
2	101	Neena Kochhar	17000	2005-09-21 00:00:00
3	102	Lex De Haan	17000	2001-01-13 00:00:00

```
UPDATE emp
   SET salary = 0
 WHERE salary < 20000;
```

```
SELECT *
  FROM EMP;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE
1	100	Steven King	24000	2003-06-17 00:00:00
2	101	Neena Kochhar	0	2005-09-21 00:00:00
3	102	Lex De Haan	0	2001-01-13 00:00:00

3. UPDATE 문 – 실습

```
ALTER TABLE emp
ADD retire_date DATE ;

UPDATE emp
      SET retire_date = SYSDATE
  WHERE emp_no = 102;

SELECT *
  FROM EMP;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	RETIRE_DATE
1	100	Steven King	24000	2003-06-17 00:00:00	(null)
2	101	Neena Kochhar	17000	2005-09-21 00:00:00	(null)
3	102	Lex De Haan	17000	2001-01-13 00:00:00	2020-06-29 20:10:10

3. UPDATE 문 – 실습

```
UPDATE EMP_INFO1
      SET emp_name = emp_name || '(middle)'
 WHERE salary BETWEEN 10000 AND 20000;


SELECT *
  FROM EMP_INFO1
 WHERE INSTR(emp_name, 'middle') > 0 ;


-- update 확인
SELECT *
  FROM EMP_INFO1
 WHERE INSTR(emp_name, 'middle') > 0
   AND salary NOT BETWEEN 10000 AND 20000;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	DEPARTMENT_NAME	COUNTRY_NAME
1	101	Neena Kochhar(middle)	17000	2005-09-21 00:00:00	Executive	United States of America
2	102	Lex De Haan(middle)	17000	2001-01-13 00:00:00	Executive	United States of America
3	108	Nancy Greenberg(middle)	12008	2002-08-17 00:00:00	Finance	United States of America
4	114	Den Raphaely(middle)	11000	2002-12-07 00:00:00	Purchasing	United States of America
5	145	John Russell(middle)	14000	2004-10-01 00:00:00	Sales	United Kingdom
6	146	Karen Partners(middle)	13500	2005-01-05 00:00:00	Sales	United Kingdom
7	147	Alberto Errazuriz(middle)	12000	2005-03-10 00:00:00	Sales	United Kingdom
8	148	Gerald Cambrault(middle)	11000	2007-10-15 00:00:00	Sales	United Kingdom
9	149	Eleni Zlotkey(middle)	10500	2008-01-29 00:00:00	Sales	United Kingdom
10	150	Peter Tucker(middle)	10000	2005-01-30 00:00:00	Sales	United Kingdom
11	156	Janette King(middle)	10000	2004-01-30 00:00:00	Sales	United Kingdom
12	162	Clara Vishney(middle)	10500	2005-11-11 00:00:00	Sales	United Kingdom
13	168	Lisa Ozer(middle)	11500	2005-03-11 00:00:00	Sales	United Kingdom
14	169	Harrison Bloom(middle)	10000	2006-03-23 00:00:00	Sales	United Kingdom
15	174	Ellen Abel(middle)	11000	2004-05-11 00:00:00	Sales	United Kingdom
16	201	Michael Hartstein(middle)	13000	2004-02-17 00:00:00	Marketing	Canada
17	204	Hermann Baer(middle)	10000	2002-06-07 00:00:00	Public Relations	Germany
18	205	Shelley Higgins(middle)	12008	2002-06-07 00:00:00	Accounting	United States of America

EMP_NO	EMP_N...	SALARY	HIRE_DA...	DEPART...	COUNT...

3. UPDATE 문 – 실습

```
UPDATE EMP_INFO1
   SET  emp_name = emp_name || ' (1)'
          ,department_name = department_name || ' (1)'
 WHERE hire_date < TO_DATE('2005-01-01', 'YYYY-MM-DD');
```

```
SELECT *
  FROM EMP_INFO1
 WHERE INSTR(department_name, '(1)') > 0 ;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	DEPARTMENT_NAME	COUNTRY_NAME
1	100	Steven King (1)	24000	2003-06-17 00:00:00	Executive (1)	United States of America
2	102	Lex De Haan(middle) (1)	17000	2001-01-13 00:00:00	Executive (1)	United States of America
3	108	Nancy Greenberg(middle) (1)	12008	2002-08-17 00:00:00	Finance (1)	United States of America
4	109	Daniel Faviet (1)	9000	2002-08-16 00:00:00	Finance (1)	United States of America
5	114	Den Raphaely(middle) (1)	11000	2002-12-07 00:00:00	Purchasing (1)	United States of America
6	115	Alexander Khoo (1)	3100	2003-05-18 00:00:00	Purchasing (1)	United States of America
7	120	Matthew Weiss (1)	8000	2004-07-18 00:00:00	Shipping (1)	United States of America
8	122	Payam Kaufling (1)	7900	2003-05-01 00:00:00	Shipping (1)	United States of America
9	133	Jason Mallin (1)	3300	2004-06-14 00:00:00	Shipping (1)	United States of America
10	137	Renske Ladwig (1)	3600	2003-07-14 00:00:00	Shipping (1)	United States of America
11	141	Trenna Rajs (1)	3500	2003-10-17 00:00:00	Shipping (1)	United States of America
12	145	John Russell(middle) (1)	14000	2004-10-01 00:00:00	Sales (1)	United Kingdom
13	156	Janette King(middle) (1)	10000	2004-01-30 00:00:00	Sales (1)	United Kingdom
14	157	Patrick Sully (1)	9500	2004-03-04 00:00:00	Sales (1)	United Kingdom
15	158	Allan McEwen (1)	9000	2004-08-01 00:00:00	Sales (1)	United Kingdom
16	174	Ellen Abel(middle) (1)	11000	2004-05-11 00:00:00	Sales (1)	United Kingdom
17	184	Nandita Sarchand (1)	4200	2004-01-27 00:00:00	Shipping (1)	United States of America
18	192	Sarah Bell (1)	4000	2004-02-04 00:00:00	Shipping (1)	United States of America
19	200	Jennifer Whalen (1)	4400	2003-09-17 00:00:00	Administration (1)	United States of America
20	201	Michael Hartstein(middle) (1)	13000	2004-02-17 00:00:00	Marketing (1)	Canada
21	203	Susan Mavris (1)	6500	2002-06-07 00:00:00	Human Resources (1)	United Kingdom
22	204	Hermann Baer(middle) (1)	10000	2002-06-07 00:00:00	Public Relations (1)	Germany
23	205	Shelley Higgins(middle) (1)	12008	2002-06-07 00:00:00	Accounting (1)	United States of America
24	206	William Gietz (1)	8300	2002-06-07 00:00:00	Accounting (1)	United States of America

3. UPDATE 문 – 실습

```
SELECT *
  FROM EMP1
 WHERE dept_id IS NULL;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	DEPT_ID
1	178	Kimberely Grant	7000	2007-05-24 00:00:00	(null)

```
UPDATE EMP1
   SET dept_id = ( SELECT MAX(department_id)
                     FROM DEPARTMENTS
                    WHERE manager_id IS NULL
                 )
 WHERE dept_id IS NULL;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	DEPT_ID
1	178	Kimberely Grant	7000	2007-05-24 00:00:00	270

```
SELECT *
  FROM EMP1
 WHERE emp_no = 178;
```


4. DELETE 문

· 테이블에 저장된 데이터를 삭제 하는 문장

· ROW 단위로 삭제됨

· WHERE 절 조건에 맞는 ROW가 삭제

· WHERE 절 생략 시 테이블에 있는 모든 ROW 삭제

4. DELETE 문

· 구문

- DELETE [FROM] 테이블명

  WHERE 조건

· FROM 은 생략 가능

· WHERE 조건을 만족하는 ROW 에 한해 삭제됨

· WHERE 조건 생략 시 테이블의 전체 ROW 삭제

4. DELETE 문 – 실습

```
SELECT *
  FROM emp;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE
1	100	Steven King	24000	2003-06-17 00:00:00
2	101	Neena Kochhar	17000	2005-09-21 00:00:00
3	102	Lex De Haan	17000	2001-01-13 00:00:00

```
DELETE emp
 WHERE emp_no in (101, 102);
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE
1	100	Steven King	24000	2003-06-17 00:00:00

4. DELETE 문 – 실습

```
DELETE emp1
 WHERE emp_name LIKE 'J%';
```

```
SELECT *
   FROM emp1
  ORDER BY emp_name;
```

```
COMMIT;
```

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	DEPT_ID
1	121	Adam Fripp	8200	2005-04-10 00:00:00	50
2	196	Alana Walsh	3100	2006-04-24 00:00:00	50
3	147	Alberto Errazuriz	12000	2005-03-10 00:00:00	80
4	103	Alexander Hunold	9000	2006-01-03 00:00:00	60
5	115	Alexander Khoo	3100	2003-05-18 00:00:00	30
6	185	Alexis Bull	4100	2005-02-20 00:00:00	50
7	158	Allan McEwen	9000	2004-08-01 00:00:00	80
8	175	Alyssa Hutton	8800	2005-03-19 00:00:00	80
9	167	Amit Banda	6200	2008-04-21 00:00:00	80
10	187	Anthony Cabrio	3000	2007-02-07 00:00:00	50

	EMP_NO	EMP_NAME	SALARY	HIRE_DATE	DEPT_ID
28	174	Ellen Abel	11000	2004-05-11 00:00:00	80
29	148	Gerald Cambrault	11000	2007-10-15 00:00:00	80
30	183	Girard Geoni	2800	2008-02-03 00:00:00	50
31	118	Guy Himuro	2600	2006-11-15 00:00:00	30
32	169	Harrison Bloom	10000	2006-03-23 00:00:00	80
33	204	Hermann Baer	10000	2002-06-07 00:00:00	70
34	126	Irene Mikkilineni	2700	2006-09-28 00:00:00	50
35	111	Ismael Sciarra	7700	2005-09-30 00:00:00	100
36	119	Karen Colmenares	2500	2007-08-10 00:00:00	30
37	146	Karen Partners	13500	2005-01-05 00:00:00	80
38	188	Kelly Chung	3800	2005-06-14 00:00:00	50
39	197	Kevin Feeney	3000	2006-05-23 00:00:00	50
40	124	Kevin Mourgos	5800	2007-11-16 00:00:00	50
41	178	Kimberely Grant	7000	2007-05-24 00:00:00	260
42	129	Laura Bissot	3300	2005-08-20 00:00:00	50
43	102	Lex De Haan	17000	2001-01-13 00:00:00	90

5. SQL Developer에서 파일 Import

· SQL Developer를 이용해 csv, excel 파일을 읽어 테이블에 데이터 저장

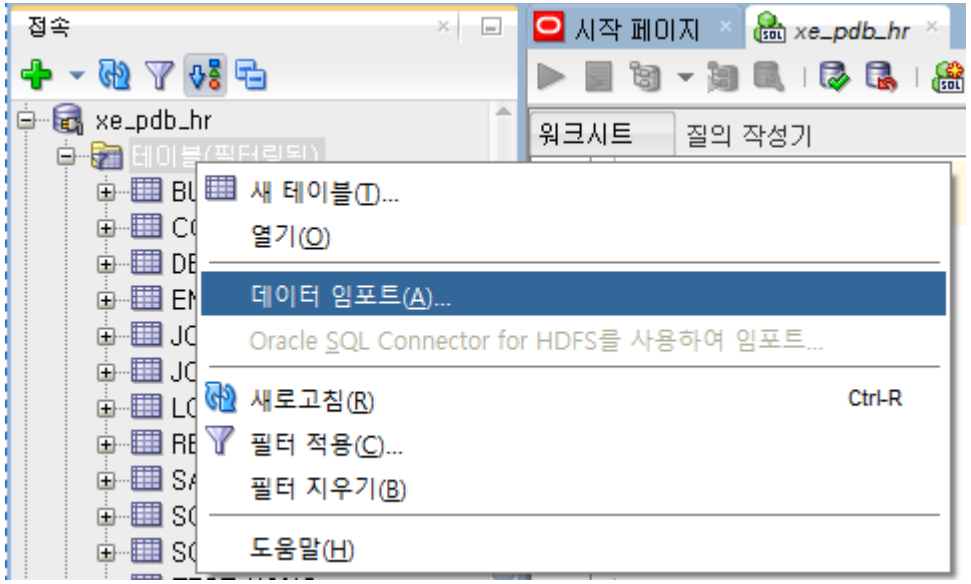
· titanic_data.csv 파일 준비

  - gitbub 사이트 8차시 폴더

· SQL Developer 실행 후 로그인

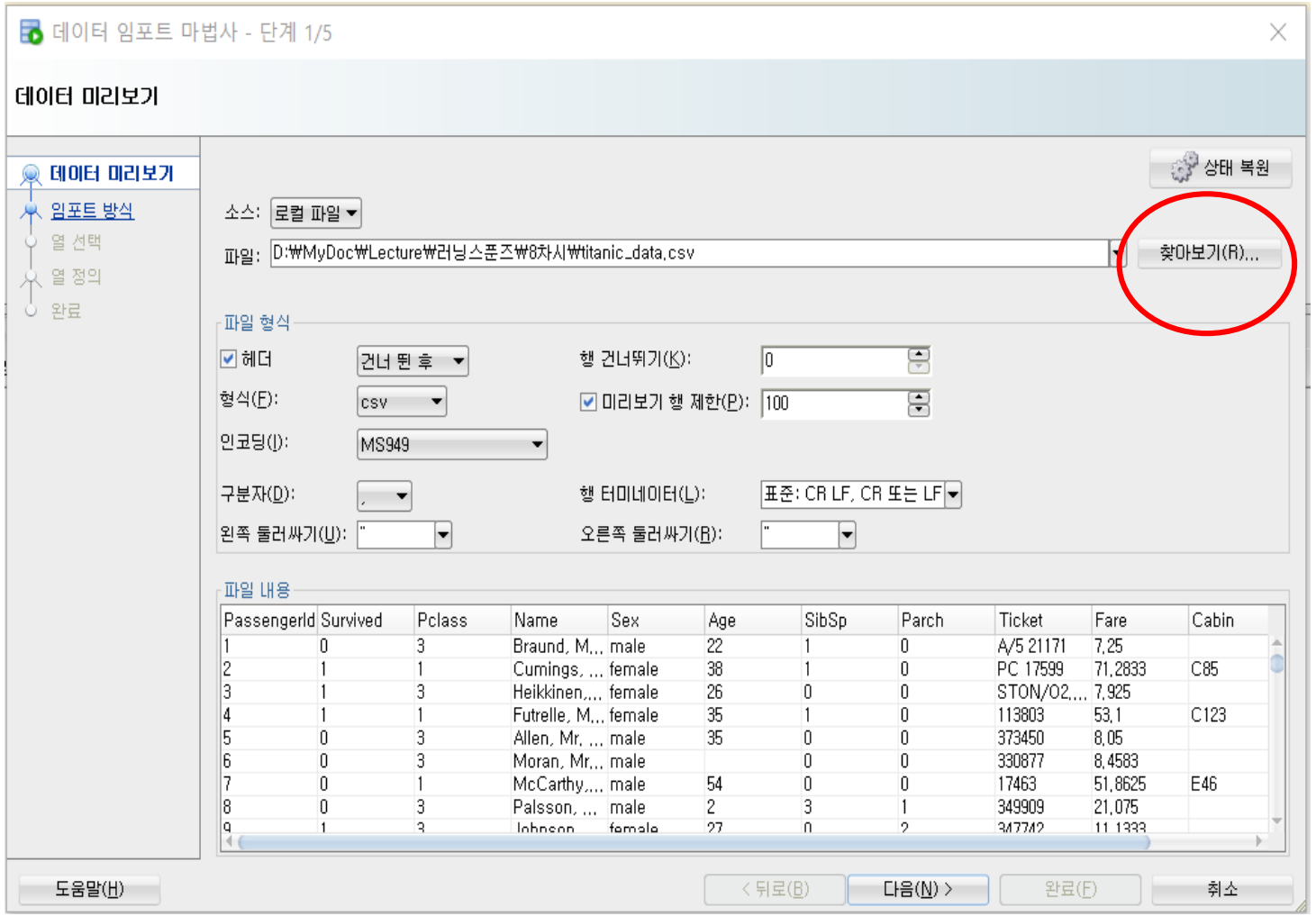
5. SQL Developer에서 파일 Import

· 왼쪽 접속 창에서 테이블 선택
→ 오른쪽 마우스 클릭
→ 데이터 임포트(A) 선택



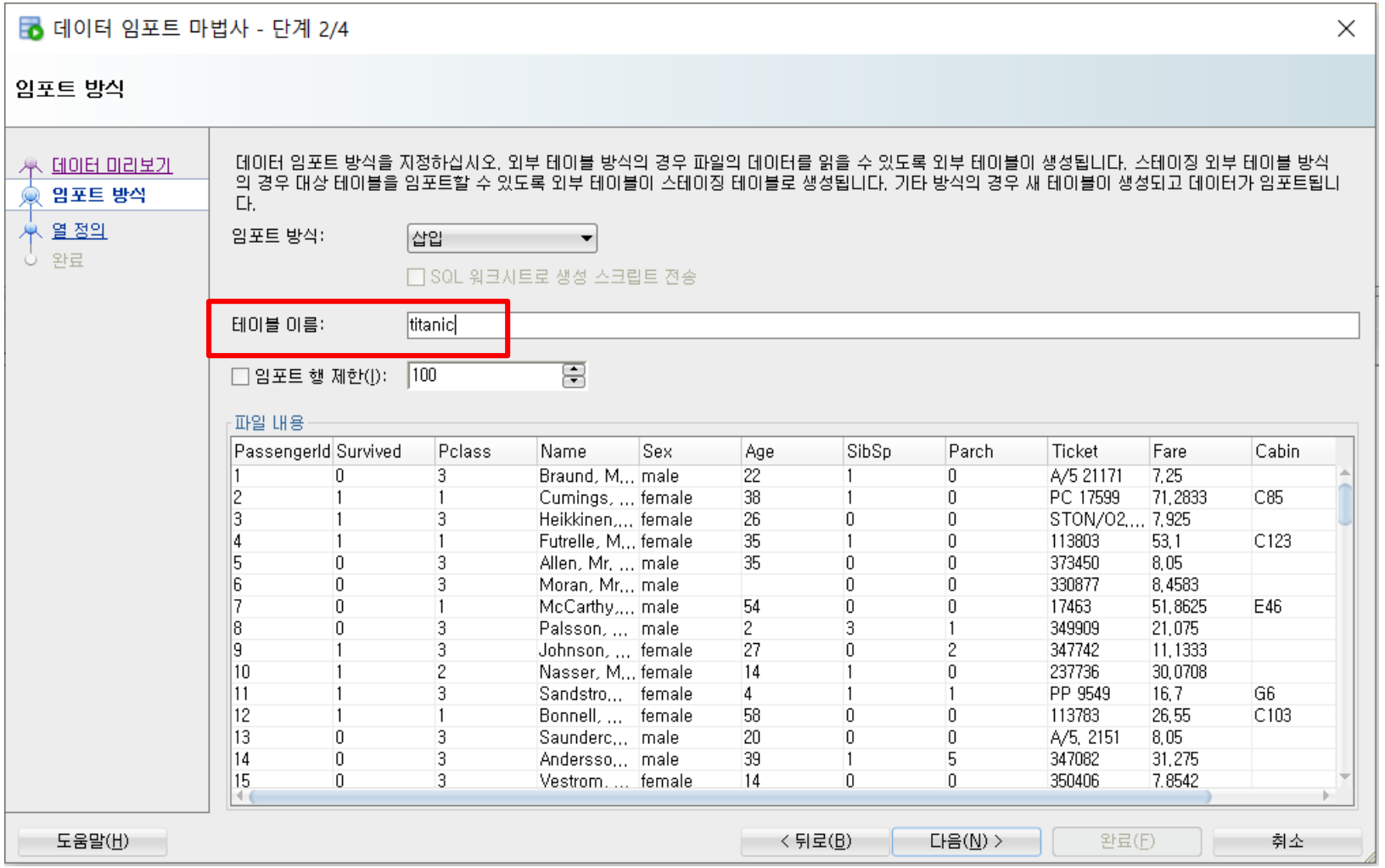
5. SQL Developer에서 파일 Import

- 데이터 임포트 마법사 창
  → 파일 선택 : titanic_data.csv
  → 다음 버튼 클릭



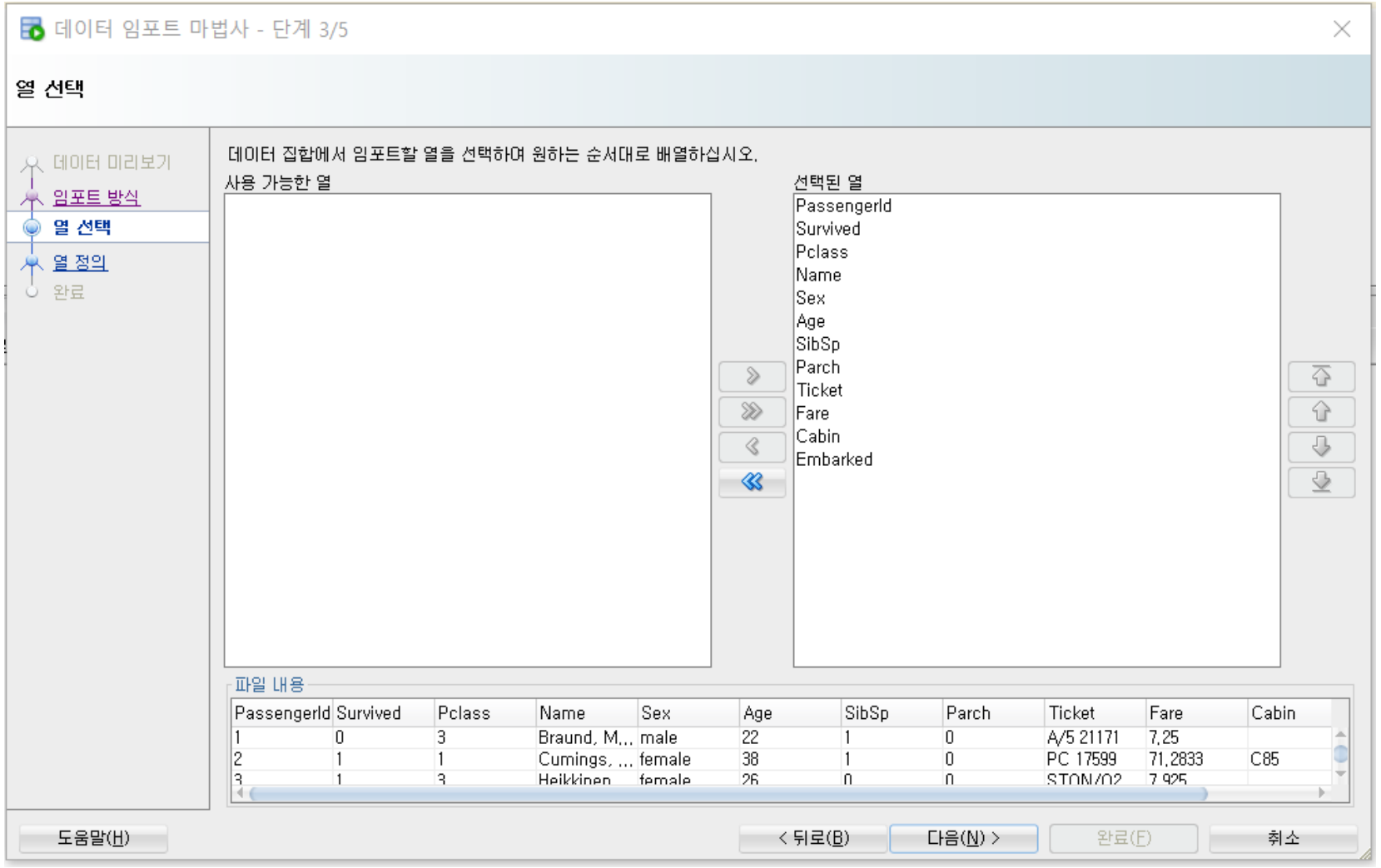
5. SQL Developer에서 파일 Import

- 데이터 임포트 마법사 창
  → 테이블 이름 : **titanic** 입력
  → 다음 버튼 클릭



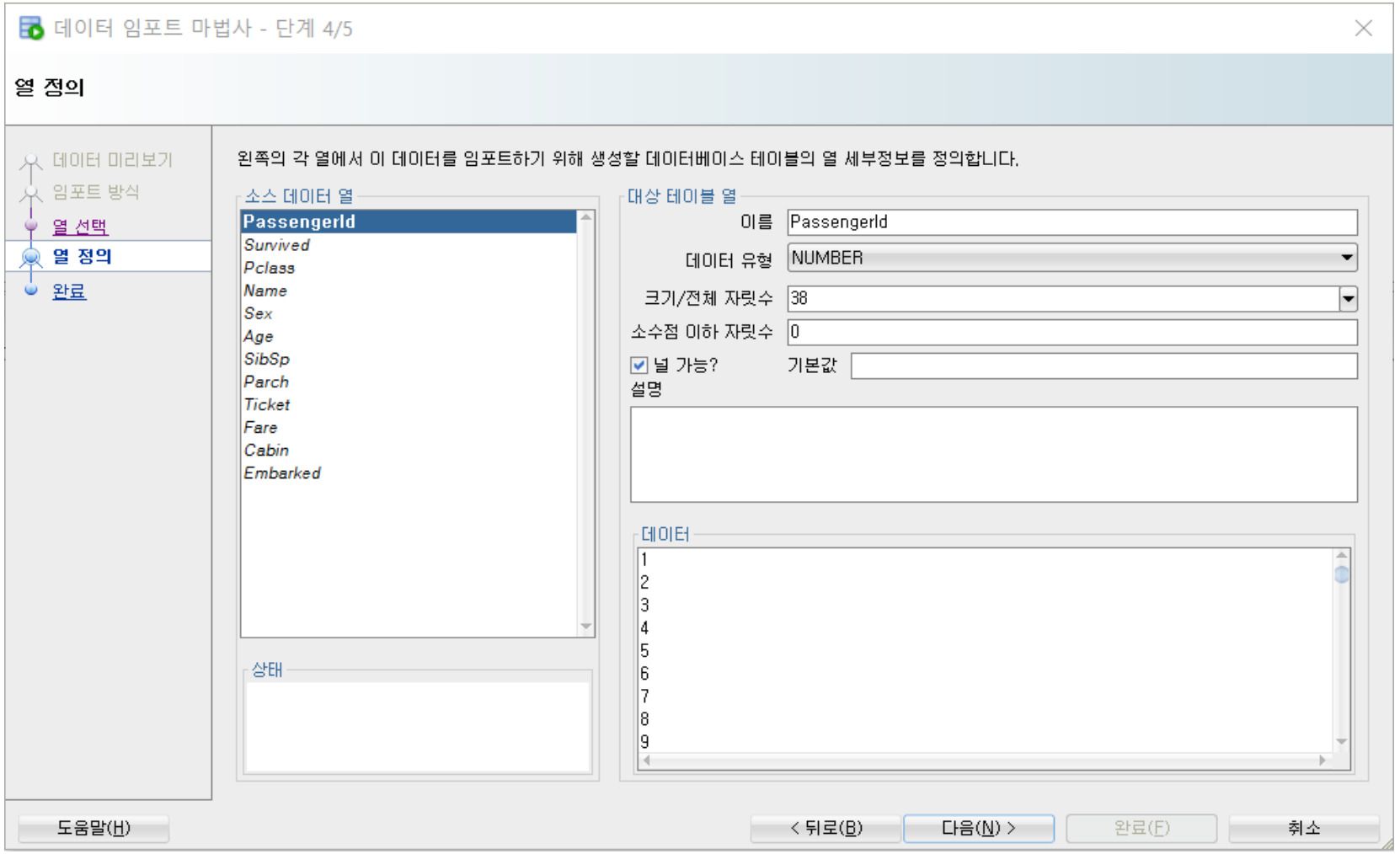
5. SQL Developer에서 파일 Import

· 데이터 임포트 마법사 창
→ 다음 버튼 클릭



5. SQL Developer에서 파일 Import

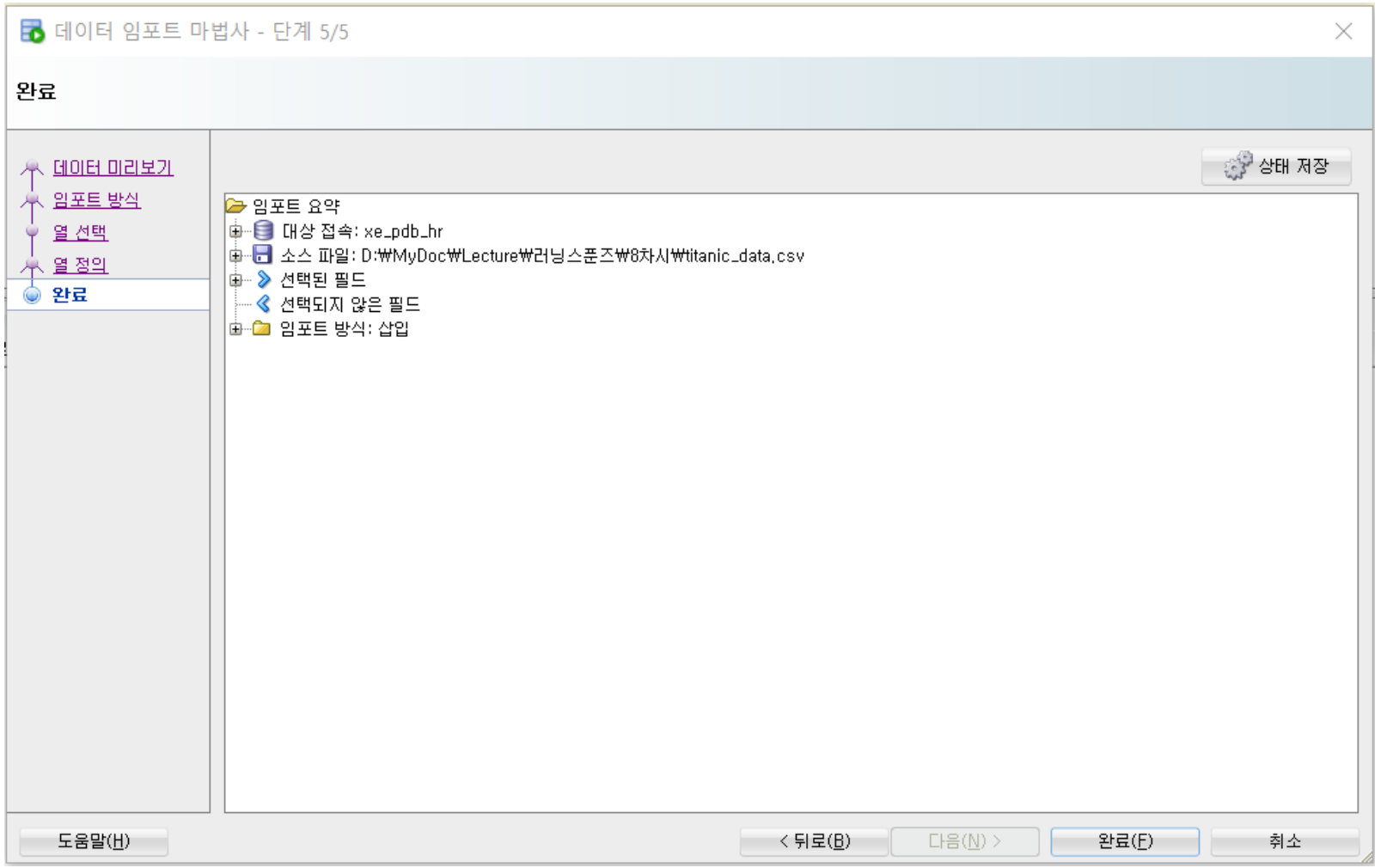
- 데이터 임포트 마법사 창
  → 각 컬럼 데이터 형 확인 후 다음 버튼 클릭



5. SQL Developer에서 파일 Import

· 데이터 임포트 마법사 창

→ 완료 버튼 클릭



5. SQL Developer에서 파일 Import

· 데이터 확인

```
SELECT *
    FROM titanic;
```

	PASSENGERID	SURVIVED	PCLASS	NAME	SEX	AGE	SIBSP	PARCH	TICKET	FARE	CABIN	EMBARKED
1	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25	(null)	S
2	2	1	1	Cumings, Mrs. John Bradley (...	female	38	1	0	PC 17599	71.2833	C85	C
3	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925	(null)	S
4	4	1	1	Futrelle, Mrs. Jacques Heath...	female	35	1	0	113803	53.1	C123	S
5	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05	(null)	S
6	6	0	3	Moran, Mr. James	male	(null)	0	0	330877	8.4583	(null)	Q
7	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075	(null)	S
9	9	1	3	Johnson, Mrs. Oscar W (Elisa...	female	27	0	2	347742	11.1333	(null)	S
10	10	1	2	Nasser, Mrs. Nicholas (Adele...	female	14	1	0	237736	30.0708	(null)	C
11	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
12	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
13	13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.05	(null)	S
14	14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31.275	(null)	S
15	15	0	3	Vestrom, Miss. Hulda Amanda ...	female	14	0	0	350406	7.8542	(null)	S
16	16	1	2	Hewlett, Mrs. (Mary D Kingco...	female	55	0	0	248706	16	(null)	S
17	17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29.125	(null)	Q
18	18	1	2	Williams, Mr. Charles Eugene	male	(null)	0	0	244373	13	(null)	S
19	19	0	3	Vander Planke, Mrs. Julius (...	female	31	1	0	345763	18	(null)	S
20	20	1	3	Masselmani, Mrs. Fatima	female	(null)	0	0	2649	7.225	(null)	C
21	21	0	2	Fynney, Mr. Joseph J	male	35	0	0	239865	26	(null)	S
22	22	1	2	Beesley, Mr. Lawrence	male	34	0	0	248698	13	D56	S
23	23	1	3	McGowan, Miss. Anna "Annie"	female	15	0	0	330923	8.0292	(null)	Q

학습정리

· SQL의 DML 중 데이터를 가공하는 문장에는 INSERT, UPDATE, DELETE 가 있다.

· 테이블에 신규 데이터를 입력하는 문장은 INSERT 문이다.

· UPDATE 문은 저장된 데이터를 수정하는 문장이다.

· DELETE 문은 저장된 데이터를 삭제하는 문장이다.

· UPDATE와 DELETE 문 사용 시 WHERE 조건절 사용에 유의해야 한다.

Quiz

1. 부서 테이블에서 부서명이 IT 부서 정보를 이용해 새로운 부서를 입력해 보세요. 새로운 부서의 부서번호는 500, 부서명이 IT2이고 manager_id와 location_id 값은 IT 부서와 동일합니다.

Quiz

2. 부서테이블에서 deparment_id 값이 280이상인 건 중 manager_id 값이 null인 건은 100, 아닌 건은 110으로 변경하는 문장을 작성해 보세요.

Quiz

3. departments 테이블에서 department_id 값이 280번 이상인 건을 삭제하는 문장을 작성해 보세요.

