

2025년 상반기 K-디지털 트레이닝

# 테이블 (심화1)

---

[KB] IT's Your Life

✓ 다음 컬럼을 가지는 userTBL과 buyTBL을 정의하세요.

- 기존에 테이블이 존재하면 삭제함
- userTBL

컬럼명	타입	필수 여부	비고
userID	고정문자(8)	필수	기본키
name	가변문자(10)	필수	
birthyear	정수	필수	

- buyTBL

컬럼명	타입	필수 여부	비고
num	정수	필수	기본키, 자동 증가
userID	고정문자(8)	필수	
prodName	고정문자(6)	필수	

- ALTER를 이용해 userID를 외래키로 수정하세요.

✓ 다음 컬럼을 가지는 userTBL과 buyTBL을 정의하세요.

```
DROP TABLE IF EXISTS buyTBL, userTBL;  
CREATE TABLE userTBL  
( userID CHAR(8) NOT NULL PRIMARY KEY,  
  name VARCHAR(10) NOT NULL,  
  birthYear INT NOT NULL  
);
```

```
CREATE TABLE buyTBL (  
  num INT AUTO_INCREMENT NOT NULL PRIMARY KEY,  
  userID CHAR(8) NOT NULL,  
  prodName CHAR(6) NOT NULL  
);
```

```
ALTER TABLE buyTBL  
  ADD CONSTRAINT FK_userTBL_buyTBL  
  FOREIGN KEY(userID)  
  REFERENCES userTBL(userID);
```

✓ 다음 조건을 만족하는 userTBL 테이블을 정의하세요.

- 기존 buyTBL, userTBL을 삭제하세요.
- 컬럼

컬럼명	타입	필수 여부	비고
userID	고정문자(8)	필수	기본키
name	가변문자(10)	필수	
birthyear	정수	필수	
email	고정문자(30)	옵션	중복불가

- ☑ 다음 조건을 만족하는 userTBL 테이블을 정의하세요.

```
USE tableDB;  
DROP TABLE IF EXISTS buyTBL, userTBL;  
CREATE TABLE userTBL (  
    userID    CHAR(8) NOT NULL PRIMARY KEY,  
    name      VARCHAR(10) NOT NULL,  
    birthYear INT NOT NULL,  
    email     CHAR(30) NULL UNIQUE  
);
```

✓ 다음 조건을 만족하는 userTBL 테이블을 정의하세요.

- 기존 userTBL을 삭제하세요.
- 컬럼

컬럼명	타입	필수 여부	비고
userID	고정문자(8)	필수	기본키
name	가변문자(10)		
birthYear	정수		1900이상 2023이하 값만 허용
mobile	고정문자(3)	필수	

✓ 다음 조건을 만족하는 userTBL 테이블을 정의하세요.

```
DROP TABLE IF EXISTS userTBL;
CREATE TABLE userTBL
( userID CHAR(8) PRIMARY KEY,
  name VARCHAR(10) ,
  birthYear INT CHECK (birthYear >= 1900 AND birthYear <= 2023),
  mobile1 char(3) NULL,
  CONSTRAINT CK_name CHECK ( name IS NOT NULL)
);
```

✓ 다음 조건을 만족하는 userTBL 테이블을 정의하세요.

- 기존 userTBL을 삭제하세요.
- 컬럼

컬럼명	타입	필수 여부	비고
userID	고정문자(8)	필수	기본키
name	가변문자(10)	필수	
birthYear	정수	필수	기본값 -1
addr	고정문자(2)	필수	기본값 서울
mobile1	고정문자(3)	옵션	
mobile2	고정문자(8)	옵션	
height	작은 정수	옵션	기본값 170
mDate	날짜	옵션	

- 기본값 추가를 확인할 수 있는 데이터를 추가하세요.



✓ 다음 조건을 만족하는 userTBL 테이블을 정의하세요.

```
DROP TABLE IF EXISTS userTBL;
CREATE TABLE userTBL
( userID      CHAR(8) NOT NULL PRIMARY KEY,
  name        VARCHAR(10) NOT NULL,
  birthYear   INT NOT NULL DEFAULT -1,
  addr        CHAR(2) NOT NULL DEFAULT '서울',
  mobile1     CHAR(3) NULL,
  mobile2     CHAR(8) NULL,
  height      SMALLINT NULL DEFAULT 170,
  mDate       DATE NULL
);
```

```
-- default문은 DEFAULT로 설정된 값을 자동 입력한다.
INSERT INTO usertbl VALUES ('LHL', '이혜리', default, default, '011', '1234567',
                             default, '2023.12.12');
-- 열 이름이 명시되지 않으면 DEFAULT로 설정된 값을 자동 입력한다.
INSERT INTO usertbl(userID, name) VALUES('KAY', '김아영');
-- 값이 직접 명기되면 DEFAULT로 설정된 값은 무시된다.
INSERT INTO usertbl VALUES ('WB', '원빈', 1982, '대전', '019', '9876543', 176,
                             '2020.5.5');
SELECT * FROM usertbl;
```

- ✓ 앞에서 만든 userTBL에 대해서 다음 조건을 처리하도록 수정하세요.
  - mobile1 컬럼을 삭제함
  - name 컬럼명을 uName으로 변경함
  - 기본키를 제거함

✓ 앞에서 만든 userTBL에 대해서 다음 조건을 처리하도록 수정하세요.

○ mobile1 컬럼을 삭제함

```
ALTER TABLE usertbl  
DROP COLUMN mobile1;
```

○ name 컬럼명을 uName으로 변경함

```
ALTER TABLE usertbl  
CHANGE COLUMN name uName VARCHAR2(20) NULL;
```

○ 기본키를 제거함

```
ALTER TABLE usertbl  
DROP PRIMARY KEY;
```

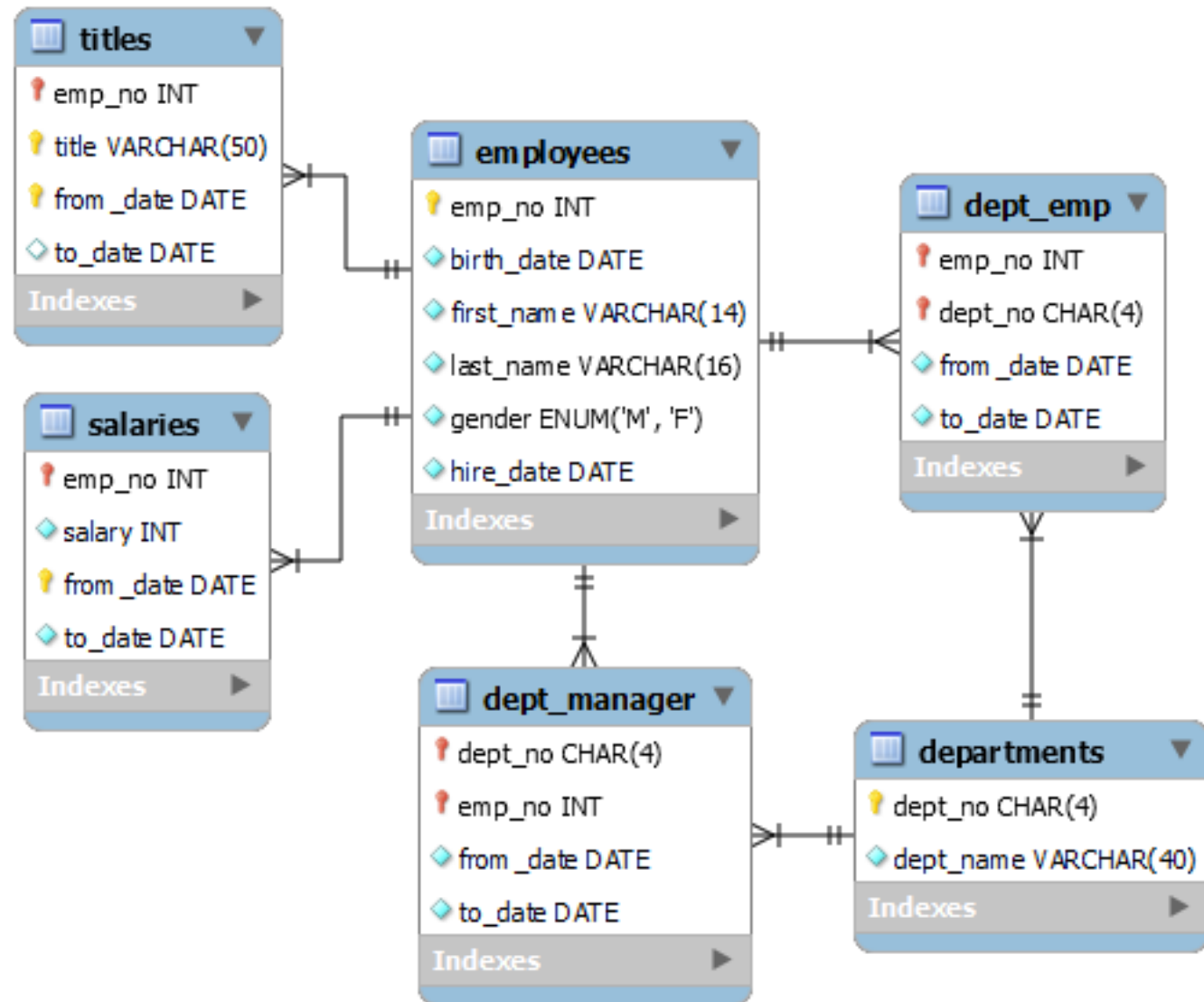
2025년 상반기 K-디지털 트레이닝

# 뷰 (심화2)

---

[KB] IT's Your Life

- 모든 문제는 employees 데이터베이스에서 수행한다.
- employees db의 ERD



✓ 다음 정보를 가지는 직원 정보를 출력하는 EMPLOYEES\_INFO 뷰를 작성하세요

```
CREATE OR REPLACE VIEW EMPLOYEES_INFO
AS
SELECT e.*,
t.title, t.from_date t_from, t.to_date t_to,
s.salary, s.from_date s_from, s.to_date s_to
FROM employees e
INNER JOIN titles t
ON e.emp_no = t.emp_no
INNER JOIN salaries s
ON e.emp_no = s.emp_no;
```

	emp_no	birth_date	first_name	last_name	gender	hire_date	title	t_from	t_to	salary	s_from	s_to
▶	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	60117	1986-06-26	1987-06-26
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	62102	1987-06-26	1988-06-25
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	66074	1988-06-25	1989-06-25
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	66596	1989-06-25	1990-06-25
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	66961	1990-06-25	1991-06-25
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	71046	1991-06-25	1992-06-24
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	74333	1992-06-24	1993-06-24
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	75286	1993-06-24	1994-06-24
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	75994	1994-06-24	1995-06-24
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	76884	1995-06-24	1996-06-23
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	80013	1996-06-23	1997-06-23
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	81025	1997-06-23	1998-06-23
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	81097	1998-06-23	1999-06-23
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	84917	1999-06-23	2000-06-22
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	85112	2000-06-22	2001-06-22
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	85097	2001-06-22	2002-06-22
	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	88958	2002-06-22	9999-01-01

✓ EMPLOYEES\_INFO 뷰에서 재직자의 현재 정보만 출력하세요.

```
SELECT *
FROM EMPLOYEES_INFO
WHERE s_to = '9999-01-01';
```

	emp_no	birth_date	first_name	last_name	gender	hire_date	title	t_from	t_to	salary	s_from	s_to
▶	10001	1953-09-02	Georgi	Facello	M	1986-06-26	Senior Engineer	1986-06-26	9999-01-01	88958	2002-06-22	9999-01-01
	10002	1964-06-02	Bezalel	Simmel	F	1985-11-21	Staff	1996-08-03	9999-01-01	72527	2001-08-02	9999-01-01
	10003	1959-12-03	Parto	Bamford	M	1986-08-28	Senior Engineer	1995-12-03	9999-01-01	43311	2001-12-01	9999-01-01
	10004	1954-05-01	Chirstian	Koblick	M	1986-12-01	Engineer	1986-12-01	1995-12-01	74057	2001-11-27	9999-01-01
	10004	1954-05-01	Chirstian	Koblick	M	1986-12-01	Senior Engineer	1995-12-01	9999-01-01	74057	2001-11-27	9999-01-01
	10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12	Senior Staff	1996-09-12	9999-01-01	94692	2001-09-09	9999-01-01
	10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12	Staff	1989-09-12	1996-09-12	94692	2001-09-09	9999-01-01
	10006	1953-04-20	Anneke	Preusig	F	1989-06-02	Senior Engineer	1990-08-05	9999-01-01	59755	2001-08-02	9999-01-01
	10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10	Senior Staff	1996-02-11	9999-01-01	88070	2002-02-07	9999-01-01
	10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10	Staff	1989-02-10	1996-02-11	88070	2002-02-07	9999-01-01

✓ 다음 정보를 가지는 부서 정보를 출력하는 EMP\_DEPT\_INFO 뷰를 작성하세요

```
CREATE OR REPLACE VIEW EMP_DEPT_INFO
AS
SELECT e.emp_no, d.dept_no, d.dept_name, de.from_date, de.to_date
FROM departments d
    INNER JOIN dept_emp de
        ON d.dept_no = de.dept_no
    INNER JOIN employees e
        ON de.emp_no = e.emp_no;
```

	emp_no	dept_no	dept_name	from_date	to_date
▶	10001	d005	Development	1986-06-26	9999-01-01
	10002	d007	Sales	1996-08-03	9999-01-01
	10003	d004	Production	1995-12-03	9999-01-01
	10004	d004	Production	1986-12-01	9999-01-01
	10005	d003	Human Resources	1989-09-12	9999-01-01
	10006	d005	Development	1990-08-05	9999-01-01
	10007	d008	Research	1989-02-10	9999-01-01



✓ EMP\_DEPT\_INFO로 현재 재직자의 부서 정보를 출력하세요.

```
SELECT *  
FROM EMP_DEPT_INFO  
WHERE to_date = '9999-01-01'
```

	emp_no	dept_no	dept_name	from_date	to_date
▶	10001	d005	Development	1986-06-26	9999-01-01
	10002	d007	Sales	1996-08-03	9999-01-01
	10003	d004	Production	1995-12-03	9999-01-01
	10004	d004	Production	1986-12-01	9999-01-01
	10005	d003	Human Resources	1989-09-12	9999-01-01
	10006	d005	Development	1990-08-05	9999-01-01
	10007	d008	Research	1989-02-10	9999-01-01