커피 판매 관리 시스템

ICT학부
정보보호학
20161267

김동준

목차

1. 데이터베이스의 생성

2. 테이블과 데이터의 생성

3. 생성한 테이블로 실습

4. 실습 및 오류 검출

5. 실습하면서 느낀 점

1.데이터베이스의 생성

실습을 진행하기에 앞서 먼저 데이터베이스를 생성시키기 위해 데이터베이스의 파일 이름, 크기, 저장 위치를 정한 후 실습을 진행한다

SQLQuery1.sql - D...4R46₩mynote (52))*

```sql
CREATE DATABASE COFFEE ON PRIMARY
(
NAME = COFFEE,
FILENAME = 'C:₩WORK₩COFFEE.mdf',
SIZE = 4MB,
MAXSIZE = 8MB,
FILEGROWTH = 4MB
)
```

100 %

메시지

```
명령이 완료되었습니다.

완료 시간: 2021-11-26T11:51:41.0467967+09:00
```

2.테이블과 데이터의 생성

프로젝트에서 필요로 하는 테이블을 생성하고 그 테이블에 맞는 데이터를 입력하여 생성해주는 과정이 필요하다.

2-1.STAFF의 TABLE을 생성해준다.

SQLQuery1.sql - D...4R46₩mynote (69)) | SQLQuery2.sql - D...4

```sql
CREATE TABLE STAFF
(
S_CODE CHAR(5) PRIMARY KEY,
S_NAME CHAR(10) NOT NULL,
S_D_CODE CHAR(5) NOT NULL,
S_ZIPCODE CHAR(7) NOT NULL,
S_ADD CHAR(50) NOT NULL,
S_PHONE CHAR(13) NOT NULL
)
```

100 %

메시지

명령이 완료되었습니다.

완료 시간: 2021-11-26T12:53:22.2928077+09:00

2.테이블과 데이터의 생성

2-2.STAFF의 DATA를 생성해준다.

SQLQuery1.sql - D...4R46₩mynote (69)) | SQLQuery2.sql - D...4R46₩mynote (60)) | SQLQuery3.sql - D...4R46₩mynote

```sql
INSERT INTO STAFF VALUES('S001', '김동준', '사장', '111-111', '경기도 김포시 구래동', '010-1111-1111')
INSERT INTO STAFF VALUES('S002', '윤은비', '직원', '222-222', '경기도 김포시 구래동', '010-2222-2222')
INSERT INTO STAFF VALUES('S003', '이정웅', '알바1', '333-333', '경기도 김포시 사우동', '010-3333-3333')
INSERT INTO STAFF VALUES('S004', '이상혁', '알바2', '444-444', '경기도 김포시 걸포동', '010-4444-4444')

SELECT * FROM STAFF
```

100 %

결과 | 메시지

	S_CODE	S_NAME	S_D_CODE	S_ZIPCODE	S_ADD	S_PHONE
1	S001	김동준	사장	111-111	경기도 김포시 구래동	010-1111-1111
2	S002	윤은비	직원	222-222	경기도 김포시 구래동	010-2222-2222
3	S003	이정웅	알바1	333-333	경기도 김포시 사우동	010-3333-3333
4	S004	이상혁	알바2	444-444	경기도 김포시 걸포동	010-4444-4444

2.테이블과 데이터의 생성

2-3.MENU의 TABLE을 생성해준다.

SQLQuery1.sql - D...4R46₩mynote (63))*

```sql
CREATE TABLE MENU
(
M_CODE CHAR(5) PRIMARY KEY,
M_NAME VARCHAR(20) NOT NULL,
M_PRICE INT NOT NULL
)
```

100 %

메시지

명령이 완료되었습니다.

완료 시간: 2021-11-26T13:19:59.1424174+09:00

2.테이블과 데이터의 생성

2-4.MENU의 DATA를 생성해준다.

SQLQuery2.sql - D...4R46₩mynote (56))* | SQLQuery1.sql - D...4R46₩

```sql
INSERT INTO MENU VALUES('M001', '에스프레소', '2500')
INSERT INTO MENU VALUES('M002', '아메리카노', '3000')
INSERT INTO MENU VALUES('M003', '모카라떼', '3500')
INSERT INTO MENU VALUES('M004', '카푸치노', '4000')

SELECT * FROM MENU
```

100 %

결과 | 메시지

	M_CODE	M_NAME	M_PRICE
1	M001	에스프레소	2500
2	M002	아메리카노	3000
3	M003	모카라떼	3500
4	M004	카푸치노	4000

2.테이블과 데이터의 생성

2-5.GOOD_SELL(상품판매)의 TABLE을 생성해준다.

SQLQuery3.sql - D...4R46₩mynote (52))* | SQLQuery2.sql - D

```sql
CREATE TABLE GOOD_SELL
(
G_NUMBER INT,
G_M_CODE CHAR(5),
G_M_DATE SMALLDATETIME,
G_M_CAPACITY INT NOT NULL,
G_M_ACCOUNT CHAR(5) NOT NULL,
G_S_CODE CHAR(5) NOT NULL,
PRIMARY KEY(G_NUMBER, G_M_CODE, G_M_DATE)
)
```

100 %

메시지

```
명령이 완료되었습니다.

완료 시간: 2021-11-26T14:46:47.1075951+09:00
```

2.테이블과 데이터의 생성

2-6.GOOD_SELL(상품판매)의 DATA를 생성해준다.

SQLQuery1.sql - D...4R46₩mynote (73))*

```sql
INSERT INTO GOOD_SELL VALUES(1, 'M001', '2021-11-11', 2, '현금', 'S002')
INSERT INTO GOOD_SELL VALUES(2, 'M001', '2021-11-11', 1, '현금', 'S002')
INSERT INTO GOOD_SELL VALUES(3, 'M002', '2021-11-15', 2, '현금', 'S002')
INSERT INTO GOOD_SELL VALUES(4, 'M003', '2021-11-15', 3, '카드', 'S002')
INSERT INTO GOOD_SELL VALUES(5, 'M004', '2021-11-16', 1, '현금', 'S003')
INSERT INTO GOOD_SELL VALUES(6, 'M002', '2021-11-16', 2, '현금', 'S003')

SELECT * FROM GOOD_SELL
```

100 %

결과 | 메시지

	G_NUMBER	G_M_CODE	G_M_DATE	G_M_CAPACITY	G_M_ACCOUNT	G_S_CODE
1	1	M001	2021-11-11 00:00:00	2	현금	S002
2	2	M001	2021-11-11 00:00:00	1	현금	S002
3	3	M002	2021-11-15 00:00:00	2	현금	S002
4	4	M003	2021-11-15 00:00:00	3	카드	S002
5	5	M004	2021-11-16 00:00:00	1	현금	S003
6	6	M002	2021-11-16 00:00:00	2	현금	S003

2.테이블과 데이터의 생성

2-7.PRODUCT(재고)의 TABLE을 생성해준다.

SQLQuery2.sql - D...4R46₩mynote (67))* | SQLQuery1.sql - D

```
CREATE TABLE PRODUCT
(
P_CODE CHAR(5) PRIMARY KEY,
P_NAME VARCHAR(20) NOT NULL,
P_PRICE INT NOT NULL
)
```

100 %

메시지

```
명령이 완료되었습니다.

완료 시간: 2021-11-26T17:55:34.0895272+09:00
```

2.테이블과 데이터의 생성

2-8.PRODUCT(재고)의 DATA를 생성해준다.

SQLQuery3.sql - D...4R46₩mynote (60))* SQLQuery2.sql - D...4R46₩myn

```sql
INSERT INTO PRODUCT VALUES('R001', '원두', '15000')
INSERT INTO PRODUCT VALUES('R002', '코코아분말', '5000')
INSERT INTO PRODUCT VALUES('R003', '라떼시럽', '5000')
INSERT INTO PRODUCT VALUES('R004', '우유', '1500')
INSERT INTO PRODUCT VALUES('W001', '종이컵', '1000')
INSERT INTO PRODUCT VALUES('W002', '스트로우', '1000')
INSERT INTO PRODUCT VALUES('W003', '휴지', '1000')

SELECT * FROM PRODUCT
```

100 %

결과 | 메시지

	P_CODE	P_NAME	P_PRICE
1	R001	원두	15000
2	R002	코코아분말	5000
3	R003	라떼시럽	5000
4	R004	우유	1500
5	W001	종이컵	1000
6	W002	스트로우	1000
7	W003	휴지	1000

2.테이블과 데이터의 생성

2-9.COMPANY(납품업체)의 TABLE을 생성해준다.

```
CREATE TABLE COMPANY
(
C_CODE CHAR(5) PRIMARY KEY,
C_NAME CHAR(20) NOT NULL,
C_PERSON CHAR(20) NOT NULL,
C_ZIPCODE CHAR(7) NOT NULL,
C_ADD CHAR(50) NOT NULL,
C_PHONE CHAR(20) NOT NULL
)
```

100 %

메시지

```
명령이 완료되었습니다.

완료 시간: 2021-11-27T13:21:56.8779299+09:00
```

2.테이블과 데이터의 생성

2-10.COMPANY(납품업체)의 DATA를 생성해준다.

SQLQuery6.sql - D...4R46₩mynote (70))* | SQLQuery7.sql - D...4R46₩mynote (72))*

```sql
INSERT INTO COMPANY VALUES('C001', '(주)서울유통', '신석연', '182-025', '서울시 강남구 양재동', '02-1577-1577')
INSERT INTO COMPANY VALUES('C002', '(주)남양우유', '이인호', '168-034', '서울시 강서구 신월동', '02-1588-0234')
INSERT INTO COMPANY VALUES('C003', '(주)세종', '정민정', '168-025', '서울시 강남구 화곡동', '02-1566-1123')

SELECT * FROM COMPANY
```

100 %

결과 | 메시지

	C_CODE	C_NAME	C_PERSON	C_ZIPCODE	C_ADD	C_PHONE
1	C001	(주)서울유통	신석연	182-025	서울시 강남구 양재동	02-1577-1577
2	C002	(주)남양우유	이인호	168-034	서울시 강서구 신월동	02-1588-0234
3	C003	(주)세종	정민정	168-025	서울시 강남구 화곡동	02-1566-1123

2.테이블과 데이터의 생성

2-11.INPUT_PRODUCT(입고관리)의 TABLE을 생성해준다.

```sql
CREATE TABLE INPUT_PRODUCT
(I_NUMBER INT,
I_P_CODE CHAR(5),
I_P_DATE SMALLDATETIME,
I_P_CAPACITY FLOAT NOT NULL,
I_P_TODATE SMALLDATETIME NULL,
I_C_CODE CHAR(5) NOT NULL,
PRIMARY KEY(I_NUMBER, I_P_CODE, I_P_DATE)
)
```

00 %

메시지

명령이 완료되었습니다.

완료 시간: 2021-11-26T19:29:39.8767035+09:00

2.테이블과 데이터의 생성

2-12.INPUT_PRODUCT(입고관리)의 DATA를 생성해준다.

```sql
INSERT INTO INPUT_PRODUCT VALUES(1, 'R001', '2021-11-11', 2, '2021-11-26', 'C001')
INSERT INTO INPUT_PRODUCT VALUES(2, 'R002', '2021-11-11', 2, '2021-11-26', 'C001')
INSERT INTO INPUT_PRODUCT VALUES(3, 'R003', '2021-11-11', 2, '2021-11-26', 'C001')
INSERT INTO INPUT_PRODUCT VALUES(4, 'R004', '2021-11-11', 10, '2021-11-26', 'C001')
INSERT INTO INPUT_PRODUCT VALUES(5, 'W001', '2021-11-11', 200, '2021-11-26', 'C002')
INSERT INTO INPUT_PRODUCT VALUES(6, 'W002', '2021-11-11', 200, '2021-11-26', 'C002')
INSERT INTO INPUT_PRODUCT VALUES(7, 'W003', '2021-11-11', 200, '2021-11-26', 'C002')

SELECT * FROM INPUT_PRODUCT
```

100 %

결과 메시지

	I_NUMBER	I_P_CODE	I_P_DATE	I_P_CAPACITY	I_P_TODATE	I_C_CODE
1	1	R001	2021-11-11 00:00:00	2	2021-11-26 00:00:00	C001
2	2	R002	2021-11-11 00:00:00	2	2021-11-26 00:00:00	C001
3	3	R003	2021-11-11 00:00:00	2	2021-11-26 00:00:00	C001
4	4	R004	2021-11-11 00:00:00	10	2021-11-26 00:00:00	C001
5	5	W001	2021-11-11 00:00:00	200	2021-11-26 00:00:00	C002
6	6	W002	2021-11-11 00:00:00	200	2021-11-26 00:00:00	C002
7	7	W003	2021-11-11 00:00:00	200	2021-11-26 00:00:00	C002

2.테이블과 데이터의 생성

2-13.USE_PRODUCT(재고사용관리)의 TABLE을 생성해준다.

```sql
CREATE TABLE USE_PRODUCT
(
U_P_CODE CHAR(5),
U_P_DATE SMALLDATETIME,
U_P_CAPACITY FLOAT NOT NULL,
U_S_CODE CHAR(5) NOT NULL,
PRIMARY KEY(U_P_CODE, U_P_DATE)
)
```

100 %

메시지

```
명령이 완료되었습니다.

완료 시간: 2021-11-26T19:46:17.3212974+09:00
```

2.테이블과 데이터의 생성

2-14.USE_PRODUCT(재고사용관리)의 DATA를 생성해준다.

```sql
INSERT INTO USE_PRODUCT VALUES('R001', '2021-11-11', 1, 'S001')
INSERT INTO USE_PRODUCT VALUES('R002', '2021-11-11', 1, 'S001')
INSERT INTO USE_PRODUCT VALUES('R003', '2021-11-11', 1, 'S001')
INSERT INTO USE_PRODUCT VALUES('R004', '2021-11-11', 5, 'S001')
INSERT INTO USE_PRODUCT VALUES('W002', '2021-11-11', 100, 'S001')
INSERT INTO USE_PRODUCT VALUES('W003', '2021-11-11', 100, 'S001')

SELECT * FROM USE_PRODUCT
```

.00 %

결과 메시지

	U_P_CODE	U_P_DATE	U_P_CAPACITY	U_S_CODE
1	R001	2021-11-11 00:00:00	1	S001
2	R002	2021-11-11 00:00:00	1	S001
3	R003	2021-11-11 00:00:00	1	S001
4	R004	2021-11-11 00:00:00	5	S001
5	W002	2021-11-11 00:00:00	100	S001
6	W003	2021-11-11 00:00:00	100	S001

3.생성한 테이블로 실습

지금까지 만들어놓은 커피판매관리시스템으로 데이터베이스를 통한 실습을 진행해보도록 한다.

3-1.상품인기도 파악을 할 수 있는 실습을 진행해보았다.

```sql
SELECT M_CODE AS 상품코드, M_NAME AS 상품명, CONVERT(CHAR(6), CAST(M_PRICE AS CHAR(4)), 1) + '원' AS단가,
CONVERT(CHAR(4),CAST(SUM(G_M_CAPACITY) AS CHAR(2)), 1) + '개' AS 총판매개수,
CONVERT(CHAR(8),CAST(M_PRICE * SUM(G_M_CAPACITY) AS CHAR(6)), 1) + '원' AS 판매가격
FROM MENU, GOOD_SELL
WHERE M_CODE = G_M_CODE
GROUP BY M_CODE, M_NAME, M_PRICE

SELECT * FROM GOOD_SELL
SELECT * FROM MENU
```

100 %

결과 | 메시지

	상품코드	상품명	AS단가	총판매개수	판매가격
1	M001	에스프레소	2500 원	3 개	7500 원
2	M002	아메리카노	3000 원	4 개	12000 원
3	M003	모카라떼	3500 원	3 개	10500 원
4	M004	카푸치노	4000 원	1 개	4000 원

	G_NUMBER	G_M_CODE	G_M_DATE	G_M_CAPACITY	G_M_ACCOUNT	G_S_CODE
1	1	M001	2021-11-11 00:00:00	2	현금	S002
2	2	M001	2021-11-11 00:00:00	1	현금	S002
3	3	M002	2021-11-15 00:00:00	2	현금	S002
4	4	M003	2021-11-15 00:00:00	3	카드	S002
5	5	M004	2021-11-16 00:00:00	1	현금	S003
6	6	M002	2021-11-16 00:00:00	2	현금	S003

	M_CODE	M_NAME	M_PRICE
1	M001	에스프레소	2500
2	M002	아메리카노	3000
3	M003	모카라떼	3500
4	M004	카푸치노	4000

3.생성한 테이블로 실습

3-2.하룻동안 판매한 제품을 결산하는 일일결산 실습을 진행해보았다.

```sql
CREATE VIEW TOTAL_DAY(판매날짜, 상품코드, 상품별총가격)
AS
SELECT TOP 100 PERCENT G_M_DATE, M_CODE, SUM(M_PRICE*G_M_CAPACITY) AS 일일정산
FROM GOOD_SELL, MENU
WHERE M_CODE = G_M_CODE
GROUP BY G_M_DATE, M_PRICE, M_CODE
```

100 %

메시지

명령이 완료되었습니다.

완료 시간: 2021-11-27T11:00:54.8507824+09:00

```sql
SELECT G_M_DATE AS 판매날짜, M_CODE AS 상품코드, SUM(M_PRICE * G_M_CAPACITY) AS 상품별총가격
FROM GOOD_SELL, MENU
WHERE M_CODE = G_M_CODE
GROUP BY G_M_DATE, M_PRICE, M_CODE
ORDER BY G_M_DATE ASC

SELECT CONVERT(CHAR(11), 판매날짜, 102) AS 판매날짜,
CONVERT(CHAR(8), CAST(SUM(상품별총가격) AS CHAR(6)), 1) + '원' AS 일일정산
FROM TOTAL_DAY
GROUP BY 판매날짜

SELECT * FROM MENU
SELECT * FROM GOOD_SELL
```

100 %

결과 | 메시지

	판매날짜	상품코드	상품별총가격
1	2021-11-11 00:00:00	M001	7500
2	2021-11-15 00:00:00	M002	6000
3	2021-11-15 00:00:00	M003	10500
4	2021-11-16 00:00:00	M002	6000
5	2021-11-16 00:00:00	M004	4000

	판매날짜	일일정산
1	2021.11.11	7500 원
2	2021.11.15	16500 원
3	2021.11.16	10000 원

	M_CODE	M_NAME	M_PRICE
1	M001	에스프레소	2500
2	M002	아메리카노	3000
3	M003	모카라떼	3500
4	M004	카푸치노	4000

	G_NUMBER	G_M_CODE	G_M_DATE	G_M_CAPACITY	G_M_ACCOUNT	G_S_CODE
1	1	M001	2021-11-11 00:00:00	2	현금	S002
2	2	M001	2021-11-11 00:00:00	1	현금	S002
3	3	M002	2021-11-15 00:00:00	2	현금	S002
4	4	M003	2021-11-15 00:00:00	3	카드	S002
5	5	M004	2021-11-16 00:00:00	1	현금	S003
6	6	M002	2021-11-16 00:00:00	2	현금	S003

3.생성한 테이블로 실습

3-3.재고관리 실습을 진행해보았다

```
SELECT P_CODE AS 재고코드, P_NAME AS 재고명,
CONVERT(VARCHAR(10), CAST(SUM(I_P_CAPACITY) AS CHAR(4)), 1) + '개' AS 출입고량,
CONVERT(VARCHAR(10), CAST(SUM(U_P_CAPACITY) AS CHAR(4)), 1) + '개' AS 총사용량,
CONVERT(VARCHAR(10), CAST(SUM(I_P_CAPACITY) - SUM(U_P_CAPACITY) AS CHAR(4)), 1) + '개' AS 남은량
FROM PRODUCT, INPUT_PRODUCT, USE_PRODUCT
WHERE P_CODE = I_P_CODE AND P_CODE = U_P_CODE
GROUP BY P_CODE, P_NAME, P_PRICE

SELECT * FROM PRODUCT
SELECT * FROM INPUT_PRODUCT
SELECT * FROM USE_PRODUCT
```

100 %

결과 | 메시지

	재고코드	재고명	출입고량	총사용량	남은량
1	R001	원두	2 개	1 개	1 개
2	R002	코코아분말	2 개	1 개	1 개
3	R003	라떼시럽	2 개	1 개	1 개
4	R004	우유	10 개	5 개	5 개
5	W002	스트로우	200 개	100 개	100 개
6	W003	휴지	200 개	100 개	100 개

	P_CODE	P_NAME	P_PRICE
1	R001	원두	15000
2	R002	코코아분말	5000
3	R003	라떼시럽	5000
4	R004	우유	1500
5	W001	종이컵	1000
6	W002	스트로우	1000
7	W003	휴지	1000

	I_NUMBER	I_P_CODE	I_P_DATE	I_P_CAPACITY	I_P_TODATE	I_C_CODE
1	1	R001	2021-11-11 00:00:00	2	2021-11-26 00:00:00	C001
2	2	R002	2021-11-11 00:00:00	2	2021-11-26 00:00:00	C001
3	3	R003	2021-11-11 00:00:00	2	2021-11-26 00:00:00	C001
4	4	R004	2021-11-11 00:00:00	10	2021-11-26 00:00:00	C001
5	5	W001	2021-11-11 00:00:00	200	2021-11-26 00:00:00	C002
6	6	W002	2021-11-11 00:00:00	200	2021-11-26 00:00:00	C002
7	7	W003	2021-11-11 00:00:00	200	2021-11-26 00:00:00	C002

	U_P_CODE	U_P_DATE	U_P_CAPACITY	U_S_CODE
1	R001	2021-11-11 00:00:00	1	S001
2	R002	2021-11-11 00:00:00	1	S001
3	R003	2021-11-11 00:00:00	1	S001
4	R004	2021-11-11 00:00:00	5	S001
5	W002	2021-11-11 00:00:00	100	S001
6	W003	2021-11-11 00:00:00	100	S001

3.생성한 테이블로 실습

3-4.재고 관리자를 확인하는 실습을 진행해보았다.

```
SELECT P_CODE AS 재고코드, P_NAME AS 재고명,
CONVERT(VARCHAR(10), CAST(SUM(I_P_CAPACITY) AS CHAR(4)), 1) + '개' AS 총입고량,
CONVERT(VARCHAR(10), CAST(SUM(U_P_CAPACITY) AS CHAR(4)), 1) + '개' AS 총사용량,
CONVERT(VARCHAR(10), CAST(SUM(I_P_CAPACITY) - SUM(U_P_CAPACITY) AS CHAR(4)),1) + '개' AS 재고량,
STAFF.S_NAME AS 사원명, STAFF.S_D_CODE AS 직급
FROM PRODUCT, INPUT_PRODUCT, USE_PRODUCT, STAFF
WHERE P_CODE = I_P_CODE AND P_CODE = U_P_CODE AND USE_PRODUCT.U_S_CODE = STAFF.S_CODE
GROUP BY P_CODE, P_NAME, P_PRICE, S_NAME, S_D_CODE

SELECT * FROM STAFF
SELECT * FROM INPUT_PRODUCT
SELECT * FROM USE_PRODUCT
```

결과 | 메시지

	재고코드	재고명	총입고량	총사용량	재고량	사원명	직급
1	R001	원두	2 개	1 개	1 개	김동준	사장
2	R002	코코아분말	2 개	1 개	1 개	김동준	사장
3	R003	라떼시럽	2 개	1 개	1 개	김동준	사장
4	R004	우유	10 개	5 개	5 개	김동준	사장
5	W002	스트로우	200 개	100 개	100 개	김동준	사장
6	W003	휴지	200 개	100 개	100 개	김동준	사장

	S_CODE	S_NAME	S_D_CODE	S_ZIPCODE	S_ADD	S_PHONE
1	S001	김동준	사장	111-111	경기도 김포시 구래동	010-1111-1111
2	S002	윤은비	직원	222-222	경기도 김포시 구래동	010-2222-2222
3	S003	이정웅	알바1	333-333	경기도 김포시 사우동	010-3333-3333
4	S004	이상혁	알바2	444-444	경기도 김포시 걸포동	010-4444-4444

	I_NUMBER	I_P_CODE	I_P_DATE	I_P_CAPACITY	I_P_TODATE	I_C_CODE
1	1	R001	2021-11-11 00:00:00	2	2021-11-26 00:00:00	C001
2	2	R002	2021-11-11 00:00:00	2	2021-11-26 00:00:00	C001
3	3	R003	2021-11-11 00:00:00	2	2021-11-26 00:00:00	C001
4	4	R004	2021-11-11 00:00:00	10	2021-11-26 00:00:00	C001
5	5	W001	2021-11-11 00:00:00	200	2021-11-26 00:00:00	C002
6	6	W002	2021-11-11 00:00:00	200	2021-11-26 00:00:00	C002
7	7	W003	2021-11-11 00:00:00	200	2021-11-26 00:00:00	C002

	U_P_CODE	U_P_DATE	U_P_CAPACITY	U_S_CODE
1	R001	2021-11-11 00:00:00	1	S001
2	R002	2021-11-11 00:00:00	1	S001
3	R003	2021-11-11 00:00:00	1	S001
4	R004	2021-11-11 00:00:00	5	S001
5	W002	2021-11-11 00:00:00	100	S001
6	W003	2021-11-11 00:00:00	100	S001

4.실습 및 오류 검출

실습하면서 오류가 생긴 부분들을 검출하고 해결하였다.

4-1.COMPANY의 DATA를 입력 시 C_NAME과 C_PHONE의 길이가 짧아 오류가 난 것을 수정을 통해 해결하였다.

```sql
CREATE TABLE COMPANY
(
C_CODE CHAR(5) PRIMARY KEY,
C_NAME CHAR(10) NOT NULL,
C_PERSON CHAR(20) NOT NULL,
C_ZIPCODE CHAR(7) NOT NULL,
C_ADD CHAR(50) NOT NULL,
C_PHONE CHAR(11) NOT NULL
)
```

```
메시지
명령이 완료되었습니다.

완료 시간: 2021-11-27T13:18:45.9886906+09:00
```

```sql
INSERT INTO COMPANY VALUES('C001', '(주)서울유통', '신석연', '182-025', '서울시 강남구 양재동', '02-1577-1577')
INSERT INTO COMPANY VALUES('C002', '(주)남양우유', '이인호', '168-034', '서울시 강서구 신월동', '02-1588-0234')
INSERT INTO COMPANY VALUES('C003', '(주)세종', '정민정', '168-025', '서울시 강남구 화곡동', '02-1566-1123')

SELECT * FROM COMPANY
```

```
결과 메시지
메시지 2628, 수준 16, 상태 1, 줄 128
테이블 'master.dbo.COMPANY'의 열 'C_NAME'에 있는 문자열 또는 이진 데이터는 잘립니다. 잘린 값: '(주)서울유'.
문이 종료되었습니다.
메시지 2628, 수준 16, 상태 1, 줄 129
테이블 'master.dbo.COMPANY'의 열 'C_NAME'에 있는 문자열 또는 이진 데이터는 잘립니다. 잘린 값: '(주)남양우'.
문이 종료되었습니다.
메시지 2628, 수준 16, 상태 1, 줄 130
테이블 'master.dbo.COMPANY'의 열 'C_PHONE'에 있는 문자열 또는 이진 데이터는 잘립니다. 잘린 값: '02-1566-112'.
문이 종료되었습니다.

(0개 행이 영향을 받음)

완료 시간: 2021-11-27T13:17:19.2088753+09:00
```

```sql
ALTER TABLE COMPANY
ALTER COLUMN C_PHONE CHAR(20) NULL
```

```
메시지
명령이 완료되었습니다.

완료 시간: 2021-11-27T13:20:09.1951119+09:00
```

```sql
ALTER TABLE COMPANY
ALTER COLUMN C_NAME CHAR(20) NULL
```

```
메시지
명령이 완료되었습니다.

완료 시간: 2021-11-27T13:20:45.5492626+09:00
```

4.실습 및 오류 검출

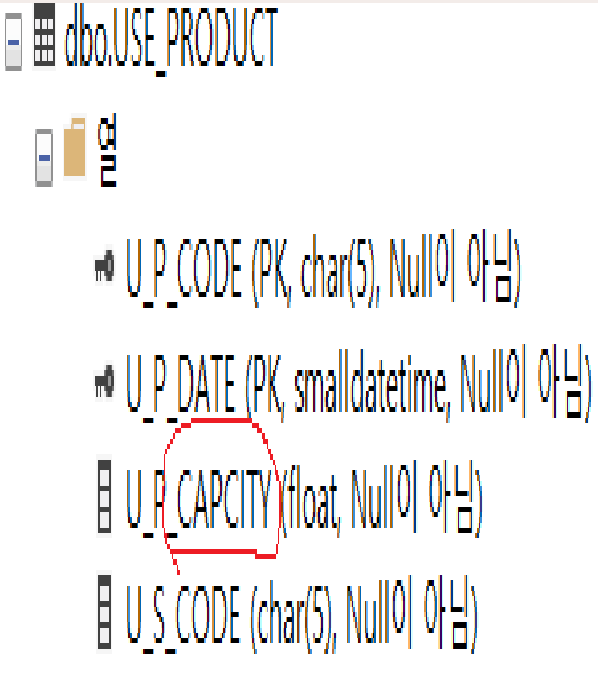
4-2.재고관리 실습 중 U_P_CAPACITY부분의 오류가나서 USE_PRODUCT의 테이블 작성해놓은 것을 확인해보니 오타가나서 실행시 오류가 나는 것을 확인하고 오타난 테이블의 이름을 수정하여서 오류를 해결하였다.

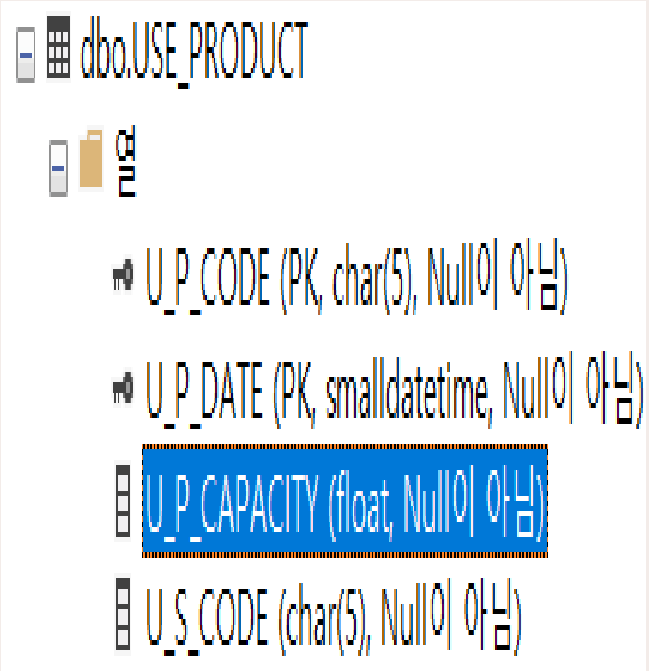
```
SELECT P_CODE AS 재고코드, P_NAME AS 재고명,
CONVERT(VARCHAR(10), CAST(SUM(I_P_CAPACITY) AS CHAR(4)), 1) + '개' AS 출입고량,
CONVERT(VARCHAR(10), CAST(SUM(U_P_CAPACITY) AS CHAR(4)), 1) + '개' AS 총사용량,
-CONVERT(VARCHAR(10), CAST(SUM(I_P_CAPACITY) - SUM(U_P_CAPACITY) AS CHAR(4)), 1) + '개' AS 남은량
FROM PRODUCT, INPUT_PRODUCT, USE_PRODUCT
WHERE P_CODE = I_P_CODE AND P_CODE = U_P_CODE
GROUP BY P_CODE, P_NAME, P_PRICE

SELECT * FROM PRODUCT
SELECT * FROM INPUT_PRODUCT
SELECT * FROM USE_PRODUCT
```

100 %

메시지

```
메시지 207, 수준 16, 상태 1, 줄 130
열 이름 'U_P_CAPACITY'이(가) 유효하지 않습니다.
메시지 207, 수준 16, 상태 1, 줄 131
열 이름 'U_P_CAPACITY'이(가) 유효하지 않습니다.

완료 시간: 2021-11-27T14:03:57.0934211+09:00
```





5.실습하면서 느낀 점

한군희 교수님의 데이터베이스 이론 및 실습 수업을 들으면서 느낀 점은 처음 수업을 듣기전에는 코로나19에 의한 비대면 수업으로 인해 실습에 대한 약간의 걱정과 두려움이 있었지만, 교수님의 같은 내용의 반복적인 강의와 중요한 부분은 재차 강조하면서 설명해주시는 부분으로 인해 이해하기가 수월하였고, 실습에 대한 부분도 충분한 자료와 중요한 부분의 반복된 설명을 통해 더 쉽고 흥미있게 진행할 수 있었습니다.

MS SQL을 통해 실습을 진행하면서 아주 작은 오류나 실수로인해 하루내내 고민하다 해결하지 못해 교수님께 연락하여 도움을 구했었는데 정말 어이없는 제 실수로 인해서 오류가 난 것을 확인하고 당황스럽고 부끄럽기도 하였었습니다.

그래도 실습 하면서 오류를 검출하고 해결하면서 뿌듯하였고, 실습에 대한 흥미도 생겨 재미있게 진행하였던 것 같습니다.

이 수업을 듣기 정말 잘한 것 같고 좋은 수업을 해주신 교수님께 감사드립니다.