



```
DROP TABLE BREAD CASCADE CONSTRAINT;
DROP TABLE CUSTOMER CASCADE CONSTRAINT;
DROP TABLE ORDER_LIST CASCADE CONSTRAINT;
```

```
-- 테이블 생성
CREATE TABLE BREAD (
  BREAD_ID CHAR(10) PRIMARY KEY,
  BREAD_NAME VARCHAR2(50),
  MADE_DATE DATE DEFAULT SYSDATE,
  COST NUMBER CHECK (COST > 0)
);
```

```
CREATE TABLE CUSTOMER (
  CUSTOMER_ID CHAR(5) PRIMARY KEY,
  CUSTOMER_NAME VARCHAR2(20),
  AGE NUMBER
);
```

```
CREATE TABLE ORDER_LIST (
  ORDER_ID CHAR(15) PRIMARY KEY,
  CUSTOMER_ID CHAR(5) REFERENCES CUSTOMER (CUSTOMER_ID),
  BREAD_ID CHAR(10) REFERENCES BREAD (BREAD_ID),
  ORDER_DATE DATE DEFAULT SYSDATE
);
```

```
-- 데이터 삽입
-- BREAD 테이블 데이터 삽입
INSERT INTO BREAD (BREAD_ID, BREAD_NAME, MADE_DATE, COST) VALUES ('B001', '크루아상', SYSDATE - 10, 3000);
INSERT INTO BREAD (BREAD_ID, BREAD_NAME, MADE_DATE, COST) VALUES ('B002', '베이글', SYSDATE - 9, 2500);
INSERT INTO BREAD (BREAD_ID, BREAD_NAME, MADE_DATE, COST) VALUES ('B003', '바게트', SYSDATE - 8, 3500);
INSERT INTO BREAD (BREAD_ID, BREAD_NAME, MADE_DATE, COST) VALUES ('B004', '사워도우', SYSDATE - 7, 4000);
INSERT INTO BREAD (BREAD_ID, BREAD_NAME, MADE_DATE, COST) VALUES ('B005', '치아바타', SYSDATE - 6, 4500);
INSERT INTO BREAD (BREAD_ID, BREAD_NAME, MADE_DATE, COST) VALUES ('B006', '포카치아', SYSDATE - 5, 2800);
INSERT INTO BREAD (BREAD_ID, BREAD_NAME, MADE_DATE, COST) VALUES ('B007', '호밀빵', SYSDATE - 4, 3200);
```

```
-- CUSTOMER 테이블 데이터 삽입
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, AGE) VALUES ('C001', '김철수', 25);
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, AGE) VALUES ('C002', '박영희', 32);
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, AGE) VALUES ('C003', '이민호', 28);
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, AGE) VALUES ('C004', '최수지', 24);
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, AGE) VALUES ('C005', '정다혜', 30);
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, AGE) VALUES ('C006', '한지민', 26);
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, AGE) VALUES ('C007', '유재석', 40);
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, AGE) VALUES ('C008', '강호동', 42);
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, AGE) VALUES ('C009', '신동엽', 38);
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, AGE) VALUES ('C010', '이효리', 35);
```

```
-- ORDER_LIST 테이블 데이터 삽입
INSERT INTO ORDER_LIST (ORDER_ID, CUSTOMER_ID, BREAD_ID, ORDER_DATE) VALUES ('O001', 'C001', 'B001', SYSDATE - 7);
INSERT INTO ORDER_LIST (ORDER_ID, CUSTOMER_ID, BREAD_ID, ORDER_DATE) VALUES ('O002', 'C002', 'B002', SYSDATE - 6);
INSERT INTO ORDER_LIST (ORDER_ID, CUSTOMER_ID, BREAD_ID, ORDER_DATE) VALUES ('O003', 'C003', 'B003', SYSDATE - 5);
INSERT INTO ORDER_LIST (ORDER_ID, CUSTOMER_ID, BREAD_ID, ORDER_DATE) VALUES ('O004', 'C004', 'B004', SYSDATE - 4);
INSERT INTO ORDER_LIST (ORDER_ID, CUSTOMER_ID, BREAD_ID, ORDER_DATE) VALUES ('O005', 'C005', 'B005', SYSDATE - 3);
INSERT INTO ORDER_LIST (ORDER_ID, CUSTOMER_ID, BREAD_ID, ORDER_DATE) VALUES ('O006', 'C006', 'B006', SYSDATE - 2);
INSERT INTO ORDER_LIST (ORDER_ID, CUSTOMER_ID, BREAD_ID, ORDER_DATE) VALUES ('O007', 'C007', 'B007', SYSDATE - 1);
INSERT INTO ORDER_LIST (ORDER_ID, CUSTOMER_ID, BREAD_ID, ORDER_DATE) VALUES ('O008', 'C008', 'B001', SYSDATE - 6);
INSERT INTO ORDER_LIST (ORDER_ID, CUSTOMER_ID, BREAD_ID, ORDER_DATE) VALUES ('O009', 'C009', 'B002', SYSDATE - 5);
INSERT INTO ORDER_LIST (ORDER_ID, CUSTOMER_ID, BREAD_ID, ORDER_DATE) VALUES ('O010', 'C010', 'B003', SYSDATE - 4);
```

이상원  
유연  
손현수  
손찬우

## BREAD

BREAD_ID	BREAD_NAME	MADE_DATE	COST
1 B001	크루아상	24/11/16	3000
2 B002	베이글	24/11/17	2500
3 B003	바게트	24/11/18	3500
4 B004	사워도우	24/11/19	4000
5 B005	치아바타	24/11/20	4500
6 B006	포카치아	24/11/21	2800
7 B007	호밀빵	24/11/22	3200

## CUSTOMER

CUSTOMER_ID	CUSTOMER_NAME	AGE
1 C001	김철수	25
2 C002	박영희	32
3 C003	이민호	28
4 C004	최수지	24
5 C005	정다혜	30
6 C006	한지민	26
7 C007	유재석	40
8 C008	강호동	42
9 C009	신동엽	38
10 C010	이효리	35

## ORDER\_LIST

ORDER_ID	CUSTOMER_ID	BREAD_ID	ORDER_DATE
1 O001	C001	B001	24/11/19
2 O002	C002	B002	24/11/20
3 O003	C003	B003	24/11/21
4 O004	C004	B004	24/11/22
5 O005	C005	B005	24/11/23
6 O006	C006	B006	24/11/24
7 O007	C007	B007	24/11/25
8 O008	C008	B001	24/11/20
9 O009	C009	B002	24/11/21
10 O010	C010	B003	24/11/22

1. 빵이름이 '바게트'인 빵의 가격을 찾아라.

```
SELECT COST FROM BREAD
WHERE BREAD_NAME='바게트';
```

	COST
1	3500

2. 3000원 이상 4000원 이하인 빵의 이름을 출력하시오.

```
SELECT BREAD_NAME
FROM BREAD
WHERE COST>=3000 AND COST<=4000;
```

	BREAD_NAME
1	크루아상
2	바게트
3	사워도우
4	호밀빵

3. 고객의 성만 나오고 나머지는 \* 로 표시하시오. (별칭은 개인정보)

```
SELECT CUSTOMER_ID, CONCAT(SUBSTR(CUSTOMER_NAME,1,
LENGTH(CUSTOMER_NAME)-2),'**') AS 개인정보
FROM CUSTOMER;
```

	CUSTOMER_ID	개인정보
1	C001	김**
2	C002	박**
3	C003	이**
4	C004	최**
5	C005	정**
6	C006	한**
7	C007	유**
8	C008	강**
9	C009	신**
10	C010	이**

4. 고객을 4개의 그룹으로 나누시오.(ROWNUM을 활용할 것, 별칭은 그룹으로 할 것)

```
SELECT CUSTOMER_ID, CUSTOMER_NAME, AGE, MOD(ROWNUM, 4) AS 그룹
FROM CUSTOMER;
```

	CUSTOMER_ID	CUSTOMER_NAME	AGE	그룹
1	C001	김철수	25	1
2	C002	박영희	32	2
3	C003	이민호	28	3
4	C004	최수지	24	0
5	C005	정다혜	30	1
6	C006	한지민	26	2
7	C007	유재석	40	3
8	C008	강호농	42	0
9	C009	신농엽	38	1
10	C010	이효리	35	2

## 함수

5. 나이가 30이상이면 OB로 30미만이면 YB로 해서 AGE\_GROUP이라는 열을 추가하시오.

```
SELECT CUSTOMER_ID, CUSTOMER_NAME, AGE,
CASE
    WHEN AGE >= 30 THEN 'OB'
    ELSE 'YB'
END AS "AGE_GROUP"
FROM CUSTOMER;
```

	CUSTOMER_ID	CUSTOMER_NAME	AGE	AGE_GROUP
1	C001	김철수	25	YB
2	C002	박영희	32	OB
3	C003	이민호	28	YB
4	C004	최수지	24	YB
5	C005	정다혜	30	OB
6	C006	한지민	26	YB
7	C007	유재석	40	OB
8	C008	강호동	42	OB
9	C009	신봉선	38	OB
10	C010	이효리	35	OB

6. BREAD 테이블에 INCREASED\_COST 라는 열을 추가하고 값은 COST에 500을 더한 값으로 채워넣으시오.

```
ALTER TABLE BREAD ADD INCREASED_COST NUMBER;
UPDATE BREAD
SET INCREASED_COST = COST + 500;
SELECT * FROM BREAD;
```

	BREAD_ID	BREAD_NAME	MADE_DATE	COST	INCREASED_COST
1	B001	크루아상	24/11/16	3000	3500
2	B002	베이글	24/11/17	2500	3000
3	B003	바게트	24/11/18	3500	4000
4	B004	사워도우	24/11/19	4000	4500
5	B005	치아바타	24/11/20	4500	5000
6	B006	포카치아	24/11/21	2800	3300
7	B007	호밀빵	24/11/22	3200	3700

7. BREAD테이블 MADE\_DATE열에서 DAY만 추출하시오.

```
SELECT BREAD_ID, BREAD_NAME, EXTRACT(DAY FROM MADE_DATE), COST
FROM BREAD;
```

	BREAD_ID	BREAD_NAME	EXTRACT(DAY FROM MADE_DATE)	COST
1	B001	크루아상	16	3000
2	B002	베이글	17	2500
3	B003	바게트	18	3500
4	B004	사워도우	19	4000
5	B005	치아바타	20	4500
6	B006	포카치아	21	2800
7	B007	호밀빵	22	3200

8. BREAD테이블의 COST데이터 뒤에 '원'을 붙이시오.

```
SELECT BREAD_ID, BREAD_NAME, MADE_DATE, CONCAT(COST, '원')
FROM BREAD;
```

	BREAD_ID	BREAD_NAME	MADE_DATE	CONCAT(COST, '원')
1	B001	크루아상	24/11/16	3000원
2	B002	베이글	24/11/17	2500원
3	B003	바게트	24/11/18	3500원
4	B004	사워도우	24/11/19	4000원
5	B005	치아바타	24/11/20	4500원
6	B006	포카치아	24/11/21	2800원
7	B007	호밀빵	24/11/22	3200원

9. (C011, '홍길동', 29) 이 데이터를 추가하고 SAVEPOINT A1을 설정하시오

```
INSERT INTO CUSTOMER (CUSTOMER_ID, CUSTOMER_NAME, AGE)
VALUES ('C011', '홍길동', 29);
SAVEPOINT A1;
```

10. ROLLBACK을 수행하시오.

```
ROLLBACK;
```

	CUSTOMER_ID	CUSTOMER_NAME	AGE
1	C001	김철수	25
2	C002	박영희	32
3	C003	이민호	28
4	C004	최수지	24
5	C005	정다혜	30
6	C006	한지민	26
7	C007	유재석	40
8	C008	강호동	42
9	C009	신동엽	38
10	C010	이효리	35
11	C011	홍길동	29

	CUSTOMER_ID	CUSTOMER_NAME	AGE
1	C001	김철수	25
2	C002	박영희	32
3	C003	이민호	28
4	C004	최수지	24
5	C005	정다혜	30
6	C006	한지민	26
7	C007	유재석	40
8	C008	강호동	42
9	C009	신동엽	38
10	C010	이효리	35