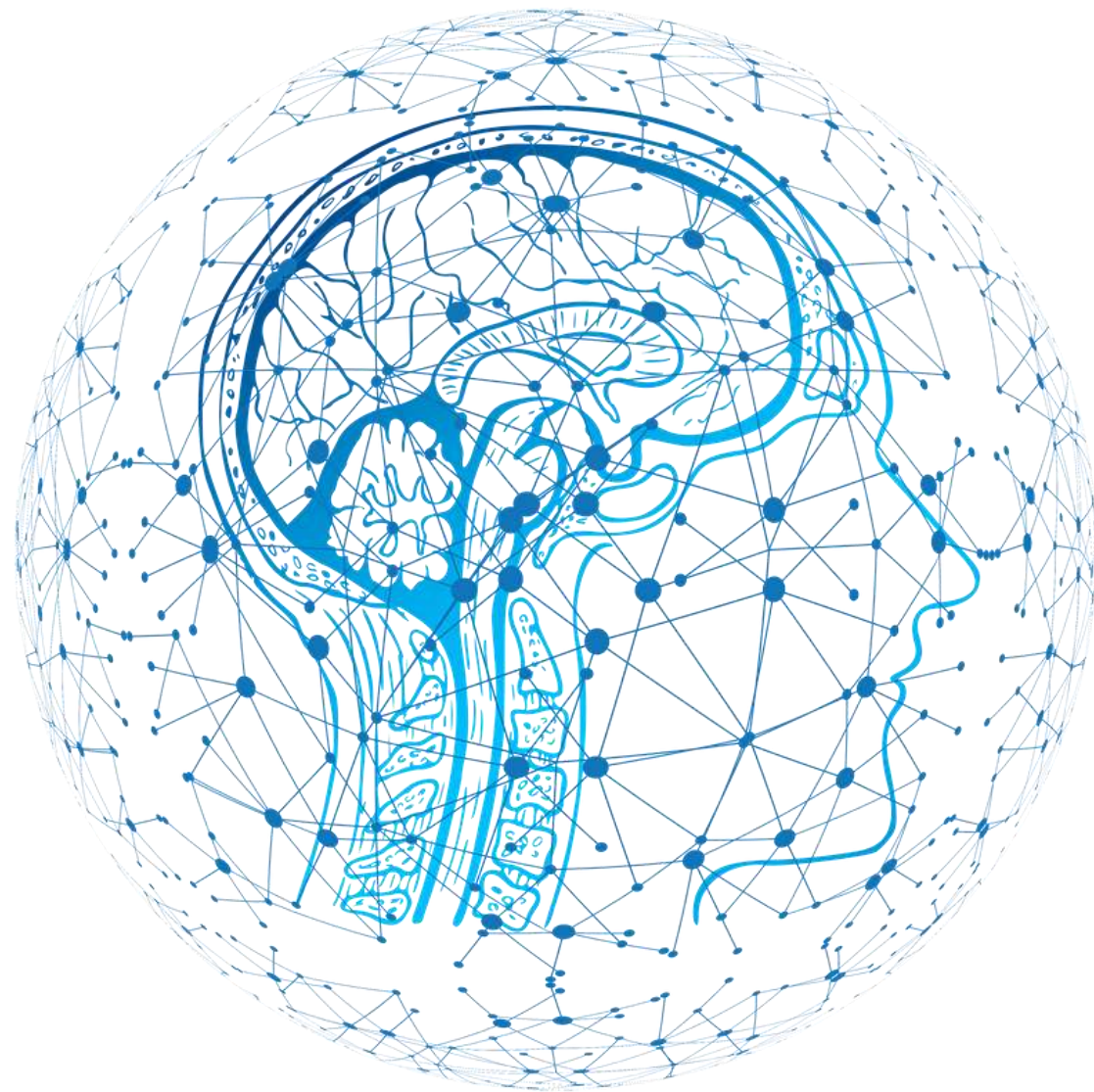


# 편의점 재고관리

팀 - 앓아씨 [정용억, 이현오, 장광수, 정인혁, 최효정]



# 목차

1

프로젝트 개요

2

팀 소개

3

구성원 소개

4

프로젝트 수행절차

5

프로젝트 수행결과

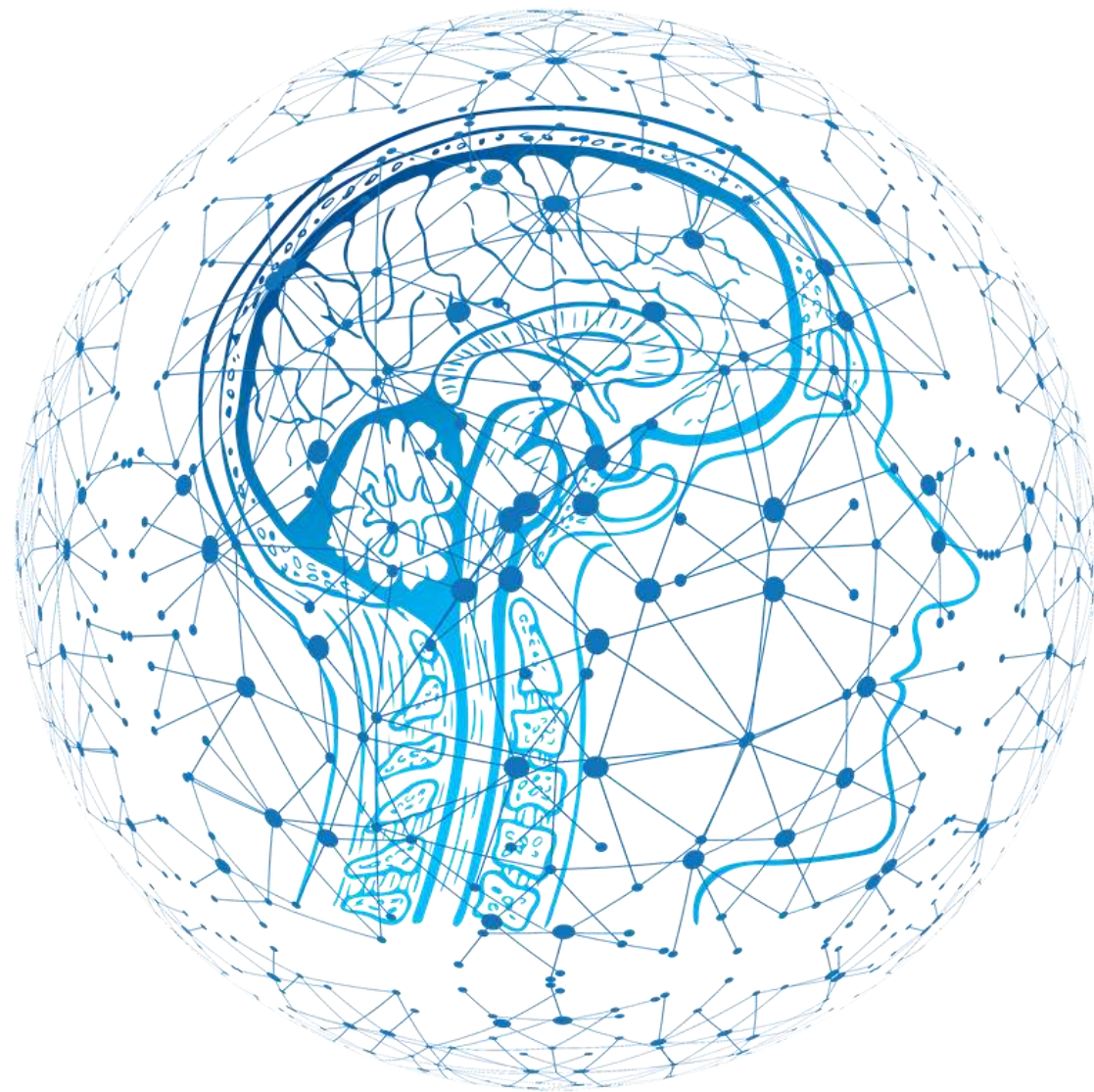
6

자체 평가의견



# 프로젝트 개요

---



# 프로젝트 개요

## 편의점 재고관리 데이터베이스

편의점 데이터 베이스를 구축하여 편의점 재고관리에 효율성을 높이고자 이번 프로젝트를 진행함.



# 팀 소개

---



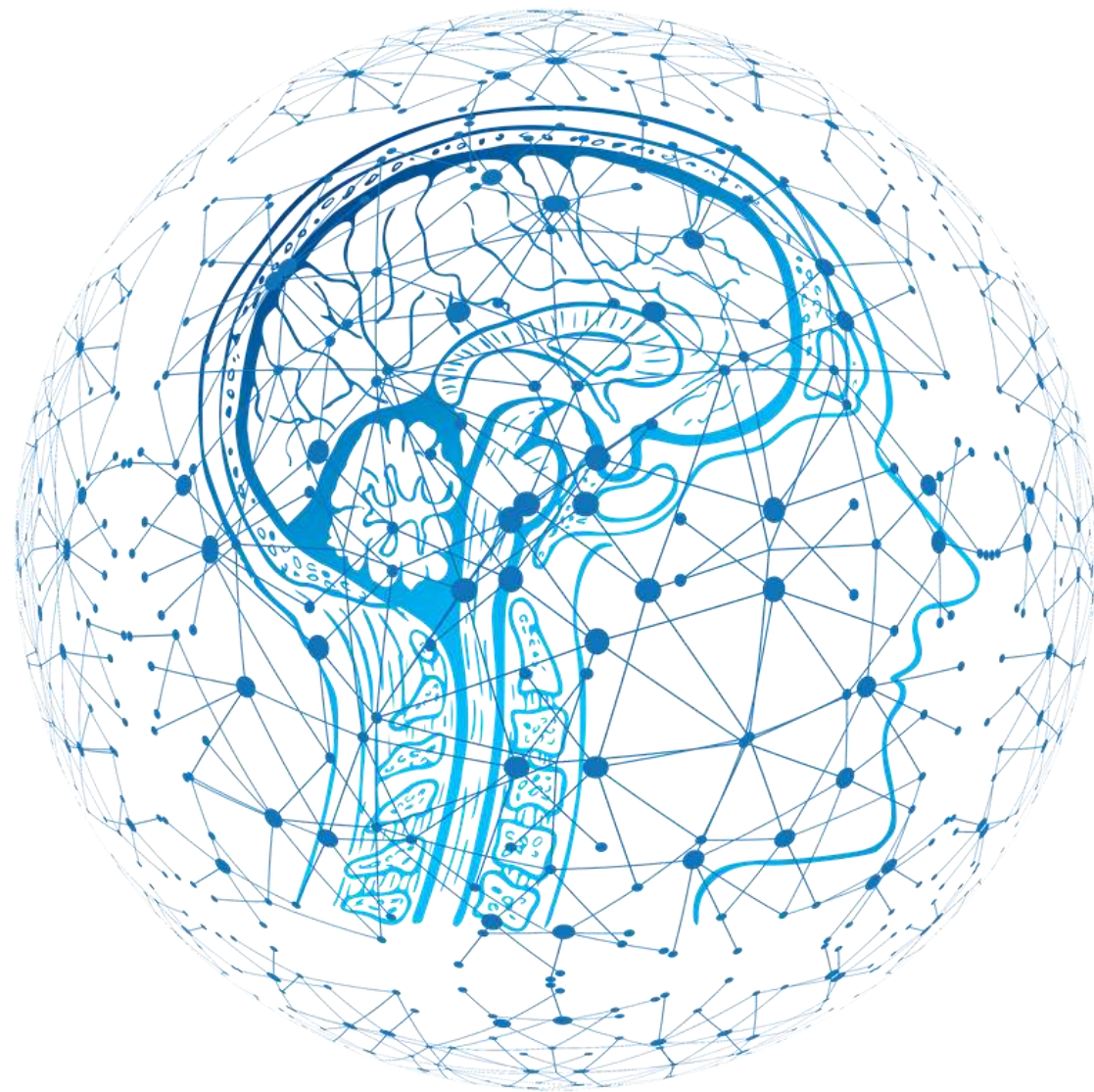
# 팀 소개

## - 앓 아 씨 -

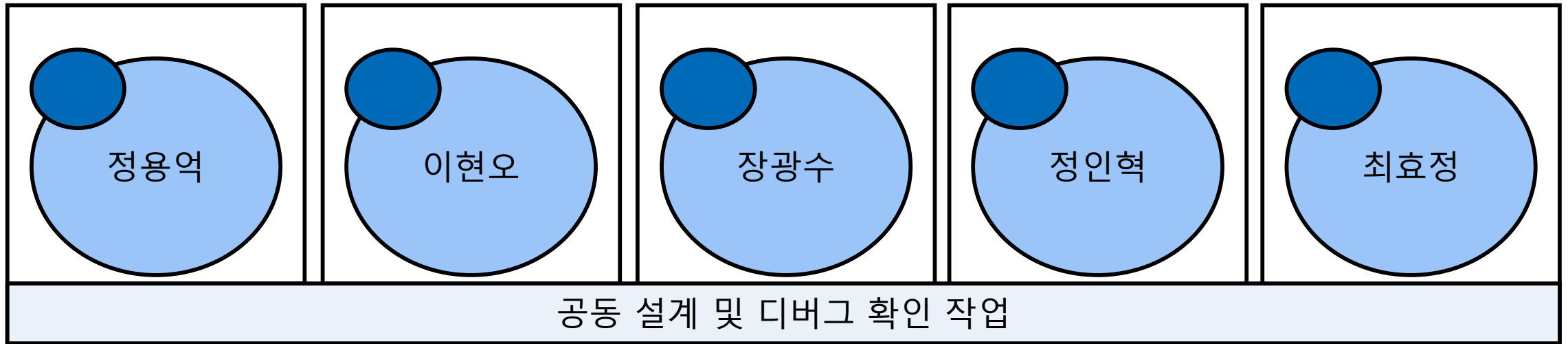
팀 ' 앓아씨 ' 는 코딩을 공부하고자 모인 **5**인이 만든 팀으로, 이 팀명은 ' 코딩을 하려면 앓아서 써야 한다 ' 라는 직관적인 아이디어에서 비롯되었습니다. 저희 팀은 코드 작성을 통해 프로젝트를 진행하고, 그 결과물을 통해 지식과 기술을 공유하며 성장하는 목표를 가지고 있습니다.

# 구성원 소개

---



# 구성원 소개



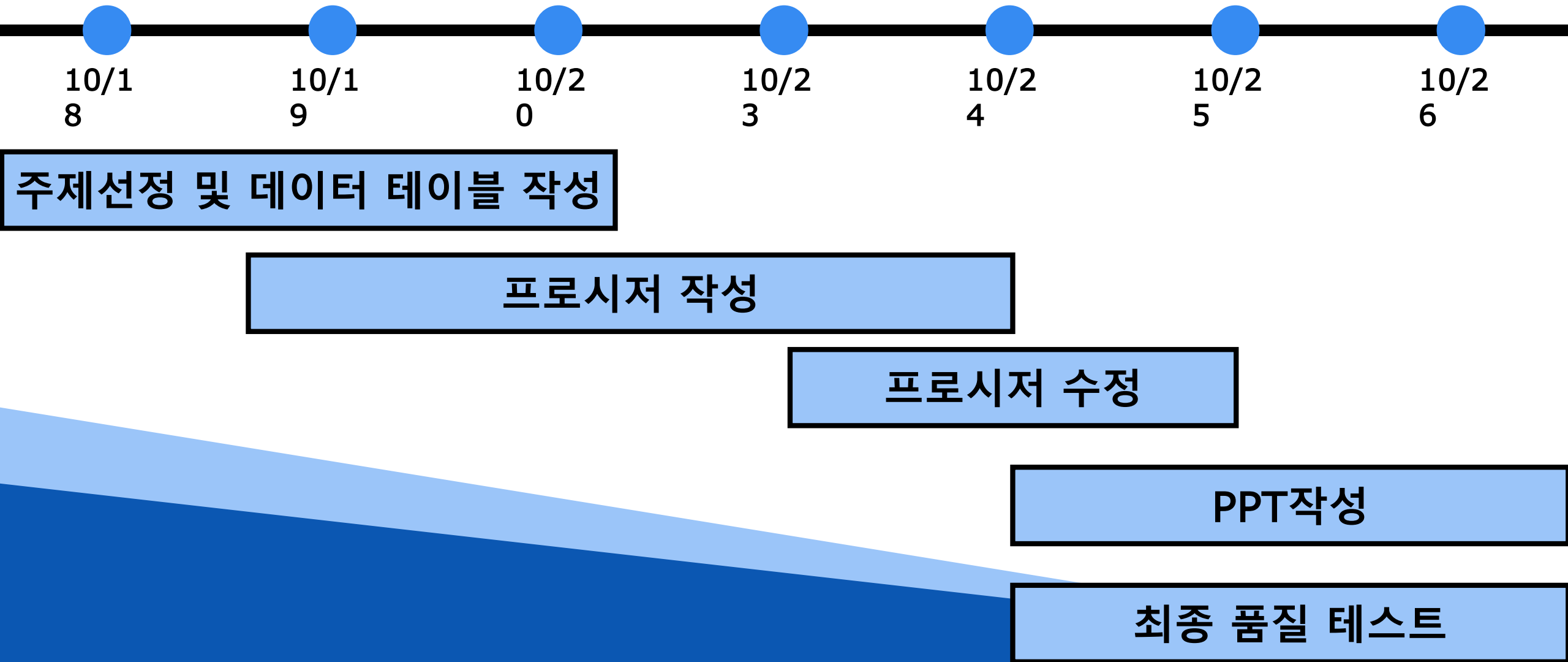


# 프로젝트 수행절차

---



# 프로젝트 수행절차



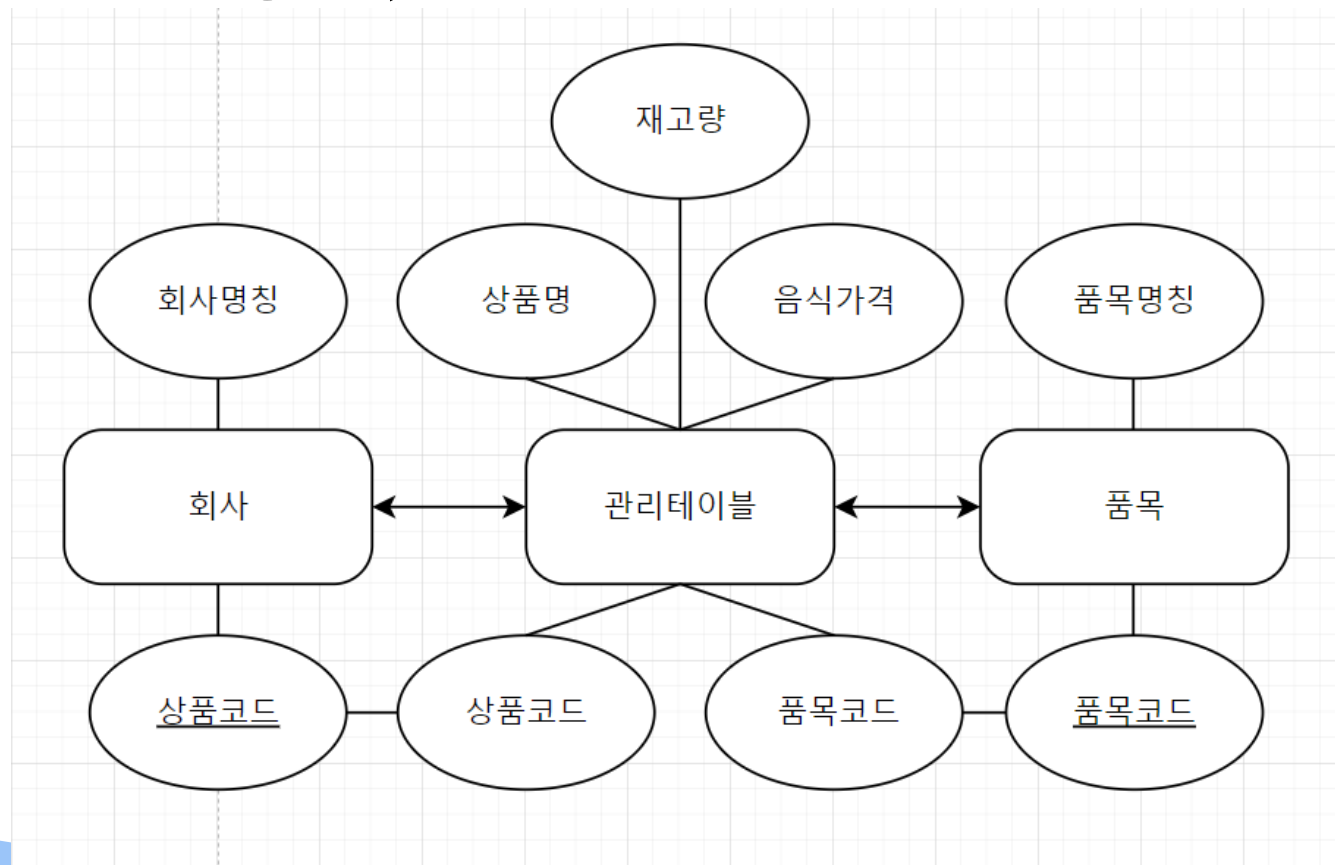
# 프로젝트 수행결과

---



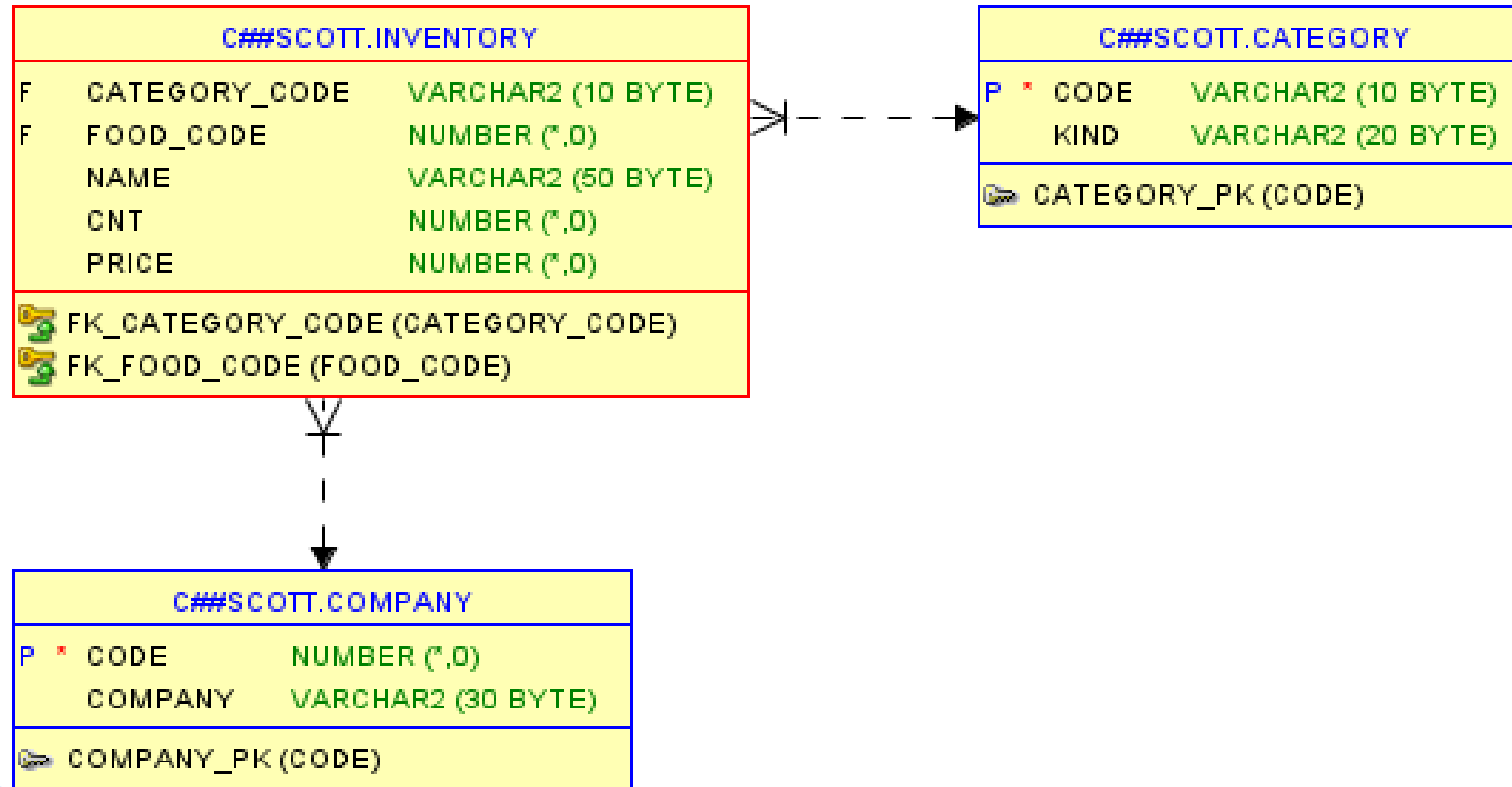
# 프로젝트 수행결과

## ERD(Entity-Relationship Diagram)



# 프로젝트 수행결과

## SQL\_MODEL





# 프로젝트 수행결과

## 테이블 생성

```
CREATE TABLE CATEGORY(  
    CODE VARCHAR2(10) PRIMARY KEY,  
    KIND VARCHAR2(20)  
);  
CREATE TABLE COMPANY(  
    CODE INT DEFAULT FOOD_BARCODE.NEXTVAL PRIMARY KEY,  
    COMPANY VARCHAR2(30)  
);  
CREATE TABLE INVENTORY(  
    CATEGORY_CODE VARCHAR2(10),  
    FOOD_CODE INT DEFAULT INVENTORY_BARCODE.NEXTVAL,  
    NAME VARCHAR2(50),  
    CNT INT ,  
    PRICE INT  
);
```

-- 품목 테이블 (이 테이블의 데이터값은 고정)  
-- 품목분류코드  
-- 품목 분류

-- 회사 테이블  
-- 상품 코드(상품마다 다름, 시퀀스 사용)  
-- 제조회사명

-- 관리 테이블  
-- 품목분류하기위한 코드 (외래키) -- CATEGORY 테이블의 CODE랑 연결  
-- 상품 코드(외래키) -- COMPANY 테이블의 CODE랑 연결  
-- 상품명  
-- 재고량  
-- 음식 가격

# 프로젝트 수행결과

## 테이블 데이터 삽입

```
-- 관리 테이블 데이터 추가/ 상품명(NAME), 품목 코드(FOOD_CODE), 재고(CNT), 가격(PRICE),  
-- 간편식사 데이터 입력  
INSERT INTO INVENTORY(NAME,CATEGORY_CODE,CNT,PRICE) VALUES ('후라이드치킨' , 'AB01' , 3, 9900);  
INSERT INTO INVENTORY(NAME,CATEGORY_CODE,CNT,PRICE) VALUES ('소시지바' , 'AB01' , 2, 2400);  
INSERT INTO INVENTORY(NAME,CATEGORY_CODE,CNT,PRICE) VALUES ('브라우니쿠키' , 'AB01' , 10, 1000);  
INSERT INTO INVENTORY(NAME,CATEGORY_CODE,CNT,PRICE) VALUES ('글레이즈도넛' , 'AB01' , 15, 1300);  
INSERT INTO INVENTORY(NAME,CATEGORY_CODE,CNT,PRICE) VALUES ('뉴자이언트지파이' , 'AB01' , 4, 2900);  
INSERT INTO INVENTORY(NAME,CATEGORY_CODE,CNT,PRICE) VALUES ('크로와상' , 'AB01' , 10, 1300);  
INSERT INTO INVENTORY(NAME,CATEGORY_CODE,CNT,PRICE) VALUES ('곰돌이치킨바' , 'AB01' , 14, 2200);  
INSERT INTO INVENTORY(NAME,CATEGORY_CODE,CNT,PRICE) VALUES ('통살닭다리꼬치' , 'AB01' , 5, 1400);  
INSERT INTO INVENTORY(NAME,CATEGORY_CODE,CNT,PRICE) VALUES ('소보로빵' , 'AB01' , 10, 1500);  
INSERT INTO INVENTORY(NAME,CATEGORY_CODE,CNT,PRICE) VALUES ('핫아메리카노L' , 'AB01' , 20, 1500);  
INSERT INTO INVENTORY(NAME,CATEGORY_CODE,CNT,PRICE) VALUES ('아메리카노L' , 'AB01' , 20, 1500);
```

# 프로젝트 수행결과

## 외래키 생성

```
-- CATEGORY 테이블의 CODE를 참조하여 INVENTORY 테이블의 CATEGORY_CODE를 외래키로 지정
ALTER TABLE INVENTORY ADD CONSTRAINT FK_CATEGORY_CODE
FOREIGN KEY(CATEGORY_CODE) REFERENCES CATEGORY(CODE);
-- COMPANY 테이블의 CODE를 참조하여 INVENTORY 테이블의 FOOD_CODE를 외래키로 지정
ALTER TABLE INVENTORY ADD CONSTRAINT FK_FOOD_CODE
FOREIGN KEY(FOOD_CODE) REFERENCES COMPANY(CODE);
```

# 프로젝트 수행결과

## 뷰 생성 및 조회

-- 모든 테이블 조회 하면 겹치는 속성값(상품코드,카테고리코드)도 나오니까 필요한 정보들만 속성 뽑아서 뷰 만들기.

```
CREATE VIEW ALL_INFO AS
SELECT KIND, NAME, PRICE, CNT, COMPANY, FOOD_CODE 음식코드
FROM COMPANY, CATEGORY, INVENTORY WHERE INVENTORY.FOOD_CODE = COMPANY.CODE
AND CATEGORY.CODE = INVENTORY.CATEGORY_CODE;
```

-- 만들어진 뷰 조회(뷰 이름 : ALL\_INFO)

```
SELECT * FROM ALL_INFO;
```

	KIND	NAME	PRICE	CNT	COMPANY	음식코드
1	간편식사	소시지바	2400	2 PB		846560
2	간편식사	브라우니쿠키	1000	10 PB		846577
3	간편식사	글레이즈도넛	1300	15 PB		846594
4	간편식사	뉴자이언트지파이	2900	4 PB		846611
5	간편식사	크로와상	1300	10 PB		846628
6	간편식사	곰돌이치킨바	2200	14 PB		846645
7	간편식사	통살닭다리꼬치	1400	5 PB		846662
8	간편식사	소보로빵	1500	10 PB		846679
9	간편식사	핫아메리카노L	1500	20 PB		846696
10	간편식사	ICE아메리카노L	1500	20 PB		846713

# 프로젝트 수행결과

## JOIN , GROUP BY, HAVING

-- 모든 테이블 JOIN해서 모두 조회

```
SELECT * FROM INVENTORY JOIN COMPANY  
ON INVENTORY.FOOD_CODE = COMPANY.CODE  
JOIN CATEGORY ON CATEGORY.CODE = INVENTORY.CATEGORY_CODE;  
COMPANY HAVING COUNT(*) >= 3;
```

	CATEGORY_CODE	FOOD_CODE	NAME	CNT	PRICE	CODE	COMPANY	CODE_1	KIND
1	AB01	846560	소시지바	2	2400	846560	PB	AB01	간편식사
2	AB01	846577	브라우니쿠키	10	1000	846577	PB	AB01	간편식사
3	AB01	846594	글레이즈도넛	15	1300	846594	PB	AB01	간편식사
4	AB01	846611	뉴자이언트지파이	4	2900	846611	PB	AB01	간편식사
5	AB01	846628	크로와상	10	1300	846628	PB	AB01	간편식사
6	AB01	846645	곰돌이치킨바	14	2200	846645	PB	AB01	간편식사
7	AB01	846662	통살닭다리꼬치	5	1400	846662	PB	AB01	간편식사
8	AB01	846679	소보로빵	10	1500	846679	PB	AB01	간편식사
9	AB01	846696	핫아메리카노L	20	1500	846696	PB	AB01	간편식사
10	AB01	846713	ICE아메리카노L	20	1500	846713	PB	AB01	간편식사
11	AB01	846730	슈크림빵	15	1400	846730	PB	AB01	간편식사



# 프로젝트 수행결과

## 중첩 질의문

-- 제일 낮은 가격의 상품 정보를 출력하기 (중첩질의문 사용)

```
SELECT * FROM ALL_INFO WHERE PRICE = (SELECT MIN(PRICE) FROM ALL_INFO);
```

	KIND	NAME	PRICE	CNT	COMPANY	음식코드
1	아이스크림	나초코	400	7	서주	847308

# 프로젝트 수행결과

## 프로시저 생성 및 결과

-- 1. 재고 수정

-- 어떤 상품의 재고를 수정 할 것인가? -> 수정할 재고, 상품 이름 쳐서 재고수정

```
CREATE OR REPLACE PROCEDURE UPDATE_CNT
```

```
(  
    UPDATE_CNT INT,  
    SEARCH_NAME VARCHAR2  
)
```

```
IS BEGIN
```

```
UPDATE INVENTORY SET CNT = UPDATE_CNT WHERE NAME = SEARCH_NAME;
```

```
END UPDATE_CNT;
```

```
/
```

-- 2. 가격 수정

-- 어떤 상품 가격을 수정 할 것인가? -> 수정할 가격, 상품 이름 쳐서 가격 수정.

```
CREATE OR REPLACE PROCEDURE UPDATE_PRICE
```

```
(  
    UPDATE_PRICE INT,  
    SEARCH_NAME VARCHAR2  
)
```

```
IS BEGIN
```

```
UPDATE INVENTORY SET PRICE = UPDATE_PRICE WHERE NAME = SEARCH_NAME;
```

```
END UPDATE_PRICE;
```

```
/
```

	CATEGORY_CODE	FOOD_CODE	NAME	CNT	PRICE
1	AB01	846543	후라이드치킨	3	9900

	CATEGORY_CODE	FOOD_CODE	NAME	CNT	PRICE
1	AB01	846543	후라이드치킨	2	9900

	CATEGORY_CODE	FOOD_CODE	NAME	CNT	PRICE
1	AB01	846543	후라이드치킨	2	9000

# 프로젝트 수행결과

## 트리거 생성 및 결과

```
-- 3. 각각 삭제될때 '데이터 삭제되었습니다.'
-- COMPANY 테이블 삭제
CREATE OR REPLACE TRIGGER ALARM_DELETE_COMPANY
BEFORE DELETE ON COMPANY
FOR EACH ROW
DECLARE BEGIN
    DBMS_OUTPUT.PUT_LINE('데이터가 삭제되었습니다. ');
END;
/
-- IVENTORY 테이블 삭제
CREATE OR REPLACE TRIGGER ALARM_DELETE_INVENTORY
BEFORE DELETE ON INVENTORY
FOR EACH ROW
DECLARE BEGIN
    DBMS_OUTPUT.PUT_LINE('데이터가 삭제되었습니다. ');
END;
/
```

데이터가 삭제되었습니다.

PL/SQL 프로시저가 성공적으로 완료되었습니다.

# 자체 평가 의견

---



# 자체 평가 의견

## 정용억

팀원들을 통하여 오라클 SQL의 기본 개념을 이해할 수 있는 좋은 기회가 된것 같다..

## 이현오

DB프로그램인 오라클 프로젝트를 통해 SQL문들 사용하여 DB 구조에 대해서 조금 더 자세하게 알게 되어서 좋은 프로젝트였습니다.

## 장광수

오라클 SQL 프로그램을 이용하여 데이터 구조를 생성하는 것과 데이터 구분이 중요한 일 임을 깨달을 수 있는 프로젝트였다.

## 정인혁

테이블 생성부터 시작해서 데이터 삽입하고, 여러 기능들을 해보면서 데이터 베이스 구조에 대해 좀 더 알게 되어서 좋았습니다!

## 최효정

데이터를 데이터베이스에 저장함으로써 데이터를 더 효율적으로 관리할 수 있다는 점을 다시 한번 상기하는 계기가 되었다.



감 사 합 니 다

---