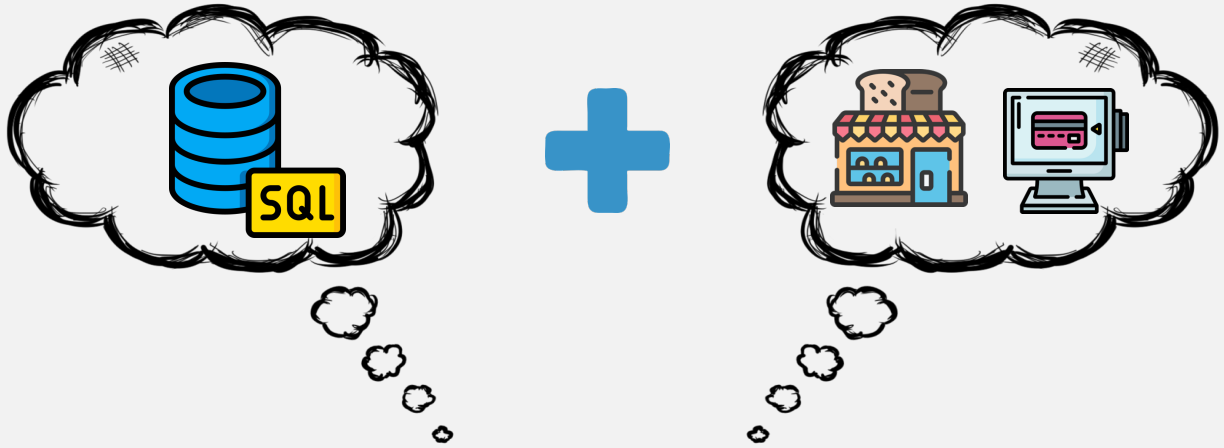
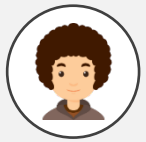


BAKERY POS



이지훈

201804088

김영찬

201600026

한성주

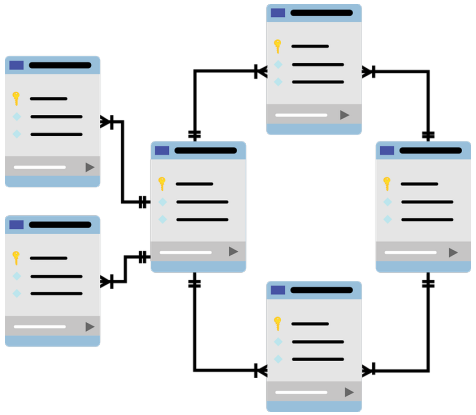
201902126

이예진

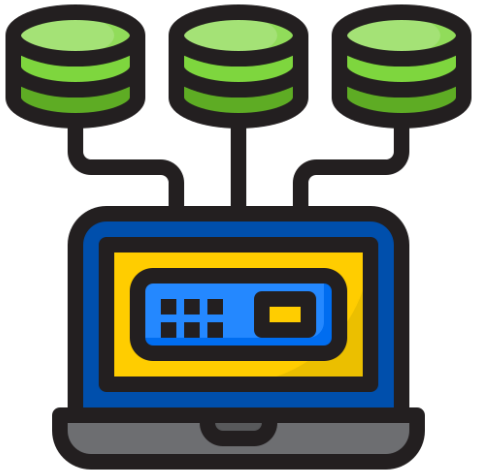
202004228

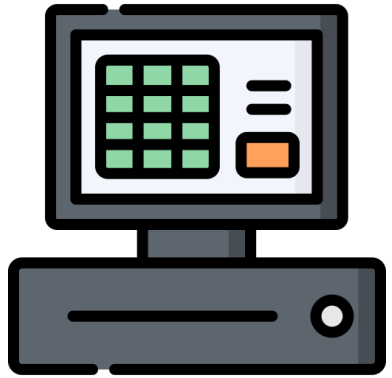



목차



- **DB Modeling**
  - ERD(Entity Relationship Diagram)
  - SQL문 작성
- **Web POS**
  - POS 구현 과정
  - POS 기능
  - 프로그램 실행
- **Q & A**

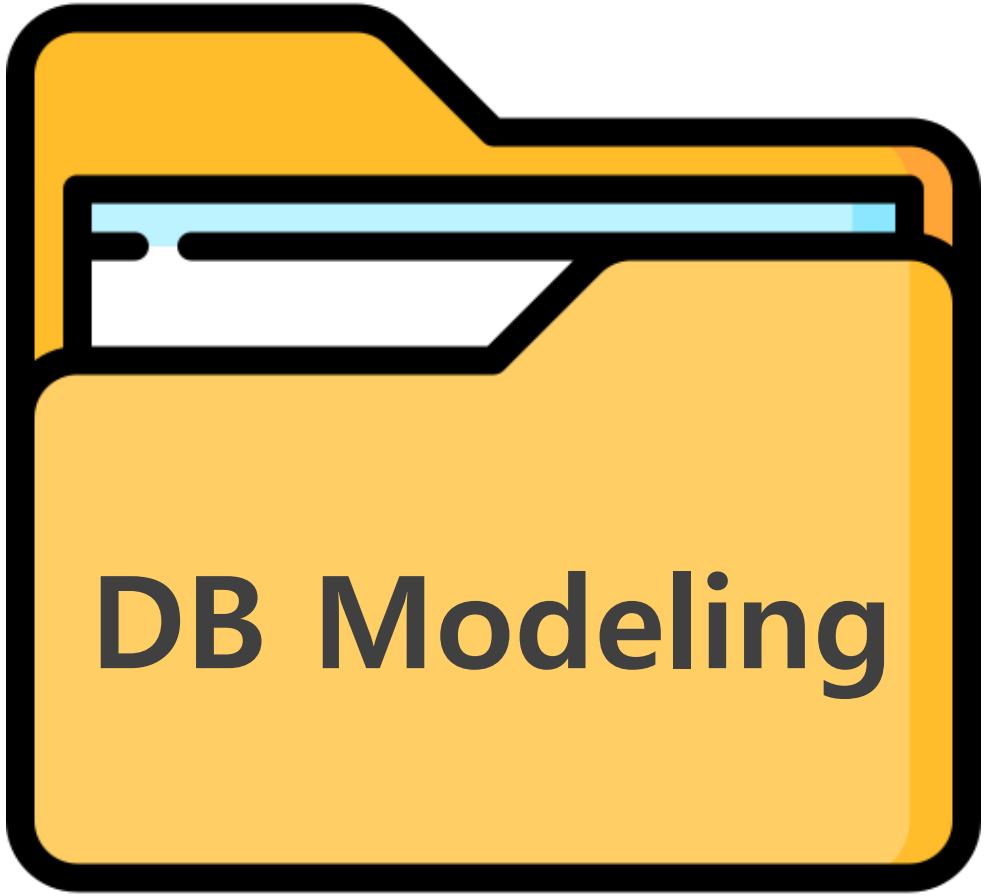


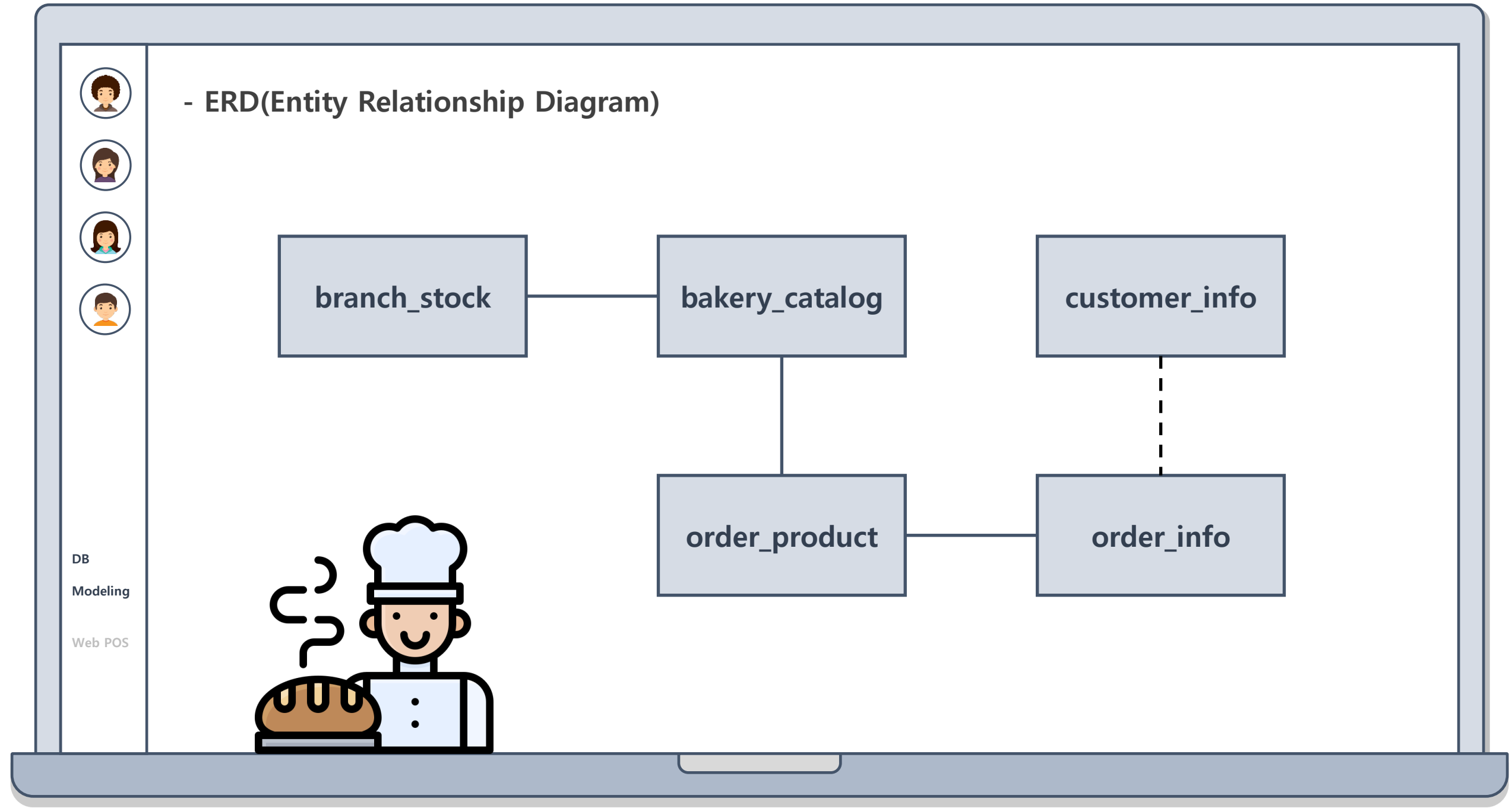


DB Modeling

Web POS




- ERD(Entity Relationship Diagram)
branch_stock
bakery_catalog
customer_info
order_product
order_info
DB
Modeling
Web POS




DB

Modeling

Web POS

- ERD(Entity Relationship Diagram)

bakery_catalog
product_name
product_price

customer_info
customer_id
customer_name

order_info
order_no
branch_name
sale_price
sale_datetime
customer_id
settlement_flag
settlement_datetime

branch_stock
branch_name
product_name
stock_volume

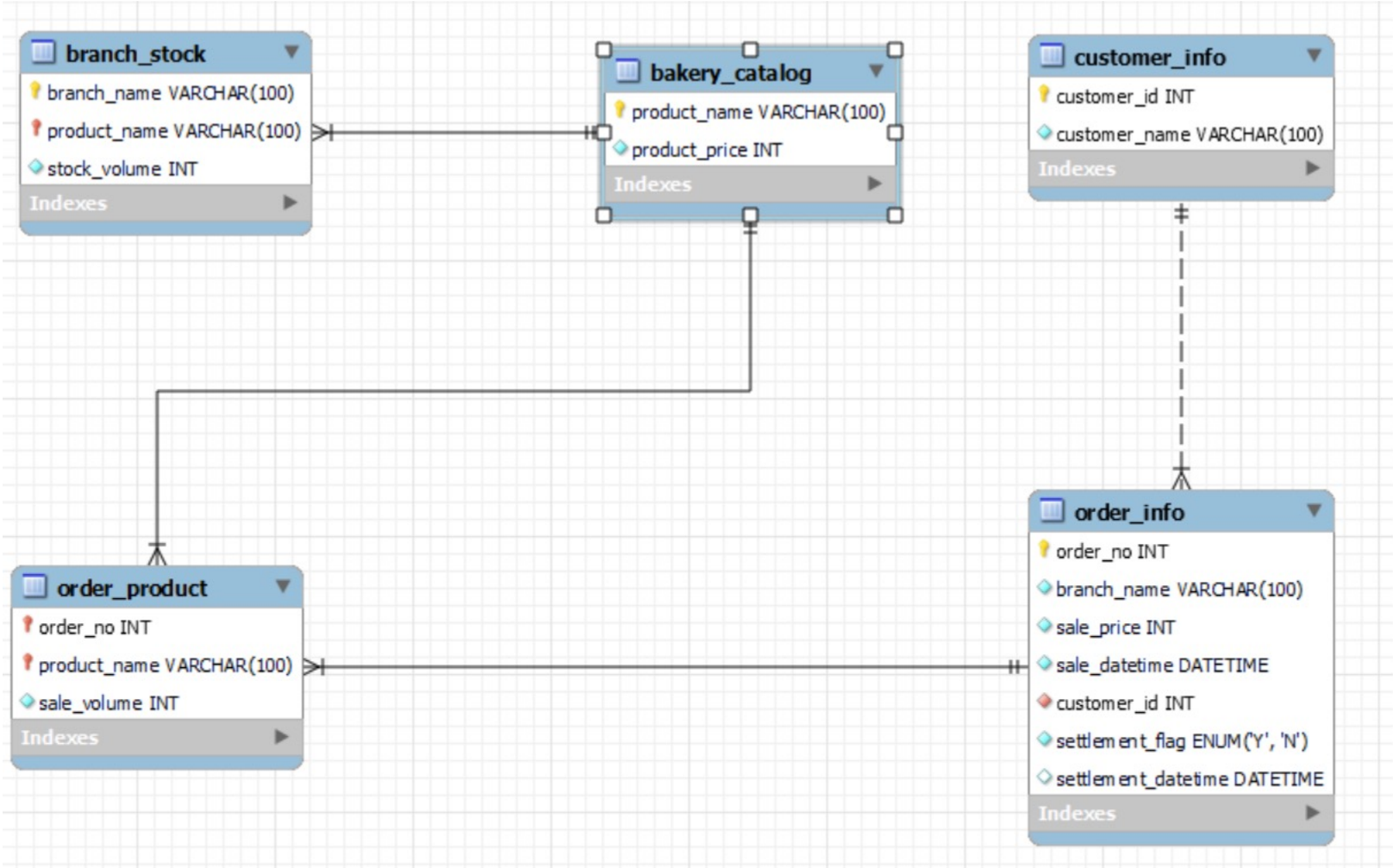
order_product
order_no
product_name
sale_volume

DB

Modeling

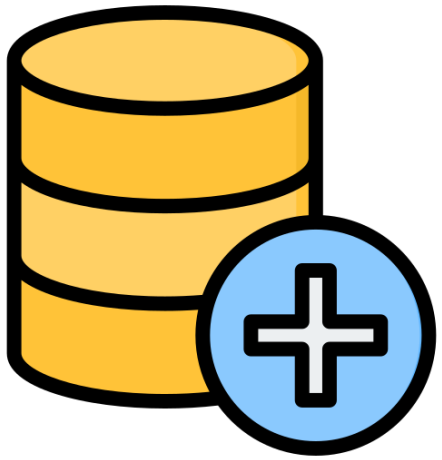
Web POS

- ERD(Entity Relationship Diagram)



- SQL문 작성

- DATABASE 생성



```
1 CREATE DATABASE bakery;
```

DB

Modeling

Web POS




- SQL문 작성

- TABLE 생성

1) 제품 카탈로그 (공급 제품명, 제품 가격)

```
3
4 CREATE TABLE `bakery_catalog` (
5   `product_name` varchar(100) NOT NULL COMMENT '제품명',
6   `product_price` int(11) NOT NULL DEFAULT '0' COMMENT '제품가격',
7   PRIMARY KEY (`product_name`)
8 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='제품 카탈로그';
9
```




DB

Modeling

Web POS

- SQL문 작성

- TABLE 생성

2) 브랜치재고 (브랜치이름, 제품명, 재고수량)

```
12
13 CREATE TABLE `branch_stock` (
14   `branch_name` varchar(100) NOT NULL COMMENT '브랜치이름',
15   `product_name` varchar(100) NOT NULL COMMENT '제품명',
16   `stock_volume` int(11) NOT NULL DEFAULT '0' COMMENT '재고수',
17   PRIMARY KEY (`branch_name`,`product_name`),
18   KEY `fk_product_name` (`product_name`),
19   CONSTRAINT `branch_stock_ibfk_1` FOREIGN KEY (`product_name`)
20     REFERENCES `bakery_catalog` (`product_name`)
21 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='브랜치재고';
22
```




- SQL문 작성

- TABLE 생성

3) 고객 정보 (고객 번호, 고객 이름)

```
22
23 CREATE TABLE `customer_info` (
24   `customer_id` int(11) NOT NULL AUTO_INCREMENT COMMENT '고객번호',
25   `customer_name` varchar(100) NOT NULL COMMENT '고객이름',
26   PRIMARY KEY (`customer_id`)
27 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='고객정보';
28
```




- SQL문 작성

- TABLE 생성

4) 주문 정보 (주문 번호, 브랜치이름, 매출액, 판매일시, 고객번호, 결제여부, 결제일시)

```
30 CREATE TABLE `order_info` (
31   `order_no` int(11) NOT NULL AUTO_INCREMENT COMMENT '주문번호',
32   `branch_name` varchar(100) NOT NULL COMMENT '브랜치이름',
33   `sale_price` int(11) NOT NULL COMMENT '매출액',
34   `sale_datetime` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '판매일시',
35   `customer_id` int(11) NOT NULL COMMENT '고객번호',
36   `settlement_datetime` datetime DEFAULT NULL COMMENT '결제일시',
37   `settlement_flag` varchar(100) NOT NULL COMMENT '결제여부',
38   PRIMARY KEY (`order_no`),
39   KEY `fk_customer_id` (`customer_id`),
40   CONSTRAINT `order_info_ibfk_1` FOREIGN KEY (`customer_id`)
41     REFERENCES `customer_info` (`customer_id`)
42 ) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COMMENT='주문정보';
```




- SQL문 작성

- TABLE 생성

5) 주문 제품 (주문번호, 제품명, 판매량)

```
44 CREATE TABLE `order_product` (
45   `order_no` int(11) NOT NULL AUTO_INCREMENT COMMENT '주문번호',
46   `product_name` varchar(100) NOT NULL COMMENT '제품명',
47   `sale_volume` int(11) NOT NULL COMMENT '판매량',
48   PRIMARY KEY (`order_no`,`product_name`),
49   KEY `fk_product_name` (`product_name`),
50   CONSTRAINT `order_product_ibfk_1` FOREIGN KEY (`product_name`)
51     REFERENCES `bakery_catalog` (`product_name`),
52   CONSTRAINT `order_product_ibfk_2` FOREIGN KEY (`order_no`)
53     REFERENCES `order_info` (`order_no`)
54 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='주문_제품';
```




DB

Modeling

Web POS




- POS 구현 과정

- 제빵 카탈로그 데이터 입력

	product_name	product_price
▶	피자빵	2000
	크림빵	1000
	단팥빵	1000

```
1 INSERT INTO bakery_catalog (product_name, product_price)
2     VALUES ('단팥빵',1000);
3 INSERT INTO bakery_catalog (product_name, product_price)
4     VALUES ('크림빵',1000);
5 INSERT INTO bakery_catalog (product_name, product_price)
6     VALUES ('피자빵',2000);
```




- POS 구현 과정

- 빵 생산재고 입력

	branch_name	product_na...	stock_volume
▶	대전지점	단팥빵	5
	대전지점	크림빵	20
	대전지점	피자빵	30
	서울지점	단팥빵	10
	서울지점	크림빵	10
	서울지점	피자빵	10

```sql
INSERT INTO branch_stock (branch_name, product_name, stock_volume) VALUES ('서울지점', '단팥빵', 10)
    ON DUPLICATE KEY UPDATE stock_volume= stock_volume + VALUES(stock_volume);
INSERT INTO branch_stock (branch_name, product_name, stock_volume) VALUES ('서울지점', '크림빵', 10)
    ON DUPLICATE KEY UPDATE stock_volume= stock_volume + VALUES(stock_volume);
INSERT INTO branch_stock (branch_name, product_name, stock_volume) VALUES ('서울지점', '피자빵', 10)
    ON DUPLICATE KEY UPDATE stock_volume= stock_volume + VALUES(stock_volume);

INSERT INTO branch_stock (branch_name, product_name, stock_volume) VALUES ('대전지점', '단팥빵', 5)
    ON DUPLICATE KEY UPDATE stock_volume= stock_volume + VALUES(stock_volume);
INSERT INTO branch_stock (branch_name, product_name, stock_volume) VALUES ('대전지점', '크림빵', 20)
    ON DUPLICATE KEY UPDATE stock_volume= stock_volume + VALUES(stock_volume);
INSERT INTO branch_stock (branch_name, product_name, stock_volume) VALUES ('대전지점', '피자빵', 30)
    ON DUPLICATE KEY UPDATE stock_volume= stock_volume + VALUES(stock_volume);

SELECT * FROM branch_stock;
```




- POS 구현 과정

- 고객정보 입력

	customer_id	customer_name
▶	1	김손님

```
1 INSERT INTO customer_info (customer_name) VALUES ('김손님');
```




- POS 구현 과정

- 마감분석

1) 12월의 결제 완료된 총매출, 결제 완료된 매출, 미결제매출 구하기

```sql
SELECT SUM(sale_price) as '총매출',
  SUM(CASE settlement_flag WHEN 'Y' THEN sale_price ELSE 0 END) AS '결제완료된 매출',
  SUM(CASE settlement_flag WHEN 'N' THEN sale_price ELSE 0 END) AS '미결제매출'
FROM order_info
WHERE MONTH(sale_datetime)=12;
```




- POS 구현 과정

- 마감분석

2) 12월의 지점별 매출액

```
1 SELECT branch_name as '지점명',
2 SUM(sale_price) as '총매출'
3 FROM order_info
4 WHERE MONTH(sale_datetime)=12
5 GROUP BY branch_name ;
```




- POS 구현 과정

- 마감분석

3) 12월의 제품별 매출액

```sql
SELECT bc.product_name as '제품',
  SUM(op.sale_volume) AS '매출수량',
  MAX(bc.product_price) AS '매출단가(현재기준)',
  SUM(op.sale_volume * bc.product_price) AS  '제품별 매출액(현재가격기준)'
FROM order_info oi  INNER JOIN
order_product op
ON oi.order_no  = op.order_no LEFT OUTER JOIN
bakery_catalog bc
ON op.product_name = bc.product_name
WHERE MONTH(sale_datetime)=12
GROUP BY bc.product_name ;
```



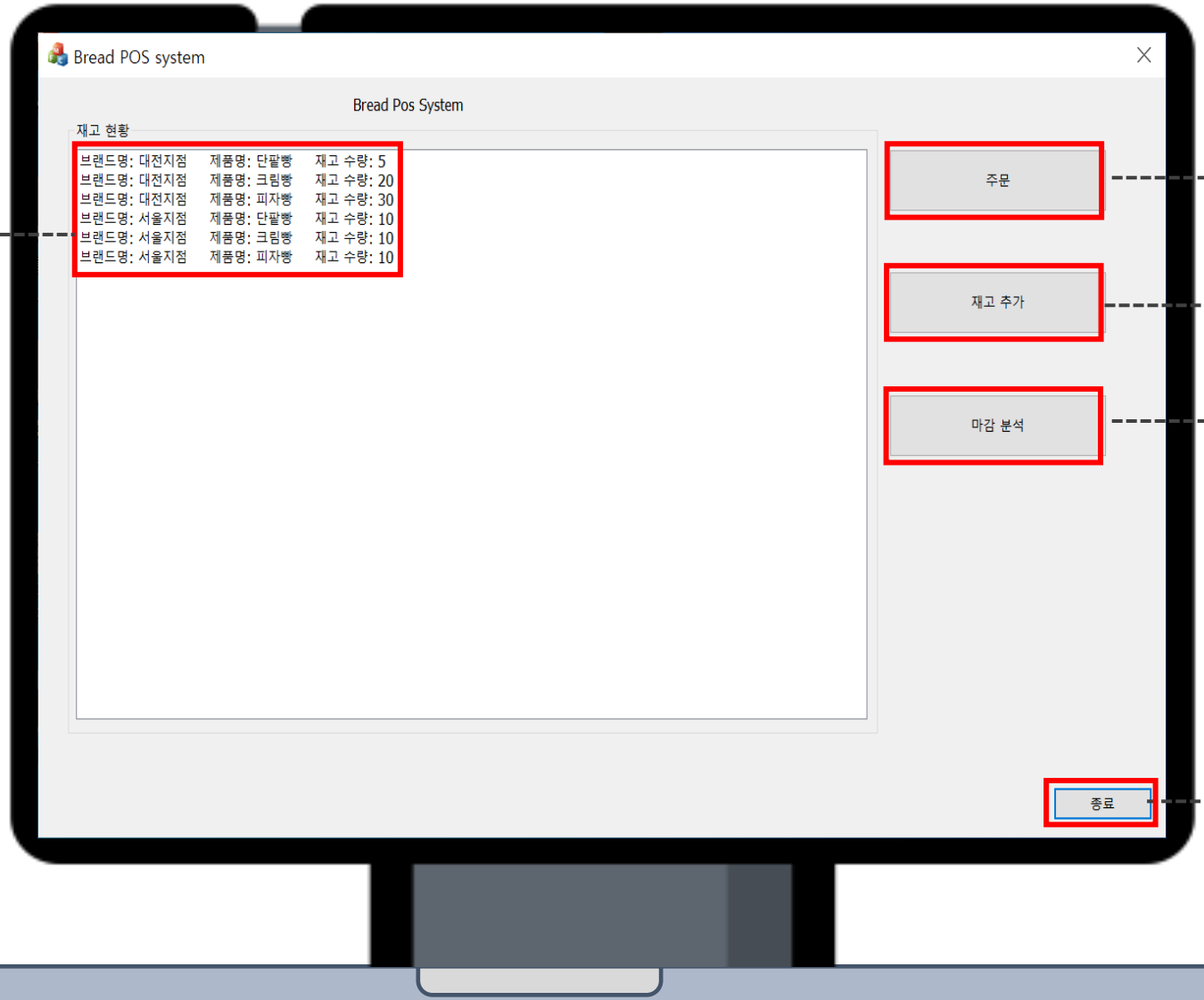

DB

Modeling

Web POS

- POS 기능

- 실행 화면



재고 현황 ⑤

① 주문 버튼

② 재고 추가

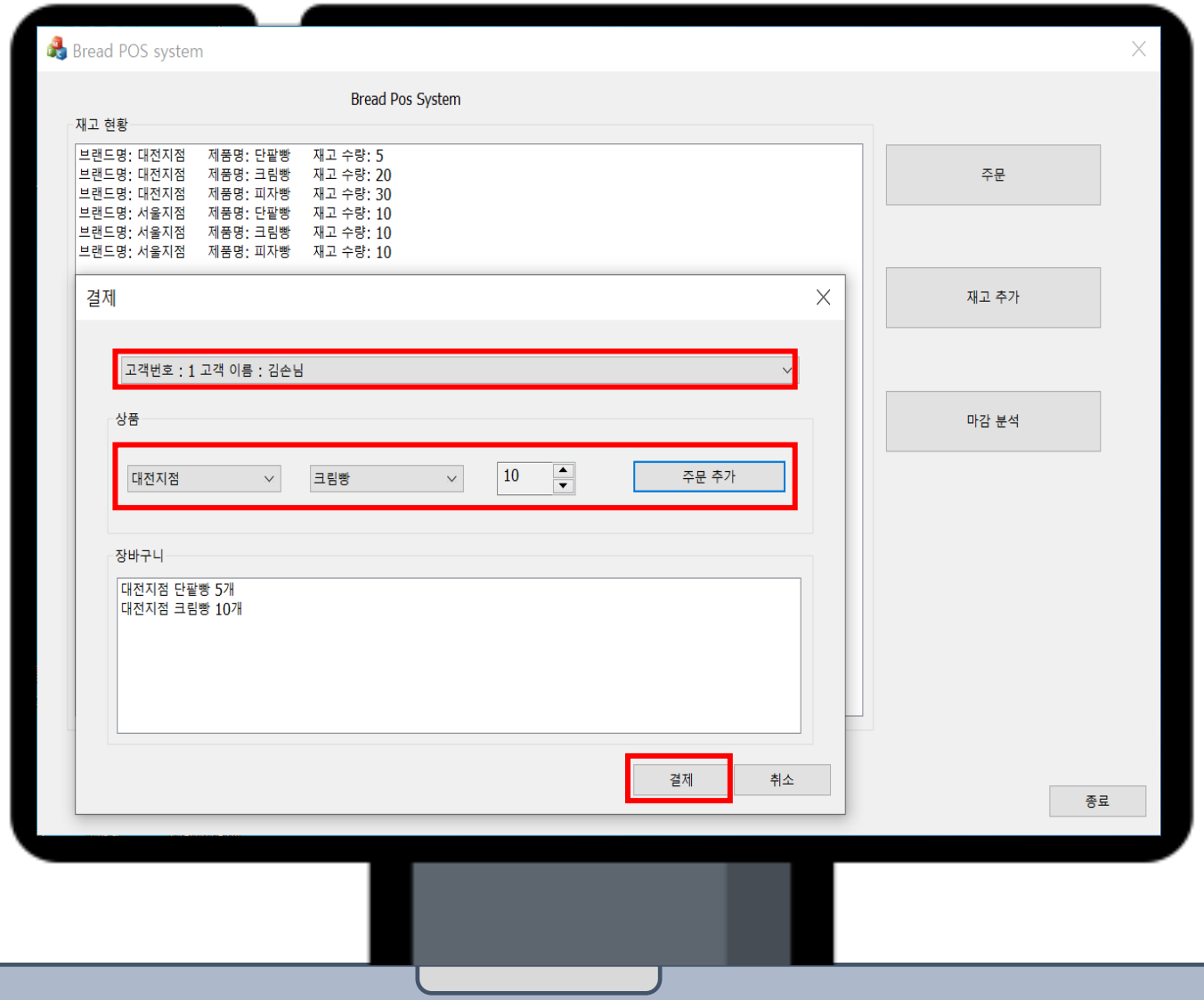
③ 마감 분석

④ 종료 버튼




- POS 기능 ①

- 주문



DB

Modeling

Web POS

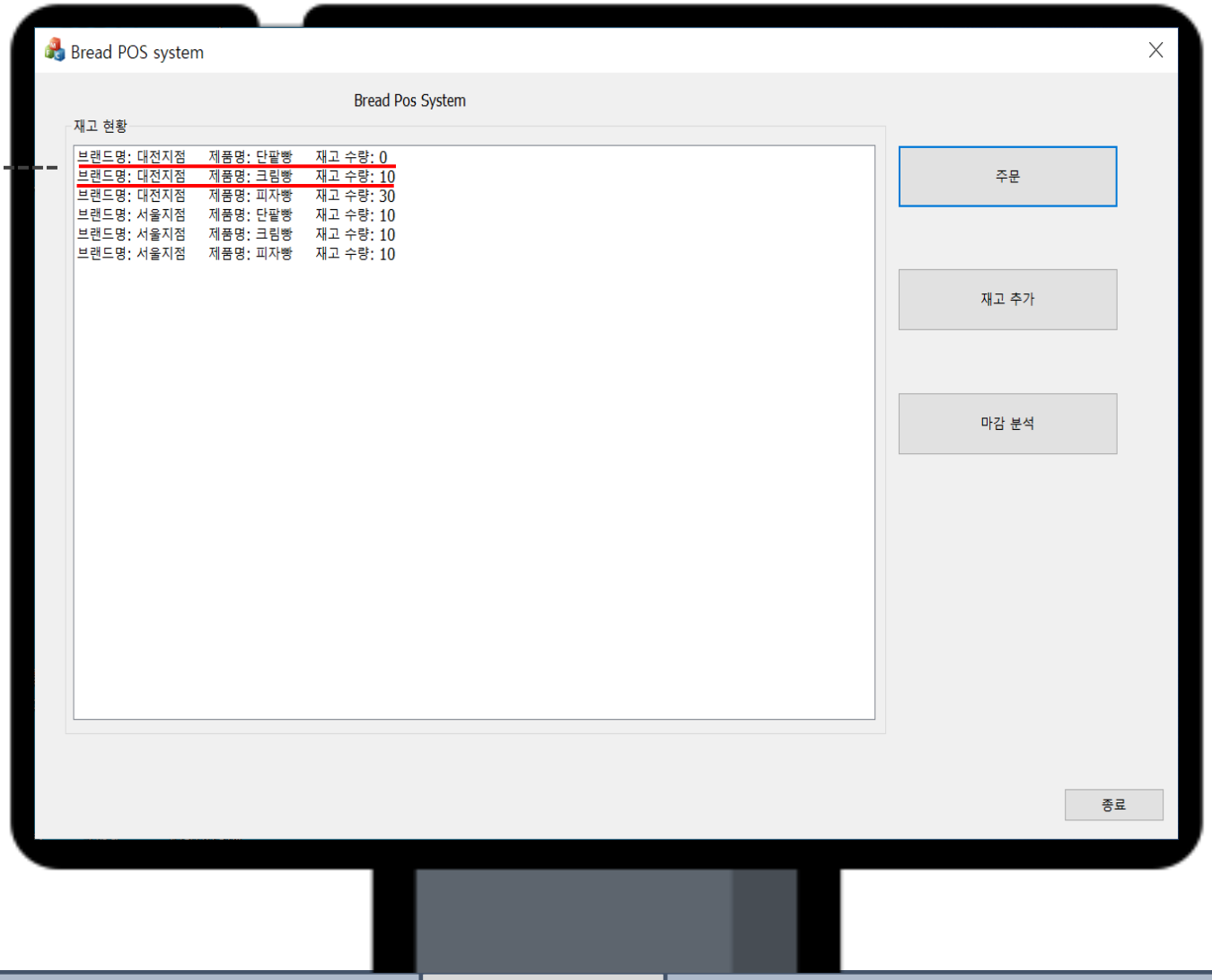



- POS 기능 ⑤

- 주문 후 재고 현황

대전지점
: 단팥빵 -5개

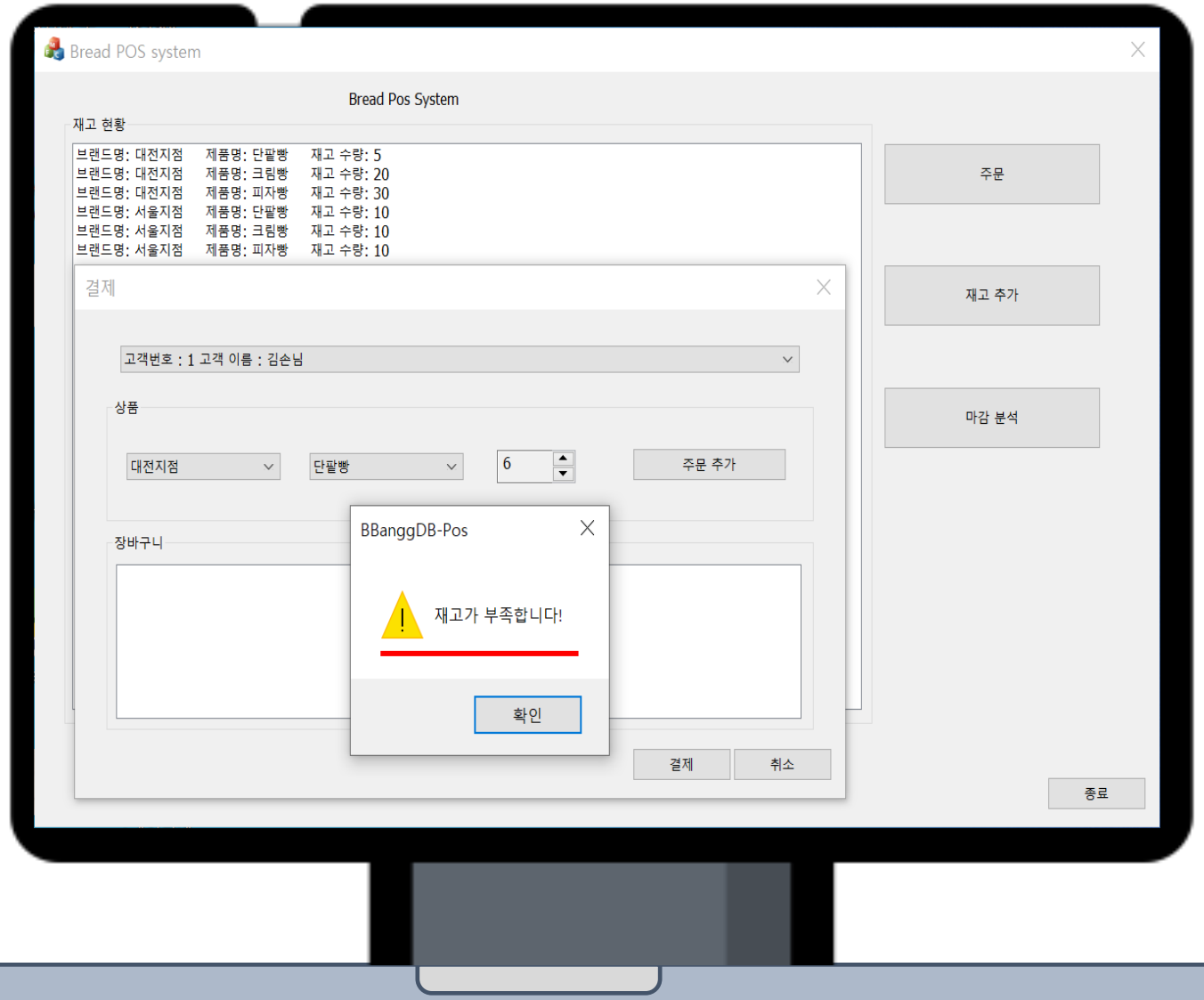
대전지점
: 크림빵 -10개






- POS 기능 ①

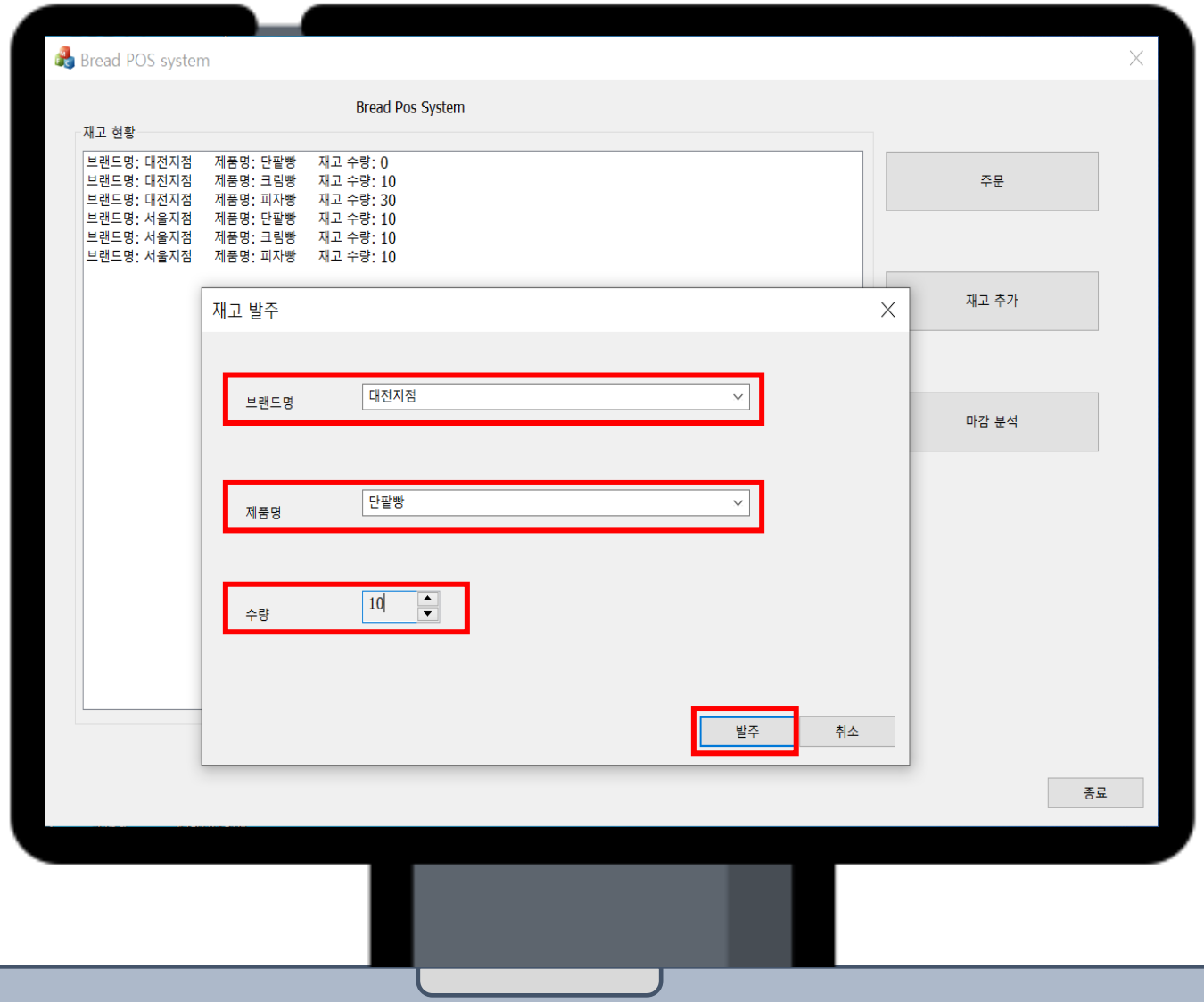
- 주문 결제 시 재고 부족






- POS 기능 ②

- 재고 추가



DB

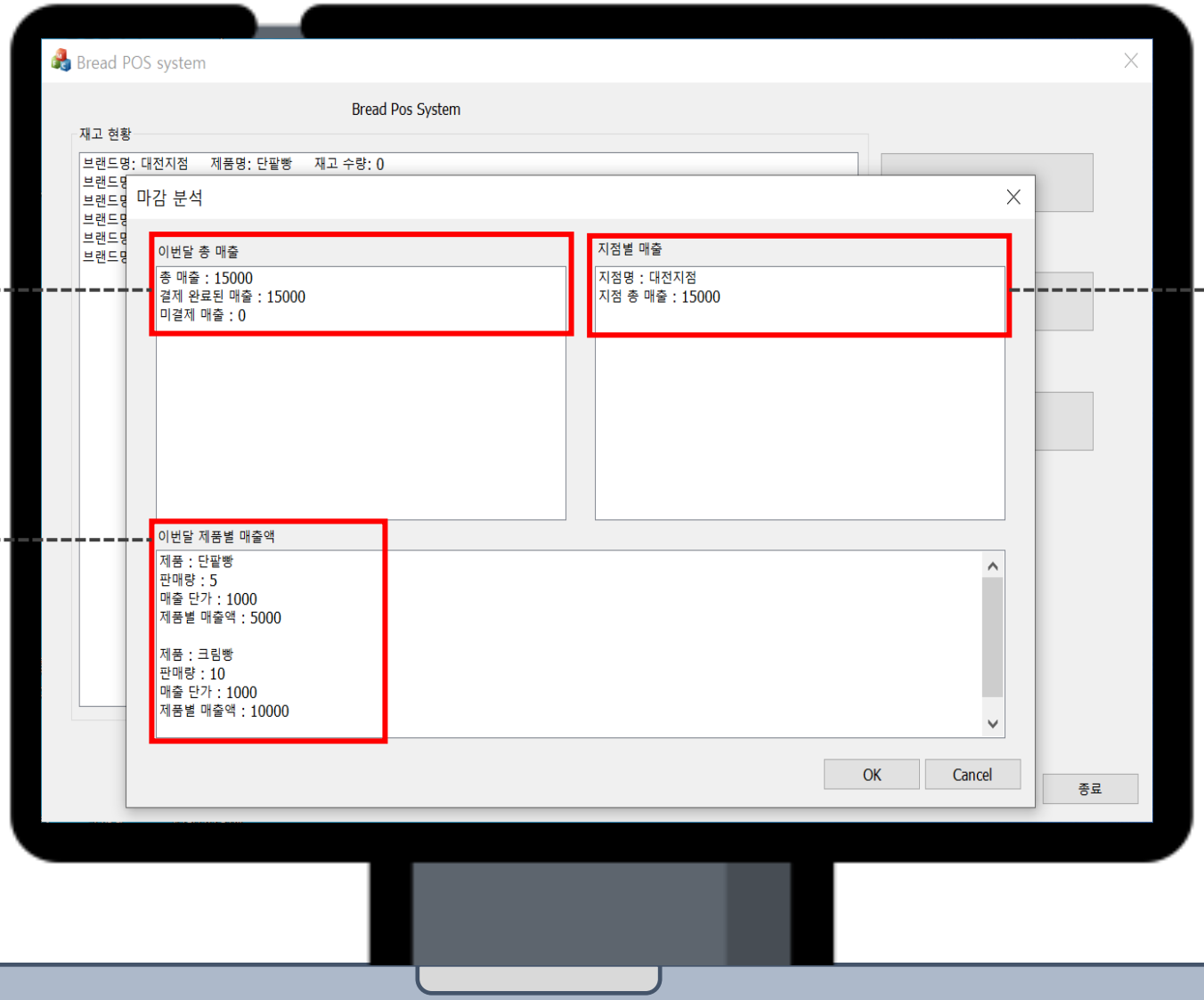
Modeling

Web POS




- POS 기능 ③

- 마감 분석



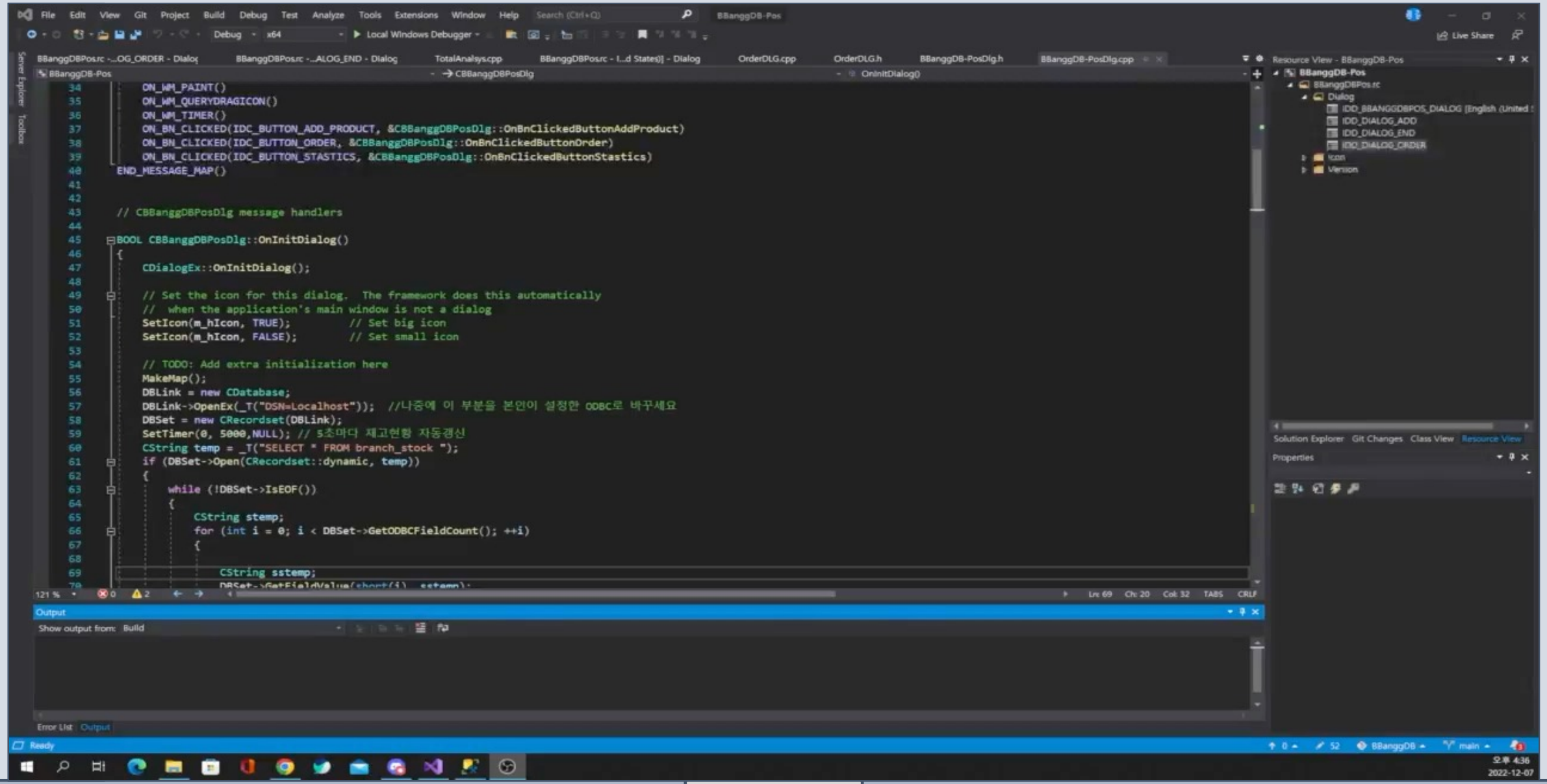
이번달 총 매출

이번달 제품별 매출액

이번달 지점별 매출

DB Modeling

Web POS


ON_WM_PAINT()
ON_WM_QUERYDRAGICON()
ON_WM_TIMER()
ON_BN_CLICKED(IDC_BUTTON_ADD_PRODUCT, &CBBanggDBPosDlg::OnBnClickedButtonAddProduct)
ON_BN_CLICKED(IDC_BUTTON_ORDER, &CBBanggDBPosDlg::OnBnClickedButtonOrder)
ON_BN_CLICKED(IDC_BUTTON_STASTICS, &CBBanggDBPosDlg::OnBnClickedButtonStastics)
END_MESSAGE_MAP()
// CBBanggDBPosDlg message handlers
BOOL CBBanggDBPosDlg::OnInitDialog()
{
CDialogEx::OnInitDialog();
// Set the icon for this dialog. The framework does this automatically
// when the application's main window is not a dialog
SetIcon(m_hIcon, TRUE); // Set big icon
SetIcon(m_hIcon, FALSE); // Set small icon
// TODO: Add extra initialization here
MakeMap();
DBLink = new CDatabase;
DBLink->OpenEx(_T("DSN=Localhost")); //나중에 이 부분을 본인이 설정한 ODBC로 바꾸세요
DBSet = new CRecordset(DBLink);
SetTimer(0, 5000,NULL); // 5초마다 재고현황 자동갱신
CString temp = _T("SELECT * FROM branch_stock ");
if (DBSet->Open(CRecordset::dynamic, temp))
{
while (!DBSet->IsEOF())
{
CString stemp;
for (int i = 0; i < DBSet->GetODBCFieldCount(); ++i)
{
CString sstemp;
Output
Show output from: Build
Error List
Output
Ready


THANK
YOU!