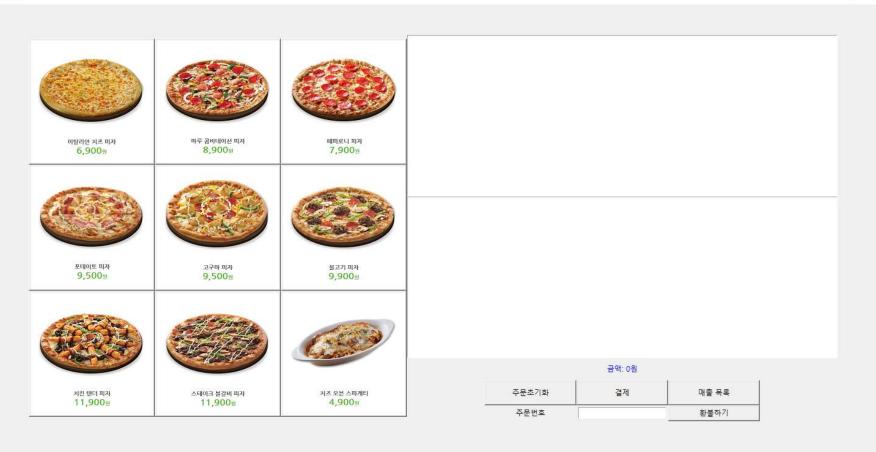
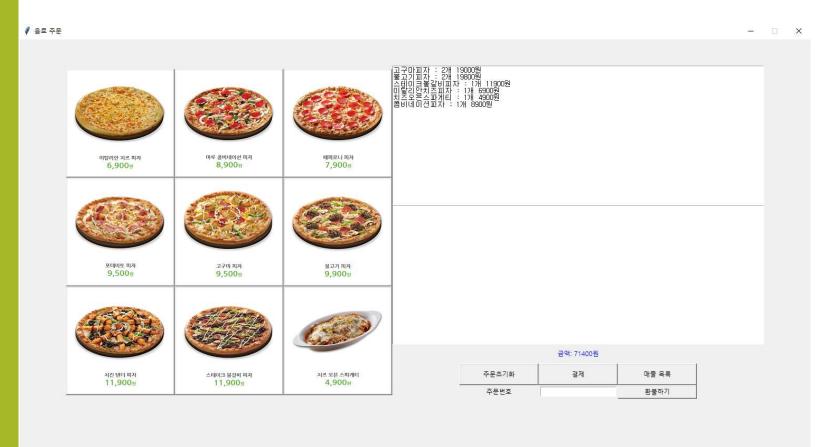
빅데이터 저장시스템 개발

화면 구성

✔ 음료 주문



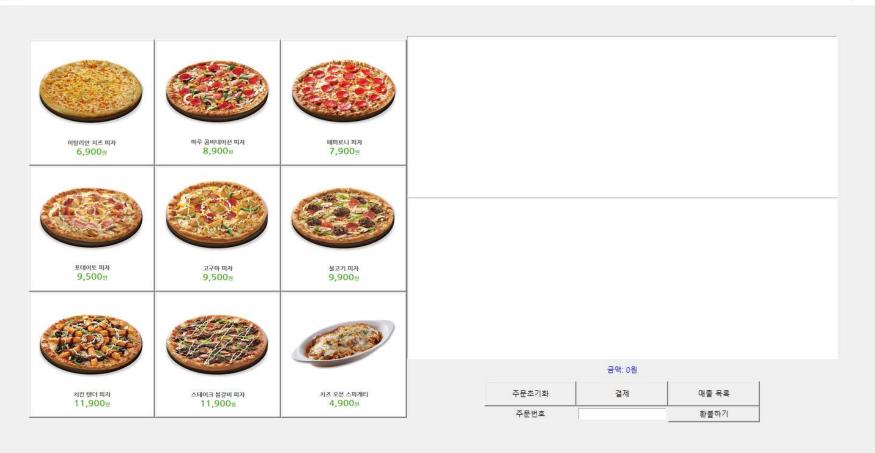
화면_제품클릭시



코드_제품 클릭 시

- 1. 클릭한 제품이 orderList(dict)에 없으면 orderList에 추가
- 2. 클릭한 제품이 orderList(dict)에 있으면 Value값 증가
- 3. 총합계산

화면_결제 버튼 클릭 시



코드 결제 버튼 클릭시

```
def order():
 global sum
                                                          1. 총합을 담는 변수 초기화
buyList.insert_buyList(sum, orderList, price)
 orderList.clear()
                                                          2. 구매 목록 리스트 초기화
 textarea.delete('1.0', END)
 sum = 0
label1['text'] = "금액: 0원"
@staticmethod
def insert_buyList(buyListTotalPrice, orderList, price):
    conn = oci.connect("scott/tiger@192.168.0.18:1521/orcl")
    cursor = conn.cursor()
    cursor.execute(sql1, (buyListTotalPrice,))
    for product in orderList.keys():
        tempPrice = int(price[product])
        cnt = int(orderList[product])
        sal2 = """
        cursor.execute(sql2, (tempPrice, product, cnt))
    cursor.close()
    conn.commit()
    conn.close()
```

- 1. 영수증 단위(구매 시간, 구매 가격 총합 등) insert
- 2. 영수증 안에서의 구매 물품 단위 (구매제품, 구매제품 개수 등) insert

화면_매출 목록 버튼 클릭 시

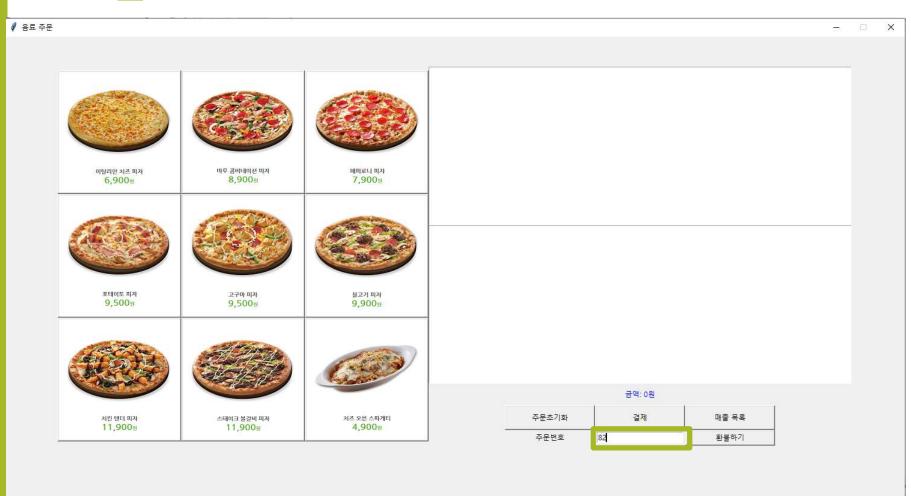


코드_매출 목록 버튼 클릭 시

```
def btn_orderList():
  textarea2.delete('1.0', END)
  textarea2.insert(tk.INSERT, buyList.select_buyList())
def select_buyList():
   returnString = ''
   cursor = conn.cursor()
   cursor.execute(sql1)
   datas = cursor.fetchall()
   for row in datas:
      returnString = returnString + "-------
      returnString = returnString + "주문번호 : " + str(row[0]) + "\n"
       returnString = returnString + "주문날짜 : " + str(row[1]) + "\n"
       returnString = returnString + "-----
      cursor.execute(sql2, (row[0], ))
      datas2 = cursor.fetchall()
      for row2 in datas2:
          returnString = returnString + str(row2[3]) + "\t\t" + str(row2[4]) + "\t\t" + <math>str(row2[2]) + "\n"
      returnString = returnString + "-----
       returnString = returnString + "가격총합 : " + str(row[2]) + "\n"
       returnString = returnString + "------
       returnString = returnString + "\n\n"
   cursor.close()
   return returnString
```

- 1. 영수증 단위(구매 시간, 구매 가격 총합 등) select
- 2. 영수증 안에서의 구매 물품 단위 (구매제품, 구매제품 개수 등) select
- 3. 앞의 값들을 원하는 형태의 string으로 만들어 반환

화면_환불 버튼 클릭 시



화면_환불 버튼 클릭 시

	引 BUYLIST	⊕ BUYLIST_DATE	⊕ BUYLIST_TOTALPRICE
1	84	20/06/24	26700
2	82	20/06/24	71400
3	81	20/06/24	110200
4	80	20/06/24	52500
5	79	20/06/24	83900
6	78	20/06/24	36200
7	77	20/06/24	14800
8	73	20/06/24	8900
9	72	20/06/24	6900
10	71	20/06/24	23700
11	70	20/06/24	41100
12	68	20/06/24	78000
13	67	20/06/24	81300

	№1 BUYLIST	⊕ BUYLIST_DATE	⊕ BUYLIST_TOTALPRICE
1	84	20/06/24	26700
2	81	20/06/24	110200
3	80	20/06/24	52500
4	79	20/06/24	83900
5	78	20/06/24	36200
6	77	20/06/24	14800
7	73	20/06/24	8900
8	72	20/06/24	6900
9	71	20/06/24	23700
10	70	20/06/24	41100
11	68	20/06/24	78000
12	67	20/06/24	81300

화면_환불 버튼 클릭 시

	⊕ BUY_ID	₩1 BUYLIS	♦	BUY_PRICE	⊕ BUY_PRODUCTNAME	⊕ BUY_CNT
1	142	8	34	9900	불고기피자	1
2	143	8	34	8900	콤비네이션피자	1
3	144	8	34	7900	페페로니피자	1
4	128	8	32	9900	불고기피자	2
5	127	8	32	9500	고구마피자	2
6	129	8	32	11900	스테이크불갈비피자	1
7	132	8	32	8900	콤비네이션피자	1
8	131	8	32	4900	치즈오븐스파게티	1
9	130	8	32	6900	이탈리안치즈피자	1
10	123	8	31	11900	치킨텐더피자	1
11	119	8	31	9900	불고기피자	2
12	120	8	31	11900	스테이크불갈비피자	1
13	122	8	31	4900	치즈오븐스파게티	1
14	118	8	31	9500	고구마피자	3
15	125	8	31	7900	페페로니피자	1
16	124	8	31	8900	콤비네이션피자	1
17	126	8	1	9500	포테이토피자	1
18	121	8	31	6900	이탈리안치즈피자	1
19	114	8	30	9500	고구마피자	2
20	117	8	30	8900	콤비네이션피자	1
21	115	8	30	9900	불고기피자	1
22	116	8	30	4900	치즈오븐스파게티	3

	⊕ BUY_ID 📲	BUYLIS	BUY_PRICE	⊕ BUY_PRODUCTNAME	⊕ BUY_CNT
1	144	84	7900	페페로니피자	1
2	143	84	8900	콤비네이션피자	1
3	142	84	9900	불고기피자	1
4	122	81	4900	치즈오븐스파게티	1
5	121	81	6900	이탈리안치즈피자	1
6	126	81	9500	포테이토피자	1
7	119	81	9900	불고기피자	2
8	125	81	7900	페페로니피자	1
9	118	81	9500	고구마피자	3
10	124	81	8900	콤비네이션피자	1
- 11	123	81	11900	치킨텐더피자	1
12	120	81	11900	스테이크불갈비피자	1
13	115	80	9900	불고기피자	1
14	117	80	8900	콤비네이션피자	1
15	116	80	4900	치즈오븐스파게티	3
16	114	80	9500	고구마피자	2

코드_환불 버튼 클릭 시

```
def btn_refund():
 refundId = orderIdEntry.get()
 buyList.delete_buyList(refundId)
 orderIdEntry.delete(0, END)
@staticmethod
def delete_buyList(orderListId):
    conn = oci.connect("scott/tiger@192.168.0.18:1521/orcl")
    cursor = conn.cursor()
    sql1 = """
                DELETE FROM buy where buylist_id = :orderListId
    cursor.execute(sql1, (orderListId,))
    sql2 = """
                DELETE FROM buylist where buylist_id = :orderListId
    cursor.execute(sql2, (orderListId,))
    cursor.close()
    conn.commit()
    conn.close()
```

영수증 안에서의 구매 물품 단위
 (구매제품, 구매제품 개수 등) delete
 영수증 단위(구매 시간, 구매 가격 총합 등) delete