# 서대문구 호프집 상권 및 주민 분석

유혜지, 안해인, 유현선, 이재용

## 1. 서대문구 호프집 상권 분석

서울시 내 25개 구에 존재하는 모든 호프집 및 간이 주점을 대상으로 각 구별 전체 매장 수, 개폐업률, 생존률, 평균 영업 기간 데이터를 수집하였다. 모든 데이터는 2018년 4분기 기준이며, 서울시 골목상권 분석 서비스(https://golmok.seoul.go.kr/regionAreaAnalysis.do)와 통계청(http://kostat.go.kr/portal/korea/index.action/ https://sgis.kostat.go.kr/view/bizStats/bizStatsMap?biz=0 )에서 수집하였다.

수집한 통계 데이터에 대하여 전체 구의 평균과 각 25개 구의 데이터 수치를 비교해 서대문구의 호프집 및 간이 주점 상권은 다른 구에 비해 얼마나 활성화되어 있는지 분석해보았다.

### 1) 점포 수

- 각 행정구역별로 호프집 점포 수를 서울시 전체 평균과 비교해보았다.

```
SELECT 행정구역, 전체 FROM HOF_NUM
WHERE 전체 < (SELECT AVG(전체) FROM HOF_NUM)
ORDER BY 전체;
```

결과 메시지

| | 행정구역 | 전체 |
|---|---|---|
| 1 | 성동구 | 385 |
| 2 | 도봉구 | 406 |
| 3 | 노원구 | 491 |
| 4 | 금천구 | 499 |
| 5 | 용산구 | 519 |
| 6 | 동작구 | 523 |
| 7 | 중구 | 543 |
| 8 | 양천구 | 579 |
| 9 | 서대문구 | 610 |
| 10 | 은평구 | 624 |
| 11 | 성북구 | 639 |
| 12 | 중랑구 | 651 |
| 13 | 서초구 | 665 |
| 14 | 동대문구 | 665 |
| 15 | 강북구 | 678 |

- 그 결과, 서대문구는 서울시 전체 평균 이하 9위였다. 즉, 타지역에 비해 서대문구에는 호프집의 수가 적었다.

## 2) 호프집 비율 및 평균 종사자 수

- 전체 외식업 사업체 중에서 호프집이 차지하는 비율과 평균적으로 호프집에 종사하는 사람의 수를 행정구역별로 서울시 전체 평균과 비교해보았다.

- 호프집 비율

```
SELECT 행정구역, 호프집_비율 FROM HOF_WORKER
WHERE 호프집_비율 < (SELECT AVG(호프집_비율) FROM HOF_WORKER)
ORDER BY 호프집_비율;
```

결과 메시지

| | 행정구역 | 호프집_비율 |
|---|---|---|
| 1 | 도봉구 | 2.1 |
| 2 | 성동구 | 2.3 |
| 3 | 금천구 | 2.7 |
| 4 | 노원구 | 2.7 |
| 5 | 중구 | 2.9 |
| 6 | 동작구 | 3.2 |
| 7 | 용산구 | 3.2 |
| 8 | 양천구 | 3.3 |
| 9 | 서초구 | 3.4 |
| 10 | 서대문구 | 3.5 |
| 11 | 성북구 | 3.8 |
| 12 | 은평구 | 3.9 |
| 13 | 동대문구 | 3.9 |

- 호프집 평균 종사자 수

```
SELECT 행정구역, 평균_종사자수 FROM HOF_WORKER
WHERE 평균_종사자수 < (SELECT AVG(평균_종사자수) FROM HOF_WORKER)
ORDER BY 평균_종사자수;
```

결과 | 메시지

| | 행정구역 | 평균_종사자수 |
|---|---|---|
| 1 | 금천구 | 179 |
| 2 | 은평구 | 180 |
| 3 | 도봉구 | 181 |
| 4 | 중랑구 | 181 |
| 5 | 양천구 | 200 |
| 6 | 구로구 | 201 |
| 7 | 영등포구 | 202 |
| 8 | 동대문구 | 203 |
| 9 | 강북구 | 210 |
| 10 | 성북구 | 212 |
| 11 | 강서구 | 216 |
| 12 | 강동구 | 218 |
| 13 | 노원구 | 219 |
| 14 | 동작구 | 220 |
| 15 | 관악구 | 220 |
| 16 | 서대문구 | 223 |

- 비교 결과, 서대문구는 호프집 비율과 평균 종사자 수 모두 서울시 평균 이하였다.

### 3) 연차별 생존률

- 서울시 내의 호프집 중 개업 후 각각 3년, 5년간 생존한 호프집의 비율을 각 행정구역별로 서울시 전체 평균과 함께 비교해보았다.

- 3년 생존률

```
SELECT 행정구역, YEAR_3 FROM SURVIVAL_RATE
WHERE YEAR_3 > (SELECT AVG(YEAR_3) FROM SURVIVAL_RATE)
ORDER BY YEAR_3 DESC;
```

결과 메시지

| | 행정구역 | YEAR_3 |
|---|---|---|
| 1 | 중구 | 50.7 |
| 2 | 서대문구 | 49.6 |
| 3 | 도봉구 | 47 |
| 4 | 종로구 | 46.3 |
| 5 | 성북구 | 45.7 |
| 6 | 동작구 | 45.6 |
| 7 | 마포구 | 45.4 |
| 8 | 노원구 | 45 |
| 9 | 광진구 | 44.6 |
| 10 | 강서구 | 43.7 |
| 11 | 금천구 | 43.3 |
| 12 | 서초구 | 43.3 |

- 5년 생존률

```
SELECT 행정구역, YEAR_5 FROM SURVIVAL_RATE
WHERE YEAR_5 > (SELECT AVG(YEAR_5) FROM SURVIVAL_RATE)
ORDER BY YEAR_5 DESC;
```

결과 메시지

| | 행정구역 | YEAR_5 |
|---|---|---|
| 1 | 중구 | 40.8 |
| 2 | 종로구 | 34.4 |
| 3 | 용산구 | 32.6 |
| 4 | 동대문구 | 28.2 |
| 5 | 성동구 | 25 |
| 6 | 영등포구 | 25 |
| 7 | 성북구 | 23.9 |
| 8 | 노원구 | 23.5 |
| 9 | 동작구 | 23.3 |
| 10 | 서대문구 | 22.6 |

- 그 결과, 서대문구는 다른 구에 비해 호프집의 생존률이 매우 높았다. 특히 3년 생존률은 전체 25개 구 중 2번째로 높았으며, 5년 생존률 역시 상위 10위권 안에 들었다.

### 4) 개폐업률

- 서울시 내 호프집의 각 구별 개폐업률을 서울시 전체 평균과 비교해보았다.

- 폐업률

```
SELECT 행정구역, 폐업률 FROM OPEN_CLOSE_RATE
WHERE 폐업률 < (SELECT AVG(폐업률) FROM OPEN_CLOSE_RATE)
ORDER BY 폐업률;
```

결과 메시지

| | 행정구역 | 폐업률 |
|---|---|---|
| 1 | 송파구 | 2.8 |
| 2 | 용산구 | 3.1 |
| 3 | 종로구 | 3.4 |
| 4 | 서초구 | 3.5 |
| 5 | 은평구 | 3.8 |
| 6 | 강서구 | 4 |
| 7 | 중구 | 4.1 |
| 8 | 서대문구 | 4.3 |
| 9 | 노원구 | 4.3 |
| 10 | 동작구 | 4.4 |
| 11 | 성북구 | 4.5 |
| 12 | 금천구 | 4.6 |
| 13 | 성동구 | 4.7 |
| 14 | 구로구 | 4.8 |

- 개업률

```
SELECT 행정구역, 개업률 FROM OPEN_CLOSE_RATE
WHERE 개업률 < (SELECT AVG(개업률) FROM OPEN_CLOSE_RATE)
ORDER BY 개업률;
```

결과 메시지

| | 행정구역 | 개업률 |
|---|---|---|
| 1 | 노원구 | 0.6 |
| 2 | 마포구 | 1.1 |
| 3 | 용산구 | 1.2 |
| 4 | 중구 | 1.3 |
| 5 | 동작구 | 1.3 |
| 6 | 서대문구 | 1.3 |
| 7 | 서초구 | 1.4 |
| 8 | 성북구 | 1.6 |
| 9 | 송파구 | 1.9 |
| 10 | 종로구 | 2 |
| 11 | 광진구 | 2 |

- 비교 결과, 서대문구는 다른 구에 비해 호프집의 개업률이 낮은 동시에 폐업률도 낮았다.

## 5) 호프집 평균 영업 기간

- 서울시 내 구별 호프집의 평균 영업 기간을 서울시 전체 평균 영업 기간과 비교해보았다.

- 최근 10년간 평균 영업 기간

```
SELECT 행정구역, RECENT_10 FROM AVG_SALES_PERIOD
WHERE RECENT_10 > (SELECT AVG(RECENT_10) FROM AVG_SALES_PERIOD)
ORDER BY RECENT_10;
```

결과 메시지

| | 행정구역 | RECENT_10 |
|---|---|---|
| 1 | 광진구 | 2.8 |
| 2 | 동대문구 | 2.8 |
| 3 | 양천구 | 2.8 |
| 4 | 영등포구 | 2.8 |
| 5 | 동작구 | 2.9 |
| 6 | 서대문구 | 2.9 |
| 7 | 마포구 | 2.9 |
| 8 | 성북구 | 2.9 |
| 9 | 성동구 | 2.9 |
| 10 | 용산구 | 3 |
| 11 | 노원구 | 3 |
| 12 | 서초구 | 3 |
| 13 | 종로구 | 3.1 |
| 14 | 중구 | 3.2 |

- 최근 30년간 평균 영업 기간

```
SELECT 행정구역, RECENT_30 FROM AVG_SALES_PERIOD
WHERE RECENT_30 > (SELECT AVG(RECENT_30) FROM AVG_SALES_PERIOD)
ORDER BY RECENT_30;
```

결과 메시지

| | 행정구역 | RECENT_30 |
|---|---|---|
| 1 | 동대문구 | 5 |
| 2 | 성북구 | 5 |
| 3 | 송파구 | 5 |
| 4 | 동작구 | 5.1 |
| 5 | 용산구 | 5.2 |
| 6 | 성동구 | 5.2 |
| 7 | 서대문구 | 5.3 |
| 8 | 서초구 | 5.5 |
| 9 | 종로구 | 5.6 |
| 10 | 중구 | 6 |

- 비교 결과, 서대문구 내 호프집들의 평균 영업 기간은 다른 구에 비해 긴 편에 속했다.

### 6) 종합 결과

위의 모든 분석 결과를 종합해본 결과, 서대문구는 호프집의 비율이 타 지역에 비해 낮은 것에 비해 생존률이 매우 높다는 것을 알 수 있다. 이는 곧 서대문구 내의 호프집들은 서로 경쟁이 치열하지 않다는 의미이므로, 서대문구에서 호프집을 창업한다면 오랜 기간 동안 안정적인 매출을 기록할 수 있을 것이다.

## 2. 서대문구 신촌동 주민 분석

우리는 서대문구 내에서도 호프집이 적은 이대역 근처(신촌동)을 창업 장소로 정했다. 호프집을 안정적으로 운영하기 위해서는 해당 상권의 거주 인구 및 유동 인구의 특성을 분석하여 타겟으로 삼을 주고객층을 선정해야 한다. 그래서 우리는 신촌동의 거주 인구 및 유동인구의 성별 및 연령대를 분석해보았다.

거주 인구 및 유동 인구에 대한 데이터는 서울시 골목상권 분석 서비스 (https://golmok.seoul.go.kr/fixedAreaAnalysis.do)의 서대문구 신촌동의 호프집 및 간이 주점의 분석 보고서에서 수집하였다.

### 1) 거주인구와 유동인구의 성별 비율

```
SELECT F1, 거주인구, 유동인구 FROM Sheet1$
WHERE F1 = '여자' OR F1 = '남자'
```

| F1 | 거주인구 | 유동인구 |
|---|---|---|
| 여자 | 1205 | 935924 |
| 남자 | 548 | 427950 |

- 거주인구와 유동인구의 성별을 조사해본 결과, 여성이 남성에 비해 2배 이상 많았다.

### 2) 거주 인구, 유동 인구 연령대

```
SELECT MAX(거주인구) AS 거주인구, MAX(유동인구) AS 유동인구 FROM Sheet1$
WHERE F1 != '여자' AND F1 != '남자'
```

| | 거주인구 | 유동인구 |
|---|---|---|
| 1 | 770 | 768995 |

```
SELECT F1 FROM Sheet1$
WHERE 거주인구 = 770 OR 유동인구 = 768995
```

| | F1 |
|---|---|
| 1 | 20대 |

- 전체 거주인구와 유동인구 중 가장 많은 연령대를 검색해본 결과 20대가 가장 많았다.

### 3) 신촌동 내 아파트 시세에 따른 주민 분포

```
SELECT 아파트가격 FROM APT
WHERE 가구수 > (SELECT AVG(가구수) FROM APT)
```

| | 아파트가격 |
|---|---|
| 1 | 1억미만 |
| 2 | 1억대 |

- 아파트 시세에 따른 주민 분포를 검색해본 결과, 대부분의 주민들이 1억대나 그 미만

의 아파트에서 거주하고 있었다. 주민 대부분이 낮은 시세의 아파트에서 거주하는 것으로 보아, 신촌동 주민의 연령대는 젊을 것이라고 추측할 수 있다.

### 4) 종합 결과

주민과 유동 인구를 분석해본 결과를 종합해보면, 신촌동의 거주민 또는 유동 인구 대부분은 20대 여성임을 알 수 있다. 따라서 우리는 20대 여성을 타겟으로 한 호프집을 창업한다면 안정적인 매출을 기대할 수 있을 것이다.

# 2. 인터페이스 및 데이터베이스 설계

## 1) 인터페이스 설계

: 매장에서 사용할 수 있는 고객정보관리, 메뉴관리, 예약관리, 주문관리 프로그램의 인터페이스

**별첨) 주문관리_인터페이스.ppt**

## 2) 데이터베이스 설계

### (1) 요구사항 정의서

각 **고객 정보**에는 고유한 고객번호, 이름, 나이, 성별, 주소, 연락처를 저장한다.

**예약 정보**에는 고유한 예약 번호, 예약자 이름, 예약 시간, 예약 인원수, 예약자 연락처를 저장한다. 또한 예약자는 반드시 고객 정보에도 저장되어야 한다. 예약 정보는 고객 정보를 참조한다.

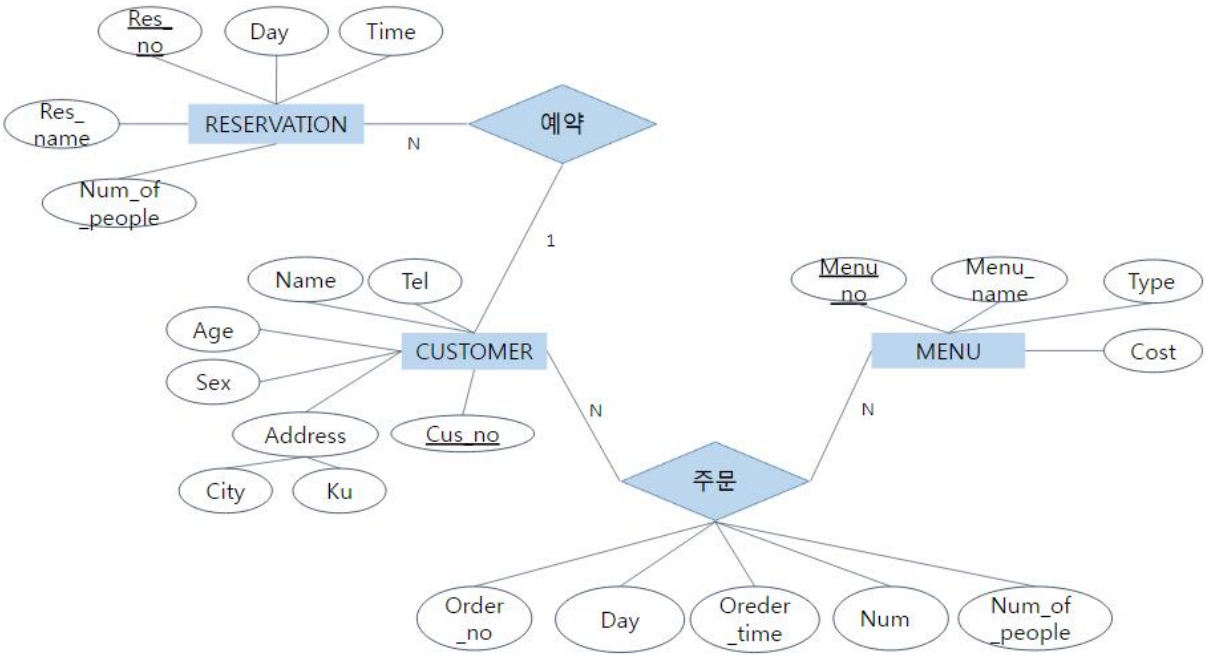
**메뉴**에는 각종 맥주 및 안주 품목이 있다. 각 품목에 대하여 고유 메뉴 번호, 메뉴 이름, 가격을 저장한다. 또한 각 품목의 종류도 저장한다. (주류/안주류)

한 테이블에서 여러 품목을 구입하는 것을 **주문**이라 한다. 각 주문에 대해 어느 시간에, 어떤 품목을, 얼마나 구매했는지를 저장한다. 주문은 고유 주문 번호, 주문 시간, 주문 품목, 각 품목에 대한 주문 수량을 저장한다.

이외에도 아래 6가지 조건을 만족해야 한다.

1. 같은 예약자가 다른 날에 예약을 2번 이상 할 수 있다.
2. 예약 번호의 경우 일주일마다 새로 갱신된다.
3. 일행 중에 대표자 한 명의 고객정보만 기록하고, 이를 기준으로 주문을 받는다.
4. 추가 주문은 받지 않는다.
5. 날짜보다는 요일별로 매출, 판매량 등의 통계를 산출할 것이므로, 모든 시간 정보는 요일과 시각을 나눠서 저장한다. 날짜는 따로 저장하지 않는다.
6. 예약을 받기 전, 반드시 고객 정보를 먼저 등록해야 한다.

### (2) 개념적 설계 : <ERD>




## (3) 논리적 설계 : ER 다이어그램 – 관계 모델 릴레이션 사상

(※ 따로 적용 해당 사항이 없는 단계는 생략하였다.)

(※ 외래키는 파란색으로 표시하였다.)

### 단계 1: 정규 엔티티 타입과 단일 값 애트리뷰트

- 복합 애트리뷰트는 해당 복합 애트리뷰트를 구성하는 단순 애트리뷰트들만 릴레이션에 포함시킨다.

**[CUSTOMER]**

| **Cus_no** | Name | Tel | Age | Sex | City | Ku |
|---|---|---|---|---|---|---|
| | | | | | | |

**[MENU]**

| **Menu_no** | Type | Menu_name | Cost |
|---|---|---|---|
| | | | |

**[RESERVATION]**

| **Res_no** | Res_name | Day | Time | Num_of_people |
|---|---|---|---|---|
| | | | | |

- [고객]의 Address의 경우, Address를 구성하는 단순 애트리뷰트인 City와 Ku만 릴레이션에 포함시킨다.

✓ CUSTOMER (Cus_no, Name, Tel, Age, Sex, City, Ku)

✓ MENU (Menu_no, Menu_name, Cost, Type)

✓ RESERVATION (Res_no, Res_name, Day, Time, num_of_people)

## 단계 4: 정규 2진 1:N 관계 타입

**[CUSTOMER]**

| **Cus_no** | Name | Tel | Age | Sex | City | Ku |
|---|---|---|---|---|---|---|
| | | | | | | |

**[RESERVATION]**

| **Res_no** | **Res_phone** | Res_name | Day | Time | Num_of_people |
|---|---|---|---|---|---|
| | | | | | |

- 정규 2진 1:N 관계 타입 <예약>에 대하여, N측의 참여 엔티티 타입에 대응되는 릴레이션 [예약]을 찾는다. 그 다음, 1측의 엔티티 타입에 대응되는 릴레이션 [고객]의 '기본키'를 릴레이션 [예약]에 외래키로 포함시킨다.

- [고객]의 기본 키는 고객번호인 Cus_no이기 때문에 원래대로라면 [예약]에서는 [고객]의 Cus_no를 참조하는 외래키를 생성해야 한다. 그러나 [예약]에 이미 고객 정보 중 하나인 Res_name이 존재하므로, 동명이인의 예약자를 구분하기 위해 [고객]의 Tel을 외래키로 참조하는 Res_phone 애트리뷰트를 생성하여 이를 [예약]의 기본키로 지정하였다.

✓ RESERVATION ( Res_no, Res_phone, Res_name, Day, Time, numofpeople)

### 단계 5: 2진 M:N 관계 타입

**[CUSTOMER]**

| **Cus_no** | Name | Tel | Age | Sex | City | Ku |
|---|---|---|---|---|---|---|
| | | | | | | |

**[ORDER]**

| **Order_no** | Cus_no | Menu_no | Day | Order_time | Num | Num_of_people |
|---|---|---|---|---|---|---|
| | | | | | | |

**[MENU]**

| **Menu_no** | Type | Menu_name | Cost |
|---|---|---|---|
| | | | |

- 2진 M:N 관계 타입 <주문>에 대해서는 릴레이션 [주문]을 생성한다. 참여 엔티티 타입에 해당하는 릴레이션들인 [고객], [메뉴]의 기본키를 릴레이션 [주문]에 외래키로 포함시키고, 이들의 조합은 곧 릴레이션 [주문]의 기본키가 된다. 또한 관계 타입 <주문>이 가지고 있는 모든 단순 애트리뷰트들을 릴레이션 [주문]에 포함시킨다.
- [주문]의 'Menu_no'는 [MENU]의 'Menu_no'를 참조하는 외래키이자 기본키이다.
- [주문]의 'Cus_no'는 [고객]의 'Cus_no'를 참조하는 외래키이자 기본키이다.

✓ ORDER (Order_no, Cus_no, Menu_no, Day, Order_time, Num)

## (4) 스키마 정제 : 데이터베이스 스키마 정규화

### 제1 정규형

- [주문] 릴레이션에서 손님 1명이 주문하는 메뉴가 많으면 **반복그룹**을 가진다.

**[ORDER]**

| **Order_no** | **Cus_no** | **Menu_no** | **Day** | **Order_time** | **Num** | **num_of_people** |
|---|---|---|---|---|---|---|
| **1** | 1 | 1 | 월 | 18:00 | 1 | 5 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | 2 | | | 1 | |
| **2** | 2 | 2 | 화 | 17:30 | 2 | 3 |
| | | 3 | | | 4 | |

- 따라서 반복 그룹 애트리뷰트에 나타나는 집합에 속한 각 값들을 하나의 튜플로 표현해주어 각 애트리뷰트가 원자 값을 갖도록 한다.

**[ORDER]**

| **Order_no** | **Cus_no** | **Menu_no** | **Day** | **Order_time** | **Num** | **num_of_people** |
|---|---|---|---|---|---|---|
| **1** | 1 | 1 | 월 | 18:00 | 1 | 5 |
| **1** | 1 | 2 | 월 | 18:00 | 1 | 5 |
| **2** | 2 | 2 | 화 | 17:30 | 2 | 3 |
| **2** | 2 | 3 | 화 | 17:30 | 4 | 3 |

## 제2정규형

### ■ [주문] 릴레이션

- 제1정규형을 만족하는 [주문] 릴레이션에서 **부분 함수적 종속성**이 존재하는지 확인해보았다.

  (**Order_no, Menu_no, Cus_no**) → Num, Day, Order_time, Num_of_people

  (**Order_no, Cus_no** ) → Day, Order_time, Num_of_people

  (**Order_no, Menu_no**) → Num

- 확인 결과, 수량(Num)에서 부분함수적 종속성이 존재하고, 요일(Day), 시간(Order_time), 인원수(Num_of_People) 애트리뷰트에서 부분 함수적 종속성이 존재하므로 부분 함수적 종속성이 존재하지 않도록 릴레이션을 분해하는 방법이 두 가지 존재한다.
- 수량(Num)만 분해하여 릴레이션을 따로 생성하는 것이 더 편리하고 효율적이므로, 수량 애트리뷰트를 따로 분리해 [주문목록]이라는 릴레이션을 생성함으로써 부분 함수적 종속성을 제거한다.

  **[ORDER]**

| Order_no | Cus_no | Day | Order_time | num_of_people |
|---|---|---|---|---|
| 1 | 1 | 월 | 18:00 | 5 |
| 2 | 2 | 화 | 17:30 | 3 |

**[ORDER_LIST]**

| Order_no | Menu_no | Num |
|---|---|---|
| **1** | 1 | 1 |
| **1** | 2 | 1 |
| **2** | 2 | 2 |
| **2** | 3 | 4 |

- 그러나, 제2정규형을 거친 [주문] 릴레이션이 여전히 복합 키를 가지고 있어서 함수 종속성을 다시 한번 확인해봤다.

**[ORDER]**

| Order_no | Cus_no | Day | Order_time | num_of_people |
|---|---|---|---|---|

(**Order_no**, **Cus_no**) → Day, Order_time, Num_of_people

(**Order_no**) → Day, Order_time, Num_of_people_no

- 확인 결과, Cus_no에서 부분 함수적 종속성이 존재하였다. 따라서 부분 함수적 종속성이 존재하지 않도록 릴레이션을 또 분해해준다.

**[ORDER]**

| Order_no | Day | Order_time | num_of_people |
|---|---|---|---|
| **1** | 월 | 18:00 | 5 |
| **2** | 화 | 17:30 | 3 |

| Order_no | Cus_no |
|---|---|
| **1** | **1** |
| **2** | **2** |

■ **[예약] 릴레이션**

| Res_no | Res_phone | Res_name | Day | time | num_of_people |
|---|---|---|---|---|---|

**(Res_no, Res_phone)** → Res_name, Day, Time, Num_of_people

**(Res_phone)** → Res_name

**(Res_no)** → Day, Time, Num_of_people

(※ 참고: 동명이인 때문에 res_no → res_name은 성립하지 않는다.)

- 위와 같이 Res_name에서 부분함수적 종속성이, Res_name 외의 나머지 애트리뷰트에서 부분 함수적 종속성이 각각 존재하므로 부분 함수적 종속성이 존재하지 않도록 릴레이션을 분해하는 방법은 두 가지가 있다.
- 두 가지 방법 중 Res_name을 분리하는 것이 더 편리하므로 Res_name을 분리하여 새로운 릴레이션을 생성한다.

**[RESERVATION]**

| Res_no | Day | Time | num_of_people |
|---|---|---|---|

| Res_no | Res_phone | Res_name |
|---|---|---|

## 제3정규형, BCNF

- 제2정규형 적용 후, 이행적 종속성이 존재하지 않아 제3정규형 과정을 진행하지 않았다.
- 또한 키가 아닌 애트리뷰트가 다른 애트리뷰트를 결정하는 경우도 없었기 때문에 BCNF 과정을 진행하지 않았다.

## 역정규화

■ **[주문] 릴레이션**

| Order_no | Day | Order_time | num_of_people |
|---|---|---|---|

| Order_no | Cus_no |
|---|---|

- 제2정규형을 통해 분해한 주문 릴레이션은 역정규화하기로 결정했다.

**[역정규화된 주문 테이블]**

| Order_no | Cus_no | Day | Order_time | num_of_people |
|---|---|---|---|---|

- 역정규화를 통해 Cus_no를 한 릴레이션에 담은 이유는, 해당 메뉴를 주문한 고객의 특성을 분석하는 등의 업무를 수행할 때 다단계의 조인을 해야 하는 불편함을 줄이기 위해서이다. 일주일동안의 주문 데이터를 분석할 때 메뉴와 고객 간의 데이터 분석이 많이 수행될텐데, 만약 릴레이션을 분해한다면 분석 과정에서 너무 많은 릴레이션에 접근해야 하기 때문이다.

## ■ [예약] 릴레이션

| Res_no | Day | Time | num_of_people |
|---|---|---|---|

| Res_no | Res_phone | Res_name |
|---|---|---|

- 제2정규형을 통해 분해한 예약 릴레이션 역시 역정규화하기로 결정했다.

**[역정규화된 예약 테이블]**

| Res_no | Res_phone | Res_name | Day | Time | Num_of_people |
|---|---|---|---|---|---|

- 역정규화해서 Res_name까지 한 릴레이션에 담은 이유 역시 다단계의 조인 연산 없이 빠른 조회를 하기 위해서이다. 뿐만 아니라, 이름을 개명하는 경우가 흔치 않으므로 수정 갱신 이상의 데미지가 크지 않을 것이기 때문에 굳이 릴레이션을 따로 분해하지 않아도 무방할 것이라 판단하였다.

### 최종적으로 정규화된 릴레이션 스키마

- ✓ CUSTOMER(Cus_no, Name, Tel, Age, Sex, City, Ku)
- ✓ MENU(Menu_no, Menu_name, Cost, Type)
- ✓ ORDER(Order_no, **Cus_no**, Day, Order_time, Num_of_people)
- ✓ ORDER_LIST(Order_no, **Menu_no**, Num)
- ✓ RESERVATION( Res_no, **Res_phone**, **Res_name**, Day, Time, Numofpeople)

## (5) 테스트 데이터 적재 : 실제와 유사한 샘플 데이터를 적재한다.(일주일)

### ■ 테스트 데이터 목록

- ✓ 고객 정보: 고객 번호, 이름, 전화번호, 나이, 성별, 시, 구
- ✓ 예약 정보: 예약 번호, 예약자 이름, 전화번호, 예약 시간, 예약 요일, 인원 수
- ✓ 메뉴: 메뉴번호, 메뉴 이름, 메뉴 종류, 가격
- ✓ 주문 정보: 주문 번호, 손님 번호, 주문 요일, 주문 시간, 인원 수
- ✓ 주문 목록: 주문 번호, 메뉴 번호, 수량

### ■ 테스트 데이터 적재

#### ① 테스트 데이터를 적재하기 위한 테이블 생성

- 고객 테이블

```
CREATE TABLE CUSTOMER(
    CUS_NO INT PRIMARY KEY NOT NULL,
    NAME CHAR(10) NOT NULL,
    TEL VARCHAR(20) UNIQUE NOT NULL,
    AGE INT NOT NULL,
    SEX CHAR(5) NOT NULL,
    CITY VARCHAR(20) NOT NULL,
    KU VARCHAR(20) NOT NULL,
)
```

- 예약 테이블

```
CREATE TABLE RESERVATION(
    RES_NO INT NOT NULL,
    RES_NAME CHAR(10) NOT NULL,
    RES_PHONE VARCHAR(20) NOT NULL,
```

```
    RES_TIME TIME NOT NULL,
    RES_DAY CHAR(5) NOT NULL,
    NUM_OF_PEOPLE INT NOT NULL,
    PRIMARY KEY(RES_NO, RES_PHONE),
    FOREIGN KEY(RES_PHONE) REFERENCES CUSTOMER(TEL) ON UPDATE CASCADE
)
```

- 메뉴 테이블

```
CREATE TABLE MENU(
    MENU_NO INT PRIMARY KEY NOT NULL,
    MENU_NAME VARCHAR(20) NOT NULL,
    MENU_TYPE CHAR(10) NOT NULL,
    COST INT NOT NULL
)
```

- 주문 테이블

```
CREATE TABLE HOF_ORDER(
    ORDER_NO INT NOT NULL,
    CUS_NO INT NOT NULL,
    ORDER_DAY CHAR(5) NOT NULL,
    ORDER_TIME TIME NOT NULL,
    NUM_OF_PEOPLE INT NOT NULL,
    PRIMARY KEY(ORDER_NO),
    FOREIGN KEY(CUS_NO) REFERENCES CUSTOMER(CUS_NO) ON UPDATE CASCADE
)
```

- 주문 목록 테이블

```
CREATE TABLE ORDER_LIST(
    ORDER_NO INT NOT NULL,
    MENU_NO INT NOT NULL,
    NUM INT NOT NULL,
    PRIMARY KEY(ORDER_NO, MENU_NO),
    FOREIGN KEY(ORDER_NO) REFERENCES HOF_ORDER(ORDER_NO),
    FOREIGN KEY(MENU_NO) REFERENCES MENU(MENU_NO)
)
```

**② 테스트 데이터가 저장된 엑셀 파일을 각각의 테이블에 삽입**

**③ 테스트 데이터 적재 결과**

- 고객 테이블

```
SELECT * FROM CUSTOMER;
```

결과 | 메시지

| | CUS_NO | NAME | TEL | AGE | SEX | CITY | KU |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 박수진 | 010-3828-3213 | 23 | 여성 | 경기도 | 구리시 |
| 2 | 2 | 강유리 | 010-4724-0277 | 24 | 여성 | 서울시 | 서대문구 |
| 3 | 3 | 송해슬 | 010-5325-9392 | 24 | 여성 | 경기도 | 고양시 |
| 4 | 4 | 도솔미 | 010-2223-2357 | 24 | 여성 | 서울시 | 서대문구 |
| 5 | 5 | 유채원 | 010-3802-0023 | 20 | 여성 | 서울시 | 광진구 |
| 6 | 6 | 곽두환 | 010-1204-3204 | 20 | 여성 | 서울시 | 서대문구 |
| 7 | 7 | 홍길동 | 010-1817-1518 | 22 | 여성 | 서울시 | 마포구 |
| 8 | 8 | 최지혜 | 010-8282-7979 | 27 | 여성 | 서울시 | 서대문구 |
| 9 | 9 | 이채영 | 010-2323-2323 | 23 | 여성 | 서울시 | 성동구 |
| 10 | 10 | 김규환 | 010-0504-1412 | 22 | 남성 | 서울시 | 관악구 |
| 11 | 11 | 김지연 | 010-3903-3833 | 23 | 여성 | 서울시 | 관악구 |
| 12 | 12 | 박예진 | 010-4989-8834 | 23 | 여성 | 경기도 | 인천시 |
| 13 | 13 | 권현식 | 010-3221-1112 | 24 | 남성 | 서울시 | 강남구 |
| 14 | 14 | 황주아 | 010-8282-5353 | 22 | 여성 | 서울시 | 강남구 |
| 15 | 15 | 김혜자 | 010-1597-3574 | 62 | 여성 | 서울시 | 마포구 |
| 16 | 16 | 정희유 | 010-7611-2179 | 21 | 여성 | 서울시 | 마포구 |
| 17 | 17 | 사은지 | 010-4444-5555 | 21 | 여성 | 서울시 | 서대문구 |
| 18 | 18 | 박민정 | 010-9879-0987 | 27 | 여성 | 서울시 | 서대문구 |
| 19 | 19 | 최지훈 | 010-5382-7777 | 23 | 남성 | 서울시 | 서대문구 |
| 20 | 20 | 황세리 | 010-2234-4458 | 31 | 여성 | 서울시 | 서대문구 |
| 21 | 21 | 손건호 | 010-5561-5603 | 22 | 남성 | 서울시 | 송파구 |

- 예약 테이블

```
SELECT * FROM RESERVATION;
```

결과 메시지

| | RES_NO | RES_NAME | RES_PHONE | RES_TIME | RES_DAY | NUM_OF_PEOPLE |
|---|---|---|---|---|---|---|
| 1 | 1 | 도솔미 | 010-2223-2357 | 19:00:00.0000000 | 월 | 3 |
| 2 | 2 | 김혜자 | 010-1597-3574 | 20:30:00.0000000 | 월 | 4 |
| 3 | 3 | 박민정 | 010-9879-0987 | 21:00:00.0000000 | 월 | 3 |
| 4 | 4 | 유예지 | 010-9823-1237 | 18:30:00.0000000 | 화 | 4 |
| 5 | 5 | 최민지 | 010-2323-3131 | 19:30:00.0000000 | 화 | 4 |
| 6 | 6 | 정미연 | 010-1004-9987 | 21:00:00.0000000 | 화 | 4 |
| 7 | 7 | 화예희 | 010-1997-1996 | 22:30:00.0000000 | 화 | 3 |
| 8 | 8 | 김지선 | 010-9802-4370 | 18:30:00.0000000 | 수 | 4 |
| 9 | 9 | 이유리 | 010-3334-1298 | 19:00:00.0000000 | 수 | 4 |
| 10 | 10 | 성수진 | 010-8732-3210 | 20:30:00.0000000 | 수 | 3 |
| 11 | 11 | 박가연 | 010-4398-3219 | 21:00:00.0000000 | 수 | 4 |
| 12 | 12 | 한도연 | 010-2840-9317 | 21:00:00.0000000 | 수 | 3 |
| 13 | 13 | 김민기 | 010-9982-2342 | 19:00:00.0000000 | 목 | 4 |
| 14 | 14 | 윤민영 | 010-4627-2392 | 20:00:00.0000000 | 목 | 4 |
| 15 | 15 | 조유진 | 010-1972-0902 | 20:30:00.0000000 | 목 | 3 |
| 16 | 16 | 나나라 | 010-4821-3218 | 21:30:00.0000000 | 목 | 4 |
| 17 | 17 | 최영주 | 010-7827-3917 | 22:00:00.0000000 | 목 | 3 |
| 18 | 18 | 성지원 | 010-6828-5728 | 22:30:00.0000000 | 목 | 4 |
| 19 | 19 | 민지원 | 010-4892-5738 | 18:00:00.0000000 | 금 | 4 |
| 20 | 20 | 한슬기 | 010-5739-4729 | 19:00:00.0000000 | 금 | 3 |
| 21 | 21 | 정민형 | 010-9872-0217 | 19:00:00.0000000 | 금 | 3 |

- 메뉴 테이블

```
SELECT * FROM MENU;
```

결과 메시지

| | MENU_NO | MENU_NAME | MENU_TYPE | COST |
|---|---|---|---|---|
| 1 | 1 | 호가든 | 맥주 | 6000 |
| 2 | 2 | 아사히 | 맥주 | 5000 |
| 3 | 3 | 하이네켄 | 맥주 | 6000 |
| 4 | 4 | 코로나 | 맥주 | 7000 |
| 5 | 5 | 블랑 | 맥주 | 5000 |
| 6 | 6 | 칭타오 | 맥주 | 6000 |
| 7 | 7 | 버드와이저 | 맥주 | 5000 |
| 8 | 8 | 스텔라 | 맥주 | 7000 |
| 9 | 9 | 밀러 | 맥주 | 6000 |
| 10 | 10 | 코젤다크 | 맥주 | 5000 |
| 11 | 11 | 기네스 | 맥주 | 8000 |
| 12 | 12 | 기린 이치방 | 맥주 | 7000 |
| 13 | 13 | 산미구엘 | 맥주 | 7000 |
| 14 | 14 | 파울라너헤페 | 맥주 | 11000 |
| 15 | 15 | 에딩거헤페 | 맥주 | 11000 |
| 16 | 16 | 삿포로 | 맥주 | 9000 |
| 17 | 17 | 버드라이트 | 맥주 | 5000 |
| 18 | 18 | 스콜 | 맥주 | 5000 |
| 19 | 19 | 카프리 | 맥주 | 4000 |
| 20 | 20 | 백스 | 맥주 | 6000 |
| 21 | 21 | 페페로니피자 | 안주 | 13000 |

- 주문 테이블

```
SELECT * FROM HOF_ORDER;
```

결과 메시지

| | ORDER_NO | CUS_NO | ORDER_DAY | ORDER_TIME | NUM_OF_PEOPLE |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 월 | 17:13:00.0000000 | 3 |
| 2 | 2 | 2 | 월 | 18:12:00.0000000 | 1 |
| 3 | 3 | 3 | 월 | 18:21:00.0000000 | 2 |
| 4 | 4 | 4 | 월 | 19:00:00.0000000 | 3 |
| 5 | 5 | 5 | 월 | 19:01:00.0000000 | 3 |
| 6 | 6 | 6 | 월 | 19:11:00.0000000 | 4 |
| 7 | 7 | 7 | 월 | 19:13:00.0000000 | 4 |
| 8 | 8 | 8 | 월 | 19:21:00.0000000 | 4 |
| 9 | 9 | 9 | 월 | 19:40:00.0000000 | 4 |
| 10 | 10 | 10 | 월 | 19:51:00.0000000 | 2 |
| 11 | 11 | 11 | 월 | 20:01:00.0000000 | 1 |
| 12 | 12 | 12 | 월 | 20:13:00.0000000 | 4 |
| 13 | 13 | 13 | 월 | 20:22:00.0000000 | 2 |
| 14 | 14 | 14 | 월 | 20:29:00.0000000 | 2 |
| 15 | 15 | 15 | 월 | 20:30:00.0000000 | 4 |
| 16 | 16 | 16 | 월 | 20:34:00.0000000 | 3 |
| 17 | 17 | 17 | 월 | 20:54:00.0000000 | 1 |
| 18 | 18 | 18 | 월 | 21:00:00.0000000 | 3 |
| 19 | 19 | 19 | 월 | 21:01:00.0000000 | 4 |
| 20 | 20 | 20 | 월 | 21:18:00.0000000 | 4 |
| 21 | 21 | 21 | 월 | 21:22:00.0000000 | 4 |
| 22 | 22 | 22 | 월 | 21:23:00.0000000 | 2 |
| 23 | 23 | 23 | 월 | 21:31:00.0000000 | 4 |

- 주문목록 테이블

```
SELECT * FROM ORDER_LIST;
```

결과 메시지

| | ORDER_NO | MENU_NO | NUM |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 2 | 1 | 3 | 1 |
| 3 | 1 | 5 | 1 |
| 4 | 1 | 22 | 1 |
| 5 | 1 | 30 | 1 |
| 6 | 2 | 6 | 1 |
| 7 | 2 | 9 | 1 |
| 8 | 2 | 24 | 1 |
| 9 | 3 | 2 | 1 |
| 10 | 3 | 3 | 1 |
| 11 | 3 | 24 | 1 |
| 12 | 3 | 26 | 1 |
| 13 | 3 | 29 | 1 |
| 14 | 4 | 1 | 1 |
| 15 | 4 | 4 | 1 |
| 16 | 4 | 14 | 1 |
| 17 | 4 | 18 | 1 |
| 18 | 4 | 28 | 1 |
| 19 | 4 | 29 | 1 |
| 20 | 5 | 1 | 1 |
| 21 | 5 | 5 | 1 |

### (6) 주문관리 프로그램의 화면 설계에 맞는 SQL 작성 및 테스트

: 함께 첨부된 주문관리_인터페이스.ppt 를 참조.

# <애플리케이션>

1. 처음 온 모든 손님은 대표자 손님의 이름, 핸드폰 번호, 나이, 성별, 주소를 입력을 해야만 주문을 할 수 있다.
2. 손님에 관한 정보는 '손님 메뉴' 버튼을 통해서 볼 수 있으며, 손님의 정보를 입력하는 순서대로 자동적으로 손님 번호가 생성된다. -> CUSTOMER table에 넣어질 값
3. 손님 정보 페이지에는 '주문' 버튼과 '예약' 버튼이 존재한다.
4. 주문 버튼을 누르면 자동적으로 비어 있는 테이블에 손님 이름이 표기되어, 해당 손님으로 주문이 가능하다.
5. 이 때, 주문 버튼을 누른 순서대로 자동적으로 주문 번호가 생성되고, 해당 손님의 손님 번호, 해당 요일, 주문 버튼을 누른 시간의 정보가 자동 저장이 되고, 해당 손님의 일행을 입력한다. -> HOF_OREDR table에 넣어질 값
6. 테이블 아래에는 '주문 받기' 버튼이 존재한다.
7. 주문 받기 버튼을 누르면 맥주와 안주 중에서 하나를 선택하고, 해당 손님이 주문한 메뉴를 눌러준다. -> 해당 메뉴에 대한 정보는 미리 MENU table로 만들어져 있는 값이다.
8. 메뉴를 하나씩 선택할 때 마다, 해당 주문 번호와 메뉴 고유의 번호 그리고 메뉴의 개수가 저장이 된다. -> ORDER_LIST table에 넣어질 값
9. 모든 결제는 후결제이며, 추가 주문은 불가능하다.
10. 예약은 20분 전에만 가능하고, 전화로 예약한다.
11. 예약도 손님이기 때문에 손님 정보를 받아야 예약을 할 수 있다.
12. 손님 정보를 받은 후 예약 버튼을 누르면, 자동적으로 해당 손님의 이름과 전화번호가 들어있는 예약 페이지가 만들어진다.
13. 이 때, 예약 버튼을 누른 순서대로 자동적으로 예약 번호가 생성되고, (해당 번호는 일주일마다 갱신) 예약 시간과 예약한 요일과 일행을 입력한다. -> RESERVATION table에 넣어질 값
14. 예약 정보는 '예약 메뉴'에서 확인 가능하다.
15. 예약 정보에도 손님 정도와 동일한 '주문' 버튼이 존재하며, 그 기능은 위와 같다.

```sql
CREATE TABLE CUSTOMER(
    CUS_NO INT PRIMARY KEY NOT NULL,
    NAME CHAR(10) NOT NULL,
    TEL VARCHAR(20) UNIQUE NOT NULL,
    AGE INT NOT NULL,
    SEX CHAR(5) NOT NULL,
    CITY VARCHAR(20) NOT NULL,
    KU VARCHAR(20) NOT NULL,
)

CREATE TABLE RESERVATION(
    RES_NO INT NOT NULL,
    RES_NAME CHAR(10) NOT NULL,
    RES_PHONE VARCHAR(20) NOT NULL,
    RES_TIME TIME NOT NULL,
    RES_DAY CHAR(5) NOT NULL,
    NUM_OF_PEOPLE INT NOT NULL,

    PRIMARY KEY(RES_NO, RES_PHONE),

    FOREIGN KEY(RES_PHONE) REFERENCES CUSTOMER(TEL) ON UPDATE CASCADE
)
```

3. HOF_ORDER : '주문' 버튼을 통해서 입력된 값을 저장할 table
4. ORDER_LIST : '주문 받기' 버튼을 통해서 입력된 값을 저장할 table

애플리케이션과 연동될 table을 미리 만든다.

1. CUSTOMER : 손님 정보에 입력된 값을 저장할 table
2. RESERVATION : 예약 정보에 입력된 값을 저장할 table

```sql
CREATE TABLE HOF_ORDER(
    ORDER_NO INT NOT NULL,
    CUS_NO INT NOT NULL,
    ORDER_DAY CHAR(5) NOT NULL,
    ORDER_TIME TIME NOT NULL,
    NUM_OF_PEOPLE INT NOT NULL,

    PRIMARY KEY(ORDER_NO),

    FOREIGN KEY(CUS_NO) REFERENCES CUSTOMER(CUS_NO) ON UPDATE CASCADE
)

CREATE TABLE ORDER_LIST(
    ORDER_NO INT NOT NULL,
    MENU_NO INT NOT NULL,
    NUM INT NOT NULL,

    PRIMARY KEY(ORDER_NO, MENU_NO),

    FOREIGN KEY(ORDER_NO) REFERENCES HOF_ORDER(ORDER_NO),
    FOREIGN KEY(MENU_NO) REFERENCES MENU(MENU_NO)
)
```

| MENU_NO | MENU_NAME | MENU_TYPE | COST |
|---|---|---|---|
| 1 | 호가든 | 맥주 | 6000 |
| 2 | 아사히 | 맥주 | 5000 |
| 3 | 하이네켄 | 맥주 | 6000 |
| 4 | 코로나 | 맥주 | 7000 |
| 5 | 블랑 | 맥주 | 5000 |
| 6 | 칭타오 | 맥주 | 6000 |
| 7 | 버드와이저 | 맥주 | 5000 |
| 8 | 스텔라 | 맥주 | 7000 |
| 9 | 밀러 | 맥주 | 6000 |
| 10 | 코젤다크 | 맥주 | 5000 |
| 11 | 기네스 | 맥주 | 8000 |
| 12 | 기린 이치방 | 맥주 | 7000 |
| 13 | 산미구엘 | 맥주 | 7000 |
| 14 | 파울라너헤페 | 맥주 | 11000 |
| 15 | 에딩거헤페 | 맥주 | 11000 |
| 16 | 삿포로 | 맥주 | 9000 |
| 17 | 버드라이트 | 맥주 | 5000 |
| 18 | 스콜 | 맥주 | 5000 |
| 19 | 카프리 | 맥주 | 4000 |
| 20 | 백스 | 맥주 | 6000 |
| 21 | 페페로니피자 | 안주 | 13000 |
| 22 | 치즈피자 | 안주 | 13000 |
| 23 | 콤비네이션... | 안주 | 13000 |

메뉴 정보는 MENU 라는 table을 만들고 미리 정보를 저장해준다.

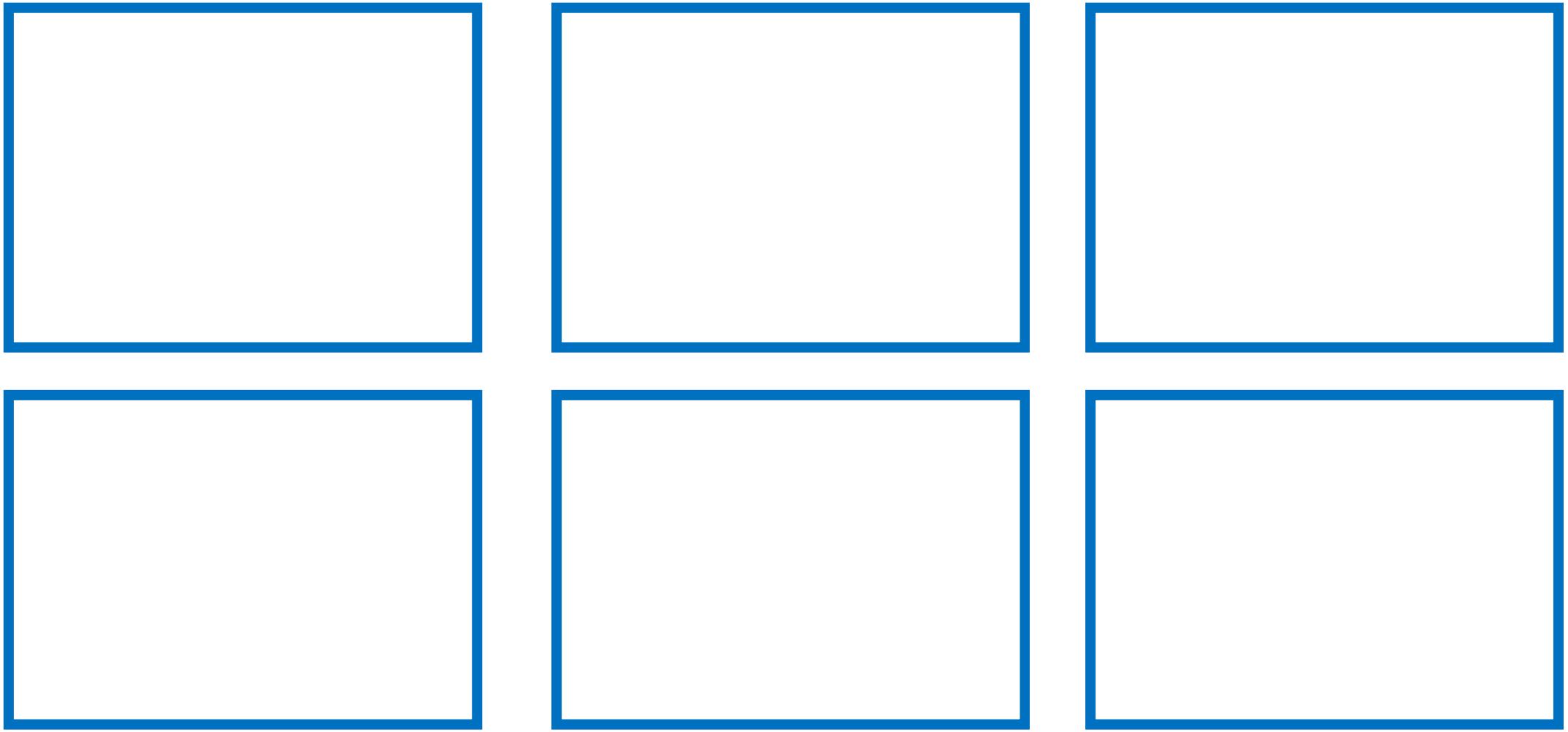
-> 메뉴는 가게에 미리 들어가 있는 정보이기 때문

```
REATE TABLE MENU(
    MENU_NO INT PRIMARY KEY NOT NULL,
    MENU_NAME VARCHAR(20) NOT NULL,
    MENU_TYPE CHAR(10) NOT NULL,
    COST INT NOT NULL
```

| | | | |
|---|---|---|---|
| 24 | 치킨 | 안주 | 11000 |
| 25 | 소세지 | 안주 | 10000 |
| 26 | 노가리 | 안주 | 9000 |
| 27 | 한치 | 안주 | 8000 |
| 28 | 나쵸 | 안주 | 5000 |
| 29 | 모듬튀김 | 안주 | 12000 |
| 30 | 감자튀김 | 안주 | 6000 |
| 31 | 마약옥수수 | 안주 | 5000 |

16:59 -> 현재 시간을 표시 (17:00~01:00 까지 운행)

해당 요일을 표기(월~일까지 운행) <- 월



애플리케이션의 초기 페이지

예약

손님

17:09

월

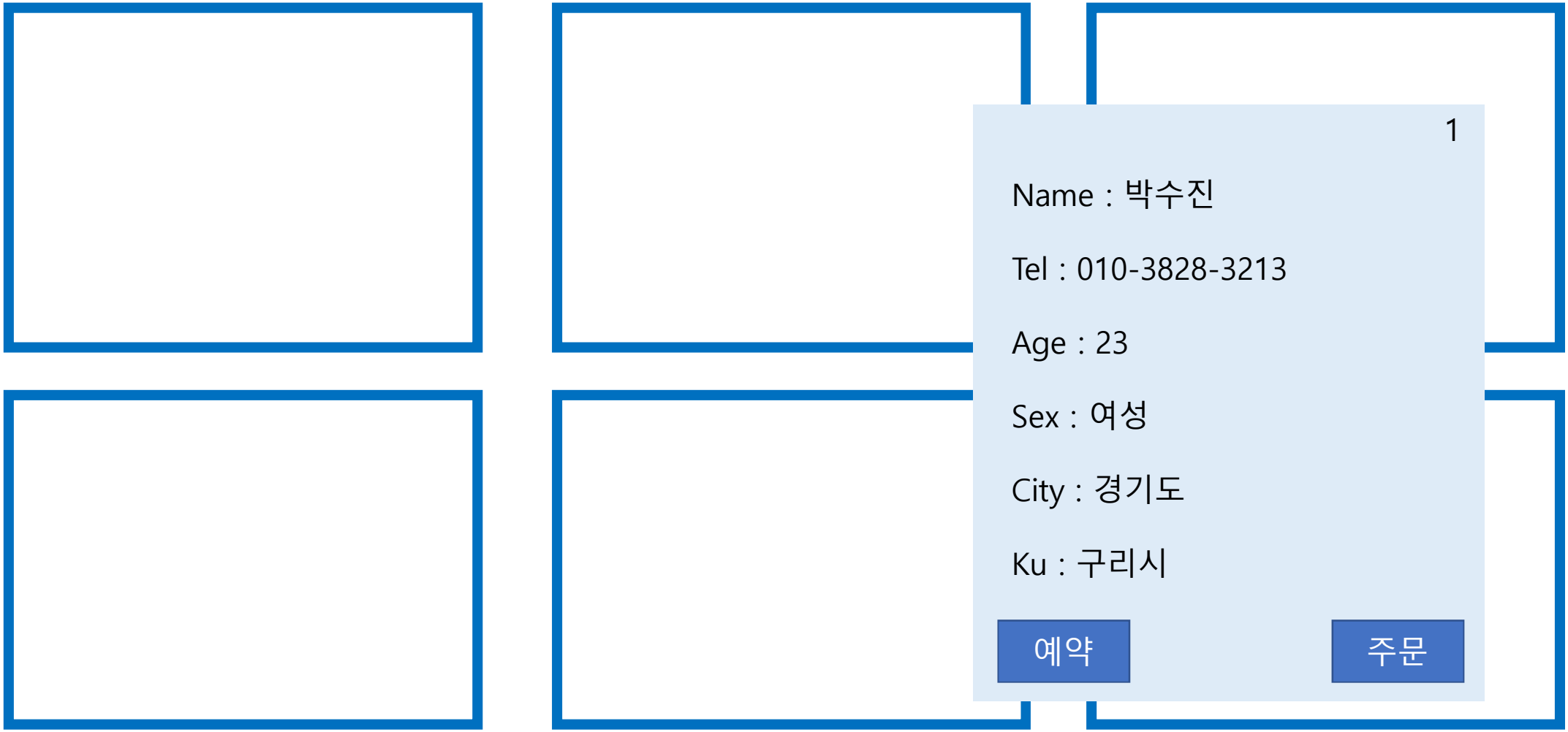
첫 번째 손님

예약

손님

17:10

월




첫 번째 손님의 손님 정보 입력

예약

손님

첫 번째 손님이 손님 정보 (이름, 전화번호, 나이, 성별, 도시, 구)를 입력한다.
 -> 손님 정보에 해당하는 번호로 손님 번호는 자동으로 저장

```
INSERT INTO CUSTOMER VALUES(1,  '박수진',  '010-3828-3213',  23, '여성', '경기도',  '구리시');
```

| CUS_NO | NAME | TEL | AGE | SEX | CITY | KU |
|---|---|---|---|---|---|---|
| 1 | 박수진 | 010-3828-3213 | 23 | 여성 | 경기도 | 구리시 |

17:13

월

1

Name : 박수진

Tel : 010-3828-3213

Age : 23

Sex : 여성

City : 경기도

Ku : 구리시

예약

주문

예약

손님

17:14 월

17:13 -> 주문 버튼을 누른 시간이 자동으로 저장 주문 버튼을 누른 시간 순서대로 자동 저장<- 1

박수진 (3) -> 손님 수는 직접 입력

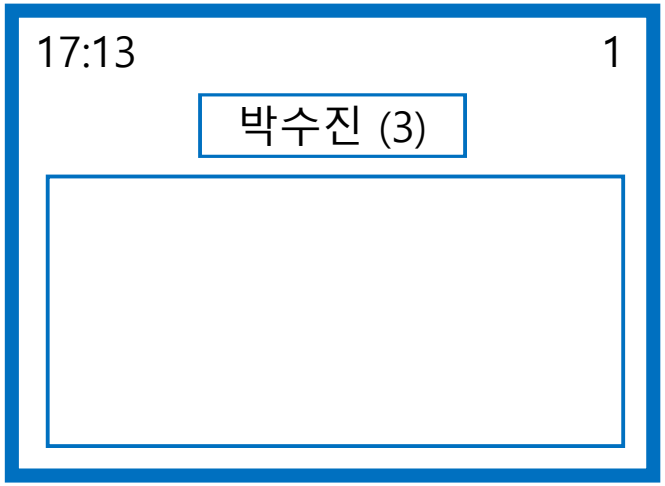
| 메뉴이름 | 메뉴 수 |
| --- | --- |

계산 -> 손님이 나가면 계산을 하고 테이블에서 삭제

주문 받기

예약 손님

17:14 월




밖으로 나갔을 때의 화면
(이전 페이지와 같은 화면으로 가고 싶을 경우에는 해당 테이블을 누르면 된다.)

손님의 일행 수를 입력한다.
나머지 정보는 자동적으로 입력이 되어, HOF_ORDER에 저장이 된다.

-> 주문 번호는 주문 버튼을 누른 시간 순서에 맞게 자동적으로 생성이 된다. 만약, 주문 버튼을 누른 시간이 동일하다면, 손님 번호가 빠른 것이 주문 번호도 빠르다.

-> 손님 번호는 해당 주문 버튼이 있었던 손님 번호를 자동 저장한다.

-> 요일 정보는 애플리케이션 (혹은 기계)에 이미 저장되어 있는 정보를 저장한다.

-> 주문한 시간은 주문 버튼을 누른 시간이 자동적으로 저장이 된다.

```
INSERT INTO HOF_ORDER VALUES(1, 1,  '월',   '17:13:00', 3);
```

| ORDER_NO | CUS_NO | ORDER_DAY | ORDER_TIME | NUM_OF_PEOPLE |
|---|---|---|---|---|
| 1 | 1 | 월 | 17:13:00.0000000 | 3 |

17:14 월

17:13

1

박수진 (3)

| 메뉴이름 | 메뉴 수 |
| --- | --- |

계산

주문 받기

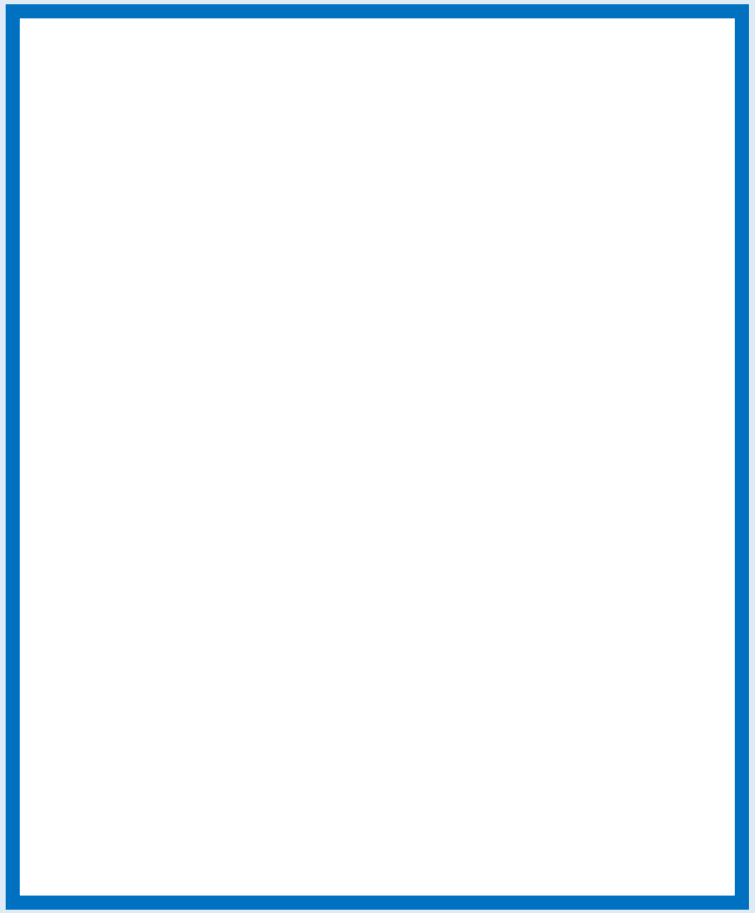
예약

손님

17:14 월

안주 맥주 1

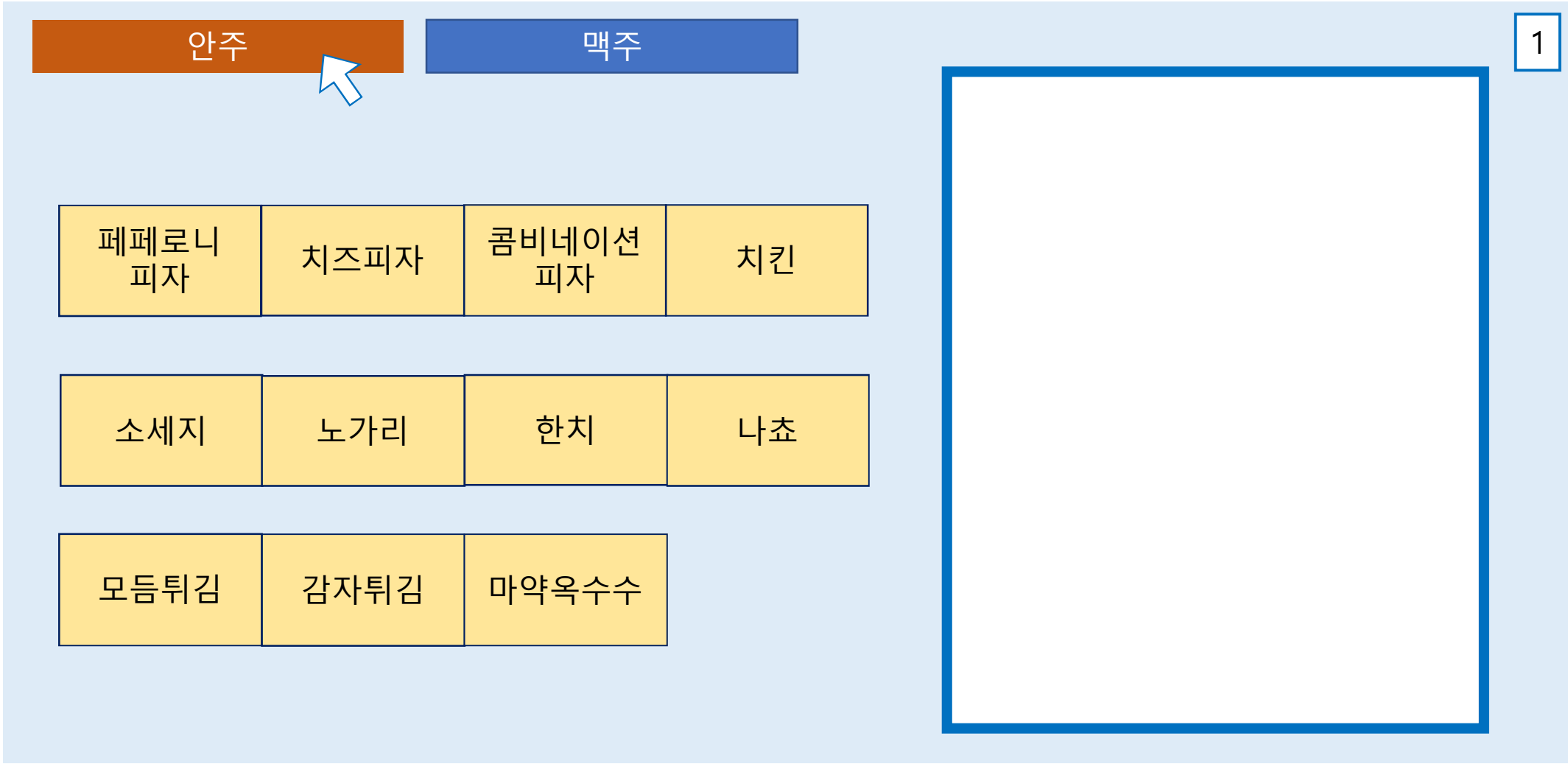
| | | | |
|---|---|---|---|
| 호가든 | 아사히 | 하이네켄 | 코로나 |
| 블랑 | 칭타오 | 버드와이저 | 스텔라 |
| 밀러 | 코젤다크 | 기네스 | 기린 이치방 |
| 산미구엘 | 파울라너헤페 | 에딩거헤페 | 삿포로 |
| 버드라이트 | 스콜 | 카프리 | 백스 |



맥주 메뉴가 들어있는 화면 (MENU에 저장되어 있는 메뉴 번호 순서)

예약 손님

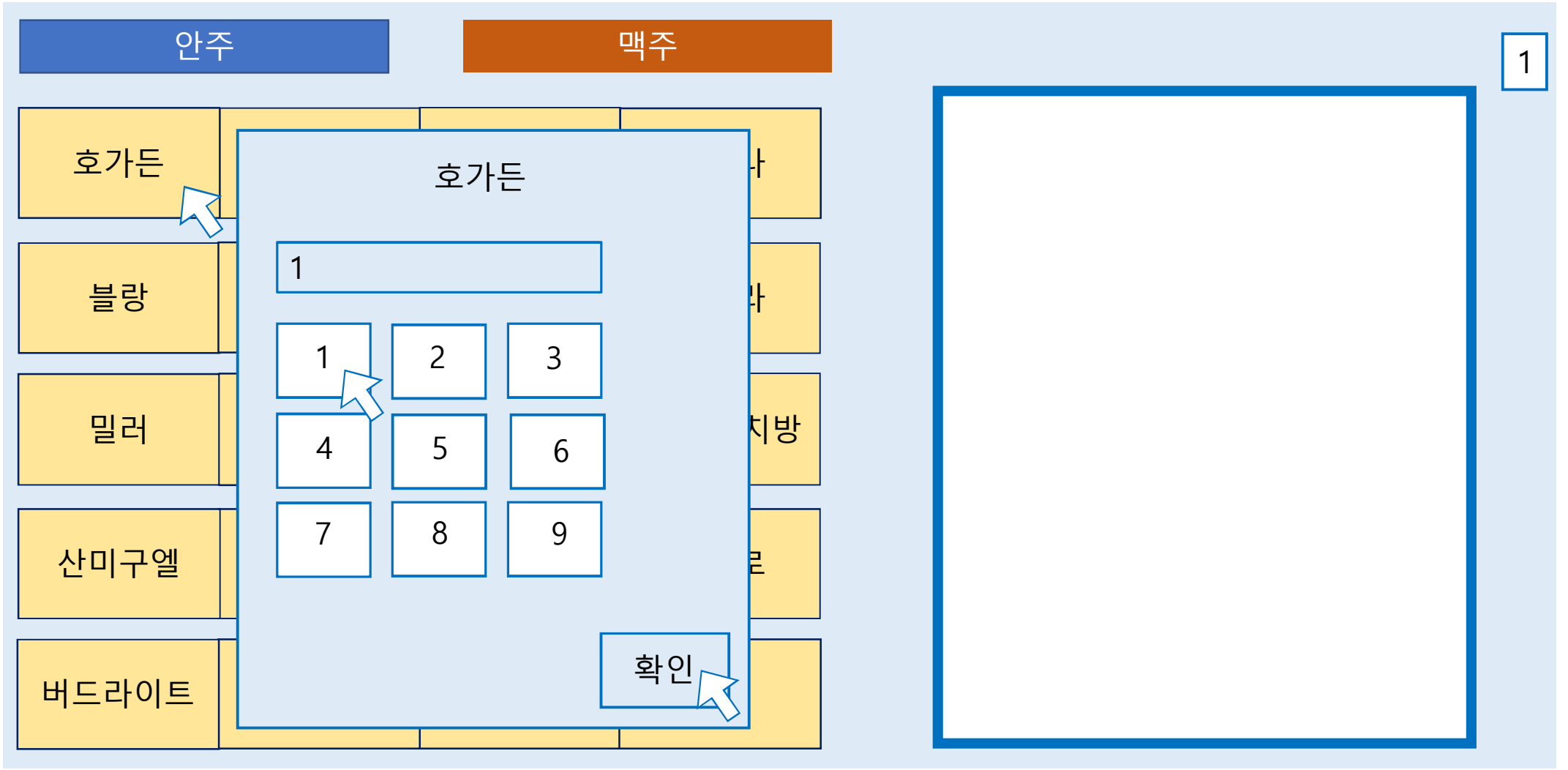
17:15 월




안주 메뉴가 들어있는 화면 (MENU에 저장되어 있는 메뉴 번호 순서)

예약 손님

17:15 월




손님이 주문한 맥주와 수량을 입력

예약 손님

손님이 주문한 해당 메뉴의 버튼을 선택하고, 수량을 입력한다.
입력한 정보가 ORDER_LIST에 저장된다.

-> 주문 번호는 해당 테이블에 저장되어 있는 주문 번호를 자동 저장한다.

-> 메뉴 번호는 MENU table에 저장되어 있는 것을 자동 저장한다.

```
INSERT INTO ORDER_LIST VALUES(1,    1,  1);
```

| ORDER_NO | MENU_NO | NUM |
|---|---|---|
| 1 | 1 | 1 |

17:15 월

안주 맥주

1

| | | | |
|---|---|---|---|
| 호가든 | 아사히 | 하이네켄 | 코로나 |
| 블랑 | 칭타오 | 버드와이저 | 스텔라 |
| 밀러 | 코젤다크 | 기네스 | 기린 이치방 |
| 산미구엘 | 파울라너헤페 | 에딩거헤페 | 삿포로 |
| 버드라이트 | 스콜 | 카프리 | 백스 |

호가든 1
하이네켄 1
블랑 1

손님이 주문한 맥주와 수량을 입력

예약 손님

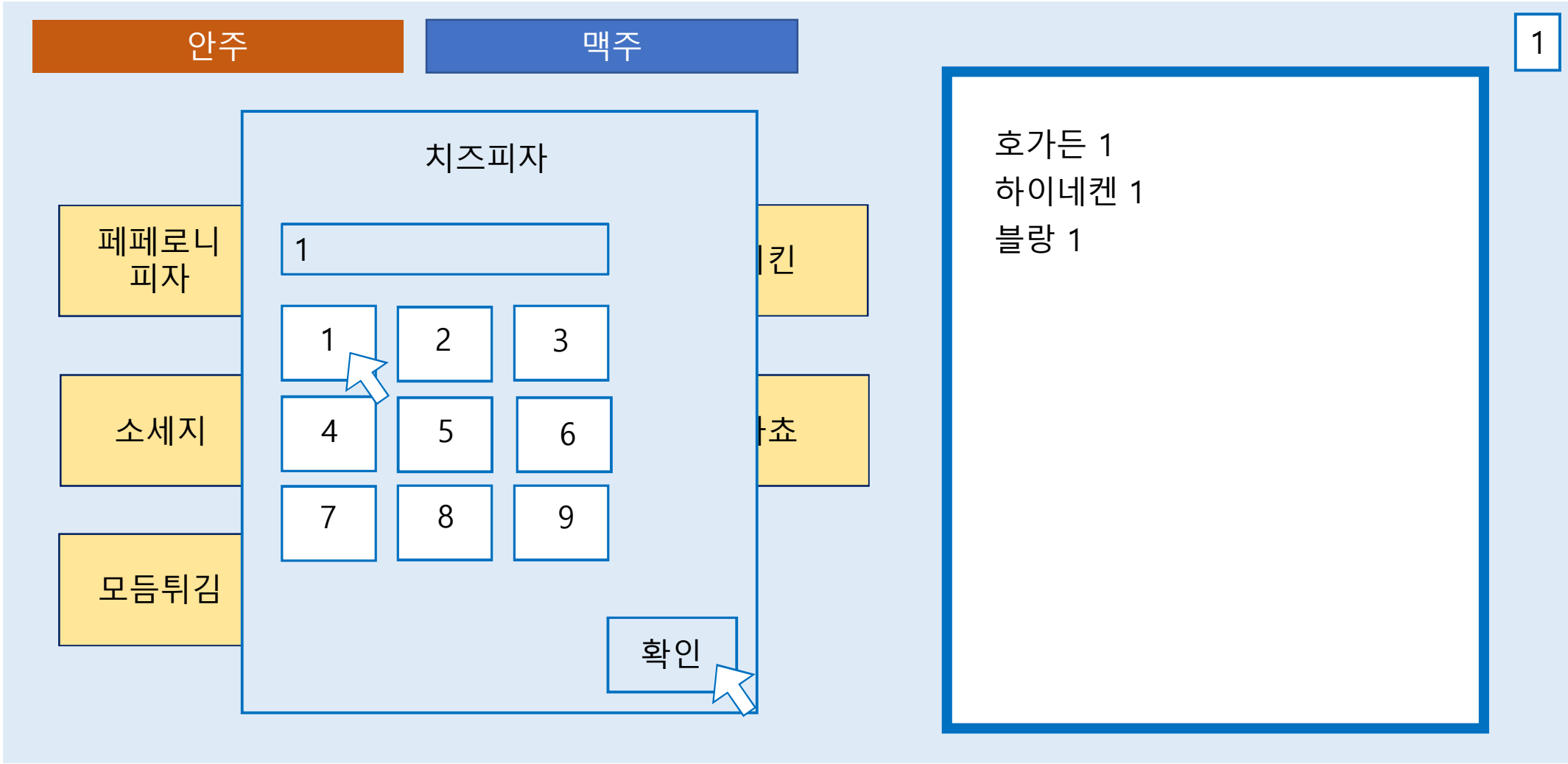
손님이 주문한 메뉴를 하나씩 입력을 한다.
입력 값 하나하나가 ORDER_LIST에 저장된다.

-> 확인 버튼을 누를 때 마다, 하나씩 값이 저장이 된다.

```
INSERT INTO ORDER_LIST VALUES(1,    1,  1);
INSERT INTO ORDER_LIST VALUES(1,    3,  1);
INSERT INTO ORDER_LIST VALUES(1,    5,  1);
```

| ORDER_NO | MENU_NO | NUM |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 3 | 1 |
| 1 | 5 | 1 |

17:15 월




손님이 주문한 안주와 수량을 입력

예약 손님

17:15 월

안주 맥주

1

| | | | |
|---|---|---|---|
| 페페로니 피자 | 치즈피자 | 콤비네이션 피자 | 치킨 |
| 소세지 | 노가리 | 한치 | 나쵸 |
| 모듬튀김 | 감자튀김 | 마약옥수수 | |

호가든 1
하이네켄 1
블랑 1
치즈피자 1
감자튀김 1

예약 손님

17:16 월

17:13

1

박수진 (3)

| 메뉴이름 | 메뉴 수 |
| --- | --- |
| 호가든 | 1 |
| 하이네켄 | 1 |
| 블랑 | 1 |
| 치즈피자 | 1 |
| 감자튀김 | 1 |

계산

주문 받기

밖으로 나갔을 때의 화면
(이전 페이지와 같은 화면으로 가고 싶을 경우에는 주문 받기를 누르면 된다.)

예약 손님

17:16 월

17:13 1

박수진 (3)

호가든 1
하이네켄 1
블랑 1
치즈피자 1
감자튀김 1

밖으로 나갔을 때의 화면

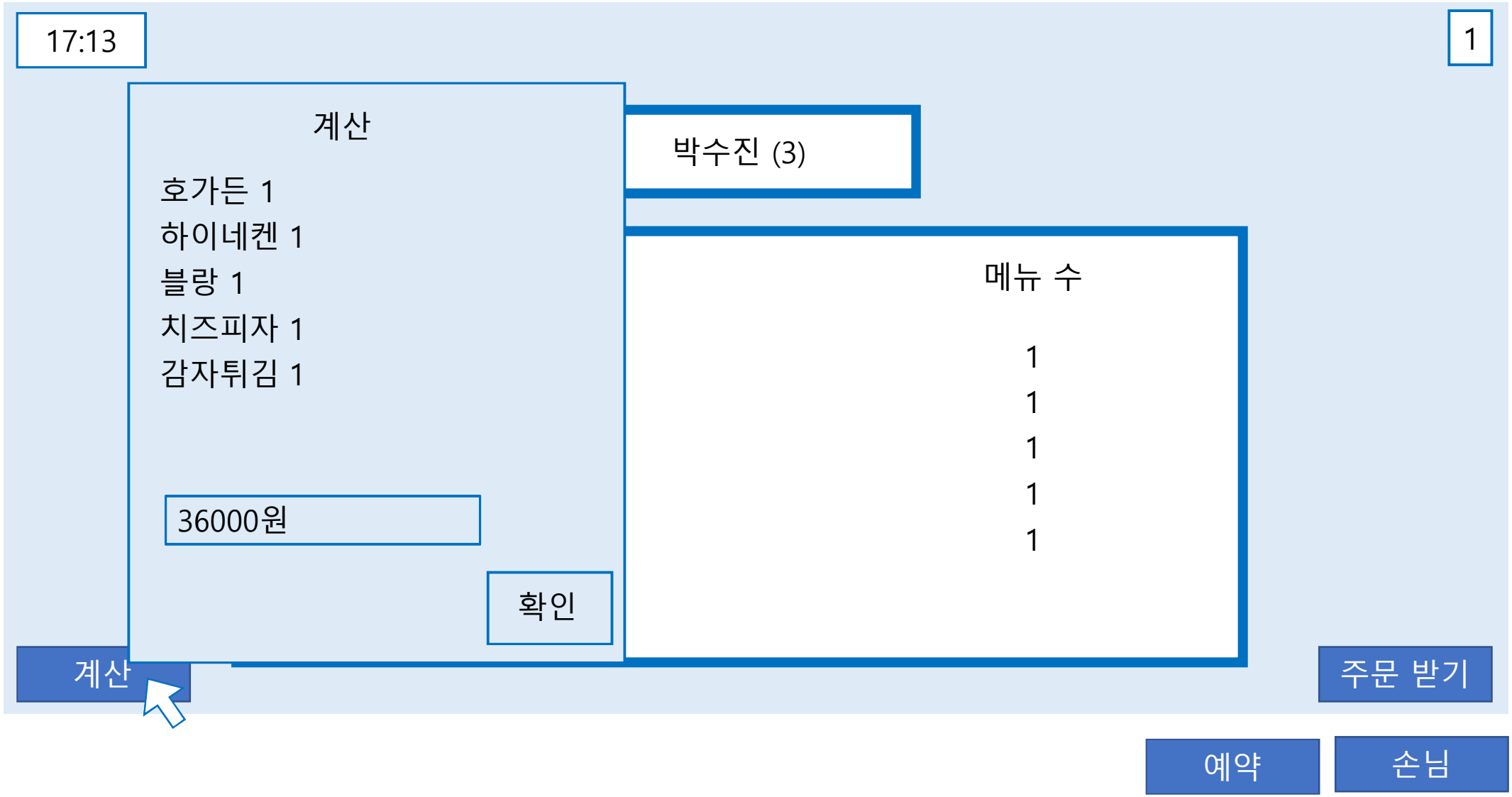
예약 손님

해당 손님이 최종적으로 시킨 메뉴와 수량

```
SELECT C.NAME, M.MENU_NAME, O.NUM FROM CUSTOMER C, MENU M, HOF_ORDER H, ORDER_LIST O
WHERE  H.ORDER_NO = O.ORDER_NO AND H.CUS_NO = C.CUS_NO AND M.MENU_NO = O.MENU_NO
```

| NAME | MENU_NAME | NUM |
|---|---|---|
| 박수진 | 호가든 | 1 |
| 박수진 | 하이네켄 | 1 |
| 박수진 | 블랑 | 1 |
| 박수진 | 치즈피자 | 1 |
| 박수진 | 감자튀김 | 1 |

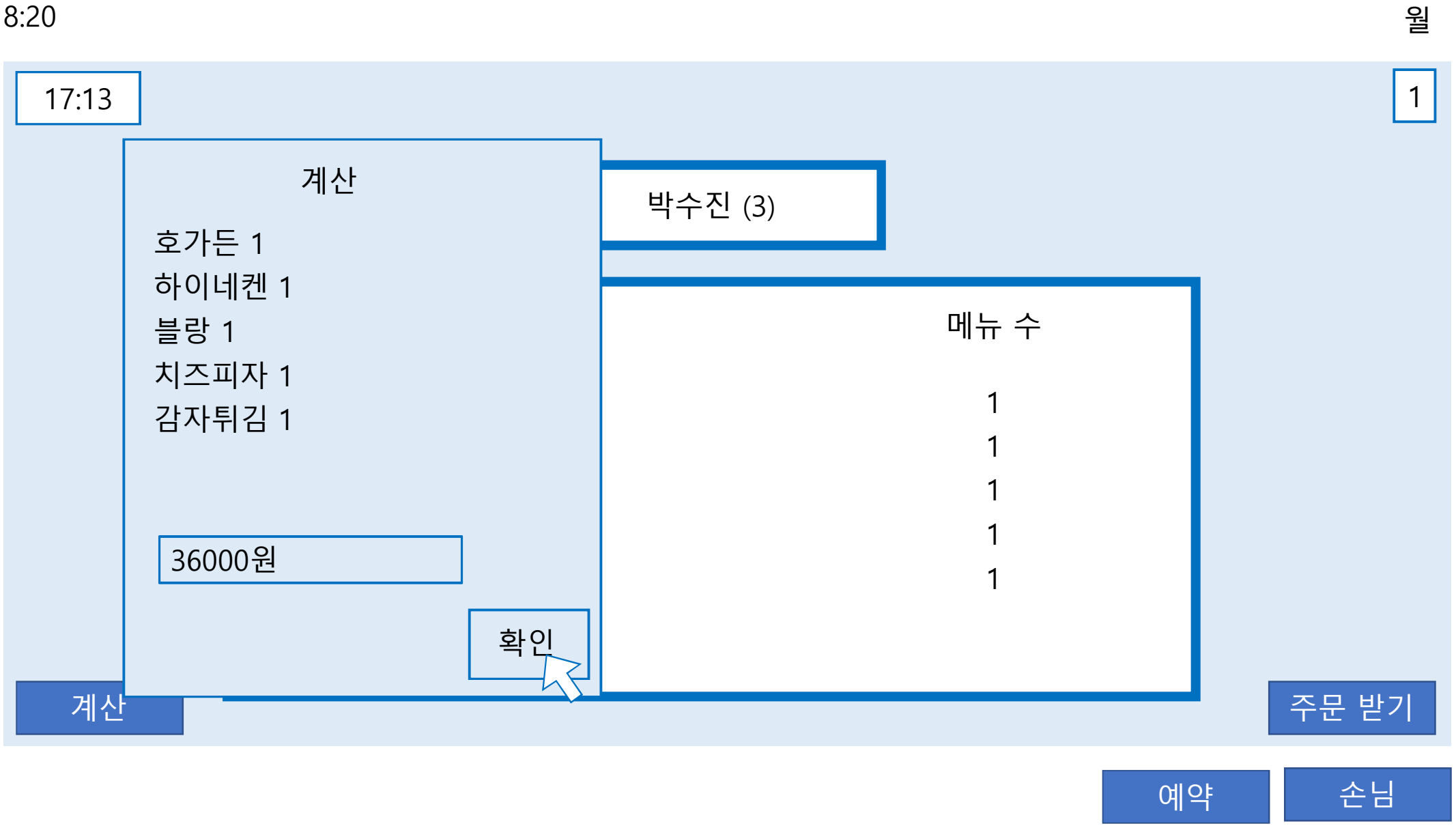
18:20 월


17:13
1
계산
호가든 1
하이네켄 1
블랑 1
치즈피자 1
감자튀김 1
36000원
확인
박수진 (3)
메뉴 수
1
1
1
1
1
계산
주문 받기
예약
손님

해당 손님이 나갈 때 SQL을 통해서 자동적으로 합계 금액이 산출
 -> 손님 번호는 해당 테이블에 저장되어 있는 손님 번호를 사용한다.

```
SELECT SUM(O.NUM*M.COST) AS ALL_COST FROM CUSTOMER C, MENU M, HOF_ORDER H, ORDER_LIST O
WHERE  C.CUS_NO = 1 AND H.ORDER_NO = O.ORDER_NO AND H.CUS_NO = C.CUS_NO AND M.MENU_NO = O.MENU_NO
```

| ALL_COST |
| --- |
| 36000 |


18:20
월
17:13
1
계산
호가든 1
하이네켄 1
블랑 1
치즈피자 1
감자튀김 1
36000원
확인
박수진 (3)
메뉴 수
1
1
1
1
1
계산
주문 받기
예약
손님

18:20

월

18:12

2

강유리 (1)

칭타오 1

밀러 1

치킨 1

예약

손님

<예약 전화>

"안녕하세요. With화연이죠? 예약을 하려고 하는데요."
"이름, 전화번호, 나이, 성별, 사시는 도시와 구를 말씀해주세요."
"최민지, 010-2323-3131, 23살, 여성, 서울시, 관악구요."
"감사합니다. 예약 하시려는 시간과 요일, 일행 수를 말씀해주세요."
"오늘(화요일), 7시 30분, 4명이요."
"화요일 7시 30분에 4명 알겠습니다. 감사합니다."

19:23 화

19:01 36

박복선 (3)

코로나 1
코젤다크 1
기네스 1
치즈피자 1

19:11 38

이루현 (4)

아사히 1
코로나 1
블랑 1
버드와이저 1
...

19:14 39

안건희 (3)

19:21 40

임주현 (3)

호가든 1
버드와이저 1
기린 이치방 1
페페로니피자 1
노가리 1

19:22

이아랑 (4)

블랑 1
스텔라 1
코젤다크 1
치킨 1
감자튀김 1

42

Name : 최민지

Tel : 010-2323-3131

Age : 23

Sex : 여성

City : 서울시

Ku : 관악구

예약 주문

예약한 손님의 손님 정보 입력

예약 손님

| CUS_NO | NAME | TEL | AGE | SEX | CITY | KU |
|---|---|---|---|---|---|---|
| 20 | 황세리 | 010-2234-4458 | 31 | 여성 | 서울시 | 서대문구 |
| 21 | 손건호 | 010-5561-5603 | 22 | 남성 | 서울시 | 송파구 |
| 22 | 한연준 | 010-6723-1277 | 24 | 여성 | 서울시 | 서대문구 |
| 23 | 이정현 | 010-4519-2954 | 22 | 여성 | 경기도 | 고양시 |
| 24 | 강모진 | 010-8117-1910 | 24 | 여성 | 서울시 | 서대문구 |
| 25 | 반지훈 | 010-9494-8282 | 48 | 남성 | 경기도 | 구리시 |
| 26 | 한지민 | 010-4256-5028 | 38 | 여성 | 서울시 | 종로구 |
| 27 | 강나라 | 010-2345-1211 | 24 | 여성 | 서울시 | 서대문구 |
| 28 | 김태희 | 010-7474-7474 | 23 | 여성 | 서울시 | 서대문구 |
| 29 | 남지호 | 010-6639-6352 | 22 | 여성 | 경기도 | 고양시 |
| 30 | 류현정 | 010-5743-4028 | 20 | 여성 | 서울시 | 종로구 |
| 31 | 김별이 | 010-0333-8999 | 22 | 여성 | 경기도 | 고양시 |
| 32 | 김류림 | 010-8543-8279 | 22 | 여성 | 서울시 | 강남구 |
| 33 | 이인기 | 010-8402-3749 | 34 | 여성 | 서울시 | 서대문구 |
| 34 | 유예지 | 010-9823-1237 | 21 | 여성 | 경기도 | 구리시 |
| 35 | 육성재 | 010-8472-7654 | 29 | 남성 | 서울시 | 중구 |
| 36 | 박복선 | 010-8613-1867 | 32 | 여성 | 경기도 | 광주시 |
| 37 | 윤동주 | 010-8399-2294 | 38 | 남성 | 서울시 | 서대문구 |
| 38 | 이루현 | 010-9451-5273 | 21 | 여성 | 서울시 | 서대문구 |
| 39 | 안건희 | 010-2869-2789 | 22 | 여성 | 서울시 | 관악구 |
| 40 | 임주현 | 010-1411-1319 | 23 | 여성 | 경기도 | 고양시 |
| 41 | 이아랑 | 010-4718-6666 | 20 | 여성 | 서울시 | 마포구 |
| 42 | 최민지 | 010-2323-3131 | 23 | 여성 | 서울시 | 관악구 |

예약한 손님의 손님 정보 (이름, 전화번호, 나이, 성별, 도시, 구)를 입력하면, CUSTOMER에 저장된다.

-> 손님 정보에 해당하는 번호로 손님 번호는 자동으로 저장

```
INSERT INTO CUSTOMER VALUES(42, '최민지',   '010-2323-3131',
                            23,    '여성', '서울시',   '관악구');
```

19:23

화

19:01 36

박복선 (3)

코로나 1
코젤다크 1
기네스 1
치즈피자 1

19:11 38

이루현 (4)

아사히 1
코로나 1
블랑 1
버드와이저 1
...

19:14 39

안건희 (3)

호가든 1

19:21 40

임주현 (3)

호가든 1
버드와이저 1
기린 이치방 1
페페로니피자 1
노가리 1

19:22

이아랑 (4)

블랑 1
스텔라 1
코젤다크 1
치킨 1
감자튀김 1

5

Name : 최민지

Tel : 010-2323-3131

Time : 19:30

Day : 화

People : 4

주문

예약한 손님의 예약 정보 입력

예약

손님

예약한 손님의 전화번호, 시간, 요일 일행 수를 입력한다.
나머지 정보는 자동으로 입력되어 RESERVATION에 저장된다.
 -> 예약 정보의 이름은 예약 정보에 있는 전화 번호와 동일한 손님 메뉴의 전화번호에서 손님의 이름을 저장한다.
 -> 예약 번호는 예약한 순서대로 자동으로 저장되며, 일주일마다 초기화된다.

```
INSERT INTO RESERVATION VALUES(5,   '최민지',   '010-2323-3131',    '19:30',    '화',   4);
```

| RES_NO | RES_NAME | RES_PHONE | RES_TIME | RES_DAY | NUM_OF_PEOPLE |
|---|---|---|---|---|---|
| 1 | 도솔미 | 010-2223-2357 | 19:00:00.0000000 | 월 | 3 |
| 2 | 김혜자 | 010-1597-3574 | 20:30:00.0000000 | 월 | 4 |
| 3 | 박민정 | 010-9879-0987 | 21:00:00.0000000 | 월 | 3 |
| 4 | 유예지 | 010-9823-1237 | 18:30:00.0000000 | 화 | 4 |
| 5 | 최민지 | 010-2323-3131 | 19:30:00.0000000 | 화 | 4 |

19:30

화

19:11 38

이루현 (4)

아사히 1
코로나 1
블랑 1
버드와이저 1
...

19:14 39

안건희 (3)

19:21 40

임주현 (3)

호가든 1
버드와이저 1
기린 이치방 1
페페로니피자 1
노가리 1

19:22

이아랑 (4)

블랑 1
스텔라 1
코젤다크 1
치킨 1
감자튀김 1

5

Name : 최민지

Tel : 010-2323-3131

Time : 19:30

Day : 화

People : 4

주문

예약한 손님이 예약한 시간에 주문

예약

손님

19:30 화

19:30

42

최민지 (4)

| 메뉴이름 | 메뉴 수 |
| --- | --- |

계산

주문 받기

예약

손님

19:30

화

19:11 38

이루현 (4)

아사히 1
코로나 1
블랑 1
버드와이저 1
...

19:14 39

안건희 (3)

호가든 1
코로나 1
블랑 1
버드와이저 1
...

5

19:21 40

임주현 (3)

호가든 1
버드와이저 1
기린 이치방 1
페페로니피자 1
노가리 1

19:22 41

이아랑 (4)

블랑 1
스텔라 1
코젤다크 1
치킨 1
감자튀김 1

19:30 42

최민지 (4)

밖으로 나갔을 때의 화면

예약

손님

모든 정보가 자동적으로 입력이 되어, HOF_ORDER에 저장이 된다.

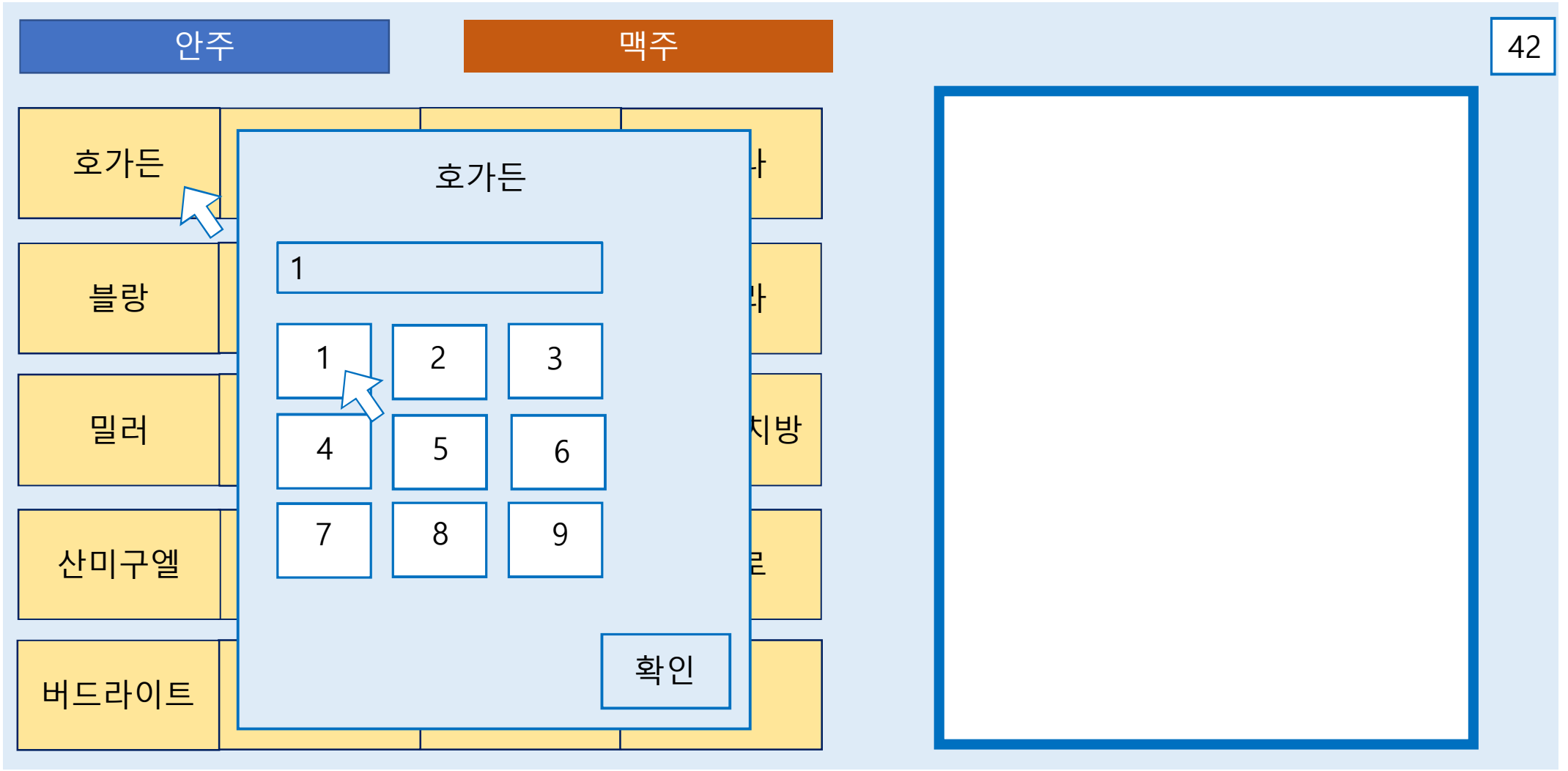
-> 주문 번호는 주문 버튼을 누른 시간 순서에 맞게 자동적으로 생성이 된다.
-> 손님 번호는 해당 주문 버튼이 있었던 손님 번호를 자동 저장한다.
-> 요일 정보는 애플리케이션 (혹은 기계)에 이미 저장되어 있는 정보를 저장한다.
-> 주문한 시간은 주문 버튼을 누른 시간이 자동적으로 저장이 된다.
-> 일행 수는 예약 정보에 저장되어 있는 정보가 자동으로 저장된다.

```
INSERT INTO HOF_ORDER VALUES(42,    42, '화',    '19:30', 4);
```

| ORDER_NO | CUS_NO | ORDER_DAY | ORDER_TIME | NUM_OF_PEOPLE |
|---|---|---|---|---|
| 25 | 25 | 월 | 22:13:00.0000000 | 4 |
| 26 | 26 | 월 | 22:54:00.0000000 | 2 |
| 27 | 27 | 월 | 23:18:00.0000000 | 1 |
| 28 | 28 | 월 | 23:34:00.0000000 | 2 |
| 29 | 29 | 화 | 17:17:00.0000000 | 1 |
| 30 | 30 | 화 | 17:52:00.0000000 | 2 |
| 31 | 31 | 화 | 18:01:00.0000000 | 3 |
| 32 | 32 | 화 | 18:13:00.0000000 | 3 |
| 33 | 33 | 화 | 18:21:00.0000000 | 2 |
| 34 | 34 | 화 | 18:30:00.0000000 | 4 |
| 35 | 35 | 화 | 18:31:00.0000000 | 1 |
| 36 | 36 | 화 | 19:01:00.0000000 | 3 |
| 37 | 37 | 화 | 19:01:00.0000000 | 2 |
| 38 | 38 | 화 | 19:11:00.0000000 | 4 |
| 39 | 39 | 화 | 19:14:00.0000000 | 3 |
| 40 | 40 | 화 | 19:21:00.0000000 | 3 |
| 41 | 41 | 화 | 19:22:00.0000000 | 4 |
| 42 | 42 | 화 | 19:30:00.0000000 | 4 |

19:30 화

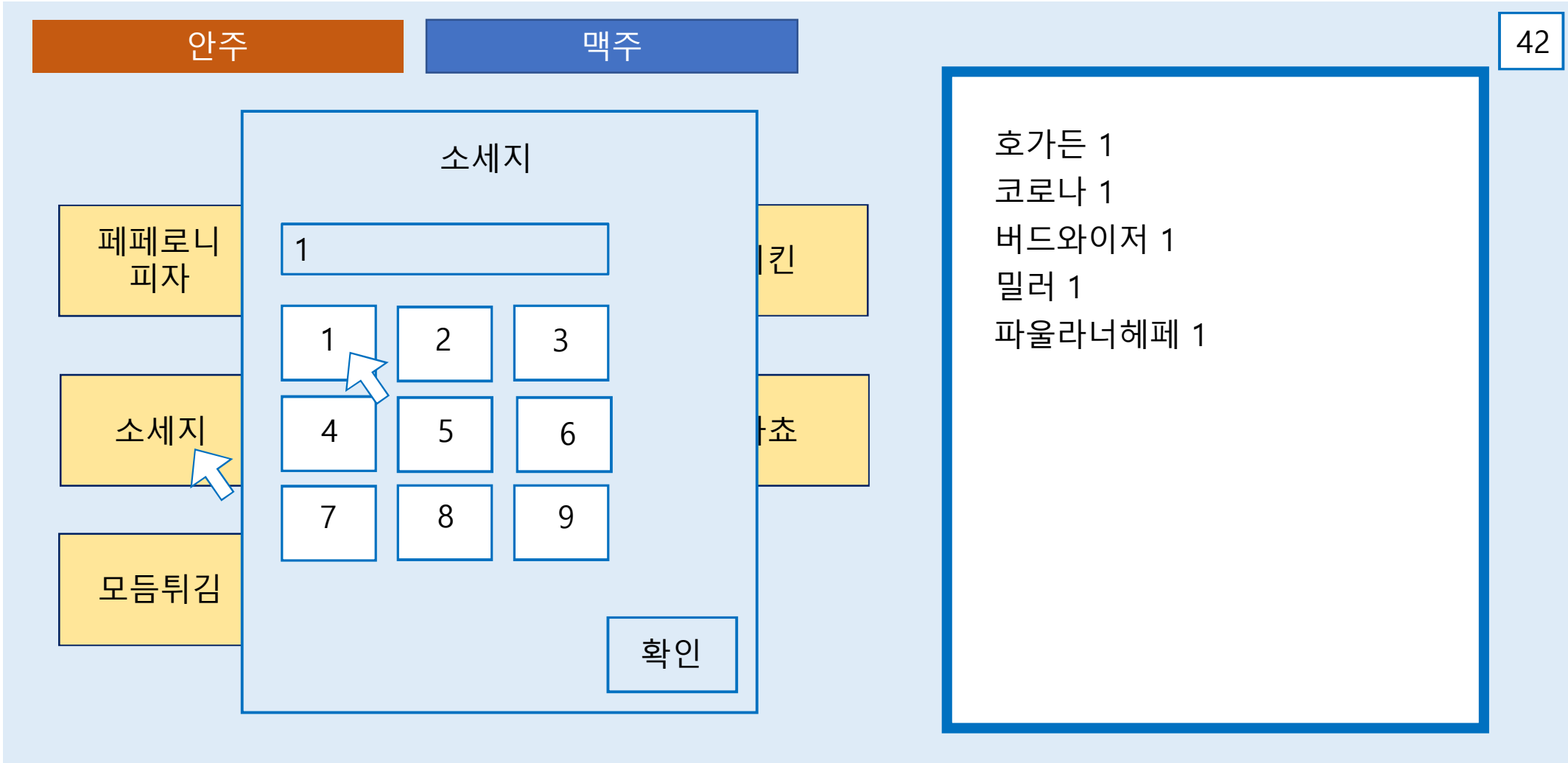
19:31 화




손님이 주문한 맥주와 수량을 입력

예약 손님

19:31 화




손님이 주문한 안주와 수량을 입력

예약 손님

19:32 화

안주 맥주

42

| | | | |
|---|---|---|---|
| 페페로니 피자 | 치즈피자 | 콤비네이션 피자 | 치킨 |
| 소세지 | 노가리 | 한치 | 나쵸 |
| 모듬튀김 | 감자튀김 | 마약옥수수 | |

호가든 1
코로나 1
버드와이저 1
밀러 1
파울라너헤페 1
소시지 1
모듬튀김 1
마약옥수수 1

예약 손님

손님이 주문한 메뉴를 하나씩 입력을 한다.
입력 값 하나하나가 ORDER_LIST에 저장된다.

-> 확인 버튼을 누를 때 마다, 하나씩 값이 저장이 된다.

```
INSERT INTO ORDER_LIST VALUES(42,  1,  1);
INSERT INTO ORDER_LIST VALUES(42,  4,  1);
INSERT INTO ORDER_LIST VALUES(42,  7,  1);
INSERT INTO ORDER_LIST VALUES(42,  9,  1);
INSERT INTO ORDER_LIST VALUES(42,  14, 1);
INSERT INTO ORDER_LIST VALUES(42,  25, 1);
INSERT INTO ORDER_LIST VALUES(42,  29, 1);
INSERT INTO ORDER_LIST VALUES(42,  31, 1);
```

| ORDER_NO | MENU_NO | NUM |
|---|---|---|
| 38 | 30 | 1 |
| 39 | 1 | 1 |
| 39 | 8 | 1 |
| 39 | 13 | 1 |
| 39 | 28 | 1 |
| 40 | 1 | 1 |
| 40 | 7 | 1 |
| 40 | 12 | 1 |
| 40 | 21 | 1 |
| 40 | 26 | 1 |
| 41 | 5 | 1 |
| 41 | 8 | 2 |
| 41 | 10 | 1 |
| 41 | 24 | 1 |
| 41 | 30 | 1 |
| 42 | 1 | 1 |
| 42 | 4 | 1 |
| 42 | 7 | 1 |
| 42 | 9 | 1 |
| 42 | 14 | 1 |
| 42 | 25 | 1 |
| 42 | 29 | 1 |
| 42 | 31 | 1 |

19:32 화

19:30

42

최민지 (4)

| 메뉴이름 | 메뉴 수 |
| --- | --- |
| 호가든 | 1 |
| 코로나 | 1 |
| 버드와이저 | 1 |
| 밀러 | 1 |
| 파울라너헤페 | 1 |
| 소시지 | 1 |
| 모듬튀김 | 1 |
| 마약옥수수 | 1 |

계산

주문 받기

예약

손님

19:32

화

19:11 38

이루현 (4)

아사히 1
코로나 1
블랑 1
버드와이저 1
...

19:14 39

안건희 (3)

호가든 1
코로나 1
블랑 1
버드와이저 1
...

5

19:21 40

임주현 (3)

호가든 1
버드와이저 1
기린 이치방 1
페페로니피자 1
노가리 1

19:22 41

이아랑 (4)

블랑 1
스텔라 1
코젤다크 1
치킨 1
감자튀김 1

19:30 42

최민지 (4)

호가든 1
코로나 1
버드와이저 1
밀러 1
...

밖으로 나갔을 때의 화면

예약

손님

해당 예약 손님이 최종적으로 시킨 메뉴와 수량

```
SELECT R.RES_NO, C.NAME, M.MENU_NAME, O.NUM FROM CUSTOMER C, MENU M, HOF_ORDER H, ORDER_LIST O, RESERVATION R
WHERE  R.RES_NO = 5  AND R.RES_PHONE = C.TEL AND H.ORDER_NO = O.ORDER_NO AND H.CUS_NO = C.CUS_NO AND M.MENU_NO = O.MENU_NO
```

| RES_NO | NAME | MENU_NAME | NUM |
|---|---|---|---|
| 5 | 최민지 | 호가든 | 1 |
| 5 | 최민지 | 코로나 | 1 |
| 5 | 최민지 | 버드와이저 | 1 |
| 5 | 최민지 | 밀러 | 1 |
| 5 | 최민지 | 파울라너헤페 | 1 |
| 5 | 최민지 | 소세지 | 1 |
| 5 | 최민지 | 모듬튀김 | 1 |
| 5 | 최민지 | 마약옥수수 | 1 |

21:20 화

19:30

42

계산

호가든 1
코로나 1
버드와이저 1
밀러 1
파울라너헤페 1
소시지 1
모듬튀김 1
마약옥수수 1

62000원

확인

최민지 (4)

| 메뉴 | 수 |
| --- | --- |
| | 1 |
| | 1 |
| | 1 |
| | 1 |
| | 1 |
| | 1 |
| | 1 |
| | 1 |

계산

주문 받기

예약

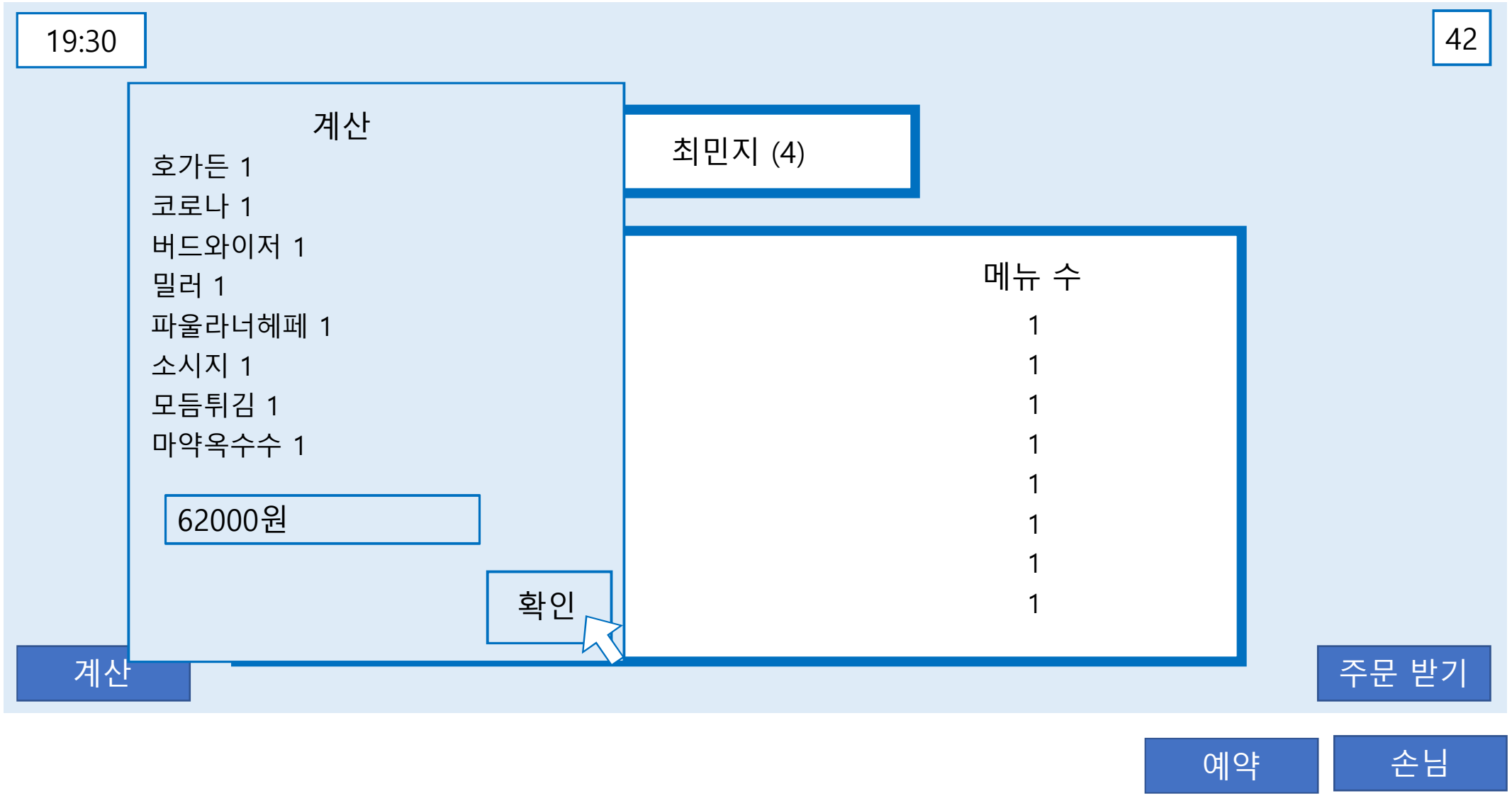
손님

해당 손님이 나갈 때 SQL을 통해서 자동적으로 합계 금액이 산출
 -> 손님 번호는 해당 테이블에 저장되어 있는 손님 번호를 사용한다.

```
SELECT SUM(O.NUM*M.COST) AS ALL_COST FROM CUSTOMER C, MENU M, HOF_ORDER H, ORDER_LIST O
WHERE  C.CUS_NO = 42 AND H.ORDER_NO = O.ORDER_NO AND H.CUS_NO = C.CUS_NO AND M.MENU_NO = O.MENU_NO
```

| ALL_COST |
| --- |
| 62000 |

21:20 화


19:30
42
계산
호가든 1
코로나 1
버드와이저 1
밀러 1
파울라너헤페 1
소시지 1
모듬튀김 1
마약옥수수 1
62000원
확인
최민지 (4)
메뉴 수
1
1
1
1
1
1
1
1
계산
주문 받기

예약 손님

21:20

화

21:02 52

김영은 (3)

아사히 3
칭타오 1
스텔라 1
치킨 1
나쵸 1

20:13 48

신애란 (4)

호가든 1
칭타오 1
코젤다크 1
기네스 1
...

21:19 53

정우성 (3)

아사히 2
삿포로 1
한치 1
모듬튀김 1

20:39 50

이기용 (4)

기네스 1
에딩거헤페 2
스콜 2
치킨 1
...

21:00 51

정미연 (4)

칭타오 1
코젤다크 1
에딩거헤페 1
페페로니피자 1
...

예약

손님

# 매출 분석 리포트

유혜지, 안해인, 유현선, 이재용

일주일동안 수집한 고객, 주문 정보 데이터를 토대로 With화연의 손님과 매출을 분석해보았다.

## 1. 전체 손님 수와 전체 매출

먼저 일주일간 With화연을 방문한 총 손님 수와 매출을 계산하여 분석해보았다.

### 1) 전체 손님 수

```
SELECT SUM(O.NUM_OF_PEOPLE) AS SUM_OF_CUSTOMER FROM HOF_ORDER O
```

결과 메시지

| | SUM_OF_CUSTOMER |
|---|---|
| 1 | 610 |

### 2) 전체 매출

```
SELECT SUM(L.NUM * M.COST) AS DAY_SALES FROM ORDER_LIST L, HOF_ORDER O, MENU M
 WHERE O.ORDER_NO = L.ORDER_NO
         AND L.MENU_NO = M.MENU_NO;
```

결과 메시지

| | DAY_SALES |
|---|---|
| 1 | 9536000 |

### 3) 결과 분석

서울시 골목상권 분석 서비스에 따른 2018년 서대문구 내 전체 호프집의 일주일 매출 평균은 5,289,737원이다. 그에 비해 With화연은 평균보다 거의 2배에 가까운 매출을 기록하였다.

## 2. 요일별 매출 및 손님 수

일주일간 전체 매출과 손님 수를 통해 With화연이 서대문구의 타 호프집들보다 매출이 높다는 것을 알 수 있었다. 그래서 이번엔 With화연의 각 요일별 매출을 비교하여 어떤 날이 장사가 제일 잘 되는지 알아보았다.

### 1) 요일별 매출

- 월~일 각 요일별 매출

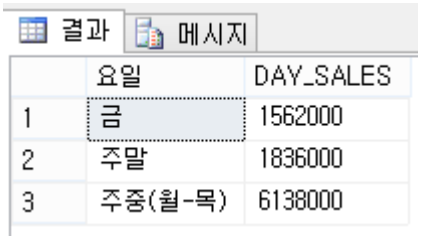
```
SELECT O.ORDER_DAY, SUM(L.NUM * M.COST) AS DAY_SALES FROM ORDER_LIST L, HOF_ORDER O, MENU M
WHERE O.ORDER_NO = L.ORDER_NO
        AND L.MENU_NO = M.MENU_NO
GROUP BY O.ORDER_DAY
ORDER BY DAY_SALES DESC;
```

결과 메시지

| | ORDER_DAY | DAY_SALES |
|---|---|---|
| 1 | 목 | 1666000 |
| 2 | 화 | 1625000 |
| 3 | 수 | 1602000 |
| 4 | 금 | 1562000 |
| 5 | 토 | 1322000 |
| 6 | 월 | 1245000 |
| 7 | 일 | 514000 |

- 주중(월-목), 금요일, 주말별 매출

```
SELECT (case when O.ORDER_DAY = '토' OR O.ORDER_DAY = '일'
             then '주말'
             when O.ORDER_DAY = '금'
             then '금'
             else '주중(월-목)'
        end) 요일,
        SUM(L.NUM * M.COST) AS DAY_SALES FROM ORDER_LIST L, HOF_ORDER O, MENU M
WHERE O.ORDER_NO = L.ORDER_NO
        AND L.MENU_NO = M.MENU_NO
GROUP BY (case when O.ORDER_DAY = '토' OR O.ORDER_DAY = '일'
             then '주말'
             when O.ORDER_DAY = '금'
             then '금'
             else '주중(월-목)'
          end);
```

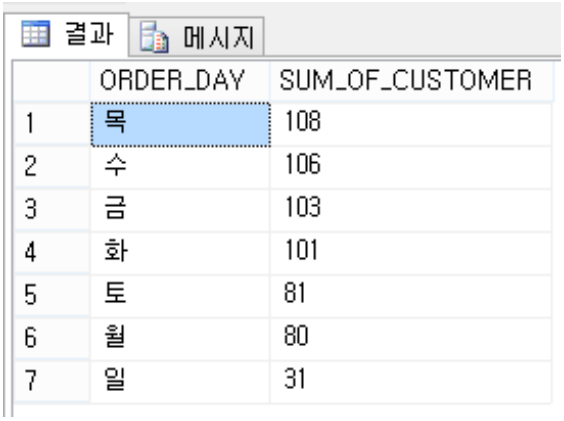


결과 메시지

| | 요일 | DAY_SALES |
|---|---|---|
| 1 | 금 | 1562000 |
| 2 | 주말 | 1836000 |
| 3 | 주중(월-목) | 6138000 |

2) 요일별 손님 수

```
SELECT O.ORDER_DAY, SUM(O.NUM_OF_PEOPLE) AS SUM_OF_CUSTOMER FROM HOF_ORDER O
GROUP BY O.ORDER_DAY
ORDER BY SUM_OF_CUSTOMER DESC;
```



결과 메시지

| | ORDER_DAY | SUM_OF_CUSTOMER |
|---|---|---|
| 1 | 목 | 108 |
| 2 | 수 | 106 |
| 3 | 금 | 103 |
| 4 | 화 | 101 |
| 5 | 토 | 81 |
| 6 | 월 | 80 |
| 7 | 일 | 31 |

3) 결과 분석

각 요일별 매출과 손님 수를 비교해본 결과, 목요일이 손님 수도 제일 많았고 매출도 가장 높았다. 또한 예상과 달리, 주중 매출에 비해 주말 매출이 저조했다. 서대문구의 주중, 금요일, 주말 매출 평균은 각각 2,731,326원, 875,399원, 1,683,011원이다. 이 수치와 With화연의 매출을 비교해보니 모두 서대문구의 평균보다 훨씬 높았다.

## 3. 메뉴별 판매량 비교

With화연에서는 무엇이 제일 잘 팔리는지 알아보고자 메뉴별 판매량을 분석해보았다.

1) 총 맥주, 안주 판매량

```
SELECT M.MENU_TYPE, SUM(L.NUM) AS SALES_NUM FROM ORDER_LIST L, HOF_ORDER O, MENU M
WHERE O.ORDER_NO = L.ORDER_NO
        AND L.MENU_NO = M.MENU_NO
GROUP BY M.MENU_TYPE
ORDER BY SALES_NUM;
```

결과 | 메시지

| | MENU_TYPE | SALES_NUM |
|---|---|---|
| 1 | 안주 | 503 |
| 2 | 맥주 | 788 |

2) 맥주, 안주 종류별 판매량

- 맥주 종류별 판매량

```
SELECT O.ORDER_DAY, SUM(L.NUM * M.COST) AS DAY_SALES FROM ORDER_LIST L, HOF_ORDER O, MENU M
WHERE O.ORDER_NO = L.ORDER_NO
        AND L.MENU_NO = M.MENU_NO
GROUP BY O.ORDER_DAY
ORDER BY DAY_SALES DESC;
```

결과 | 메시지

| | MENU_NAME | SALES_NUM |
|---|---|---|
| 1 | 호가든 | 86 |
| 2 | 아사히 | 67 |
| 3 | 칭타오 | 65 |
| 4 | 코젤다크 | 64 |
| 5 | 블랑 | 62 |
| 6 | 스텔라 | 48 |
| 7 | 기네스 | 46 |
| 8 | 버드와이저 | 43 |
| 9 | 기린 이치방 | 41 |
| 10 | 하이네켄 | 38 |
| 11 | 코로나 | 37 |
| 12 | 산미구엘 | 29 |
| 13 | 삿포로 | 29 |
| 14 | 파울라너헤페 | 28 |
| 15 | 밀러 | 25 |
| 16 | 에딩거헤페 | 19 |
| 17 | 카프리 | 17 |
| 18 | 백스 | 15 |
| 19 | 버드라이트 | 15 |
| 20 | 스콜 | 14 |

- 안주 종류별 판매량

```
SELECT M.MENU_NAME, SUM(L.NUM) AS SALES_NUM FROM ORDER_LIST L, HOF_ORDER O, MENU M
WHERE O.ORDER_NO = L.ORDER_NO
        AND L.MENU_NO = M.MENU_NO
        AND M.MENU_TYPE = '안주'
GROUP BY M.MENU_NAME
ORDER BY SALES_NUM DESC;
```

결과 메시지

| | MENU_NAME | SALES_NUM |
|---|---|---|
| 1 | 나쵸 | 71 |
| 2 | 치킨 | 71 |
| 3 | 감자튀김 | 65 |
| 4 | 마약옥수수 | 45 |
| 5 | 모듬튀김 | 42 |
| 6 | 소세지 | 42 |
| 7 | 페페로니피자 | 40 |
| 8 | 노가리 | 37 |
| 9 | 치즈피자 | 35 |
| 10 | 한치 | 32 |
| 11 | 콤비네이션피자 | 23 |

3) 결과 분석

일주일간 맥주는 총 788개, 안주는 503개가 팔리는 것으로 보아, 평균적으로 손님들은 안주 1개당 1.6개의 맥주를 주문한다고 추측할 수 있다. 맥주의 경우 호가든이 2위인 아사히와 큰 차이로 제일 많이 팔렸으며, 칭타오, 코젤다크, 블랑이 그 뒤를 이었다. 또한 안주는 나쵸와 치킨이 가장 많이 팔렸고, 감자튀김, 마약옥수수, 모듬튀김, 소세지가 그 다음으로 많이 팔렸다.

## 4. 손님 특성 분석

With화연이 지금보다 더 높은 매출을 기록하기 위해서는 방문하는 손님들의 특성을 분석할 필요가 있다. 그래서 일주일간 With화연에 방문한 손님들의 데이터를 토대로 손님 특성을 분석해보았다. 분석에 사용된 손님 데이터는 주문을 진행한 대표 손님들만 저장된 데이터이다.

1) 손님 성별 비율

```
SELECT C.SEX, COUNT(C.SEX) AS NUM FROM CUSTOMER C
GROUP BY C.SEX
ORDER BY NUM DESC;
```

결과 메시지

| | SEX | NUM |
|---|---|---|
| 1 | 여성 | 153 |
| 2 | 남성 | 45 |

2) 지역별 손님 수

- 도/시별 손님 수

```
SELECT C.CITY, COUNT(C.CITY) AS NUM FROM HOF_ORDER O, CUSTOMER C
WHERE O.CUS_NO = C.CUS_NO
GROUP BY C.CITY
ORDER BY NUM DESC;
```

결과 메시지

| | CITY | NUM |
|---|---|---|
| 1 | 서울시 | 171 |
| 2 | 경기도 | 38 |

- 구별 손님 수 (서울시 상위 5개구)

```
SELECT TOP(5) C.KU, COUNT(C.KU) AS NUM FROM HOF_ORDER O, CUSTOMER C
WHERE O.CUS_NO = C.CUS_NO
        AND C.CITY = '서울시'
GROUP BY C.KU
ORDER BY NUM DESC;
```

결과 메시지

| | KU | NUM |
|---|---|---|
| 1 | 서대문구 | 55 |
| 2 | 마포구 | 28 |
| 3 | 강남구 | 14 |
| 4 | 관악구 | 9 |
| 5 | 중구 | 6 |

- 시별 손님 수 (경기도 상위 5개 시)

```
SELECT TOP(5) C.KU, COUNT(C.KU) AS NUM FROM HOF_ORDER O, CUSTOMER C
WHERE O.CUS_NO = C.CUS_NO
        AND C.CITY = '경기도'
GROUP BY C.KU
ORDER BY NUM DESC;
```

결과 메시지

| | KU | NUM |
|---|---|---|
| 1 | 고양시 | 13 |
| 2 | 인천시 | 6 |
| 3 | 구리시 | 4 |
| 4 | 김포시 | 4 |
| 5 | 성남시 | 2 |

3) 나이대별 손님 수

```
SELECT (case when C.AGE >= 20 AND AGE< 30
             then '20대'
             when C.AGE >= 30 and C.AGE < 40
             then '30대'
             else '40대 이상'
        end) 나이대,
       count(*) AS 손님수
FROM CUSTOMER C, HOF_ORDER O
WHERE C.CUS_NO = O.CUS_NO
GROUP BY (case when C.AGE >= 20 AND AGE< 30
             then '20대'
             when C.AGE >= 30 and C.AGE < 40
             then '30대'
             else '40대 이상'
        end);
```

결과 메시지

| | 나이대 | 손님수 |
|---|---|---|
| 1 | 20대 | 173 |
| 2 | 30대 | 21 |
| 3 | 40대 이상 | 15 |

4) 시간대별 손님 수

```
SELECT (case when O.ORDER_TIME >= '17:00:00' AND O.ORDER_TIME < '19:00:00'
             then '17시 ~ 19시'
             when O.ORDER_TIME >= '19:00:00' AND O.ORDER_TIME < '21:00:00'
             then '19시 ~ 21시'
             when O.ORDER_TIME >= '21:00:00' AND O.ORDER_TIME < '23:00:00'
             then '21시 ~ 23시'
             else '23시 이후'
        end) 시간대,
       count(*) AS 손님수
FROM HOF_ORDER O
GROUP BY (case when O.ORDER_TIME >= '17:00:00' AND O.ORDER_TIME < '19:00:00'
             then '17시 ~ 19시'
             when O.ORDER_TIME >= '19:00:00' AND O.ORDER_TIME < '21:00:00'
             then '19시 ~ 21시'
             when O.ORDER_TIME >= '21:00:00' AND O.ORDER_TIME < '23:00:00'
             then '21시 ~ 23시'
             else '23시 이후'
        end);
```

결과 메시지

| | 시간대 | 손님수 |
|---|---|---|
| 1 | 17시 ~ 19시 | 34 |
| 2 | 19시 ~ 21시 | 95 |
| 3 | 21시 ~ 23시 | 63 |
| 4 | 23시 이후 | 17 |

5) 재방문한 손님

```
SELECT C.NAME, C.AGE, C.SEX, COUNT(O.CUS_NO) AS 재방문_횟수
FROM HOF_ORDER O, CUSTOMER C
WHERE O.CUS_NO = C.CUS_NO
GROUP BY C.NAME, C.AGE, C.SEX
HAVING COUNT(O.CUS_NO) > 1
ORDER BY C.AGE;
```

결과 | 메시지

| | NAME | AGE | SEX | 재방문_횟수 |
|---|---|---|---|---|
| 1 | 김아연 | 20 | 여성 | 2 |
| 2 | 유채원 | 20 | 여성 | 2 |
| 3 | 사은지 | 21 | 여성 | 2 |
| 4 | 이루현 | 21 | 여성 | 2 |
| 5 | 장보민 | 21 | 여성 | 2 |
| 6 | 화예희 | 23 | 여성 | 2 |
| 7 | 강유리 | 24 | 여성 | 2 |
| 8 | 한예리 | 28 | 여성 | 2 |
| 9 | 한지민 | 38 | 여성 | 2 |
| 10 | 주호성 | 39 | 남성 | 2 |
| 11 | 김혜자 | 62 | 여성 | 2 |

6) 결과 분석

방문하는 손님의 성별은 여성이 압도적으로 많았다. 또한 지역별 손님 수를 비교해본 결과 서울시 서대문구에서 거주하시는 손님이 가장 많았고, 손님의 나이대는 20대가 압도적으로 많았다. 또한 손님 대부분이 오후 7시에서 11시 사이에 With화연을 방문하였는데, 서대문구 내 대학생들의 하교 시간이 대부분 6시 이후라는 것을 감안해보면 With화연을 방문하는 손님들은 대부분 하루 일과가 끝난 대학생들이라는 것을 알 수 있다. 일주일 내에 재방문하는 손님들 역시 대부분 20대 여성이었다.

따라서 위 결과를 종합하면, With화연의 주고객은 서대문구에 거주하거나 서대문구에서 대학을 다니는 20대 여성임을 알 수 있다.

## 5. 주고객이 많이 주문하는 메뉴 분석

With화연의 주고객이 20대 여성이므로, 20대 여성 손님이 많이 주문하는 메뉴들을 분석해보았다. 또한 아직 주고객은 아니지만, 20대 남성을 공략하여 매출을 더 증대시킬 수 있는 방안을 고안하기 위해 20대 남성 손님이 많이 주문하는 메뉴들도 분석해보았다.

1) 20대 여성이 가장 많이 주문하는 메뉴 (상위 5가지)

- 맥주

```
SELECT TOP(5) M.MENU_NAME, SUM(L.NUM) AS SALES_NUM FROM ORDER_LIST L, HOF_ORDER O, MENU M, CUSTOMER C
WHERE O.ORDER_NO = L.ORDER_NO
        AND L.MENU_NO = M.MENU_NO
        AND C.CUS_NO = O.CUS_NO
        AND C.AGE >= 20 AND C.AGE < 30
        AND C.SEX = '여성'
        AND M.MENU_TYPE = '맥주'
GROUP BY M.MENU_NAME
ORDER BY SALES_NUM DESC;
```

결과 메시지

| | MENU_NAME | SALES_NUM |
|---|---|---|
| 1 | 호가든 | 68 |
| 2 | 블랑 | 55 |
| 3 | 아사히 | 48 |
| 4 | 코젤다크 | 45 |
| 5 | 스텔라 | 35 |

- 안주

```
SELECT TOP(5) M.MENU_NAME, SUM(L.NUM) AS SALES_NUM FROM ORDER_LIST L, HOF_ORDER O, MENU M, CUSTOMER C
WHERE O.ORDER_NO = L.ORDER_NO
        AND L.MENU_NO = M.MENU_NO
        AND C.CUS_NO = O.CUS_NO
        AND C.AGE >= 20 AND C.AGE < 30
        AND C.SEX = '여성'
        AND M.MENU_TYPE = '안주'
GROUP BY M.MENU_NAME
ORDER BY SALES_NUM DESC;
```

결과 메시지

| | MENU_NAME | SALES_NUM |
|---|---|---|
| 1 | 나쵸 | 57 |
| 2 | 감자튀김 | 51 |
| 3 | 치킨 | 51 |
| 4 | 마약옥수수 | 31 |
| 5 | 소세지 | 27 |

2) 20대 남성이 가장 많이 주문하는 메뉴 (상위 5가지)

- 맥주

```
SELECT TOP(5) M.MENU_NAME, SUM(L.NUM) AS SALES_NUM FROM ORDER_LIST L, HOF_ORDER O, MENU M, CUSTOMER C
WHERE O.ORDER_NO = L.ORDER_NO
        AND L.MENU_NO = M.MENU_NO
        AND C.CUS_NO = O.CUS_NO
        AND C.AGE >= 20 AND C.AGE < 30
        AND C.SEX = '남성'
        AND M.MENU_TYPE = '맥주'
GROUP BY M.MENU_NAME
ORDER BY SALES_NUM DESC;
```

결과 메시지

| | MENU_NAME | SALES_NUM |
|---|---|---|
| 1 | 칭타오 | 21 |
| 2 | 호가든 | 10 |
| 3 | 산미구엘 | 8 |
| 4 | 아사히 | 8 |
| 5 | 기네스 | 7 |

- 안주

```
SELECT TOP(5) M.MENU_NAME, SUM(L.NUM) AS SALES_NUM FROM ORDER_LIST L, HOF_ORDER O, MENU M, CUSTOMER C
WHERE O.ORDER_NO = L.ORDER_NO
        AND L.MENU_NO = M.MENU_NO
        AND C.CUS_NO = O.CUS_NO
        AND C.AGE >= 20 AND C.AGE < 30
        AND C.SEX = '남성'
        AND M.MENU_TYPE = '안주'
GROUP BY M.MENU_NAME
ORDER BY SALES_NUM DESC;
```

결과 메시지

| | MENU_NAME | SALES_NUM |
|---|---|---|
| 1 | 모듬튀김 | 11 |
| 2 | 치킨 | 11 |
| 3 | 마약옥수수 | 10 |
| 4 | 페페로니피자 | 10 |
| 5 | 감자튀김 | 9 |

3) 결과 분석

With화연의 주고객인 20대 여성 손님들은 주류로 호가든과 블랑을 가장 많이 주문했고, 안주류로는 나쵸와 감자튀김, 치킨을 많이 주문했다. 또한 With화연을 방문했던 20대 남성 손님들은 칭타오를 가장 많이 주문했고, 안주류로 모듬튀김과 치킨, 마약옥수수와 피자를 많이 주문했다.

## 6. 종합 결과 및 매출 증대 방안

전체 매출 및 판매량을 분석한 결과, 서대문구 내에서 With화연은 압도적으로 많은 매출을 기록하였다. 특히 주중 저녁 7시에서 11시 사이의 판매량이 가장 높았으며, 고객층 대부분이 근처 대학을 다니는 20대 여성이었다.

따라서 With화연이 매출을 더 높이기 위해서는 20대 여성 고객이 많이 주문하는 메뉴들을 더 특화할 필요가 있다. 20대 여성 고객이 맥주로 호가든을 가장 많이 시킨다는 점을 고려하여, 호가든 로제, 호가든 그랑 크루, 포비든 프룻 등 기본 호가든 외에 더 다양한 종류의 호가든을 판매하고 대신 잘 팔리지 않는 백스, 버드라이트, 스콜을 메뉴에서 제외시킨다면 전보다 매출을 더 높일 수 있을 것이다. 또한 20대 여성 손님들이 안주로 나쵸, 감자튀김, 치킨을 많이 주문하므로, 이 세가지 메뉴의 맛에 집중하는 한편 각각의 안주에 잘 어울리면서 판매량이 살짝 저조한 맥주를 안주와 함께 세트로 할인된 가격에 판매한다면 전체 매출을 높이는 동시에 호가든 외의 다른 맥주들의 판매율도 높이는 일석이조의 효과를 볼 수 있을 것으로 기대한다.

주고객에만 집중하지 않고 더 넓은 고객층을 확보하는 것도 매출을 올리는데 도움이 될 것이다. 현재의 With화연은 고객층이 주변 대학을 다니는 20대 여성에게 집중되어 있으므로 대학 방학 기간이거나 공휴일에는 매출이 평소보다 급감할 것이다. 요일별 매출만 보더라도 평일에 비해 주말의 매출이 평소보다 낮은 것으로 보아 방학 기간에는 주중에도 매출이 낮을 것이라 예상할 수 있다. 따라서 20대 여성 고객 외에도 다른 고객층을 확보할 필요가 있는데, 대학가의 특성상 20대 유동 인구가 많으므로, 단기적으로 20대 남성 고객층을 확보하는 것부터 시작하는게 좋을 것이라 판단된다. 20대 남성 고객을 끌어들이기 위해서는 20대 남성 고객이 가장 많이 주문하는 치킨, 모듬튀김과 칭타오를 중점적으로 판매하는 것이 좋다. 특히 치킨의 경우 20대 여성 고객들도 많이 주문하는 메뉴이기 때문에, 치킨을 좀 더 개발하여 종류를 다양화하거나 맛을 극대화한다면 매출이 더 증가할 것이다. 그리고 20대 남성 고객을 위해 맛과 가성비를 고루 갖춘 치맥 세트를 판매한다면 20대 남성 고객 확보에 도움이 될 것이다.