

데이터베이스 팀 프로젝트 보고서

1602069 예지혜

1615081 김성희

1617001 구재원

1717009 김지우

<목차>

- 사용 공공 데이터 정의서
- 공공 데이터 분석 SQL 및 실행 결과 분석 보고서
- 요구사항 정의서
- ERD
- 관계 데이터베이스 스키마
- 정규화된 스키마
- 테스트 데이터 적재 방법과 SQL 및 실행 결과
- 주문 관리 프로그램 SQL 및 실행 결과
- SQL 분석을 통한 분석 리포트

사용 공공 데이터 정의서

1. 서대문구의 연령 및 성별 인구

- 서대문구 지역의 연령 및 성별 인구
- 시점 : 2015년~2017년 총 3개 시점
- 항목 : 연령 - '0~4세' 항목부터 '100세 이상'까지 21개 범주
성별 - 남성, 여성 2개 범주
- 출처 : 통계청 「인구총조사」 중 연령 및 성별 인구

http://kosis.kr/statHtml/statHtml.do?orgId=101&tblId=DT_1IN1503&conn_path=I2

2. 서대문구 상권분석서비스

- 서대문구 업종별 상권 분석
- 시점 : 2018년 10월
- 항목 : 1~2년 생존률, 2~3년 생존률, 여성_매출_금액, 여성_매출_건수, 연령대_20_매출_금액, 연령대_20_매출_건수, 점포수 이용
상권코드 - 연세로 4길
서비스업종코드 - CS100001 ~ CS100010
- 출처 : 서울 열린데이터 광장 「서울시 우리마을가게 상권분석서비스」

<https://data.seoul.go.kr/dataList/datasetView.do?infd=OA-13331&srvType=S&serviceKind=1¤tPageNo=1&searchValue=&searchKey=#AXexec>

공공 데이터 분석 SQL 및 실행 결과 분석 보고서

1. 세대문구 연령 및 성별 인구 데이터 분석

1) 최대인구 연령별, 20대 남녀성비 (2015)

```
SELECT 연령별, [총인구_남자(명)] AS '2015남자', [총인구_여자(명)] AS '2015여자'
FROM dbo.데이터$
WHERE 연령별 IN ('20~24세', '25~29세')

select 연령별, max([총인구(명)]) as '2015최대인구'
FROM dbo.데이터$
group by 연령별
having max([총인구(명)]) >= ALL (select max([총인구(명)]) from dbo.데이터$ group by 연령별)
```

100 %

결과 메시지

	연령별	2015남자	2015여자
1	20~24세	13193	16427
2	25~29세	12908	13095

	연령별	2015최대인구
1	20~24세	29620

2) 최대인구 연령별, 20대 남녀성비 (2016)

```
select 연령별, max([총인구(명)]) as '2016최대인구'
FROM dbo.데이터$
group by 연령별
having max([총인구(명)]) >= ALL (select max([총인구(명)]) from dbo.데이터$ group by 연령별)

SELECT 연령별, [총인구_남자(명)] AS '2016남자', [총인구_여자(명)] AS '2016여자'
FROM dbo.데이터$
WHERE 연령별 IN ('20~24세', '25~29세')
```

100 %

결과 메시지

	연령별	2016최대인구
1	20~24세	32533

	연령별	2015남자	2015여자
1	20~24세	13650	18883
2	25~29세	13295	14107

3) 최대인구 연령별, 20대 남녀성비 (2017)

```
select 연령별, max( [총인구(명)] ) as '2017최대인구'
FROM dbo.데이터$
group by 연령별
having max( [총인구(명)] ) >= ALL (select max( [총인구(명)] ) from dbo.데이터$ group by 연령별)
```

```
SELECT 연령별, [총인구_남자(명)] AS '2017남자', [총인구_여자(명)] AS '2017여자'
FROM dbo.데이터$
WHERE 연령별 IN ( '20~24세', '25~29세' )
```

	연령별	2017최대인구
1	20~24세	33583

	연령별	2017남자	2017여자
1	20~24세	13410	20173
2	25~29세	13759	14485

2. 서대문구 상권 분석

1) 서대문구 상권 생존율 분석 (1~2년 , 2~3년)

```
SELECT 서비스_업종_코드_명, [1~2년_생존_율], [2~3년_생존_율]
FROM [dbo].[Excel_Import_1$]
WHERE 상권_코드_명 = '연세로4길'
AND 서비스_업종_코드 BETWEEN 'CS100001' AND 'CS10009'
ORDER BY [1~2년_생존_율] DESC
```

	서비스_업종_코드_명	1~2년_생존_율	2~3년_생존_율
1	패스트푸드점	83,33	50
2	호프간이주점	70	60
3	일식집	66,67	66,67
4	분식집	53,33	43,33
5	제과점	50	50
6	치킨집	50	40
7	양식집	47,37	42,11
8	한식음식점	43,4	33,96

2) 업종별 20대 매출 금액 내림차순 정렬

```

SELECT 서비스_업종_코드_명, [연령대_20_매출_금액]
FROM [dbo].[Excel_Import_1$]
WHERE 상권_코드_명 = '연세로4길'
AND 서비스_업종_코드 BETWEEN 'CS100001' AND 'CS10009'
ORDER BY [연령대_20_매출_금액] DESC

```

	서비스_업종_코드_명	연령대_20_매출_금액
1	분식집	967095113
2	패스트푸드점	868476495
3	중국집	668789631
4	한식음식점	6258532774
5	치킨집	455740743
6	제과점	407046642
7	커피음료	1561549622
8	양식집	1548979087

3) 점포수 대비 20대여성매출 금액 내림차순 정렬

```

SELECT 서비스_업종_코드_명, max(CONVERT(BIGINT,여성_매출_금액)/ CONVERT(BIGINT,점포수)) AS '점포수대비여성매출_20대'
FROM [dbo].[Excel_Import_1$]
WHERE 상권_코드_명 = '연세로4길'
AND 서비스_업종_코드 BETWEEN 'CS100001' AND 'CS10009'
group by 서비스_업종_코드_명
order by '점포수대비여성매출_20대' desc

```

	서비스_업종_코드_명	점포수대비여성매출_20대
1	패스트푸드점	22277215
2	제과점	18483910
3	중국집	15671367
4	커피음료	13022608
5	일식집	11868305
6	치킨집	11553438
7	한식음식점	11498301
8	양식집	9775259

4) 점포수 대비 20대여성매출 건수 내림차순 정렬

```

SELECT 서비스_업종_코드_명, max(CONVERT(BIGINT,여성_매출_건수)/ CONVERT(BIGINT,점포수)) AS '점포수대비건수_20대'
FROM [dbo].[Excel_Import_1$]
WHERE 상권_코드_명 = '연세로4길'
AND 서비스_업종_코드 BETWEEN 'CS100001' AND 'CS10009'
group by 서비스_업종_코드_명
order by '점포수대비건수_20대' desc

```

	서비스_업종_코드_명	점포수대비건수_20대
1	패스트푸드점	2863
2	제과점	2638
3	커피음료	2103
4	분식집	1036
5	중국집	1035
6	양식집	775
7	한식음식점	693
8	일식집	623

3. 분석 보고서

요식업 창업을 준비하기 위해 서대문구 지역의 2015~2017년 시점에 대한 연령 및 성별 분석을 진행하였다. 20~24세 사이의 인구 비율이 가장 높은 수치를 보였으며 25~29세가 뒤따랐다. 이 결과는 서대문구 지역에 20대 인구가 가장 많이 거주한다는 사실을 나타낸다. 20대 남녀 성비 분석을 진행해보니 같은 연도에 대해서는 남녀 비율이 비슷하게 나타났으나, 매년 남성 인구 대비 여성 인구의 비율에 증가 폭을 보였다. 향후 몇 년간은 이와 같은 추세를 보일 것으로 분석된다.

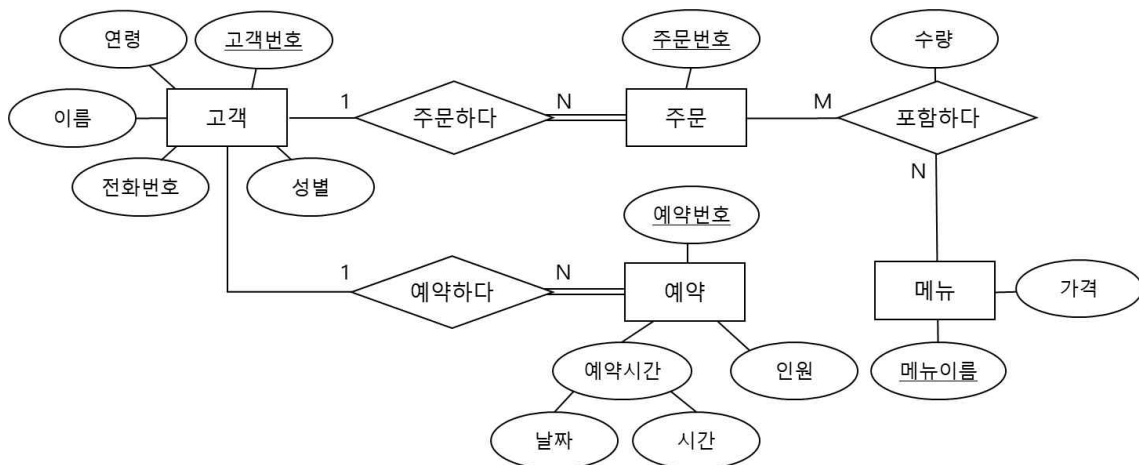
지역 인구 분석을 통해 서대문구 지역의 20대 여성을 주 타겟(target)층으로 정하고 20대 여성을 중심으로 서대문구 상권분석을 진행하였다. 요식업 업종을 대상으로 20대 매출금액을 분석해본 결과 분식집이 가장 높게 나타났으며 패스트푸드점이 뒤를 이었다. 하지만 1~2년, 2~3년 생존율을 분석한 결과 패스트푸드점의 생존율이 각 83.33%와 50%로 분식집 53.33%와 43.33에 비해 월등히 높은 수치를 보였다. 점포 수 대비 20대 여성 매출액과 건수 대비 20대 여성 매출액을 분석해보았을 때도 패스트푸드점이 각 22277215(원)와 2863(건)으로 가장 높은 수치를 나타냈다. 모든 결과를 종합해본 결과 패스트푸드점의 경쟁력이 가장 높을 것으로 분석된다.

따라서 서대문구 20대 여성을 주 고객층으로 한 패스트푸드점을 창업하기로 한다.

요구사항 정의서

1. 매장에서 다양한 고객이 주문.
2. 각 고객에 대해 고객번호(고유함), 이름, 전화번호, 성별, 연령을 저장.
3. 매장에서 다양한 메뉴를 판매.
4. 각 메뉴에 대해 메뉴이름(고유함), 가격을 저장.
5. 한 번의 주문에 대해 주문번호(고유함)를 부여. 각 주문은 한 명의 고객이 주문하며, 다양한 메뉴에 대해 다양한 수량을 주문할 수 있음.
6. 고객은 한 시점에 20명 이내의 좌석을 예약할 수 있음. 각 예약에 대해 예약번호(고유함), 예약시간, 인원을 저장. 예약시간은 날짜와 시각으로 세분화하여 나타냄.

ERD



관계 데이터베이스 스키마

1. ER 스키마를 관계 모델의 릴레이션으로 사상

단계 1 : 정규 엔티티 타입과 단일 값 애트리뷰트

고객(고객번호, 이름, 전화번호, 성별, 연령)
메뉴(메뉴이름, 가격)
주문(주문번호)
예약(예약번호, 날짜, 시간, 인원)

단계 4 : 정규 2진 1:N 관계 타입

주문(주문번호, 고객번호)
예약(예약번호, 고객번호, 날짜, 시간, 인원)

단계 5 : 2진 M:N 관계 타입

주문서(주문번호, 메뉴이름, 수량)

결과 : 총 5개의 릴레이션으로 사상

고객(고객번호, 이름, 전화번호, 성별, 연령)
메뉴(메뉴이름, 가격)
주문(주문번호, 고객번호)
예약(예약번호, 고객번호, 날짜, 시간, 인원)
주문서(주문번호, 메뉴이름, 수량)

2. 최종 관계 데이터베이스 스키마

고객

<u>고객번호</u>	이름	전화번호	성별	연령

메뉴

<u>메뉴이름</u>	가격

주문

<u>주문번호</u>	고객번호

예약

<u>예약번호</u>	고객번호	날짜	시간	인원

주문서

<u>주문번호</u>	메뉴이름	수량

정규화된 스키마

1. 함수의 종속성

- 고객 릴레이션

고객번호→이름 고객번호→전화번호 고객번호→성별 고객번호→연령
--

- 메뉴 릴레이션

메뉴이름→가격

- 주문 릴레이션

주문번호→고객번호

- 예약 릴레이션

예약번호→고객번호 예약번호→날짜 예약번호→시간 예약번호→인원
--

- 주문서 릴레이션

(주문번호, 메뉴이름)→수량

2. 정규화

- 제1정규형 : 모든 릴레이션의 모든 애트리뷰트가 원자값만 가지므로 만족
- 제2정규형 : 기본키를 두 개 가지는 주문서 릴레이션에서 부분 함수적 종속성이 존재하지 않으므로 만족
- 제3정규형 : 이행적 종속성을 가지는 릴레이션이 존재하지 않으므로 만족
- BCNF : 모든 결정자가 후보키이므로 만족

3. 결과 : 수정 사항 없음

고객(<u>고객번호</u> , 이름, 전화번호, 성별, 연령)
메뉴(<u>메뉴이름</u> , 가격)
주문(<u>주문번호</u> , 고객번호)
예약(<u>예약번호</u> , 고객번호, 날짜, 시간, 인원)
주문서(<u>주문번호</u> , <u>메뉴이름</u> , 수량)

테스트 데이터 적재 방법과 SQL 및 실행 결과

1. 데이터 생성

- 이름, 전화번호, 고객번호, 주문번호 등은 직접 부여함
- 총 50명의 고객, 200건의 주문, 20건의 예약, 375건의 주문서 데이터 생성
- 고객의 성별, 연령, 주문 내역 등은 엑셀의 rand() 함수로 난수를 발생시켜 적재

2. MSSQL 적재

- 생성된 엑셀 데이터를 플랫 데이터로 변환
- SQL Server 가져오기 및 내보내기 마법사를 통해 각 릴레이션 별 플랫 데이터를 SQL 서버로 복사함과 동시에 릴레이션 생성
- 각 릴레이션 별 Primary Key와 Foreign Key 생성

3. 적재 결과

dbo.고객

열

- 고객번호 (PK, int, Null이 아님)
- 이름 (char(50), Null)
- 전화번호 (char(50), Null)
- 성별 (char(50), Null)
- 연령 (int, Null)

dbo.메뉴

열

- 메뉴이름 (PK, char(50), Null이 아님)
- 가격 (int, Null)

dbo.예약

열

- 예약번호 (PK, int, Null이 아님)
- 고객번호 (FK, int, Null)
- 날짜 (date, Null)
- 시간 (time(0), Null)
- 인원 (int, Null)

dbo.주문

열

- 주문번호 (PK, int, Null이 아님)
- 고객번호 (FK, int, Null)

dbo.주문서

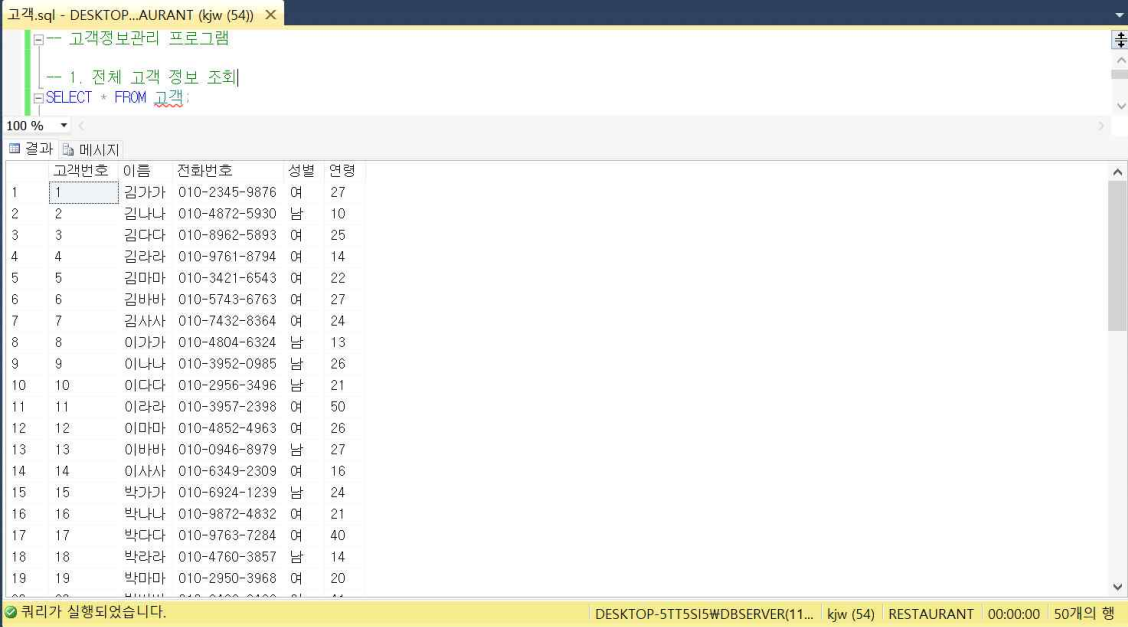
열

- 주문번호 (PK, FK, int, Null이 아님)
- 메뉴이름 (PK, FK, char(50), Null이 아님)
- 수량 (int, Null)

주문 관리 프로그램 SQL 및 실행 결과

1. 고객 정보 관리 프로그램

1) 전체 고객 정보 조회



고객.sql - DESKTOP...AURANT (kju (54))

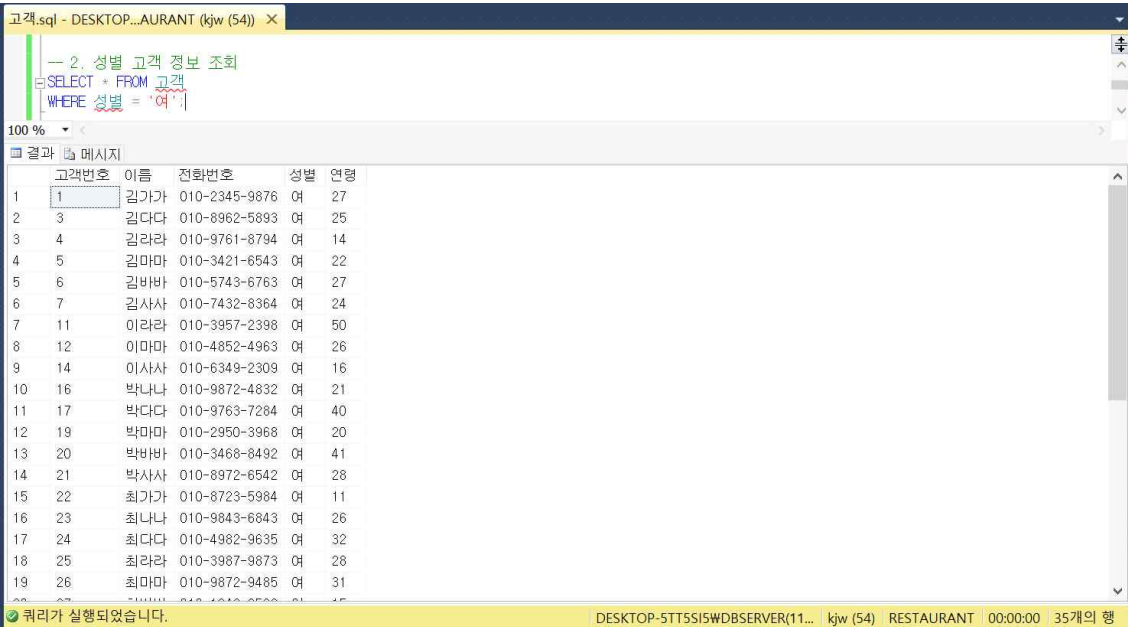
```
-- 고객정보관리 프로그램
-- 1. 전체 고객 정보 조회
SELECT * FROM 고객;
```

결과 메시지

	고객번호	이름	전화번호	성별	연령
1	1	김가가	010-2345-9876	여	27
2	2	김나나	010-4872-5930	남	10
3	3	김다다	010-8962-5893	여	25
4	4	김라라	010-9761-8794	여	14
5	5	김마마	010-3421-6543	여	22
6	6	김바바	010-5743-6763	여	27
7	7	김사사	010-7432-8364	여	24
8	8	이가가	010-4804-6324	남	13
9	9	이나나	010-3952-0985	남	26
10	10	이다다	010-2956-3496	남	21
11	11	이라라	010-3957-2398	여	50
12	12	이마마	010-4852-4963	여	26
13	13	이바바	010-0946-8979	남	27
14	14	이사사	010-6349-2309	여	16
15	15	박가가	010-6924-1239	남	24
16	16	박나나	010-9872-4832	여	21
17	17	박다다	010-9763-7284	여	40
18	18	박라라	010-4760-3857	남	14
19	19	박마마	010-2950-3968	여	20

DESKTOP-5TT5SI5WDBSERVER(11... kju (54) RESTAURANT 00:00:00 50개의 행

2) 성별 고객 정보 조회



고객.sql - DESKTOP...AURANT (kju (54))

```
-- 2. 성별 고객 정보 조회
SELECT * FROM 고객
WHERE 성별 = '여';
```

결과 메시지

	고객번호	이름	전화번호	성별	연령
1	1	김가가	010-2345-9876	여	27
2	3	김다다	010-8962-5893	여	25
3	4	김라라	010-9761-8794	여	14
4	5	김마마	010-3421-6543	여	22
5	6	김바바	010-5743-6763	여	27
6	7	김사사	010-7432-8364	여	24
7	11	이라라	010-3957-2398	여	50
8	12	이마마	010-4852-4963	여	26
9	14	이사사	010-6349-2309	여	16
10	16	박나나	010-9872-4832	여	21
11	17	박다다	010-9763-7284	여	40
12	19	박마마	010-2950-3968	여	20
13	20	박바바	010-3468-8492	여	41
14	21	박사사	010-8972-6542	여	28
15	22	최가가	010-8723-5984	여	11
16	23	최나나	010-9843-6843	여	26
17	24	최다다	010-4982-9635	여	32
18	25	최라라	010-3987-9873	여	28
19	26	최마마	010-9872-9485	여	31

DESKTOP-5TT5SI5WDBSERVER(11... kju (54) RESTAURANT 00:00:00 35개의 행

3) 연령별 고객 정보 조회

고객.sql - DESKTOP...AURANT (kju (54))*

```
-- 3. 연령별 고객 정보 조회
SELECT * FROM 고객
WHERE 연령 < 20
```

100 %

결과 메시지

	고객번호	이름	전화번호	성별	연령
1	2	김나나	010-4872-5930	남	10
2	4	김라라	010-9761-8794	여	14
3	8	이가가	010-4804-6324	남	13
4	14	이사사	010-6349-2309	여	16
5	18	박라라	010-4760-3857	남	14
6	22	최가가	010-8723-5984	여	11
7	27	최바바	010-1246-3592	여	15
8	30	강나나	010-3475-5682	여	10
9	32	강라라	010-7593-4860	여	18
10	37	조나나	010-6370-8643	여	15
11	44	서나나	010-4895-5886	남	14
12	50	서하하	010-8538-8353	여	11

쿼리가 실행되었습니다. DESKTOP-5TT5SI5WDBSERVER(11... kju (54) RESTAURANT 00:00:00 12개의 행

2. 메뉴 관리 프로그램

1) 전체 메뉴 정보 조회

메뉴.sql - DESKTOP...AURANT (kju (52)) * 고객.sql - DESKTOP...AURANT (kju (54))*

```
-- 메뉴관리 프로그램
-- 1. 전체 메뉴 정보 조회
SELECT * FROM 메뉴
```

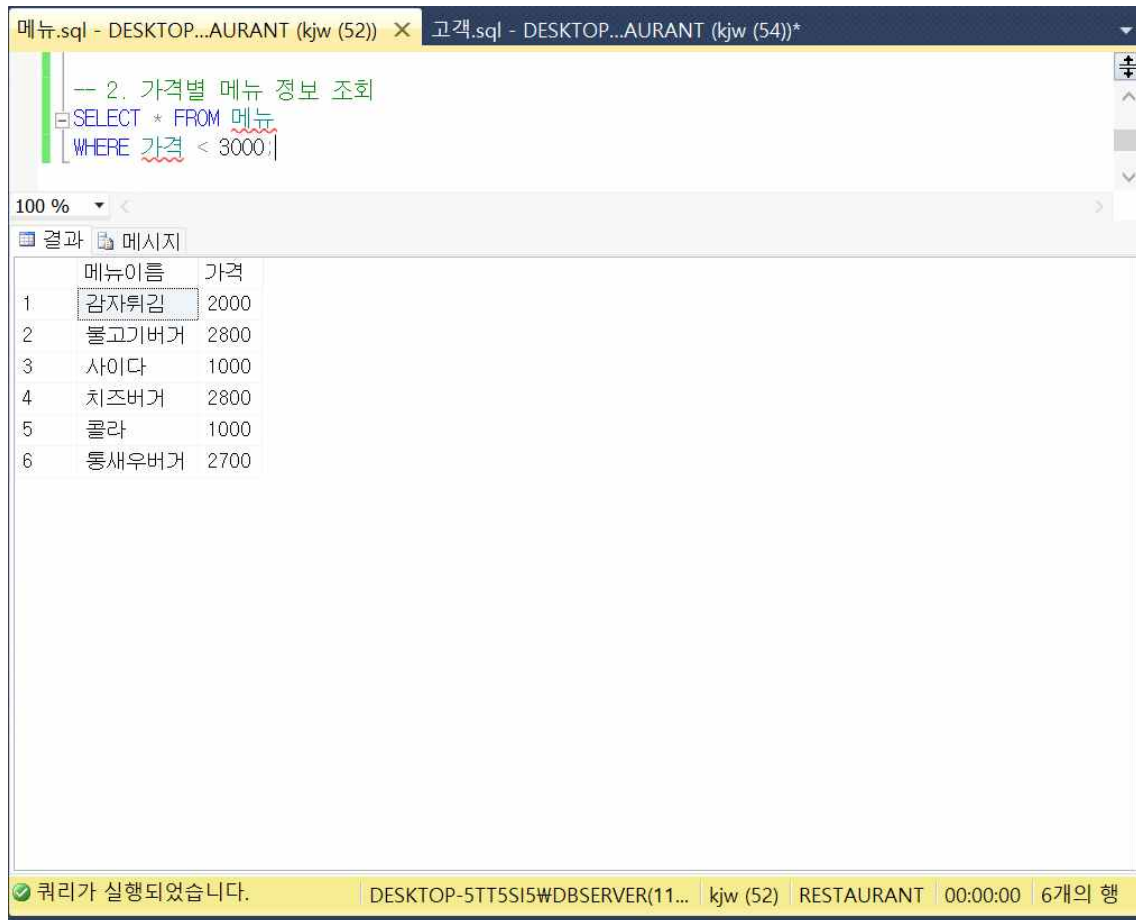
100 %

결과 메시지

	메뉴이름	가격
1	감자튀김	2000
2	불고기버거	2800
3	불고기버거세트	5000
4	사이다	1000
5	싸이버거	3200
6	싸이버거세트	5400
7	치즈버거	2800
8	치즈버거세트	5000
9	콜라	1000
10	통새우버거	2700
11	통새우버거세트	4900

쿼리가 실행되었습니다. DESKTOP-5TT5SI5WDBSERVER(11... kju (52) RESTAURANT 00:00:00 11개의 행

2) 가격별 메뉴 정보 조회



메뉴.sql - DESKTOP...AURANT (kju (52)) X 고객.sql - DESKTOP...AURANT (kju (54))*

```
-- 2. 가격별 메뉴 정보 조회  
SELECT * FROM 메뉴  
WHERE 가격 < 3000;
```

100 %

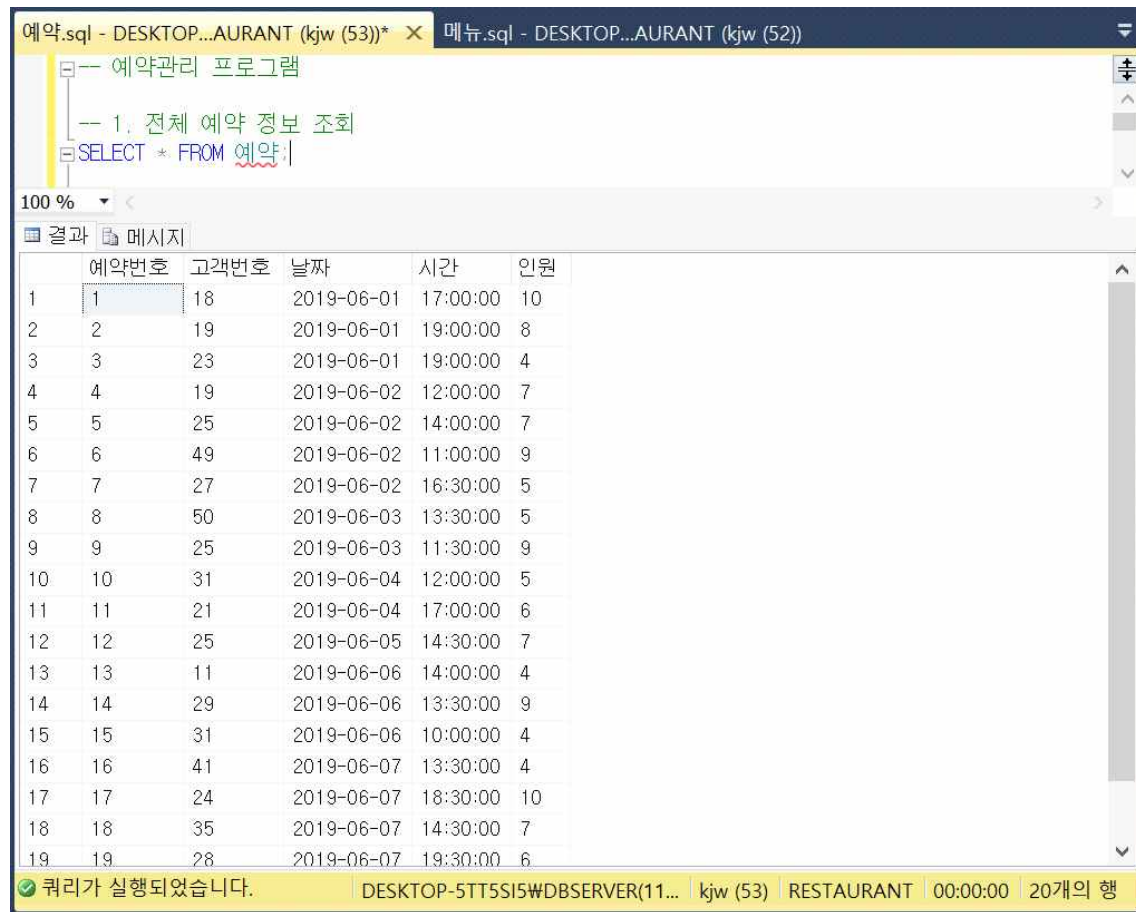
결과 메시지

	메뉴이름	가격
1	감자튀김	2000
2	불고기버거	2800
3	사이다	1000
4	치즈버거	2800
5	콜라	1000
6	통새우버거	2700

✓ 쿼리가 실행되었습니다. DESKTOP-5TT5SI5WDBSERVER(11... kju (52) RESTAURANT 00:00:00 6개의 행

3. 예약 관리 프로그램

1) 전체 예약 정보 조회



The screenshot shows a SQL query execution window with the following components:

- Query Editor:** Contains the SQL query `SELECT * FROM 예약;` and a tree view on the left showing the query structure.
- Results Panel:** Displays the query results in a table with 6 columns: 예약번호, 고객번호, 날짜, 시간, 인원. The table contains 19 rows of data.
- Status Bar:** Shows a green checkmark icon and the message "쿼리가 실행되었습니다." (Query executed successfully).
- Footer:** Displays the connection information: "DESKTOP-5TT5SI5WDBSERVER(11... | kju (53) | RESTAURANT | 00:00:00 | 20개의 행".

	예약번호	고객번호	날짜	시간	인원
1	1	18	2019-06-01	17:00:00	10
2	2	19	2019-06-01	19:00:00	8
3	3	23	2019-06-01	19:00:00	4
4	4	19	2019-06-02	12:00:00	7
5	5	25	2019-06-02	14:00:00	7
6	6	49	2019-06-02	11:00:00	9
7	7	27	2019-06-02	16:30:00	5
8	8	50	2019-06-03	13:30:00	5
9	9	25	2019-06-03	11:30:00	9
10	10	31	2019-06-04	12:00:00	5
11	11	21	2019-06-04	17:00:00	6
12	12	25	2019-06-05	14:30:00	7
13	13	11	2019-06-06	14:00:00	4
14	14	29	2019-06-06	13:30:00	9
15	15	31	2019-06-06	10:00:00	4
16	16	41	2019-06-07	13:30:00	4
17	17	24	2019-06-07	18:30:00	10
18	18	35	2019-06-07	14:30:00	7
19	19	28	2019-06-07	19:30:00	6

2) 예약 정보 날짜별 검색

예약.sql - DESKTOP...AURANT (kju (53)) X 메뉴.sql - DESKTOP...AURANT (kju (52))

```
-- 2. 예약 정보 날짜별 검색  
SELECT * FROM 예약  
WHERE 날짜 = '2019-06-01';
```

100 %

결과 메시지

	예약번호	고객번호	날짜	시간	인원
1	1	18	2019-06-01	17:00:00	10
2	2	19	2019-06-01	19:00:00	8
3	3	23	2019-06-01	19:00:00	4

쿼리가 실행되었습니다. DESKTOP-5TT5SI5WDBSERVER(11... kju (53) RESTAURANT 00:00:00 3개의 행

3) 예약 인원별 검색

예약.sql - DESKTOP...AURANT (kju (53))* X 메뉴.sql - DESKTOP...AURANT (kju (52))

```
-- 3. 예약 인원별 검색  
SELECT * FROM 예약  
WHERE 인원 > 5;
```

100 %

결과 메시지

	예약번호	고객번호	날짜	시간	인원
1	1	18	2019-06-01	17:00:00	10
2	2	19	2019-06-01	19:00:00	8
3	4	19	2019-06-02	12:00:00	7
4	5	25	2019-06-02	14:00:00	7
5	6	49	2019-06-02	11:00:00	9
6	9	25	2019-06-03	11:30:00	9
7	11	21	2019-06-04	17:00:00	6
8	12	25	2019-06-05	14:30:00	7
9	14	29	2019-06-06	13:30:00	9
10	17	24	2019-06-07	18:30:00	10
11	18	35	2019-06-07	14:30:00	7
12	19	28	2019-06-07	19:30:00	6

쿼리가 실행되었습니다. DESKTOP-5TT5SI5WDBSERVER(11... kju (53) RESTAURANT 00:00:00 12개의 행

4. 주문 관리 프로그램

1) 전체 주문 조회

주문&주문서.sql - DE...RANT (kju (55))* X

```
-- 주문관리 프로그램
-- 1. 전체 주문 조회
SELECT * FROM 주문, 주문서
WHERE 주문 주문번호 = 주문서 주문번호;
```

100 %

결과 메시지

	주문번호	고객번호	주문번호	메뉴이름	수량
1	1	17	1	불고기버거	1
2	1	17	1	치즈버거	3
3	1	17	1	콜라	2
4	2	21	2	사이다	2
5	2	21	2	싸이버거	2
6	3	37	3	감자튀김	1
7	3	37	3	사이다	1
8	3	37	3	통새우버거세트	2
9	4	26	4	사이다	1
10	4	26	4	통새우버거	2
11	5	24	5	치즈버거	1
12	5	24	5	콜라	2
13	6	44	6	불고기버거	2
14	6	44	6	사이다	1
15	6	44	6	싸이버거세트	2
16	7	39	7	불고기버거세트	2
17	7	39	7	치즈버거	3
18	8	14	8	치즈버거	2

DESKTOP-5TT5SI5WDBSERVER(11... | kju (55) | RESTAURANT | 00:00:00 | 375개의 행

2) 메뉴별 주문 조회

주문&주문서.sql - DE...RANT (kju (55))* X

```
-- 2. 메뉴별 주문 조회
SELECT * FROM 주문, 주문서
WHERE 주문 주문번호 = 주문서 주문번호
AND 메뉴이름 = '콜라';
```

100 %

결과 메시지

	주문번호	고객번호	주문번호	메뉴이름	수량
1	1	17	1	콜라	2
2	5	24	5	콜라	2
3	8	14	8	콜라	3
4	13	7	13	콜라	2
5	14	50	14	콜라	3
6	18	16	18	콜라	1
7	27	8	27	콜라	1
8	32	34	32	콜라	2
9	34	23	34	콜라	1
10	35	9	35	콜라	1
11	37	39	37	콜라	2
12	38	31	38	콜라	2
13	44	37	44	콜라	1
14	45	47	45	콜라	3
15	48	31	48	콜라	2
16	58	30	58	콜라	1
17	61	45	61	콜라	1
18	62	18	62	콜라	2

✓ 쿼리 | DESKTOP-5TT5SI5WDBSERVER(11... | kju (55) | RESTAURANT | 00:00:00 | 65개의 행

3) 고객별 주문 조회

주문&주문서.sql - DE...RANT (kju (55))* X

```
-- 3. 고객별 주문 조회
SELECT * FROM 주문, 주문서
WHERE 주문 주문번호 = 주문서 주문번호
AND 고객번호 = 17;
```

100 % < >

결과 메시지

	주문번호	고객번호	주문번호	메뉴이름	수량
1	1	17	1	불고기버거	1
2	1	17	1	치즈버거	3
3	1	17	1	콜라	2
4	24	17	24	감자튀김	1
5	24	17	24	사이다	2
6	24	17	24	싸이버거	2
7	51	17	51	불고기버거	3
8	51	17	51	싸이버거	1
9	77	17	77	감자튀김	2
10	77	17	77	사이다	1
11	77	17	77	치즈버거	1
12	77	17	77	콜라	1
13	88	17	88	통새우버거세트	3

쿼리 | DESKTOP-5TT5SI5WDBSERVER(11... | kju (55) | RESTAURANT | 00:00:00 | 13개의 행

SQL 분석을 통한 분석 리포트

1. SQL 분석

1) 주문별 총 판매액

분석쿼리.sql - DESKT...URANT (kju (52))* X

```
-- 1. 주문별 총액
SELECT 주문번호, SUM(수량*가격) AS 총금액
FROM 주문서, 메뉴
WHERE 주문서.메뉴이름 = 메뉴.메뉴이름
GROUP BY 주문번호
ORDER BY 주문번호;
```

100 %

결과 메시지

	주문번호	총금액
1	1	13200
2	2	8400
3	3	12800
4	4	6400
5	5	4800
6	6	17400
7	7	18400
8	8	8600
9	9	10000
10	10	14800
11	11	10000
12	12	6000
13	13	7600
14	14	5000
15	15	14000
16	16	6400
17	17	11400

✓ 쿼리가 실행되었습니다.

2) 메뉴별 총 판매량 및 총 판매액

```
-- 2. 메뉴별 총판매량 총판매액
SELECT 메뉴.메뉴이름, SUM(수량) AS 총판매량, SUM(수량*가격) AS 총판매액
FROM 주문서, 주문, 메뉴
WHERE 주문서.메뉴이름 = 메뉴.메뉴이름
AND 주문서.주문번호 = 주문.주문번호
GROUP BY 메뉴.메뉴이름
ORDER BY 총판매량 DESC;
```

100 %

결과 메시지

	메뉴이름	총판매량	총판매액
1	콜라	111	111000
2	사이다	100	100000
3	싸이버거세트	54	291600
4	치즈버거세트	54	270000
5	통새우버거세트	50	245000
6	불고기버거세트	49	245000
7	감자튀김	46	92000
8	불고기버거	43	120400
9	싸이버거	40	128000
10	통새우버거	38	102600
11	치즈버거	36	100800

콜라, 사이다가 판매량이 가장 많았지만, 주메뉴를 살펴보면 싸이버거세트, 치즈버거세트가 가장 많이 팔렸다. 단품보다는 세트가 더 많이 팔리는 것을 볼 수 있다.

3) 성별 총 판매액

```
-- 3. 성별 총판매액
SELECT 고객.성별, SUM(수량*가격) AS 총판매액
FROM 주문서, 메뉴, 고객
WHERE 주문서.메뉴이름 = 메뉴.메뉴이름
GROUP BY 고객.성별;
```

100 % <

결과 메시지

	성별	총판매액
1	여	63224000
2	남	27096000

남성 고객보다 여성 고객에 의한 매출이 더 높았다.

4) 연령별 총 판매액

```
-- 4. 연령별 총판매액
( SELECT '10-19세' AS 연령, SUM(수량*가격) AS 총판매액
FROM 주문서, 주문, 메뉴, 고객
WHERE 주문서.메뉴이름 = 메뉴.메뉴이름
AND 주문서.주문번호 = 주문.주문번호
AND 주문.고객번호 = 고객.고객번호
AND 연령 BETWEEN 10 AND 19)
UNION
( SELECT '20-29세' AS 연령, SUM(수량*가격) AS 총판매액
FROM 주문서, 주문, 메뉴, 고객
WHERE 주문서.메뉴이름 = 메뉴.메뉴이름
AND 주문서.주문번호 = 주문.주문번호
AND 주문.고객번호 = 고객.고객번호
AND 연령 BETWEEN 20 AND 29)
UNION
( SELECT '30-39세' AS 연령, SUM(수량*가격) AS 총판매액
FROM 주문서, 주문, 메뉴, 고객
WHERE 주문서.메뉴이름 = 메뉴.메뉴이름
AND 주문서.주문번호 = 주문.주문번호
AND 주문.고객번호 = 고객.고객번호
AND 연령 BETWEEN 30 AND 39)
UNION
( SELECT '40-49세' AS 연령, SUM(수량*가격) AS 총판매액
FROM 주문서, 주문, 메뉴, 고객
WHERE 주문서.메뉴이름 = 메뉴.메뉴이름
AND 주문서.주문번호 = 주문.주문번호
AND 주문.고객번호 = 고객.고객번호
AND 연령 BETWEEN 40 AND 49)
UNION
( SELECT '50세 이상' AS 연령, SUM(수량*가격) AS 총판매액
FROM 주문서, 주문, 메뉴, 고객
WHERE 주문서.메뉴이름 = 메뉴.메뉴이름
AND 주문서.주문번호 = 주문.주문번호
AND 주문.고객번호 = 고객.고객번호
AND 연령 >= 50);
```

결과 메시지		
	연령	총판매액
1	10-19세	502600
2	20-29세	793700
3	30-39세	290500
4	40-49세	170600
5	50세 이상	49000

연령별 판매액을 살펴본 결과, 20대 고객에 의한 매출이 가장 높았다. 그 다음으로 10대, 30대 순이었고, 50세 이상 고객의 방문이 가장 적은 것으로 나타났다.

5) 20대의 최다 판매 메뉴

-- 5. 20대 최다판매메뉴

SELECT 메뉴.메뉴이름, SUM(수량) AS 판매량, SUM(수량*가격) AS 판매액
FROM 주문서, 주문, 메뉴, 고객
WHERE 주문서.메뉴이름 = 메뉴.메뉴이름
AND 주문서.주문번호 = 주문.주문번호
AND 주문.고객번호 = 고객.고객번호
AND 연령 BETWEEN 20 AND 29
GROUP BY 메뉴.메뉴이름
ORDER BY 판매액 DESC;

91 %

결과

메시지

	메뉴이름	판매량	판매액
1	싸이버거세트	29	156600
2	불고기버거세트	29	145000
3	통새우버거세트	18	88200
4	치즈버거세트	17	85000
5	싸이버거	26	83200
6	통새우버거	19	51300
7	사이다	43	43000
8	감자튀김	21	42000
9	콜라	35	35000
10	치즈버거	12	33600
11	불고기버거	11	30800

가장 매출이 높았던 연령대인 20대에게 가장 많이 팔리는 메뉴를 살펴보았다. 싸이버거세트와 불고기버거세트가 동일하게 많이 팔렸음을 알 수 있다.

6) 성별 최대 판매 메뉴

- 여성 고객 최대 판매 메뉴

-- 6. 여성 최대판매메뉴

```
SELECT 메뉴.메뉴이름, SUM(수량) AS 판매량, SUM(수량*가격) AS 판매액
FROM 주문서, 주문, 메뉴, 고객
WHERE 주문서.메뉴이름 = 메뉴.메뉴이름
AND 주문서.주문번호 = 주문.주문번호
AND 주문.고객번호 = 고객.고객번호
AND 성별 = '여'
GROUP BY 메뉴.메뉴이름
ORDER BY 판매량 DESC;
```

91 %

결과 메시지

	메뉴이름	판매량	판매액
1	콜라	76	76000
2	사이다	69	69000
3	치즈버거세트	41	205000
4	통새우버거세트	41	200900
5	사이버거세트	37	199800
6	감자튀김	37	74000
7	불고기버거세트	34	170000
8	사이버거	29	92800
9	통새우버거	27	72900
10	치즈버거	25	70000
11	불고기버거	23	64400

여성 고객은 주요 메뉴 중 치즈버거세트와 통새우버거세트를 가장 선호하였다.

- 남성 최다 판매 메뉴

7. 남성 최다판매메뉴

```
SELECT 메뉴.메뉴이름, SUM(수량) AS 판매량, SUM(수량*가격) AS 판매액
FROM 주문서, 주문, 메뉴, 고객
WHERE 주문서.메뉴이름 = 메뉴.메뉴이름
AND 주문서.주문번호 = 주문.주문번호
AND 주문.고객번호 = 고객.고객번호
AND 성별 = '남'
GROUP BY 메뉴.메뉴이름
ORDER BY 판매량 DESC;
```

91 %

결과 메시지

	메뉴이름	판매량	판매액
1	콜라	35	35000
2	사이다	31	31000
3	불고기버거	20	56000
4	싸이버거세트	17	91800
5	불고기버거세트	15	75000
6	치즈버거세트	13	65000
7	치즈버거	11	30800
8	싸이버거	11	35200
9	통새우버거	11	29700
10	통새우버거세트	9	44100
11	감자튀김	9	18000

남성 고객은 주요 메뉴 중 불고기버거를 가장 선호하였다.

2. 분석 리포트

분석 결과, 20대 · 여성에 의한 매출이 가장 높았으며, 가장 인기 있는 메뉴는 싸이버거세트와 치즈버거세트였다. 가장 매출이 높은 연령 집단인 20대의 선호 메뉴는 싸이버거세트와 불고기버거세트이며, 가장 매출이 높은 성별 집단인 여성의 선호 메뉴는 치즈버거세트와 통새우버거세트이다. 주로 단품보다는 세트가 매출이 높으므로 세트 제품을 빠르게 준비할 수 있도록 해야 한다. 또한, 주요 메뉴 이외의 사이드 메뉴의 판매량도 높은 것으로 보아 다양한 사이드 메뉴를 준비하면 매출에 도움이 될 것으로 보인다. 버거와 잘 어울리는 사이드 메뉴 또는 비건 메뉴를 추가하는 것도 좋을 것이다.

고객이 주로 10대, 20대에 몰려있다는 단점이 존재하며, 연령대가 높은 집단을 만족시키는 메뉴를 개발하면 매출을 증대할 수 있을 것으로 보인다. 패스트푸드의 단점보다는 추억을 자극하는 메뉴를 추가하면 좋을 것 같다.