

## [보고서의 목차]

### I. 기존 대비 차이점, 발전성

- i) 제시한 주제
- ii) 제안서에 제출된 내용과의 비교

### II. 요구사항 분석 및 개념적 모델링

- i) 요구사항 명세서
- ii) ER Diagram

### III. 논리적 모델링

- i) 테이블 정의서
- ii) SQL DDL 문을 통한 스키마 생성

### IV. 구현

- i) SQL DML 문을 이용한 실행
  - 1) INSERT
  - 2) SELECT
- ii) 소스 파일(별도 첨부)

### V. 기타

## I. 기존 대비 차이점, 발전성

### i) 제시한 주제

국내 음식점과 고객의 관계를 통한 예약 관리 및 거래처 관리

### ii) 제안서에 제출된 내용과의 비교

#### 1) 차이점

기존 기말과제 제안서로 제출한 요구사항 명세서에서 나타난 현실적이지 못하거나 불필요한 속성 부분(음식점의 전화번호, 월 매출 등)을 수정했습니다. 예를 들면, 음식점의 전화번호가 여러 개 가지고 있는 경우는 그 규모가 매우 큰 점포가 아닌 이상 현실적으로 가능성이 희박하고, 여러 개라 해도 대표번호로 연결하였을 때 통화 중인 경우에는 다른 번호로 연결해주는 (링크) 서비스가 존재하기 때문에 요구사항 명세서에서는 대표번호 하나만 존재한다고 가정하였고, 이를 통한 ER Diagram에서도 해당 부분을 수정하였습니다.

또한, 음식점의 별점의 경우 기존에는 제한이 없었으나, 대부분 별점(혹은 평점)을 5점 만점으로 계산하고 있으므로 최대 점수를 5점으로 제한을 두었고, 이에 대한 테이블 제작 시, 제약조건을 통하여 5점 초과한 평점은 입력할 수 없도록 하였습니다.

마지막으로 ER 다이어그램 작성 시 누락된 음식점에 상호(이름) 속성을 추가하였고, 음식점이 중추적 역할이므로 이를 약한 개체가 아닌 강한 개체로 식별되도록 하였습니다.

#### 2) 발전성

기말과제 제안서에는 개념적 모델링의 결과인 ERD(ER 다이어그램)까지만 제시하였지만, 현재 제출될 기말과제 보고서에는 논리적 모델링 결과인 관계형 모델로 변환하여 테이블의 스키마를 설계하고 그 결과가 테이블 정의서로 기술됩니다.

요구사항 명세서로만 설계를 마친 결과를 데이터베이스 언어인 SQL 문을 이용하여 명세서에 제시된 조건들을 끼워 맞춰가며 제약조건을 설정하고, 전체 테이블을 실제 구현함으로써, 원하는 결과값을 탐색하고자 할 때, 성공적인 탐색을 할 수 있었습니다.

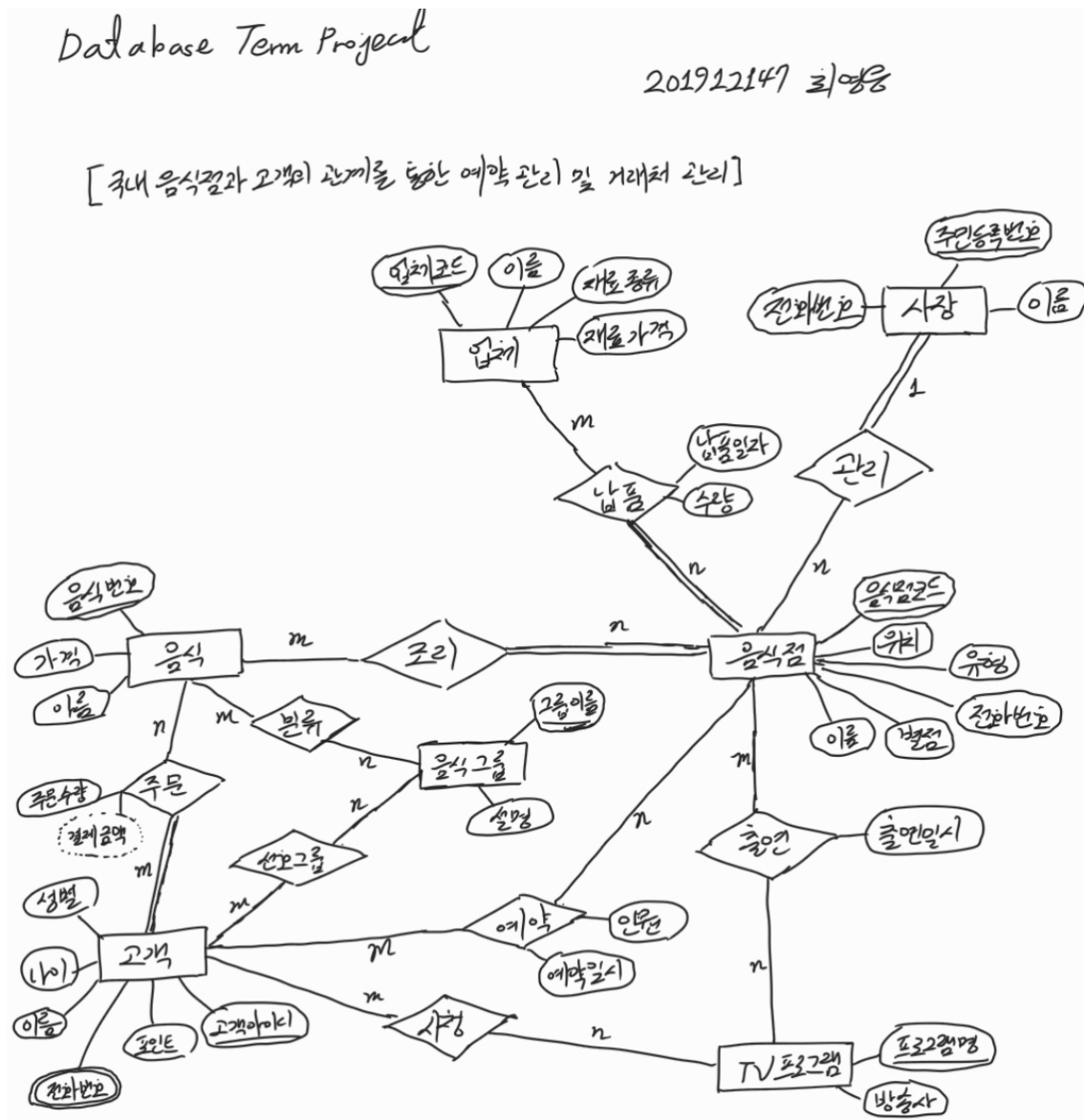
## II. 요구사항 분석 및 개념적 모델링

### i) 요구사항 명세서

- 음식은 음식 번호(고유함), 음식 이름, 가격에 대한 정보가 저장된다.

- 음식들은 여러 음식 종류(예를 들면, 한식, 중식 등)로 분류되며, 각 음식은 여러 종류로 분류될 수 있다. 각 음식 종류는 이름(같은 이름은 없다고 가정한다.), 그에 대한 설명을 갖는다.
- 음식점은 음식점 코드, 음식점 유형, 음식점 이름, 음식점 위치, 전화번호, 별점에 대한 정보를 가지며 음식점 코드는 각 음식점을 구분할 수 있는 유일한 속성이다. 단, **별점은 최대 5점 만점으로 가정**한다.
- 음식점의 전화번호는 대표번호 **하나만 존재한다**고 가정한다.
- 사장은 음식점을 전적으로 책임지고 관리하는데, 한 사장은 여러 음식점을 관리할 수 있지만 한 음식점에 여러 사장이 존재하진 않는다. 사장은 무조건 하나 이상의 음식점을 관리한다.
- 사장은 주민등록번호, 사장 이름, 전화번호의 정보를 갖는다.
- 음식점은 TV 프로그램에 출연할 수 있는데, 출연하게 된다면 출연일시에 대한 정보도 함께 저장된다. 또한, 한 음식점은 여러 TV 프로그램에 출연할 수 있고, 역시 한 TV 프로그램에 여러 음식점이 출연하는 것도 가능하다.
- TV 프로그램은 프로그램명(고유하다고 하자.), 방송사의 정보를 갖는다.
- 음식점은 음식을 조리하는데, 한 음식점은 여러 음식을 조리할 수 있고, 한 음식은 여러 음식점에서 조리될 수 있다. 이때 음식점은 음식을 반드시 하나 이상은 조리하여야 한다.
- 고객은 고객 아이디(고유함), 고객 이름, 나이, 성별, 전화번호, 포인트로 구성되고, 고객별 선호하는 음식 종류들도 존재한다. 고객의 전화번호는 여러 개일 수 있다.
- 고객은 TV 프로그램을 시청하는데, 한 고객은 여러 TV 프로그램을 시청할 수 있고, 한 TV 프로그램은 여러 고객에 의해 시청될 수 있다.
- 고객이 음식을 먹기 위해서 음식점에 예약할 수 있는데, 예약할 때 **예약(방문할)** 일시와 인원을 **같이** 저장한다. 한 음식점은 여러 고객에게 예약을 받을 수 있고, **한 고객은 (방문일시가 다르다면, ) 여러 음식점을 예약**할 수 있다.
- 고객은 음식점에서 음식을 주문하는데, 한 고객은 여러 음식을 주문할 수 있지만, 하나도 주문하지 않을 수는 없으며, 한 음식은 여러 고객에 의해 주문될 수 있다. 주문 시에는 주문 수량이 같이 저장된다. (결제 금액은 주문 수량과 음식별 가격으로 유도 가능하다고 본다.)
- 음식점은 거래처에서 음식에 필요한 재료를 납품받는다. 거래처는 고유한 업체코드가 존재하고, 업체 이름, **취급**재료 종류, 재료 가격으로 구성된다. 각 음식점은 여러 거래처로부터 납품받을 수 있고, 각 거래처는 여러 음식점으로 납품할 수 있지만, 음식점은 최소한 한 곳 이상의 거래처로부터 납품받아야 한다. 납품할 때, 납품 일자와 수량도 함께 저장된다.

ii) ER 다이어그램(ER Diagram)



### III. 논리적 모델링

i) 테이블 정의서

강의에서 배웠듯이, ER 다이어그램에 나타난 여러 관계들(1:1, 1:N, M:N 등)을 관계 데이터 모델로 변환하는 규칙에 따라 테이블로 변환하였습니다. 해당 프로젝트에는 개체의 특성상 관계의 대부분이 M:N이고, 일부 필요한 관계에만 1:N 관계가 적용되었습니다.

1) 1:N 관계: N측의 릴레이션을 찾고, 해당 릴레이션에는 1측의 기본키(사장-주민 등록번호)를 외래키로서 참조하도록 추가하였습니다. 이때, 1:N은 N측 개체가 1측 인스턴스와 최대 1개만 연결되는 것이 보장됩니다.

2) M:N 관계: 해당 관계에 속하는 두 릴레이션의 기본키를 새로운 관계 릴레이션의 외래키로서 두고, 이 둘을 조합한 키를 기본키로 합니다. 관계에 부가적 속성이 존재할 경우, 새로이 생성된 릴레이션의 속성으로서 포함합니다.

따라서 변환된 릴레이션은 다음과 같습니다.

- 업체(업체코드, 이름, 취급재료 종류, 가격)
- 사장(주민등록번호, 이름, 전화번호)
- 음식점(음식점코드, 유형, 이름, 위치, 별점, 전화번호, 주민등록번호(FK))
- 납품(음식점코드(FK), 업체코드(FK), 납품일자, 수량)
- 음식(음식번호, 이름, 가격)
- 조리(음식점코드(FK), 음식번호(FK))
- 음식그룹(그룹이름, 설명)
- 분류(그룹이름(FK), 음식번호(FK))
- 고객(고객아이디, 이름, 성별, 나이, 포인트)
- 고객연락처(고객아이디(FK), 전화번호)
- 선호그룹(고객아이디(FK), 그룹이름(FK))
- 주문(고객아이디(FK), 음식번호(FK), 주문수량)
- 예약(고객아이디(FK), 음식점코드(FK), 예약일시(U), 인원)
- TV프로그램(프로그램명, 방송사)
- 시청(고객아이디(FK), 프로그램명(FK))
- 출연(프로그램명(FK), 음식점코드(FK), 출연일시)

## ii) SQL DDL 문을 통한 스키마 생성

(공통)

해당 프로그램에서 한글 1자를 3바이트로 취급하므로, 해당 부분을 고려하여 필드의 입력가능한 최대 크기를 설정하였습니다.

간혹 테이블 이름 외부에 따옴표가 설정된 경우는 예약어(Reserved Word)와 해당 테이블 이름이 서로 중복되어 테이블 생성 시 오류를 발생시켰기 때문입니다.

```
CREATE TABLE supplier (
    supplier_code VARCHAR2(6) PRIMARY KEY,
    supplier_name VARCHAR2(24) NOT NULL,
    handling_type VARCHAR2(35),
    supplier_price NUMBER(8) NOT NULL

    CONSTRAINT CHK_supplier_price
        CHECK (supplier_price > 0)
);
```

(공급) 업체: 해당 릴레이션의 속성들을 모두 포함했으며, 취급재료의 종류가 정해져 있지 않고 여러 종류를 다양하게 취급하면 해당 속성은 입력되지 않아도 되도록 설정했습니다. 또한, 공급가가 0원 이하인 경우는 입력되지 않도록 제약조건을 설정하였습니다.

```
CREATE TABLE "owner" (
    id_number CHAR(14) PRIMARY KEY,
    owner_name VARCHAR2(12) NOT NULL,
    owner_phone_number CHAR(13),

    CHECK (id_number LIKE '%-%-'),
    CONSTRAINT CHK_owner_number_type
        CHECK (owner_phone_number LIKE '%-%-%-')
);
```

사장: 해당 릴레이션의 속성들을 모두 포함했으며, 제약조건을 통해 사장의 주민등록 번호의 형식, 사장의 전화번호 형식이 각각 제시된 형식과 일치하는 튜플만 추가 가능합니다.

```
CREATE TABLE restaurant (
    restaurant_code VARCHAR2(6) PRIMARY KEY,
    restaurant_type VARCHAR2(12),
    restaurant_name VARCHAR2(30) NOT NULL,
    restaurant_location VARCHAR2(20) NOT NULL,
    rating NUMBER(3, 2),
    restaurant_phone_number CHAR(13),
    fk_owner_id_number CHAR(14) NOT NULL,

    FOREIGN KEY (fk_owner_id_number)
        REFERENCES "owner"(id_number),

    CONSTRAINT CHK_restaurant_number_type
        CHECK (restaurant_phone_number LIKE '%-%-%-'),

    CONSTRAINT CHK_rating
        CHECK (rating <= 5.0)
);
```

음식점: 해당 릴레이션의 속성들을 모두 포함했으며, 사장이 음식점을 관리하므로 관

계에 따른 릴레이션 변환 과정에 의해 추가된 외래키 주민등록번호를 추가해주었습니다. 이때, 해당 외래키는 사장 테이블의 주민등록번호(기본키)를 참조합니다.

```
CREATE TABLE supply (
    fk_restaurant_code VARCHAR2(6) NOT NULL,
    fk_supplier_code VARCHAR2(6) NOT NULL,
    supply_date DATE NOT NULL,
    supplying_quantity NUMBER(3) NOT NULL,

    CONSTRAINT CHK_supplying_quantity
        CHECK (supplying_quantity > 0),

    PRIMARY KEY(fk_restaurant_code, fk_supplier_code),

    FOREIGN KEY (fk_restaurant_code)
    REFERENCES restaurant(restaurant_code),

    FOREIGN KEY (fk_supplier_code)
    REFERENCES supplier(supplier_code)
);
```

납품: 납품 테이블은 M:N 관계의 변환규칙을 적용하여 변환된 테이블입니다. 기본키는 음식점의 기본키인 음식점 코드, 업체의 기본키인 업체코드의 결합인 (음식점코드(FK), 업체코드(FK))입니다. 납품을 받을 때, 무조건 1단위 이상 납품받아야 하므로 제약조건으로 납품 개수에 대한 조건을 추가하였고, 납품일이 NULL이면 투명한 납품이 되지 못하므로, NULL 무결성 제약조건을 추가하였습니다.

```
CREATE TABLE food (
    food_number VARCHAR2(6) PRIMARY KEY,
    food_name VARCHAR2(20) NOT NULL,
    food_price NUMBER(6) NOT NULL
);
```

음식: 해당 릴레이션의 속성들을 모두 포함하여 작성하였습니다. 요구사항 명세서에 제시되어 있듯, 음식은 음식번호로 유일하게 구별되고 식별되므로 기본키로 설정하였습니다. 이때, 음식은 음식점에서 판매되는 것이므로, (명시되진 않았으나) 이름과 가격 역시 NULL값이 입력될 수 없도록 하였습니다.

```
CREATE TABLE cook (
    fk_restaurant_code VARCHAR2(6) NOT NULL,
    fk_food_number VARCHAR2(6) NOT NULL,

    PRIMARY KEY (fk_restaurant_code, fk_food_number),

    FOREIGN KEY (fk_restaurant_code)
    REFERENCES restaurant(restaurant_code),

    FOREIGN KEY (fk_food_number)
    REFERENCES food(food_number)
);
```

조리: 조리 테이블은 M:N 관계의 변환규칙을 적용하여 변환된 테이블입니다. 조리는 음식점에서 음식을 하면서 생성되는 관계이므로, 음식점의 기본키인 음식점 코드와 음식의 기본키인 음식 번호를 각각 해당 릴레이션의 외래키로 가져온 후, 이 둘의 순서쌍을 기본키로 설정하였습니다.

```
CREATE TABLE food_group (
    group_name VARCHAR2(12) PRIMARY KEY,
    "description" VARCHAR2(100)
);
```

음식그룹: 음식그룹은 그룹의 이름과 그에 대한 설명을 갖습니다. 음식의 그룹을 식별하는 키는 (이름이 중복되지 않는 가정이 존재하므로) 그룹 이름을 음식 그룹 테이블의 기본키로 하였습니다.

```
CREATE TABLE "category" (
    fk_group_name VARCHAR2(6) NOT NULL,
    fk_food_number VARCHAR2(6) NOT NULL,

    PRIMARY KEY (fk_group_name, fk_food_number),

    FOREIGN KEY (fk_group_name)
    REFERENCES food_group(group_name),

    FOREIGN KEY (fk_food_number)
    REFERENCES food(food_number)
);
```

분류: 분류는 그룹의 이름과 그 그룹에 해당하는 음식 번호를 매핑해놓은 관계 테이블로, 그룹, 음식의 기본키를 참조하는 방식(외래키)을 사용하였습니다. (참조 무결성 유지)



```
CREATE TABLE customer (
    customer_id VARCHAR2(15) PRIMARY KEY,
    customer_name VARCHAR2(12) NOT NULL,
    customer_gender CHAR(3) NOT NULL,
    customer_age NUMBER(2),
    "point" NUMBER(5) NOT NULL
);
```

고객: 고객의 정보를 갖는 테이블입니다. 고객은 고객 아이디로서 구별되므로 해당 테이블의 기본키로 설정하였으며, 나이 정보를 제외한 모든 속성은 무조건 입력되도록 (즉, NULL 값 삽입이 되지 않도록) 하였습니다. 그리고 성별은 '남', '여'로만 입력되므로, 해당 필드의 도메인을 고정길이 최대 3바이트(한글 1자) 문자열로 도메인을 제한하였습니다.

```
CREATE TABLE preferred_group (
    fk_customer_id VARCHAR2(15) NOT NULL,
    fk_group_name NOT NULL,

    PRIMARY KEY (fk_customer_id, fk_group_name),

    FOREIGN KEY (fk_customer_id)
    REFERENCES customer(customer_id),

    FOREIGN KEY (fk_group_name)
    REFERENCES food_group(group_name)

    ON DELETE CASCADE
);
```

선호그룹: 고객별 선호하는 (음식들의) 그룹을 정리한 테이블입니다. 이 역시 M:N 관계이므로 해당 필드 각각 고객, 음식그룹을 참조하는 외래키가 되고, 고객아이디와 그룹이름의 순서쌍을 기본키로 갖는 테이블이 됩니다.

```
CREATE TABLE customer_contact (
    fk_customer_id VARCHAR2(15) NOT NULL,
    customer_number CHAR(13) NOT NULL,

    PRIMARY KEY (fk_customer_id, customer_number),

    CONSTRAINT CHK_customer_number_type
    CHECK (customer_number LIKE '0-0-0-0-0'),

    FOREIGN KEY (fk_customer_id)
    REFERENCES customer(customer_id)
    ON DELETE CASCADE
);
```

고객연락처: 고객의 전화번호가 다중값 속성이므로, 고객의 기본키를 외래키로 갖고, (Foreign Key, Value) 형태의 기본키로 구성된 새로운 릴레이션으로 변환하였습니다. 해당 릴레이션 역시 전화번호 정보를 삽입할 때, 정해진 형태('000-0000-0000')가 아니면 입력되지 않도록 제약조건을 설정했습니다.

```

CREATE TABLE customer_order (
    fk_customer_id VARCHAR2(15) NOT NULL,
    fk_food_number VARCHAR2(6) NOT NULL,
    ordered_amount NUMBER(2) NOT NULL,

    PRIMARY KEY (fk_customer_id, fk_food_number),

    FOREIGN KEY (fk_customer_id)
    REFERENCES customer(customer_id),

    FOREIGN KEY (fk_food_number)
    REFERENCES food(food_number),

    CONSTRAINT CHK_ordered_amount
    CHECK (ordered_amount > 0)
);

```

주문: 이 관계는 고객이 음식을 주문함으로써 형성되는 관계로 기본적으로 M:N 매핑 카디널리티를 갖습니다. 그러므로 고객의 아이디는 고객 테이블의 기본키를 참조하는 외래키로 하고, 음식 번호는 음식 테이블의 기본키를 참조하는 외래키로 설정하였고, 이 둘을 결합한 키 쌍을 주문 관계의 기본키로 설정하였습니다. 이때, 주문이 성립하기 위해서는 무조건 1개 이상은 주문해야 하므로, 제약조건을 통하여 주문 수량 입력에 대한 조건을 설정했습니다.

```

CREATE TABLE reservation (
    fk_customer_id VARCHAR2(15) NOT NULL,
    fk_restaurant_code VARCHAR2(6) NOT NULL,
    reserved_time DATE NOT NULL,
    companions NUMBER(2),

    CONSTRAINT UNIQUE_time UNIQUE(reserved_time),

    PRIMARY KEY (fk_customer_id, fk_restaurant_code),

    FOREIGN KEY (fk_customer_id)
    REFERENCES customer(customer_id),

    FOREIGN KEY (fk_restaurant_code)
    REFERENCES restaurant(restaurant_code)
);

```

예약: 예약 테이블 역시 고객이 음식점에 방문하는 것을 예약하면서 생성되는 M:N 관계의 테이블이므로, 관련된 테이블의 기본키를 각각 외래키로 하고, 둘의 조합을 예약 테이블의 기본키로 하였습니다. 나머지 속성은 관계가 형성되면서 생기는 속성이므로 예약 테이블의 필드로 추가하였습니다. 예약일시 속성에 대해서는, 날짜 포맷으로 받았습니다. 이와 더불어 예약 시간이 다르다는 조건하에서만 여러 음식점을 예약할 수 있으므로 고유키(Unique Key)로 설정하였습니다.

```
CREATE TABLE "program" (
    program_name VARCHAR2(30) PRIMARY KEY,
    broadcasting_company VARCHAR2(12)
);
```

TV프로그램: TV 프로그램에 대한 정보를 갖는데, 이는 프로그램명과 프로그램을 방영, 제작하는 방송사 칼럼으로 구성됩니다. 프로그램 이름은 유일하므로 해당 필드를 기본키로 설정하였습니다.

```
CREATE TABLE broadcasting (
    fk_program_name VARCHAR2(30) NOT NULL,
    fk_restaurant_code VARCHAR2(6) NOT NULL,
    broadcasted_date DATE NOT NULL,

    PRIMARY KEY(fk_program_name, fk_restaurant_code),

    FOREIGN KEY (fk_program_name)
    REFERENCES "program"(program_name),

    FOREIGN KEY (fk_restaurant_code)
    REFERENCES restaurant(restaurant_code)
);
```

출연: 방송 프로그램에 음식점이 출연함으로써 형성되는 관계 테이블로, M:N 관계입니다. 그러므로 관계에 참여하는 테이블들의 기본키를 출연 테이블의 외래키가 참조하도록 하였고, 외래키들의 쌍(프로그램 이름, 음식점 코드)을 기본키로 설정하였습니다. 방영일시는 DML 문 사용에 용이하도록 DATE 포맷으로 도메인을 설정했습니다.

```
CREATE TABLE watching (
    fk_customer_id VARCHAR2(10) NOT NULL,
    fk_program_name VARCHAR2(30) NOT NULL,

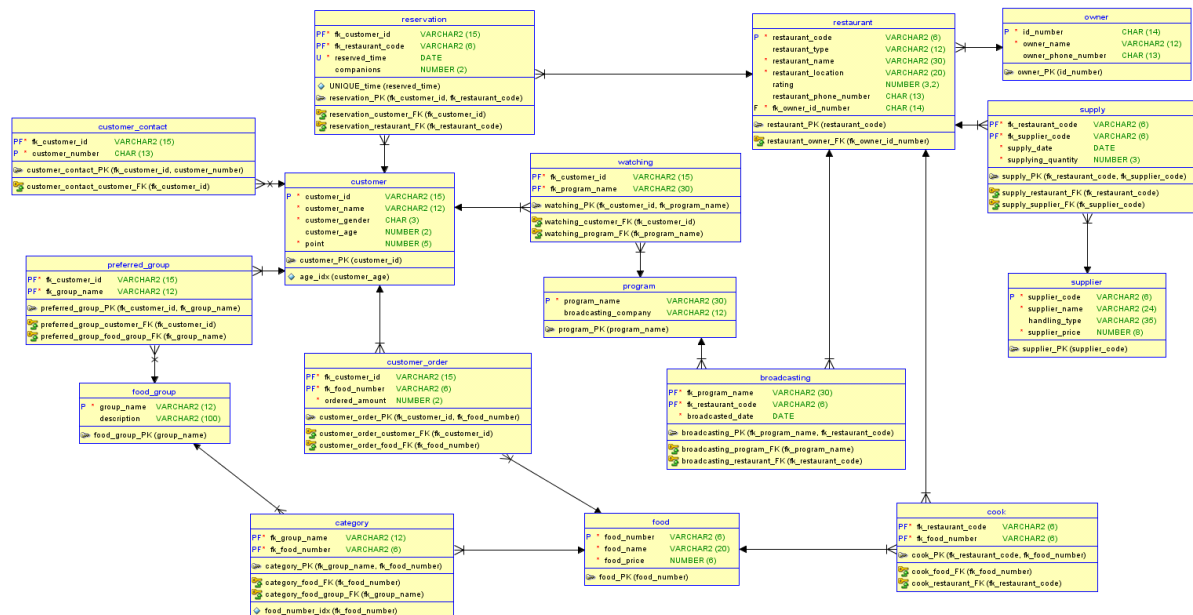
    PRIMARY KEY (fk_customer_id, fk_program_name),

    FOREIGN KEY (fk_customer_id)
    REFERENCES customer(customer_id),

    FOREIGN KEY (fk_program_name)
    REFERENCES "program"(program_name)
);
```

시청: 고객이 해당 프로그램을 시청한 경우, 어떤 프로그램을 시청하였는지를 저장하는 테이블입니다. 기본적으로 M:N 매핑 카디널리티를 가지므로, 관계에 참여하는 테이블들의 기본키를 출연 테이블의 외래키가 참조하도록 하였고, 외래키들의 쌍(고객 아이디, 프로그램 이름)을 기본키로 설정하였습니다.

해당 과정을 통해 생성된 테이블을 관계형 데이터 모델로 표현하면, 다음과 같습니다.



## IV. 구현

### i) SQL DML 문을 이용한 실행

#### 1) 데이터 삽입(INSERT)

각 테이블에 대해 샘플 데이터를 삽입하였고, 이 시스템에서 핵심이 되는 음식점 상호, 음식점 위치, 음식 종류 및 가격, 방송 방영정보, 가게 번호 등은 실제 존재하는 데이터를 가져왔음을 명확히 밝힙니다.

[업체]

```

INSERT INTO supplier VALUES('S-001', '신수', '쌈채소류', 32500);
INSERT INTO supplier VALUES('S-002', '전북농협', '버섯류', 22700);
INSERT INTO supplier VALUES('S-003', '주문진수협', '오징어/문어', 42000);
INSERT INTO supplier VALUES('S-004', '산포수산', '고등어/갈치/꽂치', 24000);
INSERT INTO supplier VALUES('S-005', '다람정육', '계육', 17200);
INSERT INTO supplier VALUES('S-006', '상우정육', '돈육/우육 가공', 58000);
INSERT INTO supplier VALUES('S-007', '정광', '머육가공', 18900);
INSERT INTO supplier VALUES('S-008', '황금양계장', '계육/계란', 16300);
INSERT INTO supplier VALUES('S-009', '향성정미', '쌀 및 기타 곡류', 33600);
INSERT INTO supplier VALUES('S-010', '강원전선농협', '산나물류', 19400);
INSERT INTO supplier VALUES('S-011', '여주일품정미', '쌀 및 기타 곡류', 34200);
INSERT INTO supplier VALUES('S-012', '진미육가공', '계육/돈육 가공', 23000);
    
```

[사장]

페이지가 길 경우,  
Skip 가능합니다.

```

INSERT INTO "owner"(id_number, name) VALUES('840102-1234517', '황연희');
INSERT INTO "owner" VALUES('931107-2534112', '정수진', '010-1011-2233');
INSERT INTO "owner" VALUES('000424-3012345', '최영웅', '010-2650-9477');
INSERT INTO "owner" VALUES('700902-1423991', '신승민', '010-1103-0444');
INSERT INTO "owner" VALUES('650209-2234954', '이윤화', '010-3745-2291');
INSERT INTO "owner" VALUES('950804-1133220', '장민혁', '010-3348-9127');
INSERT INTO "owner" VALUES('801128-1211943', '윤성태', '010-7454-2690');

```

## [음식점]

```

INSERT INTO restaurant VALUES('R-0001', '한식', '갈비명가 서서갈비', '수원 권선구', 4.21, '031-296-9233');
INSERT INTO restaurant VALUES('R-0002', '분식', '돈파스 김동현 가게', '수원 영통구', 4.4, '031-214-3438');
INSERT INTO restaurant VALUES('R-0003', '분식', '애플하우스', '서울 서초구', 4.37, '02-595-1629');
INSERT INTO restaurant VALUES('R-0004', '분식', '명동교차', '서울 중구', 4.39, '02-776-5348');
INSERT INTO restaurant VALUES('R-0005', '일식주점', '면무동 면어집', '수원 장안구', 4.08, '070-8860-9457');
INSERT INTO restaurant VALUES('R-0006', '중식', '목란', '서울 서대문구', 4.27, '02-732-1245');
INSERT INTO restaurant VALUES('R-0007', '일식', '스시미치루', '서울 종로구', 4.67, '02-761-4090');
INSERT INTO restaurant VALUES('R-0008', '분식', '중앙떡볶이', '대구 중구', 4.28, '053-424-7692');
INSERT INTO restaurant VALUES('R-0009', '중식', '리안', '대구 수성구', 4.47, '053-746-0203');
INSERT INTO restaurant VALUES('R-0010', '치킨', '뉴욕통닭', '대구 중구', 4.52, '053-253-0070');
INSERT INTO restaurant VALUES('R-0011', '치킨', '용성통닭', '수원 팔달구', 4.34, '031-242-8226');
INSERT INTO restaurant VALUES('R-0012', '치킨', '진미통닭', '수원 팔달구', 4.31, '031-255-3401');
INSERT INTO restaurant VALUES('R-0013', '한식', '조점례남문피순대', '전주 완산구', 4.31, '063-232-5006');
INSERT INTO restaurant VALUES('R-0014', '중식', '노벨만점', '전주 완산구', 4.24, '063-284-4318');
INSERT INTO restaurant VALUES('R-0015', '한식', '용진집막걸리', '전주 완산구', 3.96, '063-224-8164');
INSERT INTO restaurant VALUES('R-0016', '한식', '호남각', '전주 덕진구', 4.39, '063-278-8150');
INSERT INTO restaurant VALUES('R-0017', '중식', '공화춘', '인천 중구', 4.11, '032-765-0571');
INSERT INTO restaurant VALUES('R-0018', '한식', '해성횃집', '강릉 성남동', 4.29, '033-648-4313');
INSERT INTO restaurant VALUES('R-0019', '한식', '미가곰치국', '강릉 주문진읍', 4.17, '033-661-8944');
INSERT INTO restaurant VALUES('R-0020', '한식', '해운대 가마밀면', '부산 해운대구', 4.54, '051-747-9404');

```

## [납품]

```

INSERT INTO supply VALUES('R-0001', 'S-001', to_date('2020/12/1', 'YYYY/MM/DD'), 50);
INSERT INTO supply VALUES('R-0001', 'S-006', to_date('2020/12/4', 'YYYY/MM/DD'), 100);
INSERT INTO supply VALUES('R-0001', 'S-009', to_date('2020/12/2', 'YYYY/MM/DD'), 140);
INSERT INTO supply VALUES('R-0002', 'S-006', to_date('2020/11/30', 'YYYY/MM/DD'), 20);
INSERT INTO supply VALUES('R-0002', 'S-008', to_date('2020/12/3', 'YYYY/MM/DD'), 15);
INSERT INTO supply VALUES('R-0002', 'S-009', to_date('2020/10/5', 'YYYY/MM/DD'), 30);
INSERT INTO supply VALUES('R-0002', 'S-012', to_date('2020/12/1', 'YYYY/MM/DD'), 10);
INSERT INTO supply VALUES('R-0003', 'S-006', to_date('2020/11/27', 'YYYY/MM/DD'), 20);
INSERT INTO supply VALUES('R-0003', 'S-007', to_date('2020/11/13', 'YYYY/MM/DD'), 30);
INSERT INTO supply VALUES('R-0004', 'S-005', to_date('2020/12/2', 'YYYY/MM/DD'), 10);
INSERT INTO supply VALUES('R-0004', 'S-007', to_date('2020/11/13', 'YYYY/MM/DD'), 20);
INSERT INTO supply VALUES('R-0004', 'S-009', to_date('2020/11/25', 'YYYY/MM/DD'), 33);
INSERT INTO supply VALUES('R-0005', 'S-003', to_date('2020/12/5', 'YYYY/MM/DD'), 15);
INSERT INTO supply VALUES('R-0005', 'S-004', to_date('2020/12/5', 'YYYY/MM/DD'), 10);
INSERT INTO supply VALUES('R-0005', 'S-005', to_date('2020/12/2', 'YYYY/MM/DD'), 10);
INSERT INTO supply VALUES('R-0005', 'S-011', to_date('2020/10/27', 'YYYY/MM/DD'), 40);
INSERT INTO supply VALUES('R-0006', 'S-001', to_date('2020/12/1', 'YYYY/MM/DD'), 35);
INSERT INTO supply VALUES('R-0006', 'S-002', to_date('2020/11/29', 'YYYY/MM/DD'), 20);

```



```

INSERT INTO supply VALUES('R-0006', 'S-003', to_date('2020/12/3', 'YYYY/MM/DD'), 45);
INSERT INTO supply VALUES('R-0006', 'S-005', to_date('2020/12/1', 'YYYY/MM/DD'), 20);
INSERT INTO supply VALUES('R-0006', 'S-006', to_date('2020/11/26', 'YYYY/MM/DD'), 100);
INSERT INTO supply VALUES('R-0006', 'S-007', to_date('2020/9/11', 'YYYY/MM/DD'), 50);
INSERT INTO supply VALUES('R-0006', 'S-009', to_date('2020/9/16', 'YYYY/MM/DD'), 230);
INSERT INTO supply VALUES('R-0006', 'S-012', to_date('2020/12/5', 'YYYY/MM/DD'), 20);
INSERT INTO supply VALUES('R-0007', 'S-004', to_date('2020/12/2', 'YYYY/MM/DD'), 15);
INSERT INTO supply VALUES('R-0007', 'S-009', to_date('2020/9/30', 'YYYY/MM/DD'), 50);
INSERT INTO supply VALUES('R-0008', 'S-003', to_date('2020/12/2', 'YYYY/MM/DD'), 8);
INSERT INTO supply VALUES('R-0008', 'S-007', to_date('2020/10/21', 'YYYY/MM/DD'), 30);
INSERT INTO supply VALUES('R-0008', 'S-008', to_date('2020/12/6', 'YYYY/MM/DD'), 15);
INSERT INTO supply VALUES('R-0008', 'S-009', to_date('2020/10/5', 'YYYY/MM/DD'), 30);
INSERT INTO supply VALUES('R-0009', 'S-003', to_date('2020/12/2', 'YYYY/MM/DD'), 15);
INSERT INTO supply VALUES('R-0009', 'S-004', to_date('2020/11/30', 'YYYY/MM/DD'), 5);
INSERT INTO supply VALUES('R-0009', 'S-005', to_date('2020/12/3', 'YYYY/MM/DD'), 10);
INSERT INTO supply VALUES('R-0009', 'S-011', to_date('2020/11/17', 'YYYY/MM/DD'), 30);
INSERT INTO supply VALUES('R-0010', 'S-005', to_date('2020/11/29', 'YYYY/MM/DD'), 100);
INSERT INTO supply VALUES('R-0011', 'S-008', to_date('2020/12/1', 'YYYY/MM/DD'), 120);
INSERT INTO supply VALUES('R-0012', 'S-008', to_date('2020/12/2', 'YYYY/MM/DD'), 85);
INSERT INTO supply VALUES('R-0012', 'S-012', to_date('2020/12/2', 'YYYY/MM/DD'), 20);
INSERT INTO supply VALUES('R-0013', 'S-006', to_date('2020/11/28', 'YYYY/MM/DD'), 40);
INSERT INTO supply VALUES('R-0013', 'S-009', to_date('2020/11/1', 'YYYY/MM/DD'), 60);
INSERT INTO supply VALUES('R-0013', 'S-011', to_date('2020/11/15', 'YYYY/MM/DD'), 30);
INSERT INTO supply VALUES('R-0014', 'S-002', to_date('2020/12/1', 'YYYY/MM/DD'), 15);
INSERT INTO supply VALUES('R-0014', 'S-003', to_date('2020/11/24', 'YYYY/MM/DD'), 30);
INSERT INTO supply VALUES('R-0014', 'S-006', to_date('2020/11/28', 'YYYY/MM/DD'), 40);
INSERT INTO supply VALUES('R-0014', 'S-008', to_date('2020/11/27', 'YYYY/MM/DD'), 20);
INSERT INTO supply VALUES('R-0014', 'S-009', to_date('2020/12/1', 'YYYY/MM/DD'), 70);
INSERT INTO supply VALUES('R-0014', 'S-010', to_date('2020/12/2', 'YYYY/MM/DD'), 4);
INSERT INTO supply VALUES('R-0015', 'S-001', to_date('2020/11/29', 'YYYY/MM/DD'), 20);
INSERT INTO supply VALUES('R-0015', 'S-002', to_date('2020/12/2', 'YYYY/MM/DD'), 17);
INSERT INTO supply VALUES('R-0015', 'S-004', to_date('2020/12/1', 'YYYY/MM/DD'), 20);
INSERT INTO supply VALUES('R-0015', 'S-005', to_date('2020/11/29', 'YYYY/MM/DD'), 30);
INSERT INTO supply VALUES('R-0015', 'S-006', to_date('2020/12/1', 'YYYY/MM/DD'), 25);
INSERT INTO supply VALUES('R-0015', 'S-007', to_date('2020/11/19', 'YYYY/MM/DD'), 15);

INSERT INTO supply VALUES('R-0015', 'S-009', to_date('2020/11/1', 'YYYY/MM/DD'), 60);
INSERT INTO supply VALUES('R-0015', 'S-010', to_date('2020/11/28', 'YYYY/MM/DD'), 20);
INSERT INTO supply VALUES('R-0016', 'S-001', to_date('2020/11/28', 'YYYY/MM/DD'), 25);
INSERT INTO supply VALUES('R-0016', 'S-002', to_date('2020/12/4', 'YYYY/MM/DD'), 13);
INSERT INTO supply VALUES('R-0016', 'S-004', to_date('2020/12/2', 'YYYY/MM/DD'), 30);
INSERT INTO supply VALUES('R-0016', 'S-005', to_date('2020/11/22', 'YYYY/MM/DD'), 40);
INSERT INTO supply VALUES('R-0016', 'S-006', to_date('2020/12/5', 'YYYY/MM/DD'), 35);
INSERT INTO supply VALUES('R-0016', 'S-007', to_date('2020/11/9', 'YYYY/MM/DD'), 19);
INSERT INTO supply VALUES('R-0016', 'S-009', to_date('2020/11/11', 'YYYY/MM/DD'), 90);
INSERT INTO supply VALUES('R-0016', 'S-010', to_date('2020/11/26', 'YYYY/MM/DD'), 30);

INSERT INTO supply VALUES('R-0017', 'S-002', to_date('2020/11/19', 'YYYY/MM/DD'), 19);
INSERT INTO supply VALUES('R-0017', 'S-003', to_date('2020/11/25', 'YYYY/MM/DD'), 35);
INSERT INTO supply VALUES('R-0017', 'S-006', to_date('2020/11/23', 'YYYY/MM/DD'), 55);

```

```

INSERT INTO supply VALUES('R-0017', 'S-008', to_date('2020/11/30', 'YYYY/MM/DD'), 30);
INSERT INTO supply VALUES('R-0017', 'S-009', to_date('2020/12/3', 'YYYY/MM/DD'), 100);
INSERT INTO supply VALUES('R-0017', 'S-010', to_date('2020/12/4', 'YYYY/MM/DD'), 14);
INSERT INTO supply VALUES('R-0018', 'S-003', to_date('2020/12/1', 'YYYY/MM/DD'), 20);
INSERT INTO supply VALUES('R-0018', 'S-004', to_date('2020/11/30', 'YYYY/MM/DD'), 15);
INSERT INTO supply VALUES('R-0018', 'S-009', to_date('2020/11/5', 'YYYY/MM/DD'), 40);
INSERT INTO supply VALUES('R-0018', 'S-010', to_date('2020/11/28', 'YYYY/MM/DD'), 13);
INSERT INTO supply VALUES('R-0019', 'S-003', to_date('2020/11/29', 'YYYY/MM/DD'), 22);
INSERT INTO supply VALUES('R-0019', 'S-004', to_date('2020/11/30', 'YYYY/MM/DD'), 18);
INSERT INTO supply VALUES('R-0019', 'S-009', to_date('2020/12/1', 'YYYY/MM/DD'), 34);
INSERT INTO supply VALUES('R-0019', 'S-010', to_date('2020/11/28', 'YYYY/MM/DD'), 17);
INSERT INTO supply VALUES('R-0020', 'S-008', to_date('2020/11/29', 'YYYY/MM/DD'), 13);
INSERT INTO supply VALUES('R-0020', 'S-009', to_date('2020/11/1', 'YYYY/MM/DD'), 65);

```

## [음식]

```

INSERT INTO food VALUES('F-001', '돼지갈비', 7900);
INSERT INTO food VALUES('F-002', '소양념갈비', 9900);
INSERT INTO food VALUES('F-003', '소생갈비', 20000);
INSERT INTO food VALUES('F-004', '돈까스', 5500);
INSERT INTO food VALUES('F-005', '떡볶이', 4000);
INSERT INTO food VALUES('F-006', '비빔국수', 9000);
INSERT INTO food VALUES('F-007', '만두', 5000);
INSERT INTO food VALUES('F-008', '연어회', 18000);
INSERT INTO food VALUES('F-009', '짬뽕', 7500);
INSERT INTO food VALUES('F-010', '동파육', 45000);
INSERT INTO food VALUES('F-011', '탕수육', 20000);
INSERT INTO food VALUES('F-012', '짜장면', 6000);
INSERT INTO food VALUES('F-013', '오마카세', 60000);
INSERT INTO food VALUES('F-014', '순대', 3000);
INSERT INTO food VALUES('F-015', '후라이드치킨', 18000);
INSERT INTO food VALUES('F-016', '양념치킨', 19000);
INSERT INTO food VALUES('F-017', '홀어썸', 12000);
INSERT INTO food VALUES('F-018', '보리굴비', 12000);
INSERT INTO food VALUES('F-019', '낙지탕탕이', 15000);
INSERT INTO food VALUES('F-020', '알탕', 10000);
INSERT INTO food VALUES('F-021', '삼숙이탕', 10000);
INSERT INTO food VALUES('F-022', '곰치국', 12000);

```

## [조리]

```

INSERT INTO cook VALUES('R-0001', 'F-001');
INSERT INTO cook VALUES('R-0001', 'F-002');
INSERT INTO cook VALUES('R-0001', 'F-003');
INSERT INTO cook VALUES('R-0002', 'F-004');
INSERT INTO cook VALUES('R-0002', 'F-006');
INSERT INTO cook VALUES('R-0002', 'F-007');
INSERT INTO cook VALUES('R-0003', 'F-005');
INSERT INTO cook VALUES('R-0004', 'F-006');
INSERT INTO cook VALUES('R-0004', 'F-007');
INSERT INTO cook VALUES('R-0005', 'F-008');

```

```

INSERT INTO cook VALUES ('R-0005', 'F-013');
INSERT INTO cook VALUES ('R-0005', 'F-009');
INSERT INTO cook VALUES ('R-0007', 'F-013');
INSERT INTO cook VALUES ('R-0008', 'F-004');
INSERT INTO cook VALUES ('R-0008', 'F-005');
INSERT INTO cook VALUES ('R-0008', 'F-006');
INSERT INTO cook VALUES ('R-0008', 'F-007');
INSERT INTO cook VALUES ('R-0008', 'F-014');
INSERT INTO cook VALUES ('R-0009', 'F-009');
INSERT INTO cook VALUES ('R-0009', 'F-011');
INSERT INTO cook VALUES ('R-0009', 'F-012');
INSERT INTO cook VALUES ('R-0010', 'F-015');
INSERT INTO cook VALUES ('R-0010', 'F-016');
INSERT INTO cook VALUES ('R-0011', 'F-015');
INSERT INTO cook VALUES ('R-0011', 'F-016');
INSERT INTO cook VALUES ('R-0012', 'F-015');
INSERT INTO cook VALUES ('R-0012', 'F-016');
INSERT INTO cook VALUES ('R-0013', 'F-014');
INSERT INTO cook VALUES ('R-0014', 'F-009');
INSERT INTO cook VALUES ('R-0014', 'F-011');
INSERT INTO cook VALUES ('R-0014', 'F-012');
INSERT INTO cook VALUES ('R-0015', 'F-017');
INSERT INTO cook VALUES ('R-0015', 'F-018');
INSERT INTO cook VALUES ('R-0015', 'F-019');
INSERT INTO cook VALUES ('R-0016', 'F-002');
INSERT INTO cook VALUES ('R-0016', 'F-003');
INSERT INTO cook VALUES ('R-0016', 'F-017');
INSERT INTO cook VALUES ('R-0016', 'F-018');
INSERT INTO cook VALUES ('R-0017', 'F-009');
INSERT INTO cook VALUES ('R-0017', 'F-010');
INSERT INTO cook VALUES ('R-0017', 'F-011');
INSERT INTO cook VALUES ('R-0017', 'F-012');
INSERT INTO cook VALUES ('R-0018', 'F-020');
INSERT INTO cook VALUES ('R-0018', 'F-021');
INSERT INTO cook VALUES ('R-0019', 'F-020');
INSERT INTO cook VALUES ('R-0019', 'F-021');
INSERT INTO cook VALUES ('R-0019', 'F-022');
INSERT INTO cook VALUES ('R-0020', 'F-006');
INSERT INTO cook VALUES ('R-0020', 'F-007');

```

#### [음식그룹]

```

INSERT INTO food_group VALUES ('한식', '한국 음식 (순댓국, 갈비, 회, 한정식 등)을 포함합니다. ');
INSERT INTO food_group VALUES ('중식', '중국 음식과 중화풍 요리 (탕수육, 짜장, 짬뽕 등)을 포함합니다. ');
INSERT INTO food_group VALUES ('일식', '일본풍 요리 (사시미, 오마카세 등)을 포함합니다. ');
INSERT INTO food_group VALUES ('분식', '간편하게 즐길 수 있는 떡볶이, 순대, 만두 등을 포함합니다. ');
INSERT INTO food_group VALUES ('치킨', '여러 종류의 모든 치킨을 포함하는 그룹입니다. ');

```



## [분류]

```
INSERT INTO "category" VALUES('한식', 'F-001');
INSERT INTO "category" VALUES('한식', 'F-002');
INSERT INTO "category" VALUES('한식', 'F-003');
INSERT INTO "category" VALUES('분식', 'F-004');
INSERT INTO "category" VALUES('분식', 'F-005');
INSERT INTO "category" VALUES('분식', 'F-006');
INSERT INTO "category" VALUES('한식', 'F-006');
INSERT INTO "category" VALUES('분식', 'F-007');
INSERT INTO "category" VALUES('한식', 'F-008');
INSERT INTO "category" VALUES('일식', 'F-008');
INSERT INTO "category" VALUES('중식', 'F-009');
INSERT INTO "category" VALUES('중식', 'F-010');
INSERT INTO "category" VALUES('중식', 'F-011');
INSERT INTO "category" VALUES('중식', 'F-012');
INSERT INTO "category" VALUES('일식', 'F-013');
INSERT INTO "category" VALUES('한식', 'F-014');
INSERT INTO "category" VALUES('분식', 'F-014');
INSERT INTO "category" VALUES('치킨', 'F-015');
INSERT INTO "category" VALUES('치킨', 'F-016');
INSERT INTO "category" VALUES('한식', 'F-017');
INSERT INTO "category" VALUES('한식', 'F-018');
INSERT INTO "category" VALUES('한식', 'F-019');
INSERT INTO "category" VALUES('한식', 'F-020');
INSERT INTO "category" VALUES('한식', 'F-021');
```

## [고객]

```
INSERT INTO customer VALUES('hero00', '최영웅', '남', 21, 3500);
INSERT INTO customer VALUES('cyjeon95', '전창윤', '남', 26, 750);
INSERT INTO customer VALUES('kangyunhee3843', '강윤희', '여', 23, 1200);
INSERT INTO customer VALUES('jung10', '정일영', '남', 31, 800);
INSERT INTO customer VALUES('seowool972', '전서우', '여', 49, 200);
INSERT INTO customer VALUES('cym1970', '최용문', '남', 51, 340);
INSERT INTO customer VALUES('doyoung1192', '이도영', '여', 25, 790);
INSERT INTO customer VALUES('qwe1293', '신소연', '여', 24, 1000);
INSERT INTO customer VALUES('easy0', '이지영', '여', 20, 820);
INSERT INTO customer VALUES('kyt0901', '김용태', '남', 17, 750);
INSERT INTO customer VALUES('jsh0407', '전선희', '여', 19, 880);
INSERT INTO customer VALUES('chris1024', '박영수', '남', 23, 1000);
```

## [고객연락처]

```
INSERT INTO customer_contact VALUES('hero00', '010-2650-9477');
INSERT INTO customer_contact VALUES('hero00', '010-4462-9477');
INSERT INTO customer_contact VALUES('cyjeon95', '010-3392-1023');
INSERT INTO customer_contact VALUES('kangyunhee3843', '010-5320-3843');
INSERT INTO customer_contact VALUES('jung10', '010-9023-1140');
INSERT INTO customer_contact VALUES('seowool972', '010-5341-9477');
INSERT INTO customer_contact VALUES('cym1970', '010-6213-5440');
INSERT INTO customer_contact VALUES('doyoung1192', '010-7430-1192');
INSERT INTO customer_contact VALUES('qwe1293', '010-4522-1293');
INSERT INTO customer_contact VALUES('easy0', '010-2346-9981');
INSERT INTO customer_contact VALUES('kyt0901', '010-6389-3874');
INSERT INTO customer_contact VALUES('jsh0407', '010-9392-8902');
INSERT INTO customer_contact VALUES('chris1024', '010-8895-1024');
```

### [선호그룹]

```
INSERT INTO preferred_group VALUES('hero00', '일식');
INSERT INTO preferred_group VALUES('hero00', '분식');
INSERT INTO preferred_group VALUES('hero00', '치킨');
INSERT INTO preferred_group VALUES('cyjeon95', '한식');
INSERT INTO preferred_group VALUES('cyjeon95', '중식');
INSERT INTO preferred_group VALUES('kangyunhee3843', '일식');
INSERT INTO preferred_group VALUES('kangyunhee3843', '분식');
INSERT INTO preferred_group VALUES('jung10', '치킨');
INSERT INTO preferred_group VALUES('seowool972', '한식');
INSERT INTO preferred_group VALUES('cym1970', '한식');
INSERT INTO preferred_group VALUES('cym1970', '치킨');
INSERT INTO preferred_group VALUES('cym1970', '일식');
INSERT INTO preferred_group VALUES('doyoung1192', '중식');
INSERT INTO preferred_group VALUES('qwe1293', '한식');
INSERT INTO preferred_group VALUES('qwe1293', '중식');
INSERT INTO preferred_group VALUES('easy0', '치킨');
INSERT INTO preferred_group VALUES('easy0', '분식');
INSERT INTO preferred_group VALUES('kyt0901', '중식');
INSERT INTO preferred_group VALUES('kyt0901', '치킨');
INSERT INTO preferred_group VALUES('jsh0407', '치킨');
INSERT INTO preferred_group VALUES('jsh0407', '중식');
INSERT INTO preferred_group VALUES('chris1024', '한식');
```

### [주문]

```
INSERT INTO customer_order VALUES('hero00', 'F-001', 3);
INSERT INTO customer_order VALUES('hero00', 'F-004', 1);
INSERT INTO customer_order VALUES('hero00', 'F-016', 1);
INSERT INTO customer_order VALUES('cyjeon95', 'F-009', 1);
INSERT INTO customer_order VALUES('cyjeon95', 'F-011', 1);
INSERT INTO customer_order VALUES('kangyunhee3843', 'F-005', 1);
INSERT INTO customer_order VALUES('kangyunhee3843', 'F-014', 1);
INSERT INTO customer_order VALUES('jung10', 'F-020', 2);
INSERT INTO customer_order VALUES('seowool972', 'F-006', 1);
INSERT INTO customer_order VALUES('seowool972', 'F-015', 1);
INSERT INTO customer_order VALUES('cym1970', 'F-022', 1);
INSERT INTO customer_order VALUES('easy0', 'F-005', 2);
INSERT INTO customer_order VALUES('kyt0901', 'F-012', 1);
INSERT INTO customer_order VALUES('chris1024', 'F-016', 1);
```

### [예약]

```
INSERT INTO reservation VALUES('hero00', 'R-0001', to_date('2020/12/8 13:45', 'YYYY/MM/DD HH24:MI'), 3);
INSERT INTO reservation VALUES('hero00', 'R-0011', to_date('2020/12/7 18:30', 'YYYY/MM/DD HH24:MI'), 2);
INSERT INTO reservation VALUES('kangyunhee3843', 'R-0007', to_date('2020/12/10 17:20', 'YYYY/MM/DD HH24:MI'), 6);
INSERT INTO reservation VALUES('jung10', 'R-0017', to_date('2020/12/6 12:20', 'YYYY/MM/DD HH24:MI'), 4);
INSERT INTO reservation VALUES('cym1970', 'R-0019', to_date('2020/12/11 11:00', 'YYYY/MM/DD HH24:MI'), 8);
INSERT INTO reservation VALUES('jsh0407', 'R-0001', to_date('2020/12/8 18:00', 'YYYY/MM/DD HH24:MI'), 4);
INSERT INTO reservation VALUES('chris1024', 'R-0016', to_date('2020/12/9 16:20', 'YYYY/MM/DD HH24:MI'), 5);
INSERT INTO reservation VALUES('hero00', 'R-0006', to_date('2020/12/24 17:30', 'YYYY/MM/DD HH24:MI'), 2);
```

## [TV프로그램]

```
INSERT INTO "program" VALUES('모닝와이드', 'SBS');
INSERT INTO "program" VALUES('2TV 생생정보', 'KBS');
INSERT INTO "program" VALUES('백종원의 3대천왕', 'SBS');
INSERT INTO "program" VALUES('알쓸신잡', 'tvN');
INSERT INTO "program" VALUES('수요미식회', 'Olive');
INSERT INTO "program" VALUES('전지적 참견시점', 'MBC');
```

## [시청]

```
INSERT INTO watching VALUES('hero00', '2TV 생생정보');
INSERT INTO watching VALUES('hero00', '백종원의 3대천왕');
INSERT INTO watching VALUES('hero00', '전지적 참견시점');
INSERT INTO watching VALUES('cyjeon95', '백종원의 3대천왕');
INSERT INTO watching VALUES('cyjeon95', '수요미식회');
INSERT INTO watching VALUES('kangyunhee3843', '알쓸신잡');
INSERT INTO watching VALUES('kangyunhee3843', '전지적 참견시점');
INSERT INTO watching VALUES('jung10', '2TV 생생정보');
INSERT INTO watching VALUES('jung10', '모닝와이드');
INSERT INTO watching VALUES('seowool1972', '전지적 참견시점');
INSERT INTO watching VALUES('seowool1972', '2TV 생생정보');
INSERT INTO watching VALUES('doyoung1192', '모닝와이드');
INSERT INTO watching VALUES('doyoung1192', '알쓸신잡');
INSERT INTO watching VALUES('qwe1293', '수요미식회');
INSERT INTO watching VALUES('easy0', '알쓸신잡');
INSERT INTO watching VALUES('kyt0901', '백종원의 3대천왕');
INSERT INTO watching VALUES('kyt0901', '전지적 참견시점');
INSERT INTO watching VALUES('chris1024', '백종원의 3대천왕');
INSERT INTO watching VALUES('chris1024', '수요미식회');
INSERT INTO watching VALUES('chris1024', '전지적 참견시점');
```

## [출연]

```
INSERT INTO broadcasting VALUES('수요미식회', 'R-0003', to_date('2018/1/17', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('수요미식회', 'R-0004', to_date('2015/1/28', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('2TV 생생정보', 'R-0006', to_date('2015/9/10', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('전지적 참견시점', 'R-0008', to_date('2020/2/22', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('모닝와이드', 'R-0010', to_date('2013/3/08', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('백종원의 3대천왕', 'R-0010', to_date('2015/12/11', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('모닝와이드', 'R-0011', to_date('2015/5/18', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('2TV 생생정보', 'R-0011', to_date('2019/4/29', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('2TV 생생정보', 'R-0012', to_date('2019/4/8', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('모닝와이드', 'R-0013', to_date('2013/12/6', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('2TV 생생정보', 'R-0013', to_date('2016/3/14', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('백종원의 3대천왕', 'R-0014', to_date('2015/12/4', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('2TV 생생정보', 'R-0014', to_date('2019/5/10', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('모닝와이드', 'R-0015', to_date('2013/12/6', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('수요미식회', 'R-0015', to_date('2015/9/16', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('백종원의 3대천왕', 'R-0015', to_date('2017/1/7', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('알쓸신잡', 'R-0015', to_date('2017/7/21', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('2TV 생생정보', 'R-0017', to_date('2017/8/17', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('백종원의 3대천왕', 'R-0018', to_date('2016/7/9', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('2TV 생생정보', 'R-0018', to_date('2017/5/30', 'YYYY/MM/DD'));
INSERT INTO broadcasting VALUES('모닝와이드', 'R-0020', to_date('2013/7/26', 'YYYY/MM/DD'));
```

## 2) 데이터 검색(SELECT)

- 2020년 11월에 재료를 납품받은 음식점의 이름과 납품일, 납품받은 종류를 날짜 기준 오름차순으로 정렬하여 검색하라. (날짜가 같으면 음식점 이름 기준으로 오름차순 정렬한다.)

```
select restaurant_name, supply_date, handling_type from restaurant, supply, supplier
where fk_restaurant_code = restaurant_code and
fk_supplier_code = supplier_code and
supply_date between '2020-11-01' and '2020-11-30'
order by supply_date, restaurant_name ASC;
```

RESTAURANT_NAME	SUPPLY_DATE	HANDLING_TYPE	노벨반점	2020/11/28 00:00	돈육/우육 가공
용진집막걸리	2020/11/01 00:00	쌀 및 기타 곡류	RESTAURANT_NAME	SUPPLY_DATE	HANDLING_TYPE
조점레남문피순대	2020/11/01 00:00	쌀 및 기타 곡류			
해운대 가야밀면	2020/11/01 00:00	쌀 및 기타 곡류	미가공치국	2020/11/28 00:00	산나물류
해성회집	2020/11/05 00:00	쌀 및 기타 곡류	용진집막걸리	2020/11/28 00:00	산나물류
호남각	2020/11/09 00:00	머육가공	조점레남문피순대	2020/11/28 00:00	돈육/우육 가공
호남각	2020/11/11 00:00	쌀 및 기타 곡류	해성회집	2020/11/28 00:00	산나물류
명동교자	2020/11/13 00:00	머육가공	호남각	2020/11/28 00:00	쌈채소류
애플하우스	2020/11/13 00:00	머육가공	뉴욕통닭	2020/11/29 00:00	계육
조점레남문피순대	2020/11/15 00:00	쌀 및 기타 곡류	묵란	2020/11/29 00:00	버섯류
리안	2020/11/17 00:00	쌀 및 기타 곡류	미가공치국	2020/11/29 00:00	오징어/문어
공화춘	2020/11/19 00:00	버섯류	용진집막걸리	2020/11/29 00:00	계육
RESTAURANT_NAME	SUPPLY_DATE	HANDLING_TYPE	용진집막걸리	2020/11/29 00:00	쌈채소류
용진집막걸리	2020/11/19 00:00	머육가공	해운대 가야밀면	2020/11/29 00:00	계육/계란
호남각	2020/11/22 00:00	계육	RESTAURANT_NAME	SUPPLY_DATE	HANDLING_TYPE
공화춘	2020/11/23 00:00	돈육/우육 가공	공화춘	2020/11/30 00:00	계육/계란
노벨반점	2020/11/24 00:00	오징어/문어	돈까스 김동현 가게	2020/11/30 00:00	돈육/우육 가공
공화춘	2020/11/25 00:00	오징어/문어	리안	2020/11/30 00:00	고등어/갈치/꽂치
명동교자	2020/11/25 00:00	쌀 및 기타 곡류	미가공치국	2020/11/30 00:00	고등어/갈치/꽂치
묵란	2020/11/26 00:00	돈육/우육 가공	해성회집	2020/11/30 00:00	고등어/갈치/꽂치
호남각	2020/11/26 00:00	산나물류			
노벨반점	2020/11/27 00:00	계육/계란			
애플하우스	2020/11/27 00:00	돈육/우육 가공			

36개 행이 선택되었습니다.

- 사장은 평소에 재료를 공급받을 때, 재료를 공급받는데 필요한 비용에 대해 궁금해했다고 한다. 사장이 운영하는 가게별 재료에 대한 총비용(단위: 만원)을 검색하라. (1차적으로 사장의 이름에 대해 오름차순으로 정렬 후, 사장이 관리하는 음식점이 여러 곳이면 2차적으로 음식점 이름에 대해 오름차순 정렬을 수행한다.)

```
select "owner".name, restaurant_name, sum(supplier_price * supplying_quantity)/10000 AS total_price
from "owner", restaurant, supply, supplier
where fk_owner_id_number = id_number and
fk_restaurant_code = restaurant_code and
fk_supplier_code = supplier_code
group by "owner".name, restaurant_name
order by "owner".name, restaurant_name ASC;
```

NAME	RESTAURANT_NAME	TOTAL_PRICE
신승민	뉴욕통닭	172
신승민	명동교자	165.88
윤성태	미가곰치국	282.82
윤성태	중앙떡볶이	215.55
윤성태	진미통닭	184.55
이윤희	돈까스 김동현 가게	264.25
이윤희	용성통닭	195.6
이윤희	해운대 가야밀면	239.59
장민혁	연무동 연어집	241
장민혁	용진집막걸리	616.94
정수진	노벨반점	667.61

NAME	RESTAURANT_NAME	TOTAL_PRICE
정수진	애플하우스	172.7
정수진	해성횬집	279.62
최영웅	공화춘	921.19
최영웅	리안	194.8
최영웅	스시미치루	204
황연희	갈비명가 서서갈비	1212.9
황연희	목란	1875.85
황연희	조점레남문피순대	536.2
황연희	호남각	851.07

20개 행이 선택되었습니다.

- 방송 출연을 1회도 하지 않은 음식점의 이름을 검색하라.

```
select restaurant_name from restaurant
where restaurant_code not in (select fk_restaurant_code from broadcasting);
```

RESTAURANT\_NAME

연무동 연어집  
호남각  
리안  
갈비명가 서서갈비  
미가곰치국  
돈까스 김동현 가게  
스시미치루

7개 행이 선택되었습니다.

- 맛집을 선정하는 TV프로그램이 있다고 가정하자. 해당 프로그램에서는 평점이 5점 만점 중 4.3 이상인 음식점을 ‘찐 맛집!’이라고 한다고 하자. ‘찐 맛집’의 이름과 위치, 실제 평점을 검색하라. (평점이 높은 음식점이 상위 결과로 나오도록 정렬 기준을 설정하라.)

```
select restaurant_name, restaurant_location, rating from restaurant
where rating >= 4.3
order by rating DESC;
```

RESTAURANT_NAME	RESTAURANT_LOCATION	RATING
스시미치루	서울 종로구	4.67
해운대 가야밀면	부산 해운대구	4.54
뉴욕통닭	대구 중구	4.52
리안	대구 수성구	4.47
돈까스 김동현 가게	수원 영통구	4.4
호남각	전주 덕진구	4.39
명동교자	서울 중구	4.39
애플하우스	서울 서초구	4.37
용성통닭	수원 팔달구	4.34
진미통닭	수원 팔달구	4.31
조점레남문피순대	전주 완산구	4.31

11개 행이 선택되었습니다.

## V. 기타

### i) 인덱스

```
CREATE INDEX food_number_idx
ON "category"(fk_food_number);
```

‘분류’ 테이블의 음식번호 외래키에 대한 인덱스 추가

```
CREATE INDEX age_idx
ON customer(customer_age);
```

‘고객’ 테이블의 나이 필드에 대한 인덱스 추가

### ii) 데이터 출력포맷 변경

```
SELECT VALUE FROM NLS_SESSION_PARAMETERS WHERE PARAMETER = 'NLS_DATE_FORMAT';
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY/MM/DD HH24:MI';
/* 기본 날짜 출력방식 변경 */
```

기본적으로 제공되는 DATE에 대한 데이터 포맷은 RR/MM/DD (예를 들면, 2020년 11월 1일이면 20/11/01)인데, 좀 더 편하게 보고 이해할 수 있도록 날짜(또는 시간)에 대한 출력 포맷을 해당 코드를 이용하여 바꾸었습니다.