

2020-2학기 데이터베이스 Term Project

# 배달 앱 데이터베이스 구현

광고홍보학과

2017113427 이명성

# 목차

- 주제 설명
- 주제 선정 이유
- 사용자 정의, 요구사항 (~11/12)
- E-R 다이어그램, 관계형 데이터베이스 스키마 테이블(~11/26)
- 테이블 생성, 사용자별 작업 처리 SQL문 (~12/14)

# 주제 설명

- 최근 스마트폰에 '배달의민족', '요기요' 등 **배달 서비스 어플리케이션**을 설치해 주문을 하는 사용자 수가 크게 늘고 있다. 최근의 **배달 앱**은 더 이상 직접 배달 서비스에 한정된 것이 아니라, 방문포장 대행과 생필품을 판매.배달하는 자체적인 마트의 기능까지 갖추는 등 기능이 복잡해지고 있으며, **관계된 이해관계자도 늘어나고 있다**. 시스템에 접근할 수 있는 사용자는 **관리자 외에도** 크게는 배달을 맡은 **배달원**, 배달앱에 업체를 등록한 **식당 주인**, 배달앱을 통해 원하는 가게를 검색하고 음식을 구매하는 **고객**이 있고 그 외에도 배달앱 **CS 부서 직원** 등이 필요시 해당 시스템에 접근할 수 있다.

# 주제 선정 이유

- 평소 배달 앱을 많이 사용하는 사람 중 한 사람으로서 문득 배달 앱에 대해 생각해보니 **굉장히 많은 이들이 사용자로 얹혀 있는 구조**라는 것을 깨달았다.  
그래서 많은 사용자들이 접근하고 있는 **배달 앱의 시스템**은 어떠할지 **도식화해보고** **싶다는** 생각이 들었고, **배달 앱의 사용자들을 위한 데이터베이스**를 구성해보고 **싶다는** 생각이 들었다.

# 사용자 정의 (~11/12)

- 1) **고객**: 배달 앱을 이용하여 **주문**을 하는 고객. 데이터베이스 사용자 유형 중 **최종 사용자**에 해당한다.
- 2) **점주**: 배달 앱 상에 가게를 등록하여 고객들이 배달 앱 서비스를 통해 해당 가게에 주문을 할 수 있도록 한다. 고객의 주문을 위해 메뉴, **리뷰 이벤트** 등을 제공한다. 데이터베이스 사용자 유형 중 **최종 사용자**에 해당한다.
- 3) **배달원**: 점주가 만든 음식을 고객에게 가져다 주는 사용자. 데이터베이스 사용자 유형 중 **최종 사용자**에 해당한다.
- 4) **CS부서**: 배달이나 주문 과정에서 **문제가 생겼을 때**에 고객들에게 도움을 주는 부서이다. 데이터베이스 사용자 유형 중 **최종 사용자**에 해당한다.

사용자	요구사항	통합할 내용
고객	<p>1) 배달앱에 회원가입하려면 <u>고객아이디, 고객이름, 주소, 전화번호</u>를 입력해야 한다.</p> <p>2) 가입한 회원에게는 <u>등급과 쿠폰</u>이 부여된다.</p> <p>3) 등급은 silver, gold, vip 3종류이며 처음 가입하면 silver 등급을 부여받는다.</p> <p>4) 쿠폰은 각 고객 당 1개가 필수적으로 제공된다. 고객은 한 개의 쿠폰만 가질 수 있으며 한 개의 쿠폰은 한 명의 고객에게만 <b>제공</b>된다.</p> <p>5) 쿠폰은 <u>쿠폰번호, 할인가격, 유효기간</u> 정보를 유지하며 쿠폰번호로 식별할 수 있다.</p> <p>6) 고객은 <u>고객아이디</u>로 식별한다.</p> <p>7) 메뉴는 <u>메뉴이름</u>으로 식별한다.</p> <p>8) 고객은 메뉴가 제공하는 <u>메뉴이름, 가격, 제조가게명</u>을 확인하고 메뉴를 <b>주문</b>한다.</p> <p>9) 고객이 메뉴를 주문하면 주문에 대한 <u>주문번호, 주문고객(ID), 주문 메뉴, 주문수량, 주문일자</u> 정보를 유지해야 한다.</p>	<p>리뷰(작성자ID) -&gt; 고객(고객아이디)</p> <p>메뉴(제조가게명) -&gt; 가게(가게이름)</p> <p>고객(보유쿠폰) -&gt; 쿠폰(쿠폰번호)</p> <p>주문(주문메뉴) -&gt; 메뉴(메뉴이름)</p> <p>찜한가게(가게명) -&gt; 가게(가게이름)</p> <p>주문(주문고객) -&gt; 고객(고객아이디)</p> <p>선물(송신자ID, 수신자ID)-&gt;</p> <p>고객(고객아이디)</p> <p>찜(고객아이디) -&gt; 고객(고객아이디)</p>

사용자	요구사항	통합할 내용
고객	<p>10) 고객은 여러 메뉴를 주문할 수 있고, 하나의 메뉴를 여러 고객이 주문할 수 있다.</p> <p>11) 음식을 받은 고객은 맛있게 음식을 먹고 리뷰를 작성할 수 있다.</p> <p>12) 고객은 리뷰를 여러 개 작성할 수 있고, 리뷰 하나는 한 명의 고객만 <b>작성</b>할 수 있다.</p> <p>13) 리뷰에 대한 <u>리뷰번호, 작성자ID, 가게명, 리뷰내용, 작성일자, 별점</u> 정보를 유지한다.</p> <p>14) 고객은 다른 고객에게 <b>선물</b>이 가능하다.</p> <p>15) 선물은 5000원과 10000원 두 종류의 정액권으로 줄 수 있고 선물을 주고받으면 <u>송신자ID, 발신자ID, 가격, 제공일자</u> 정보를 유지한다.</p> <p>16) 고객은 자신이 좋아하는 가게들을 '<b>찜</b>'하여 '찜한가게'로 저장할 수 있다. (*종속적 관계, *다중값) 찜한가게는 <u>고객아이디, 가게명</u>의 정보를 가진다.</p> <p>17) 고객은 처음 등록할 때 자신이 좋아하는 <b>최향</b>의 음식을 2개 이상 입력한다. 예를 들면 이런 식이다. (초밥, 우동), (카레, 돈까스, 샐러드) (*다중값)</p>	<p>리뷰(작성자ID) -&gt; 고객(고객아이디)</p> <p>메뉴(제조가게명) -&gt; 가게(가게이름)</p> <p>고객(보유쿠폰) -&gt; 쿠폰(쿠폰번호)</p> <p>주문(주문메뉴) -&gt; 메뉴(메뉴이름)</p> <p>찜한가게(가게명) -&gt; 가게(가게이름)</p> <p>주문(주문고객) -&gt; 고객(고객아이디)</p> <p>선물(송신자ID, 수신자ID)-&gt;</p> <p>고객(고객아이디)</p> <p>찜(고객아이디) -&gt; 고객(고객아이디)</p>

사용자	요구사항	통합할 내용
점주	<p>1) 배달앱에 가게를 등록하려면 <u>가게이름, 전화번호, 가게주소</u>를 입력해야 한다.</p> <p>2) 가게는 <u>가게이름</u>으로 식별한다.</p> <p>3) 한 가게는 여러 메뉴를 <b>제조</b>할 수 있지만, 각 메뉴는 한 가게에서 제조할 수 있다.</p> <p>4) 가게주소와 각 고객의 주소와의 거리차이를 통해 가게는 고객마다 다른 '<u>거리</u>'를 가진다.</p> <p>5) 고객이 메뉴를 <b>주문</b>하면, 가게는 배달원에게 배달<b>신청</b>을 한다.</p> <p>6) 배달원은 여러 개의 가게의 신청을 받을 수 있지만, 가게는 한 명의 배달원에게만 신청을 넣을 수 있다.</p>	



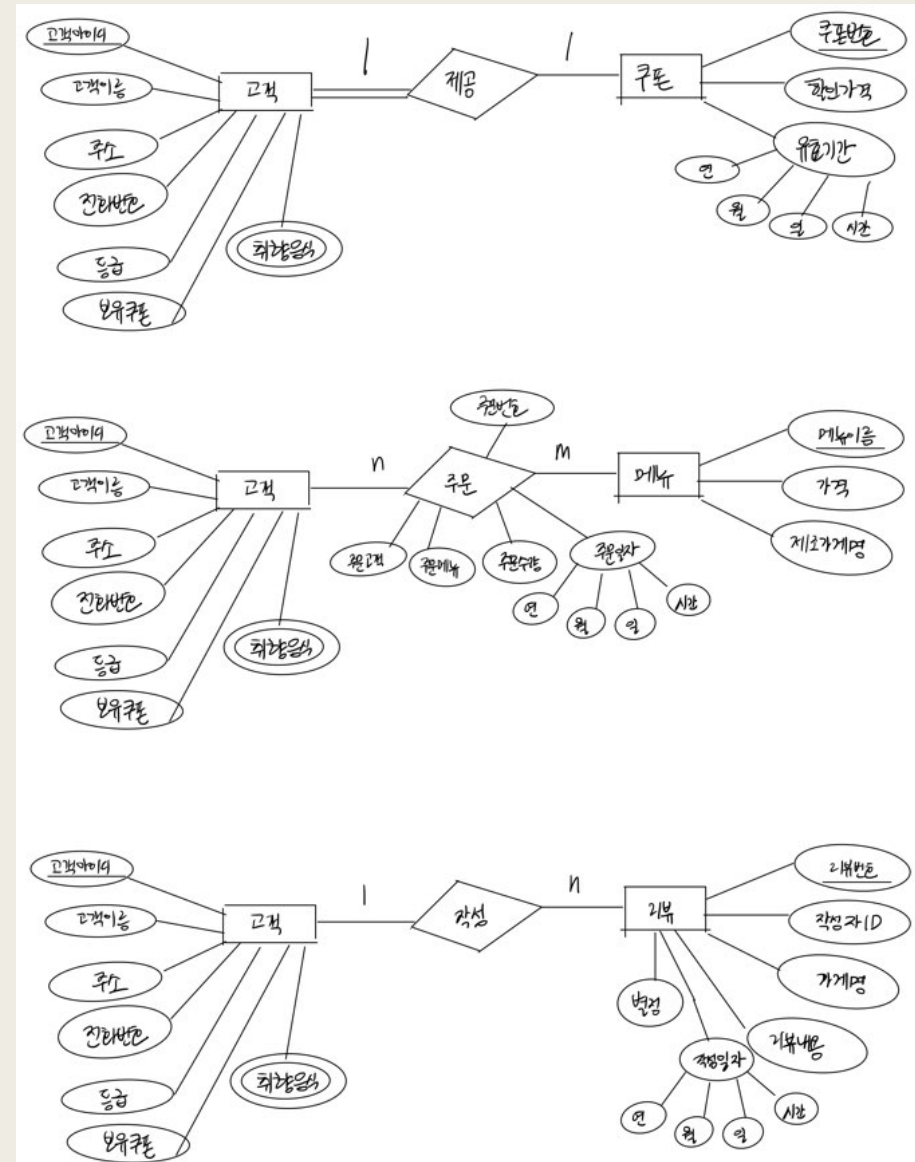
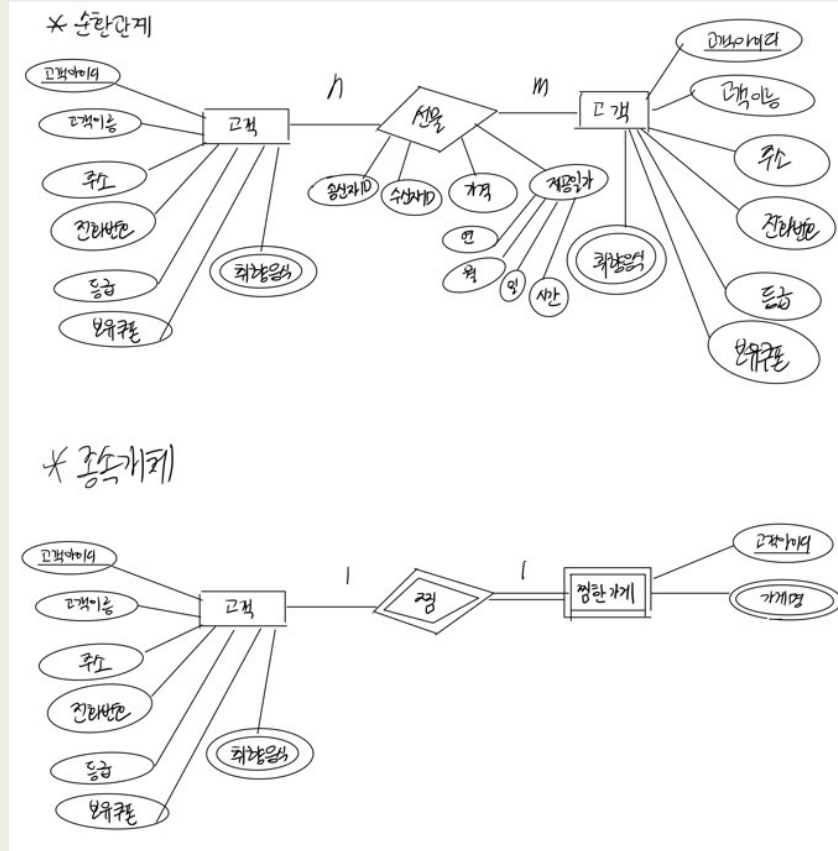
사용자	요구사항	통합할 내용
배달원	<p>1) 배달원은 배달앱에 <u>아이디, 이름, 전화번호</u> 정보를 입력해 등록한다.</p> <p>2) 배달원은 아이디로 식별한다.</p> <p>3) 가게가 배달 <b>신청</b>을 하면 <u>신청우선순위, 가게명, 배달원, 주문번호</u> 정보가 적히고, 우선순위가 높은 가게부터 배달원이 방문한다.</p> <p>4) 배달원은 고객에게 <b>배달</b>한다. 배달은 배달원의 <u>주문고객, 배달지, 출발시간, 출발위치, 배달원, 가게명, 주문번호</u> 정보를 유지한다.</p> <p>5) '출발위치'는 배달하는 '가게주소'와 같고 고객의 주소 정보와의 '거리' 정보를 통해 '<u>예상도착시간</u>' 정보가 주어진다.</p>	<p>신청(가게명) -&gt; 가게(가게이름)</p> <p>신청(배달원) -&gt; 배달원(아이디)</p> <p>배달(출발위치) -&gt; 가게(주소)</p> <p>배달(주문고객) -&gt; 주문(주문고객)</p> <p>배달(배달원) -&gt; 신청(배달원)</p> <p>배달(가게명) -&gt; 신청(가게명)</p> <p>배달(배달지) -&gt; 고객(주소)</p>

사용자	요구사항	통합할 내용
CS부서	<p>1) CS부서는 <u>사안</u>과 사안에 따른 <u>담당자 이름</u> 정보를 유지한다.</p> <p>2) 고객이 CS부서에 <b>불만제기</b>를 하면 불만제기에는 '<u>고객아이디</u>', '<u>불만사항</u>', '<u>불만제기일자</u>', '<u>담당자</u>' 정보를 유지한다.</p> <p>3) 고객은 여러 불만사항을 제기할 수 있지만, 한 개의 불만사항은 한명의 고객에 의해서만 제기된다.</p>	<p>불만제기(불만사항) -&gt; CS부서(사안)</p> <p>불만제기(고객아이디) -&gt; 고객(고객아이디)</p> <p>불만제기(담당자) -&gt; CS부서(담당자)</p>

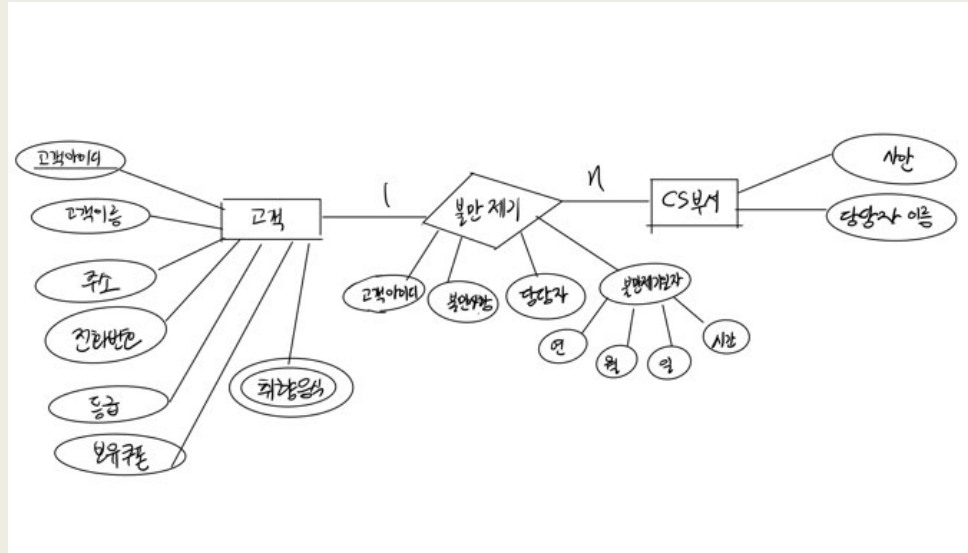
### \* 데이터 모델을 만들기 위한 조건

- 같은 이름을 가진 메뉴는 없다!
- 같은 이름을 가진 가게는 없다!

# E-R 다이어그램 (~11/26)



# E-R 다이어그램 (~11/26)



총 개체 수: **9개** (고객, 가게, 배달원, CS부서, 리뷰, 메뉴, 쿠폰, 찜한가게 (종속개체), 취향음식(종속개체))

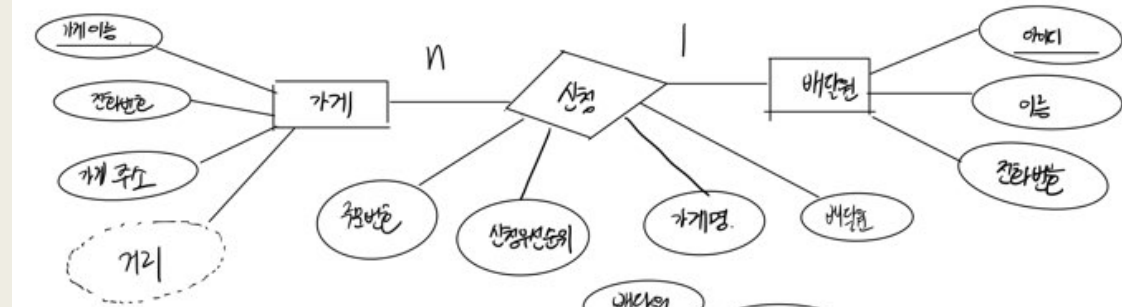
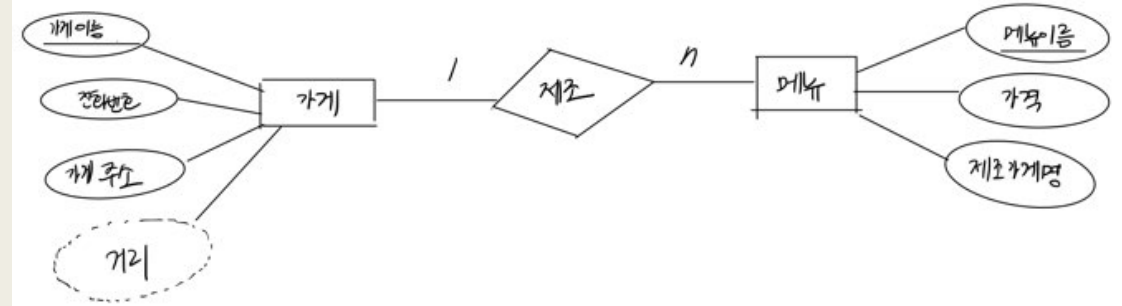
총 관계 수: **10개** (제조, 주문, 신청, 배달, 불만제기, 선물, 찜(종속), 제공, 작성, 취향음식(종속))

\* 순환관계(선물), 삼항관계(배달) 포함.

전체 속성 수: **30개 이상**

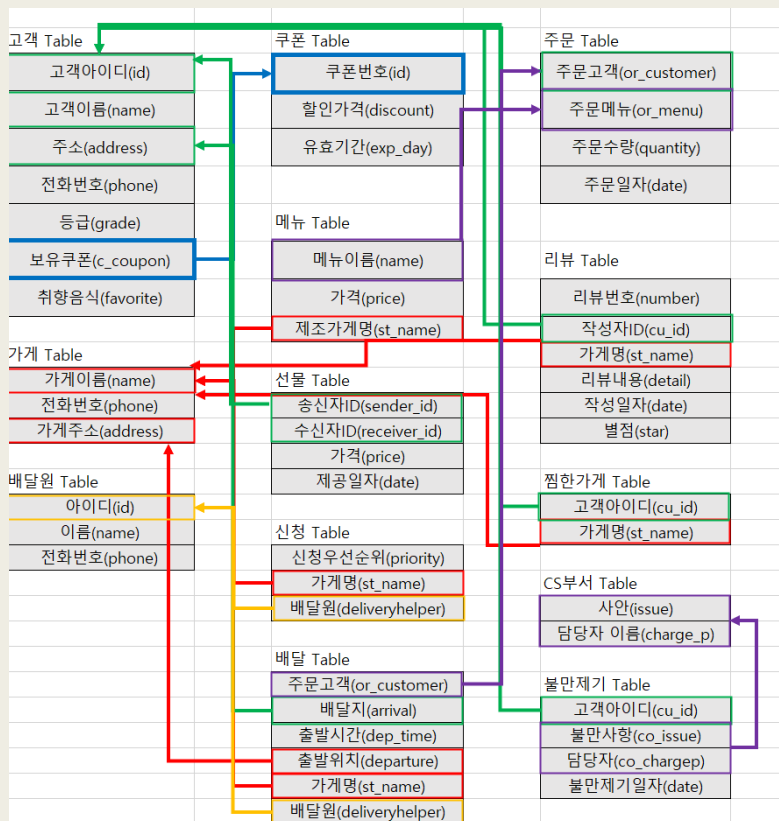
유도속성 2개 : 가게(거리), 배달(예상도착시간)

다중 값 속성 2개: 고객(취향음식), 찜한가게(가게명)



# 스키마 테이블 (~11/26)

테이블 이름			고객(customer)					
속성이름	데이터 타입	널 허용여부	기본값	기본키	외래키	제약조건	인스턴스 예시1	인스턴스 예시2
고객아이디(id)	CHAR(8)	N		PK			20000001	20000002
고객이름(name)	VARCHAR(20)	N					강한나	김진주
주소(address)	VARCHAR(50)	N					서울시 중구 장충동 13-2 202호	서울시 중구 필동 235-15
전화번호(phone)	VARCHAR(14)	N					010-1111-2222	010-3333-4444
등급(grade)	VARCHAR(10)	N	silver			silver, gold, vip만 허용	silver	gold
보유쿠폰(c_coupon)	VARCHAR(20)	N			O		S1	S2



TermProject 2차 제출 스키마 엑셀 파일 참조  
(발표자료에는 일부만 이미지 첨부했습니다.)

외래키 참조 다이어그램

# 테이블 생성 SQL문(~12/14)

```
MariaDB [termproject1]> CREATE TABLE customer (id CHAR(8) not null, name VARCHAR(20) not null, address VARCHAR(50) not null, phone VARCHAR(14) not null, grade VARCHAR(10) not null DEFAULT 'silver', PRIMARY KEY(id));
Query OK, 0 rows affected (0.019 sec)

MariaDB [termproject1]> CREATE TABLE coupon (id VARCHAR(20) not null, discount int not null DEFAULT '1000', exp_day DATETIME not null, PRIMARY KEY(id));
Query OK, 0 rows affected (0.022 sec)

MariaDB [termproject1]> ALTER TABLE customer ADD c_coupon VARCHAR(20) not null;
Query OK, 0 rows affected (0.017 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [termproject1]> describe customer;
```

Field	Type	Null	Key	Default	Extra
id	char(8)	NO	PRI	NULL	
name	varchar(20)	NO		NULL	
address	varchar(50)	NO		NULL	
phone	varchar(14)	NO		NULL	
grade	varchar(10)	NO		silver	
c_coupon	varchar(20)	NO		NULL	

```
6 rows in set (0.013 sec)

MariaDB [termproject1]> ALTER TABLE customer ADD CONSTRAINT couponID FOREIGN KEY(c_coupon) REFERENCES coupon(id);
Query OK, 0 rows affected (0.065 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

1) CREATE TABLE 활용해 customer와 coupon 테이블 구현

2) coupon의 id를 참조하는 customer 테이블의 외래키 'c\_coupon' 구현 위해 테이블 생성 이후에 ALTER TABLE 명령문을 활용해 속성 추가 및 FOREIGN KEY 조건 추가

3) DESCRIBE '테이블 이름' 명령어로 확인한 완성된 테이블 모습

# 테이블 생성 SQL문(~12/14)

1) CREATE TABLE 활용해 테이블 구현

2) FOREIGN KEY REFERENCES 문으로  
외래키 관계 설정

```
MariaDB [termproject1]> describe coupon;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | varchar(20) | NO   | PRI | NULL    |       |
| discount | int(11)    | NO   |     | 1000    |       |
| exp_day | datetime   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.009 sec)

MariaDB [termproject1]> CREATE TABLE favorite (id CHAR(8)
Query OK, 0 rows affected (0.020 sec)

MariaDB [termproject1]> describe favorite;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | char(8)   | NO   | PRI | NULL    |       |
| favorite | varchar(50) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.006 sec)

MariaDB [termproject1]> CREATE TABLE menu (name VARCHAR(50)
Query OK, 0 rows affected (0.020 sec)

MariaDB [termproject1]> describe menu;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name   | varchar(50) | NO   | PRI | NULL    |       |
| price  | int(11)    | NO   |     | NULL    |       |
| st_name | varchar(50) | NO   | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.008 sec)

MariaDB [termproject1]> describe store;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name   | varchar(50) | NO   | PRI | NULL    |       |
| phone  | varchar(14) | NO   |     | NULL    |       |
| address | varchar(50) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.006 sec)
```

# 테이블 생성 SQL문(~12/14)

1) INSERT INTO 문을 활용해 인스턴스를 테이블에 삽입

2) SELECT \* FROM 문으로  
테이블 내의 모든 스키마와 튜플 확인 가능

```
MariaDB [termproject1]> select * from customer;
```

id	name	address	phone	grade
20000001	강한나	서울시 중구 장충동 13-2 202호	010-1111-2222	silver
20000002	김진주	서울시 중구 필동 235-15	010-3333-4444	gold

```
2 rows in set (0.002 sec)

MariaDB [termproject1]> select * from coupon
-> ;
```

id	discount	exp_day
S1	1000	2020-11-26 00:00:00
S2	1500	2020-12-25 00:00:00

```
2 rows in set (0.000 sec)

INSERT INTO store VALUES ('마시따파스타','02-555...' at line 1
MariaDB [termproject1]> INSERT INTO store VALUES ('마시따파스타','02-5555-444
Query OK, 2 rows affected (0.002 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [termproject1]> select*from store;
```

name	phone	address
동국우동	02-3333-2222	서울시 중구 장충동 777-58
마시따파스타	02-5555-4444	서울시 중구 장충동 55-2

```
2 rows in set (0.000 sec)

MariaDB [termproject1]> describe `order`;
```

Field	Type	Null	Key	Default	Extra
or_customer	char(8)	NO	MUL	NULL	
or_menu	varchar(50)	NO	MUL	NULL	
quantity	int(11)	NO		1	
date	datetime	NO		NULL	

```
4 rows in set (0.004 sec)

MariaDB [termproject1]> select * from review;
```

number	cu_id	st_name	detail	date	star
1	20000001	동국우동	별로예요! 배달도 느리고 비싸요.	2020-11-26 00:00:00	2
2	20000002	마시따파스타	맛있어요. 양이 좀만 더 많았으면 좋겠네요.	2020-12-25 00:00:00	4



# 사용자별 작업 처리 SQL문 (~12/14)

사용자: 고객

- 고객이 받은 할인 쿠폰의 금액을 확인하고자 한다면?
- 고객이 리뷰 별점이 높은 순서대로 가게를 정렬해서 보고 싶다면?

```
MariaDB [termproject1]> SELECT cu.name, SUM(co.discount) AS '총할인금액'  
-> FROM customer cu, coupon co  
-> WHERE cu.c_coupon = co.id  
-> GROUP BY cu.name;
```

name	총할인금액
강하나	1000
김진주	1500

2 rows in set (0.010 sec)

```
MariaDB [termproject1]> SELECT st_name, star  
-> FROM review  
-> ORDER BY star DESC;
```

st_name	star
마시따파스타	4
동국우동	2

2 rows in set (0.000 sec)

# 사용자별 작업 처리 SQL문 (~12/14)

사용자: 점주

■ 하루 매출을 알고 싶다면?

■ 리뷰를 적어주고 **3점 이상 별점**을 준 고객이 주문했을 때에만  
리뷰 이벤트를 제공하려면?

```
MariaDB [termproject1]> SELECT st.name, SUM(me.price * ord.quantity) AS '총매상'
-> FROM store st, menu me, `order` ord
-> WHERE ord.or_menu = me.name AND me.st_name = st.name
-> GROUP BY st.name;
+-----+-----+
| name | 총매상 |
+-----+-----+
| 동국우동 | 7500 |
| 마시따파스타 | 21000 |
+-----+-----+
2 rows in set (0.003 sec)

MariaDB [termproject1]>
```

```
MariaDB [termproject1]> SELECT st.name, cu.name, ord.or_number
-> FROM customer cu, `order` ord, store st, review re
-> WHERE re.star >= 3 AND re.cu_id = cu.id AND ord.or_customer = cu.id AND re.st_name = st.name;
+-----+-----+-----+
| name | name | or_number |
+-----+-----+-----+
| 마시따파스타 | 김진주 | 0002 |
+-----+-----+-----+
1 row in set (0.002 sec)
```

테이블 이름			리뷰(review)				인스턴스 예시1	인스턴스 예시2
속성이름	데이터 타입	널 허용여부	기본값	기본키	외래키	제약조건		
리뷰번호(number)	INT	N		PK			1	2
작성자ID(cu_id)	CHAR(8)	N			O		20000001	20000002
가게명(st_name)	VARCHAR(50)	N			O		동국우동	마시따파스타
리뷰내용(detail)	VARCHAR(150)	N					별로예요! 배달도 느리고, 비싸요.	맛있어요. 양이 좀만 더 많았으면
작성일자(date)	DATETIME	N					2020-11-26	2020-12-25
별점(star)	INT	N	3			0이상 5이하	2	4

# 사용자별 작업 처리 SQL문 (~12/14)

사용자: 배달원

- 우선순위가 가장 높은 가게의 위치를 확인하고 싶다면?

```
MariaDB [termproject1]> SELECT reg.priority, reg.st_name, st.address  
-> FROM register reg, store st  
-> WHERE reg.st_name = st.name  
-> ORDER BY priority;
```

priority	st_name	address
1	마시따파스타	서울시 중구 장충동 55-2
2	동국우동	서울시 중구 장충동 777-58

2 rows in set (0.000 sec)

# 사용자별 작업 처리 SQL문 (~12/14)

사용자: CS센터 직원

- 배송 지연에 대해 문의를 준 고객에게 배달기사의 출발시간을 알려주고자 한다면?

```
MariaDB [termproject1]> SELECT cu.name, del.dep_time  
-> FROM customer cu, complain com, delivery del  
-> WHERE com.co_issue = '배달 지연 관련' AND com.cu_id = cu.id AND del.or_customer = cu.id;
```

name	dep_time
김진주	2020-12-25 01:17:32

1 row in set (0.002 sec)

- 위생 관련 문의를 준 고객이 주문한 가게와 주문한 메뉴를 찾고자 한다면?

```
MariaDB [termproject1]> SELECT cu.name, ord.or_menu, me.st_name  
-> FROM customer cu, `order`ord, menu me, complain com  
-> WHERE com.co_issue = '위생 관련' AND com.cu_id = cu.id AND ord.or_customer = cu.id AND ord.or_menu = me.name;
```

name	or_menu	st_name
강한나	봉골레파스타	마시따파스타

1 row in set (0.001 sec)

2020-2학기 데이터베이스 Term Project

**들어 주셔서  
감사합니다**