

---

# 과목 명: 데이터베이스시스템

담당 교수 명: 정 성 원

<<Project 2>>

서강대학교 컴퓨터공학과

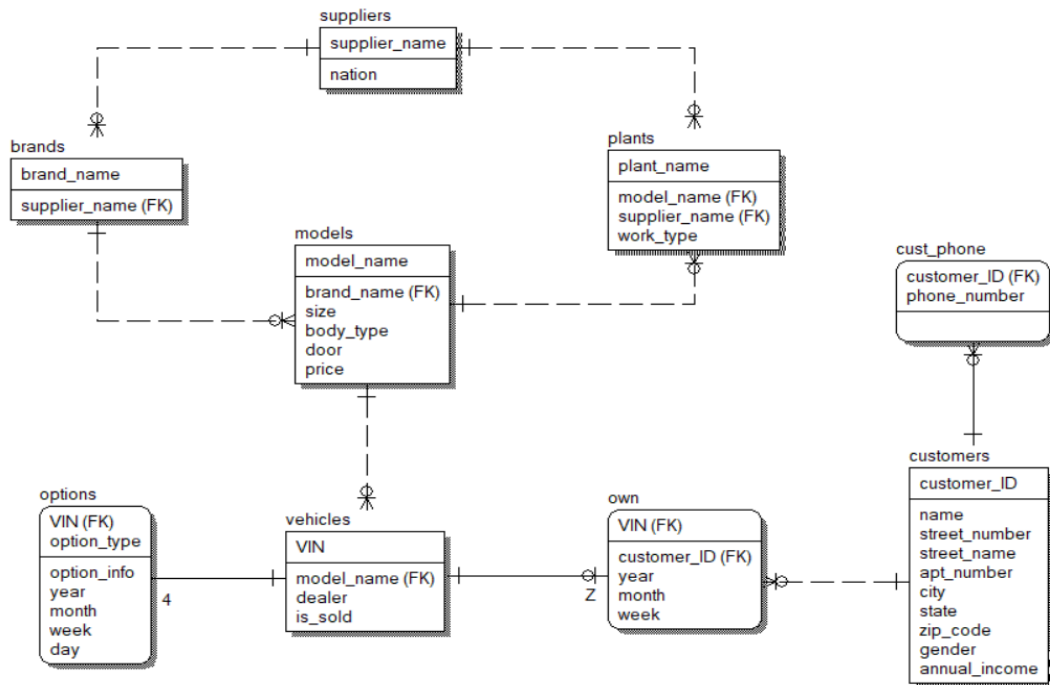
[학번] 20191574

[이름] 김예진

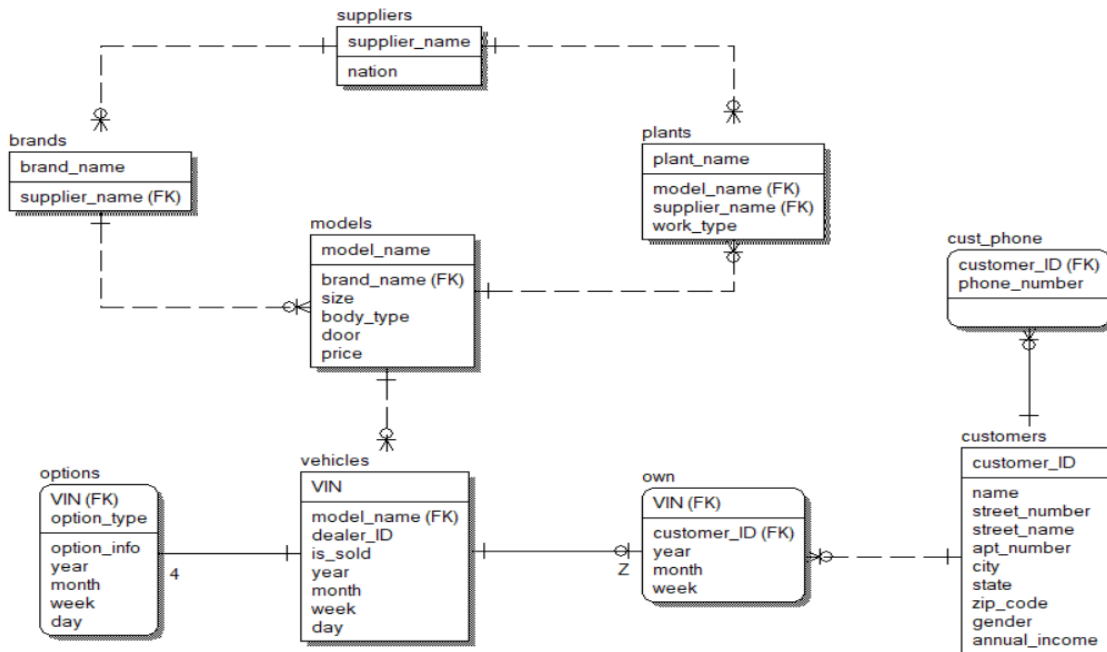
# 목 차

1. 프로젝트 1 의 Relation Schema	3
1.1 suppliers	4
1.2 brands	4
1.3 models	4
1.4 plants	4
1.5 vehicles	5
1.6 options	5
1.7 own	5
1.8 customers	5
1.9 cust_phone	5
2. Physical Schema Diagram	6
3. Queries	6
3.1 (TYPE 1) Show the sales trends for a particular brand over the past k years.	6
3.1.1 (TYPE 1-1) Then break these data out by gender of the buyer.	6
3.1.2 (TYPE 1-1-1) Then by income range.	8
3.2 (TYPE 2) Show sales trends for various brands over the past k months.	9
3.2.1 (TYPE 2-1) Then break these data out by gender of the buyer.	9
3.2.2 (TYPE 2-1-1) Then by income range.	10
3.3 (TYPE 3) Find that transmissions made by supplier (company name) between two given dates are defective.	11
3.3.1 (TYPE 3-1) Find the VIN of each car containing such a transmission and the customer to which it was sold.	11
3.3.2 (TYPE 3-2) Find the dealer who sold the VIN and transmission for each vehicles containing these transmissions.	12
3.4 (TYPE 4) Find the top k brands by dollar-amount sold by the year.	13
3.5 (TYPE 5) Find the top k brands by unit sales by the year.	14
3.6 (TYPE 6) In what month(s) do convertibles sell best?	15
3.7 (TYPE 7) Find those dealers who keep a vehicles in inventory for the longest average time.	15

## 1. 프로젝트 1 의 Relation Schema



일부 query 에서 어떤 vehicle 이 특정 dealer 의 판매 대상으로 지정되는 날짜가 필요하다고 판단하여 vehicles 에 날짜에 대한 속성을 다음과 같이 추가하여 진행하고자 한다.



## 1.1 suppliers

suppliers(supplier\_name, nation)에서 functional dependency 는 다음과 같다.

- supplier\_name -> nation

supplier\_name 은 suppliers 의 super key 이므로 suppliers 는 BCNF form 을 만족한다.

## 1.2 brands

brands(brand\_name, supplier\_name)에서 functional dependency 는 다음과 같다.

## 1.3 models

models(model\_name, brand\_name, size, body\_type, door, price)에서 functional dependency 는 다음과 같다.

- model\_name -> brand\_name, size, body\_type, door, price

model\_name 은 models 의 super key 이므로 models 는 BCNF form 을 만족한다.

## 1.4 plants

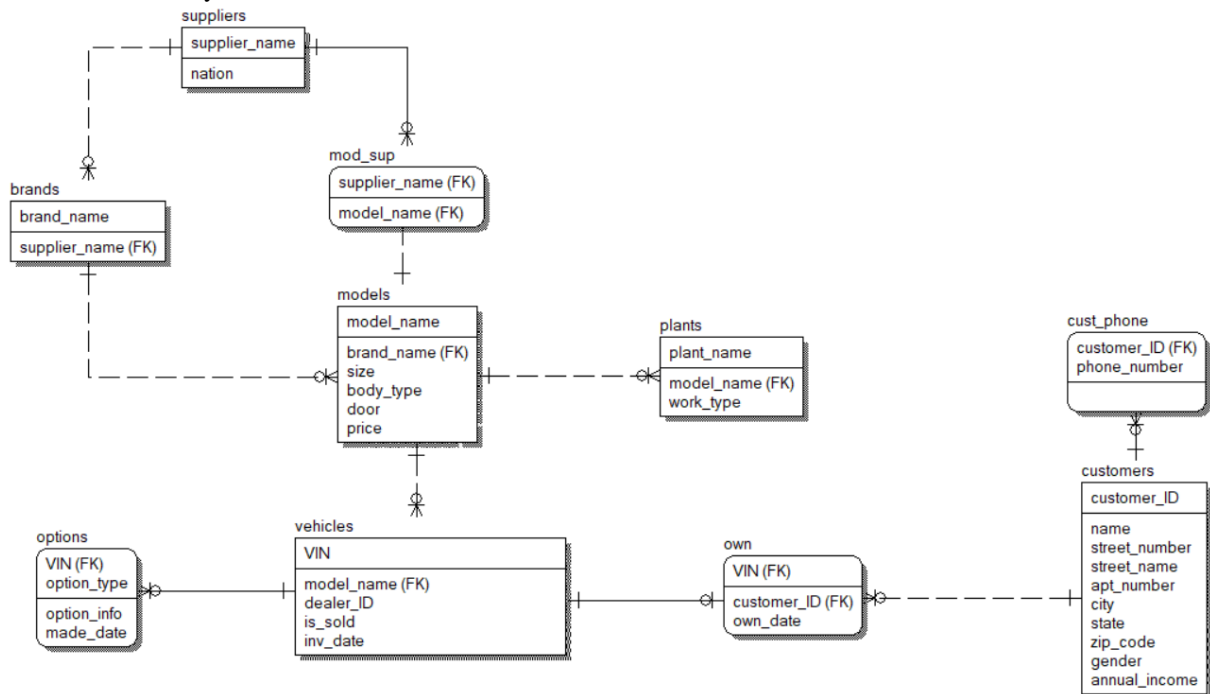
plants(plant\_name, model\_name, supplier\_name, work\_type)에서 functional dependency 는 다음과 같다.

- plant\_name -> model\_name, supplier\_name, work\_type (각 공장은 특정 회사의 특정 모델의 특정 work 를 진행한다.)
- **model\_name -> supplier\_name**

plant\_name 은 plants 의 super key 이므로 plants 는 BCNF form 을 만족한다.

하지만 model\_name -> supplier\_name 은 trivial 하지도 않고, model\_name 은 plants 의 super key 도 아니므로 BCNF form 에 부합하지 않는다.

따라서 이를 BCNF form 에 맞게 plants(plant\_name, model\_name, work\_type)과 mod\_sup(model\_name, supplier\_name)으로 분해하였다. 추가적으로 날짜에 대한 부분은 date 형식으로 간단하게 처리하기 위해 year, month, week, day 를 하나의 attribute 로 묶어서 적용하였다.



분해한 relation 을 다시 확인해보자.

plants(plant\_name, model\_name, work\_type)에는 다음과 같은 functional dependency 가 존재한다.

- plant\_name -> model\_name, work\_type

plant\_name 은 plants 의 super key 이므로 plants 는 BCNF form 을 만족한다.

mod\_sup(model\_name, supplier\_name)에는 다음과 같은 functional dependency 가 존재한다.

- model\_name -> supplier\_name

model\_name 은 mod\_sup 의 super key 이므로 plants 는 BCNF form 을 만족한다.

## 1.5 vehicles

vehicles(VIN, model\_name, dealer, is\_sold, year, month, week, day)에서 functional dependency 는 다음과 같다.

- VIN -> model\_name, dealer, is\_sold, year, month, week, day

VIN 은 vehicles 의 super key 이므로 vehicles 는 BCNF form 을 만족한다.

## 1.6 options

options(VIN, option\_type, option\_info, year, month, week, day)에서 functional dependency 는 다음과 같다.

- VIN, option\_type -> option\_info, year, month, week, day

{VIN, option\_type} 는 options 의 super key 이므로 options 는 BCNF form 을 만족한다.

## 1.7 own

own(VIN, customer\_ID, year, month, week)에서 functional dependency 는 다음과 같다.

- VIN -> customer\_ID, year, month, week

VIN 은 own 의 super key 이므로 own 은 BCNF form 을 만족한다.

## 1.8 customers

customers(customer\_ID, name, street\_number, street\_name, apt\_number, city, state, zip\_code, gender, annual\_income)에서 functional dependency 는 다음과 같다.

- customer\_ID -> name, street\_number, street\_name, apt\_number, city, state, zip\_code, gender, annual\_income

customer\_ID 는 customer 의 super key 이므로 customers 는 BCNF form 을 만족한다.

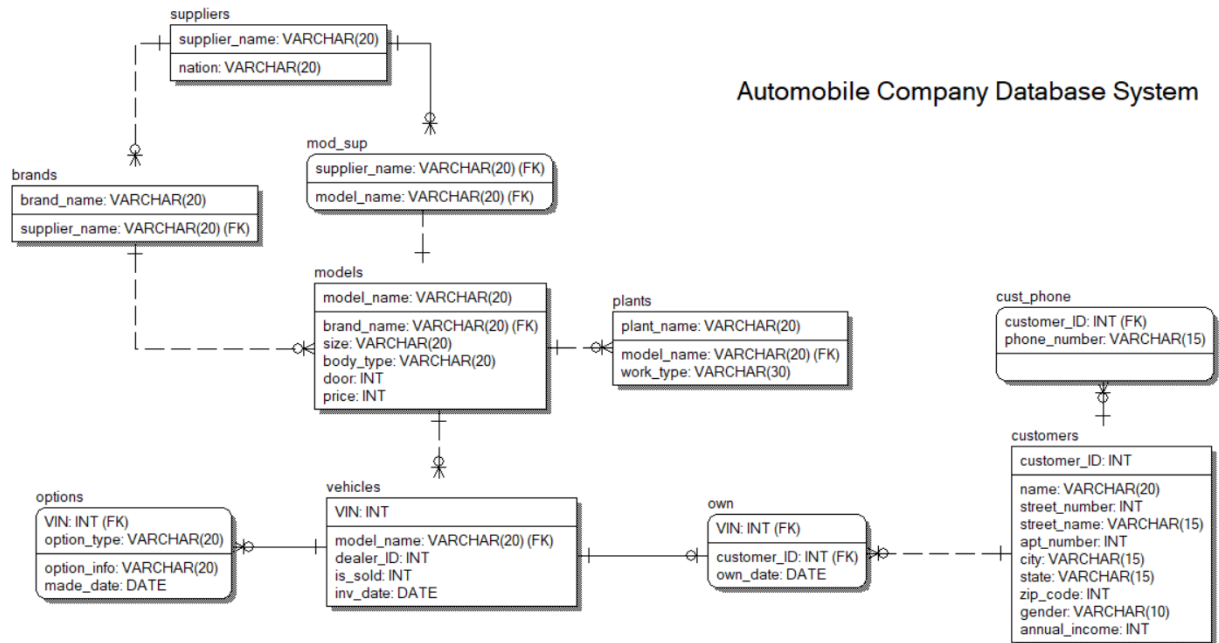
## 1.9 cust\_phone

cust\_phone(customer\_ID, phone\_number)에서 functional dependency 는 다음과 같다.

- customer\_ID, phone\_number -> customer\_ID, phone\_number

위의 함수 종속은 trivial 하므로 cust\_phone 은 BCNF form 을 만족한다.

## 2. Physical Schema Diagram



기본적으로 문자열을 저장해야 하는 datatype 은 기본적으로 VARCHAR(20)으로 설정하였다. 그리고 ID에 대한 부분은 INT 형으로, 날짜에 관한 속성은 모두 DATE 형으로 두었다.

이를 제외한 부분의 attribute 에 대해서는 다음과 같이 설정하였다. models 의 door 의 경우 차체의 문의 개수를 의미하는 것으로 INT 형으로, price 의 경우 해당 model 의 가격이므로 INT 형으로 나타냈다. plants 의 work\_type 의 경우 해당 공장이 하는 일에 대한 부분인데 추후 다양한 work 에 대해 처리 가능하도록 VARCHAR(30)으로 나타냈다. vehicles 의 is\_sold 는 해당 차량이 팔린 상태인지, inventory 내에 있는 지 여부를 저장하기 위한 것으로 INT 형으로 두었다. customers 에서는 예외적으로 주소 정보의 문자열 부분을 모두 VARCHAR(15)로 할당하였다. 주소의 숫자 부분은 INT 형으로 나타낼 수 있도록 했으며 gender 의 경우에는 male, female 에 대해 저장하기 위해 VARCHAR(10)으로 나타냈다. annual\_income 에 대해서는 연봉에 대한 정보이므로 INT 형으로 두었다.

## 3. Queries

### 3.1 (TYPE 1) Show the sales trends for a particular brand over the past k years.

Particular brand 와 k 에 대한 정보를 입력 받아 query 를 처리한다.

#### 3.1.1 (TYPE 1-1) Then break these data out by gender of the buyer.

```
select model_name, size, body_type, door, price, count(model_name) as sold, gender
from customers natural join own natural join vehicles natural join models
where brand_name = 'brand' and (DATEDIFF(own_date, 'date') >= 0 )
group by model_name, gender order by gender
```

customer, own, vehicles 와 models 를 join 함으로써 고객이 구입한 차량들의 model 정보를 가져올 수 있도록 하고, 그 중 user 가 입력한 brand 와 past k year 내에 해당하는 정보만 추출한다. 이 때, date 는 현재가 2021 년

이므로 2021 에서 입력한 k 를 뺀 값으로 yyyy 라고 하면 'yyyy-01-01'의 값을 가지고 있다. 그리고 이를 model 명과 gender 의 정보로 grouping 한 뒤, gender 로 ordering 하여 성별에 따라 해당 model 의 정보를 얻을 수 있도록 하였다. select 에서 모델에 대한 정보와 더불어 모델의 개수를 count 하여 각 모델에 대한 판매량 또한 볼 수 있다.

실행 결과는 다음과 같다.

### 1) Jaguar, 20

```

** Show the sales trends for a particular brand over the past k years **
Which brand ? Jaguar
Which k ? 20

----- Subtypes in TYPE 1 -----
1. TYPE 1-1
2. TYPE 1-1-1

SELECT SUBOPTION : 1

** Then break these data out by gender of the buyer **

  model_name      size      body_type  door      price      sold      gender
  -----
Jaguar XJ         full      sedan      4      195000000      1      female
Jaguar XF         semi-large station wagon 5      68400000      1      male
Jaguar XJ         full      sedan      4      195000000      1      male

```

### 2) Maserati, 8

```

** Show the sales trends for a particular brand over the past k years **
Which brand ? Maserati
Which k ? 8

----- Subtypes in TYPE 1 -----
1. TYPE 1-1
2. TYPE 1-1-1

SELECT SUBOPTION : 1

** Then break these data out by gender of the buyer **

  model_name      size      body_type  door      price      sold      gender
  -----

```

### 3) Maserati, 20

```

** Show the sales trends for a particular brand over the past k years **
Which brand ? Maserati
Which k ? 20

----- Subtypes in TYPE 1 -----
1. TYPE 1-1
2. TYPE 1-1-1

SELECT SUBOPTION : 1

** Then break these data out by gender of the buyer **

  model_name      size      body_type  door      price      sold      gender
  -----
Levante          semi-large SUV      4      132100000      2      female
Levante          semi-large SUV      4      132100000      2      male

```

### 3.1.2 (TYPE 1-1-1) Then by income range.

```
select model_name, size, body_type, door, price, annual_income
from customers natural join own natural join vehicles natural join models
where brand_name = 'brand' and (DATEDIFF(own_date, 'date') >= 0 )
order by annual_income
```

(TYPE 1-1)과 유사한 방식으로 작성하되, annual\_income 에 따른 정보 분석이므로 별도의 grouping 은 진행하지 않고 annual\_income 으로 ordering 하였다.

실행 결과는 다음과 같다.

#### 1) Jaguar, 20

```
** Show the sales trends for a particular brand over the past k years **
Which brand ? Jaguar
Which k ? 20

----- Subtypes in TYPE 1 -----
1. TYPE 1-1
2. TYPE 1-1-1

SELECT SUBOPTION : 2

** Then by income range **

  model_name      size      body_type  door      price      annual_income
-----
Jaguar XJ         full      sedan      4      195000000      40155000
Jaguar XJ         full      sedan      4      195000000      53090000
Jaguar XF         semi-large station wagon 5      68400000      166790000
```

#### 2) Maserati, 8

```
** Show the sales trends for a particular brand over the past k years **
Which brand ? Maserati
Which k ? 8

----- Subtypes in TYPE 1 -----
1. TYPE 1-1
2. TYPE 1-1-1

SELECT SUBOPTION : 2

** Then by income range **

  model_name      size      body_type  door      price      annual_income
```

#### 3) Maserati, 20

```
** Show the sales trends for a particular brand over the past k years **
Which brand ? Maserati
Which k ? 20

----- Subtypes in TYPE 1 -----
1. TYPE 1-1
2. TYPE 1-1-1

SELECT SUBOPTION : 2

** Then by income range **

  model_name      size      body_type  door      price      annual_income
-----
Levante          semi-large SUV      4      132100000      53090000
Levante          semi-large SUV      4      132100000      66870000
Levante          semi-large SUV      4      132100000      119475000
Levante          semi-large SUV      4      132100000      119475000
```



### 3.2 (TYPE 2) Show sales trends for various brands over the past k months.

k에 대한 정보를 입력 받아 query를 처리한다.

#### 3.2.1 (TYPE 2-1) Then break these data out by gender of the buyer.

```
select model_name, brand_name, size, body_type, door, price, count(model_name) as sold, gender
from customers natural join own natural join vehicles natural join models
where DATEDIFF(own_date, DATE_SUB(CURDATE(), INTERVAL mm MONTH)) >= 0
group by model_name, gender
order by gender
```

(TYPE 1-1)과 유사한 방식으로 작성하되, 다양한 brand에 대해서 정보를 가져와야 하므로 where 문에 brand에 대한 지정은 하지 않고, k month 만큼 현재에서 뺀 날짜보다 최근인 경우 정보를 가져올 수 있도록 작성하였다.

실행 결과는 다음과 같다.

1) 5

```
** Show sales trends for various brands over the past k months **
Which k ? 5

----- Subtypes in TYPE 2 -----
1. TYPE 2-1
2. TYPE 2-1-1

SELECT SUBOPTION : 1

** Then break these data out by gender of the buyer **
```

model_name	brand_name	size	body_type	door	price	sold	gender
Aventador	Lamborghini	mid	coupe	2	577300000	1	female

2) 20

```
** Show sales trends for various brands over the past k months **
Which k ? 20

----- Subtypes in TYPE 2 -----
1. TYPE 2-1
2. TYPE 2-1-1

SELECT SUBOPTION : 1

** Then break these data out by gender of the buyer **
```

model_name	brand_name	size	body_type	door	price	sold	gender
Aventador	Lamborghini	mid	coupe	2	577300000	1	female
Aventador	Lamborghini	mid	coupe	2	577300000	1	male
Jaguar XJ	Jaguar	full	sedan	4	195000000	1	male
Malibu	Chevrolet	mid	sedan	4	23640000	1	male
Rogue	Nissan	mid	SUV	4	31000000	1	male

3) 50

```

** Show sales trends for various brands over the past k months **
Which k ? 50

----- Subtypes in TYPE 2 -----
1. TYPE 2-1
2. TYPE 2-1-1

SELECT SUBOPTION : 1

** Then break these data out by gender of the buyer **

  model_name    brand_name    size    body_type    door    price    sold    gender
-----
      A5         Audi        mid    convertible    2    49355000    1    female
Aventador     Lamborghini    mid        coupe    2    577300000    1    female
      CLA         Benz        mid        coupe    4    55200000    1    female
Aventador     Lamborghini    mid        coupe    2    577300000    1    male
Cherokee      Jeep          mid    station wagon    4    48400000    1    male
Jaguar XF      Jaguar       semi-large    station wagon    5    68400000    1    male
Jaguar XJ      Jaguar        full        sedan    4    195000000    1    male
Malibu        Chevrolet    mid        sedan    4    23640000    2    male
Rouge         Nissan        mid         SUV    4    31000000    1    male

```

### 3.2.2 (TYPE 2-1-1) Then by income range.

```

select model_name, brand_name, size, body_type, door, price, annual_income
from customers natural join own natural join vehicles natural join models
where DATEDIFF(own_date, DATE_SUB(CURDATE(), INTERVAL mm MONTH)) >= 0
order by annual_income

```

(TYPE 2-1)과 유사한 방식으로 작성하되, annual\_income 에 따른 정보 분석이므로 별도의 grouping 은 진행하지 않고 annual\_income 으로 ordering 하였다.  
실행 결과는 다음과 같다.

1) 5

```

** Show sales trends for various brands over the past k months **
Which k ? 5

----- Subtypes in TYPE 2 -----
1. TYPE 2-1
2. TYPE 2-1-1

SELECT SUBOPTION : 2

** Then by income range **

  model_name    brand_name    size    body_type    door    price    annual_income
-----
Aventador     Lamborghini    mid        coupe    2    577300000    40155000

```

2) 20

\*\* Show sales trends for various brands over the past k months \*\*  
Which k ? 20

----- Subtypes in TYPE 2 -----  
1. TYPE 2-1  
2. TYPE 2-1-1

SELECT SUBOPTION : 2

\*\* Then by income range \*\*

model_name	brand_name	size	body_type	door	price	annual_income
Aventador	Lamborghini	mid	coupe	2	577300000	40155000
Jaguar XJ	Jaguar	full	sedan	4	195000000	53090000
Rouge	Nissan	mid	SUV	4	31000000	82115000
Aventador	Lamborghini	mid	coupe	2	577300000	112690000
Malibu	Chevrolet	mid	sedan	4	23640000	169345000

3) 50

\*\* Show sales trends for various brands over the past k months \*\*  
Which k ? 50

----- Subtypes in TYPE 2 -----  
1. TYPE 2-1  
2. TYPE 2-1-1

SELECT SUBOPTION : 2

\*\* Then by income range \*\*

model_name	brand_name	size	body_type	door	price	annual_income
Aventador	Lamborghini	mid	coupe	2	577300000	40155000
Cherokee	Jeep	mid	station wagon	4	48400000	44135000
Jaguar XJ	Jaguar	full	sedan	4	195000000	53090000
CLA	Benz	mid	coupe	4	55200000	53645000
Rouge	Nissan	mid	SUV	4	31000000	82115000
A5	Audi	mid	convertible	2	49355000	109035000
Aventador	Lamborghini	mid	coupe	2	577300000	112690000
Malibu	Chevrolet	mid	sedan	4	23640000	127690000
Jaguar XF	Jaguar	semi-large	station wagon	5	68400000	166790000
Malibu	Chevrolet	mid	sedan	4	23640000	169345000

### 3.3 (TYPE 3) Find that transmissions made by supplier (company name) between two given dates are defective.

Particular supplier 의 이름과 start date, end date 를 입력 받아 query 를 진행한다.

#### 3.3.1 (TYPE 3-1) Find the VIN of each car containing such a transmission and the customer to which it was sold.

with defect (VIN, model\_name, customer\_ID) as

```
(select VIN, model_name, customer_ID
from vehicles natural join own natural join options
where option_type = 'transmission' and DATEDIFF(made_date, start_date) >= 0 and DATEDIFF(end_date,
made_date)>=0)
select VIN, customer_ID
from defect natural join mod_sup
```

where supplier\_name = 'supplier'

with 절에서 vehicles, own 과 options 를 join 하여 소비자가 구매한 차량에 대해 option 정보를 얻고 그 중 transmission 에 해당하는 option 과 start\_date 부터 end\_date 사이에 제조된 것들의 정보만 추출한 뒤, 이 transmission 을 가진 차량에 대한 정보를 attribute 으로 한 임시 릴레이션 defect 를 정의한다.

defect 와 mod\_sup 을 join 하여 defect 가 있는 차량들의 supplier 정보를 입력으로 들 온 값과 비교하여 해당 supplier 에서 제조된 option 만 추려내어 정보를 얻는다.

실행 결과는 다음과 같다.

#### 1) Tata, 2000-01-01, 2016-01-01

```
** Find that transmissions made by supplier (company name) between two given dates are defective **
Which supplier ? Tata
Start date (form : yyyy-mm-dd) ? 2000-01-01
End date (form : yyyy-mm-dd) ? 2016-01-01

----- Subtypes in TYPE 3 -----
1. TYPE 3-1
2. TYPE 3-2

SELECT SUBOPTION : 1

** Find the VIN of each car containing such a transmission and the customer to which it was sold **

VIN      customer_ID
22        1
3         12
31        21
```

#### 2) Tata, 2000-06-11, 2007-12-01

```
** Find that transmissions made by supplier (company name) between two given dates are defective **
Which supplier ? Tata
Start date (form : yyyy-mm-dd) ? 2000-06-11
End date (form : yyyy-mm-dd) ? 2007-12-01

----- Subtypes in TYPE 3 -----
1. TYPE 3-1
2. TYPE 3-2

SELECT SUBOPTION : 1

** Find the VIN of each car containing such a transmission and the customer to which it was sold **

VIN      customer_ID
3         12
31        21
```

### 3.3.2 (TYPE 3-2) Find the dealer who sold the VIN and transmission for each vehicles containing these transmissions.

```
with defect (VIN, dealer_ID, model_name) as
  (select VIN, dealer_ID, model_name
   from vehicles natural join own natural join options
   where option_type = 'transmission' and DATEDIFF(made_date, start_date) >= 0 and DATEDIFF(end_date,
made_date)>=0)
  select VIN, dealer_ID
```

```
from defect natural join mod_sup
where supplier_name = 'supplier'
```

(TYPE 2-1)과 유사한 방식으로 작성하되, dealer 의 정보를 필요로 하기 때문에 defect 를 정의할 때 customer\_ID 대신 dealer\_ID 를 추출하였다.

실행 결과는 다음과 같다.

#### 1) Tata, 2000-01-01, 2016-01-01

```
** Find that transmissions made by supplier (company name) between two given dates are defective **
Which supplier ? Tata
Start date (form : yyyy-mm-dd) ? 2000-01-01
End date (form : yyyy-mm-dd) ? 2016-01-01

----- Subtypes in TYPE 3 -----
1. TYPE 3-1
2. TYPE 3-2

SELECT SUBOPTION : 2

** Find the dealer who sold the VIN and transmission for each vehicles containing these transmissions **

VIN      dealer_ID
22        15
3         8
31        15
```

#### 2) Tata, 2000-06-11, 2007-12-01

```
** Find that transmissions made by supplier (company name) between two given dates are defective **
Which supplier ? Tata
Start date (form : yyyy-mm-dd) ? 2000-06-11
End date (form : yyyy-mm-dd) ? 2007-12-01

----- Subtypes in TYPE 3 -----
1. TYPE 3-1
2. TYPE 3-2

SELECT SUBOPTION : 2

** Find the dealer who sold the VIN and transmission for each vehicles containing these transmissions **

VIN      dealer_ID
3         8
31        15
```

### 3.4 (TYPE 4) Find the top k brands by dollar-amount sold by the year.

k 와 year 에 대한 정보를 입력 받아 query 를 처리한다.

```
select brand_name, sum(price) as dollar_amount_sold
from vehicles natural join own natural join models
where YEAR(own_date) = yyyy
group by brand_name order by dollar_amount_sold desc
```

vehicles, own 과 models 를 join 하여 고객이 구매한 차량의 모델 정보를 가져오고, 그 중 해당 년도에 구입한 정보만 추출한다. 이 정보를 brand\_name 에 따라 grouping 하고 그 brand 에 속한 차량들의 가격을 모두 더하여 해당 년도의 dollar-amount sold 를 구한다. 그리고 dollar\_amount\_sold 를 기준으로 내림차순으로 정렬해주었다.

실행 결과는 다음과 같다.

1) 4, 2012

```
** Find the top k brands by dollar-amount sold by the year **
Which k ? 4
Which year (form : yyyy) ? 2012
    brand_name  dollar_amount_sold
    Lamborghini 577300000
    Maserati    132100000
    Jeep        50900000
    Nissan      31000000
```

2) 2, 2012

```
** Find the top k brands by dollar-amount sold by the year **
Which k ? 2
Which year (form : yyyy) ? 2012
    brand_name  dollar_amount_sold
    Lamborghini 577300000
    Maserati    132100000
```

3) 3, 2018

```
** Find the top k brands by dollar-amount sold by the year **
Which k ? 3
Which year (form : yyyy) ? 2018
    brand_name  dollar_amount_sold
    Jaguar      68400000
    Jeep         48400000
    Chevrolet    23640000
```

### 3.5 (TYPE 5) Find the top k brands by unit sales by the year.

k 와 year 에 대한 정보를 입력 받아 query 를 처리한다.

```
select brand_name, count(VIN) as unit_sales
from vehicles natural join own natural join models
where YEAR(own_date) = yyyy
group by brand_name order by unit_sales desc
```

(TYPE 4)과 유사한 방식으로 작성하되, 가격의 합이 아닌 판매 차량의 대수에 대한 정보를 필요로 하기 때문에 sum(price) 대신 count(VIN)의 정보를 추출하였고, 판매 대수를 기준으로 내림차순 정렬하였다.

실행 결과는 다음과 같다.

1) 4, 2012

```
** Find the top k brands by unit sales by the year **
Which k ? 4
Which year (form : yyyy) ? 2012
    brand_name  unit sales
    Jeep        1
    Lamborghini 1
    Maserati    1
    Nissan      1
```

2) 2, 2012

```
** Find the top k brands by unit sales by the year **
```

```
Which k ? 2
```

```
Which year (form : yyyy) ? 2012
```

brand_name	unit sales
Nissan	1
Lamborghini	1
Maserati	1
Jeep	1

➔ 위와 같이 공동 rate 에 대해서는 모두 출력하도록 code 를 작성하였다.

➔ 만약 위의 4 개의 brand 의 unit sales 가 2 이고, 다른 A 라는 brand 의 unit sales 가 1 이었다면 위의 4 개 brand 의 순위는 모두 1 위이고, A 의 경우 5 위이므로 A 는 출력하지 않는 메커니즘으로 구현하였다.

+) TYPE 4 는 가격에 대한 정보를 추출하는 것이므로 위와 같은 상황의 발생 확률은 낮지만 이에 대해서도 동일한 메커니즘으로 작동될 수 있도록 구현하였다.

3) 3, 2018

```
** Find the top k brands by unit sales by the year **
```

```
Which k ? 3
```

```
Which year (form : yyyy) ? 2018
```

brand_name	unit sales
Jeep	1
Chevrolet	1
Jaguar	1

### 3.6 (TYPE 6) In what month(s) do convertibles sell best?

```
select MONTH(own_date), count(VIN) as unit_sales  
from vehicles natural join own natural join models  
where body_type = 'convertible'  
group by MONTH(own_date)  
order by unit_sales desc
```

vehicles, own, model 을 join 하여 고객이 구매한 차량의 model 정보를 얻고, 그 중 convertible 차량에 대한 정보만을 선별한다. 고객이 이 차량을 구매한 날짜의 ‘월(month)’ 정보를 얻고 이를 기준으로 grouping 하여 각 ‘월’에 따른 convertible 차량의 판매량을 계산한다. 그리고 이를 판매량 기준으로 내림차순 정렬하였다. 실행 결과는 다음과 같다.

```
** In what month(s) do convertibles sell best ? **
```

```
Which month do convertible sell best : 10, 3, 1
```

➔ 10 월, 3 월, 1 월에 convertible 차량의 판매 대수가 공동으로 가장 많다.

### 3.7 (TYPE 7) Find those dealers who keep a vehicle in inventory for the longest average time.

```
select dealer_ID, avg(DATEDIFF(CURDATE(), inv_date)) as avg_time  
from vehicles  
where is_sold = 0  
group by dealer_ID  
order by avg_time desc
```

vehicles table 에서 is\_sold 가 0, 즉 아직 판매되지 않은 차량의 정보를 얻고 이 정보를 dealer\_ID 를

기준으로 grouping 한다. 이 때, 각 dealer 별로 가지고 있는 차량들이 inventory 에 keep 되어 있던 평균 일 수를 구하여 이를 기준으로 내림차순 정렬한다.

실행 결과는 다음과 같다.

```
** Find those dealers who keep a vehicle in inventory for the longest average time **  
Which dealers keep a vehicle in inventory longest : 9
```

➔ 9 호 딜러점이 평균적으로 차량을 inventory 에 오래 보관했다.