
과목 명: 데이터베이스시스템

담당 교수 명: 정 성 원

<<Project 1>>

서강대학교 컴퓨터공학과

[학번] 20191574

[이름] 김예진

목 차

1. E-R model	3
1.1 개체와 속성 선택	3
1.1.1 vehicles	3
1.1.2 brands	3
1.1.3 models	3
1.1.4 options	3
1.1.5 suppliers	3
1.1.6 plants	3
1.1.7 customers	4
1.2 중복된 속성 제거	4
1.2.1 vehicles	4
1.2.2 brands	4
1.2.3 models	4
1.2.4 options	5
1.2.5 suppliers	5
1.2.6 plants	5
1.2.7 customers	6
1.3 개체들 간의 관계 정리	6
1.3.1 launch	6
1.3.2 br_mod	6
1.3.3 sup_pl	7
1.3.4 manufacture	7
1.3.5 mod_veh	8
1.3.6 veh_op	8
1.3.7 own	8
1.4 Visualization	10
2. Relational Schema	11
2.1 E-R model에서 Relational Schema로의 변환	11
2.1.1 Strong entity set with simple attributes	11
2.1.2 Strong entity set with Composite Attributes	11
2.1.3 Weak entity set	11
2.1.4 Relationship set	11
2.2 스키마의 중복성	12
2.2.1 veh_op	12
2.3 스키마의 조합	12
2.3.1 launch	12
2.3.2 br_mod	12
2.3.3 sup_pl	12
2.3.4 mod_veh	12
2.3.5 manufacture	12
2.4 Visualization	13

1. E-R model

1.1 개체와 속성 선택

데이터 베이스 내에 포함되어야 할 개체 집합을 결정하고 각 개체마다 적절한 속성을 선택한다.

1.1.1 *vehicles*

속성으로 VIN, model_name, customer_ID, dealer, is_sold 를 포함하며, VIN 을 primary key 로 갖는다. VIN 은 자동차를 식별할 수 있게 해주는 번호이고, model_name 은 해당 차량의 모델명, customer_ID 는 이 차량을 소유하고 있는 고객의 고객 ID, dealer 는 해당 차량을 판매했거나 판매할 예정인 판매점의 이름, is_sold 는 차량이 판매되었는지 여부에 대한 정보 제공한다. 아직 판매되지 않은 차량에 대해서는 customer_ID 가 null 이 될 수 있다.

1.1.2 *brands*

속성으로 brand_name 을 포함하며, brand_name 을 primary key 로 갖는다. brand_name 은 브랜드명을 의미한다.

1.1.3 *models*

속성으로 model_name, brand_name, size, body_type, door, price 를 포함하며, model_name 을 primary key 로 갖는다. model_name 은 모델명, brand_name 은 해당 모델을 출시한 브랜드의 이름, size 는 해당 모델의 차량 크기(경형, 소형, ...), body_type 은 해당 모델의 차량 형태(세단, 쿠페, 컨버터블, 웨건, ...), door 은 문의 개수(2-door, 4-door, ...), price 는 모델 가격을 의미한다.

1.1.4 *options*

속성으로 VIN, option_type, option_info, (year, month, week, day)를 포함하며, VIN, option_type 을 primary key 로 갖는다. VIN 은 자동차를 식별할 수 있는 번호, option_type 은 옵션의 종류로 color, engine, transmission 이 해당 속성의 인스턴스로 올 수 있다. option_info 는 option_type 에 대한 부가적인 정보로, option_type 이 color 라면 blue, black 등의 색 정보, engine 이라면 engine 의 한 종류와 같이 추가 정보를 저장할 수 있다. (year, month, week, day)는 해당 자동차에 부착된 option 이 제조된 날짜를 저장한다. 이 option 개체는 vehicle 과의 관계에서 vehicle 에 추가된 option 의 정보를 기술하는 용도로 사용된다.

1.1.5 *suppliers*

속성으로 supplier_name, nation 을 포함하며, supplier_name 을 primary key 로 갖는다. supplier_name 은 회사의 이름, nation 은 회사의 소속 국가를 의미한다.

1.1.6 *plants*

속성으로 plant_name, work_type, model_name, supplier_name 을 포함하며, plant_name 을 primary key 로 갖는다. plant_name 은 공장의 이름, work_type 은 assemble, color, engine, transmissioin 중 하나를 인스턴스로

갖는 속성으로 해당 공장이 하는 일의 종류가 어떤 것인지를 저장할 수 있다. `model_name`은 이 공장에서 만드는 부품이 어떤 `model`의 것인지를 나타낼 수 있는 모델명 속성이고, `supplier_name`은 이 공장을 소유하고 있는 회사의 이름을 나타낸다.

1.1.7 customers

속성으로 ID, name, ((street_number, street_name, apt_number), city, state, zip_code), {phone}, gender, annual income 을 포함하며 ID 를 primary key 로 갖는다.

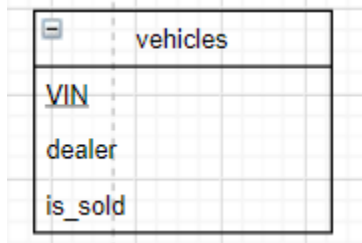
ID 는 고객을 식별할 수 있도록 부여한 식별자고, name 은 고객명이다. ((street_number, street_name, apt_number), city, state, zip_code)는 고객의 주소를 나타내고, phone 은 고객의 전화번호이다. 고객이 전화번호를 여러 개 소유할 수 있으므로 이는 multi-value 로 나타낸다. gender 는 고객의 성별, annual_income 은 고객의 연봉을 나타낸다.

1.2 중복된 속성 제거

개체들 간에 불필요하게 중복된 속성을 제거한다.

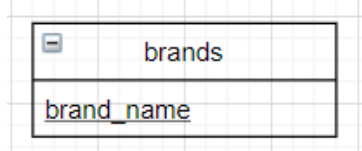
1.2.1 vehicles

vehicles(VIN, dealer, is_sold)



1.2.2 brands

brands(brand_name)



1.2.3 models

models(model_name, size, body_type, door, price)

models
<u>model_name</u>
size
body_type
door
price

1.2.4 options

options(VIN, option_type, option_info, (year, month, week, day))

options
<u>option_type</u>
option_info
date
year
month
week
day

1.2.5 suppliers

suppliers(supplier_name, nation)

suppliers
<u>supplier_name</u>
nation

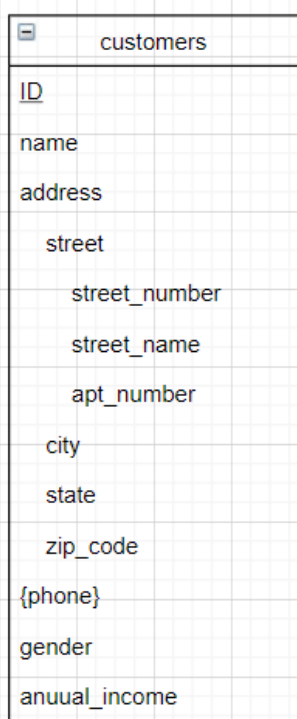
1.2.6 plants

plants(plant_name, work_type, model_name, supplier_name)

plants
<u>plant_name</u>
work_type

1.2.7 customers

customers(ID, name, ((street_number, street_name, apt_number), city, state, zip_code), {phone}, gender, annual income)

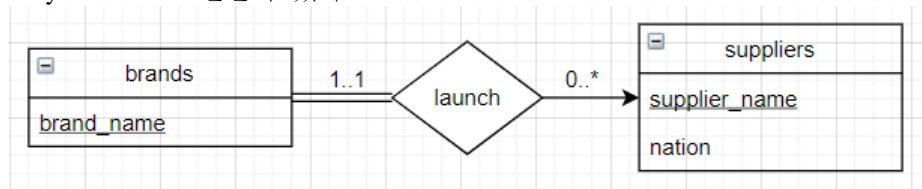


1.3 개체들 간의 관계 정리

개체들 간의 관계를 정리한다.

1.3.1 launch

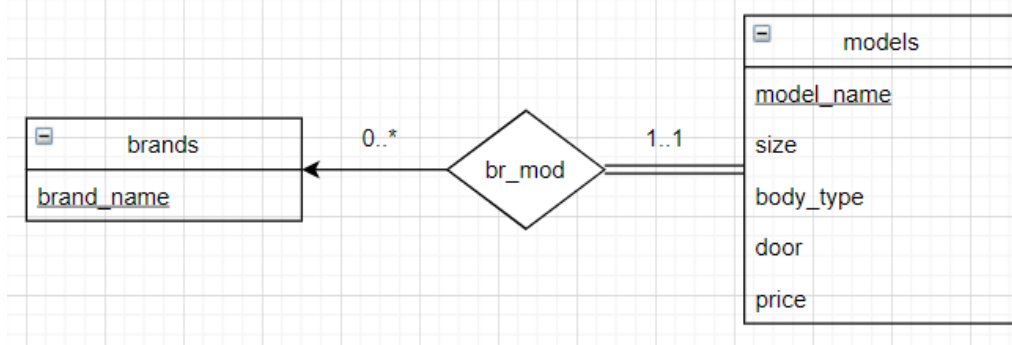
launch 는 brands 와 suppliers 의 관계를 나타낸다. brand 는 하나의 supplier 에 속해 있고, 하나의 supplier 는 여러 개의 brand 를 갖는 관계이므로 many to one 관계이고, 모든 brand 는 하나의 supplier 와 항상 관계를 가져야 하므로 이 관계에 전체적으로 참가한다. brand 는 무조건 하나의 supplier 에 연결되므로 cardinality 는 1..1, supplier 는 0 부터 무한 개의 brand 와 연결될 수 있으므로 cardinality 는 0..*로 표현할 수 있다.



1.3.2 br_mod

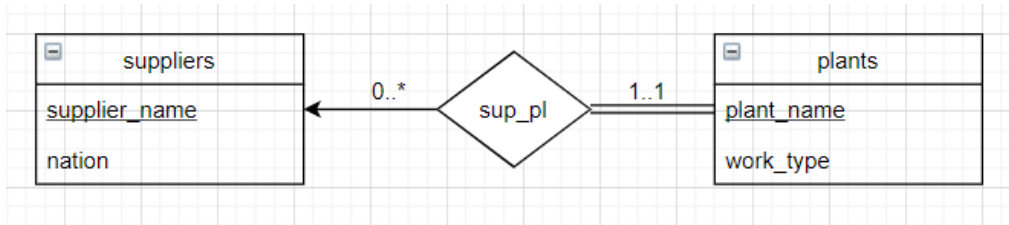
br_mod 는 brands 와 models 의 관계를 나타낸다. 하나의 brand 가 여러 개의 model 을 가지며, model 은 하나의 brand 에 속해 있으므로 one to many 관계이고, 모든 model 은 하나의 brand 와 항상 관계를 가져야 하므로 이 관계에 전체적으로 참가한다. model 은 무조건 하나의 brand 에 연결되므로

cardinality 는 1..1, brand 는 0 부터 무한 개의 model 과 연결될 수 있으므로 cardinality 는 0..*로 표현할 수 있다.



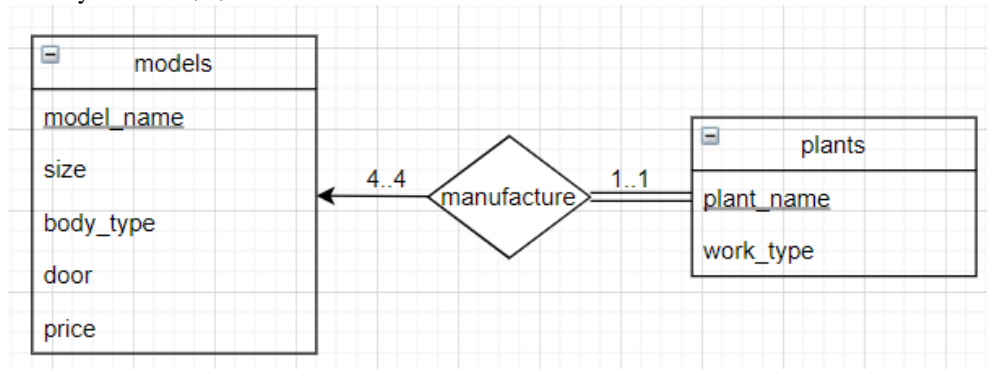
1.3.3 sup_pl

sup_pl 은 suppliers 와 plants 의 관계를 나타낸다. 하나의 supplier 는 여러 개의 plant 를 가지며 plant 는 하나의 supplier 에 속해 있는 관계이므로 one to many 관계이고, 모든 plant 는 하나의 supplier 와 항상 관계를 가져야 하므로 이 관계에 전체적으로 참가한다. plant 는 무조건 하나의 supplier 와 연결되므로 cardinality 는 1..1, supplier 는 0 부터 무한 개의 plant 와 연결될 수 있으므로 cardinality 는 0..*로 표현할 수 있다.



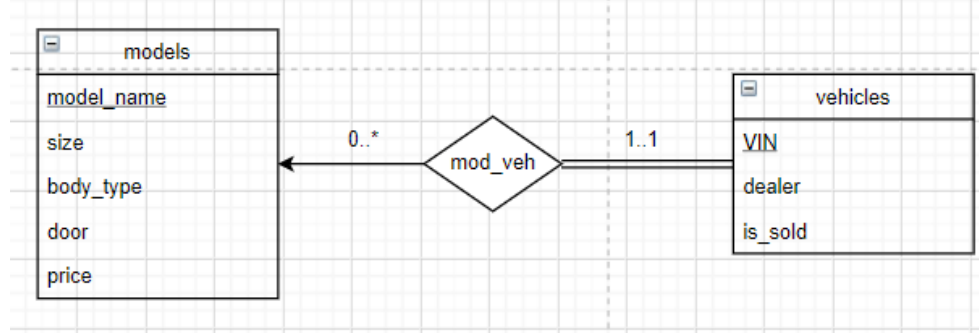
1.3.4 manufacture

manufacture 은 models 와 plants 의 관계를 나타낸다. 하나의 모델은 각각 assemble, color, engine, transmission 에 대한 작업을 하는 plant 를 각각 가져 총 4 개의 plant 와 관계를 맺는다. 또한 하나의 plant 는 항상 하나의 model 에만 연결될 수 있으므로 manufacture 은 one to many 관계다. plant 는 항상 특정 model 에 대한 작업을 수행해야 하므로 이 관계에 전체적으로 참가한다. model 은 4 개의 plant 와 관계를 맺으므로 cardinality 는 4..4 이고, plant 는 한 개의 model 과만 관계를 맺으므로 cardinality 는 1..1 이다.



1.3.5 mod_veh

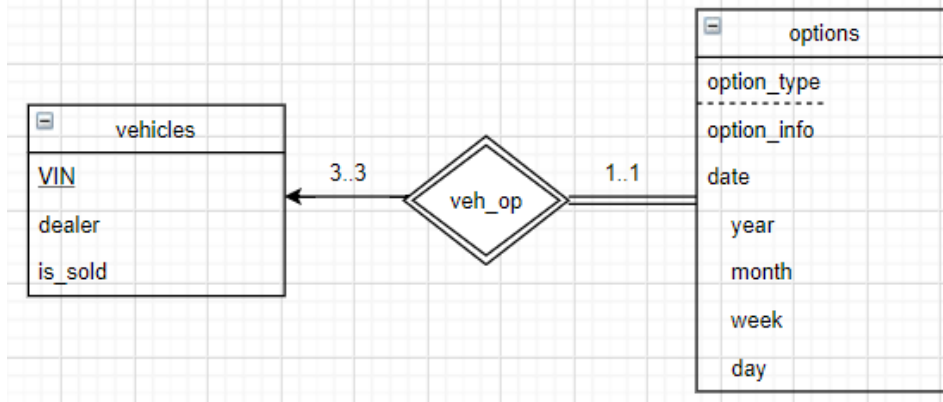
mod_veh 는 models 와 vehicles 의 관계를 나타낸다. 하나의 model 은 여러 개의 vehicle 을 가지며, vehicle 은 하나의 model 에 속하므로 one to many 관계이고, 모든 vehicle 은 하나의 model 과 관계를 맺어야 하므로 이 관계에 전체적으로 참가한다. vehicle 은 model 과 무조건 하나 연결되므로 cardinality 가 1..1, model 은 vehicle 을 0 부터 무한 개까지 가질 수 있으므로 cardinality 가 0..* 이다.



1.3.6 veh_op

veh_op 은 vehicles 와 options 의 관계를 나타낸다. 하나의 vehicle 은 color, engine, transmission 으로 총 3 개의 option 을 가지고 있고, 하나의 option 은 하나의 차량에 포함된 정보로 vehicle 하나와 관계를 맺으므로 one-to-many 관계이다. 모든 option 은 vehicle 과 관계를 맺어야 하므로 이 관계에 전체적으로 참가한다. vehicle 은 option 과 무조건 3 개 연결되므로 cardinality 가 3..3, option 은 vehicle 과 무조건 하나 연결되므로 cardinality 가 1..1 이다.

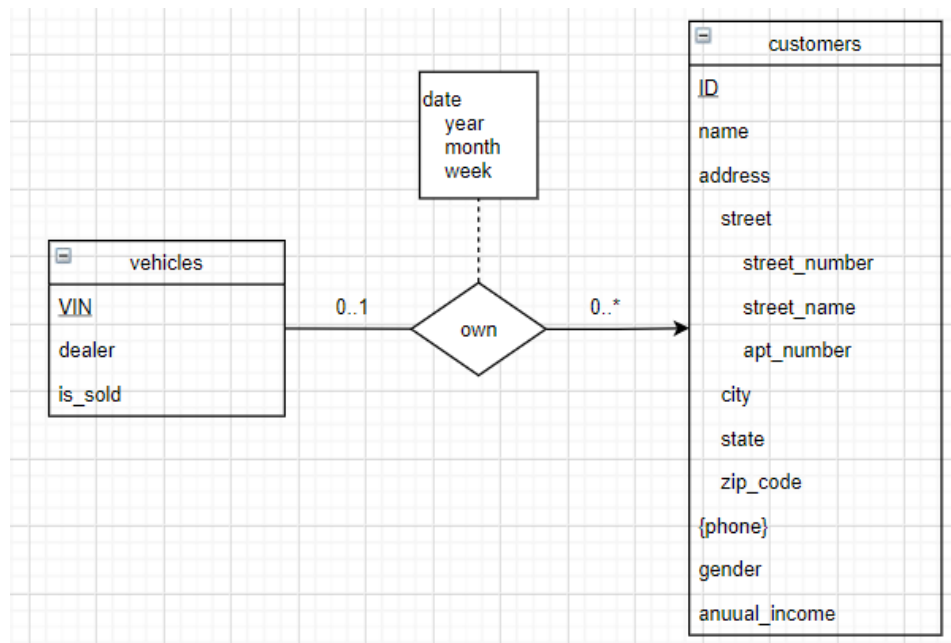
또한 options 는 option 개체 내부에 primary key 를 형성하기 위한 충분한 속성을 지니고 있지 않다. options 는 차량에 속하는 속성으로도 볼 수 있기 때문에 VIN 으로 식별이 가능한데, VIN 은 vehicle 의 primary key 이므로 options 가 vehicles 에 dependent 한 weak entity 가 되도록 나타낸다.



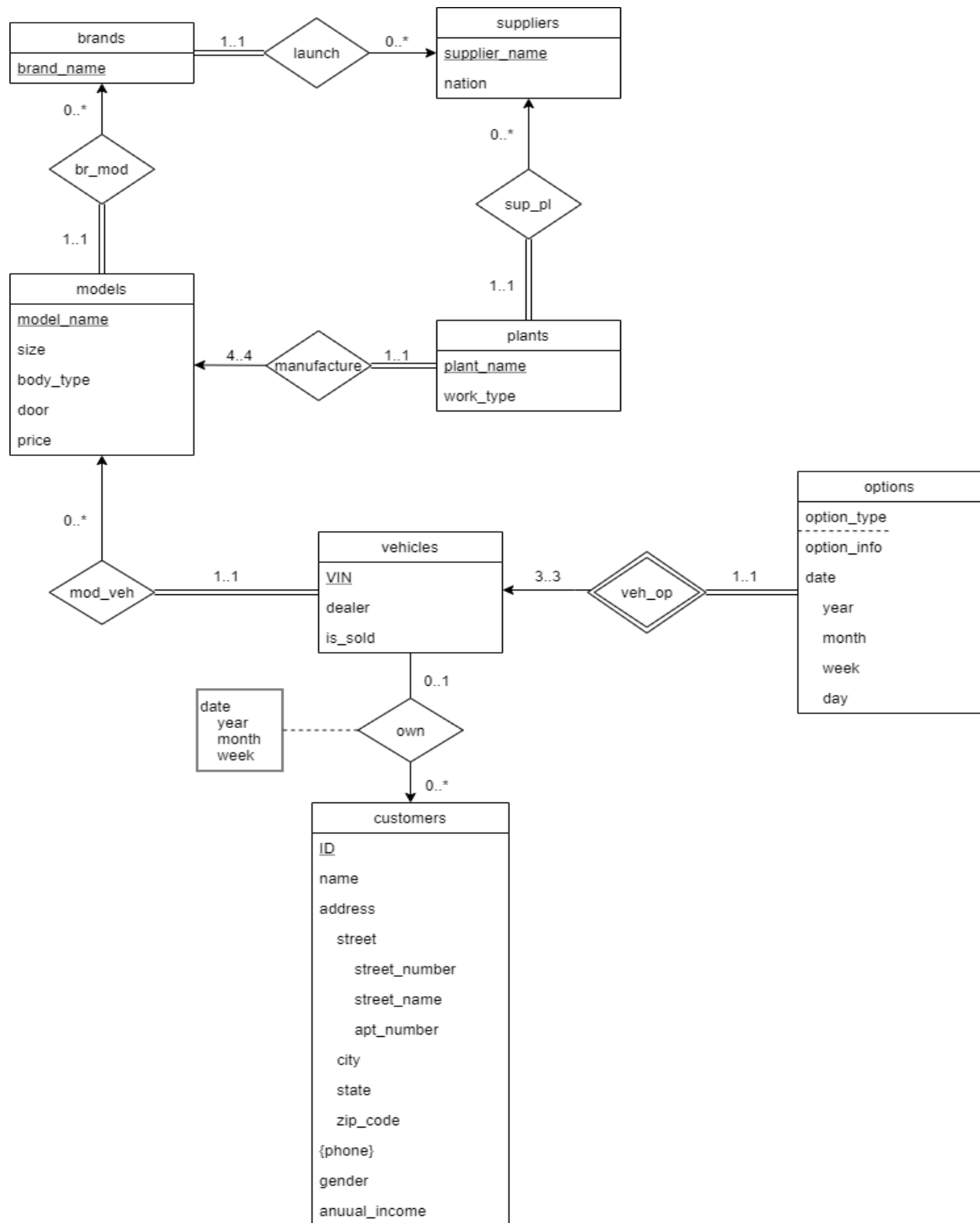
1.3.7 own

own 은 vehicles 와 customers 의 관계를 나타낸다. 하나의 vehicle 는 한 명의 customer 과 관계를 맺지만 한 명의 customer 은 여러 vehicle 을 소유하고 있을 수 있다. 따라서 many-to-one 의 관계다. vehicle 은 아직 팔리지 않은 경우에 고객과 연결되지 않거나 내지는 한 명의 고객과 연결되어 있는 것이 전체 경우의 수이기 때문에 cardinality 는 0..1 이고, customer 은 vehicle 을 가지고 있지 않거나 무한 대의 차를 소유할 수 있기 때문에 cardinality 가 0..*이다.

또한 이 관계에서 추가적으로 고객이 해당 차량을 구매한 날짜에 대한 정보를 저장할 수 있도록 하기 위해 date(year, month, week) 속성을 관계에 점선으로 연결해 주었다.



1.4 Visualization



2. Relational Schema

2.1 E-R model 에서 Relational Schema 로의 변환

2.1.1 Strong entity set with simple attributes

E-R model 에서 표현한 속성을 모두 지는 스키마로 나타낸다.

vehicles(VIN, dealer, is_sold)
brands(brand_name)
models(model_name size, body_type, door, price)
suppliers(supplier_name, nation)
plants(plant_name, work_type)

2.1.2 Strong entity set with Composite Attributes

복합 속성은 속성들을 단층화하고, 다중 값 속성은 새로운 스키마를 만든다.

customers(ID, name, street_number, street_name, apt_number, city, state, zip_code, gender, annual_income)
cust_phone(ID, phone_number)

2.1.3 Weak entity set

의존하는 strong entity 의 primary key 를 해당 weak entity 의 primary key set 에 포함시키고 릴레이션 스키마를 표현한다.

options(VIN, option_type, option_info, year, month, week, day)

2.1.4 Relationship set

관계 집합을 구성하는 개체들의 primary key 들과 해당 relation 의 추가 속성을 포함하여 relation schema 를 표현한다.

primary key 는 다음과 같이 설정한다.

one-to-one : 한 쪽 개체 집합의 primary key 를 임의로 선택

one-to-many or many-to-one : many 인 부분의 개체 집합의 primary key 를 선택

many-to-many : 개체 집합에 참여하는 속성들의 primary key 의 합집합.

launch(brand_name, supplier_name)
br_mod(model_name, brand_name)
sup_pl(plant_name, supplier_name)
manufacture(plant_name, model_name)
mod_ven(VIN, model_name)
veh_op(VIN, option_type)
own(VIN, ID)

2.2 스키마의 중복성

2.2.1 veh_op

관계 집합 veh_op(VIN, option_type)와 개체 집합 options(VIN, option_type, option_info, year, month, week, day)를 보면 veh_op의 정보가 options에 모두 포함되어 있다는 것을 알 수 있다. veh_op가 불필요하게 중복되었으므로 테이블에 나타내지 않는다.

2.3 스키마의 조합

many-to-one 혹은 one-to-many 관계에서 many에 해당하는 개체가 해당 관계에 전체적으로 참가하면 이를 조합할 수 있다.

2.3.1 launch

brands와 suppliers는 many-to-one의 관계이고, brands가 launch 관계에 전체적으로 참가하므로 조합이 가능하다.

brands & launch

→ brands(brand_name, supplier_name)

2.3.2 br_mod

models와 brands는 many-to-one의 관계이고, models가 br_mod 관계에 전체적으로 참가하므로 조합이 가능하다.

models & br_mods

→ models(model_name, brands_name, size, body_type, door, price)

2.3.3 sup_pl

plants와 suppliers는 many-to-one의 관계이고, plants가 sup_pl 관계에 전체적으로 참가하므로 조합이 가능하다.

plants & sup_pl

→ plants(plant_name, work_type, supplier_name)

2.3.4 mod_veh

vehicles와 models는 many-to-one의 관계이고, vehicles가 mod_veh 관계에 전체적으로 참가하므로 조합이 가능하다.

vehicles & mod_veh

→ vehicles(VIN, model_name, dealer, is_sold)

2.3.5 manufacture

plants와 models는 many-to-one의 관계이고, plants가 manufacture 관계에 전체적으로 참가하므로 조합이 가능하다.

plants & manufacture

➔ plants(plant_name, model_name, work_type, supplier_name)

2.4 Visualization

