과목명: 데이터베이스시스템

담당 교수명: 정성원

<<Project 1>>

**서강대학교 컴퓨터학과**

**[20171630]**

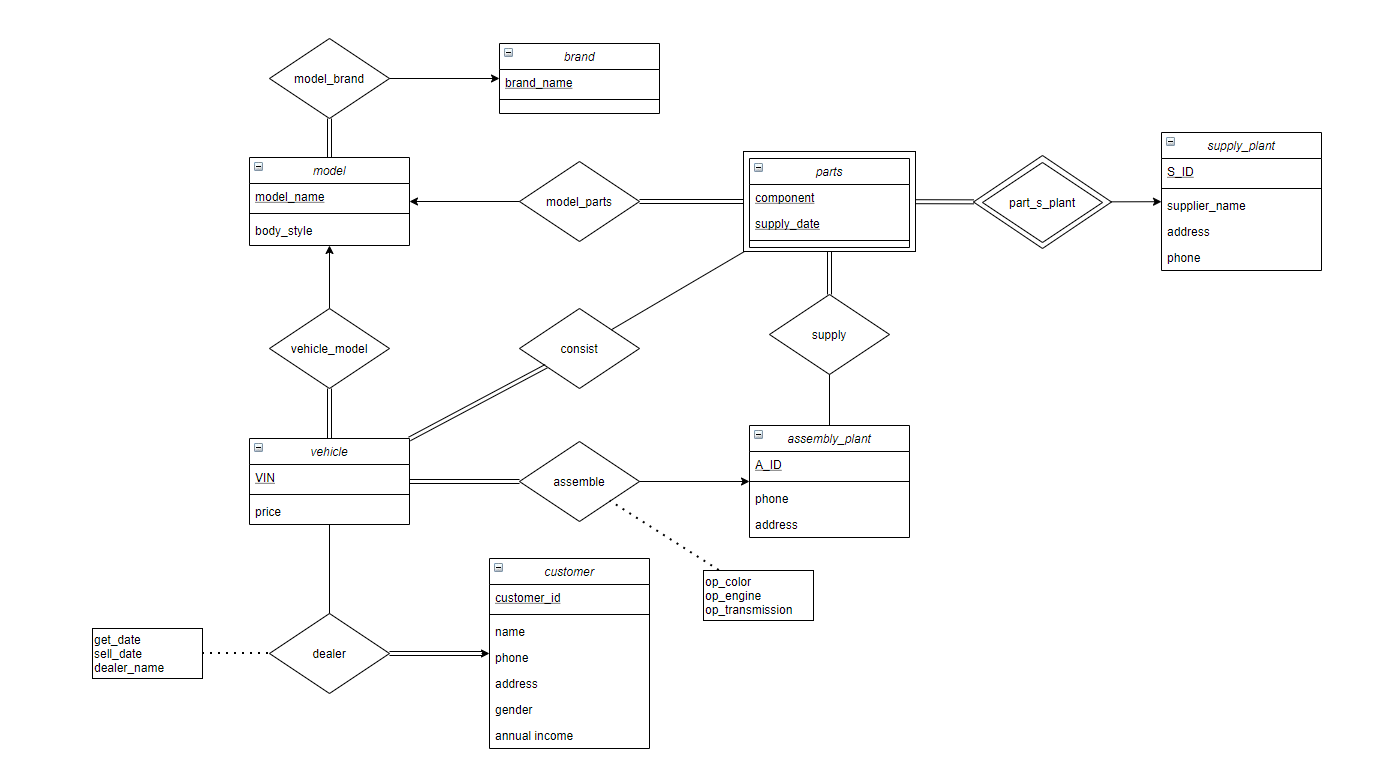
**[남주형]**

**1. 프로젝트 개요**

가상의 자동차 회사의 DBA가 되어 회사 운영의 기반이 되는 데이터베이스의 주요 부분을 재설계한다

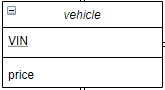
**2. E-R Diagram**

**1) Diagram**



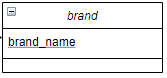
**2) Entity**

* vehicle



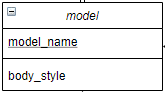
차량의 vehicle identification number (VIN)와 차량의 가격을 저장하고 있다. VIN만으로 구분이 가능하므로 primary key는 VIN이 된다.

* brand



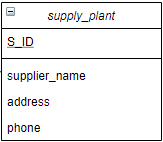
brand의 이름을 저장하고 있다. brand\_name만으로 구분이 가능하므로 primary key는 brand\_name이 된다.

* model



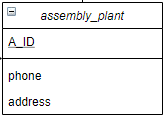
model의 이름과 body\_style에 대한 정보를 저장하고 있다. model\_name만으로 구분이 가능하므로 primary key는 model\_name이 된다.

* supply\_plant



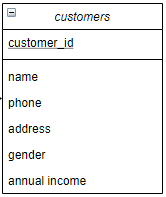
공급 공장의 ID와 공급 업체의 이름, 주소, 전화번호를 저장하고 있다. 여기서 공급 업체의 이름에는 공급 업체가 있을 경우 공급 업체의 이름을 저장하고 자신의 회사가 소유한 공장이면 자신의 회사 이름을 저장한다. 공장의 ID만으로 구별이 가능하므로 S\_ID를 primary key로 설정해주었다.

* assembly\_plant



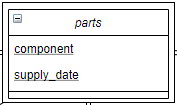
조립 공장의 ID와 주소와 번호에 대한 정보를 저장하고 있다. A\_ID만으로 구분이 가능하므로 primary key는 A\_ID이 된다.

* customers



고객의 ID, 이름, 전화번호, 주소, 성별, 연소득에 대한 정보를 저장하고 있다. 고객의 고유한 ID로 구분이 가능하므로 primary key는 customer\_id가 된다.

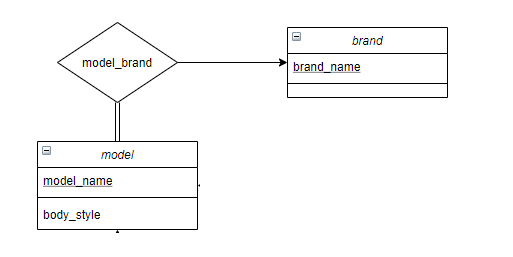
* parts



부품과 공급날짜와 공급 공장의 ID를 저장하고 있다. weak entity로 공급공장의 id는 supply\_plant entity로 부터 받아온다. 세개의 attribute가 다 있어야 구분이 가능하므로 component, supply\_date, S\_ID가 primary key이다.

**3) Relationship**

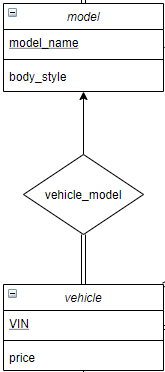
* brand - model\_brand - model



brand는 여러개의 model을 가질 수 있지만 model은 여러개의 brand를 가질 수 없으므로 brand와 model은 one to many 관계이다.

또 모든 model은 relationship set에서 적어도 하나의 relationship에 참여 하므로 inst\_brand에서 model의 참여는 total이다. 이에 반해 일부 brand는 relationship set에서 어떤 relationship에도 참여 하지 않을 수 있으므로 inst\_brand에서 brand의 참여는 partial이다.

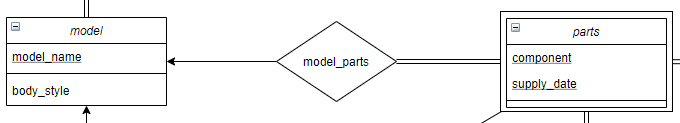
* vehicle - vehicle\_model - model



model는 여러개의 vehicle을 가질 수 있지만 vehicle은 여러개의 model를 가질 수 없으므로 model과 vehicle은 one to many 관계이다.

또 모든 vehicle은 relationship set에서 적어도 하나의 relationship에 참여 하므로 inst\_model에서 vehicle의 참여는 total이다. 이에 반해 일부 model는 relationship set에서 어떤 relationship에도 참여 하지 않을 수 있으므로 inst\_model에서 model의 참여는 partial이다.

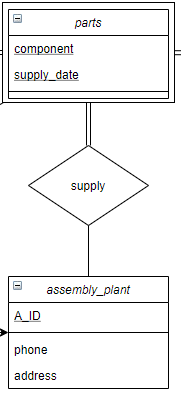
* model - model\_parts- parts



model는 여러개의 parts을 가질 수 있지만 parts은 여러개의 model를 가질 수 없으므로 model와 parts은 one to many 관계이다.

또 모든 parts은 relationship set에서 적어도 하나의 relationship에 참여 하므로 model\_parts에서 parts의 참여는 total이다. 이에 반해 일부 model는 relationship set에서 어떤 relationship에도 참여 하지 않을 수 있으므로 model\_parts에서 model의 참여는 partial이다.

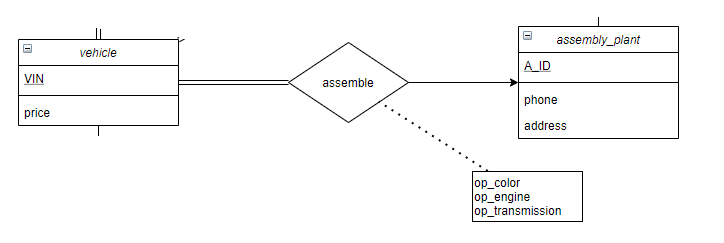
* parts - supply - assembly\_plant



하나의 parts 가 여러개의 assembly\_plant에 공급될 수 있고 하나의 assembly\_plant가 여러개의 parts를 공급 받을 수 있으므로 many to many 관계이다.

또 공급 받는 공장인 assembly\_plant는 부품을 공급받지 않는 경우도 있기 때문에 partial이고 parts는 무조건 공급해야 한다고 설정하여 total로 해주었다.

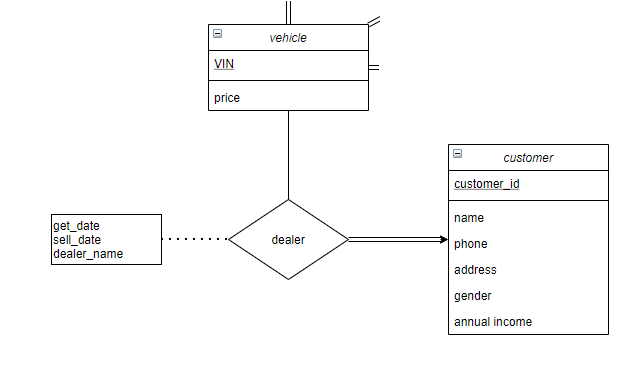
* vehicle - assemble - assembly\_plant



하나의 assembly\_plant에서 여러개의 vehicle을 조립할 수 있고 하나의 vehicle은 여러개의 assembly\_plant에서 조립될 수 없으므로 vehicle과 assembly\_plant는 many to one 관계이다. assemble하는 과정에서 option인 color, engine, transmission의 정보를 저장하기 위해 attributes를 추가해주었다.

또 모든 vehicle은 relationship set에서 적어도 하나의 relationship에 참여 하므로 assemble에서 vehicle의 참여는 total이다. 이에 반해 일부 assembly\_plant는 relationship set에서 어떤 relationship에도 참여 하지 않을 수 있으므로 assemble에서 assembly\_plant의 참여는 partial이다.

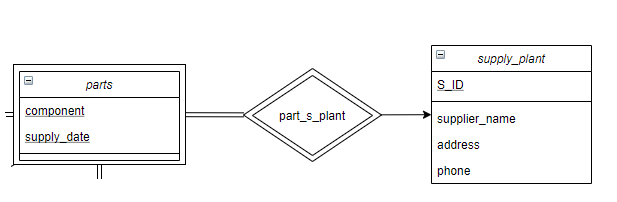
* vehicle - dealer - customer



하나의 customer은 여러개의 vehicle을 구매할 수 있고 하나의 vehicle은 여러명의 custormer에게 판매될 수 없으므로 vehicle과 customer는 many to one의 관계이다. 여기서 get\_date는 dealer가 vehicle을 얻은 날짜이고 sell\_date는 dealer가 vehicle을 판매한 날짜이다. vehicle이 판매 되지 않았다면 costomer와 sell\_date 정보가 null값이 들어간다. 그리고 dealer의 이름이 attribute로 추가된다.

또 모든 customer은 relationship set에서 적어도 하나의 relationship에 참여 하므로 dealer에서 customer의 참여는 total이다. 이에 반해 일부 vehicle는 relationship set에서 어떤 relationship에도 참여 하지 않을 수 있으므로 dealer에서 vehicle의 참여는 partial이다.

* parts - par\_s\_plant - supply\_plant

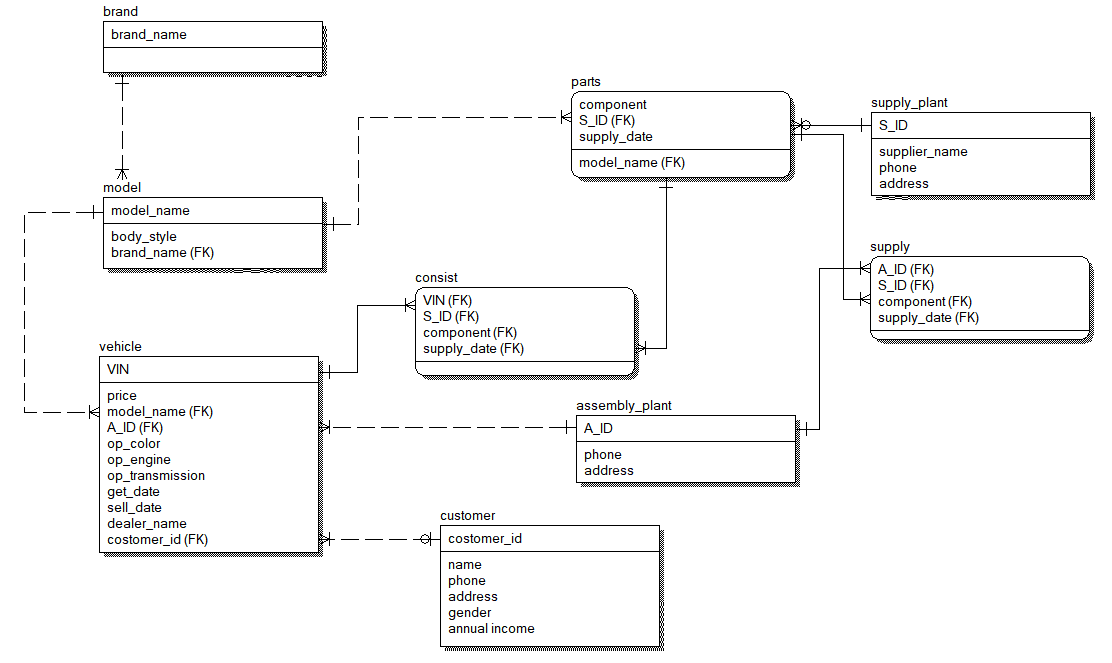


이 relationship은 weak entity set을 identifying strong entity set으로 연결하는 relationship이다. 하나의 공장이 여러개의 부품을 생산할 수 있고 하나의 부품이 여러 공장에서 생성될 수 없으므로 many to one의 관계이다.

parts는 무조건 공장에서 생성되어야 하므로 total이고 공장은 parts를 생산하지 않을 수도 있으므로 partial이다.

**3. Schema diagram(erwin-logical mode)**

**1) Diagram**



**2) Entity**

이론) E-R Diagram을 Schema diagram으로 바꾸는 과정에서 one to many 관계는 one 쪽 entity의 primary key를 many 쪽 entity의 attribute로 넣어주고 foreign key로 설정해준다. many to many 관계는 새로운 entity를 생성하여 양쪽의 primary key를 이 entity의 primary key로 설정해준다. 이와 같은 방법으로 아래의 entity들이 완성되었다.

* vehicle

|  |  |  |
| --- | --- | --- |
| **attribute 조건** | **attribute 이름** | **Domain** |
| 차량 식별 번호(PK) | VIN | Number |
| 가격 | price | Number |
| model 이름(FK) | model\_name | String |
| 조립 공장 이름(FK) | A\_ID | String |
| 색깔 옵션 | op\_color | String |
| 엔진 옵션 | op\_engine | String |
| 변속기 옵션 | op\_transmission | String |
| 얻은(만들어진) 날짜 | get\_date | Datetime |
| 팔린 날짜 | sell\_date | Datetime |
| 딜러 이름 | dealer\_name | String |
| 고객 ID(FK) | costomer\_id | String |

* brand

|  |  |  |
| --- | --- | --- |
| **attribute 조건** | **attribute 이름** | **Domain** |
| brand 이름(PK) | brand\_name | String |

* model

|  |  |  |
| --- | --- | --- |
| **attribute 조건** | **attribute 이름** | **Domain** |
| model 이름(PK) | model\_name | String |
| 차체 스타일 | body\_style | String |
| brand 이름(FK) | brand\_name | String |

* supply\_plant

|  |  |  |
| --- | --- | --- |
| **attribute 조건** | **attribute 이름** | **Domain** |
| 공급 공장 이름(PK) | S\_ID | String |
| 공급 업체 이름 | supplier\_name | String |
| 전화 번호 | phone | Number |
| 주소 | address | String |

* assembly\_plant

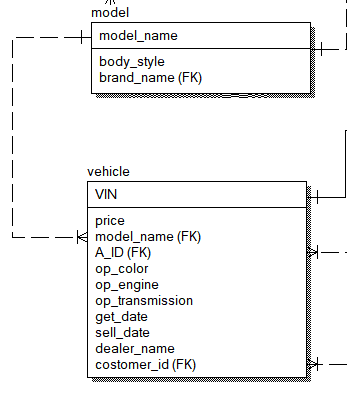
|  |  |  |
| --- | --- | --- |
| **attribute 조건** | **attribute 이름** | **Domain** |
| 조립 공장 이름(PK) | A\_ID | String |
| 전화 번호 | phone | Number |
| 주소 | address | String |

* supply

|  |  |  |
| --- | --- | --- |
| **attribute 조건** | **attribute 이름** | **Domain** |
| 부품(PK,FK) | component | String |
| 공급 공장 이름(PK,FK) | S\_ID | String |
| 조립 공장 이름(PK,FK) | A\_ID | String |
| 공급 일자(PK,FK) | supply\_date | Datetime |

**3) Relation**

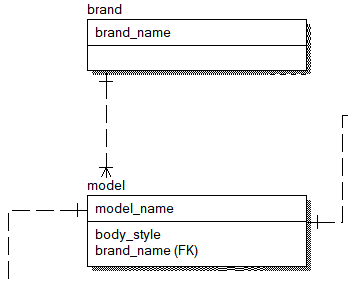
* model to vehicle



Cardinality: One-to-One-or-More

model과 vehicle사이의 관계를 나타낸 것이다. 한 개의 model이 한 개 이상의 vehicle을 포함하고 있으므로 one to one or more의 cardinality를 가진다.

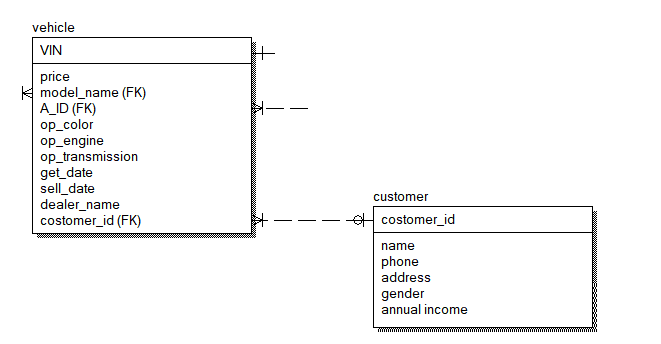
* brand to model



Cardinality: One-to-One-or-More

brand와 model사이의 관계를 나타낸 것이다. 한 개의 brand가 한 개 이상의 model을 포함하고 있으므로 one to one or more의 cardinality를 가진다.

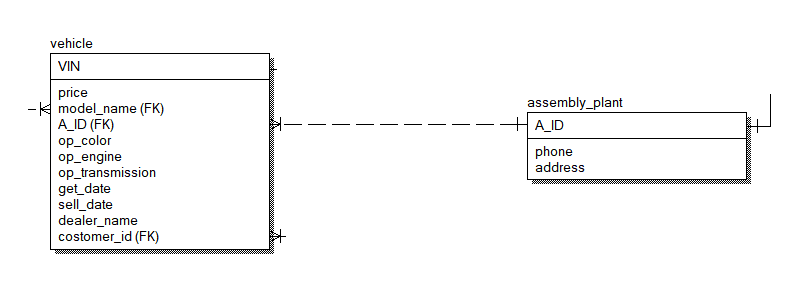
* customer to vehicle



Cardinality: Zero-or-One-to-One-or-More

customer과 vehicle사이의 관계를 나타낸 것이다. customer가 null이냐 아니냐에 따라 vehicle이 팔렸나 안팔렸나를 알 수 있으므로 한 개의 customer 가 한 개 이상의 model을 포함하고 있거나 영 개의 customer가 한 개 이상의 model을 포함하고 있다. 따라서 zero or one to one or more의 cardinality를 가진다.

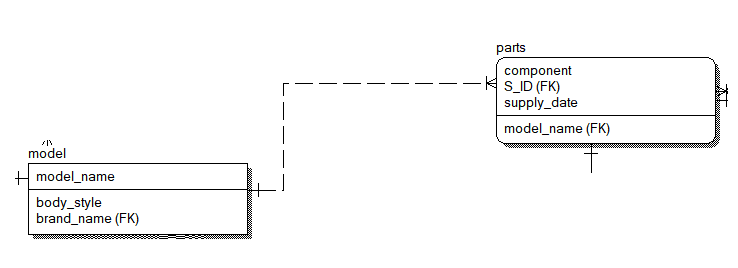
* assembly\_plant to vehicle



Cardinality: One-to-One-or-More

assembly\_plant와 vehicle사이의 관계를 나타낸 것이다. 한 개의 assembly\_plant가 한 개 이상의 vehicle을 포함하고 있으므로 one to one or more의 cardinality를 가진다.

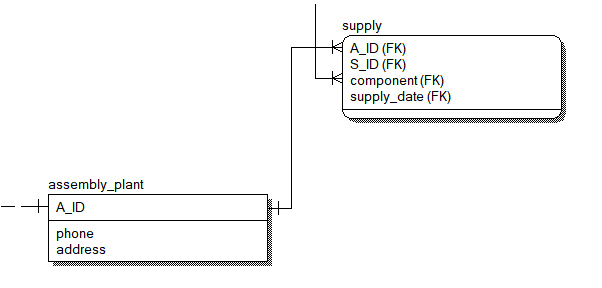
* model to parts



Cardinality: One-to-One-or-More

model과 parts사이의 관계를 나타낸 것이다. 한 개의 model이 한 개 이상의 parts을 포함하고 있으므로 one to one or more의 cardinality를 가진다.

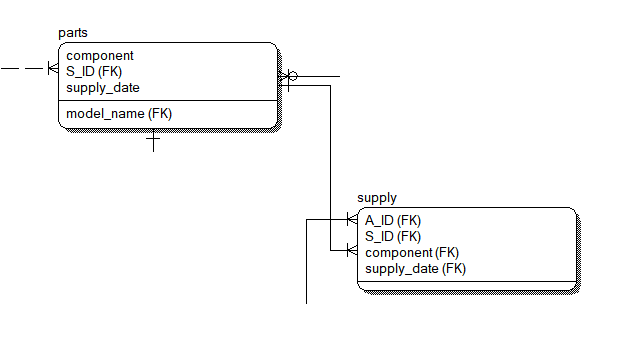
* assembly\_plant to supply



Cardinality: One-to-One-or-More

assembly\_plant과 supply사이의 관계를 나타낸 것이다. 한 개의 assembly\_plant가 한 개 이상의 supply을 포함하고 있으므로 one to one or more의 cardinality를 가진다.

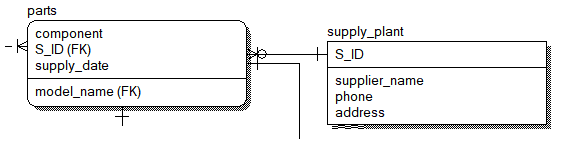
* parts to supply



Cardinality: One-to-One-or-More

parts와 supply사이의 관계를 나타낸 것이다. 한 개의 parts가 한 개 이상의 supply을 포함하고 있으므로 one to one or more의 cardinality를 가진다.

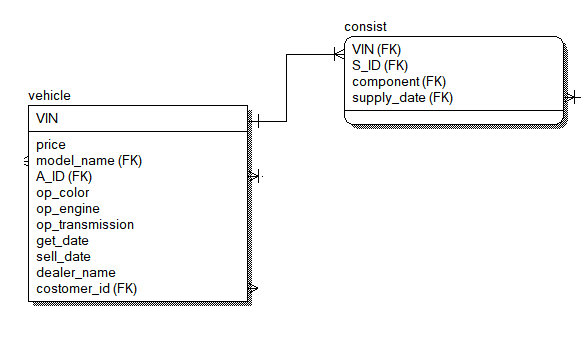
* supply\_plant to parts



Cardinality: One-to-Zero-One-or-More

supply\_plant와 parts사이의 관계를 나타낸 것이다. 한 개의 supply\_plant가 0개 또는 한 개 이상의 parts을 포함하고 있으므로 one to zero, one or more의 cardinality를 가진다.

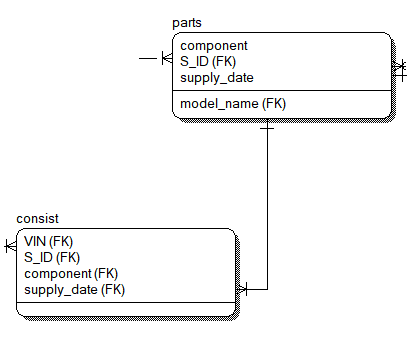
* vehicle to consist



Cardinality: One-to-One-or-More

vehicle와 parts사이의 관계를 나타낸 것이다. 한 개의 vehicle가 한 개 이상의 consist을 포함하고 있으므로 one to one or more의 cardinality를 가진다.

* parts to consist



Cardinality: One-to-One-or-More

parts와 consist사이의 관계를 나타낸 것이다. 한 개의 parts가 한 개 이상의 consist을 포함하고 있으므로 one to one or more의 cardinality를 가진다.