

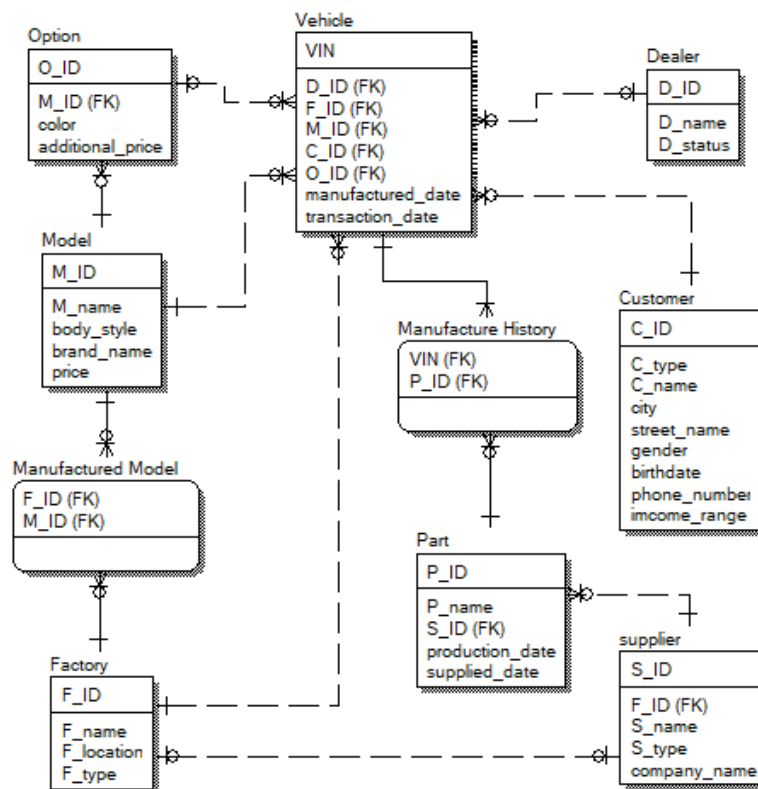
Database System Project 2

Normalization and Query Processing

20171678 이찬희

BCNF Decomposition

Decomposition을 진행하기 전, 기존의 schema diagram은 다음과 같다.



이 중 BCNF를 만족하지 않는 스키마는 Model(M_ID, M_name, body_style, brand_name, price)이며 이유는 다음과 같다.

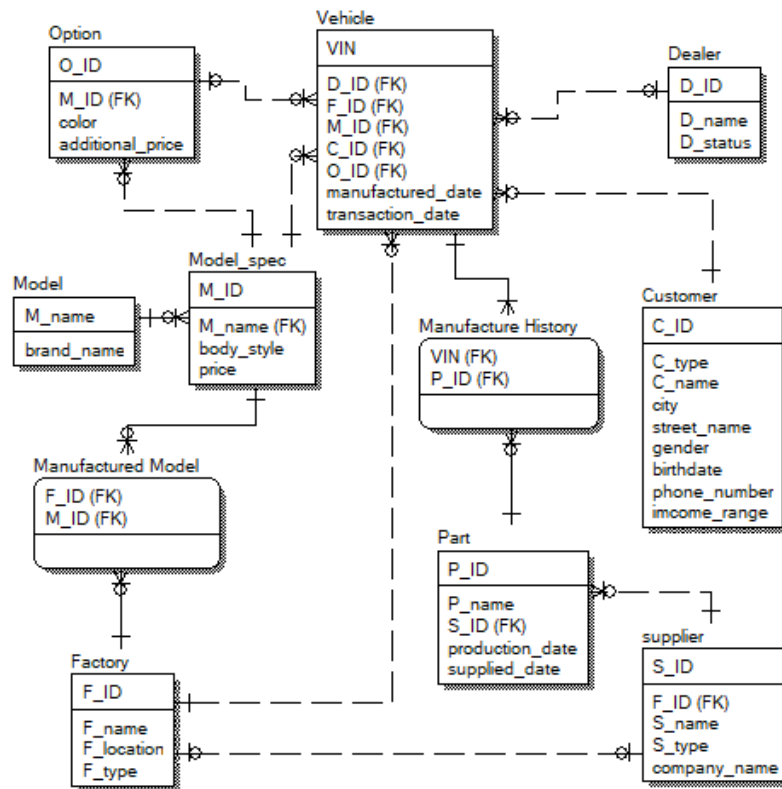
- M_name → brand_name 이 성립하지만, M_name 이 superkey가 아니다.

이를 고려하여 decompose 하면, Model은 다음의 두 스키마로 분리될 수 있다.

- Model_spec(M_ID, M_name, body_style, price)
- Model(M_name, brand_name)

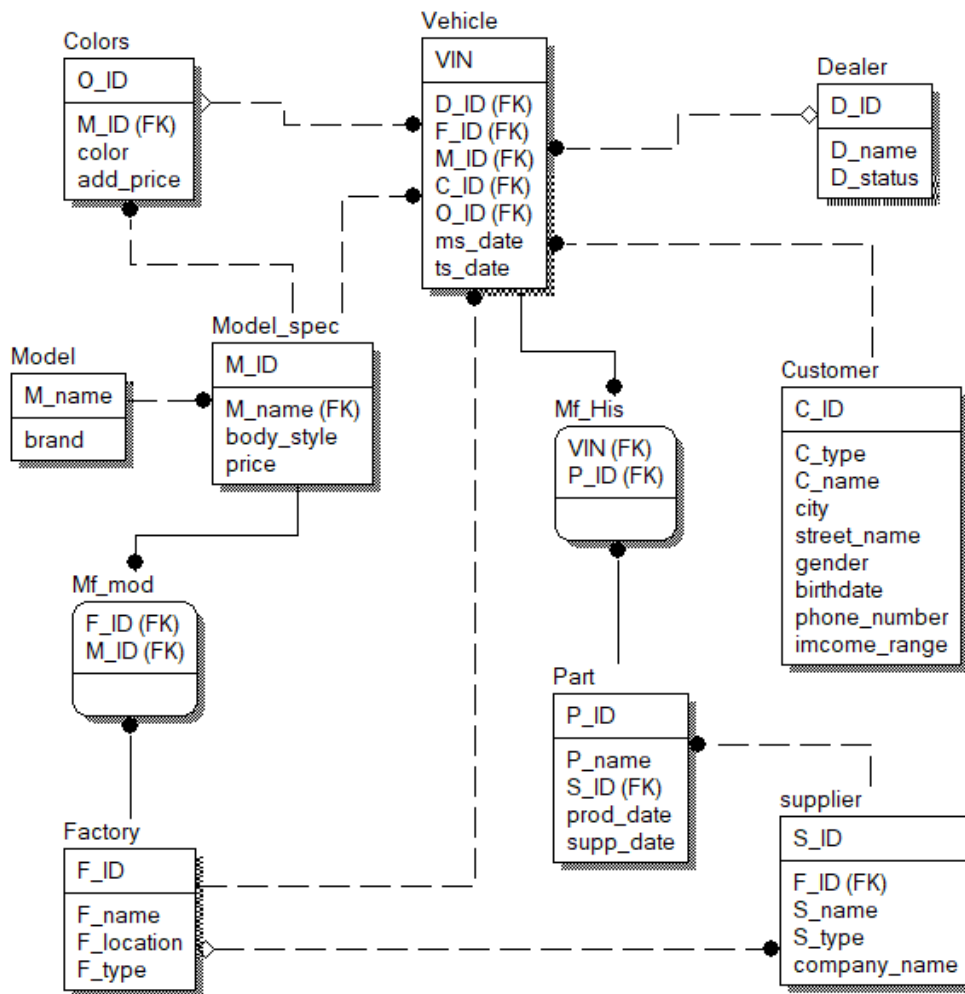
이 외에 M_name, body_style → price 는 가격에 변동이 있는 상황이 있을 수 있다고 판단하여 성립하지 않는다고 가정하였다.

이를 토대로 다시 그린 Logical schema diagram은 다음과 같다.



모든 상세 모델(Model_spec)은 하나의 모델을 가지지만, 한 모델은 여러개의 상세 모델과 연결될 수 있고 (같은 모델에서 서로 다른 body_type과 price를 지니는 상세 모델들이 출시되는 경우), 연결되지 않을 수도 있으므로 (모델이 아직 이름만 존재하며, body_type이나 price가 정해지지 않아 상세 모델이 존재하지 않는 경우) Model 과 Model_spec 사이의 relationship은 one-to-zero-one-or-more relationship이다.

Physical Schema Diagram



Logical schema diagram을 schema diagram으로 옮기면서 너무 entity 이름이나 entity set 이름등을 축약하였다. Option 역시 테이블 이름으로 사용할 수 없어 color로 변경되었다.

- Option
 - (PK) O_ID
숫자로 고유한 ID값을 나타내기 위해 type은 numeric(8,0)으로 설정하였다.
PK이므로 null은 허용되지 않는다.
 - (FK) M_ID
 - color
색상의 이름을 저장해야하므로 type은 varchar(20)으로 설정하였다.
색상의 이름이 없는 경우 테이블에 저장할 이유가 없으므로 null은 허용되지 않는다.
 - add_price
추가 금액을 나타내기 위해 type은 numeric(10,0)으로 설정하였다.

추가 금액이므로 금액이 음수일 수 있다고 판단하여 값의 크기에 대한 제약은 걸지 않았으며, null은 허용되지 않는다.

- Model

- (PK) M_name

모델의 이름을 나타내므로 type은 varchar(30)으로 설정하였다.

PK이므로 null은 허용되지 않는다.

- brand

브랜드의 이름을 나타내므로 type은 varchar(30)으로 설정하였다.

모든 모델은 해당 모델이 출시된 브랜드를 가져야하므로 null은 허용되지 않는다.

- Model_spec

- (PK) M_ID

숫자로 고유한 ID값을 나타내기 위해 type은 numeric(8,0)으로 설정하였다.

PK이므로 null은 허용되지 않는다.

- (FK) M_name

모든 모델은 모델명을 가지므로 null은 허용되지 않는다.

- body_style

모델의 body style로, varchar(20)으로 설정하였다. 'default', '4-door', 'wagon' 세 값 중 하나를 지닐 수 있으며, null은 허용되지 않는다.

- price

금액을 나타내므로 varchar(10,0)으로 설정하였다. 기본 금액이므로 항상 양수여야 하며, null은 허용되지 않는다.

- Factory

- (PK) F_ID

숫자로 고유한 ID값을 나타내기 위해 type은 numeric(8,0)으로 설정하였다.

PK이므로 null은 허용되지 않는다.

- F_name

공장의 이름을 나타내기 위해 type은 varchar(20)으로 설정하였다. 공장의 이름을 모르더라도 F_ID로 구분할 수 있으며, 필수적인 값이 아니므로 null이 허용된다.

- F_location

공장의 위치를 나타내기 위해 type은 varchar(20)으로 설정하였다. 부수적인 정보이므로 null이 허용된다.

- F_type

공장이 조립을 담당하는지, 부품 제조를 담당하는지 값을 저장하며, type은 varchar(15)로 설정하였다. 'assemble' 또는 'manufacture' 두 값 중 하나를 가진다.

- Supplier

- (PK) S_ID
숫자로 고유한 ID값을 나타내기 위해 type은 numeric(8,0)으로 설정하였다.
PK이므로 null은 허용되지 않는다.
- (FK) F_ID
- S_name
Supplier의 이름을 나타내기 위해 type은 varchar(20)으로 설정하였다. 부수적인 정보이므로 null이 허용된다.
- S_type
Supplier가 자사 소속인지, 타사 소속인지를 나타내는 값으로, type은 varchar(15)로 설정하였다.
'company' 또는 'third-party' 두 값 중 하나를 가진다.
- company_name
타사 소속 supplier의 경우 기업명을 저장하기 위한 column으로, type은 varchar(20)으로 설정하였다. 자사 소속의 경우 따로 저장하는 값이 없으므로 null이 허용된다.
- Part
 - (PK) P_ID
숫자로 고유한 ID값을 나타내기 위해 type은 numeric(8,0)으로 설정하였다.
PK이므로 null은 허용되지 않는다.
 - P_name
부품의 이름을 나타내기 위해 type은 varchar(20)으로 설정하였고, 부수적인 정보이므로 null이 허용된다.
 - (FK) S_ID
 - prod_date
날짜값이므로 type은 date으로 설정하였으며, 필수적인 정보이므로 null이 허용되지 않는다.
 - supp_date
날짜값이므로 type은 date으로 설정하였으며, 필수적인 정보이므로 null이 허용되지 않는다.
- Customer
 - (PK) C_ID
숫자로 고유한 ID값을 나타내기 위해 type은 numeric(8,0)으로 설정하였다.
PK이므로 null은 허용되지 않는다.
 - C_name
고객의 이름을 나타내기 위해 type은 varchar(20)으로 설정하였으며, 부수적인 정보이므로 null이 허용된다.
 - C_type
개인 고객과 기업 고객을 구분하기 위해 'individual', 'business' 두 값 중 하나를 가지며
varchar(10)으로 설정하였다.

- city
고객의 주소를 나타내기 위해 type은 varchar(20)으로 설정하였으며, 부수적인 정보이므로 null이 허용된다.
- street
고객의 주소를 나타내기 위해 type은 varchar(20)으로 설정하였으며, 부수적인 정보이므로 null이 허용된다.
- gender
고객의 성별을 나타내기 위해 type은 varchar(20)으로 설정하였으며, 부수적인 정보이므로 null이 허용된다.
- birthdate
날짜값이므로 date로 설정하였으며, 부수적인 정보이므로 null이 허용된다.
- phon_number
고객의 연락처를 나타내기 위해 type은 varchar(20)으로 설정하였으며, 부수적인 정보이므로 null이 허용된다.
- income_range
고객의 연봉을 나타내기 위해 type은 numeric(10, 0)으로 설정하였으며, 부수적인 정보이므로 null이 허용된다.
- Dealer
 - (PK) D_ID
숫자로 고유한 ID값을 나타내기 위해 type은 numeric(8,0)으로 설정하였다.
PK이므로 null은 허용되지 않는다.
 - D_name
딜러의 이름을 나타내기 위해 type은 varchar(20)으로 설정하였으며, 부수적인 정보이므로 null이 허용된다.
 - D_status
현재 영업중인지 여부를 나타내기 위해 'open', 'closed' 두 값 중 하나를 가지며, varchar(10)으로 설정하였다.
- Vehicle
 - (PK) VIN
숫자로 고유한 ID값을 나타내기 위해 type은 numeric(10,0)으로 설정하였다.
PK이므로 null은 허용되지 않는다.
 - (FK) F_ID
모든 차량은 해당 차량이 생산된 공장값을 가지므로, null이 허용되지 않는다.
 - (FK) M_ID
모든 차량은 특정 모델에 해당하므로, null이 허용되지 않는다.

- (FK) C_ID
아직 거래되지 않았을 수 있으므로, null이 허용된다.
- (FK) D_ID
아직 딜러가 배정되지 않았을 수 있으므로, null이 허용된다.
- (FK) O_ID
추가적인 옵션이 선택되지 않았을 수 있으므로, null이 허용된다.
- mf_date
날짜값이므로 date로 설정하였으며, 생산일은 필수적인 값으로, null이 허용되지 않는다.
- ts_date
날짜값이므로 date로 설정하였으며, 아직 거래되지 않았을 수 있으므로, null이 허용된다.

Query

명세서에서 요구한 13개의 쿼리를 모두 프로그램 상에서 구현하였다.

Code implementation

MySQL server 8.0 을 사용하여 database를 구축하였으며, ODBC를 사용하여 C++ 코드로 DBMS에 연결하고 쿼리를 실행하였다.

Initailization

```
MySQL Succesfully Connected.
Table created.
Rows inserted.

----- Welcome! -----

** Due to insufficient amount of data, unexpected results may occur.
```

가장 먼저 MySQL 서버와 연결 한 후 txt 파일에서 쿼리를 읽어 Otable을 생성하고 data를 입력한다. 성공 하였다면, welcome message를 출력한다.

Main Menu

```
----- SELECT QUERY TYPES -----

1. TYPE 1
2. TYPE 2
3. TYPE 3
4. TYPE 4
5. TYPE 5
6. TYPE 6
7. TYPE 7
0.QUIT

Type:
```

7가지 type 중 한 가지를 선택할 수 있다. 0~7 이외의 값을 입력하는 경우 그 값은 무시되며 0을 입력하여 프로그램을 종료할 때 까지 반복적으로 출력된다.

Type 1

```

----- TYPE 1 -----
** Show the sales trends for a particular brand over the past k years.
Which Brand?
    1. Hyundai
    2. Hyundai Truck & Bus
    3. Genesis
Brand: 3
Which K?: 2

| YEAR | SALES |
-----
| 2019 | 53900000 |
| 2020 | 244170000 |

----- Subtypes in TYPE 1 -----
    1. TYPE 1-1
    0. QUIT
TYPE: 1

----- TYPE 1-1 -----
** Break these data out by gender of the buyer.

| YEAR | GENDER | SALES |
-----
| 2019 | (null) | 53900000 |
| 2020 | (null) | 136100000 |
| 2020 | Female | 108070000 |

----- Subtypes in TYPE 1-1 -----
    1. TYPE 1-1-1
    0. QUIT
TYPE: 1

----- TYPE 1-1-1 -----
** Break these data out by income range.
Enter the criteria for income (Unit: KRW 10,000): 5000

| YEAR | GENDER | INCOME RANGE | SALES |
-----
| 2019 | (null) | >= 5000 | 53900000 |
| 2020 | (null) | (null) | 95000000 |
| 2020 | (null) | >= 5000 | 41100000 |
| 2020 | Female | (null) | 53970000 |
| 2020 | Female | < 5000 | 54100000 |

```

지난 K 년간 선택한 브랜드의 sales trend를 보여준다. 브랜드와 K 값을 유저가 입력할 수 있으며, 연간 총 판매금액의 합이 출력된다. type 1-1을 선택하면 해당 판매 기록을 구매자의 성별로 나누어 출력하며, type 1-1-1을 선택하면 그 기록을 다시 income range로 나누어 출력한다. income range의 경우, 유저가 입력한 값 이상이거나 미만인 경우, 또는 null 인 경우 세 가지로 나누어 출력된다.

Type 2


```

----- TYPE 2 -----

** Show sales trends for various brands over the past k months.
Which K?: 10

<< Genesis >>

| MONTH | SALES |
-----|-----
| 2020.08 | 53900000 |
| 2020.09 | 53970000 |
| 2020.12 | 54100000 |
| 2021.01 | 53900000 |

<< Hyundai >>

| MONTH | SALES |
-----|-----
| 2021.02 | 40510000 |

<< Hyundai Truck & Bus >>

| MONTH | SALES |
-----|-----
| 2020.08 | 57680000 |
| 2020.12 | 37350000 |
| 2021.03 | 37350000 |

```

```

----- Subtypes in TYPE 2 -----

1. TYPE 2-1
0. QUIT
TYPE: 1

----- TYPE 2-1 -----

** Break these data out by gender of the buyer.

<< Genesis >>

| MONTH | GENDER | SALES |
-----|-----|-----
| 2021.01 | Male | 53900000 |
| 2020.08 | (null) | 53900000 |
| 2020.09 | Female | 53970000 |
| 2020.12 | Female | 54100000 |

<< Hyundai >>

| MONTH | GENDER | SALES |
-----|-----|-----
| 2021.02 | (null) | 15700000 |
| 2021.02 | Male | 24810000 |

<< Hyundai Truck & Bus >>

| MONTH | GENDER | SALES |
-----|-----|-----
| 2021.03 | (null) | 37350000 |
| 2020.08 | Female | 57680000 |
| 2020.12 | (null) | 37350000 |

```

```

----- Subtypes in TYPE 2-1 -----
      1. TYPE 2-1-1
      0. QUIT
TYPE: 1
----- TYPE 2-1-1 -----

** Break these data out by income range.
Enter the criteria for income (Unit: KRW 10,000): 3000

<< Genesis >>

| MONTH | GENDER | INCOME RANGE | SALES |
|-----|-----|-----|-----|
| 2021.01 | Male   | >= 3000      | 53900000 |
| 2020.08 | (null) | (null)       | 53900000 |
| 2020.09 | Female | (null)       | 53970000 |
| 2020.12 | Female | >= 3000      | 54100000 |

<< Hyundai >>

| MONTH | GENDER | INCOME RANGE | SALES |
|-----|-----|-----|-----|
| 2021.02 | (null) | >= 3000      | 15700000 |
| 2021.02 | Male   | >= 3000      | 24810000 |

<< Hyundai Truck & Bus >>

| MONTH | GENDER | INCOME RANGE | SALES |
|-----|-----|-----|-----|
| 2021.03 | (null) | (null)       | 37350000 |
| 2020.08 | Female | >= 3000      | 57680000 |
| 2020.12 | (null) | (null)       | 37350000 |

```

지난 K달간 브랜드별 sales trend를 보여준다. K 값은 유저의 입력으로 결정되며, 브랜드별로 월간 총 판매 금액의 합이 출력된다. 판매기록이 없는 달은 출력되지 않는다. type 2-1을 선택하면 해당 판매 기록을 구매자의 성별로 나누어 출력하며, type 2-1-1을 선택하면 그 기록을 다시 income range로 나누어 출력한다. income range의 경우, 유저가 입력한 값 이상이거나 미만인 경우, 또는 null 인 경우 세 가지로 나누어 출력된다.

Type 3

```

----- TYPE 3 -----
** Find that transmissions made by supplier (company name) between two given dates are defective.
Enter start date (ex. 2019-03-14): 2019-01-01
Enter end date (ex. 2019-03-14): 2021-05-30
Enter the name of the supplier (Among Supplier1 ~ Supplier15): Supplier1

| NAMES | PRODUCED DATE | SUPPLIED DATE |
-----|-----|-----|
| Part 2 | 2019-03-19 | 2019-04-01 |
| Part 1 | 2020-01-03 | 2020-01-08 |
| Part 9 | 2020-06-03 | 2020-06-13 |
| Part 15 | 2021-05-03 | 2021-05-26 |

----- Subtypes in TYPE 3 -----
1. TYPE 3-1
2. TYPE 3-2
0. QUIT
TYPE: 1

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

| NAMES | VIN | CUSTOMER | CONTACT |
-----|-----|-----|-----|
| Part 2 | 00000100000001 | Jax | 01012345678 |
| Part 2 | 00000100000002 | Kim | 01033226587 |
| Part 1 | 00000100000003 | Sam | (null) |
| Part 1 | 00000100000004 | Rose | 01012345678 |

```

```

----- TYPE 3 -----
** Find that transmissions made by supplier (company name) between two given dates are defective.
Enter start date (ex. 2019-03-14): 2019-01-01
Enter end date (ex. 2019-03-14): 2021-05-30
Enter the name of the supplier (Among Supplier1 ~ Supplier15): Supplier1

| NAMES | PRODUCED DATE | SUPPLIED DATE |
-----|-----|-----|
| Part 2 | 2019-03-19 | 2019-04-01 |
| Part 1 | 2020-01-03 | 2020-01-08 |
| Part 9 | 2020-06-03 | 2020-06-13 |
| Part 15 | 2021-05-03 | 2021-05-26 |

----- Subtypes in TYPE 3 -----
1. TYPE 3-1
2. TYPE 3-2
0. QUIT
TYPE: 2

----- TYPE 3-2 -----
** Find the dealer who sold the VIN and transmission for each vehicle containing these transmissions.

| NAMES | VIN | DEALER |
-----|-----|-----|
| Part 2 | 00000100000001 | Dealer3 |
| Part 2 | 00000100000002 | Dealer4 |
| Part 1 | 00000100000003 | Dealer5 |
| Part 1 | 00000100000004 | Dealer5 |

```

선택한 기간동안 선택한 공급자가 공급한 부품에 결함이 있다고 할 때, 그 부품의 정보를 보여준다. 본 프로젝트에서는 공급자 이름과 해당 공급자가 속한 회사의 이름을 다른 것으로 취급하므로, 공급자의 이름을 사용하여 쿼리를 실행하였다. type 3에서는 공급된 부품 정보를 부품명, 생산일, 공급일 세 가지로 나누어 보여주며, type 3-1에서는 해당 부품이 사용된 차량의 VIN과 이를 구매한 고객의 이름, 그리고 연락처를 보여준다. type 3-2에서는 해당 부품이 사용된 차량의 VIN과 이를 판매한 dealer의 이름을 보여준다.

Type 4

```
----- TYPE 4 -----  
  
** Find the top k brands by dollar-amount sold by the year.  
Which K?: 2  
Since when? (Year): 2019  
  
|          BRAND          |    SALES    |  
2019 -----  
1 |          Genesis      | 53900000    |  
2 |          Hyundai      | 31950000    |  
2020 -----  
1 |          Genesis      | 244170000   |  
2 | Hyundai Truck & Bus   | 152710000   |  
2021 -----  
1 |          Genesis      | 53900000    |  
2 |          Hyundai      | 40510000    |
```

연도별로 브랜드별 판매금액 순위를 보여준다. 몇년도부터의 통계를 보고 싶은지, 상위 몇 개의 브랜드를 보고 싶은 지 유저가 입력할 수 있다.

Type 5

```
----- TYPE 5 -----  
  
** Find the top k brands by unit sales by the year.  
Which K?: 2  
Since when? (Year): 2019  
  
|          Brand          |    Unit    |  
2019 -----  
1 |          Hyundai      | 2 |  
2 |          Genesis      | 1 |  
2020 -----  
1 |          Genesis      | 5 |  
2 | Hyundai Truck & Bus   | 3 |  
2021 -----  
1 |          Hyundai      | 2 |  
2 |          Genesis      | 1 |
```

연도별로 브랜드별 판매 대수 순위를 보여준다. 몇년도부터의 통계를 보고 싶은지, 상위 몇 개의 브랜드를 보고 싶은 지 유저가 입력할 수 있다.

Type 6

```
----- TYPE 6 -----  
  
** In what month(s) do convertibles sell best?  
Until what rank?: 5  
  
|    MONTH    |    SALES    |  
-----  
1 |    August    | 165480000   |  
2 |    January   | 152680000   |  
3 |    December  | 91450000    |  
4 |    February  | 81910000    |  
5 |    September | 53970000    |
```

월별 판매량 순위를 확인할 수 있다. 월별로 순위와 판매량을 보여주며, 상위 몇 개를 출력할 지 사용자가 결정할 수 있다.

Type 7

```
----- TYPE 7 -----  
  
** Find those dealers who keep a vehicle in inventory for the longest average time.  
Until what rank?: 3  
  
  |   NAME   | AVG. INVENTORY DATE |  
-----  
1 | Dealer10 |      204.0000 days |  
2 |  Dealer4 |      186.3333 days |  
3 |  Dealer5 |      167.2000 days |
```

차량을 판매하는데 가장 오래 걸리는 딜러들이 누구인 지 확인 할 수 있다. 차량의 inventory date는 차량 판매일과 제조일의 차로 계산하였으며, 상위 몇 개를 출력할 지 사용자가 결정할 수 있다.

Exit

```
----- SELECT QUERY TYPES -----  
  
1. TYPE 1  
2. TYPE 2  
3. TYPE 3  
4. TYPE 4  
5. TYPE 5  
6. TYPE 6  
7. TYPE 7  
0.QUIT  
Type: 0  
  
Table Dropped.
```

0을 선택하면 테이블들을 드롭하고 프로그램을 종료한다.