



명지대학교
MYONGJI UNIVERSITY

DB설계 및 구현2

팀 프로젝트

Building a HYUNDAI Motors Database System

‘Project MileStrone’

학 과 : 융합소프트웨어학부

팀 원 : 60191651 류채현

60141540 이규연

60191674 이상은

60171662 차유미

목차

I. 프로젝트 제안서 :: Project Proposal.....	3
1. 서론	3
2. 회사 설명.....	3
3. 새로운 DB 가 필요한 이유	4
4. 프로젝트를 진행하기 위해 알아야할 정보.....	4
II. 고객 요구사항 :: User Requirement.....	5
III. 개념 모델링 :: conceptual modeling	7
1. Entity Relation Model.....	7
2. How to make E-R Model.....	7
IV. 논리 모델링 :: Logical Modeling	9
1. Relational data model with Text notation	9
2. How to make Relational data model	10
V. Implementation in Oracle DBMS.....	11
1. Table Information.....	11
2. All Queries Used(DDL)	14
3. Useful Query(DML)	31
VI. 역할 분담	43
1. 제안서 작성	43
2. User Requirement, ER Model	43
3. Logical Modeling.....	43
4. Implementation in Oracle DBMS	43

I. 프로젝트 제안서 :: Project Proposal

1. 서론

현재 신종 코로나바이러스 감염증의 여파로 전세계적으로 자동차 시장이 위축되어 있다. 특히 이번 코로나19로 인해 디지털 채널의 중요성이 부각되고 변화에 대해 소극적인 태도를 보였던 자동차 산업에서의 디지털 전략은 더욱 중요해지고 있다. Tesla를 비롯하여 BMW, 벤츠와 같은 주요 브랜드에서는 이미 온라인 자동차 판매 전략을 수립하고 도입했지만 국내 브랜드인 현대/기아 자동차의 경우 해외에서만 도입하고 국내에서는 시행이 되고 있지 않다.

이번 팬데믹 사태로 인해 부각된 변화의 대한 인식 다시 말해, 국내 브랜드 자동차 산업에서의 디지털 전략을 더욱 완성도 있게 하기위해 여러 정보들을 데이터화하고 새롭게 구축해 국내 자동차 산업의 디지털화를 시킬 필요가 있다.

2. 회사 설명

(1) 산업 / 비즈니스 목표

- i. 글로벌 기업의 위상에 맞는 선진 문화를 구축하며, 시대에 맞는 트렌디한 혁신 기술과 포괄적 서비스를 기반으로 개인에게 맞는 최상의 서비스를 구현하여 이를 고객에게 제공한다.
- ii. 모든 가치의 중심은 고객이며, 고객과 끊임없이 상호 소통하며 현재에 안주하지 않고 새로운 가능성에 도전하고 현대자동차만의 열정과 창의적 사고로 반드시 목표를 달성한다.
- iii. 현대자동차만이 제공하는 고객 맞춤형 서비스를 누구나 제공받을 수 있도록 적용 범위를 넓힐 것이며, 고객의 삶을 더욱 가치 있게 만들기 위해 변화와 혁신을 이끌어간다.
- iv. 차별화된 서비스와 신뢰성 있는 데이터 분석으로 1:1 고객 관리를 하며, 최상의 만족도를 이끌어낸다.

(2) 회사의 제품 / 서비스

- i. 자동차를 판매하는 회사이며, 소비자들의 성향에 맞는 세분화되고 다양한 요구를 반영하기 위해 새로운 서비스를 구축하고자 한다
- ii. 고객의 개인정보, 판매정보, 할부정보 등 다양한 정보를 하나의 DB에 통합함으로써 고객과 관련된 모든 정보를 쉽게 처리, 분석할 수 있는 서비스 시스템을 통해 개별 고객을 대상으로 소득 수준, 구매시기, 구매연령, 성별 등 타겟 마케팅을 할 수 있는 기반을 확보한다.
- iii. 데이터 분석을 통해 고객의 구매패턴과 판매예측 서비스를 도입하여 판매량을 증가시키고 고객에 대한 서비스를 최적화한다.

3. 새로운 DB가 필요한 이유

현대자동차에서는 새로운 DB를 통해 더 효율적이고, 빠르게 분석 결과를 얻고 싶다고 요청하였다. 기존의 엑셀 등을 이용하는 방식으로는 분석을 할 때나 데이터를 수정하고, 추가하는 등의 과정에서 오류가 생하는 일이 잦아 이를 해결하기 위해 DB로 시스템을 바꿀 것을 결정했다고 말하였다.

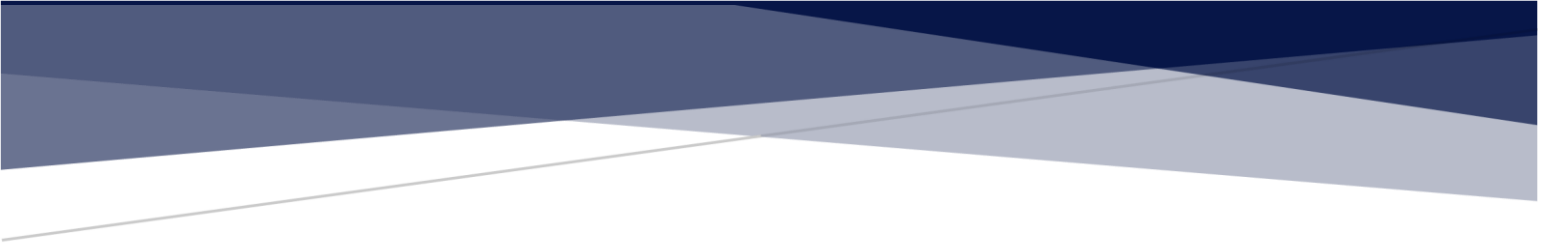
또한, 현대자동차에서 요구하는 정보들을 모두 담은 Database를 제작하기 위해 Database에 저장되어야 하는 정보들, 해당 정보들간의 관계를 기본적으로 알아야하며, 이를 고객 요구사항을 수집하는 과정에서 회사측에서 전달받아 프로젝트를 진행하였다.

4. 프로젝트를 진행하기 위해 알아야할 정보

현대자동차에서 요구하는 정보들을 모두 담은 Database를 제작하기 위해 Database에 저장되어야 하는 정보들, 해당 정보들간의 관계를 기본적으로 알아야하며, 이를 고객 요구사항을 수집하는 과정에서 회사측에서 전달받아 프로젝트를 진행하였다.

II. 고객 요구사항 :: User Requirement

- 1) 현대자동차는 자동차를 만들고 판매하는 회사이다. 현대자동차는 고객에게 최상의 서비스를 제공하기 위하여 고안하던 중, 타겟 마케팅 시스템을 도입하기로 결정하였다. 이 시스템을 구축하기 위해서는 새로운 DBMS이 필요하기 때문에, 현대자동차는 우리회사에 시스템을 구축하는 일을 의뢰하였다. 새로운 시스템을 설계하기 위한 요구사항들은 다음과 같다. 현대자동차는 특정 집단을 타겟으로 하기위한 요소들을 '효율적이고, 빠르게' 비교 분석하여 원하는 결과를 얻을 수 있는 View를 제공받기를 원한다.
- 2) 현대자동차 측에서는 데이터베이스에 현대자동차의 자동차, 고객, 직원(판매직원, 엔지니어), 거래처, 부품(사용 부품, 거래처 부품)의 정보가 포함되어야 한다고 요청하였다.
- 3) 고객에 대한 정보로는 주민등록번호(identify), 이름(성, 이름으로 구분), 성별, 이메일, 전화번호, 직업, 주소, 보험 가입 여부, 면허증 번호가 포함된다. 고객은 한 번 계약을 할 때 한 명의 판매직원을 통해 한 대의 자동차만을 구매할 수 있고, 계약을 한 적이 있는 고객들의 정보만 데이터베이스에 저장된다. 계약을 할 때에는 계약 번호, 계약 날짜, 수령 날짜, 차량 번호, 가격의 정보가 함께 저장되어야 한다고 요청했다.
- 4) 이 회사에서 직원은 판매직원과 엔지니어, 그리고 그 외의 직원들이 있다. 직원은 공통적으로 직원ID(identify), 이름(성, 이름으로 구분), 전화번호, 이메일, 주소, 직급의 정보를 포함한다. 판매직원의 경우 판매대수의 정보를 함께 포함하며, 엔지니어는 해당 엔지니어의 경력에 대한 정보를 함께 포함한다. 직원은 판매직원인면서 엔지니어일 수 없다.
- 5) 거래처는 사업자등록번호(identify), 이름, 주소, 이메일, 연락처의 정보를 포함한다. 거래처는 엔지니어를 통해 부품을 납품하며, 엔지니어와 거래를 한 적이 있는 거래처만을 이 데이터베이스에 저장한다. 엔지니어 또한 하나 이상의 거래처와 거래한 적이 있어야한다. 한 명의 엔지니어는 여러 거래처와 거래할 수 있으며, 거래처는 이 회사에서 한 명의 엔지니어와 거래한다.
- 6) 부품은 모델명(identify), 용도의 정보를 포함한다. 이 때, 부품은 반드시 사용 부품이거나 거래처부품, 혹은 둘 다이어야 하며, 사용 부품은 실제 자동차를 만들 때 사용되는 부품을 의미한다. 이 사용부품에는 수량 정보가 포함되어 있다. 거래처 부품의 경우, 반드시 거래처에서 취

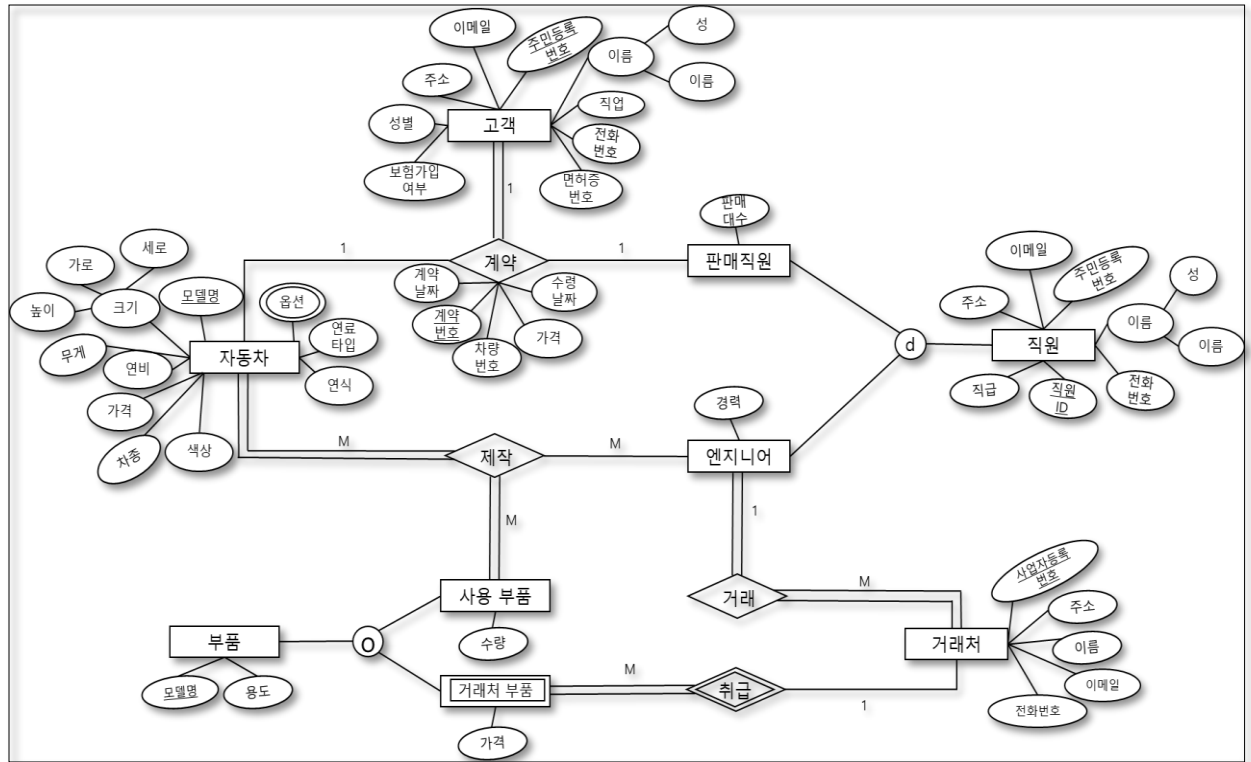


급이 되는 것이어야 하며, 서로 다른 거래처에서 같은 부품을 취급할 수 있다. 이 거래처부품에는 가격과 부품번호의 정보가 포함되어 있고, 거래처에서는 부품을 하나 이상 반드시 취급한다.

7) 자동차는 모델명(identify), 크기(가로, 세로, 높이로 구분), 무게, 연비, 가격, 차종, 색상, 연식, 연료타입, 옵션(들)의 정보를 가진다. 자동차는 여러 부품들을 합쳐서 만들어지게 되며, 하나의 부품이 여러 종류의 자동차를 만들 때에 사용되기도 한다. 또한, 자동차는 여러 명의 엔지니어에 의해 만들어지며, 엔지니어는 여러 종류의 자동차를 만들 수 있다.

III. 개념 모델링 :: conceptual modeling

1. Entity Relation Model

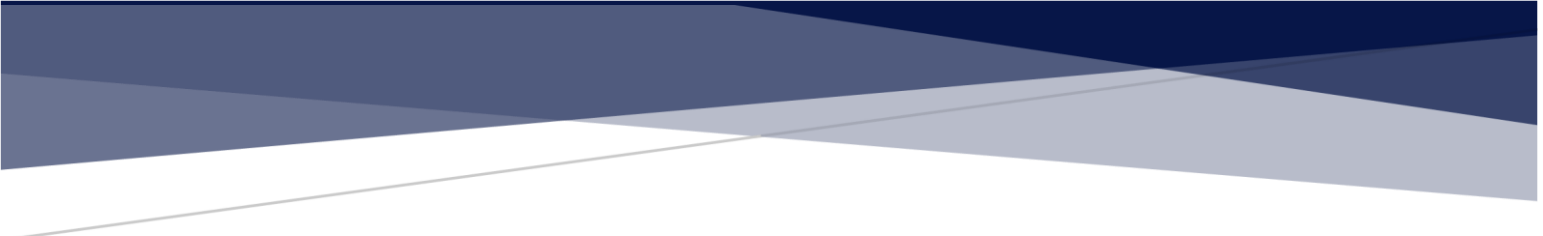


2. How to make E-R Model

ER모델은 위와 같은 형태로 완성되었으며, 어떤 정보들이 반드시 포함되어야 하는지, 그리고 서로 어떻게 연결이 되어있어야 하는지를 회사측으로부터 전달받은 요구사항에 맞추어 구상했다.

차량 구매에 대한 계약을 진행할 때, 하나의 계약에서는 하나의 자동차에 대해, 한 명의 고객, 그리고 한 명의 판매직원이 이를 진행하게 되며, 계약은 계약 번호를 통해 구분된다. 따라서, 이를 one-to-one relationship으로 연결하여 완성하였고, 계약에는 계약할 때마다 생성되는 총 5개의 attribute를 추가하여 완성했다.

두번째로 자동차 제작의 경우, 여러명의 엔지니어가 동시에 여러 개의 자동차를 제작 가능하며, 또한 여러 개의 사용 부품을 자동차 제작에 사용하고 여러 종류의 차량에 사용 가능하기 때문에 many-to-many relationship으로 연결하여 완성했다.



직원의 경우, 판매직원과 엔지니어가 존재하는데, 엔지니어와 판매직원에게 공통적으로 적용되는 데이터의 경우 직원 entity에, 그리고 각자 추가적으로 필요한 데이터는 각각의 entity로 나누어 추가하였습니다. 다만, 판매직원이면서 동시에 엔지니어인 직원은 존재할 수 없다고 제한했다.

부품은 사용부품과 거래처 부품으로 나누어지며, 이 또한 직원과 비슷하게 만들었다. 공통적인 데이터는 부품 entity에, 그리고 개별적인 데이터는 개별 entity에 저장해두었다. 다만, 사용 부품이면서 거래처 부품인 부품이 존재할 수 있다고 표시해주었다.

거래처 부품의 경우 하나의 거래처에서 여러 개 존재하며, weak entity type로 하였다. 여기서 거래처는 사업자등록번호로 구분되도록 하였다. 거래처는 한 명의 엔지니어와 거래하며, 엔지니어는 여러 개의 거래처와 거래를 하므로 one-to-many relationship의 관계를 주어 완성했다.

IV. 논리 모델링 :: Logical Modeling

1. Relational data model with Text notation

- Customer(Customer_SSN, FName, LName, Sex, Address, Email, License#, Phone#, Job, Ins_JoinStatus)
- Car(Car_ModelName, Width, Height, Depth, Weight, FuelEfficiency, Price, CarType, Color, FuelType, Modelyear)
- CarOption(Car_ModelName, CarOption)
Foreign Key(Car_ModelName) references Car(Car_ModelName)
- Part(Part_ModelName, Usage)
- UsingPart(UP_ModelName, Quantity)
Foreign Key(UP_ModelName) references Part(Part_ModelName)
- WholesalePart(WP_ModelName, Wholesale_License#, Price)
Foreign Key(WP_ModelName) references Part(Part_ModelName)
Foreign Key(Wholesale_License#) references Wholesale(License#)
- Employee(Employee_ID, Employee_SSN, FName, LName, Phone#, Address, Email, Position)
- Engineer(Engineer_ID, Career)
Foreign Key(Engineer_ID) references Employee(Employee_ID)
- SalesEmployee(SalesEmployee_ID, Sales_Volume)
Foreign Key(SalesEmployee_ID) references Employee(Employee_ID)
- Contract(Contract#, Car_ModelName, Customer_SSN, SalesEmployee_ID, Contract_Date, Car#, Price, Receiving_Date)
Foreign Key(Car_ModelName) references Car(Car_ModelName)
Foreign Key(Customer_SSN) references Customer(Customer_SSN)
Foreign Key(SalesEmployee_ID) references SalesEmployee(SalesEmployee_ID)
- Wholesale(License#, Engineer_ID, Address, Name, Email, Phone#)
Foreign Key(Engineer_ID) references Engineer(Engineer_ID)
- Production(Car_ModelName, UP_ModelName, Engineer_ID)
Foreign Key(Car_ModelName) references Car(Car_ModelName)
Foreign Key(UP_ModelName) references UsingPart(UP_ModelName)
Foreign Key(Engineer_ID) references Engineer(Engineer_ID)

2. How to make Relational data model

앞서 개념 모델링을 통해 도출된 E-R Model을 좀더 상세화하여 데이터의 집합, 관계 등을 정의한 모델을 문자로 표기했다.

이 작업은 전체 데이터 구조에서 가장 핵심을 이루기 때문에 도출된 Relational data model에서는 데이터 구조 전체가 상세하게 표현되어 모든 정보를 포함해야하고, 모든 엔티티의 정보와 엔티티들 간의 관계도 꼭 명시되어 있어야 한다.

Oracle DBMS를 효율적으로 사용하기 위해 한글로 명명되어 있던 E-R Model들을 모두 미리 번역을 해 영문으로 표기했으며, 사용되어질 테이블의 이름을 표기하고 소괄호안에 그 엔티티의 해당 속성들을 모두 표시해주었다. 또한, 엔티티의 주 식별자를 밑줄로 표기했으며, 다른 엔티티들과의 관계를 나타내기 위해 외부 식별자를 엔티티의 속성 값으로 넣어 주었다. 외부 식별자로 놓여진 속성값들이 어느 엔티티에서 참조되는 것인지도 나타내기 위해 Foreign key(외부 식별자) references 참조엔티티(참조엔티티의 주 식별자)와 같은 형태로 표기했다.

V. Implementation in Oracle DBMS

1. Table Information

Logical Modeling을 통해 만들어진 Relational data model을 기반으로 실제 데이터베이스를 Oracle DBMS를 통해 구현해 보았다.

(1) Customer

Attribute	Description	Data Type	Constraint
<u>CUSTOMER_SSN</u>	고객 주민 번호	VARCHAR2(15)	NOT NULL
LNAME	고객 성	VARCHAR2(10)	NOT NULL
FNAME	고객 이름	VARCHAR2(10)	NOT NULL
SEX	고객 성별	VARCHAR2(5)	NOT NULL
ADDRESS	고객 주소	VARCHAR2(60)	
EMAIL	고객 이메일	VARCHAR2(25)	
LICENSE#	고객 면허증 번호	VARCHAR2(15)	
PHONE#	고객 전화번호	VARCHAR2(15)	
JOB	고객 직업	VARCHAR2(20)	
INS_JOINSTATUS	고객 보험가입여부	VARCHAR2(5)	NOT NULL

(2) Car

Attribute	Description	Data Type	Constraint
<u>Car_ModelName</u>	자동차 모델명	VARCHAR2(20)	NOT NULL
Width	전폭	Number(11,2)	
Height	전장	Number(11,2)	
Depth	전고	Number(11,2)	
Weight	공차중량	Number(11,2)	
FuelEfficiency	연비	Number(11,2)	NOT NULL
Price	가격	Number(13,0)	NOT NULL
CarType	차종	VARCHAR2(10)	NOT NULL
Color	색상	VARCHAR2(10)	NOT NULL
FuelType	연료 타입	VARCHAR2(10)	NOT NULL
Modelyear	연식	VARCHAR2(4)	NOT NULL

(3) CarOption

Attribute	Description	Data Type	Constraint
<u>Car_ModelName</u>	자동차 모델명	VARCHAR2(20)	NOT NULL references Car(Car_ModelName)
CarOption	자동차 옵션	VARCHAR2(30)	NOT NULL

(4) Part

Attribute	Description	Data Type	Constraint
<u>Part_ModelName</u>	자동차 부품명	VARCHAR2(35)	NOT NULL
Usage	부품 사용 용도	VARCHAR2(30)	

(5) UsingPart

Attribute	Description	Data Type	Constraint
<u>UP_ModelName</u>	사용 부품명	VARCHAR2(35)	NOT NULL references Part(Part_ModelName)
Quantity	수량	Number(10,0)	

(6) WholesalePart

Attribute	Description	Data Type	Constraint
<u>WP_ModelName</u>	거래 부품명	VARCHAR2(35)	NOT NULL references Part(Part_ModelName)
Wholesale_License#	거래처 번호	VARCHAR2(15)	NOT NULL references WholeSale(License#)
Price	부품 가격	Number(13,0)	

(7) Employee

Attribute	Description	Data Type	Constraint
<u>EMPLOYEE_ID</u>	직원 ID 번호	VARCHAR2(20)	NOT NULL
EMPLOYEE_SSN	직원 주민번호	VARCHAR2(15)	NOT NULL
FNAME	직원 이름	VARCHAR2(10)	NOT NULL
LNAME	직원 성	VARCHAR2(10)	NOT NULL
PHONE#	직원 전화번호	VARCHAR2(15)	
ADDRESS	직원 주소	VARCHAR(60)	
EMAIL	직원 이메일	VARCHAR(25)	
POSITION	직원 직급	VARCHAR(10)	

(8) Engineer

Attribute	Description	Data Type	Constaint
<u>ENGINEER_ID</u>	엔지니어 ID 번호	VARCHAR2(20)	NOT NULL references EMPLOYEE(EMPLOYEE_ID)
CAREER	엔지니어 연차	VARCHAR2(10)	

(9) SalesEmployee

Attribute	Description	Data Type	Constaint
<u>SALESEMPLOYEE_ID</u>	판매직원 ID 번호	VARCHAR2(20)	NOT NULL references EMPLOYEE(EMPLOYEE_ID)
SALES_VOLUME	판매직원 판매량	VARCHAR(10)	

(10) Contract

Attribute	Description	Data Type	Constraint
<u>CONTRACT#</u>	계약 번호	VARCHAR2(20)	NOT NULL
CAR_MODELNAME	자동차 모델명	VARCHAR2(20)	NOT NULL references Car(Car_ModelName)
CUSTOMER_SSN	고객 주민 번호	VARCHAR2(15)	NOT NULL references Customer(Customer_SSN)
SALESEMPLOYEE_ID	판매직원 ID 번호	VARCHAR2(20)	NOT NULL references SalesEmployee(SalesEmployee_ID)
CONTRACT_DATE	계약 날짜	DATE	
CAR#	차량 번호	VARCHAR2(10)	
PRICE	판매 가격	NUMBER(13)	
RECEIVING_DATE	자동차 수령 날짜	DATE	

(11) Wholesale

Attribute	Description	Data Type	Constaint
<u>LICENSE#</u>	사업자등록번호	VARCHAR2(20)	NOT NULL
ENGINEER_ID	해당 거래처와 거래 하는 엔지니어의 ID	VARCHAR2(20)	NOT NULL references ENGINEER(ENGINEER_ID)
ADDRESS	거래처의 주소	VARCHAR2(70)	
NAME	거래처의 이름	VARCHAR2(15)	
EMAIL	거래처의 이메일	VARCHAR2(25)	
PHONE#	거래처의 전화번호	VARCHAR(15)	

(12) Production

Attribute	Description	Data Type	Constraint
<u>CAR_MODELNAME</u>	자동차 모델명	VARCHAR2(20)	NOT NULL references ENGINEER(ENGINEER_ID)
<u>UP_MODELNAME</u>	사용 부품명	VARCHAR2(35)	NOT NULL references ENGINEER(ENGINEER_ID)
<u>ENGINEER_ID</u>	엔지니어 ID 번호	VARCHAR2(20)	NOT NULL references ENGINEER(ENGINEER_ID)

2. All Queries Used(DDL)

(1) CREATE TABLE

--CREATE Customer

CREATE TABLE Customer

(Customer_SSN varchar2(15) not null,

LName varchar2(10) not null,

FName varchar2(10) not null,

Sex varchar2(5) not null,

Address varchar2(60),

Email varchar2(25),

License# varchar2(15),

Phone# varchar2(15),

Job varchar2(20),

Ins_JoinStatus varchar2(5) not null,

CONSTRAINT Customer_pk PRIMARY KEY (Customer_SSN));

--CREATE Car

CREATE TABLE Car

(Car_ModelName varchar2(20) not null,

Width number(11,2),

Height number(11,2),

Depth number(11,2),

Weight number(11,2),

FuelEfficiency number(11,2) not null,

```

Price number(13,0) not null,
CarType varchar2(10) not null,
Color varchar2(10) not null,
FuelType varchar2(10) not null,
Modelyear varchar2(4) not null,
CONSTRAINT Car_pk PRIMARY KEY (Car_ModelName));

--CREATE CarOption
CREATE TABLE CarOption
(Car_ModelName varchar2(20) not null,
CarOption varchar2(30) not null,
CONSTRAINT Option_pk PRIMARY KEY (Car_ModelName, CarOption),
CONSTRAINT Option_fk FOREIGN KEY (Car_ModelName) REFERENCES Car(Car_ModelName));

--CREATE Employee
CREATE TABLE Employee
(Employee_ID varchar2(20) not null,
Employee_SSN varchar2(15) not null,
FName varchar2(10) not null,
LName varchar2(10) not null,
Phone# varchar2(15),
Address varchar2(60),
Email varchar2(25),
Position varchar2(10),
CONSTRAINT Employee_pk PRIMARY KEY (Employee_ID));

--CREATE Engineer
CREATE TABLE Engineer
(Engineer_ID varchar2(20) not null,
Career varchar2(10),
CONSTRAINT Engineer_pk PRIMARY KEY (Engineer_ID),
CONSTRAINT Engineer_fk FOREIGN KEY (Engineer_ID) REFERENCES Employee(Employee_ID));

--CREATE SalesEmployee

```

```

CREATE TABLE SalesEmployee
(SalesEmployee_ID varchar2(20) not null,
Sales_Volume varchar2(10),
CONSTRAINT SalesEmployee_pk PRIMARY KEY (SalesEmployee_ID),
CONSTRAINT SalesEmployee_fk FOREIGN KEY (SalesEmployee_ID) REFERENCES Employee(Employee_ID));

--CREATE Wholesale
CREATE TABLE Wholesale
(License# varchar2(20) not null,
Engineer_ID varchar2(20) not null,
Address varchar2(70),
Name varchar2(15),
Email varchar2(25),
Phone# varchar2(15),
CONSTRAINT Wholesale_pk PRIMARY KEY (License#),
CONSTRAINT Wholesale_fk FOREIGN KEY (Engineer_ID) REFERENCES Engineer(Engineer_ID));

--CREATE Part
CREATE TABLE Part
(Part_ModelName varchar2(35) not null,
Usage varchar2(30),
CONSTRAINT Part_pk PRIMARY KEY (Part_ModelName));

--CREATE UsingPart
CREATE TABLE UsingPart
(UP_ModelName varchar2(35) not null,
Quantity number(10,0),
CONSTRAINT UsingPart_pk PRIMARY KEY (UP_ModelName),
CONSTRAINT UsingPart_fk FOREIGN KEY (UP_ModelName) REFERENCES Part(Part_ModelName));

--CREATE WholeSalePart
CREATE TABLE WholeSalePart
(WP_ModelName varchar2(35) not null,
Wholesale_License# varchar2(15) not null,

```



```
Price number(13,0),
CONSTRAINT WholeSalePart_pk PRIMARY KEY (WP_ModelName),
CONSTRAINT WholeSalePart_fk1 FOREIGN KEY (WP_ModelName) REFERENCES Part(Part_ModelName),
CONSTRAINT WholeSalePart_fk2 FOREIGN KEY (Wholesale_License#) REFERENCES Wholesale(License#));
```

--CREATE Contract

```
CREATE TABLE Contract
(Contract# varchar2(20) not null,
Car_ModelName varchar2(20) not null,
Customer_SSN varchar2(15) not null,
SalesEmployee_ID varchar2(20) not null,
Contract_Date date,
Car# varchar2(10),
Price number(13,0),
Receiving_Date date,
CONSTRAINT Contract_pk PRIMARY KEY (Contract#),
CONSTRAINT Contract_fk1 FOREIGN KEY (Car_ModelName) REFERENCES Car(Car_ModelName),
CONSTRAINT Contract_fk2 FOREIGN KEY (Customer_SSN) REFERENCES Customer(Customer_SSN),
CONSTRAINT Contract_fk3 FOREIGN KEY (SalesEmployee_ID) REFERENCES SalesEmployee(SalesEmployee_ID));
```

--CREATE Production

```
CREATE TABLE Production
(Car_ModelName varchar2(20) not null,
UP_ModelName varchar2(35) not null,
Engineer_ID varchar2(20) not null,
CONSTRAINT Production_pk PRIMARY KEY (Car_ModelName, UP_ModelName, Engineer_ID),
CONSTRAINT Production_fk1 FOREIGN KEY (Car_ModelName) REFERENCES Car(Car_ModelName),
CONSTRAINT Production_fk2 FOREIGN KEY (UP_ModelName) REFERENCES UsingPart(UP_ModelName),
CONSTRAINT Production_fk3 FOREIGN KEY (Engineer_ID) REFERENCES Engineer(Engineer_ID));
```

(2) INSERT INTO VALUES

-- INSERT Employee

```
INSERT INTO employee VALUES ('E001', '9703211297428', '광현', '정', '01087235343', '
서울시 마포구 고산2길 11', 'wjdrhkdus@gmail.com', '매니저');
```

INSERT INTO employee VALUES ('E002', '9610232391827', '나은', '이', '01082756264', '경기도 광주시 광주대로 224번길', 'dlskdms@gmail.com', '사원');

INSERT INTO employee VALUES ('E003', '9305292194821', '다현', '최', '01012948375', '충청남도 예산군 덕산면 광천중앙길 7', 'chlekgus@gmail.com', '사원');

INSERT INTO employee VALUES ('E004', '9406112938472', '하은', '최', '01037915736', '경기도 하남시 서하남로 390번길', 'chlgkdms@gmail.com', '사원');

INSERT INTO employee VALUES ('E005', '9810281103927', '보훈', '채', '01093185627', '인천광역시 부평구 평천로 402(삼산동)', 'coqhngns@gmail.com', '매니저');

INSERT INTO employee VALUES ('E006', '9603291928472', '동혁', '이', '01039284726', '경기도 용인시 기흥구 예현로 41(서천동)', 'dlehdgur@gmail.com', '팀원');

INSERT INTO employee VALUES ('E007', '9208052093847', '하나', '기', '01029582712', '서울특별시 강남구 논현로 28(개포동)', 'rlgksk@gmail.com', '팀원');

INSERT INTO employee VALUES ('E008', '8960211284756', '한별', '이', '01072847162', '경기도 광주시 중앙로 335번길 28', 'dlgksquf@gmail.com', '팀원');

INSERT INTO employee VALUES ('E009', '9912111928475', '현식', '김', '01082762262', '전라북도 전주시 완산구 삼산길 74-5', 'rlagustlr@gmail.com', '팀원');

INSERT INTO employee VALUES ('E010', '9709281139485', '하진', '김', '01092847261', '경상남도 밀양시 명예실로2길 45-3', 'rlagkwls@gmail.com', '팀장');

INSERT INTO employee VALUES ('E011', '9807292029485', '승희', '손', '01062756271', '경기도 하남시 신우실로 15', 'thstmdgml@gmail.com', '사원');

INSERT INTO employee VALUES ('E012', '9801272039857', '한이', '배', '01082726428', '대전광역시 서구 신갈마로 2', 'qogksdl@gmail.com', '사원');

INSERT INTO employee VALUES ('E013', '9702321029372', '민석', '김', '01084731672', '경상북도 울진군 매화면 갈면갈마길 9-4', 'rlaalstjr@gmail.com', '사원');

INSERT INTO employee VALUES ('E014', '9405122192827', '채연', '한', '01024263416', '경기도 파주시 문산읍 문향로 75번길 52', 'gkscodus@gmail.com', '사원');

INSERT INTO employee VALUES ('E015', '9602131928374', '지완', '양', '01019284625', '전라북도 군산시 요죽안2길 57-25', 'didwldhks@gmail.com', '매니저');

INSERT INTO employee VALUES ('E016', '9407121204821', '현재', '이', '01012376512', '서울특별시 강남구 도산대로 101길 34', 'dlguswo@gmail.com', '팀원');

INSERT INTO employee VALUES ('E017', '9602102415783', '가연', '김', '01082367552', '전라북도 장수군 계남면 계남중길 52', 'rlarkdus@gmail.com', '팀원');

INSERT INTO employee VALUES ('E018', '9712021948264', '하윤', '최', '01026847253', '전라북도 군산시 임피면 계남1길 35', 'chlgkdus@gmail.com', '팀원');

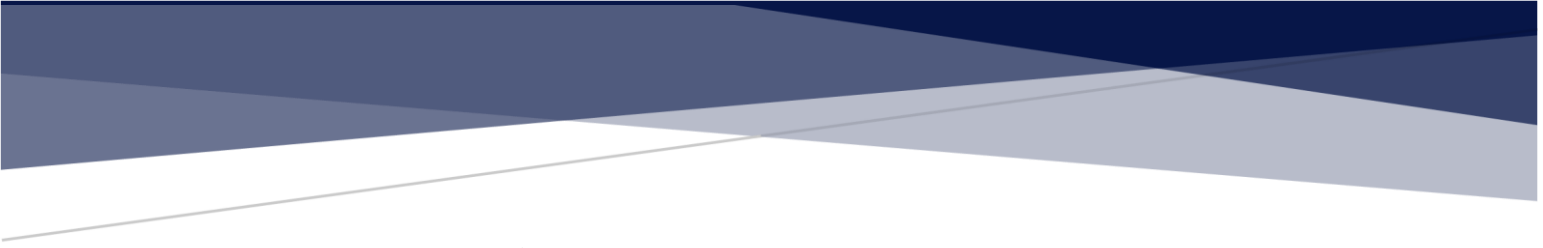
INSERT INTO employee VALUES ('E019', '9801212294857', '예진', '한', '01059284521', '경기도 하남시 신우실로 15', 'thstmdgml@gmail.com', '사원');

```

서울특별시 종로구 대학로12길 61, 'gksdPwls@gmail.com', '팀원');
INSERT INTO employee VALUES ('E020', '9612211938562', '지원', '김', '01017253742', '
경기도 시흥시 계수로 19, 'rlawldnjs@gmail.com', '팀원');
INSERT INTO employee VALUES ('E021', '9710281927462', '정훈', '노', '01017375261', '
강원도 삼척시 도계읍 도계로 316, 'shwjdgns@gmail.com', '사원');
INSERT INTO employee VALUES ('E022', '8901171859821', '진호', '홍', '01072514221', '
강원도 정선군 임계면 송계3길 3, 'ghdwlsgh@gmail.com', '사원');
INSERT INTO employee VALUES ('E023', '9001262562514', '예희', '박', '01062846445', '
충청남도 천안시 동남구 목천읍 충절로 813, 'qkrdPgml@gmail.com', '사원');
INSERT INTO employee VALUES ('E024', '9702212837461', '나연', '한', '01072552841', '
전라북도 순창군 동계면 동계로 11, 'gksskdus@gmail.com', '사원');
INSERT INTO employee VALUES ('E025', '9610231029846', '원상', '최', '01028246182', '
전라북도 임실군 삼계면 충효로 1293, 'chldnjstkd@gmail.com', '사원');
INSERT INTO employee VALUES ('E026', '9802122912746', '다빈', '최', '01082625191', '
경기도 광주시 도척면 상림마을길 10, 'chlekqls@gmail.com', '팀원');
INSERT INTO employee VALUES ('E027', '9305231928372', '상엽', '김', '01068726482', '
전라북도 부안군 보안면 상림안길 6, 'rlatkdduq@gmail.com', '팀원');
INSERT INTO employee VALUES ('E028', '9702152082715', '수현', '기', '01072654112', '
경상북도 경산시 진량읍 상림2길 17, 'rltngus@gmail.com', '팀원');
INSERT INTO employee VALUES ('E029', '9401281923781', '진우', '한', '01027234578', '
경상북도 구미시 장천면 장천상림 1길 10, 'gkswlsdn@gmail.com', '팀원');
INSERT INTO employee VALUES ('E030', '9602122298191', '아연', '김', '01032947215', '
서울특별시 강서구 공항대로 58가길 72-6, 'rladkdus@gmail.com', '팀원');
INSERT INTO employee VALUES ('E031', '9702131981234', '석우', '정', '010847917123', '
서울특별시 관악구 봉천로 13나길 63-7, 'wjdtjrdn@gmail.com', '사원');
INSERT INTO employee VALUES ('E032', '9102311328345', '현우', '김', '010962534212', '
서울특별시 강남구 도곡로77길 29, 'rlagusdn@gmail.com', '사원');
INSERT INTO employee VALUES ('E033', '9610262948926', '예은', '김', '010271386791', '
대전광역시 서구 관저중로 33, 'rladPdms@gmail.com', '팀원');
INSERT INTO employee VALUES ('E034', '9711222230183', '은지', '박', '010726356165', '
서울특별시 노원구 한글비석로 1길 15, 'qkrdmswl@gmail.com', '팀원');
INSERT INTO employee VALUES ('E035', '9810252912482', '나현', '임', '010826367132', '
인천광역시 미추홀구 독배로 305, 'dlaskgus@gmail.com', '팀원');

```

-- INSERT Engineer



```
INSERT INTO engineer VALUES('E006', '3');
INSERT INTO engineer VALUES('E007', '2');
INSERT INTO engineer VALUES('E008', '6');
INSERT INTO engineer VALUES('E009', '2');
INSERT INTO engineer VALUES('E010', '4');
INSERT INTO engineer VALUES('E016', '2');
INSERT INTO engineer VALUES('E017', '1');
INSERT INTO engineer VALUES('E018', '5');
INSERT INTO engineer VALUES('E019', '7');
INSERT INTO engineer VALUES('E020', '1');
INSERT INTO engineer VALUES('E026', '2');
INSERT INTO engineer VALUES('E027', '4');
INSERT INTO engineer VALUES('E028', '2');
INSERT INTO engineer VALUES('E029', '3');
INSERT INTO engineer VALUES('E030', '2');
INSERT INTO engineer VALUES('E033', '5');
INSERT INTO engineer VALUES('E034', '1');
INSERT INTO engineer VALUES('E035', '3');
```

```
--INSERT SlaesEmployee
```

```
INSERT INTO salesemployee VALUES('E001', '3');
INSERT INTO salesemployee VALUES('E002', '5');
INSERT INTO salesemployee VALUES('E003', '3');
INSERT INTO salesemployee VALUES('E004', '3');
INSERT INTO salesemployee VALUES('E005', '1');
INSERT INTO salesemployee VALUES('E011', '1');
INSERT INTO salesemployee VALUES('E012', '1');
INSERT INTO salesemployee VALUES('E013', '1');
INSERT INTO salesemployee VALUES('E014', '3');
INSERT INTO salesemployee VALUES('E015', '2');
INSERT INTO salesemployee VALUES('E021', '2');
INSERT INTO salesemployee VALUES('E022', '1');
INSERT INTO salesemployee VALUES('E023', '2');
INSERT INTO salesemployee VALUES('E024', '1');
INSERT INTO salesemployee VALUES('E025', '1');
```

INSERT INTO salesemployee VALUES('E031', '1');

INSERT INTO salesemployee VALUES('E032', '3');

--INSERT wholesale

INSERT INTO wholesale VALUES('W001', 'E006', '전라북도 전주시 완산구 우림로 610-2',
'우림부품', 'dhckdParts@gmail.com', '01018361348');

INSERT INTO wholesale VALUES('W002', 'E007', '경상북도 배리1길 1', '배리부품',
'qoflparts@gmail.com', '01061365137');

INSERT INTO wholesale VALUES('W003', 'E007', '강원도 원주시 치악로 1881', '원주부품',
'dnjswnparts@gmail.com', '01031263523');

INSERT INTO wholesale VALUES('W004', 'E008', '충청북도 청주시 청원구 오창읍 중부
로 1359-83', '오창부품', 'dhckdparts', '01053612432');

INSERT INTO wholesale VALUES('W005', 'E009', '충청남도 금산군 남일면 금산로 267', '
금산부품', 'rmatksparts@gmail.com', '01029438134');

INSERT INTO wholesale VALUES('W006', 'E010', '충청북도 천안시 서북구 성환읍 울금4
길 40-16', '성환부품', 'tjdghksparts@gmail.com', '01048316782');

INSERT INTO wholesale VALUES('W007', 'E016', '전라북도 전주시 완산구 우림로 1090',
'완산부품', 'dhksks@gmail.com', '01053713652');

INSERT INTO wholesale VALUES('W008', 'E016', '강원도 화천군 성서면 영서로 7737', '
영서부품', 'dudtjparts@gmail.com', '010148362326');

INSERT INTO wholesale VALUES('W009', 'E017', '서울특별시 강동구 양재대로 89가길
66-1', '양재부품', 'didwoparts@gmail.com', '01046213946');

INSERT INTO wholesale VALUES('W010', 'E018', '서울특별시 강북구 도봉로 95길 21', '
도봉부품', 'ehqhdparts@gmail.com', '01081763498');

INSERT INTO wholesale VALUES('W011', 'E019', '서울특별시 관악구 봉천로23바길 13-
17', '봉천부품', 'qhdcjparts@gmail.com', '01013481274');

INSERT INTO wholesale VALUES('W012', 'E019', '전라남도 나주시 산포면 산포로 461', '
나주부품', 'skwnparts@gmail.com', '01038497237');

INSERT INTO wholesale VALUES('W013', 'E020', '서울특별시 관악구 행운길 32-14', '행
운부품', 'goddnsparts@gmail.com', '01083471983');

INSERT INTO wholesale VALUES('W014', 'E026', '서울특별시 구로구 경인로 113', '경인
부품', 'ruddlsparts@gmail.com', '01019436174');

INSERT INTO wholesale VALUES('W015', 'E027', '경기도 가평군 설악면 가마소길 34-50',
'설악부품', 'tjfdkrparts@gmail.com', '01015313432');

INSERT INTO wholesale VALUES('W016', 'E028', '충청북도 단양군 영춘면 김삿갓로 137-

```

69', '김삿갓부품', 'rlatkrktparts@gmail.com', '01082438179');
INSERT INTO wholesale VALUES('W017', 'E029', '경상북도 경주시 산내면 대현길 128-5',
'대현부품', 'eogusparts@gmail.com', '01041893225');
INSERT INTO wholesale VALUES('W018', 'E030', '경상남도 산청군 산청읍 웅석봉로 154
번길 10-10', '웅석봉부품', 'dndtjrghdparts@gmail.com', '010764835235');
INSERT INTO wholesale VALUES('W019', 'E033', '경상남도 양산시 영동면 풍호대길 51',
'양산부품', 'didtksparts@gmail.com', '01071237195');
INSERT INTO wholesale VALUES('W020', 'E034', '경기도 부천시 안곡로30번길 29', '부천
부품', 'qncjsparts@gmail.com', '01049873213');
INSERT INTO wholesale VALUES('W021', 'E034', '충청남도 당진시 고대면 대호만로', '고
대부품', 'rheoparts@gmail.com', '01078414332');
INSERT INTO wholesale VALUES('W022', 'E035', '부산광역시 남구 천제등로 28번길 29',
'천제부품', 'cjswpparts@gmail.com', '01098172346');
INSERT INTO wholesale VALUES('W023', 'E035', '부산광역시 동래구 중앙대로 1381번길
71', '중앙부품', 'wnddkdparts@gmail.com', '01071329847');

```

-- INSERT Car

```

INSERT INTO Car VALUES('GENESIS_G90', 1880, 4670, 1430, 1390, 14, 94550000, 'Sedan',
'Black', 'Gasoline', 2020);
INSERT INTO Car VALUES('GENESIS_G80', 1870, 4600, 1450, 1300, 13, 52910000, 'Sedan',
'White', 'Gasoline', 2020);
INSERT INTO Car VALUES('GENESIS_G70', 1875, 4640, 1450, 1300, 12, 48300000, 'Sedan',
'Black', 'Gasoline', 2020);
INSERT INTO Car VALUES('GENESIS_GV80', 1975, 4945, 1715, 2105, 11, 64670000, 'SUV',
'Blue', 'Diesel', 2020);
INSERT INTO Car VALUES('GENESIS_GV70', 1970, 4940, 1720, 2005, 12, 64670000, 'SUV',
'Black', 'Gasoline', 2020);
INSERT INTO Car VALUES('PALISADE', 1975, 4980, 1750, 1880, 9, 45300000, 'SUV', 'Black',
'Gasoline', 2019);
INSERT INTO Car VALUES('PALISADE_AWD', 1975, 4980, 1750, 1880, 9, 50030000, 'SUV',
'Black', 'Diesel', 2019);
INSERT INTO Car VALUES('GRANDEUR', 1865, 4930, 1470, 1640, 10, 41080000, 'Sedan',
'Black', 'Gasoline', 2018);
INSERT INTO Car VALUES('GRANDEUR_HYBRID', 1875, 4990, 1470, 1700, 16, 43410000,
'Sedan', 'Black', 'Electric', 2020);

```



```

INSERT INTO Car VALUES('AVANTE', 1830, 4570, 1440, 1220, 8, 22200000, 'Sedan',
'White', 'Gasoline', 2020);
INSERT INTO Car VALUES('AVANTE_HYBRID', 1830, 4570, 1440, 1220, 8, 22200000,
'Sedan', 'White', 'Electric', 2020);
INSERT INTO Car VALUES('SONATA', 1865, 4855, 1475, 1465, 11, 23860000, 'Sedan',
'White', 'Diesel', 2019);
INSERT INTO Car VALUES('SANTAFE', 1880, 4700, 1680, 1790, 12, 29750000, 'SUV',
'White', 'Gasoline', 2018);
INSERT INTO Car VALUES('TUCSON', 1885, 4475, 1650, 1730, 11, 24400000, 'SUV', 'Black',
'Diesel', 2020);
INSERT INTO Car VALUES('GrandStarex', 1920, 5150, 1935, 2060, 11, 22090000, 'VAN',
'Black', 'Diesel', 2020);
INSERT INTO Car VALUES('KONA',1800,4205,1550,1350,12,20310000,'SUV','RED','Diesel',
2020);
INSERT INTO Car VALUES('POTER2', 1740, 5175, 1970, 1825, 9, 16700000, 'Truck', 'Blue',
'Diesel', 2020);
INSERT INTO Car VALUES('Venue', 1770, 4040, 1565, 1215, 13, 18720000, 'SUV', 'Blue',
'Gasoline', 2020);

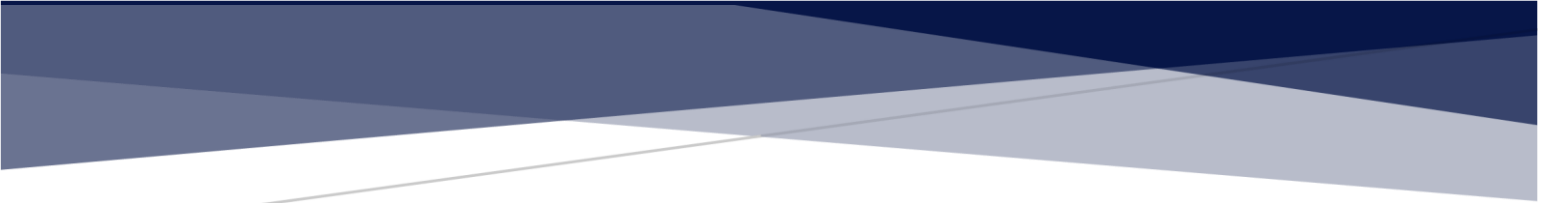
```

```
-- INSERT CarOption
```

```

INSERT INTO CarOption VALUES('GENESIS_G90', 'EmergencyBrakingSystem');
INSERT INTO CarOption VALUES('GENESIS_G80', 'AutomaticParking');
INSERT INTO CarOption VALUES('GENESIS_G70', 'ECM_SideMirror');
INSERT INTO CarOption VALUES('GENESIS_GV80', 'Hi_Pass');
INSERT INTO CarOption VALUES('GENESIS_GV70', 'TiltSteering');
INSERT INTO CarOption VALUES('PALISADE', 'SurroundMonitor');
INSERT INTO CarOption VALUES('GRANDEUR','VoiceRecognition');
INSERT INTO CarOption VALUES('AVANTE', 'WirelessDoorLock');
INSERT INTO CarOption VALUES('SONATA', 'Navigation');
INSERT INTO CarOption VALUES('SANTAFE', 'USB_AddInput');
INSERT INTO CarOption VALUES('TUCSON','SmartKey');
INSERT INTO CarOption VALUES('GrandStarex','FoldingRearSeat');
INSERT INTO CarOption VALUES('KONA','CoolingSeat');
INSERT INTO CarOption VALUES('POTER2', 'RearCamera');
INSERT INTO CarOption VALUES('Venue', 'Airbag');

```



```
-- INSERT Part
INSERT INTO Part VALUES('AirConditionerFilter', 'Filter');
INSERT INTO Part VALUES('FuelFilter', 'Filter');
INSERT INTO Part VALUES('OilFilter', 'Filter');
INSERT INTO Part VALUES('CylinderHead', 'Engine');
INSERT INTO Part VALUES('TensionerPully', 'Engine');
INSERT INTO Part VALUES('BeltDrive', 'Engine');
INSERT INTO Part VALUES('Radiator', 'Cooling');
INSERT INTO Part VALUES('Condenser', 'Cooling');
INSERT INTO Part VALUES('HeatExchanger', 'Heating');
INSERT INTO Part VALUES('Stabiliser', 'Wheel');
INSERT INTO Part VALUES('CoilSpring', 'Suspension');
INSERT INTO Part VALUES('PulseGenerator', 'Sensor');
INSERT INTO Part VALUES('BrakeDisc', 'BrakeSystem');
INSERT INTO Part VALUES('HandBrake', 'BrakeSystem');
INSERT INTO Part VALUES('StarterBattery', 'BrakeSystem');
INSERT INTO Part VALUES('WheelSensor', 'BrakeSystem');
INSERT INTO Part VALUES('SpotLight', 'BrakeSystem');
INSERT INTO Part VALUES('HeadLight', 'BrakeSystem');
INSERT INTO Part VALUES('CrankshaftPulse', 'Sensor');
INSERT INTO Part VALUES('IgnitionCable', 'Spark');
INSERT INTO Part VALUES('WiperBlade', 'Wiper');
INSERT INTO Part VALUES('SideWall', 'Interior');
INSERT INTO Part VALUES('PlateLight', 'Interior');
INSERT INTO Part VALUES('VentilationGrille', 'Interior');
INSERT INTO Part VALUES('JackingPoint', 'Interior');
INSERT INTO Part VALUES('HydraulicFilter', 'Filter');
INSERT INTO Part VALUES('Gasket', 'Engine');
INSERT INTO Part VALUES('OilDrainPlug', 'Engine');
INSERT INTO Part VALUES('LambdaSensor', 'Sensor');
INSERT INTO Part VALUES('TuningModule', 'Engine');
INSERT INTO Part VALUES('PariculateFilter', 'Filter');
INSERT INTO Part VALUES('Manifold', 'Engine');
INSERT INTO Part VALUES('MiddleSilencer', 'Engine');
INSERT INTO Part VALUES('CatalyticConverter', 'AirConditioning');
```



```
INSERT INTO Part VALUES('Evaporator', 'AirConditioning');
INSERT INTO Part VALUES('HighPressureLine', 'AirConditioning');
INSERT INTO Part VALUES('ExpansionValve', 'AirConditioning');
INSERT INTO Part VALUES('RubberBuffer', 'Suspension');
INSERT INTO Part VALUES('DustCover', 'Suspension');
INSERT INTO Part VALUES('GasSpring', 'Interior');
INSERT INTO Part VALUES('MirrorGlass', 'Interior');
INSERT INTO Part VALUES('InnerWingPanel', 'Interior');
INSERT INTO Part VALUES('FootwellTray', 'Interior');
INSERT INTO Part VALUES('PedalLining', 'Interior');
INSERT INTO Part VALUES('Harness', 'ElectricDevice');
```

```
-- INSERT UsingPart
```

```
INSERT INTO UsingPart VALUES('AirConditionerFilter', 550);
INSERT INTO UsingPart VALUES('FuelFilter', 400);
INSERT INTO UsingPart VALUES('OilFilter', 600);
INSERT INTO UsingPart VALUES('CylinderHead', 250);
INSERT INTO UsingPart VALUES('TensionerPully', 230);
INSERT INTO UsingPart VALUES('BeltDrive', 450);
INSERT INTO UsingPart VALUES('Radiator', 300);
INSERT INTO UsingPart VALUES('Condenser', 530);
INSERT INTO UsingPart VALUES('HeatExchanger', 70);
INSERT INTO UsingPart VALUES('Stabiliser', 100);
INSERT INTO UsingPart VALUES('CoilSpring', 600);
INSERT INTO UsingPart VALUES('PulseGenerator', 430);
INSERT INTO UsingPart VALUES('BrakeDisc', 190);
INSERT INTO UsingPart VALUES('HandBrake', 290);
INSERT INTO UsingPart VALUES('FootwellTray', 600);
INSERT INTO UsingPart VALUES('PedalLining', 100);
INSERT INTO UsingPart VALUES('Harness', 130);
INSERT INTO UsingPart VALUES('StarterBattery', 4000);
INSERT INTO UsingPart VALUES('WheelSensor', 195);
INSERT INTO UsingPart VALUES('SpotLight', 750);
INSERT INTO UsingPart VALUES('HeadLight', 242);
INSERT INTO UsingPart VALUES('CrankshaftPulse', 480);
```

```
INSERT INTO UsingPart VALUES('IgnitionCable',110);
INSERT INTO UsingPart VALUES('WiperBlade',300);
INSERT INTO UsingPart VALUES('SideWall', 234);
INSERT INTO UsingPart VALUES('PlateLight', 50);
INSERT INTO UsingPart VALUES('VentilationGrille',208);
INSERT INTO UsingPart VALUES('JackingPoint', 112);
INSERT INTO UsingPart VALUES('HydraulicFilter',250);
INSERT INTO UsingPart VALUES('Gasket', 320);
INSERT INTO UsingPart VALUES('OilDrainPlug', 231);
INSERT INTO UsingPart VALUES('LambdaSensor', 100);
```

```
-- INSERT WholeSalePart
```

```
INSERT INTO WholeSalePart VALUES('StarterBattery', 'W001', 40000);
INSERT INTO WholeSalePart VALUES('WheelSensor', 'W002', 89500);
INSERT INTO WholeSalePart VALUES('SpotLight', 'W003', 7500);
INSERT INTO WholeSalePart VALUES('HeadLight', 'W004', 9400);
INSERT INTO WholeSalePart VALUES('CrankshaftPulse', 'W005', 4800);
INSERT INTO WholeSalePart VALUES('IgnitionCable','W006', 12300);
INSERT INTO WholeSalePart VALUES('WiperBlade','W007', 8300);
INSERT INTO WholeSalePart VALUES('SideWall', 'W008', 4300);
INSERT INTO WholeSalePart VALUES('PlateLight', 'W009', 500);
INSERT INTO WholeSalePart VALUES('VentilationGrille','W010', 30800);
INSERT INTO WholeSalePart VALUES('JackingPoint', 'W011', 11400);
INSERT INTO WholeSalePart VALUES('HydraulicFilter', 'W012', 25000);
INSERT INTO WholeSalePart VALUES('Gasket', 'W013', 3200);
INSERT INTO WholeSalePart VALUES('OilDrainPlug', 'W014', 2300);
INSERT INTO WholeSalePart VALUES('LambdaSensor', 'W015', 228100);
INSERT INTO WholeSalePart VALUES('TuningModule','W016', 13200);
INSERT INTO WholeSalePart VALUES('PariculateFilter','W017', 18000);
INSERT INTO WholeSalePart VALUES('Manifold','W018', 1300);
INSERT INTO WholeSalePart VALUES('MiddleSilencer', 'W018', 215700);
INSERT INTO WholeSalePart VALUES('CatalyticConverter', 'W018', 632300);
INSERT INTO WholeSalePart VALUES('Evaporator','W018', 161500);
INSERT INTO WholeSalePart VALUES('HighPressureLine', 'W018',34000);
INSERT INTO WholeSalePart VALUES('ExpansionValve', 'W018', 153600);
```

```

INSERT INTO WholeSalePart VALUES('RubberBuffer','W018', 1700);
INSERT INTO WholeSalePart VALUES('DustCover', 'W018', 59000);
INSERT INTO WholeSalePart VALUES('GasSpring', 'W021', 15100);
INSERT INTO WholeSalePart VALUES('MirrorGlass','W021', 13400);
INSERT INTO WholeSalePart VALUES('InnerWingPanel', 'W021', 16000);
INSERT INTO WholeSalePart VALUES('Harness', 'W004', 6300);
INSERT INTO WholeSalePart VALUES('AirConditionerFilter', 'W015',11550);
INSERT INTO WholeSalePart VALUES('FuelFilter','W020', 23000);
INSERT INTO WholeSalePart VALUES('OilFilter', 'W017',33500);
INSERT INTO WholeSalePart VALUES('CylinderHead','W006',45000);
INSERT INTO WholeSalePart VALUES('TensionerPully', 'W011',202900);
INSERT INTO WholeSalePart VALUES('BeltDrive', 'W014',180200);
INSERT INTO WholeSalePart VALUES('Radiator', 'W013',46000);
INSERT INTO WholeSalePart VALUES('Condenser', 'W011',34000);
INSERT INTO WholeSalePart VALUES('HeatExchanger','W017', 56000);
INSERT INTO WholeSalePart VALUES('Stabiliser', 'W019',6800);
INSERT INTO WholeSalePart VALUES('CoilSpring','W001', 282000);
INSERT INTO WholeSalePart VALUES('PulseGenerator','W011', 220000);
INSERT INTO WholeSalePart VALUES('BrakeDisc', 'W020',19500);
INSERT INTO WholeSalePart VALUES('HandBrake','W012', 20500);
INSERT INTO WholeSalePart VALUES('FootwellTray','W003', 11600);
INSERT INTO WholeSalePart VALUES('PedalLining', 'W017',10200);

```

-- INSERT Customer

```

INSERT INTO Customer VALUES('8723409182309','김','태희','F','부산광역시 해운대구 동
백로 67','sdf3783@naver.com','378239867895312','01029837463','배우','Y');
INSERT INTO Customer VALUES('8220394826554','이','영훈','M','서울특별시 용산구 용산
동 6가','younghoon12@gmail.com','378291049576820','01093857261','사업가','N');
INSERT INTO Customer VALUES('8655122983280','최','재민','M','서울특별시 종로구 자하
문로 4길 21','cjmin43@hanmail.com','300928166738421','01099836271','변호사','Y');
INSERT INTO Customer VALUES('9883029374933','박','지훈','M','대전광역시 동구 산내로
321-35','jihoon529@naver.com','345666812903481','01090482764','개발자','N');
INSERT INTO Customer VALUES('9019222583745','최','아린','F','강원도 속초시 청호로
122','dkflsdl00@daum.net','377896251738495','01056672391','회사원','N');
INSERT INTO Customer VALUES('8799037741978','김','수진','F','제주도 서귀포시 안덕면

```

중산간서로 1610','cnjdopsf@gmail.com','399820193654875','01043217564','제조업자','N');

INSERT INTO Customer VALUES('8459776110322','이','한이','M','경상북도 안동시 풍천면 전서로 186','onetwo22@hanmail.com','329074504387462','01084561126','의사','Y');

INSERT INTO Customer VALUES('9760259686681','배','수지','F','충청북도 단양군 매포읍 삼봉로 644','suzysuzy@naver.com','339647947251749','01075569322','공무원','Y');

INSERT INTO Customer VALUES('9641635790339','차','동혁','M','경상북도 경주시 보불로 216-8','deligtcha4242@daum.net','365570098123486','01035641889','물리치료사','Y');

INSERT INTO Customer VALUES('8502847249814','장','희영','F','전라북도 전주시 완산구 기린대로 99','wkgmldud@gmail.com','311005672983614','01012278645','교수','Y');

INSERT INTO Customer VALUES('8989762351666','정','준우','M','경기도 파주시 탄현면 헤이리마을길 70-21','joonwu@hanmail.com','353564996212765','01089903547','공장 근로자','N');

INSERT INTO Customer VALUES('8360191663389','김','새봄','F','경기도 양평군 양서면 양수리 32','newspringkim@naver.com','398317048645743','01079880354','피디','Y');

INSERT INTO Customer VALUES('8908903423055','양','지연','F','충청북도 청주시 상당구 문의면 청남대길 646','delayyang@naver.com','369276581203486','01019226837','간호사','N');

INSERT INTO Customer VALUES('9354827699576','박','재환','M','전라북도 군산시 해망로 240','nevermdow@gmail.com','358799012112545','01074658902','연구원','Y');

INSERT INTO Customer VALUES('9484623098581','정','민현','M','서울특별시 중구 덕수궁길 61','oiwedsfjk@hanmail.com','335566669090712','01064587791','판사','Y');

INSERT INTO Customer VALUES('8472309669263','김','동현','M','서울특별시 성동구 서울숲2길 32-14','nondong@naver.com','387265948716253','01027394665','회계사','Y');

-- INSERT contract

INSERT INTO contract VALUES('202012205019', 'GENESIS_G90', '9019222583745', 'E001','2020-12-20','429837','95000000','2020-12-31');

INSERT INTO contract VALUES('202011148392', 'AVANTE', '8723409182309', 'E002','2020-11-14','642142','25000000','2020-12-31');

INSERT INTO contract VALUES('202010238473', 'SONATA', '8220394826554', 'E003','2020-10-23','864633','23900000','2020-11-15');

INSERT INTO contract VALUES('202010102993', 'AVANTE_HYBRID', '9883029374933', 'E003','2020-10-10','085356','23000000','2020-11-17');

INSERT INTO contract VALUES('202012128372', 'SANTAFE', '8908903423055',

```

'E004','2020-12-12','832452','30000000','2021-02-14');
INSERT INTO contract VALUES('202011308931', 'GrandStarex', '9760259686681',
'E005','2020-11-30','547783','23000000','2021-01-23');
INSERT INTO contract VALUES('202010034331', 'GRANDEUR', '8655122983280',
'E011','2020-10-03','980342','41100000','2020-12-25');
INSERT INTO contract VALUES('202009218021', 'TUCSON', '8360191663389',
'E012','2020-09-21','225472','25000000','2020-11-28');
INSERT INTO contract VALUES('202009247361', 'GENESIS_G80', '8655122983280',
'E014','2020-09-24','386313','54000000','2020-11-01');
INSERT INTO contract VALUES('202003143329', 'KONA', '8799037741978', 'E014','2020-
03-14','586797','20400000','2020-06-10');
INSERT INTO contract VALUES('202006043910', 'POTER2', '9760259686681',
'E015','2020-06-04','023892','16700000','2020-09-24');
INSERT INTO contract VALUES('202007312811', 'GENESIS_GV70', '9641635790339',
'E021','2020-07-31','776212','48300000','2020-10-10');
INSERT INTO contract VALUES('202008192299', 'GRANDEUR_HYBRID', '8360191663389',
'E022','2020-08-19','823452','43410000','2020-11-17');
INSERT INTO contract VALUES('202004112911', 'GENESIS_G70', '8908903423055',
'E023','2020-04-11','612232','48300000','2020-07-30');
INSERT INTO contract VALUES('201908221227', 'Venue', '9354827699576', 'E024','2019-
08-22','909452','19720000','2019-12-03');
INSERT INTO contract VALUES('201912073928', 'GrandStarex', '9484623098581',
'E031','2019-12-07','123456','23000000','2020-03-04');
INSERT INTO contract VALUES('201903152019', 'PALISADE', '8472309669263',
'E032','2019-03-15','661124','46000000','2019-05-07');
INSERT INTO contract VALUES('202011114392', 'AVANTE', '8723409182309',
'E002','2020-11-11','642142','25000000','2020-12-31');
INSERT INTO contract VALUES('202009233473', 'SONATA', '8220394826554',
'E003','2020-09-23','864633','23900000','2020-11-15');
INSERT INTO contract VALUES('202008128372', 'SANTAFE', '8908903423055',
'E004','2020-08-12','832452','30000000','2021-02-14');
INSERT INTO contract VALUES('202001308931', 'GrandStarex', '9760259686681',
'E004','2020-01-30','547783','23000000','2021-01-23');
INSERT INTO contract VALUES('202006034331', 'GRANDEUR', '8655122983280',
'E002','2020-06-03','980342','41100000','2020-12-25');

```

```

INSERT INTO contract VALUES('202002218021', 'TUCSON', '9883029374933',
'E001','2020-02-21','225472','25000000','2020-11-28');
INSERT INTO contract VALUES('202007123392', 'TUCSON', '8502847249814',
'E001','2020-07-12','488752','25000000','2020-12-17');
INSERT INTO contract VALUES('202008291873', 'GRANDEUR', '8908903423055',
'E002','2020-08-29','345234','41100000','2020-11-19');
INSERT INTO contract VALUES('202006069287', 'KONA', '8908903423055', 'E002','2020-
06-06','085356','20400000','2020-08-08');
INSERT INTO contract VALUES('202012139382', 'POTER2', '8989762351666',
'E013','2020-12-13','392018','16700000','2021-02-28');
INSERT INTO contract VALUES('202004293847', 'Venue', '8502847249814', 'E015','2020-
04-29','349283','19720000','2021-07-21');
INSERT INTO contract VALUES('202005251982', 'GRANDEUR', '8459776110322',
'E014','2020-05-25','122343','41100000','2020-08-17');
INSERT INTO contract VALUES('202003190098', 'TUCSON', '8472309669263',
'E021','2020-03-19','655843','25000000','2020-05-21');
INSERT INTO contract VALUES('202007182732', 'GENESIS_G80', '8360191663389',
'E025','2020-07-18','876543','54000000','2020-10-09');
INSERT INTO contract VALUES('202011284762', 'PALISADE', '8220394826554',
'E032','2020-11-28','986242','46000000','2021-03-08');
INSERT INTO contract VALUES('202005192837', 'GENESIS_G70', '9641635790339',
'E032','2020-05-19','012948','48300000','2020-09-18');

```

-- INSERT production

```

INSERT INTO production VALUES('GENESIS_G80', 'BeltDrive', 'E006');
INSERT INTO production VALUES('GENESIS_GV70', 'AirConditionerFilter', 'E007');
INSERT INTO production VALUES('GRANDEUR', 'Stabiliser', 'E008');
INSERT INTO production VALUES('GRANDEUR_HYBRID', 'HeatExchanger', 'E009');
INSERT INTO production VALUES('SONATA', 'FootwellTray', 'E009');
INSERT INTO production VALUES('GENESIS_G90', 'CoilSpring', 'E010');
INSERT INTO production VALUES('SANTAFE', 'Harness', 'E016');
INSERT INTO production VALUES('POTER2', 'FuelFilter', 'E017');
INSERT INTO production VALUES('PALISADE_AWD', 'OilFilter', 'E018');
INSERT INTO production VALUES('AVANTE_HYBRID', 'PedalLining', 'E019');
INSERT INTO production VALUES('Venue', 'Radiator', 'E020');

```



```

INSERT INTO production VALUES('AVANTE', 'PulseGenerator', 'E020');
INSERT INTO production VALUES('PALISADE', 'CylinderHead', 'E027');
INSERT INTO production VALUES('TUCSON', 'TensionerPully', 'E029');
INSERT INTO production VALUES('GENESIS_GV80', 'Condenser', 'E035');
INSERT INTO production VALUES('GrandStarex', 'HandBrake', 'E033');
INSERT INTO production VALUES('KONA', 'BrakeDisc', 'E028');
INSERT INTO production values('Venue', 'BeltDrive', 'E007');
INSERT INTO production values('AVANTE', 'BeltDrive', 'E007');
INSERT INTO production values('TUCSON', 'Condenser', 'E010');
INSERT INTO production values('AVANTE', 'CoilSpring', 'E018');
INSERT INTO production values('GrandStarex', 'BrakeDisc', 'E035');

```

3. Useful Query(DML)

우리 회사에서는 Customer부터 Production테이블까지 시스템에서 사용되는 모든 테이블들의 데이터베이스를 좀 더 효율적으로 사용하기 위해 몇가지 유용한 쿼리를 제시하고자 한다.

(1) 각 차량별 매출

자동차를 생산하기 위해서는 많은 인력과 부품, 시간이 들어간다. 자동차를 생산할 때 회사가 최대의 이익을 얻기 위해서는 인력과 부품을 효율적으로 활용하여 자동차 생산성을 높여 시간을 절약하여야 한다. 무엇보다 회사의 손해를 줄이는 일이 중요한데, 회사의 손해를 줄이기 위해서는 각 차량 별 매출을 분석하여 소비자들에게 인기가 있는 자동차가 무엇이고 인기가 없는 자동차가 무엇인지 파악하여 인력과 부품을 효율적으로 관리할 필요가 있다.

이 쿼리는 각 차량 별 매출을 보여주는 쿼리로, 각 자동차 이름별로 계약 시 팔렸던 자동차의 판매 가격의 합을 보여줍니다. 이를 통해 매출이 낮은 자동차의 경우 생산을 줄이거나 생산을 중단해야 하고 매출이 높은 것은 생산을 늘려야 한다는 것을 한눈에 알 수 있습니다.

i. Query

```

SELECT car.car_modelname as 자동차_모델명, count(contract.car_modelname)*contract.price as 총매출
FROM car, contract
WHERE car.car_modelname = contract.car_modelname

```

GROUP BY car.car_modelname, contract.price
ORDER BY COUNT(contract.car_modelname)*contract.price DESC;

ii. Result

자동차_모델명	총매출		
1 GRANDEUR	164400000	9 AVANTE	50000000
2 GENESIS_G80	108000000	10 GENESIS_GV70	48300000
3 TUCSON	100000000	11 SONATA	47800000
4 GENESIS_G70	96600000	12 GRANDEUR_HYBRID	43410000
5 GENESIS_G90	95000000	13 KONA	40800000
6 PALISADE	92000000	14 Venue	39440000
7 GrandStarex	69000000	15 POTER2	33400000
8 SANTAFE	60000000	16 AVANTE_HYBRID	23000000

(2) 판매 실적

현대자동차에서 근무하는 판매사원들의 판매량, 총판매가격을 알아보려고 한다. 이 쿼리를 통해 판매사원의 기여도 및 실적을 확인할 수 있으며 판매 실적을 통해 인센티브를 차등 부여, 혹은 진급에 도움이 되는 지표로 사용할 수 있다.

i. Query

SELECT salesemployee_id AS 판매자ID, CONCAT(lname,fname) AS 판매자명,
COUNT(salesemployee_id) AS 판매량, SUM(price) AS 총_판매가격
FROM contract,employee
WHERE salesemployee_id = employee_id
GROUP BY salesemployee_id, lname,fname
ORDER BY 총_판매가격 DESC;

ii. Result

판매자ID	판매자명	판매량	총_판매가격				
1 E002	이나은	5	152600000	9 E023	박예희	1	48300000
2 E001	정광현	3	145000000	10 E022	홍진호	1	43410000
3 E032	김현우	3	140300000	11 E011	손승희	1	41100000
4 E014	한채연	3	115500000	12 E015	양지완	2	36420000
5 E004	최하은	3	83000000	13 E012	배한미	1	25000000
6 E021	노정훈	2	73300000	14 E031	정석우	1	23000000
7 E003	최다현	3	70800000	15 E005	채보훈	1	23000000
8 E025	최원상	1	54000000	16 E024	한나연	1	19720000
				17 E013	김민석	1	16700000

(3) 월(month)별 판매 수량

어떤 달에 가장 많은 자동차 판매가 이루어졌는지, 어떤 달에 판매가 가장 부진한지를 월별로 비교할 수 있다. 2020년 올해, 현대자동차는 11월에 가장 많은 판매가 이루어졌으며, 1월과 2월에 가장 적은 판매가 이루어진 것을 조회할 수 있다. 이를 통해, 판매가 부진했던 다음달에는 더 많은 마케팅과 고객 유치를 통하여 판매량을 증가시켜야함을 알 수 있다.

i. Query

```
SELECT TO_CHAR(contract_date,'yyyy-mm') AS month, COUNT(*) AS 판매수량
FROM contract
GROUP BY TO_CHAR(contract_date,'yyyy-mm')
ORDER BY TO_CHAR(contract_date,'yyyy-mm');
```

ii. Result

MONTH	판매수량	MONTH	판매수량
1 2019-03	1	8 2020-05	2
2 2019-08	1	9 2020-06	3
3 2019-12	1	10 2020-07	3
4 2020-01	1	11 2020-08	3
5 2020-02	1	12 2020-09	3
6 2020-03	2	13 2020-10	3
7 2020-04	2	14 2020-11	4
		15 2020-12	3

(4) 자동차 제작에 참여중인 인원 수

이 query는 자동차를 제작하는 일에 있어 추가적인 엔지니어가 필요한를 파악하기 위해서 효율적으로 작용한다. 즉, 새롭게 인원을 배치할 일이 생길 때 이 query를 활용하여 각 자동차별 제작 인원을 파악한 뒤, 인원이 부족한 곳에 배치할 수 있다는 것이다. 이 query는 구동할 때 별다른 과정 없이 실행하는 것만으로도 결과를 볼 수 있다.

i. Query

```
SELECT CAR_MODELNAME AS 자동차_모델명, COUNT(engineer_id) AS 작업중인_엔지니어_수
FROM production
GROUP BY car_modelname
ORDER BY car_modelname;
```

ii. Result

자동차_모델명	작업중인_엔지니어_수		
1 AVANTE	3	9 GrandStarex	2
2 AVANTE_HYBRID	1	10 KONA	1
3 GENESIS_G80	1	11 PALISADE	1
4 GENESIS_G90	1	12 PALISADE_AWD	1
5 GENESIS_GV70	1	13 POTER2	1
6 GENESIS_GV80	1	14 SANTAFE	1
7 GRANDEUR	1	15 SONATA	1
8 GRANDEUR_HYBRID	1	16 TUCSON	2
		17 Venue	2

(5) 특정 엔지니어가 작업중인 차량을 함께 작업하고 있는 사람들

모든 엔지니어는 모든 차량을 생산할 수 있는 것이 아니며, 각자 하나 이상의 자동차를 생산 할 수 있으며 두 개이상의 차량을 생산할 수 있는 능력을 가진 엔지니어도 존재한다. 이때 특정 엔지니어가 빠지는 상황에서 작업 중인 차량을 마무리 하기 위해서는 인원 충당이 필요하다. 인원 충당을 하기 위해서는 아무나 충당하는 것이 아닌 해당 차량을 작업할 수 있는 엔지니어를 충당해야 하고 따라서 해당 차량을 함께 작업하고 있는 사람들 중에 한 명을 충당해야 한다.

따라서 이 쿼리는 특정 engineer가 작업 중인 차량을 함께 작업하고 있는 사람들의 리스트를 보여주며 인원이 빠졌을 경우 빠르게 인원 충당이 될 수 있도록 도와준다. 실제 공장에서 굉장히 유용하게 쓰일 쿼리이고 빠른 회전을 위해서 필요한 쿼리이다.

i. Query

```

SESELECT DISTINCT engineer_id AS 엔지니어ID, car_modelname AS 자동차_모델명
FROM production
WHERE car_modelname IN
(SELECT car_modelname FROM production WHERE Engineer_ID='E020')
MINUS
SELECT engineer_id AS 엔지니어ID, car_modelname AS 자동차_모델명
FROM production
WHERE Engineer_ID='E020';

```

ii. Result

	엔지니어ID	자동차_모델명
1	E007	AVANTE
2	E007	Venue
3	E018	AVANTE

(6) 해당 부품이 사용되는 자동차의 모델명

자동차 별로 사용되는 부품들의 이름을 확인 할 수 있다. 이 쿼리를 통해 직원들이 차 차량 제조할 때 어떤 부품이 사용되는지 알 수 있고 더 나아가 해당 부품을 구매하기 위해 거래를 해야하는 거래처 정보를 얻기 위한 데이터로 사용될 수 있다.

i. Query

```
SELECT car_modelname AS 자동차_모델명, up_modelname AS 부품이름
FROM production
ORDER BY car_modelname, up_modelname;
```

ii. Result

자동차_모델명	부품이름		
1 AVANTE	BeltDrive	12 GrandStarex	HandBrake
2 AVANTE	CoilSpring	13 KONA	BrakeDisc
3 AVANTE	PulseGenerator	14 PALISADE	CylinderHead
4 AVANTE_HYBRID	PedalLining	15 PALISADE_AWD	OilFilter
5 GENESIS_G80	BeltDrive	16 POTER2	FuelFilter
6 GENESIS_G90	CoilSpring	17 SANTAFE	Harness
7 GENESIS_GV70	AirConditionerFilter	18 SONATA	FootwellTray
8 GENESIS_GV80	Condenser	19 TUCSON	Condenser
9 GRANDEUR	Stabiliser	20 TUCSON	TensionerPully
10 GRANDEUR_HYBRID	HeatExchanger	21 Venue	BeltDrive
11 GrandStarex	BrakeDisc	22 Venue	Radiator

(7) 해당 부품을 구매하기 위해 이용가능한 거래처명과 거래처번호

현대자동차에서 취급하는 45개의 모든 부품명들이 조회된다. 필요한 부품을 구매하기 위해서는 어떤 거래처와 거래해야하는지 해당 부품의 검색을 통해 쉽게 알 수 있다.

i. Query

```
SELECT part.part_modelname AS 부품명,wholesale.name AS 거래처명,wholesale.Phone# AS 거래처번호
FROM part, wholesalepart,wholesale
WHERE part.part_modelname=wholesalepart.wp_modelname
AND wholesalepart.wholesale_license#=wholesale.license#;
```

ii. Result

부품명	거래처명	거래처번호			
1 StarterBattery	우림부품	01018361348	23 ExpansionValve	웅석봉부품	010764835235
2 WheelSensor	배리부품	01061365137	24 RubberBuffer	웅석봉부품	010764835235
3 SpotLight	원주부품	01031263523	25 DustCover	웅석봉부품	010764835235
4 HeadLight	오창부품	01053612432	26 GasSpring	고대부품	01078414332
5 CrankshaftPulse	금산부품	01029438134	27 MirrorGlass	고대부품	01078414332
6 IgnitionCable	성환부품	01048316782	28 InnerWingPanel	고대부품	01078414332
7 WiperBlade	완산부품	01053713652	29 Harness	오창부품	01053612432
8 SideWall	영서부품	010148362326	30 AirConditionerFilter	설악부품	01015313432
9 PlateLight	양재부품	01046213946	31 FuelFilter	부천부품	01049873213
10 VentilationGrille	도봉부품	01081763498	32 OilFilter	대현부품	01041893225
11 JackingPoint	봉천부품	01013481274	33 CylinderHead	성환부품	01048316782
12 HydraulicFilter	나주부품	01038497237	34 TensionerPully	봉천부품	01013481274
13 Gasket	행운부품	01083471983	35 BeltDrive	경인부품	01019436174
14 OilDrainPlug	경인부품	01019436174	36 Radiator	행운부품	01083471983
15 LambdaSensor	설악부품	01015313432	37 Condenser	봉천부품	01013481274
16 TuningModule	김삿갓부품	01082438179	38 HeatExchanger	대현부품	01041893225
17 PariculateFilter	대현부품	01041893225	39 Stabiliser	양산부품	01071237195
18 Manifold	웅석봉부품	010764835235	40 CoilSpring	우림부품	01018361348
19 MiddleSilencer	웅석봉부품	010764835235	41 PulseGenerator	봉천부품	01013481274
20 CatalyticConverter	웅석봉부품	010764835235	42 BrakeDisc	부천부품	01049873213
21 Evaporator	웅석봉부품	010764835235	43 HandBrake	나주부품	01038497237
22 HighPressureLine	웅석봉부품	010764835235	44 FootwellTray	원주부품	01031263523
			45 PedalLining	대현부품	01041893225

(8) 차량에 활용되는 부품의 목록

이 query를 활용하여 검색한 차량에 어떤 부품들이 사용되는지를 보여준다. 즉, 이는 자동차를 생산하며 어떤 부품이 필요한지 알고 싶을 때 활용이 가능하다. 이 query는 차량 이름을 직접 입력하여 (예: 'AVANTE'라고 입력) 해당 입력받은 자동차의 이름을 통해 검색을 진행하게 되는데, 이 때 작은 따옴표("")를 반드시 포함하여 입력하여야 한다.

i. Query

```
SELECT car_modelname AS 자동차_모델명, wp_modelname AS 부품_이름, price AS 가격
FROM production, wholesalepart
WHERE production.up_modelname = wholesalepart.wp_modelname
AND production.car_modelname = &SEARCHMODEL
ORDER BY price;
```

ii. Result

자동차_모델명	부품_이름	가격
1 AVANTE	BeltDrive	180200
2 AVANTE	PulseGenerator	220000
3 AVANTE	CoilSpring	282000

(9) 고객의 조건(성별 등)에 따른 선호 차량

고객이 자동차를 구입하려고 할 때 1:1 고객 맞춤 서비스를 제공하기 위해서 고객과 같은 조건의 고객의 데이터를 분석하여 이에 맞추어 추천하면 더 높은 만족도를 이끌어 낼 수 있다. 이때 성별에 따라 선호하는 자동차의 종류가 달라지는데 성별에 따라 구입한 자동차의 리스트를 중심으로 고객에게 추천하면 구매 확률이 더 높아질 것이다.

이 쿼리는 계약을 했던 고객의 데이터와 차량 데이터를 바탕으로 성별에 따른 선호 차량을 보여준다. 본인이 여성일 경우, 남성이 선호하는 차량을 보길 원한다면 남성이 선호하는 차량도 제공받을 수 있다.

i. Query

```
SELECT customer.sex AS 성별, contract.car_modelname AS 자동차_모델명
FROM customer, contract
WHERE customer.customer_ssn=contract.customer_ssn
GROUP BY customer.sex, contract.car_modelname
ORDER BY customer.sex;
```

ii. Result

성별	자동차_모델명	12 F	Venue
1 F	AVANTE	13 M	AVANTE_HYBRID
2 F	GENESIS_G70	14 M	GENESIS_G70
3 F	GENESIS_G80	15 M	GENESIS_G80
4 F	GENESIS_G90	16 M	GENESIS_GV70
5 F	GRANDEUR	17 M	GRANDEUR
6 F	GRANDEUR_HYBRID	18 M	GrandStarex
7 F	GrandStarex	19 M	PALISADE
8 F	KONA	20 M	POTER2
9 F	POTER2	21 M	SONATA
10 F	SANTAFE	22 M	TUCSON
11 F	TUCSON	23 M	Venue

(10) 보험가입여부에 따른 차량 구입 횟수

시대에 따라 기존에 보험을 가입했는지 여부에 따라 차량을 구입한 횟수는 크게 차이가 난다. 보험에 가입하지 않아도 차량을 구입할 수 있지만 예전에는 보험에 가입하지 않으면 차량을 구입하지 않는 경우가 많았다. 하지만 최근에는 보험에 가입하지 않아도 관상용이나 재산의 목적으로 차량을 구입하는 경우가 많다. 따라서 시대에 따라 보험가입여부에 따라 차량 구입 횟수를 확인할 필요가 있고 이에 맞게 마케팅을 할 필요가 있다.

예를 들어 마케팅을 할 때 최근에 보험에 가입한 고객들이 차량 구입을 더 많이 한다면 모든 고객들을 대상으로 할 것이 아니라 보험가입여부에 따른 차량 구입 횟수의 데이터를 바탕으로 보험에 가입한 고객들에게 마케팅을 집중적으로 한다면 작은 마케팅으로 큰 효과를 볼 수 있을 것이다.

따라서 보험가입여부에 따른 차량 구입 횟수를 보여주는 이 쿼리는 유용하게 활용될 것이며, 주기적으로 확인한다면 마케팅을 할 때 크게 효과를 볼 수 있을 것이다.

i. Query

```
SELECT customer.ins_joinstatus AS 보험가입여부, COUNT(contract.contract#) AS 차량구입횟수
FROM customer, contract
WHERE customer.customer_ssn=contract.customer_ssn
GROUP BY customer.ins_joinstatus
ORDER BY customer.ins_joinstatus;
```

ii. Result

보험가입여부	차량구입횟수
1 N	13
2 Y	20

(11) 차량 가격별 판매량 차이

제조업체에서 만들어지는 차량의 기본가격별 판매량 추이를 알아보려고 한다. 이 쿼리문을 통해 차량의 모델명별로 책정된 기본가격과 해당 모델의 계약건수를 확인할 수 있으며, 이 지표로 차량에 옵션을 붙이기전 가격과 판매량 사이의 상관관계를 분석해 새로 만들 차량의 가격책정에 도움을 줄 수 있다.

i. Query

```
SELECT car.car_modelname AS 모델명, car.price AS 차량가격_옵션X, COUNT(contract#) AS 판매량
FROM car, contract
WHERE car.car_modelname = contract.car_modelname
GROUP BY car.car_modelname, car.price
ORDER BY 판매량 DESC, car.price;
```

ii. Result

모델명	차량가격_옵션X	판매량				
1 TUCSON	24400000	4	9 SANTAFE	29750000	2	
2 GRANDEUR	41080000	4	10 PALISADE	45300000	2	
3 GrandStarex	22090000	3	11 GENESIS_G70	48300000	2	
4 POTER2	16700000	2	12 GENESIS_G80	52910000	2	
5 Venue	18720000	2	13 AVANTE_HYBRID	22200000	1	
6 KONA	20310000	2	14 GRANDEUR_HYBRID	43410000	1	
7 AVANTE	22200000	2	15 GENESIS_GV70	64670000	1	
8 SONATA	23860000	2	16 GENESIS_G90	94550000	1	

(12) 거래처별, 총 거래수

거래처별로 가장 많이 거래한 거래처 순서로 조회된다. 현대자동차는 '웅석봉부품'과 총 8회로 가장 많은 거래를 했음을 알 수 있고, '대현부품', '봉천부품'이 총 4회로 다음 순위를 차지하였다. 이를 통해 현대자동차가 필요한 부품의 재고는 '웅석봉부품'에 많이 존재함을 알 수 있다.

i. Query

```
SELECT wholesale.name AS 거래처명, COUNT(wholesale_license#) AS 총거래수
FROM wholesale,wholesalepart
WHERE wholesale_license#=wholesalepart.wholesale_license#
GROUP BY wholesale.name ORDER BY COUNT(wholesale_license#) DESC;
```

ii. Result

거래처명	총거래수		
1 웅석봉부품	8	11 행운부품	2
2 대현부품	4	12 경인부품	2
3 봉천부품	4	13 나주부품	2
4 고대부품	3	14 도봉부품	1
5 오창부품	2	15 양재부품	1
6 부천부품	2	16 완산부품	1
7 우림부품	2	17 배리부품	1
8 원주부품	2	18 양산부품	1
9 성환부품	2	19 영서부품	1
10 설악부품	2	20 김삿갓부품	1
		21 금산부품	1

(13) 거래처와 거래 가능한 물품들의 재고량

이 query는 어떤 물품들이 거래처와 거래가 가능한지, 그리고 해당 물품이 얼마만큼의 재고량이 남아있는지를 알 수 있다. 즉, 재고량을 수시로 확인하여 필요할 경우 주문해야

하는 engineer의 경우 자주 사용 가능하다. 또한, 만일 필요한 물품이 없다면 새로운 거래처와 거래를 시작하는 등의 작업이 이어질 수 있다. 이 query는 별도의 입력값 없이 실행하여 결과를 볼 수 있다.

i. Query

```
SELECT wp_modelname AS 부품_이름, SUM(quantity) AS 재고량
FROM usingpart, wholesalepart
WHERE wholesalepart.wp_modelname = usingpart.up_modelname
GROUP BY wp_modelname
ORDER BY wp_modelname;
```

ii. Result

부품_이름	재고량		
1 AirConditionerFilter	550	17 JackingPoint	112
2 BeltDrive	450	18 LambdaSensor	100
3 BrakeDisc	190	19 OilDrainPlug	231
4 CoilSpring	600	20 OilFilter	600
5 Condenser	530	21 PedalLining	100
6 CrankshaftPulse	480	22 PlateLight	50
7 CylinderHead	250	23 PulseGenerator	430
8 FootwellTray	600	24 Radiator	300
9 FuelFilter	400	25 SideWall	234
10 Gasket	320	26 SpotLight	750
11 HandBrake	290	27 Stabiliser	100
12 Harness	130	28 StarterBattery	4000
13 HeadLight	242	29 TensionerPully	230
14 HeatExchanger	70	30 VentilationGrille	208
15 HydraulicFilter	250	31 WheelSensor	195
16 IgnitionCable	110	32 WiperBlade	300 +

(14) 연비에 따른 자동차 고르기

자동차를 고를 때 차량의 종류, 너비, 높이, 색깔, 가격 등도 많이 고려하겠지만 가장 많이 고려하는 부분 중 하나가 연비이다. 연비란 연료 1L로 갈 수 있는 거리를 뜻하며, 연비가 높다는 것은 같은 연료량으로 많은 거리를 주행할 수 있다는 의미한다. 또한 연비가 높으면 높을수록 연료비도 아끼고 환경에도 이롭다. 연료비를 아끼기 위해서라도 연비가 높은 제품을 고르려고 하는데 연비가 높은 제품들은 가격이 비싸기 마련이다. 그래서 제품을 구매하려는 고객들마다 연비에 따른 기준이 있다. 예를 들어 연비가 최소 10이상 15이하 였으면 좋겠다는 고객들이 있고, 11이상은 되어야 한다는 고객들이 있다. 따라서 고객들은 자신의 조건을 충족하는 자동차의 리스트만을 보길 원하고 이 쿼리를 적용하면 훨씬 쉽게 볼 수 있을 것이다.

이 쿼리는 연비가 일정한 범위 사이인 것의 자동차 리스트와 연비를 편리하게 볼 수 있으며 연비가 오름차순으로 정렬되어 있어 연비별로 자동차 모델의 이름을 한눈에 파악할 수 있는 장점이 있다.

i. Query

```
SELECT car_modelname, fuelefficiency
FROM car
WHERE fuelefficiency BETWEEN 10 AND
(SELECT MAX(fuelefficiency) FROM car
ORDER BY fuelefficiency;
```

ii. Result

	CAR_MODELNAME	FUELEFFICIENCY
1	GRANDEUR	10
2	TUCSON	11
3	GENESIS_GV80	11
4	GrandStarex	11
5	SONATA	11
6	GENESIS_G70	12
7	GENESIS_GV70	12
8	KONA	12
9	SANTAFE	12
10	GENESIS_G80	13
11	Venue	13
12	GENESIS_G90	14
13	GRANDEUR_HYBRID	16

(15) 일정 대수 이상 혹은 일정 금액 이상의 차량을 구매한 고객의 SSN 명단

이 query는 일정 금액, 혹은 일정 대수 이상의 차량을 구매한 고객의 명단을 보여줌으로서 해당 고객에 대한 혜택을 제공하거나, 해당 고객이 가지고 있는 특징(예: 성별 등)을 통해 이후의 차량을 개발하는데 있어서 어떠한 방식으로 제작한다면 좋을지에 대한 정보를 얻을 수 있다.

이 query를 실행할 때에는 NUM값(구매한 차의 대수), 그리고 TOTALP값(구매 가격)을 숫자로 입력해야한다.

i. Query

```
SELECT customer_SSN AS 고객_주민번호 FROM contract
GROUP BY customer_ssn
```

```
HAVING COUNT(customer_ssn) >= &NUM
UNION
SELECT customer_SSN AS 고객_주민번호 FROM contract
GROUP BY customer_ssn
HAVING SUM(price) >= &TOTALP;
```

ii. Result

	고객_주민번호
1	8908903423055

(16) 올해 나온 신차 중 판매가 완료된 자동차 조회

현대자동차는 2020년 올해, 다양한 신차를 출시하였다. 올해 출시한 자동차 중 어떤 모델이 판매되었는지 쉽게 조회가능하며 이 쿼리를 통해서 내년 2021년에 어떤 모델을 더 제조해야할지 예측할 수 있다.

i. Query

```
SELECT car_modelname FROM car
WHERE Modelyear=2020
INTERSECT
SELECT car_modelname FROM contract;
```

ii. Result

	CAR_MODELNAME
1	AVANTE
2	AVANTE_HYBRID
3	GENESIS_G70
4	GENESIS_G80
5	GENESIS_G90
6	GENESIS_GV70
7	GRANDEUR_HYBRID
8	GrandStarex
9	KONA
10	POTER2
11	TUCSON
12	Venue

VI. 역할 분담

1. 제안서 작성

- ✓ 류채현: 산업/비즈니스 목표 작성
- ✓ 이규연: 서론 작성
- ✓ 이상은: 새로운 DB가 필요한 이유 작성
- ✓ 차유미: 회사의 제품/서비스 작성

2. User Requirement, ER Model

- ✓ 류채현: 산업/비즈니스 목표
- ✓ 이규연: 서론
- ✓ 이상은: 새로운 DB가 필요한 이유
- ✓ 차유미: 회사의 제품/서비스

3. Logical Modeling

- ✓ 공통: 초안 작성 및 취합

4. Implementation in Oracle DBMS

- ✓ 류채현: customer, contract, production 테이블의 INSERT문 작성,
- ✓ 이규연: DDL 작성, 보고서 초안 작성 및 취합
- ✓ 이상은: employee, engineer, salesemployee 테이블의 INSERT문 작성, 보고서의 'How to make E-R Model' 작성
- ✓ 차유미: car, option, part, usingpart, wholesalepart 테이블의 INSERT문 작성
- ✓ 공통: query 작성