



소주 유통관리 시스템

과목명	데이터베이스
제출일	2019.06.07(금)

지도교수	강윤철
팀명	소확행
팀원	B711045 김찬미
	B386074 하승민
	B611241 홍가희
	B735542 홍혜림

목차

1. Project Definition

1.1 프로젝트 목표	1 P
1.2 시스템 구매자 및 사용자	2 P
1.3 프로젝트 범위와 조직도	2 P
1.4 Rich Picture 전제 시스템	3 P
1.5 Rich Picture 심화	3 P
1.5.1 주문관리 Rich Picture	4 P
1.5.2 재고관리 Rich Picture	4 P
1.5.3 판매계약 Rich Picture	5 P
1.5.4 광고계약 Rich Picture	5 P

2. Requirement Analysis

2.1 업무조사 &프로세스 모델링	6 P
2.1.1 주문관리에 대한 시나리오	6 P
2.1.2 판매계약에 대한 시나리오	7 P
2.1.3 광고 마케팅에 대한 시나리오	8 P
2.1.4 재고관리에 대한 시나리오	9 P
2.1.5 발주계획에 대한 시나리오	9 P
2.2 정보 요구사항	11 P
2.2.1 주문 정보 요구사항	11 P
2.2.2 재고관리 정보 요구사항	11 P
2.2.3 광고마케팅 정보 요구사항	12 P
2.2.4 판매계약 정보 요구사항	12 P
2.3 기능요구사항	13 P
2.3.1 기본 정보관리 기능요구사항	13 P
2.3.2 복합 정보관리 기능요구사항	14 P
(주문,재고,판매,광고,기업,제품)	
2.4 Business Rules	15 P

3. DB System Design

3.1 논리적 데이터 모델 설계	16 P
3.1.1 주문관리	17 P
3.1.2 재고관리	18 P
3.1.3 판매계약	18 P
3.1.4 마케팅	19 P
3.2 논리적 데이터 흐름 설계	20 P

3.2.1 DFD Level 0 (Context Diagram)	-----20 P
3.2.2 DFD Level 1 소주 유통관리 전체	-----21 P
3.2.3 DFD Level 2 소주 주문관리	-----21 P
소주 재고관리	----- 22 P
판매 계약관리	----- 22 P
3.3 Data Dictionary	-----23 P
3.4 Schema 정의	-----26 P
3.5 Table 정의	-----31 P
3.6 Procedure 설계	-----34 P
4. 결론 및 소감	-----42 p
5. 부록	-----42 p
6. 참고문헌	-----49 p

그림목차

- [그림 1] 소주유통관리 시스템의 범위
- [그림 2] 프로젝트 조직도
- [그림 3] 전체 시스템 Rich Picture
- [그림 4] 주문관리 Rich Picture
- [그림 5] 재고관리 Rich Picture
- [그림 6] 판매계약 Rich Picture
- [그림 7] 광고계약 Rich Picture
- [그림 8] 주문관리 Swim-Lane Chart
- [그림 9] 판매 계약 Swim-Lane Chart
- [그림 10] 광고 마케팅 Swim Lane Chart
- [그림 11] 재고관리 Swim Lane Chart
- [그림 12] 발주계획 Swim Lane Chart
- [그림 13] 주문 정보 요구사항
- [그림 14] 재고 관리 정보요구사항
- [그림 15] 광고 마케팅 정보요구사항
- [그림 16] 판매계약 정보 요구사항
- [그림 17] 기본정보관리 기능
- [그림 18] 복합정보관리 기능
- [그림 19] 시스템 전체 ERD
- [그림 20] 주문 정보 관리 ERD
- [그림 21] 재고 정보 관리 ERD
- [그림 22] 판매계약 정보 관리 ERD
- [그림 23] 마케팅 정보 관리 ERD
- [그림 24] DFD Level 0 (Context Diagram)
- [그림 25] DFD Level 1 - 소주 유통관리
- [그림 26] DFD Level 2 - 소주 주문 관리
- [그림 27] DFD Level 2 - 소주 재고 관리
- [그림 28] DFD Level 2 - 판매 계약 관리
- [그림 29] Schemas-(8+3) 단계
- [그림 30] DDL문 예시
- [그림 31] Diagram
- [그림 32] 주문 Table
- [그림 33] 상세 주문 Table
- [그림 34] 재고 Table
- [그림 35] 판매계약 Table
- [그림 36] 광고계약 Table

- [그림 37] 입력, 수정, 삭제 및 조회 가능 프로시저
- [그림 38] 주문서 확인 프로시저
- [그림 39] 소주별 주문통계 프로시저
- [그림 40] 제품 특성 확인 프로시저
- [그림 41] 출하정보관리 프로시저
- [그림 42] 판매계약조건 프로시저
- [그림 43] 마트별 판매계약 프로시저
- [그림 44] 모델별 광고 종류 및 횟수 프로시저
- [그림 45] 직원정보관리 프로시저
- [그림 46] 부서 실적 조회 프로시저

표목차

- [표 1] Schemas-1단계
- [표 2] Schemas-2단계
- [표 3] Schemas-3단계
- [표 4] Schemas-6단계
- [표 5] Schemas-8단계
- [표 6] Schemas-(8+3) 단계
- [표 7] Schemas-9단계

1. Project Definition

회식이나 각종 모임에서 빠질 수 없는 것 중 하나가 ‘술’이다. 그 중 제일 흔한 술은 ‘소주’이다. 전 세계적으로 한국의 소주는 판매량 1위를 차지하고 있고 그중 94%를 자국민이 차지한다. 그만큼 우리나라 사람들의 소주 소비율이 상당하다.

소주는 우리나라에서 흔하게 볼 수 있으며 현대 한국인들은 소주와 밀접한 관련을 갖고 있다. 술집에서는 물론이거니와 각종 음식점 및 편의점 등 다양한 곳에 상당한 부분을 차지하고 있다. 많은 사람들이 삶의 피로를 위로하기 위해, 주변 사람들과 친밀한 자리를 갖기 위해 소주를 즐겨 찾아 마시고 있다. 또한 학교 학과나 동아리 행사, 회사 등에서 행사를 위한 구매, 뿐만 아니라 음식점이나 마트에서 소매업을 위해 도매로 많은 양을 구매하기도 한다. 그렇기에 소주의 유통이 빠르게 돌아가고 이에 대해 생산량도 상당할 것 이라고 생각된다.

한국의 대표적인 술인 소주는 편의점, 마트, 음식점, 술집 등 흔하게 접할 수 있다. 그래서 대부분의 사람들은 공장에서 소주를 만들면 파는 단순한 시스템이라 생각할 수 있다. 하지만 이 소주 하나를 생산하고 유통하고 판매하기까지의 과정에는 생각보다 많은 프로세스들이 있다. 그래서 소주를 중심으로 본사, 공장, 연구소, 유통망 등의 관계를 파악하고 데이터 베이스 구축을 개발하고자 이 주제를 선정하게 되었다.

소주라는 제품의 생산과정을 생각해 보았을 때 많은 사람들과 단체들이 연관되어 있었고 생산과 소비 공정 또한 객체를 구분하고 관계성을 분석할 수 있도록 경계가 명확하다는 장점이 있었다. 그래서 우리는 소주 공장과 본사, 소비자의 유형의 관계성을 분석할 수 있는 데이터베이스를 구축해 소주 생산 기업이 용이한 데이터를 이용할 수 있도록 할 예정이다.

1.1 프로젝트 목표

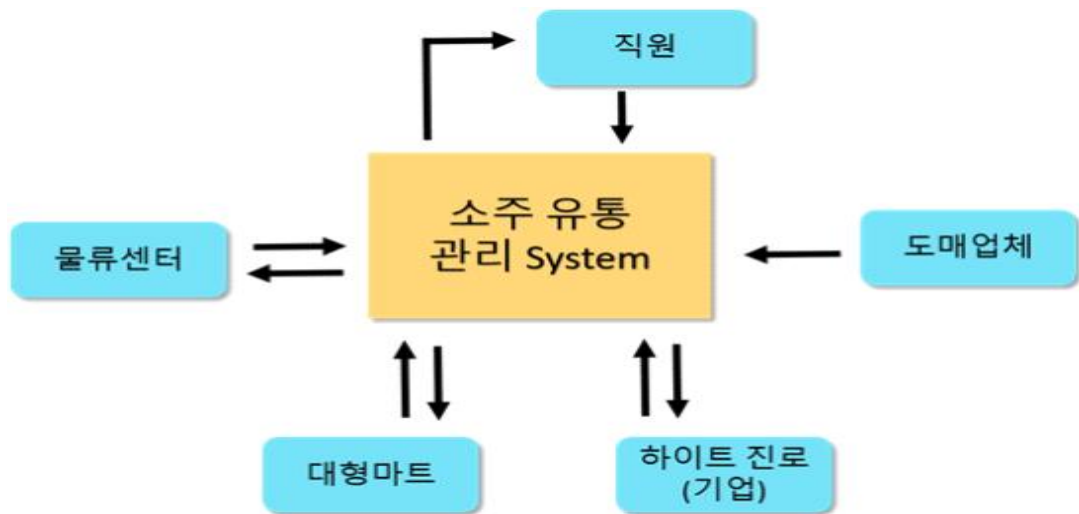
소주 생산 본사에서 소주의 생산과 운영, 관리를 할 수 있는 시스템으로 회사 대표가 생산과정 관계성과 현황을 한눈에 파악할 수 있도록 도움을 주는 것을 목표로 한다. 본사는 이 시스템을 통해 각 process에서 업무가 어떻게 진행될 것이고, 진행 결과가 어떤지 확인할 수 있고, 지난 자료를 통해 앞으로의 의사결정을 내릴 때 본사에 최적의 이익을 낼 수 있는 대안을 선택할 수 있게 한다.

즉, 소주가 우리가 즐겨 마실 수 있는 술에서 끝날 것이 아니라 이것에 대한 상당한 양의 생산과정 및 유통과정 그리고 그것을 판매하기 위한 마케팅 전략, 소비자와 판매자의 유형, 그리고 제일 상위 계층인 소주 판매 기업들까지 전반적인 부분에 대해서 연관되어있는 과정들에 대해 알아보고자 한다.

1.2 시스템 구매자 및 사용자

- 시스템 구매자 : 하이트 진로(기업)
: 데이터 베이스 시스템을 이용해 마케팅, 재고관리, 판매계약, 주문관리 등 소주 유통과정 현
황을 효율적으로 관리하고 개선하고자 하는 객체이다.
- 시스템 사용자 : 하이트 진로(기업), 직원, 물류센터, 도매업체, 대형마트
:데이터 베이스 시스템을 직접 이용하는 사용자으로써 업무 담당자이다.

1.3 프로젝트 범위와 조직도



[그림 1] 소주유통관리 시스템의 범위

[그림1]은 소주유통관리 시스템의 범위를 나타낸 것으로 시스템 구매자인 하이트진로(기업)와 도매업체, 직원, 물류센터, 대형마트의 정보교환 과정을 나타내고 있다.

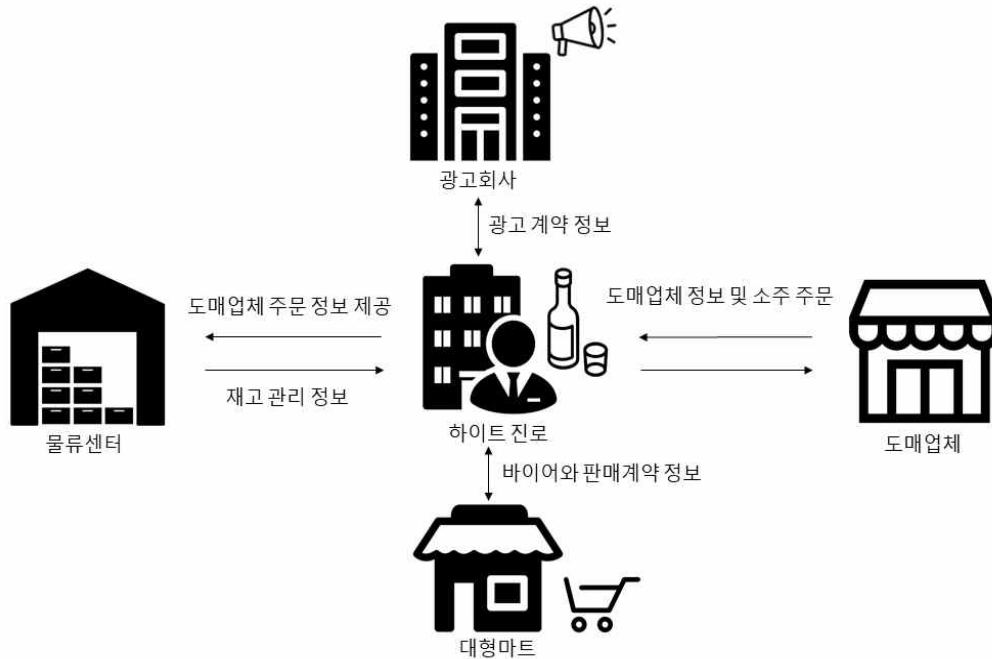


[그림 2]는 소주유통관리시스템의 데이터베이스 구축을 위한 프로젝트 조직도이다. 각자 팀원들에게 적절한 업무분배를 통해 모두가 함께하
는 프로젝트를 진행하고자 했으며 해당 시스템의 사용자는 하이트진로(기업)가 될 것이다.

[그림 2] 프로젝트 조직도

1.4 Project Rich Picture 전체 시스템

시스템 선정 및 범위를 구체화하기 위해서 상위 Rich Picture를 먼저 그려보았다.



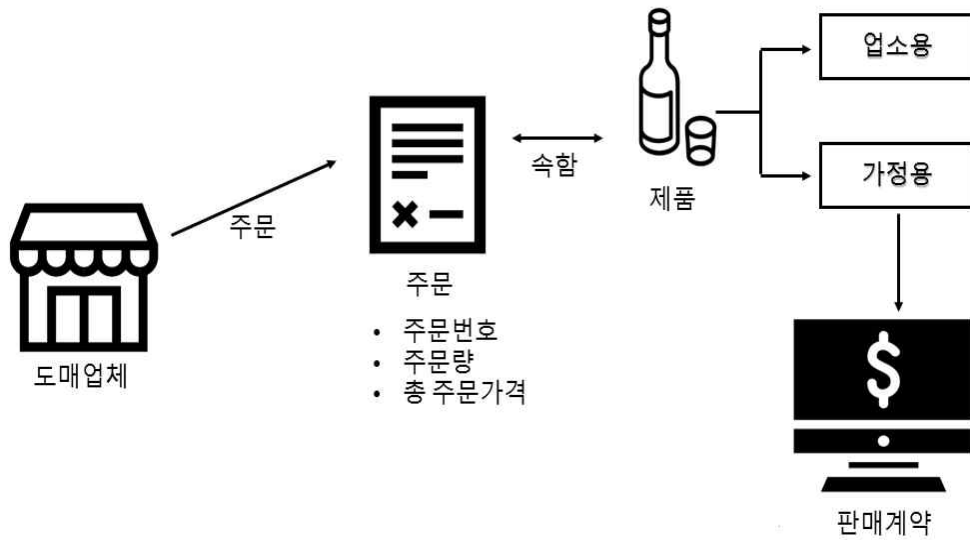
[그림 3] 전체 시스템 Rich Picture

[그림 3]은 소주유통관리 시스템 전체의 복잡한 데이터 흐름 과정을 Rich Picture로 나타낸 것이다. 시스템은 하이트진로(기업)를 중심으로 광고회사, 도매업체, 대형마트, 물류센터 사이에 상호작용을 통해 다양한 정보와 활동들이 발생한다. 소주 유통과정 전체 시스템을 Rich Picture로 표현함으로써 데이터베이스를 통해 알고자 하는 내용들에 대해 생각해 보았다. 먼저, 도매업체가 주문을 하면 도매업체의 정보와 소주 주문 정보가 기업으로 들어가고, 이 주문 정보는 물류센터로 넘어가게 된다. 물류센터는 재고 관리 정보를 기업에게 준다. 그리고 대형마트의 바이어와 판매계약을 통해 대형마트로 소주를 판매하는 정보를 가질 수 있고, 광고회사와도 광고계약 정보를 주고 받는다.

1.5 Project Rich Picture 심화

상위 Rich Picture를 보고 더 자세히 다뤄져야 할 주문관리, 재고관리, 판매계약, 마케팅(광고계약)에 대해 심화 Rich Picture를 그려보았다. 각 Rich Picture를 순서대로 하나씩 설명 하겠다.

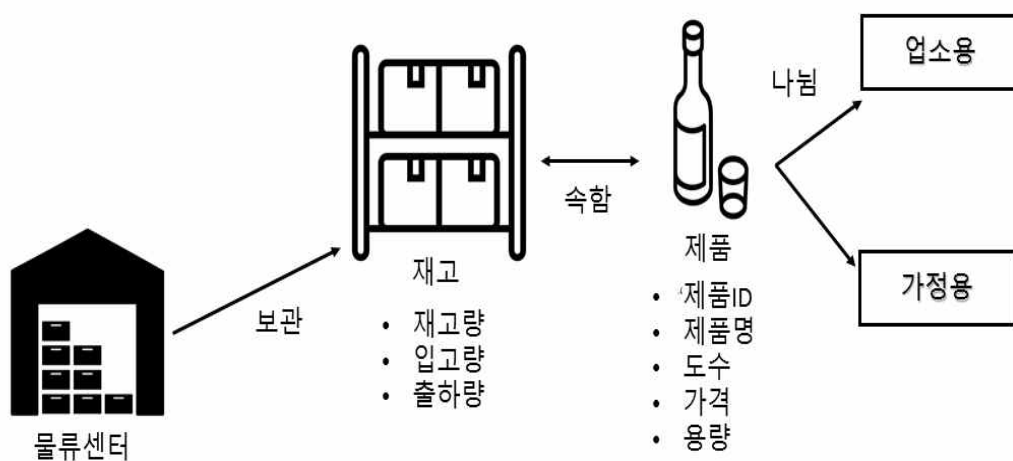
1.5.1 주문관리 Rich Picture



[그림 4] 주문관리 Rich Picture

[그림 4] 와 같이 도매업체에서 주문 번호, 주문량, 총 주문가격 등의 데이터를 바탕으로 주문을 한다. 이때 주문과 제품은 서로 속하는 관계에 있으며 제품 자체는 업소용과 가정용으로 나누어진다. 특히 제품 중에서 가정용 판매 목적의 제품이 대형마트와 판매계약이 이루어지게 된다.

1.5.2 재고관리 Rich Picture

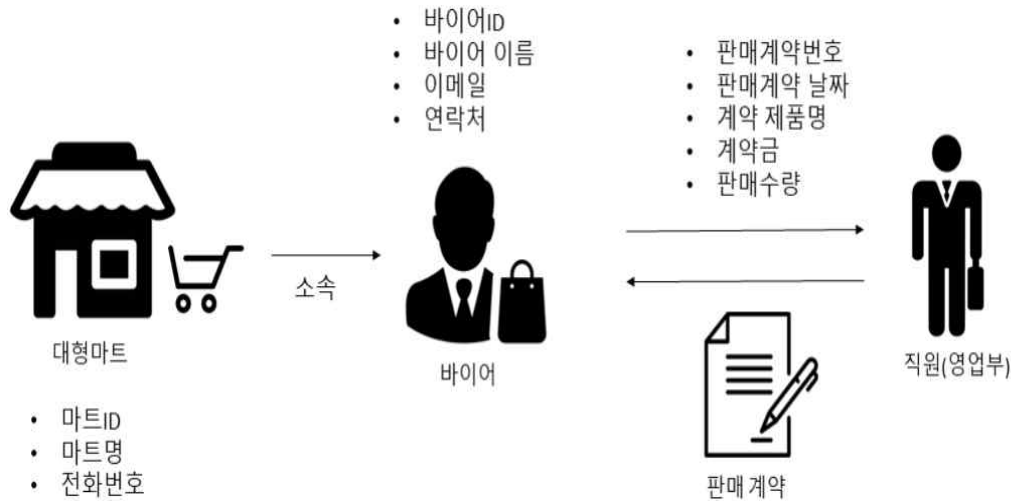


[그림 5] 재고관리 Rich Picture

[그림 5]은 물류센터에 보관 되어지는 재고관리 데이터의 흐름을 보여주는 그림이다. 물류센터에서 재고를 보관하면서 재고량, 입고량, 출하량, 입력날짜 등의 데이터를 입력하고 관리한

다. 또한 재고와 제품이 서로 속하는 관계를 가지고 있다. 여기서 제품은 제품ID, 제품명, 도수, 가격, 용량 등의 정보를 가지고 있다.

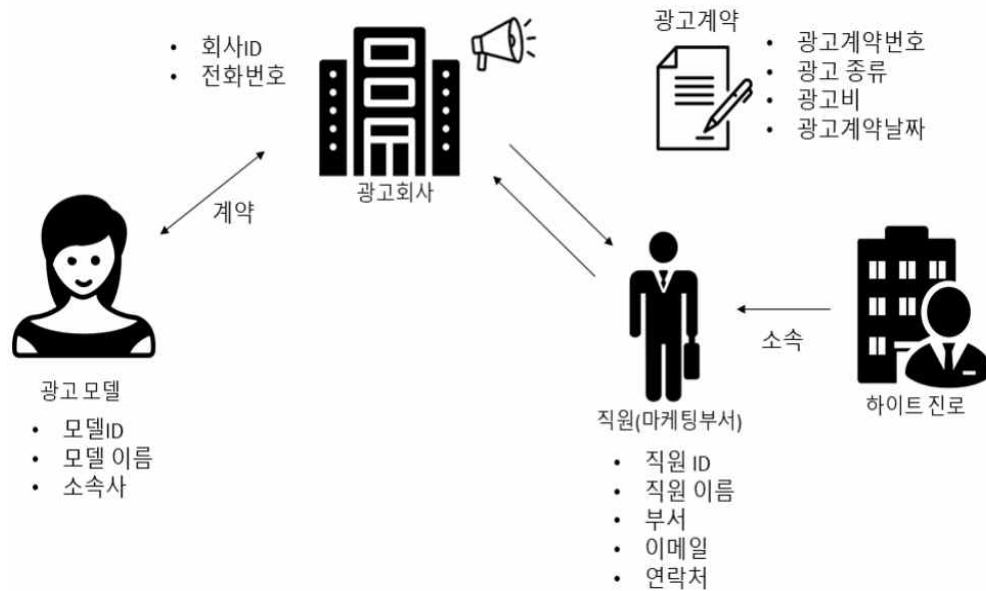
1.5.3 판매계약 Rich Picture



[그림 6] 판매계약 Rich Picture

[그림 6]은 대형마트의 바이어 중심 데이터 흐름과정을 Rich Picture로 간략히 나타낸 것이다. 대형마트에 속해있는 바이어가 하이트 진로(기업) 소속 영업부 직원과 판매계약번호, 판매계약날짜, 계약제품명, 계약금, 판매수량 등의 데이터를 바탕으로 판매계약을 맺는 Rich Picture이다.

1.5.4 마케팅 Rich Picture



[그림 7] 마케팅 Rich Picture

[그림 7]는 광고회사와 화이트 진로(기업)가 광고계약을 맺는 과정이다. 하이트진로 소속 마케팅부서 직원이 광고회사와 광고계약번호, 광고종류, 광고비, 광고계약날짜 등에 대해서 광고 계약을 맺는다. 또 광고회사에서는 원하는 광고 모델과 모델 ID, 모델이름, 소속사 등에 대해 광고 계약을 맺게 된다.

2. Requirement Analysis

앞 장에서는 구현하고자 하는 데이터베이스 시스템을 개략적인 Rich Picture를 통해 도식화해 보았다. 이번 장에서는 해당 시스템을 실제로 사용하게 될 소주 유통관리 시스템의 업무 분석을 통해, 사용자에게 제공되어야 할 적합하고 유익한 기능들을 찾아 정의하려한다.

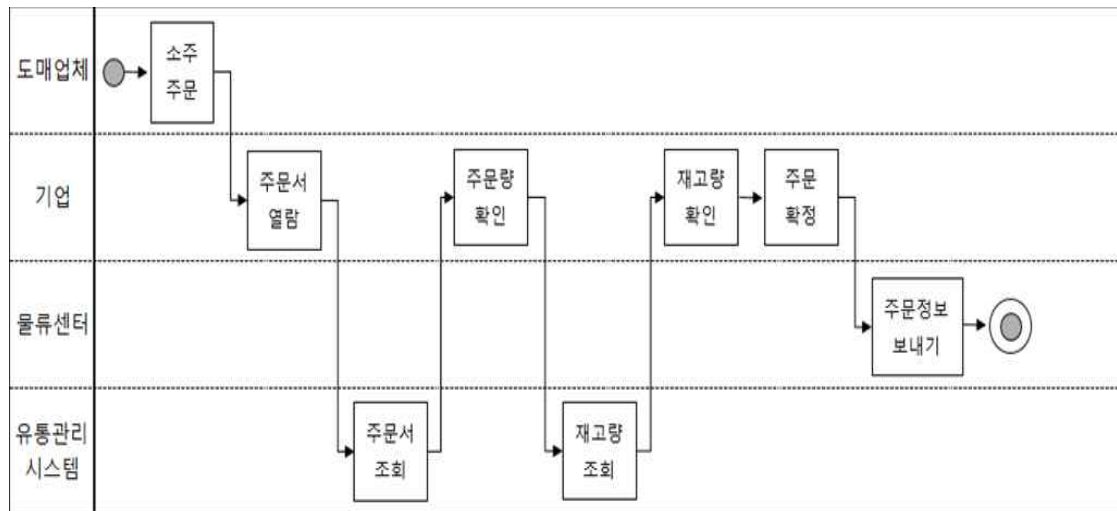
2.1 업무조사 및 프로세스 모델링

소확행 팀은 소주유통관리 시스템의 효율성을 도모하기 위해 하이트 진로 사 고객센터에 여러 차례 전화를 걸어 소주의 생산부터 소비자에 이르기까지의 과정을 물어보았다. 뿐만 아니라 인터넷을 통해 소주 제품에 따른 유통절차의 차이 또한 알아볼 수 있었다. 기존에 이러한 유통과정에 대한 데이터베이스 시스템이 도입되어 있지 않았다. 따라서 본 데이터베이스를 구축하여 관리했을 때, 하이트 진로 사에 유익한 정보를 줌과 동시에 데이터베이스를 통해 효율적인 관리가 이루어져야 한다. 업무조사 및 프로세스 모델링 절에서는 데이터베이스를 통해 관리하고자 하는 업무 별로 나누어 시나리오를 작성하고, 시나리오를 통해 새로운 시스템 상에서 사용자 업무를 파악했다. 또 시나리오를 토대로 프로세스 모델링을 실시하고 Swim Lane Chart 로 나타내 보았다.

2.1.1 주문관리에 대한 시나리오

1. 도매업체는 기업에게 소주를 주문한다.
2. 기업은 도매업체의 주문서를 열람한다.
3. 기업은 주문서의 주문량을 확인한다.
4. 기업은 물류센터의 재고량을 확인한다.
5. 기업은 주문을 확정짓고 물류센터로 주문정보를 보낸다.

© 주문관리에 대한 시나리오를 토대로 작성된 Swim-Lane Chart



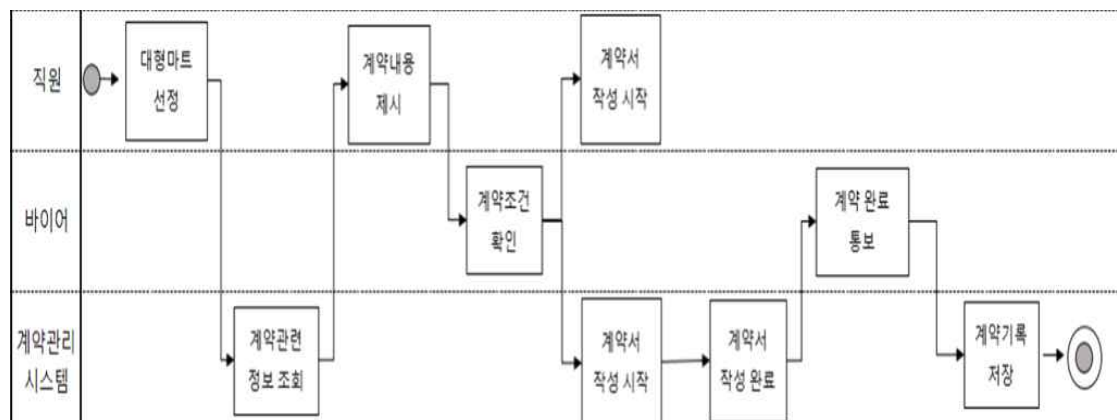
[그림 8] 주문관리 Swim-Lane Chart

위의 [그림 8]은 주문관리에 대한 시나리오를 통해 주요 Act를 도식화 한 Swim-Lane 차트이다. 이 경우 도매업체에서 소주 주문을 하는 데서 시작한다. 그 다음 기업에서 데이터베이스를 통해 주문서를 열람하고 확인한다. 또 주문을 확정 짓기 위해 기업은 데이터베이스를 통해 재고량을 확인하고 주문을 확정 짓는다. 마지막으로 기업은 확정된 주문서를 바탕으로 주문정보를 물류센터에 보내게 된다.

2.1.2 판매 계약에 관한 시나리오

1. 영업직원이 판매계약 조건을 제시한다.
2. 대형마트 바이어가 판매계약 조건을 확인한다.
3. 직원은 계약서를 작성한다.
4. 계약서 작성을 완료한다.
5. 판매 계약을 체결한다.
6. 계약 기록을 저장한다.

◎ 판매 계약에 관한 시나리오를 토대로 작성된 Swim-Lane Chart



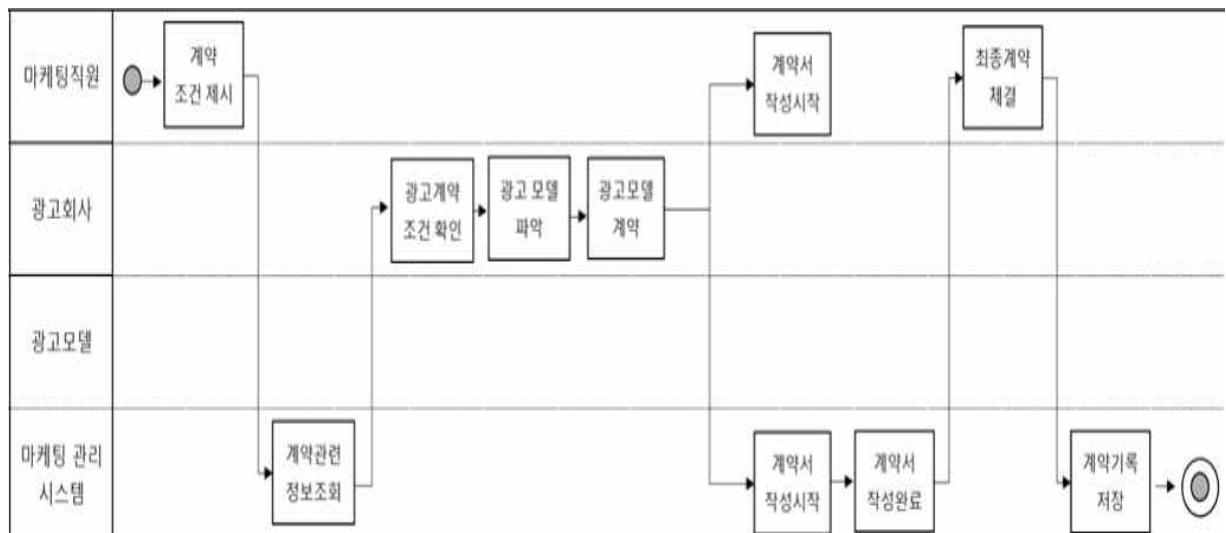
[그림 9] 판매 계약 Swim-Lane Chart

위의 [그림 9]는 판매 계약에 대한 시나리오를 통해 주요 Act를 도식화 한 Swim-Lane 차트이다. 판매 계약을 맺기 위해 직원이 적당한 대형마트를 선정한다. 그 후 데이터베이스를 통해 계약관련 정보를 조회하고 바이어에게 계약내용을 제시한다. 바이어는 직원이 제시한 계약조건을 확인하고 직원이 계약서를 작성하게 한 후 계약 완료를 통보한다. 이 때 계약관리 시스템 내의 데이터베이스에 계약관련 기록이 저장된다.

2.1.3 광고 마케팅에 대한 시나리오

1. 마케팅 직원이 광고계약 조건을 제시한다.
2. 광고회사가 광고계약 조건을 확인한다.
3. 광고회사가 조건에 맞는 광고모델과 계약한다.
4. 직원은 계약서를 완료한다.
5. 계약서 작성을 완료한다.
6. 광고계약을 체결한다.
7. 계약 기록을 저장한다.

◎ 광고 마케팅에 대한 시나리오를 토대로 작성된 Swim Lane chart



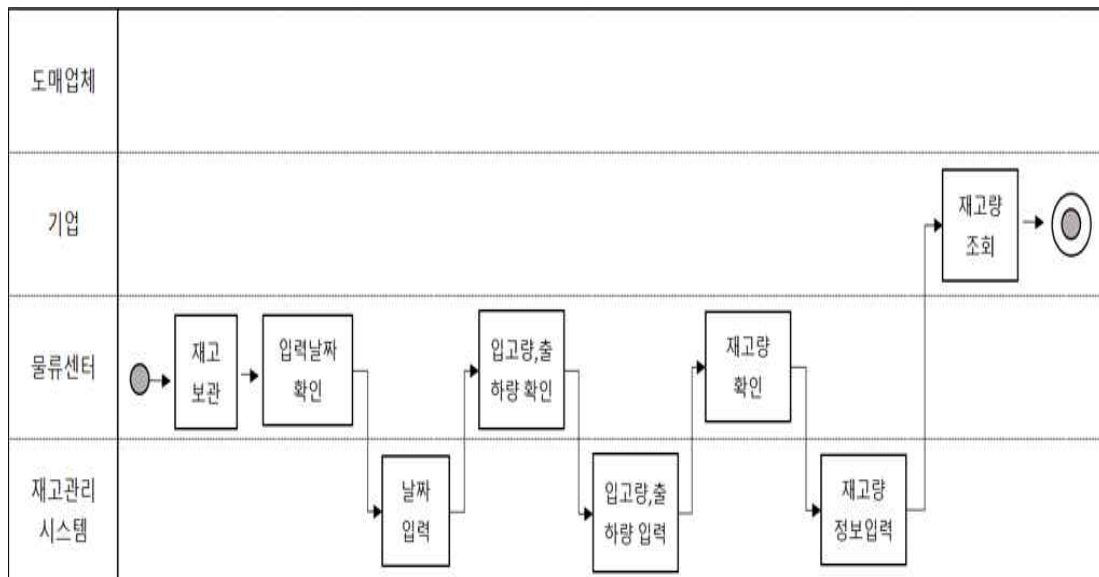
[그림 10] 광고 마케팅 Swim Lane Chart

위의 [그림 10]은 광고 마케팅에 대한 시나리오를 통해 주요 Act를 도식화 한 Swim Lane 차트이다. 광고 계약을 맺기 위해 하이트 진로(기업)의 마케팅 직원이 계약 조건을 제시한다. 마케팅 관리 시스템을 통해 광고 계약관련 정보를 조회할 수 있다. 이에 따라 광고회사는 광고계약 조건을 확인하고 그에 맞는 광고 모델을 찾고 계약을 맺는다. 광고 조건과 광고모델 선정이 완료되면 하이트 진로의 마케팅 직원이 광고 계약서를 작성하고 최종계약을 체결한 뒤 계약 기록을 데이터베이스에 저장한다.

2.1.4 재고관리에 대한 시나리오

1. 물류센터에서 재고를 보관한다.
2. 재고 보관 날짜를 입력한다.
3. 재고의 입고량과 출하량을 기록한다.
4. 재고량을 확인한다.
5. 재고량 정보를 저장한다.

◎ 재고관리에 대한 시나리오를 토대로 작성된 Swim Lane Chart



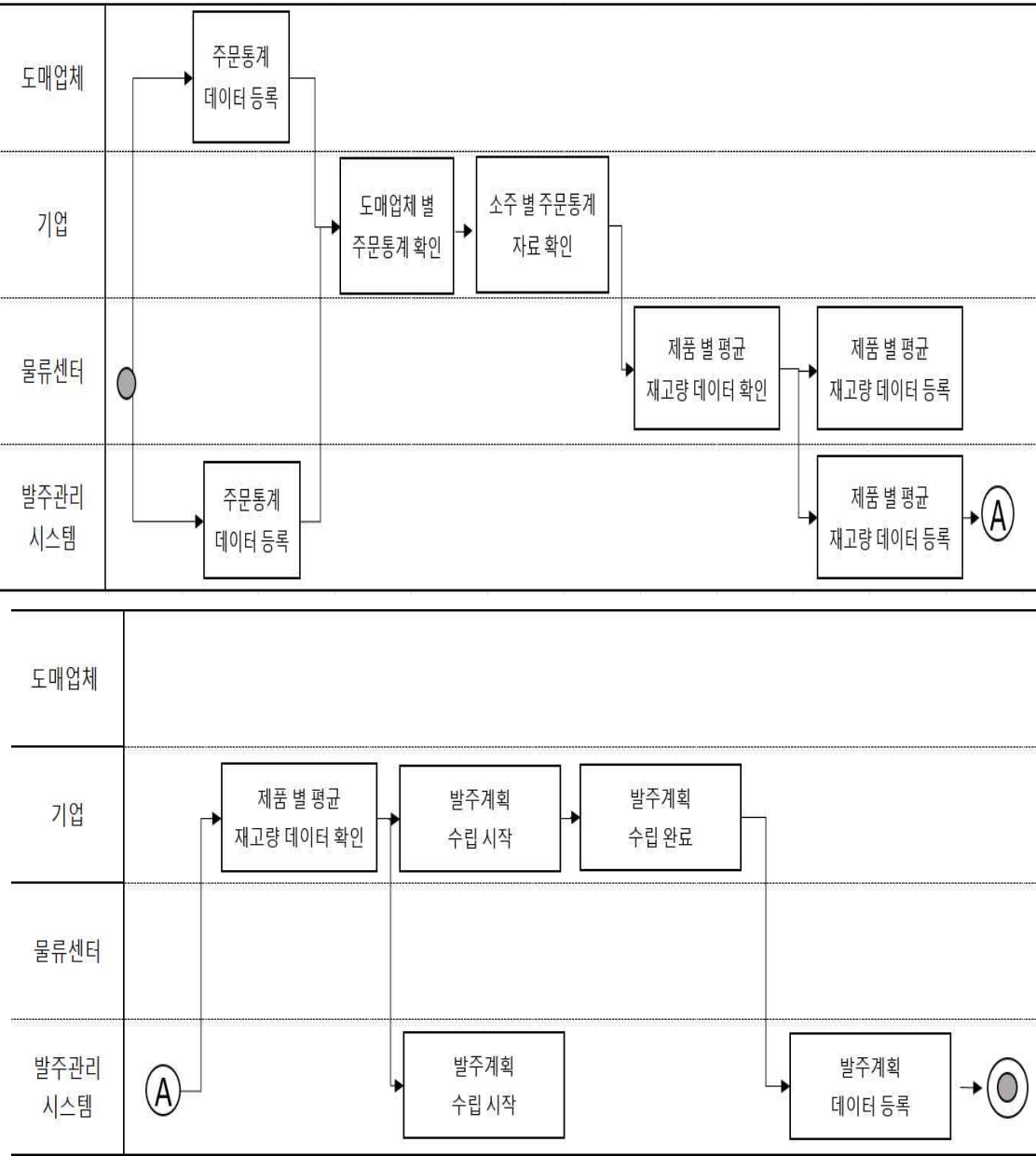
[그림 11] 재고관리 Swim Lane Chart

위의 [그림 11]은 재고관리에 대한 시나리오를 통해 주요 Act를 도식화 한 Swim Lane 차트이다. 물류센터를 통해 재고가 보관되면서 시작한다. 물류센터에서 재고의 입력날짜를 확인하고 데이터베이스에 날짜 데이터를 저장한다. 그 후 재고 입고량, 출하량을 확인하고 또 입고량, 출하량 데이터를 데이터베이스에 저장한다. 마지막으로 물류센터에서 입고량, 출하량 정보를 통해 재고량 정보를 확인하고 그 정보를 데이터베이스에 저장한다. 이렇게 데이터베이스에 저장된 재고량 정보를 기업이 조회에서 볼 수 있게 된다.

2.1.5 발주계획에 대한 시나리오

1. 기업이 도매업체 별로 주문 통계를 확인한다.
2. 소주 별 주문 통계자료를 확인한다.
3. 물류센터의 제품 별 평균 재고량을 확인한다.
4. 통계 자료를 기반으로 기업이 발주 계획을 세운다.

◎ 발주계획에 대한 시나리오를 토대로 작성된 Swim Lane Chart



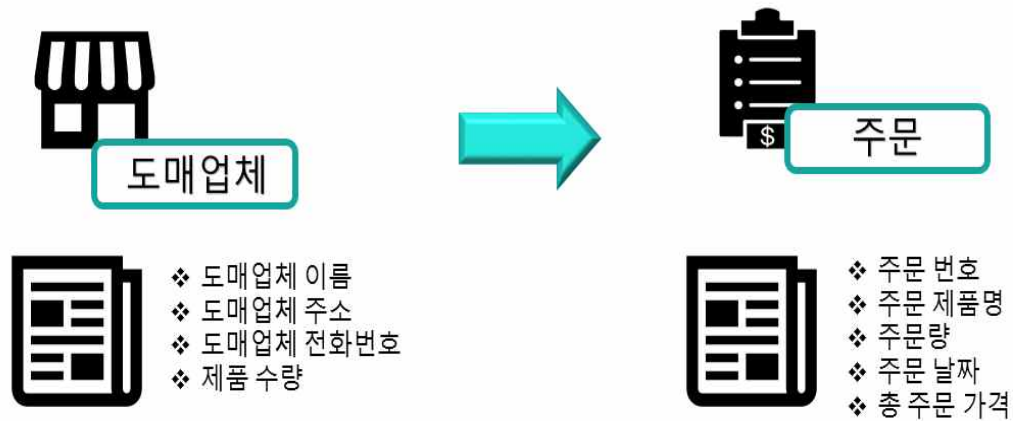
[그림 12] 발주계획 Swim Lane Chart

위의 [그림 12]은 발주계획에 대한 시나리오를 통해 주요 Act를 도식화 한 Swim Lane 차트이다. 발주계획은 도매업체에서 주문통계 데이터를 발주관리 시스템에 등록하고 기업이 도매업체 별 주문통계를 확인하면서 시작된다. 그 다음 기업이 소주 별 주문통계 자료를 확인한다. 기업이 재고 데이터를 열람하기 위해 물류센터는 제품 별 평균 재고량 데이터를 확인한 후 발주관리 시스템에 제품 별 평균 재고량 데이터를 등록한다. 등록된 데이터를 기업이 확인하고 발주계획 수립을 시작한다. 그 후 기업은 발주계획 수립을 완료하고 발주계획 데이터를 발주관리 시스템에 등록한다.

2.2 정보 요구사항

업무 조사 단계의 문제 정의 및 시나리오에서 Swim-Lane Chart를 통해 소주 유통과정에서 일어나는 업무들의 과정을 알아보았다. 위의 정보를 통해 각 업무 별 필요한 정보에 대해 정의해본다.

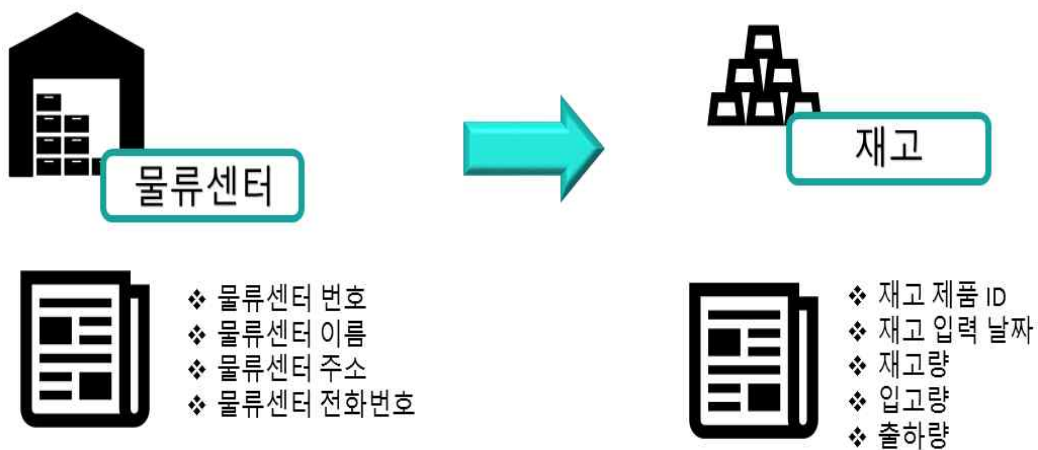
2.2.1 주문 정보요구사항



[그림 13] 주문 정보 요구사항

[그림 13]은 주문 관리를 위한 정보 요구사항을 나타낸 것이다. 모든 주문은 주문번호를 통해 관리된다. 도매업체에서 도매업체 이름, 도매업체 주소, 제품 수량정보를 바탕으로 하이트 진로(기업)에 주문을 요청하면 주문번호, 주문 제품명, 주문량, 주문날짜, 총 주문가격정보를 바탕으로 데이터가 관리된다.

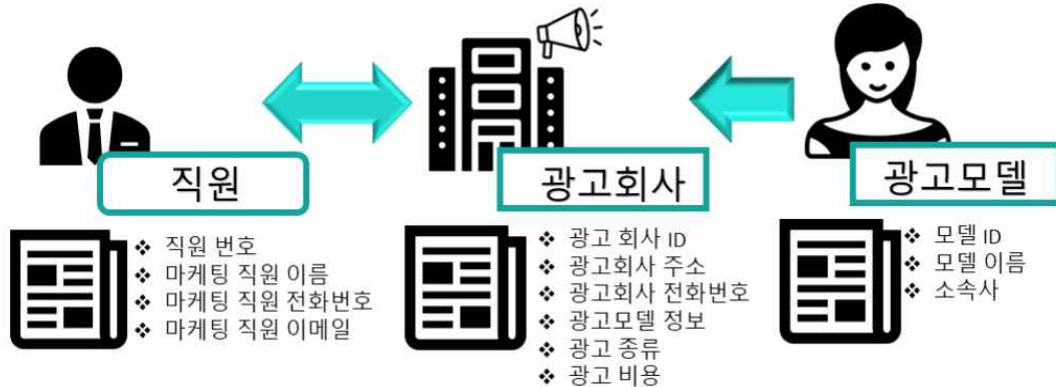
2.2.2 재고 관리 정보요구사항



[그림 14] 재고 관리 정보요구사항

[그림 14]은 재고 관리를 위한 정보 요구사항을 나타낸 것이다. 모든 재고는 재고 제품 ID와 재고코드를 통해 관리된다. 재고는 재고 제품 ID, 재고코드, 재고 입력날짜, 재고량, 입고량, 출하량 정보를 바탕으로 물류센터를 통해 관리된다. 물류센터는 재고관리 정보와 함께 물류센터번호를 기본으로 물류센터이름, 물류센터주소, 물류센터 전화번호 등 물류센터 데이터 정보도 요구된다.

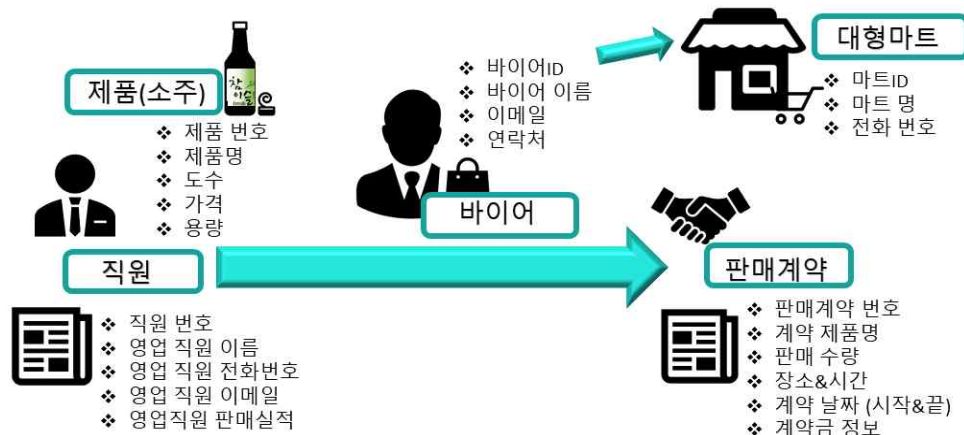
2.2.3 광고 마케팅 정보요구사항



[그림 15] 광고 마케팅 정보요구사항

[그림 15]는 광고마케팅을 위한 정보요구사항을 나타낸 것이다. 광고 마케팅은 직원, 광고회사, 광고모델에 의해서 이루어진다. 첫 번째 광고계약은 하이트 진로(기업)의 마케팅 직원과 광고회사 간의 계약이며 두 번째 광고계약은 광고회사와 광고모델 간의 계약에 의해 나타난다. 광고회사는 광고회사 정보와 광고모델 정보는 각각 광고회사 ID, 광고모델 ID를 통해 관리된다. 첫 번째 광고계약에서 필요한 정보요구사항은 직원번호, 마케팅 직원이름, 마케팅 직원 전화번호, 마케팅 직원 이메일과 관련된 데이터를 통해 직원정보 데이터가 요구되며 광고회사 ID, 광고회사 주소, 광고회사 전화번호, 광고모델정보, 광고 종류, 광고비용 등의 광고회사 데이터가 요구된다. 두 번째 광고계약에서는 광고회사 데이터와 모델ID, 모델 이름, 소속사와 같은 광고 모델 데이터 정보요구사항이 요구된다.

2.2.4 판매계약 정보요구사항



[그림 16] 판매계약 정보 요구사항

[그림 16]은 판매계약을 위한 정보요구사항을 나타낸 것이다. 모든 판매계약은 판매계약번호를 통해서 이루어진다. 판매계약 데이터에는 판매계약번호 외에, 계약제품명, 판매수량, 장소 및 시간, 계약날짜(시작과 끝), 계약금 정보에 관한 것이 요구된다. 판매계약 데이터는 하이트진로(기업)의 직원에 의해서 생성된다. 직원에 대한 정보에는 직원번호, 영업직원 이름, 전화번호, 이메일, 판매실적 데이터가 있다. 직원과 판매계약을 맺는 주체는 대형마트 소속 바이어가 되며 대형마트와 바이어에 대한 정보가 요구된다. 뿐만 아니라 판매계약 물품인 소주에 관한 제품 데이터도 요구된다.

2.3 기능 요구사항

소주 유통관리 시스템을 통해 저장한 정보들을 이용하여 기업을 기반으로 한 전체 유통 관련 업체들의 제품에 대한 어떤 기능들을 수행할 수 있는지 알아보았다.

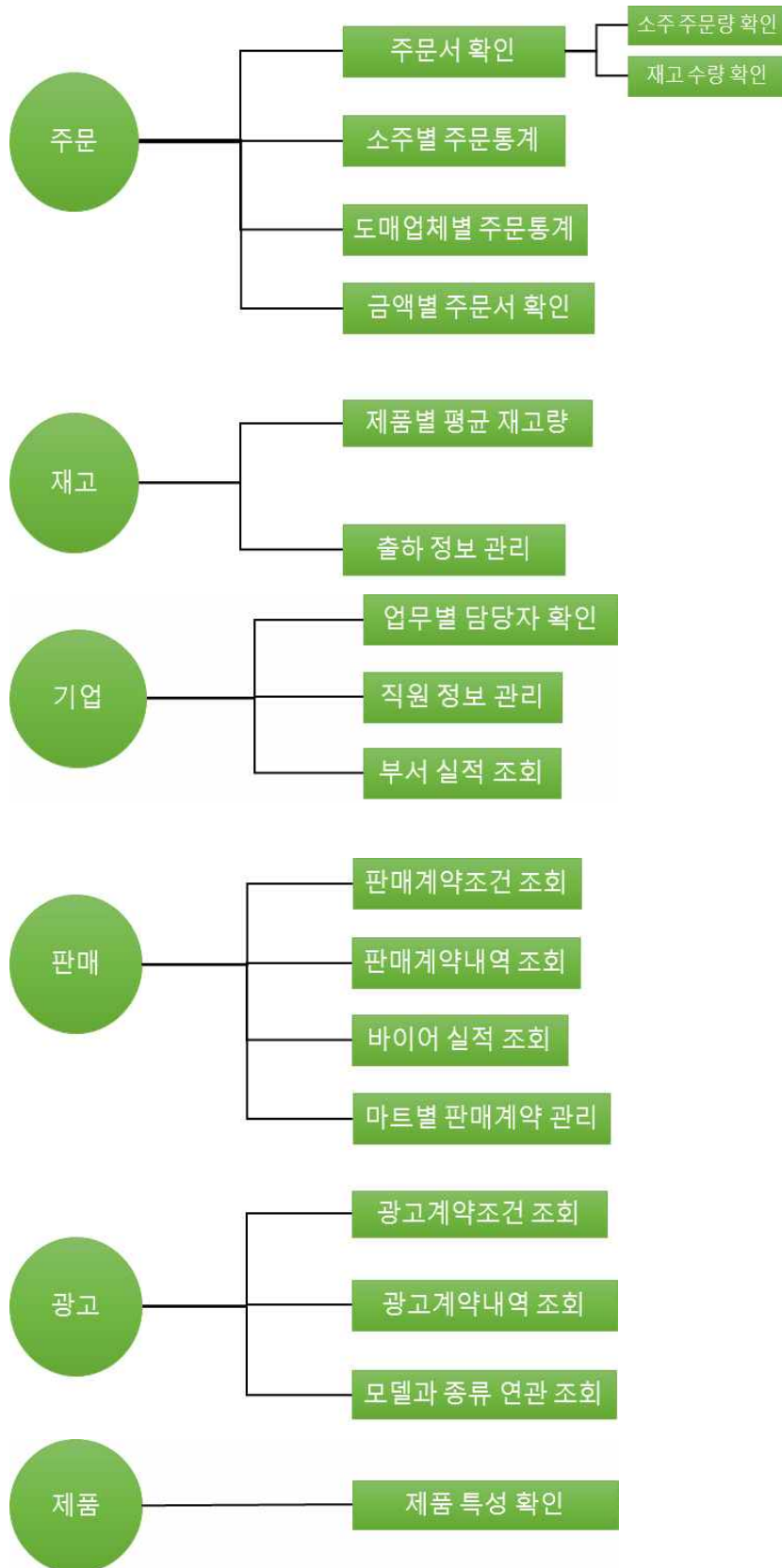
2.3.1 기본 정보 관리 기능요구사항



[그림 17] 기본정보관리 기능

기본적으로 소주 유통 관리 시스템을 통해 제품에 대한 정보를 등록하고 조회할 수 있으며 수정과 삭제가 가능하다. 이러한 정보를 활용하여 기업 및 업체들은 제품을 주문하고 재고량이 어떻게 되는지 판매계약과 광고계약에 대한 부분에 대해서도 열람할 수 있고 각 정보를 관리할 수 있다.

2.3.2 복합 정보 관리 기능요구사항 (주문, 재고, 판매, 광고, 기업, 제품)



[그림 18] 복합정보관리 기능

복합적으로는 더 나아가 각 그룹별로 여러 기능들을 요구한다. 먼저, 주문 관련으로는 주문서 확인으로 소주 주문량과 재고 수량을 한 번에 확인 할 수 있는 기능과 소주별, 도매업체별 주문 통계 기능이 있다. 재고의 경우에는 제품별 평균 재고량을 통해 통계적으로 활용하기 원하고 제품이 출하되어 어느 도매업체로 갔는지 확인할 수 있는 출하 정보 관리 기능도 필요하다. 또한, 기업은 업무별로 어떤 직원이 담당했는지 알고 싶어 하며, 부서 실적 조회 기능을 통해 인사 이동이나 승진 평가 등에 참고 하려 한다. 판매와 광고는 비슷하게 계약 조건을 조회하거나 내역을 볼 수 있으며, 판매계약의 경우 마트별로 관리를 할 수 있는 기능을 원한다. 마지막으로 기업이 판매하는 제품들에 대한 정보를 얻을 수 있는 기능을 원한다.

2.4 Business Rules

앞에서 정의한 활동들과 사용자의 요구 사항 분석을 통하여 소주 관련 유통에 관한 규정과 시스템 개발 시 고려해야 할 중요한 사항들을 아래 Business Rules을 통해 정의해 보았다.

(1) 기업 관련 Rules

- 기업은 부서별로 나누어 일을 담당한다.
- 도매업체에서 소주 주문이 들어오면 기업은 주문서 순서대로 처리한다.
- 기업은 소주 발주를 위해 재고 데이터를 참고한다.

(2) 주문 관련 Rules

- 하나의 도매업체는 여러 개의 제품을 주문을 할 수 있다.
- 하나의 주문은 반드시 하나의 도매업체의 주문을 받는다.
- 소주 주문은 100짜 이상부터 할 수 있다.
- 도매업체는 소주(제품)의 가정용, 업소용 둘 다 주문이 가능하다.
- 각각의 주문은 고유한 주문번호를 가진다.

(3) 대형마트 관련 Rules

- 하나의 대형마트는 여러 명의 바이어를 가진다.
- 하나의 바이어는 반드시 하나의 대형마트에 속한다.

(4) 판매계약 관련 Rules

- 한 직원은 여러 개의 판매계약을 할 수 있다.
- 하나의 판매계약은 반드시 한 명의 직원에 의해 계약된다.
- 하나의 판매계약은 반드시 하나의 가정용 제품을 가진다.
- 하나의 가정용 제품은 여러 개의 판매계약에 속한다.
- 계약시 영업 직원은 반드시 대형마트의 바이어와 접촉해야한다.
- 각각의 판매계약은 고유한 판매계약번호를 가진다.

(5) 재고 관련 Rules

- 하나의 재고는 반드시 하나의 물류센터에 보관된다.
- 하나의 물류센터는 여러 개의 재고를 보관할 수 있다.
- 하나의 제품은 여러 개의 재고에 속한다.
- 하나의 재고는 반드시 하나의 제품을 가져야 한다.
- 재고량이 일정수준 이하일 경우 기업에게 알린다.
- 관리자는 매일 재고량과 입고량, 출하량을 입력한다.
- 주문서 순서대로 도매업체로 운송한다.

(6) 광고 계약 관련 Rules

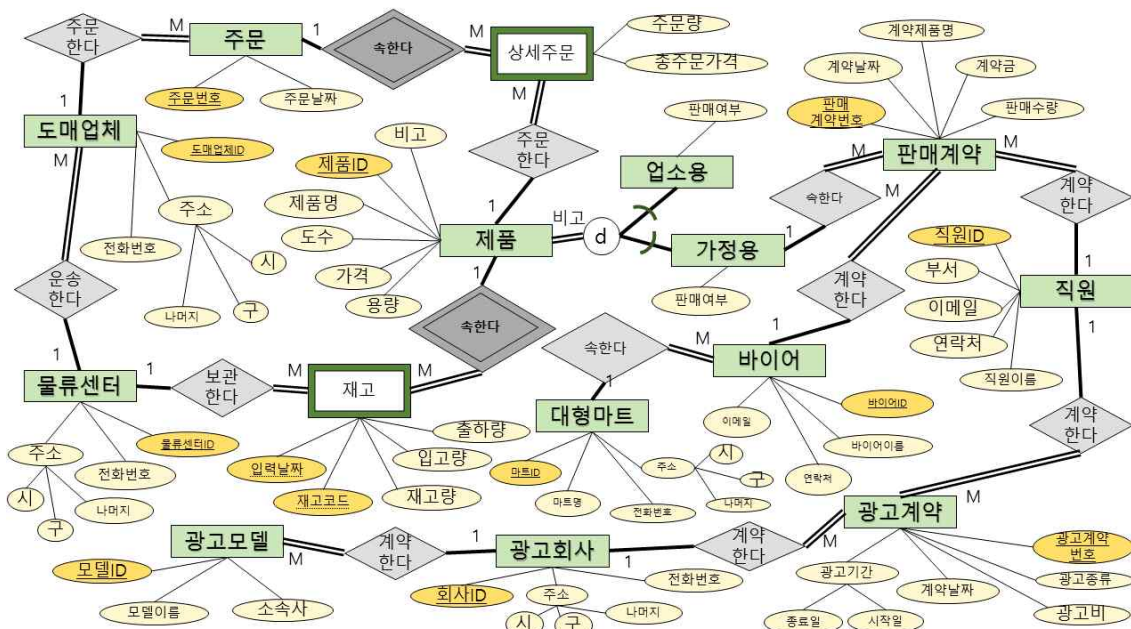
- 하나의 광고계약은 반드시 하나의 광고회사와 계약하여야 한다.
- 하나의 광고회사는 '소주'가 아닌 여러 광고계약을 할 수 있다.
- 하나의 광고회사는 여러 광고모델과 계약할 수 있다.
- 하나의 소주 광고모델은 반드시 하나의 소주 광고회사랑 계약하여야 한다.

3. DB System Design

지금까지 Project Definition과 Requirement Analysis 과정을 통해 시스템을 구축하기 위한 밑바탕이 되는 작업을 했다. 이번 DB System Design 단계에서는 앞에서 수행했던 결과들을 토대로 하여 시스템을 구축하기 위해서 설계를 할 것이다. 설계는 ERD (Entity-Relationship Diagram)를 이용한 논리적 데이터 모델 설계와 DFD(Data-Flow Diagram)를 이용한 논리적 데이터 흐름 설계를 한 후 완성된 ERD를 통해 Schemas를 작성하는 순서로 진행한다. 먼저 최상위 레벨의 ERD, DFD를 제시한 후 업무 및 기능에 따른 분류를 진행해본다.

3.1 논리적 데이터 모델 설계

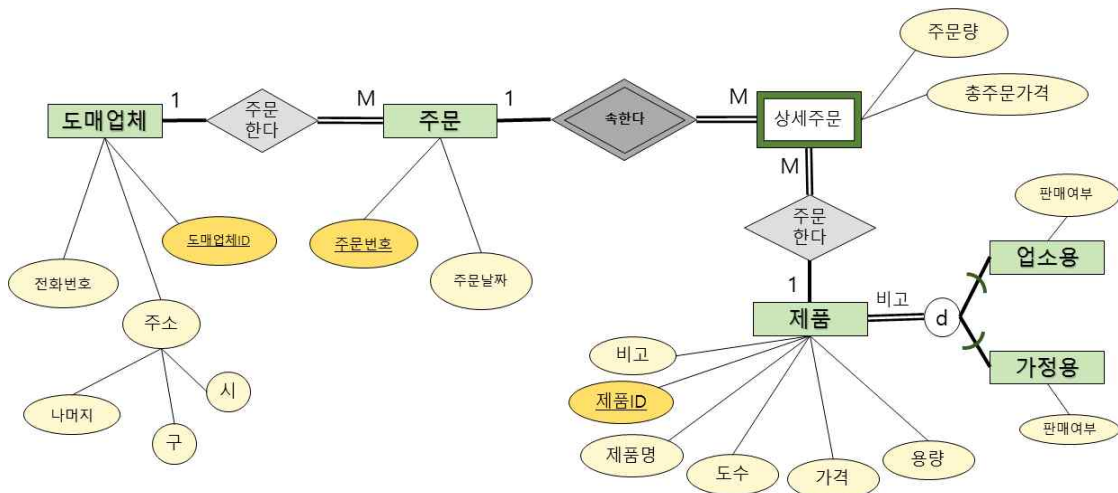
앞의 3절까지 수행한 프로젝트 정의, 시스템 분석 그리고 프로세스 모델링과 요구 사항 정리 및 Business Rules 정의를 토대로 소주 유통 관리 시스템의 전체 ERD를 그려보았다. 시스템 전체의 ERD는 다음 [그림] 과 같다.



[그림 19] 시스템 전체 ERD

앞서 정보 요구 사항에서 정의한 정보들을 기준으로 하여 Entity 들을 설정한 후 각 Entity 의 Attribute 들을 결정하였다. 그 후 각 Entity 간의 관계를 결정하고 Relationship type 에 대해서도 결정하였다. 크게 이 시스템의 주된 개체는 도매업체, 물류센터, 대형마트, 광고회사라 할 수 있다.

3.1.1 주문 정보 관리

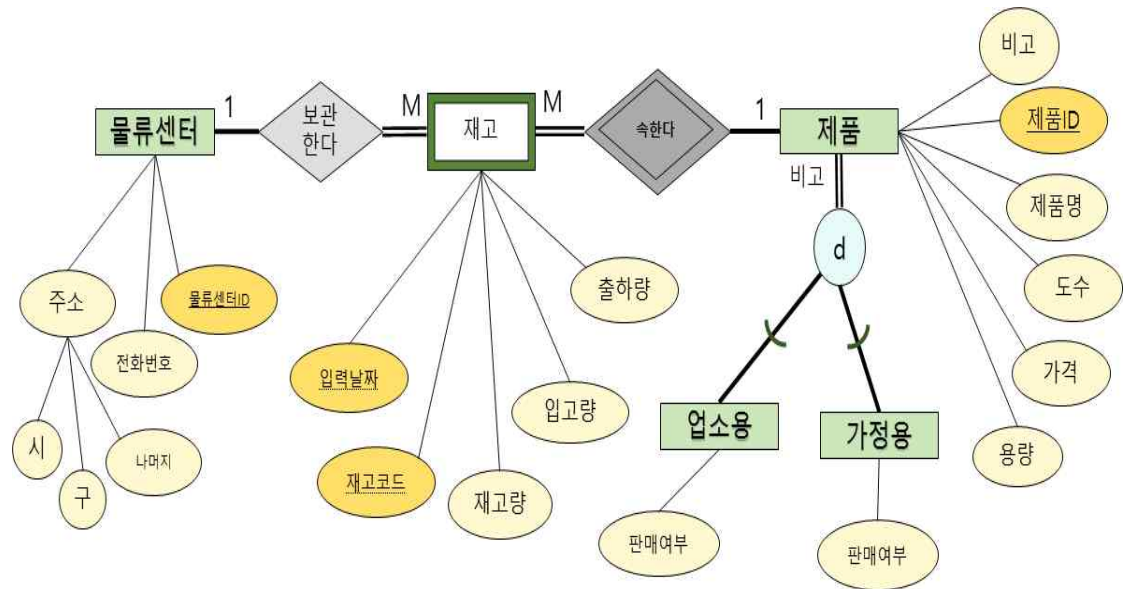


[그림 20] 주문 정보 관리 ERD

[그림 20]과 같이 도매업체는 각 도매업체ID를 키로 가지고 있으며, 한 도매업체는 주문을 여러번 할 수 있고, 하나의 주문은 하나의 도매업체에 의해 이루어지므로 도매업체와 주문은 1대다 관계를 가지고 있다. 주문은 주문번호를 키로 가지며, 주문과 제품을 이어주는 상세주

문이라는 약한 개체를 통해 주문과 제품은 상세주문과 1대다 관계를 가진다. 그리고 제품은 계승관계로서 비고라는 속성을 통해 업소용과 가정용으로 나누어지며, 제품은 반드시 업소용 또는 가정용 이어야 한다.

3.1.2 재고 정보 관리

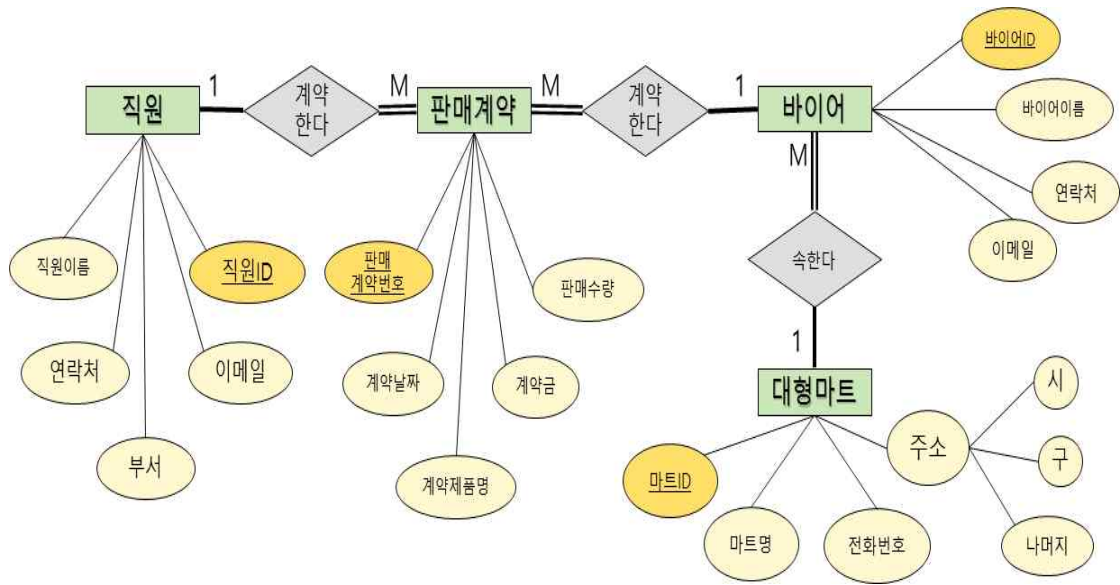


[그림 21] 재고 정보 관리 ERD

[그림 21]에서 보면 제품을 기준으로 재고라는 약한 개체를 연결해 주었다. 제품ID를 통해 재고에서 입력날짜와 재고코드를 정보를 더해주면서 하나의 재고 정보 관련 데이터가 되었다. 또한 제품에서는 비고라는 속성을 활용하여 업소용 가정용으로 나누었다. 이것은 판매 여부에 따라 판매 가능한 가정용과 판매 불가능한 업소용으로 나뉜 것이다. 물류센터는 여러 재고를 입력날짜, 재고코드를 활용하여 보관하게 된다. 재고는 반드시 하나의 물류센터에만 보관되어야 한다.

cf) 제품에서 '제조날짜'에 대해 다루지 않은 이유는 소주와 같은 주류는 유통기한이 존재하지 않는다. 예를 들어 김치와 같은 시간이 지나도 상하지 않고 숙성되는 것이기 때문이다. 그러나 술에는 '제조날짜'라는 것이 포함되어 있다. 하지만 '제조날짜' 별로 구분하게 되면 물류센터에서 재고에 대한 관리를 할 때 100개의 주문을 하였다고 가정하면 그 중 50개 제품은 3달 전에 만들어진 제품, 나머지 50개 제품은 1달 전에 만들어진 제품으로 섞여서 보내질 수 밖에 없다. 그렇기 때문에 우리는 '선입선출'의 방식을 원칙으로 하고 '제조날짜'를 데이터 정보에서 제외하기로 하였다.

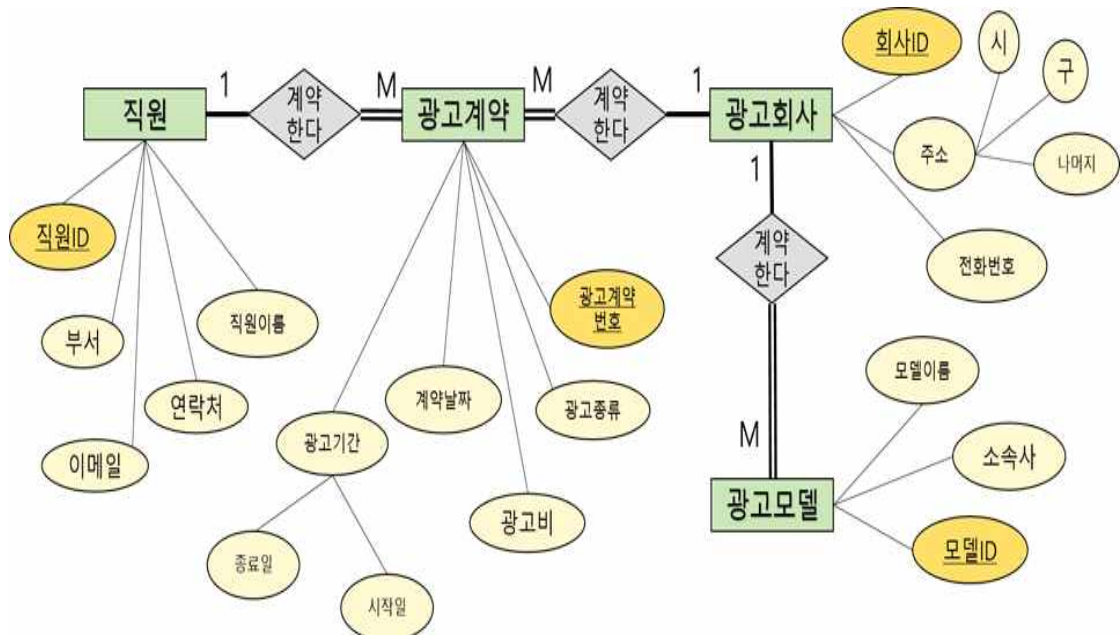
3.1.3 판매계약 정보 관리



[그림 22] 판매계약 정보 관리 ERD

[그림 22]에서 직원은 대형마트의 바이어와 판매계약을 한다. 판매계약을 할 때 계약을 하는 순서대로 판매계약번호가 주어져서, 판매계약은 한 명의 직원, 한 명의 바이어와 체결되기 때문에 직원과 바이어 모두 1대1 관계를 가진다. 하지만, 대형마트에는 여러 명의 바이어가 있을 수 있고, 바이어는 꼭 하나의 대형마트에 속해야 하기 때문에 바이어와 대형마트는 다대1 관계이다. 대형마트 또한 마트ID를 키로 가지고 있다.

3.1.4 마케팅 정보 관리



[그림 23] 마케팅 정보 관리 ERD

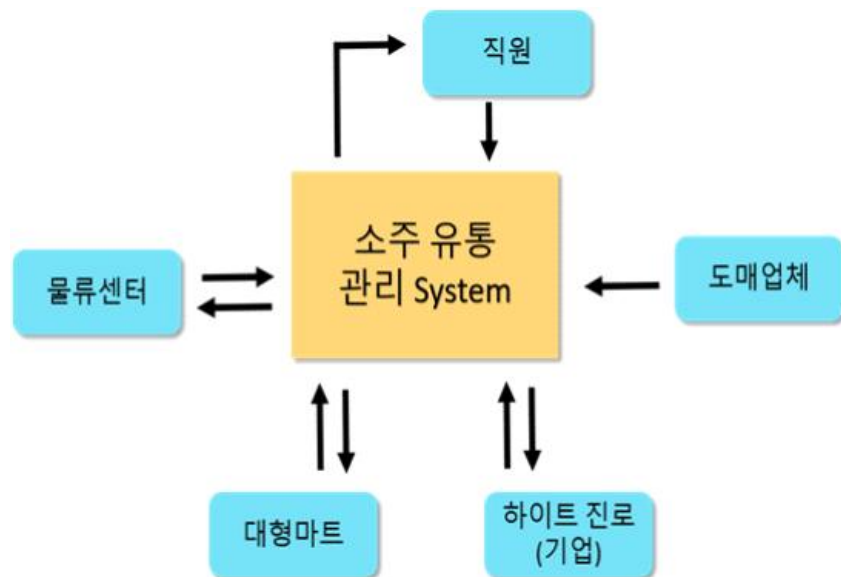
[그림 23]에서 직원은 광고회사와 광고계약을 한다. 광고계약에는 광고계약 번호에 따라 계

약이 이루어진다. 광고계약 번호를 기준으로 계약을 한 직원ID와 광고회사의 ID가 저장된다. 또한 광고회사는 광고모델과도 계약을 하는데 여기서 중요한 점은 광고회사는 여러 광고모델과 계약할 수 있기 때문에 여러 모델ID가 저장되지만 앞서 광고계약 정보 요구사항에서 말했던 예외로 광고모델은 하나의 광고회사와만 계약을 하기 때문에 하나의 회사ID에 저장된다.

3.2 논리적 데이터 흐름 설계

시스템의 데이터 흐름을 한눈에 보기 쉽게 하기 위해서, 또한 시스템의 세부 기능과 데이터 흐름을 표현하여 시스템의 동적인 측면을 설계하기 위해 DFD를 작성하였다. DFD는 ERD 작성과 병행하여 작성, 수정이 진행되었고 상세화에서는 기존에 작성된 기능요구사항도 참조하였다.

3.2.1 DFD Level 0 (Context Diagram)

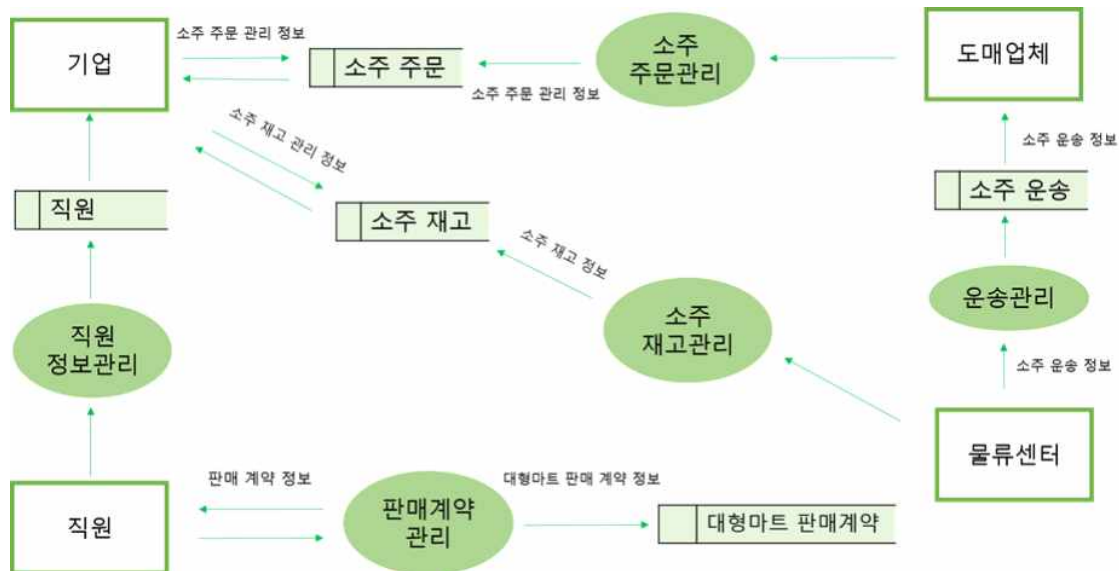


[그림 24] DFD Level 0 (Context Diagram)

ERD에 나타난 소주 유통관리 정보 시스템의 주요 객체인 도매업체, 직원, 물류센터, 대형마트를 기준으로 하여 최상위 DFD인 DFD Level 0을 Context Diagram을 통해 작성하였고 이는 위의 그림과 같다.

- 도매업체는 등록한 도매업체 정보를 기업에게 전달한다.
- 도매업체는 기업에게 소주 주문을 한다.
- 기업은 물류센터에 도매업체 주문 정보를 제공한다.
- 물류센터는 기업에게 재고 관리 정보에 대해 전달한다.
- 대형마트의 바이어와 기업의 판매부서 직원은 판매계약을 하고 그에 관한 정보는 서로 열람 가능하다.
- 광고회사와 기업의 마케팅부서 직원은 광고계약을 하고 그에 관한 정보는 서로 열람 가능하다.

3.2.2 DFD Level 1 - 소주 유통 관리 전체

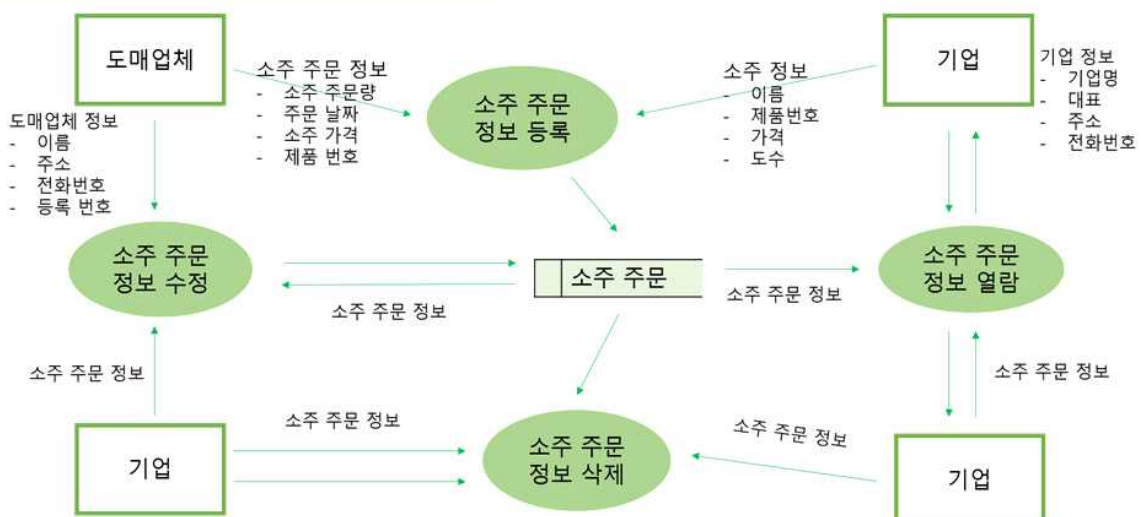


[그림 25] DFD Level 1 - 소주 유통관리

ERD 상에서의 소주 유통관리 부분을 DFD level 1로 나타낸 것이다. 물류센터의 소주 재고 관리가 소주의 재고 데이터 정보로 입력되고 그것은 기업과 재고 관리 정보로 주고받는다. 또한 물류센터의 운송관리에 대한 정보가 입력되고 그것은 도매업체로부터 전달한다. 도매업체는 소주 주문관리에 대해 소주 주문 정보를 기업과 주고받는다. 직원은 대형마트의 입력된 판매계약 데이터를 직원 정보 데이터로 입력하여 기업에게 전달한다. 이러한 구체적인 사항들 주문, 재고, 판매 관리에 대해 Level 2에서 상세히 나타내었다.

3.2.3 DFD Level 2

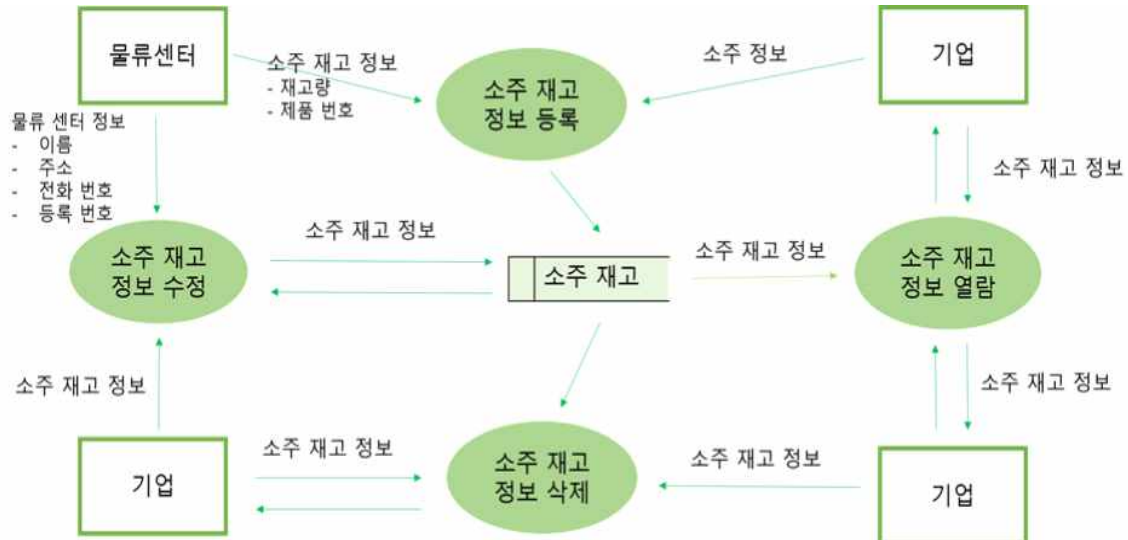
(1) DFD Level 2 - 소주 주문 관리



[그림 26] DFD Level 2 - 소주 주문 관리

소주 유통관리의 일부분인 소주 주문관리에 대해 나타낸 것이다. 기업은 도매업체가 등록한 소주 주문 정보에 대해서 열람, 삭제, 수정 할 수 있으며 도매업체는 소주 주문 정보를 수정 할 수 있다. 그리고 기업은 소주 주문을 위한 소주에 대한 정보를 수정할 수 있다.

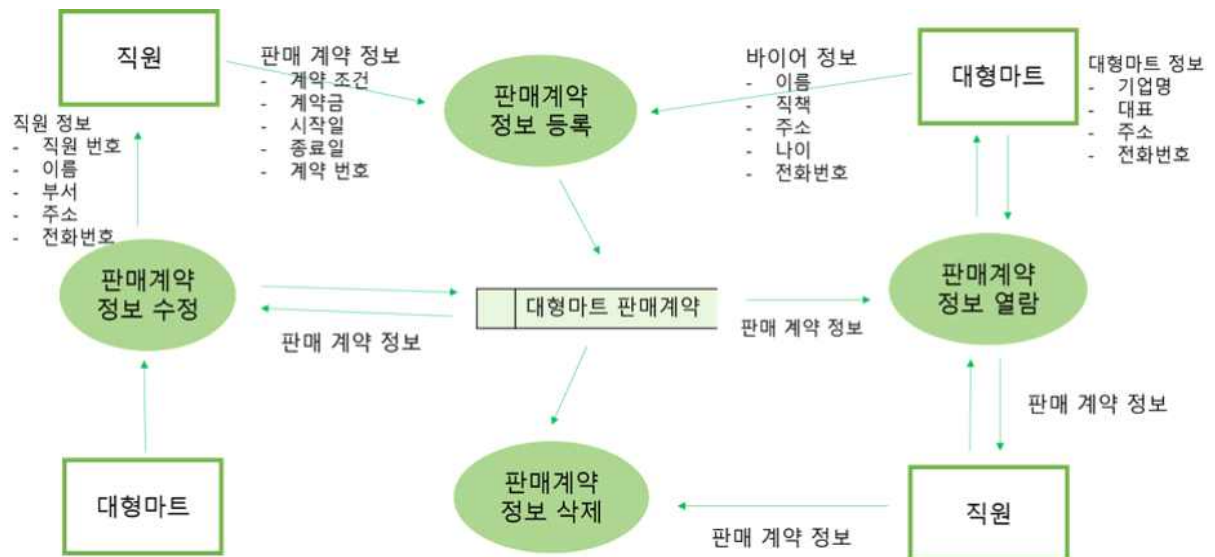
(2) DFD Level 2 - 소주 재고 관리



[그림 27] DFD Level 2 - 소주 재고 관리

소주 유통관리의 일부분인 소주 재고 관리에 대해 나타낸 것이다. 기업은 물류센터가 등록한 소주 재고 정보에 대해 열람, 삭제, 수정 할 수 있으며 물류센터는 소주 재고 정보를 수정할 수 있다. 그리고 기업은 소주 재고를 위한 소주에 대한 정보를 수정할 수 있다.

(3) DFD Level 2 - 판매 계약 관리



[그림 28] DFD Level 2 - 판매 계약 관리

소주 유통관리의 일부분인 판매 계약 관리에 대해 나타난 것이다. 대형마트는 바이어를 통해 판매계약 정보를 등록할 수 있고 직원 또한 판매계약에 대한 정보를 등록할 수 있다. 직원과 대형마트는 각각 판매계약에 대해 정보를 열람할 수 있으며 삭제도 가능하다. 대형마트는 판매계약을 한 직원에게 수정을 요구할 수 있다.

3.3 Data Dictionary

ERD상에 나타난 데이터에 대해 이해하기 쉽도록 사전형식으로 정리한다.
Primary Key와 Foreign Key는 table 정의에서 한다.

◇도매업체

- ◆ 도매업체ID int - 도매업체명을 대체하여 ID 속성을 만들었다.
- ◆ 도매업체전화번호 char(100)
- ◆ 도매업체주소(시,구,나머지) : 복합 속성
 - 도매업체주소_시 char(100)
 - 도매업체주소_구 char(100)
 - 도매업체주소_나머지 char(100)

◇물류센터

- ◆ 물류센터ID int - 물류센터명을 대체하여 ID 속성을 만들었다.
- ◆ 물류센터전화번호 char(100)
- ◆ 물류센터주소(시,구,나머지) : 복합 속성
 - 물류센터주소_시 char(100)
 - 물류센터주소_구 char(100)
 - 물류센터주소_나머지 char(100)

◇주문

- ◆ 주문번호 int
- ◆ 주문날짜 date

◇상세주문

- ◆ 주문량 int
- ◆ 총주문가격 money - 주문량에 따른 총 주문 가격 금액

◇제품 - 업소용과 가정용으로 나누어 진다.

- ◆ 제품ID int
- ◆ 제품명 char(100)
- ◆ 도수 char(100)
- ◆ 가격 money
- ◆ 용량 char(100)
- ◆ 비고 char(100) - 가정용, 업소용을 나누는 속성

◇업소용 - 제품의 속성들을 모두 상속

- ◆ 판매여부 char(100) - 판매 불가를 기록하기 위한 속성

◇가정용 - 제품의 속성들을 모두 상속

- ◆ 판매여부 char(100) - 판매 가능을 기록하기 위한 속성

◇재고 - 약한 개체 클래스

- ◆ 입력날짜 datetime - 부분키, 재고를 입력한 날짜
- ◆ 재고코드 char(100) - 물류센터에 따른 재고코드
- ◆ 재고량 int - 당일 총 재고량을 입력
- ◆ 입고량 int - 당일 입고된 수량을 입력
- ◆ 출하량 int - 당일 출하된 수량을 입력

◇직원 - 기업의 직원

- ◆ 직원ID int - 기업의 직원 고유 ID
- ◆ 직원이름 char(100)
- ◆ 부서 char(100) - 직원의 부서명
- ◆ 직원이메일 char(100) - 직원의 이메일 주소
- ◆ 직원연락처 char(100) - 직원의 전화번호

◇바이어 - 기업의 직원과 계약하는 대형마트의 바이어

- ◆ 바이어ID int - 마트의 바이어 고유 ID
- ◆ 바이어이름 char(100)

- ◆ 바이어이메일 char(100)
- ◆ 바이어연락처 char(100)

◇대형마트

- ◆ 마트ID int
- ◆ 마트명 char(100)
- ◆ 마트전화번호 char(100)
- ◆ 마트주소(시,구,나머지) : 복합 속성
 - 마트주소_시 char(100)
 - 마트주소_구 char(100)
 - 마트주소_나머지 char(100)

◇판매계약

- ◆ 판매계약번호 int - 계약서에 기록되는 계약번호
- ◆ 판매계약날짜 char(100)
- ◆ 계약제품명 char(100) - 계약하는 제품의 이름
- ◆ 계약금 money
- ◆ 판매수량 int

◇광고계약

- ◆ 광고계약번호 int
- ◆ 광고종류 char(100) - tv, 인터넷 등 광고의 종류를 기록하는 속성
- ◆ 광고비 money
- ◆ 광고계약날짜 char(100) - 광고계약을 체결한 날짜
- ◆ 광고기간(시작일, 종료일) : 복합속성
 - 광고시작일 char(100) - 광고 게시 시작 날짜
 - 광고종료일 char(100) - 광고 게시 종료 날짜

◇광고회사

- ◆ 회사ID int
- ◆ 회사전화번호 char(100)
- ◆ 회사주소(시,구,나머지) : 복합 속성
 - 회사주소_시 char(100)
 - 회사주소_구 char(100)
 - 회사주소_나머지 char(100)

◇광고모델

- ◆ 모델ID int
- ◆ 소속사 char(100)
- ◆ 모델이름 char(100)

3.4 Schemas 정의

완성된 ERD를 바탕으로 실제 데이터베이스를 구축하기 위해 먼저 데이터베이스 스키마를 제시하여 ERD 를 데이터베이스 구축에 적합하게 정의했다. 스키마 정의는 8단계 과정에 맞추어 차례대로 제시하였다.

(1) 1단계(Strong Entity)

스키마	속성
물류센터	<u>물류센터ID</u> string, 전화번호 number
도매업체	<u>도매업체ID</u> string, 전화번호 number
주문	<u>주문번호</u> string, 주문날짜 date
제품	<u>제품ID</u> string, 제품명 string, 도수 string, 가격 number, 용량 string, 비고 string
직원	<u>직원ID</u> string, 직원이름 string, 부서 string, 직원이메일 string, 직원연락처 string
대형마트	<u>마트ID</u> string, 마트명 string, 마트전화번호 string
바이어	<u>바이어ID</u> string, 바이어이름 string, 바이어이메일 string, 바이어연락처 string
판매계약	<u>판매계약번호</u> string, 판매계약날짜 date, 계약제품명 string, 계약금 number, 판매수량 number
광고회사	<u>회사ID</u> string, 회사전화번호 string
광고계약	<u>광고계약번호</u> string, 광고종류 string, 광고비 number, 광고계약날짜 date
광고모델	<u>모델ID</u> string, 소속사 string, 모델이름 string

[표 1] Schemas-1단계

1단계에서는 Strong Entity와 Strong Entity의 Simple Attribute들을 정의한다. 강한 개체 클래스의 이름을 그대로 스키마 이름으로 가져왔다. 그리고 단순 속성들의 이름 또한 그대로 가져와 정의하였다. 각 ID 또는 번호를 Primary Key로 설정 하였다. 1단계에서 정의된 강한 개체 스키마는 총 11개 이다.

(2) 2단계 (Composite attribute)

2단계에서는 Strong Entity의 Composite attribute들을 그들의 구성요소들로 분해한다. 강한 개체의 복합 속성들로는 물류센터, 도매업체, 대형마트, 광고회사, 광고계약이 있다. 5가지의 개체 중 물류센터, 도매업체, 대형마트, 광고회사에서는 주소가 복합속성으로 표현된다. 예로 도매업체의 ‘도매업체주소’ Composite attribute는 도매업체주소_시string, 도매업체주소_구string, 도매업체주소_나머지string으로 분해되어야 한다. 마찬가지로 나머지 스키마들도 [표2]와 같이 다음의 구성요소로 분해되어 진다. 그리고 광고계약의 경우 광고기간이 광고시작

일과 광고종료일로 나누어 구성된다.

스키마	속성
물류센터	<u>물류센터ID</u> string, 전화번호 number, <u>물류센터주소_시</u> string, <u>물류센터주소_구</u> string, <u>물류센터주소_나머지</u> string
도매업체	<u>도매업체ID</u> string, 전화번호 number, <u>도매업체주소_시</u> string, <u>도매업체주소_구</u> string, <u>도매업체주소_나머지</u> string
대형마트	<u>마트ID</u> string, 마트명 string, 마트전화번호 string, <u>마트주소_시</u> string, <u>마트주소_구</u> string, <u>마트주소_나머지</u> string
광고회사	<u>회사ID</u> string, 회사전화번호 string, <u>회사주소_시</u> string, <u>회사주소_구</u> string, <u>회사주소_나머지</u> string
광고계약	<u>광고계약번호</u> string, 광고종류 string, 광고비 number, 광고계약날짜 date, <u>광고시작일</u> date, <u>광고종료일</u> date

[표 2] Schemas-2단계

(3) 3단계 (1 대 M Cardinality)

3단계에서는 1-M의 Cardinality 관계 타입을 위한 관계 스키마를 생성한다. 여기서 1의 입장에 있는 Entity 의 Key를 M의 관계에 있는 Entity로 가져와 Foreign Key로 사용한다. 도매업체에서 물류센터의 물류센터ID를, 주문에서 도매업체의 도매업체ID를 가져오는 것이 이에 해당된다. 3단계에서 정의된 스키마는 도매업체, 주문, 바이어, 판매계약, 광고계약, 광고모델 총 6개 이다. 그 중 판매계약과 광고계약은 이어진 두 개체에서 모두 키를 가져와 외래키가 2개가 된다.

스키마	속성
도매업체	<u>도매업체ID</u> string, 전화번호 number, <u>도매업체주소_시</u> string, <u>도매업체주소_구</u> string, <u>도매업체주소_나머지</u> string, <u>물류센터ID</u> string references <u>물류센터</u>
주문	<u>주문번호</u> string, 주문날짜 date, <u>도매업체ID</u> string references <u>도매업체</u>
바이어	<u>바이어ID</u> string, 바이어이름 string, 바이어이메일 string, 바이어연락처 string, <u>마트ID</u> string references <u>대형마트</u>
판매계약	<u>판매계약번호</u> string, 판매계약날짜 date, 계약제품명 string, 계약금 number, 판매수량 number, <u>바이어ID</u> string references <u>바이어</u> , <u>직원ID</u> string references <u>직원</u>
광고계약	<u>광고계약번호</u> string, 광고종류 string, 광고비 number, 광고계약날짜 date, <u>광고시작일</u> date, <u>광고종료일</u> date, <u>회사ID</u> string references <u>광고회사</u> , <u>직원ID</u> string references <u>직원</u>
광고모델	<u>모델ID</u> string, 소속사 string, 모델이름 string, <u>회사ID</u> string references <u>광고회사</u>

[표 3] Schemas-3단계

(4) 4단계 (1 대 1 Cardinality)

소주 유통 관리 시스템의 ERD에서는 1 대 1 카디널리티 관계는 나타나지 않는다.

(5) 5단계(M 대 M Cardinality)

소주 유통 관리 시스템의 ERD에서는 M 대 M 카디널리티 관계는 나타나지 않는다.

(6) 6단계 (Weak Entity)

6단계에서는 Weak Entity들을 위해 관계 스키마들을 생성한다. Weak Entity는 key값이 없으므로 이는 식별 관계를 갖는 소유자 Entity에서 가져온다. 소유자 Entity 외에 또 다른 관계를 가지고 있다면 그 관계를 표현하기 위해 관계를 맺고 있는 Entity에서 외래키를 가져온다. [표6]에서는 재고 Week Entity가 소유자 Entity인 '제품'에서 제품ID를 Primary key로 가져오고, 제품ID 만으로는 구분이 안 되어 부분키인 입력날짜, 재고코드와 함께 Primary key로 사용한다. 그리고 물류센터 Entity에서 외래키를 가져왔다. 상세주문의 경우 식별 관계를 갖는 소유자 Entity인 주문에서 주문번호를 가져와 Primary키로 사용하고, 제품에서 제품ID를 가져와 외래키로 사용한다.

스키마	속성
재고	<u>입력날짜</u> date, <u>제품ID</u> string references 제품, <u>재고코드</u> string, 재고량 number, 입고량 number, 출하량 number, 물류센터ID string references 물류센터
상세주문	<u>주문번호</u> string references 주문, 주문량 number, 총주문가격 number, 제품ID string references 제품

[표 4] Schemas-6단계

(7) 7단계(다중값 속성)

소주 데이터베이스에서는 다중값 속성으로 사용할 만한 것이 특별히 존재하지 않는다.

(8) 8단계

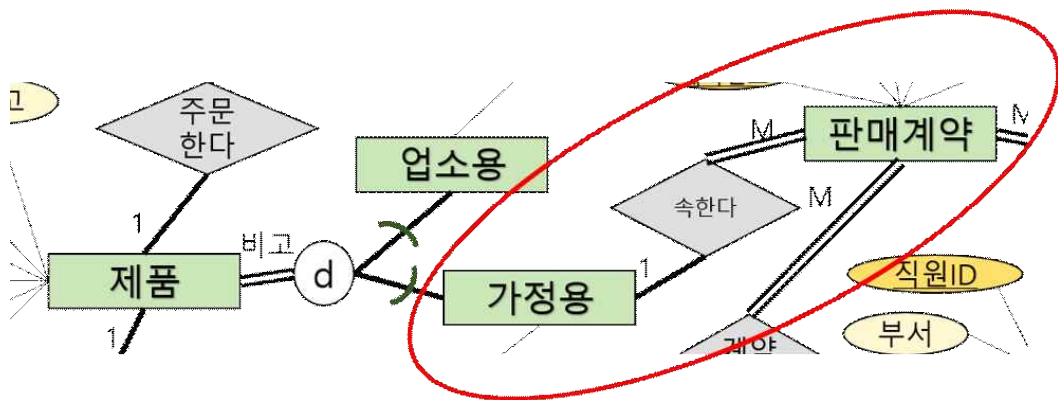
8단계에서는 Inheritance 관계 타입과 Subclass를 표현한다. Subclass들은 Superclass의 key값을 계승 받는다. 여기서는 Subclass인 업소용과 가정용이 Superclass인 제품에서 제품ID를 key값으로 계승 받았다. 그리고 업소용과 가정용을 구분하기 위해 제품구분코드를 defining attribute로 사용하였다.

스키마	속성
업소용	<u>제품ID</u> string references 제품, 판매여부 string, 제품구분코드 string (defining attribute)
가정용	<u>제품ID</u> string references 제품, 판매여부 string, 제품구분코드 string (defining attribute)

[표 5] Schemas-8단계

◇ 8단계+ 3단계표현

판매계약은 대형마트와 하는 계약이므로 업소용이 아니라 가정용과만 계약된다. 그렇기 때문에 제품의 Subclass인 가정용과 1 대 다 관계로 이어진다. 그래서 8단계 + 3단계로 표현하였다. 따라서 판매계약번호가 Primary key가 되며, 가정용에서 제품ID를 외래키로 가져온다.



[그림 29] Schemas-(8+3) 단계

스키마	속성
판매계약	판매계약번호 string, 판매계약날짜 date, 계약제품명 string, 계약금 number, 판매수량 number, 바이어ID string references 바이어, 직원ID string references 직원, 제품ID string references 가정용

[표 6] Schemas-(8+3) 단계

(9) 최종 스키마

최종적으로 만들어진 15개의 스키마이다.

스키마	속성
물류센터	물류센터ID string, 전화번호 number, 물류센터주소_시 string, 물류센터주소_구 string, 물류센터주소_나머지 string
도매업체	도매업체ID string, 전화번호 number, 도매업체주소_시 string, 도매업체주소_구 string, 도매업체주소_나머지 string, 물류센터ID string references 물류센터
주문	주문번호 string, 주문날짜 date, 도매업체ID string references 도매업체
제품	제품ID string, 제품명 string, 도수 string, 가격 number, 용량 string, 비고 string
직원	직원ID string, 직원이름 string, 부서 string, 직원이메일 string, 직원연락처 string
대형마트	마트ID string, 마트명 string, 마트전화번호 string, 마트주소_시 string, 마트주소_구 string, 마트주소_나머지 string
바이어	바이어ID string, 바이어이름 string, 바이어이메일 string, 바이어연락처 string, 마트ID string references 대형마트
판매계약	판매계약번호 string, 판매계약날짜 date, 계약제품명 string, 계약금 number, 판매수량 number, 바이어ID string references 바이어, 직원ID string references 직원, 제품ID string references 가정용

스키마	속성
광고회사	회사ID string, 회사전화번호 string, 회사주소_시 string, 회사주소_구 string, 회사주소_나머지 string
광고계약	광고계약번호 string, 광고종류 string, 광고비 number, 광고계약날짜 date, 광고시작일 date, 광고종료일 date, 회사ID string references 광고회사, 직원ID string references 직원
광고모델	모델ID string, 소속사 string, 모델이름 string, 회사ID string references 광고회사
재고	입력날짜 date, 제품ID string references 제품, 재고코드 string, 재고량 number, 입고량 number, 출하량 number, 물류센터ID string references 물류센터
상세주문	주문번호 string references 주문, 주문량 number, 총주문가격 number, 제품ID string references 제품
업소용	제품ID string references 제품, 판매여부 string, 제품구분코드 string (defining attribute)
가정용	제품ID string references 제품, 판매여부 string, 제품구분코드 string (defining attribute)

[표 7] Schemas-9단계

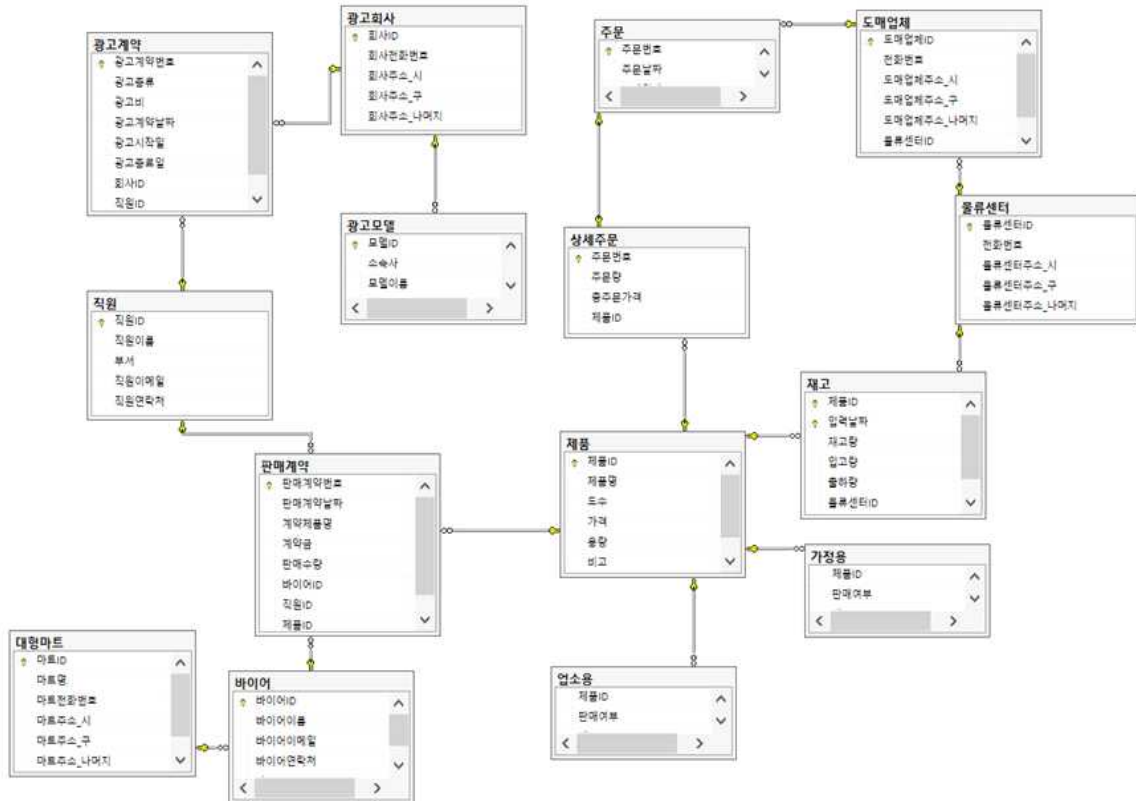
3.5 Table 정의

위의 최종 스키마 실현을 위해 DDL문을 사용하여 SQL에서 테이블을 만들었다. 그 결과 총 15개의 테이블이 생성되었다. 그 결과는 다음과 같다. 그리고 INSERT INTO를 통해 데이터를 입력하였다.



[그림 30] DDL문 예시

다음 그림은 만들어진 Table 의 Diagram이다.



[그림 31] Diagram

15개의 테이블 중 중요한 테이블과 특이사항이 있는 테이블 위주로 설명하겠다.

(1) 주문 및 상세주문

	주문번호	주문일자	도매업체ID
1	400001	2019-03-01	201
2	400002	2019-03-04	202
3	400003	2019-03-05	201
4	400004	2019-03-09	204
5	400005	2019-03-12	203
6	400006	2019-03-15	204
7	400007	2019-03-16	202
8	400008	2019-03-19	201
9	400009	2019-03-24	203

[그림 32] 주문 Table

	주문번호	주문량	총주문가격	제품ID
1	400001	200	8000000	101
2	400002	200	6000000	102
3	400003	200	6400000	104
4	400004	200	8000000	101
5	400005	200	8000000	101
6	400006	100	3600000	105
7	400007	200	5200000	106
8	400008	300	7800000	106

[그림 33] 상세 주문 Table

주문의 경우 Primary 키인 주문번호와 주문날짜, 주문을 한 도매업체ID로 이루어져 있다. 주문의 상세내용은 상세주문이라는 Table에서 따로 나타난다. 상세주문에서는 주문번호와 주문량, 총 주문가격과 주문한 제품의 ID가 나타난다. 이 두 테이블을 통해 어떤 도매업체가 어떤 제품을 몇 개 주문했는지 볼 수 있다.

(2) 재고

	입력날짜	제품ID	재고코드	재고량	입고량	출하량	물류센터ID
1	2019-03-01	101	C3	600	100	200	303
2	2019-03-04	102	C1	800	300	200	301
3	2019-03-05	104	C3	500	100	200	303
4	2019-03-09	101	C2	900	100	200	302
5	2019-03-12	101	C4	1000	0	200	304
6	2019-03-15	105	C2	800	100	100	302
7	2019-03-16	106	C1	700	0	200	301
8	2019-03-19	106	C3	1100	200	300	303
9	2019-03-24	102	C2	400	100	500	302

[그림 34] 재고 Table

재고의 경우 약한 개체이기 때문에 제품에서 제품ID를 가져와 프라이머리키로 사용하며, 제품ID 만으로는 유니크한 식별이 불가능하기 때문에 입력날짜와 재고코드라는 2개의 부분키를 사용하여 각각의 재고량 등을 식별하였다. 그리고 소주의 경우 유통기한이 거의 없다시피 하기 때문에 따로 제조일자 라던가 유통기한은 재고의 속성으로 넣지 않았다. 그리고 원래라면 날짜마다, 물류센터마다의 재고 데이터가 모두 있어야 하지만 현실적으로 프로젝트의 경우 수백개나 되는 데이터를 입력하기 어렵기 때문에 주문 테이블의 주문 날짜들과 판매계약 테이블의 판매계약날짜들만 재고 데이터로 만들었다.

(3) 판매계약

	판매계약번호	판매계약날짜	계약제품명	계약금	판매수량	바이어ID	직원ID	제품ID
1	911001	2019-03-24	참이슬 fresh	5000000.00	500	822001	811002	102
2	911002	2019-03-26	일품진로	2000000.00	100	822002	811005	108
3	911003	2019-03-31	참이슬 오리지널	4000000.00	400	822001	811005	104
4	911004	2019-04-01	참이슬 fresh	2000000.00	200	822003	811002	102
5	911005	2019-04-05	진로	3000000.00	300	822002	811002	106
6	911006	2019-04-08	참이슬 fresh	3000000.00	300	822004	811005	102
7	911007	2019-04-11	진로	1000000.00	100	822001	811002	106
8	911008	2019-04-14	참이슬 오리지널	2000000.00	200	822003	811005	104

[그림 35] 판매계약 Table

판매계약의 경우 Key인 판매계약번호와 판매계약날짜 그리고 제품 Table의 제품명과 다른 계약제품명이라는 속성이 따로 있으며, 계약금과 판매 수량이 나타나 있다. 그리고 이 판매계약을 체결한 바이어ID와 직원ID, 그리고 계약한 제품ID까지 Foreign Key 로 포함되어 있는 것을 볼 수 있다.

(4) 광고계약

	광고계약번호	광고종류	광고비	광고계약날짜	광고시작일	광고종료일	회사ID	직원ID
1	922001	TV	10000000.00	2019-03-15	2019-04-01	2019-04-30	7001	811004
2	922002	배너	3000000.00	2019-03-21	2019-04-01	2019-04-10	7003	811006
3	922003	SNS	1000000.00	2019-03-27	2019-04-01	2019-04-05	7004	811004
4	922004	라디오	5000000.00	2019-04-03	2019-04-10	2019-04-15	7002	811006
5	922005	SNS	1000000.00	2019-04-05	2019-04-05	2019-04-10	7004	811004
6	922006	유튜브	8000000.00	2019-04-07	2019-04-15	2019-04-30	7003	811004
7	922007	배너	3000000.00	2019-04-10	2019-04-11	2019-04-20	7003	811006
8	922008	SNS	2000000.00	2019-04-12	2019-04-15	2019-04-25	7004	811004
9	922009	라디오	5000000.00	2019-04-12	2019-04-15	2019-04-20	7001	811006

[그림 36] 광고계약 Table

광고계약의 경우 Key인 광고계약번호와 광고종류, 광고비, 광고계약날짜와 이 계약한 광고를 시작하는 날짜와 종료하는 날짜가 나와 있다. 또, 이 계약을 체결하는 광고회사의 ID와 직원 ID가 Foreign Key로 포함되어 있는 것을 볼 수 있다.

3.6 Procedure 설계

지금까지 테이블을 통해 필요한 데이터들을 입력하고 테이블 간의 관계를 설정하였다. 그리고 Procedure를 설계하여 고객요구사항들을 만족하는 기능들을 구현하였다. 16개의 기능 프로시저와 단순 조회 프로시저 1개, 입력 및 수정, 삭제가 가능한 프로시저 1개, 총 18개의 프로시저가 생성되었다. 보고서 페이지 제한으로 인해 중요하거나 복잡한 프로시저들만 설명하겠다.

(1) 입력, 수정, 삭제 및 조회가 가능한 프로시저

```

uu
ALTER procedure [dbo].[데이터관리]
(
    @type as char(100)='',
    @물류센터 ID as int,
    @전화번호 as char(100),
    @물류센터주소_시 as char(100),
    @물류센터주소_구 as char(100),
    @물류센터주소_나머지 as char(100)
)
as
begin
    begin
        if @type = '입력'
        begin
            INSERT INTO 물류센터 (물류센터 ID, 전화번호, 물류센터주소_시, 물류센터주소_구, 물류센터주소_나머지)
            VALUES (@물류센터 ID, @전화번호, @물류센터주소_시, @물류센터주소_구, @물류센터주소_나머지)
        end
        if @type = '수정'
        begin
            UPDATE 물류센터 SET 전화번호=@전화번호, 물류센터주소_시=@물류센터주소_시, 물류센터주소_구=@물류센터주소_구, 물류센터주소_나머지=@물류센터주소_나머지
            WHERE 물류센터 ID=@물류센터 ID
        end
        else if @type = '삭제'
        begin
            DELETE FROM 물류센터 WHERE 물류센터 ID=@물류센터 ID
        end
        select * from 물류센터
    end
end;

```

Parameter	Data Type	Output Pa...	Pass Null ...	Value
@type	char(100)	No	<input type="checkbox"/>	입력
@물류센터 ...	int	No	<input type="checkbox"/>	305
@전화번호	char(100)	No	<input type="checkbox"/>	02-1256-...
@물류센터 ...	char(100)	No	<input type="checkbox"/>	서울
@물류센터 ...	char(100)	No	<input type="checkbox"/>	마포구
@물류센터 ...	char(100)	No	<input type="checkbox"/>	999

	물류센터ID	전화번호	물류센터주소_시	물류센터주소_구	물류센터주소_나머지
1	301	031-2114-2365	경기도	과천시	165-3
2	302	031-4968-9654	경기도	성남시	34
3	303	031-6585-5932	경기도	하남시	64-2
4	304	031-7569-2651	경기도	고양시	121-5
5	305	02-1256-5984	서울	마포구	999

[그림 37] 입력, 수정, 삭제 및 조회 가능 프로시저

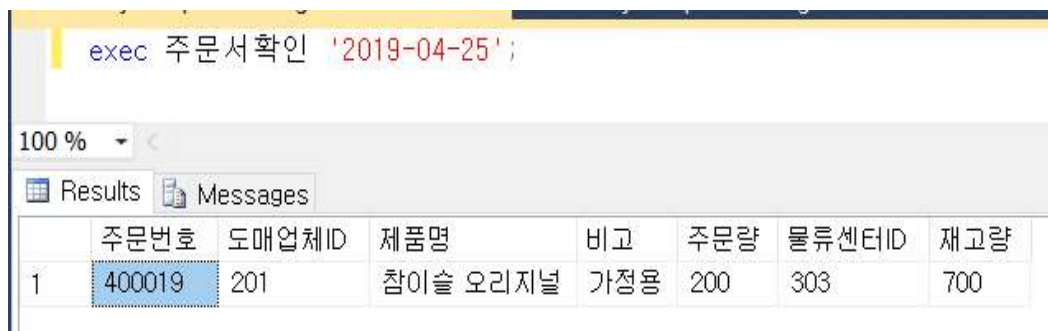
하나의 프로시저 실행으로 입력 또는 수정, 삭제 후 바로 결과를 확인 가능한 프로시저를 만들었다. If 문과 type 이라는 Parameter를 통해 '입력' 또는 '수정', '삭제' 라는 단어를 입력하면 그에 해당하는 DML문이 실행되도록 설계하였다. 예로 type에 '입력'이라는 단어를 넣은 후 추가하고 싶은 물류센터ID, 전화번호, 주소를 입력하고 확인을 누르면 처음 테이블에서 추가로 데이터가 생성된 것을 볼 수 있다. 이 하나의 프로시저를 통해 입력, 수정, 삭제가 모두 가능하다.

(2) 주문

(2-1) 주문서 확인

```
ALTER procedure [dbo].[주문서확인] (@주문날짜 AS char(100))
as
begin
    select O.주문번호, D.도매업체ID, J.제품명, J.비고, S.주문량, Z.물류센터ID, Z.재고량
    from (((도매업체 as D INNER JOIN 주문 as O ON D.도매업체ID=O.도매업체ID)
        INNER JOIN 상세주문 as S ON O.주문번호=S.주문번호)
        INNER JOIN 재고 as Z ON S.제품ID=Z.제품ID)
        INNER JOIN 제품 as J ON S.제품ID=J.제품ID

    where O.주문날짜=Z.입력날짜
    and @주문날짜=주문날짜
end;
```



exec 주문서확인 '2019-04-25';

	주문번호	도매업체ID	제품명	비고	주문량	물류센터ID	재고량
1	400019	201	참이슬 오리지널	가정용	200	303	700

[그림 38] 주문서 확인 프로시저

이 기능은 우리가 처음 보고자 했던 기능이다. 주문날짜를 Parameter로 설정하여 주문날짜를 입력하면 주문번호와 주문한 도매업체의 ID, 제품명, 비고(업소용, 가정용 구분), 주문량, 물류센터ID, 물류센터의 해당 제품의 재고량을 볼 수 있다. 이 프로시저에는 도매업체, 주문, 상세주문, 재고, 제품이라는 총 5개의 테이블이 INNER JOIN을 통해 연결되어 있다. 그리고 주문의 주문날짜와 재고의 입력날짜가 같을 경우의 데이터를 보여준다. 이 기능을 통해 기업은 쉽게 주문서의 필요한 내용들을 볼 수 있을 것이다.

(2-2) 소주별 주문통계

```

ALTER procedure [dbo].[소주별주문통계]
as
begin
    select J.제품명, J.비고, SUM(S.주문량) as 주문량합계, AVG(S.주문량) as 평균주문량, COUNT(S.주문량) as 주문횟수
    from 제품 as J INNER JOIN 상세주문 as S ON J.제품ID=S.제품ID
    group by 제품명, 비고
    order by 제품명
end;
exec 소주별주문통계;

```

	제품명	비고	주문량합계	평균주문량	주문횟수
1	진로	가정용	1100	220	5
2	진로	업소용	400	133	3
3	참이슬 오리지널	가정용	1000	200	5
4	참이슬 오리지널	업소용	400	200	2
5	참이슬 fresh	가정용	900	180	5
6	참이슬 fresh	업소용	1000	200	5

[그림 39] 소주별 주문통계 프로시저

기업은 소주별 주문통계를 통해 어떤 소주가 더 잘 팔리고 얼마나 팔리는 지에 대해 볼 수 있다. 그리고 이 통계를 이 후 소주 발주계획에 참고 할 수 있다. 먼저 group by를 통해 제품명과 비고로 나누어 결과를 볼 수 있게 만들었으며, SUM을 통해 제품별 주문량 합계와 AVG를 통해 제품별 평균 주문량을 볼 수 있고, COUNT를 통해 제품마다 몇 번 주문되었는지 주문횟수를 알 수 있다.

(3) 제품 - 제품 특성 확인

```

ALTER procedure [dbo].[제품특성확인] (@용도 as char(100))
as
begin
    if @용도='업소용'
    begin
        select J.*, BU.판매여부
        from 제품 as J RIGHT OUTER JOIN 업소용 as BU ON J.제품ID=BU.제품ID
    end
    if @용도='가정용'
    begin
        select J.*, FU.판매여부
        from 제품 as J RIGHT OUTER JOIN 가정용 as FU ON J.제품ID=FU.제품ID
    end
end;
exec 제품특성확인 업소용;

```

	제품ID	제품명	도수	가격	용량	비고	판매여부
1	101	참이슬 fresh	17도	2000,00	360ml	업소용	판매 불가
2	103	참이슬 오리지널	20,1도	2100,00	360ml	업소용	판매 불가
3	105	진로	16,9도	1800,00	360ml	업소용	판매 불가
4	107	일품진로	25도	12000,00	375ml	업소용	판매 불가

[그림 40] 제품 특성 확인 프로시저

이는 제품의 특성을 볼 수 있는 기능이다. 먼저 업소용 또는 가정용을 입력할 '용도'를 파라미터로 만들고 만약 업소용을 입력하였을 때 비고에 업소용이라고 적힌 제품들만 조회될 수 있도록 RIGHT OUTER JOIN을 이용하여 만들었다. 처음 만들 때는 제품명을 파라미터로 하고, '%와일드카드%'를 사용하여 만약 '참이슬'을 입력하였을 경우 '참이슬 fresh', '참이슬 오리지널' 로서 참이슬이 들어가는 제품이 모두 조회되도록 만들려고 하였지만 오류가 생기는 바람에 보고서에 담지 못하였다.

(4) 재고 - 출하정보관리

```

ALTER procedure [dbo].[출하정보관리] (@날짜입력 as char(100))
as
begin
begin
select Z.물류센터ID, Z.입력날짜, J.제품명, Z.출하량, D.도매업체ID, D.도매업체주소_시, D.도매업체주소_구, D.도매업체주소_나머지, D.전화번호
from (재고 as Z INNER JOIN 도매업체 as D ON Z.물류센터ID=D.물류센터ID) INNER JOIN 제품 as J ON Z.제품ID=J.제품ID
where @날짜입력=입력날짜
end
IF @@ROWCOUNT=0
print '해당 날짜의 출하기록이 없습니다.';
end;
exec 출하정보관리 '2019-04-25';

```

	물류센터ID	입력날짜	제품명	출하량	도매업체ID	도매업체주소_시	도매업체주소_구	도매업체주소_나머지	전화번호
1	303	2019-04-25	참이슬 오리지널	200	201	서울	마포구	22-2	02-2549-4582
2	301	2019-04-25	참이슬 fresh	300	202	서울	서대문구	25-1	02-5954-1265

[그림 41] 출하정보관리 프로시저

이는 날짜를 파라미터로 하여 원하는 날짜를 입력했을 때 그 날짜에 출하를 한 기록이 있다면 출하를 한 물류센터와 제품명, 출하량, 제품을 받은 도매업체 정보 등을 볼 수 있도록 만들었다. 이를 통해 기업과 물류센터는 해당 출하제품이 어느 도매업체로 갈 것인지, 혹은 갖는지 알 수 있다. 그리고 IF @@ROWCOUNT = 0을 통해 입력한 날짜에 출하가 없었다면 '해당 날짜의 출하기록이 없습니다.' 라는 문구가 나오도록 만들었다.

(5) 판매

(5-1) 판매계약조건

```

ALTER procedure [dbo].[판매계약조건] (@판매계약번호 AS char(100))
as
begin
select D.마트명, P.계약제품명, P.계약금, P.판매수량, P.판매계약날짜
from (판매계약 as P INNER JOIN 바이어 as B ON P.바이어ID=B.바이어ID)
INNER JOIN 대형마트 as D ON B.마트ID=D.마트ID
where @판매계약번호=판매계약번호
end;
exec 판매계약조건 911010;

```

	마트명	계약제품명	계약금	판매수량	판매계약날짜
1	대형마트	진로	2000000.00	200	2019-04-19

[그림 42] 판매계약조건 프로시저

이 기능은 기업 직원과 바이어가 계약을 맺을 때 보여줄 판매계약조건이다. 판매계약번호를 치면 계약할 마트명과 계약제품명, 계약금, 판매 수량, 판매계약날짜를 볼 수 있다. 이를 통해 직원과 바이어는 계약의 조건을 확인할 수 있다.

(5-2) 마트별 판매계약

```

ALTER procedure [dbo].[마트별판매계약]
as
begin
    select M.마트명, P.계약제품명, COUNT(P.판매계약번호) as 계약횟수, SUM(P.계약금) as 총계약금, SUM(P.판매수량) as 총수량
    from (판매계약 as P INNER JOIN 바이어 as B ON P.바이어ID=B.바이어ID)
    INNER JOIN 대형마트 as M ON B.마트ID=M.마트ID
    group by 마트명, 계약제품명
    order by 마트명
end;
exec 마트별판매계약;

```

	마트명	계약제품명	계약횟수	총계약금	총수량
1	대농마트	진로	1	2000000.00	200
2	대농마트	참이슬 오리지널	1	2000000.00	200
3	대농마트	참이슬 fresh	2	4000000.00	400
4	롯데마트	진로	2	3000000.00	300
5	롯데마트	참이슬 오리지널	2	6000000.00	600
6	롯데마트	참이슬 fresh	2	8000000.00	800
7	이마트	참이슬 오리지널	1	3000000.00	300
8	이마트	참이슬 fresh	3	6000000.00	600
9	홈플러스	일품진로	3	6000000.00	300
10	홈플러스	진로	2	4000000.00	400
11	홈플러스	참이슬 fresh	1	1000000.00	100

[그림 43] 마트별 판매계약 프로시저

기업의 마케팅부서는 각 마트별로 진행한 판매계약의 횟수와 계약금 등을 한 눈에 볼 수 있는 기능이 통계를 위해 필요하다 생각했다. COUNT로 판매계약을 체결한 횟수를, SUM으로 총 계약금과 총 계약수량을 볼 수 있으며 그 결과를 대형마트별, 계약제품별로 상세히 보여준다. 이 기능을 실행함으로써 어떤 마트가 어떤 제품을 몇 번 계약했는지 총 몇 개의 수량을 계약했는지 한 눈에 파악할 수 있다.

(6) 광고

(6-1) 광고계약조건 - 판매계약조건과 속성은 다르지만 형식이 거의 동일하다. 위의 판매계약 조건 조회와 같으므로 사진과 설명은 생략한다.

(6-2) 모델별 광고 종류 및 횟수

<pre> ALTER procedure [dbo].[모델별 광고종류와횟수] (@모델이름 as char(100)) as begin select M.소속사, M.모델이름, C.광고종류, COUNT(M.소속사) as 광고횟수, SUM(C.광고비) as 총광고비 from 광고계약 as C INNER JOIN 광고모델 as M ON C.회사ID=M.회사ID WHERE @모델이름=M.모델이름 group by 소속사, 모델이름, 광고종류 end; exec 모델별 광고종류와횟수 강윤철; </pre>																							
<div>100 %</div> <div>Results Messages</div> <table> <tr> <th></th><th>소속사</th><th>모델이름</th><th>광고종류</th><th>광고횟수</th><th>총광고비</th></tr> <tr> <td>1</td><td>hi</td><td>강윤철</td><td>라디오</td><td>2</td><td>11000000.00</td></tr> <tr> <td>2</td><td>hi</td><td>강윤철</td><td>TV</td><td>2</td><td>20000000.00</td></tr> </table>							소속사	모델이름	광고종류	광고횟수	총광고비	1	hi	강윤철	라디오	2	11000000.00	2	hi	강윤철	TV	2	20000000.00
	소속사	모델이름	광고종류	광고횟수	총광고비																		
1	hi	강윤철	라디오	2	11000000.00																		
2	hi	강윤철	TV	2	20000000.00																		

[그림 44] 모델별 광고 종류 및 횟수 프로시저

각 광고 모델 별로 우리 기업의 광고의 어떤 종류를 몇 회 했는지에 대해 볼 수 있는 기능이 다. 이 또한 마케팅을 계획하기 위해 필요한 정보라고 생각한다. 광고계약과 광고모델을 회사 ID로 INNER JOIN 하여 모델명을 입력하면 그 모델이 계약한 광고종류와 광고 횟수를 볼 수 있다.

(7) 기업

(7-1) 직원 정보 관리

```

ALTER procedure [dbo].[직원정보관리]
(
    @type as char(100)='',
    @직원 ID as int,
    @직원이름 as char(100),
    @부서 as char(100),
    @직원이메일 as char(100),
    @직원연락처 as char(100)
)
as
begin
    begin
        if @type = '입력'
        begin
            INSERT INTO 직원 (직원 ID, 직원이름, 부서, 직원이메일, 직원연락처)
            VALUES (@직원 ID, @직원이름, @부서, @직원이메일, @직원연락처)
        end
        if @type = '조회'
        begin
            SELECT * FROM 직원 WHERE 직원 ID=@직원 ID
        end
        if @type = '수정'
        begin
            UPDATE 직원 SET 직원이름=@직원이름, 부서=@부서, 직원이메일=@직원이메일, 직원연락처=@직원연락처
            WHERE 직원 ID=@직원 ID
        end
        else if @type = '삭제'
        begin
            DELETE FROM 직원 WHERE 직원 ID=@직원 ID
        end
        select * from 직원
    end
end;

```

[그림 45] 직원정보관리 프로시저

이 기능은 처음 설명했던 (1) 기능과 유사하므로 짧게 설명 하고 넘어가겠다. type이라는 parameter을 추가하여 type에 '입력', '조회', '수정', '삭제'를 입력하고 직원정보를 입력하면 직원 정보가 자동으로 변경되는 간편한 기능이다.

(7-2) 부서 실적 조회

```
ALTER procedure [dbo].[부서실적조회] (@부서 as char(100))
as
begin
    IF @부서='영업부서'
    begin
        select E.부서, E.직원이름, COUNT(P.판매계약번호) as 판매계약횟수
        from 직원 as E INNER JOIN 판매계약 as P ON E.직원 ID=P.직원 ID
        group by 부서, 직원이름
    end
    IF @부서='마케팅부서'
    begin
        select E.부서, E.직원이름, COUNT(G.광고계약번호) as 광고계약횟수
        from 직원 as E INNER JOIN 광고계약 as G ON E.직원 ID=G.직원 ID
        group by 부서, 직원이름
    end
    else
    begin
        print '실적 조회 되는 부서가 아닙니다.'
    end
end;
```

exec 부서실적조회 마케팅부서 ;

100 %

Results Messages

	부서	직원이름	광고계약횟수
1	마케팅부서	박보검	7
2	마케팅부서	홍혜림	8

[그림 46] 부서 실적 조회 프로시저

부서 실적 조회는 기업에게 꼭 필요한 기능이다. 이는 실적에 따라 후에 인사 평가라던지 승진 평가에 참고될 수 있기 때문이다. 현재 우리가 만든 시스템에서는 판매계약과 광고계약으로 계약이 2가지가 있기 때문에 2가지 계약을 기준으로 만들었다. 우선 부서명을 Parameter로 하여 실적 조회를 원하는 부서명을 입력하도록 하였다. 그리고 판매계약은 영업부서만이 하고 광고계약은 마케팅부서만이 계약할 수 있다. 그래서 '영업부서'를 입력한다면 판매계약 table에서 COUNT를 통해 직원별 판매계약 횟수를 알 수 있고, '마케팅부서'를 입력한다면 광고계약 table에서 COUNT를 통해 직원별 광고계약 횟수를 알 수 있다. 이를 통해 각 부서 별 계약 체결 횟수를 알 수 있으며, 현재로서는 영업부서, 마케팅부서 외의 다른 부서명을 입력할 경우 '실적 조회 되는 부서가 아닙니다.'라는 문구가 나오도록 프로시저를 설계하였다.

4. 결론 및 소감

‘소주’라는 주제 선정 배경부터 획기적이라고 생각했다. 무엇보다 20대를 대표하고 우리가 가장 많이 접할 수 있으며 트렌드를 이끌어 나갈 수 있는 것이 바로 ‘소주’였다. 하지만 그것이 우리 소비자로 오기까지의 기업과 관련하여 어떠한 각종 업체들에 의해 전달(유통) 되어지는가 의문이었다. 또한 유통에 대한 전반적인 시스템을 구축하게 되면 그것을 사용하는 기업 및 업체의 입장에서 재고에 대한 비용 감소 효과로 효율적일 것이라 생각했다. 처음 유통에 관한 정보를 검색해볼 때 많은 어려움이 있었다. 주류라는 자체가 만 19세 이상의 품목이기도 하고 자세한 유통에 대한 구조는 정보 검색으로는 한계가 있었다. 현실에 부딪혀 보기로 했다. 직접 ‘하이트진로’ 고객센터에 전화를 하였다. “안녕하세요. 저희는 홍익대학교 학생입니다. 과제를 선정하던 중 ‘소주’의 유통과정에 대해 알고 싶어서 이렇게 전화 드렸습니다.”. 돌아온 답변은 굉장히 만족스러웠다. 친절하게 우리의 질문에 응해주셨고 기업의 기밀이기 때문에 자세하게는 알려드리지 못하지만 전반적인 과정에 대해서는 말씀드리겠습니다. 라는 답변을 들을 수 있었다. 그 후 전반적인 프로젝트 수행과정에서 큰 어려움을 느끼지 못했지만 ERD 작성하는 부분에서 많은 시행착오가 있었다. 교수님께도 주기적으로 질문하고 조원들 또한 하루에 두 번씩 만나면서 머리를 맞대고 수정을 반복했다. 결과적으로 쿼리를 작성하고 프로시저를 만드는 부분에서 큰 오류 없이 우리가 원하는 기능 요구 사항에 대해 모두 적용할 수 있었다. 처음으로 팀 프로젝트를 하면서 팀원들과 12시간 이상 같이 보내기도 하며 정말 많은 노력이 담긴 것 같다. 한 학기 동안 데이터베이스 수업을 들으며 실질적으로 적용해볼 수 있던 좋은 기회였던 것 같다.

5. 부록 - SQL로 구현한 모든 Procedure

SQL 상으로 구현한 모든 Procedure 문을 부록으로 첨부한다.

```
/* 데이터 단순 조회 */
create procedure 물류센터조회
as
begin
    select 물류센터.*
    from 물류센터
end;
```

```
/* 입력, 수정, 삭제 후 조회 */
create procedure 데이터관리
(
    @type as char(100)='',
    @물류센터ID as int,
    @전화번호 as char(100),
    @물류센터주소_시 as char(100),
    @물류센터주소_구 as char(100),
```

```

@물류센터주소_나머지 as char(100)
)
as
begin
    begin
        if @type = '입력'
        begin
            INSERT INTO 물류센터 (물류센터ID, 전화번호, 물류센터주소_시,
물류센터주소_구, 물류센터주소_나머지)
            VALUES (@물류센터ID, @전화번호, @물류센터주소_시, @물류센
터주소_구, @물류센터주소_나머지)
        end
        if @type = '수정'
        begin
            UPDATE 물류센터 SET 전화번호=@전화번호, 물류센터주소_시=@
물류센터주소_시, 물류센터주소_구=@물류센터주소_구, 물류센터주소_나머지=@물류센터주소_
나머지
            WHERE 물류센터ID=@물류센터ID
        end
        else if @type ='삭제'
        begin
            DELETE FROM 물류센터 WHERE 물류센터ID=@물류센터ID
        end
        select * from 물류센터
    end
end;

/* 주문 - 주문서 확인 */
create procedure 주문서확인 (@주문날짜 AS char(100))
as
begin
    select O.주문번호, D.도매업체ID, J.제품명, J.비고 ,S.주문량, Z.물류센터ID, Z.재고
량
    from (((도매업체 as D INNER JOIN 주문 as O ON D.도매업체ID=O.도매업체ID)
        INNER JOIN 상세주문 as S ON O.주문번호=S.주문번호)
        INNER JOIN 재고 as Z ON S.제품ID=Z.제품ID)
        INNER JOIN 제품 as J ON S.제품ID=J.제품ID

    where O.주문날짜=Z.입력날짜
        and @주문날짜=주문날짜
end;

```



```

/* 주문 - 소주별주문통계 */
create procedure 소주별주문통계
as
begin
    select J.제품명, J.비고, SUM(S.주문량) as 주문량합계, AVG(S.주문량) as 평균주
문량, COUNT(S.주문량) as 주문횟수
    from 제품 as J INNER JOIN 상세주문 as S ON J.제품ID=S.제품ID
    group by 제품명, 비고
    order by 제품명
end;

```

```

/* 주문 - 도매업체별주문통계 */
create procedure 도매업체별주문통계
as
begin
    select O.도매업체ID, J.제품명, SUM(S.주문량) as 전체주문량, AVG(S.주문량) as 평
균주문량, COUNT(S.주문번호) as 주문횟수
    from (주문 as O INNER JOIN 상세주문 as S ON O.주문번호=S.주문번호) INNER
JOIN 제품 as J ON S.제품ID=J.제품ID
    group by 도매업체ID, 제품명
    order by 도매업체ID
end;

```

```

/* 주문 - 금액별주문서 */
create procedure 금액별주문서 (@최저금액 as money, @최고금액 as money)
as
begin
    select O.주문번호, J.제품명, S.주문량, S.총주문가격, J.가격
    from (상세주문 as S INNER JOIN 주문 as O ON S.주문번호=O.주문번호) INNER
JOIN 제품 as J ON S.제품ID=J.제품ID
    where S.총주문가격 >= @최저금액 and
           S.총주문가격 <= @최고금액
    order by S.총주문가격
end;

```

```

/* 제품 - 제품특성확인 */
create procedure 제품특성확인 (@용도 AS char(100))
as
begin
    if @용도='업소용'
    begin

```

```

        select J.*, BU.판매여부
        from 제품 as J RIGHT OUTER JOIN 업소용 as BU ON J.제품ID=BU.제품
ID
    end
    if @용도='가정용'
    begin
        select J.*, FU.판매여부
        from 제품 as J RIGHT OUTER JOIN 가정용 as FU ON J.제품ID=FU.제품
ID
    end
end;

```

```

/* 재고 - 제품별 평균 재고량 */
create procedure 제품별평균재고량
as
begin
    select J.제품명, AVG(Z.재고량) as 평균재고량, AVG(Z.출하량) as 평균출하량
    from 재고 as Z INNER JOIN 제품 as J ON Z.제품ID=J.제품ID
    group by 제품명
end;

```

```

/* 재고 - 출하정보관리 */
create procedure 출하정보관리 (@날짜입력 as char(100))
as
begin
    begin
        select Z.물류센터ID, Z.입력날짜, J.제품명, Z.출하량, D.도매업체ID ,D.도매업체주소
        _시, D.도매업체주소_구, D.도매업체주소_나머지, D.전화번호
        from (재고 as Z INNER JOIN 도매업체 as D ON Z.물류센터ID=D.물류센터ID)
INNER JOIN 제품 as J ON Z.제품ID=J.제품ID
        where @날짜입력=입력날짜
        end
        IF @@ROWCOUNT=0
            print '해당 날짜의 출하기록이 없습니다.';
end;

```

```

/* 판매 - 판매계약조건 */
create procedure 판매계약조건(@판매계약번호 AS char(100))
as
begin
    select D.마트명, P.계약제품명, P.계약금, P.판매수량, P.판매계약날짜

```

```

        from (판매계약 as P INNER JOIN 바이어 as B ON P.바이어ID=B.바이어ID)
            INNER JOIN 대형마트 as D ON B.마트ID=D.마트ID
        where @판매계약번호=판매계약번호
end;

/* 판매 - 판매계약내역조회 */
create procedure 판매계약내역조회 (@판매계약번호 as int)
as
begin
    select P.판매계약번호, M.마트명, B.바이어이름, E.직원이름, P.판매계약날짜, P.계약
    제품명, P.판매수량, P.계약금
    from ((판매계약 as P INNER JOIN 바이어 as B ON P.바이어ID=B.바이어ID)
        INNER JOIN 직원 as E ON P.직원ID=E.직원ID) INNER JOIN 대형마트 as
    M ON B.마트ID=M.마트ID
    where @판매계약번호=판매계약번호
end;

/* 판매 - 바이어실적조회 */
create procedure 바이어실적조회 (@바이어ID as int)
as
begin
    select B.바이어이름, COUNT(P.판매계약번호) as 계약체결횟수, SUM(P.계약금) as
    총계약금
    from 판매계약 as P INNER JOIN 바이어 as B ON P.바이어ID=B.바이어ID
    where @바이어ID=B.바이어ID
    group by 바이어이름
end;

/* 판매 - 마트별판매계약 */
create procedure 마트별판매계약
as
begin
    select M.마트명, P.계약제품명, COUNT(P.판매계약번호) as 계약횟수, SUM(P.계약
    금) as 총계약금, SUM(P.판매수량) as 총수량
    from (판매계약 as P INNER JOIN 바이어 as B ON P.바이어ID=B.바이어ID)
        INNER JOIN 대형마트 as M ON B.마트ID=M.마트ID
    group by 마트명, 계약제품명
    order by 마트명
end;

```

```

/* 광고 - 광고계약조건 조회 */
create procedure 광고계약조건(@광고계약번호 AS char(100))
as
begin
    select A.광고종류, A.광고비, A.광고시작일, A.광고종료일, B.모델이름
    from 광고계약 as A INNER JOIN 광고모델 as B
        ON A.회사ID=B.회사ID
    where @광고계약번호=광고계약번호
end;

/* 광고 - 광고계약내역조회 */
create procedure 광고계약내역조회 (@광고계약번호 as int)
as
begin
    select G.광고계약번호, E.직원이름 ,G.광고계약날짜, G.광고종류, M.소속사, M.모델
    이름, G.광고시작일, G.광고종료일
    from (광고계약 as G INNER JOIN 광고모델 as M ON G.회사ID=M.회사ID) INNER
    JOIN 직원 as E ON G.직원ID=E.직원ID
    where @광고계약번호=광고계약번호
end;

/* 광고 - 모델별 광고종류 및 횟수 */
create procedure 모델별광고종류와횟수 (@모델이름 as char(100))
as
begin
    select M.소속사, M.모델이름, C.광고종류, COUNT(M.소속사) as 광고횟수, SUM(C.
    광고비) as 총광고비
    from 광고계약 as C INNER JOIN 광고모델 as M ON C.회사ID=M.회사ID
    WHERE @모델이름=M.모델이름
    group by 소속사, 모델이름, 광고종류
end;

/* 기업 - 업무별 담당자확인 */
create procedure 담당자확인 (@부서명 as char(100))
as
begin
    select *
    from 직원
    where @부서명=부서
end;

```

```

/* 기업 - 직원 정보 관리 */
create procedure 직원정보관리
(
    @type as char(100)='',
    @직원ID as int,
    @직원이름 as char(100),
    @부서 as char(100),
    @직원이메일 as char(100),
    @직원연락처 as char(100)
)
as
begin
    begin
        if @type = '입력'
        begin
            INSERT INTO 직원 (직원ID, 직원이름, 부서, 직원이메일, 직원연
            락처)
            VALUES (@직원ID, @직원이름, @부서, @직원이메일, @직원연락
            처)

        end
        if @type = '조회'
        begin
            SELECT * FROM 직원 WHERE 직원ID=@직원ID
        end
        if @type = '수정'
        begin
            UPDATE 직원 SET 직원이름=@직원이름, 부서=@부서, 직원이메일
            =@직원이메일, 직원연락처=@직원연락처
            WHERE 직원ID=@직원ID
        end
        else if @type = '삭제'
        begin
            DELETE FROM 직원 WHERE 직원ID=@직원ID
        end
        select * from 직원
    end
end;

```

```

/* 기업 - 부서실적조회 */
create procedure 부서실적조회 (@부서 as char(100))
as

```

```

begin
    IF @부서='영업부서'
    begin
        select E.부서, E.직원이름, COUNT(P.판매계약번호) as 판매계약횟수
        from 직원 as E INNER JOIN 판매계약 as P ON E.직원ID=P.직원ID
        group by 부서, 직원이름
    end
    IF @부서='마케팅부서'
    begin
        select E.부서, E.직원이름, COUNT(G.광고계약번호) as 광고계약횟수
        from 직원 as E INNER JOIN 광고계약 as G ON E.직원ID=G.직원ID
        group by 부서, 직원이름
    end
    else
    begin
        print '실적 조회 되는 부서가 아닙니다.'
    end
end
end;

```

6. 참고문헌

한국주류산업협회(주류 제조 및 유통 과정)

http://www.kalia.or.kr/bbs/board.php?tbl=bbs11&mode=VIEW&num=8&category=&findType=&findWord=&sort1=&sort2=&page=1&mobile_flag=

하이트진로

고객센터 대표전화 : 080-210-0150