2019 Database System

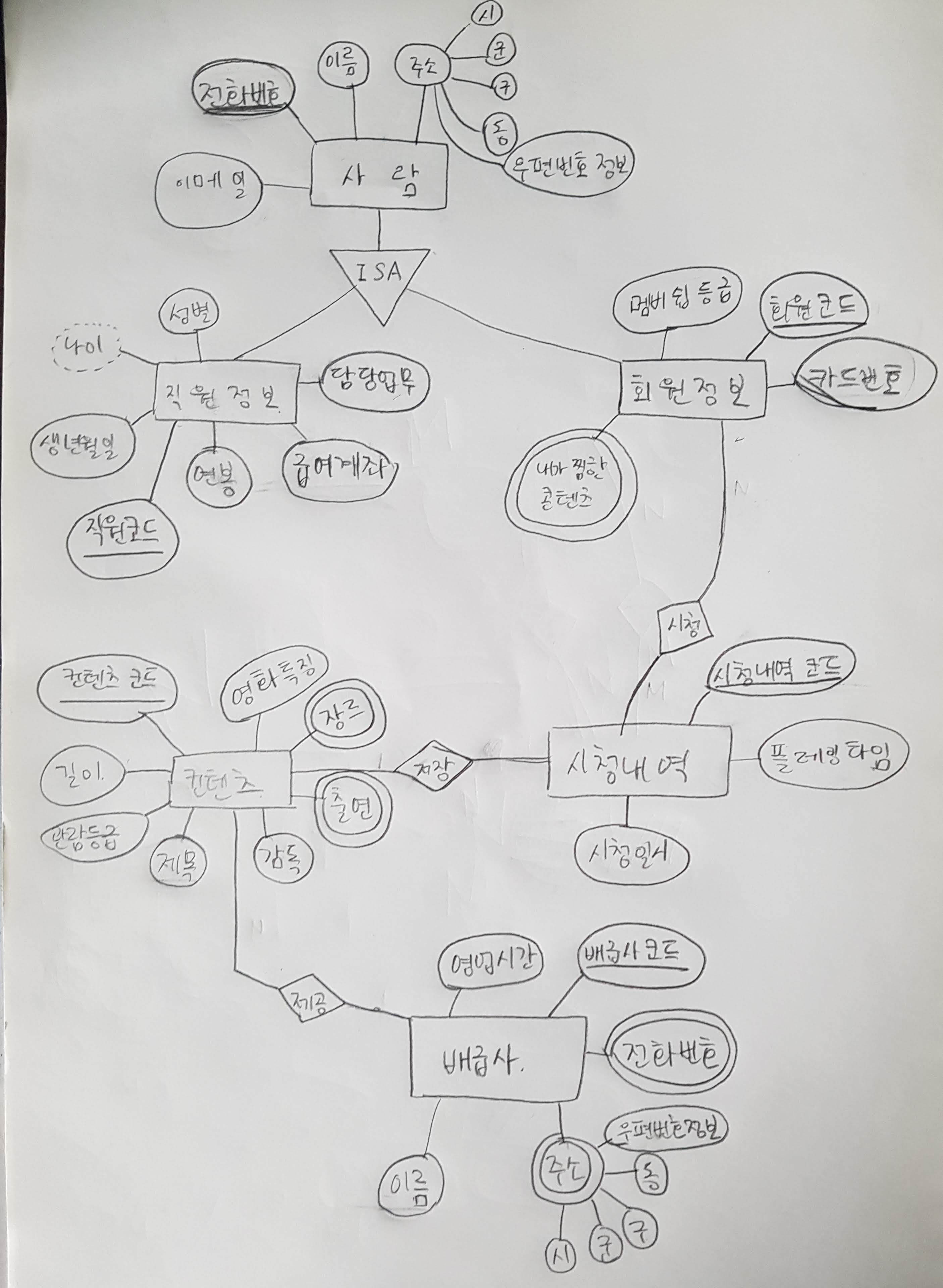
Project #1

이름 : 이승철

학과 : 컴퓨터공학과

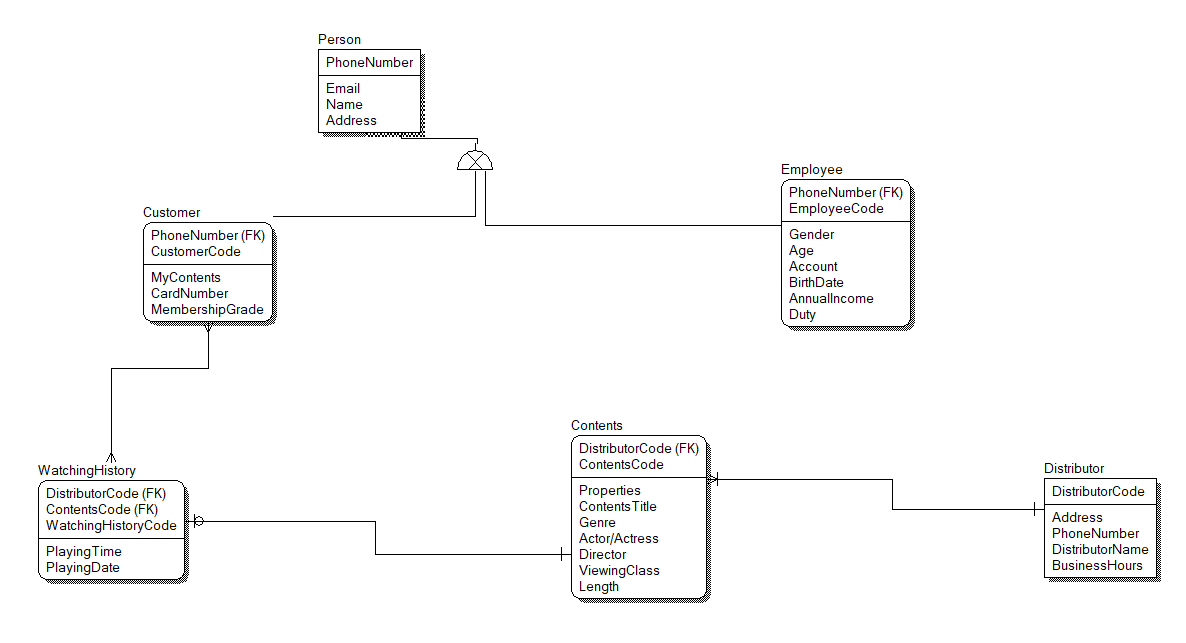
학번 : 20141561

**1. ER 다이어그램**

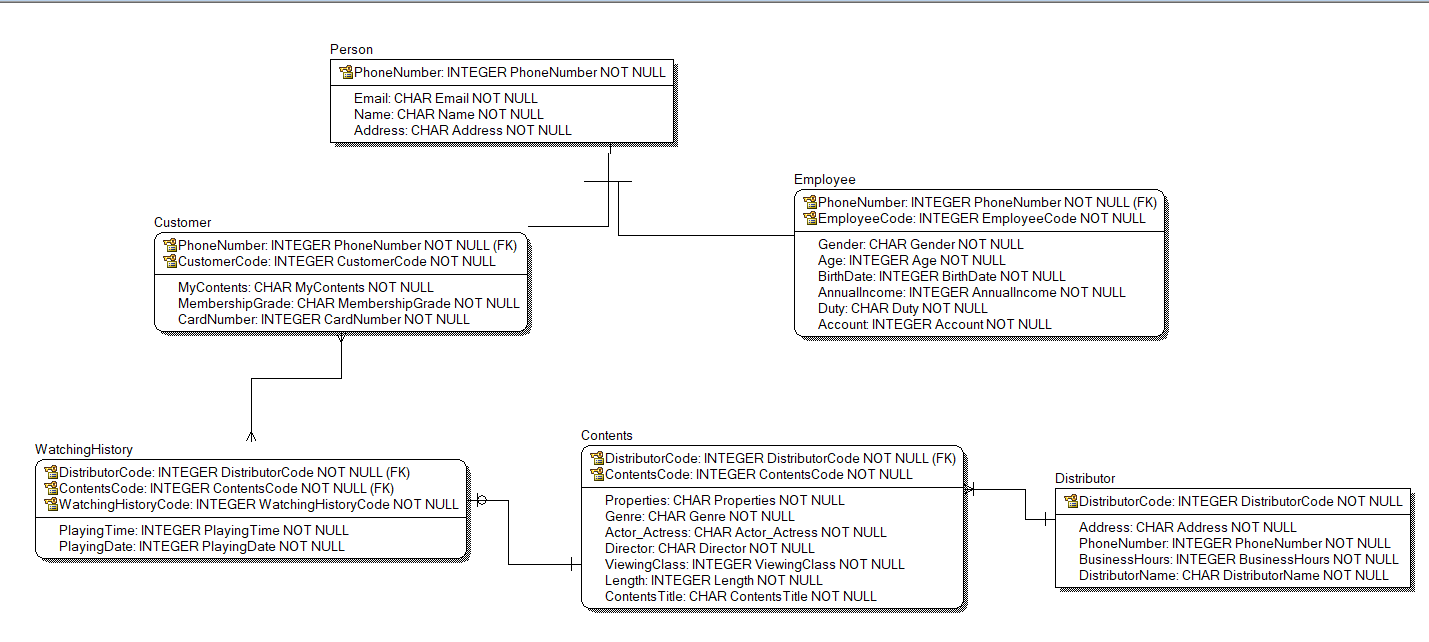
****

* 각 파트에 대한 설명
  + 사람 entity : 회원과 직원 entity가 같은 사람으로 묶일 수 있으므로 사람 entity를 만들어서 회원과 직원의 상위 entity로 설정을 해주었다. 그리고 회원과 직원 entity의 공통 attribute를 새로운 attribute로 설정을 해주었다. 그래서 이메일, 주소, 이름, 전화번호, 주소가 attribute가 되고 그 중에 주소는 시, 군, 구, 동, 우편번호 정보를 갖는 복합 attribute가 된다. 그리고 전화번호 attribute를 primary key로 설정을 했다. 왜냐하면 한 사람은 하나의 전화번호를 갖고 있다고 판단을 했기 때문이다.
  + 직원 정보 entity : 직원 정보에 관한 것을 저장하고 관리하는 entity이다. 이 entity에는 사람 entity에서 있는 attribute도 물론 포함하고 성별, 나이, 생년월일, 연봉, 급여계좌, 담당업무, 직원 코드를 attribute 포함한다. 그 중에서 나이는 성별로부터 파생되어질 수 있으므로 derived attribute로 나타내어진다. 그리고 직원코드 attribute를 추가해서 한 직원 당 하나의 integer형식의 겹치지 않는 고유 직원코드를 갖는다고 가정을 했기 때문에 이 attribute가 primary key가 된다고 판단을 했다.
  + 회원 정보 entity : 회원 정보에 관한 것을 저장하고 관리하는 entity이다. 이 entity에는 사람 entity에 있는 attribute도 물론 포함하고 내가 찜한 콘텐츠, 카드 번호, 회원 코드, 멤버쉽 등급 attribute를 포함한다. 그 중에서 내가 찜한 콘텐츠는 한 회원이 여러 개의 콘텐츠를 찜할 수 있으므로 multi-valued attribute로 설정을 했다. 그리고 회원코드 attribute를 추가해서 한 회원 당 하나의 integer형식의 겹치지 않는 고유 회원 코드를 갖는다고 가정을 했기 때문에 이 attribute가 primary key가 된다고 판단을 했다.
  + 배급사 entity : 배급사에 관한 정보를 저장하고 관리하는 entity이다. 이 entity에는 영업시간, 배급사 코드, 전화번호, 주소, 이름 attribute가 있고 그 중에서 전화번호 attribute는 하나의 배급사 내에 여러 개의 전화번호가 있을 수 있으므로 multi-valued attribute로 설정을 했다. 그리고 주소 attribute는 배급사가 여러 군데 있을 수 있으므로 multi-valued attribute로 설정을 하고 그 attribute는 시, 군, 구, 동, 우편번호 정보 attribute를 포함하므로 복합 attribute도 된다고 설정을 했다. 요구사항에서는 컨텐츠 리스트를 포함하는데 이는 컨텐츠 entity와 relation이 되어 있으므로 따로 attribute를 만들지 않았다. 그리고 배급사 코드 attribute를 추가해서 한 배급사 당 하나의 integer형식의 겹치지 않는 고유 코드를 갖는다고 가정을 했기 때문에 이 attribute가 primary key가 된다고 판단을 했다.
  + 컨텐츠 entity : 컨텐츠에 관한 정보를 저장하고 관리하는 entity이다. 이 entity에는 컨텐츠 코드, 길이, 관람등급, 제목, 감독, 출연, 장르, 영화특징 attribute가 있다. 그 중에서 장르는 한 영화에서 미스터리, 스릴러 같이 여러 개의 장르가 있을 수 있으므로 multi-valued attribute로 설정을 했다. 그리고 출연 attribute는 한 영화에 여러 명의 배우가 출연하기 때문에 이 또한 multi-valued attribute로 설정을 했다. 그리고 컨텐츠 코드 attribute를 추가해서 하나의 컨텐츠 당 하나의 integer형식의 겹치지 않는 고유 코드를 갖는다고 가정을 했기 때문에 이 attribute가 primary key가 된다고 판단을 했다. 요구사항에는 이 entity에 배급사에 관한 정보가 있는데 이는 배급사 entity와 relation이 되어 있으므로 따로 attribute를 만들지 않았다.
  + 시청내역 entity : 회원이 시청한 시청내역에 관한 정보를 저장하고 관리하는 entity이다. 이 entity에는 시청내역 코드, 플레잉 타임, 시청 일시 attribute가 있다. 그 중에서 시청내역 코드는 하나의 시청 내역 당 하나의 integer형식의 겹치지 않는 고유 코드를 갖는다고 가정을 했기 때문에 이 attribute가 primary key가 된다고 판단을 했다. 요구사항에서는 고객과 동영상의 정보를 포함한다고 했다. 고객은 시청내역 entity가 회원 정보의 entity와 relation이 되어 있고 동영상의 정보는 컨텐츠 entity와 relation이 되어 있으므로 따로 attribute를 만들지 않았다.
* Relation & Mapping Cardinality 설명
  + 회원정보 – 시청내역 : N:M관계이다. 한 명의 회원은 여러 개의 컨텐츠를 시청할 수 있으므로 여러 개의 시청내역이 존재하고 하나의 시청내역에는 여러 명의 회원이 존재할 수 있기 때문에 N:M의 관계이다.. 그리고 회원정보와 시청내역 사이에는 “시청”이라는 relation을 두고 있는데 이는 회원이 “시청”을 통해서 시청내역에 기록이 된다는 relation이다.
  + 컨텐츠 – 시청내역 : 1:1(or zero)관계이다. 하나의 컨텐츠는 회원이 볼 수도 있고 안 볼 수도 있기 때문에 하나의 시청내역을 갖거나 아예 없을 수 있다. 그리고 하나의 시청내역에는 하나의 컨텐츠가 존재하기 때문에 1:1(or zero)관계이다. 이 관계는 “저장”이라는 relation을 두고 있는데 이는 어떠한 컨텐츠가 시청내역에 “저장”된다는 relation이다.
  + 컨텐츠 – 배급사 : N:1관계이다. 하나의 배급사에는 여러 개의 컨텐츠가 있을 수 있지만 하나의 컨텐츠는 하나의 배급사로 연결 지을 수 밖에 없다. 그리고 그 관계를 “제공”이라는 relation을 두고 있는데 이는 어떠한 배급사가 어떠한 콘텐츠를 “제공”한다는 relation이다.
  + 사람 – 직원정보, 회원정보 : ISA계층을 사용했다. 회원과 직원은 똑같은 상위 entity인 “사람”으로 묶을 수 있기 때문에 “사람”이라는 entity를 통해서 둘 사이의 공통된 attribute를 합쳤고 직원정보와 회원정보를 서브클래스로 설정을 했다.

**2. ER Win을 통한 설계 내용 (Logical Mode)**



**3. ER Win을 통한 설계 내용 (Physical Mode)**



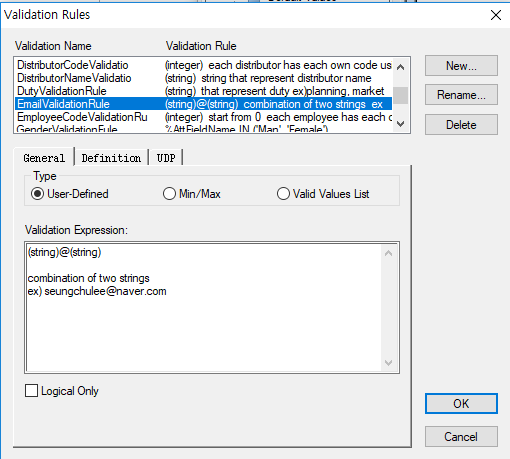
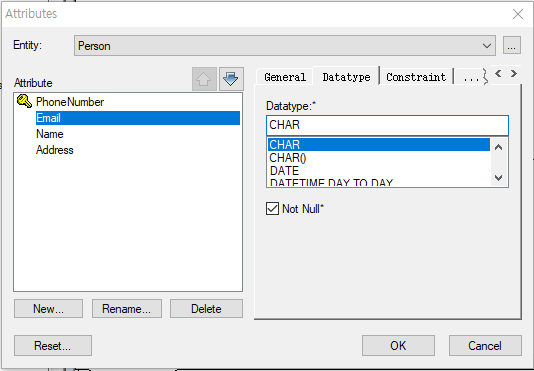
**4. ER Win 모든 설계에 대한 요구사항**

**- Entity & Attributes & key**

1) Person entity

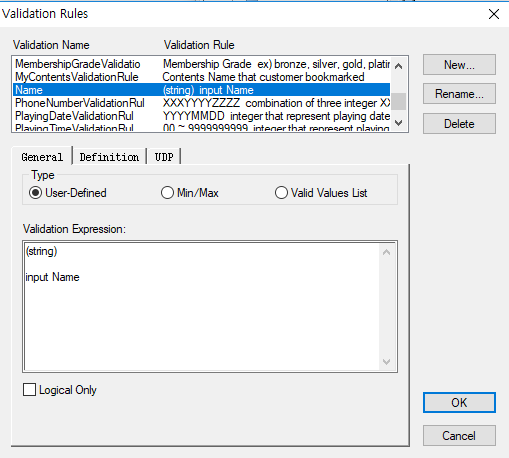
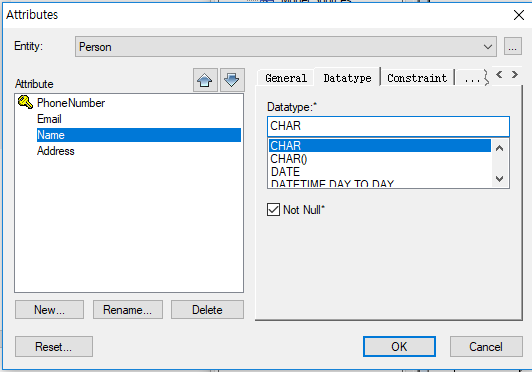
- Email Attribute

사람의 이메일에 관한 정보로 char형으로 저장을 하고 형식은 (string)@(string)형식으로 이루어져 있다.



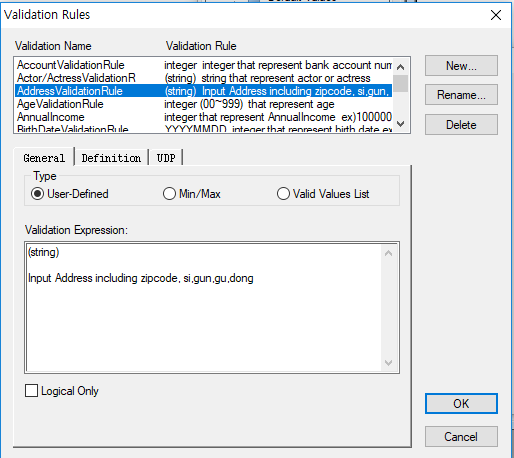
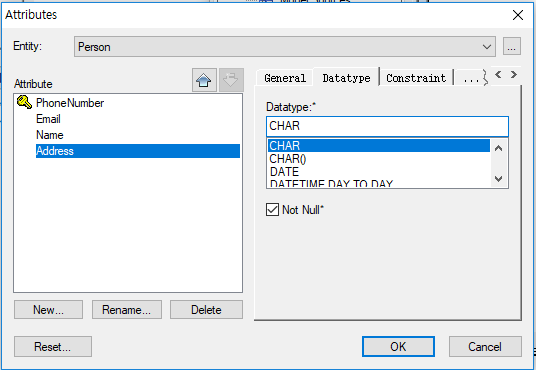
- Name Attribute

사람 이름에 관한 정보로 char형식으로 이루어져 있고 이름을 입력한다.



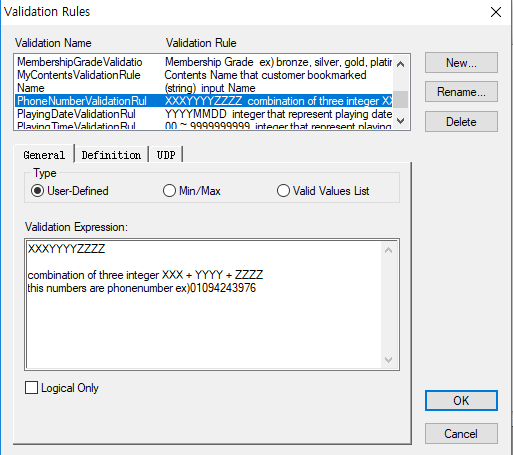
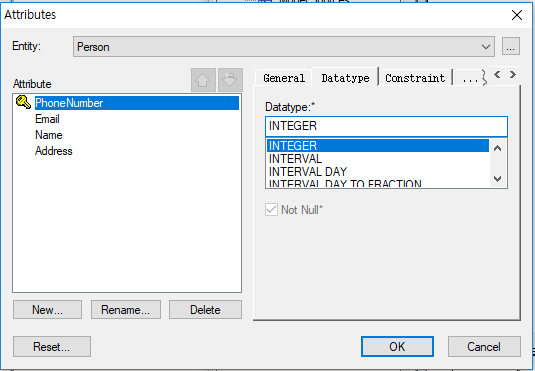
- Address Attribute

사람 주소에 관한 정보로 char형식으로 이루어져 있고 이 주소는 시, 군, 구, 동, 우편번호 정보로 이루어져 있다.



- PhoneNumber Attribute (primary key)

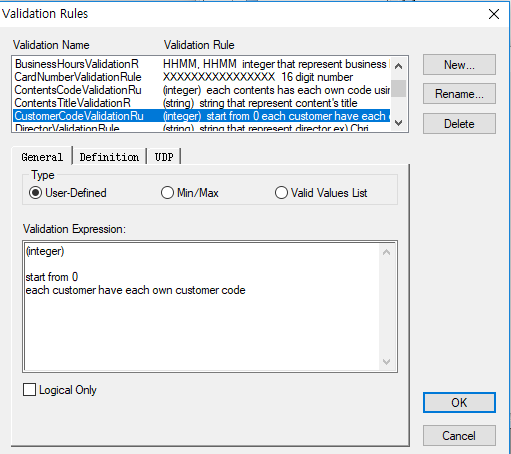
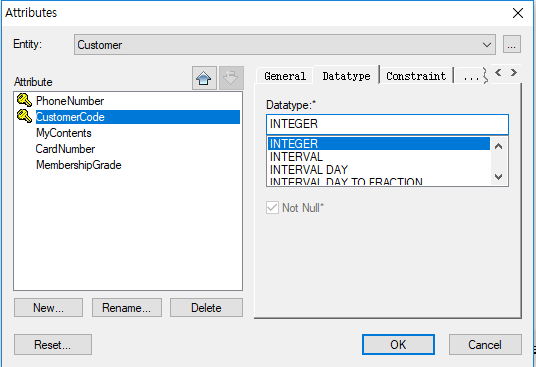
사람 전화번호에 관한 정보로 integer형식으로 이루어져있고 사람마다 핸드폰 번호는 하나로 가정을 했고 이는 primary key가 될 수 있다. XXXYYYYZZZZ형식으로 전화번호가 이루어져 있다고 판단했다.



2) Customer entity

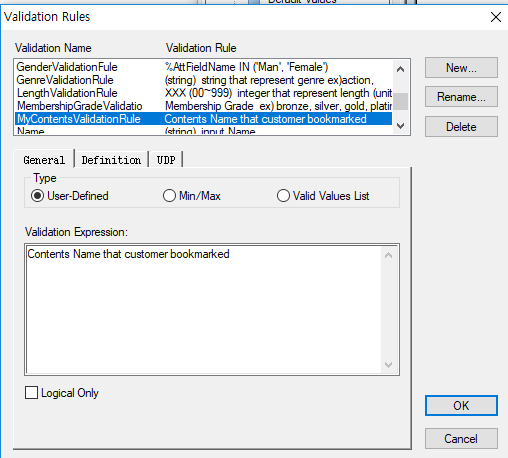
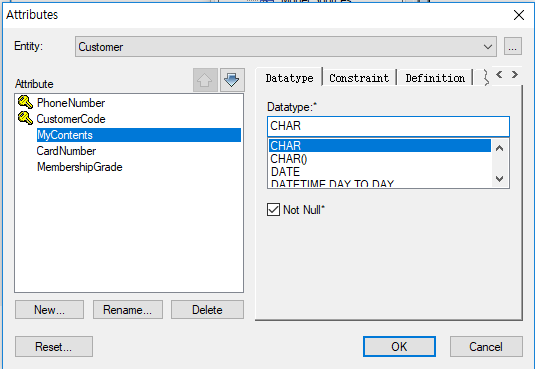
- Customer Code Attribute (primary key)

회원 코드에 관한 정보로 0부터 시작해서 회원 한 명 당 integer형식의 고유 번호를 부여하기 때문에 이는 primary key가 될 수 있고 integer형식으로 이루어져 있다.



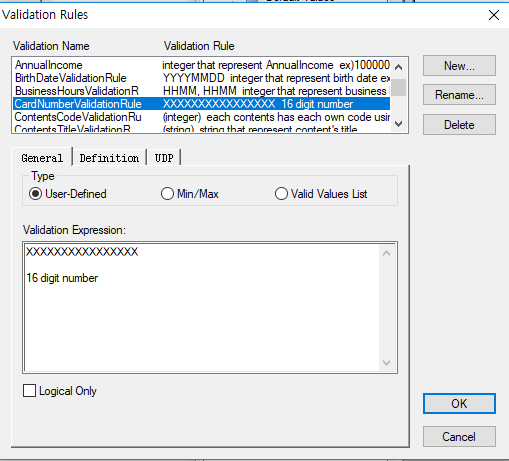
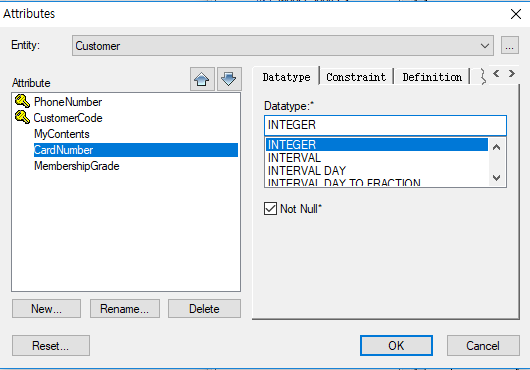
- MyContents attribute

내가 찜한 컨텐츠에 관한 정보로 char형식으로 이루어져 있고 회원들이 찜한 컨텐츠를 저장한다. 여러 컨텐츠를 찜할 수 있으므로 여러 개를 입력할 수 있다.



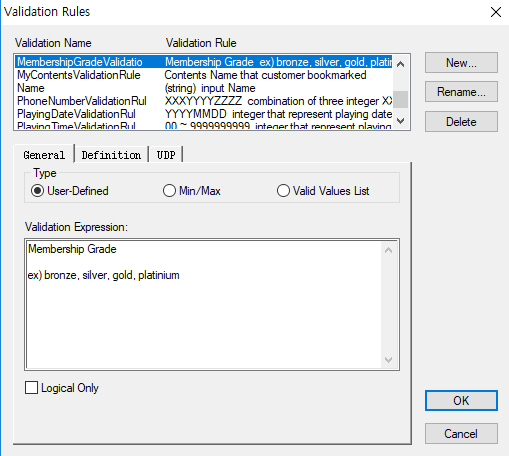
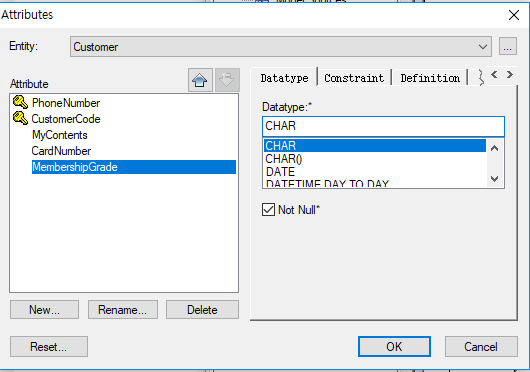
* CardNumber attribute

회원 카드 번호에 관한 정보로 integer형식으로 이루어져 있고 카드번호 16자리를 입력하게 되어있다.

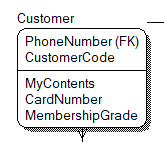


* MembershipGrade attribute

멤버쉽 등급에 관한 정보로 char 형식으로 이루어져 있고 bronze, silver, gold 같은 등급을 저장한다.



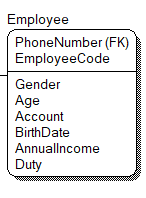
* PhoneNumber (Foreign Key)



사람 entity에서 PhoneNumber가 Primary key로 판단을 했기 때문에 그 subclass인 customer entity에서 foreign key로 판단을 할 수 있다.

3) Employee entity

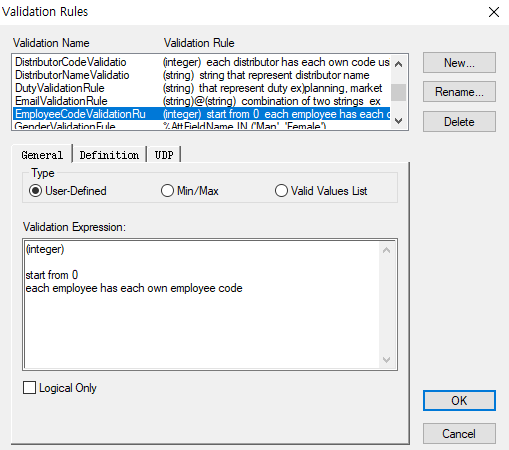
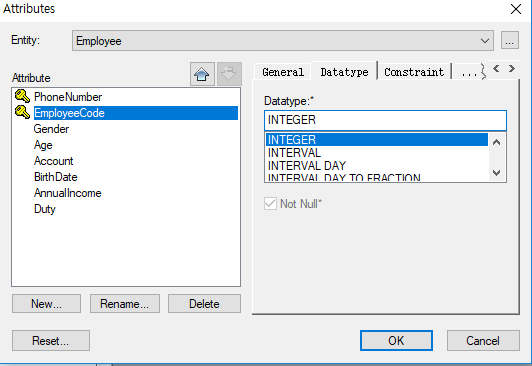
- PhoneNumber (Foreign Key)



사람 entity에서 PhoneNumber가 Primary key로 판단을 했기 때문에 그 subclass인 employee entity에서 foreign key로 판단을 할 수 있다.

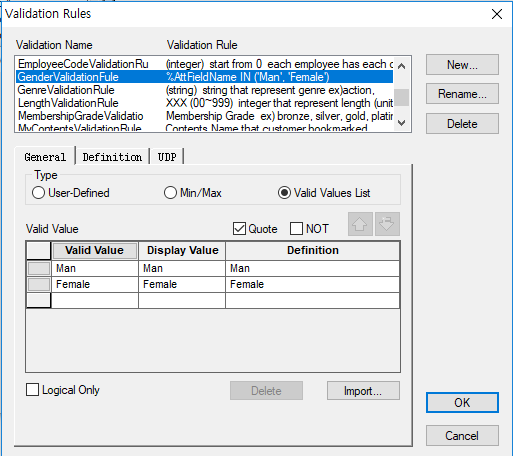
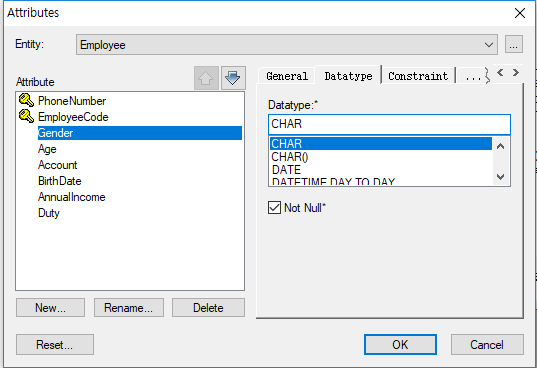
- EmployeeCode attribute (primary key)

직원 코드에 관한 정보로 각 직원마다 0부터 시작하는 integer형식의 고유 번호가 부여되어 primary key의 기능을 할 수 있고 integer형식으로 이루어져 있다.



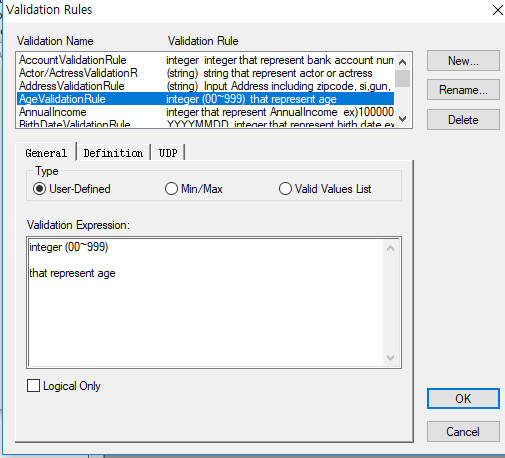
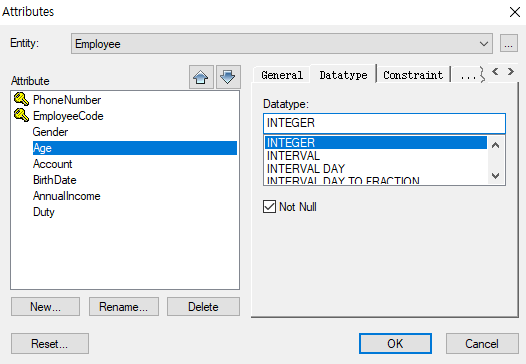
* Gender attribute

직원의 성별에 관한 정보로 char 형식으로 이루어져 있고 Male과 Female 두 개의 값을 가진다.



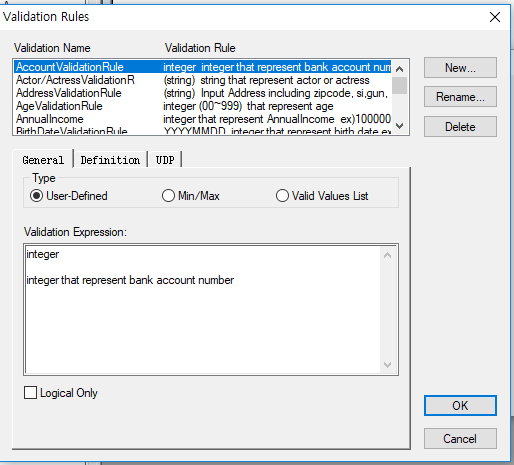
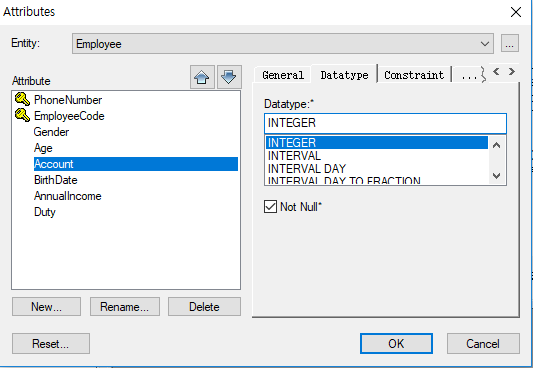
* Age attribute

직원의 나이에 관한 정보로 integer형식으로 이루어져 있고 나이를 저장하게 된다.



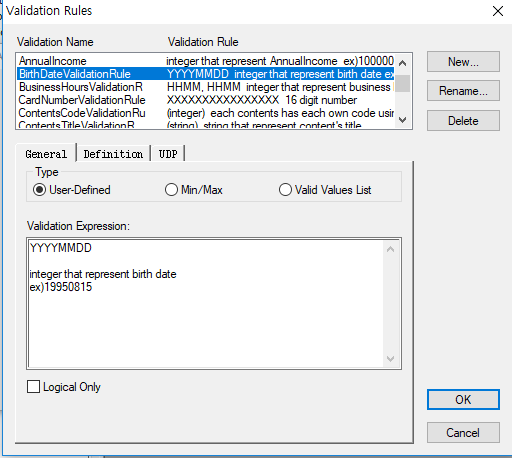
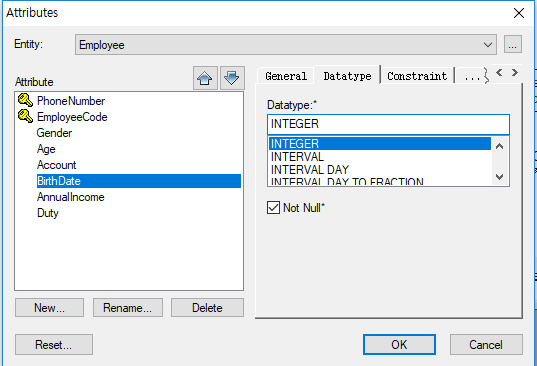
* Account attribute

직원 급여계좌에 관한 정보로 integer형식으로 이루어져 있고 계좌번호를 저장하게 된다.



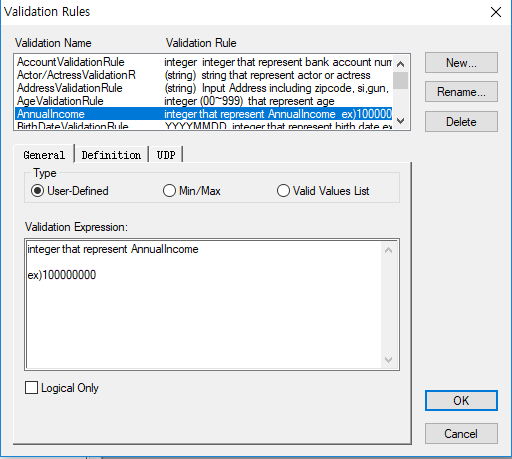
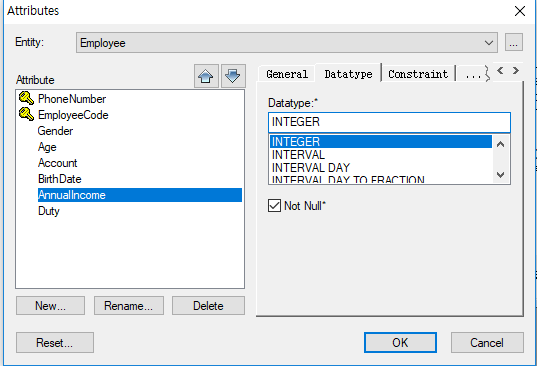
* BirthDate attribute

직원의 생년월일에 관한 정보로 integer형식으로 이루어져 있고 YYYYMMDD형식으로 생년월일을 저장한다.



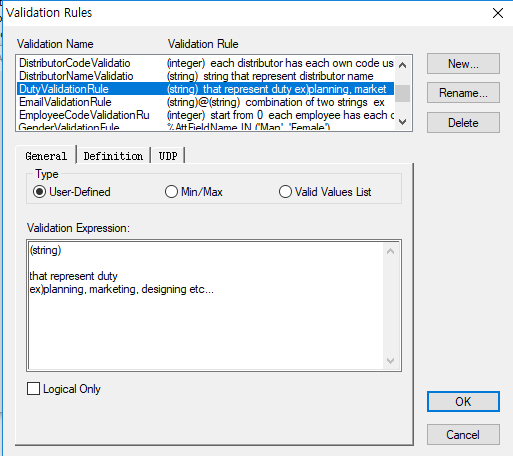
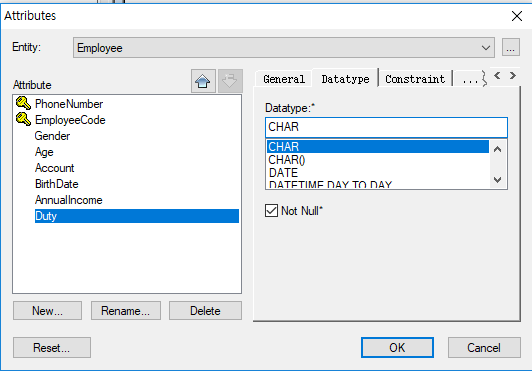
* AnnalIncome attribute

직원의 연봉에 관한 정보로 integer형식으로 이루어져 있고 연봉 금액을 저장한다.

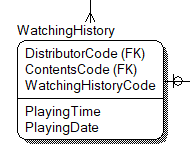


* Duty attribute

직원의 담당 업무에 관한 정보로 char형식으로 이루어져 있고 담당 업무를 저장한다.



4) WatchingHistory entity

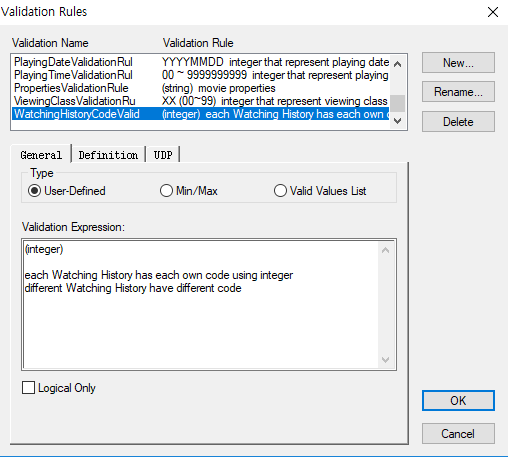
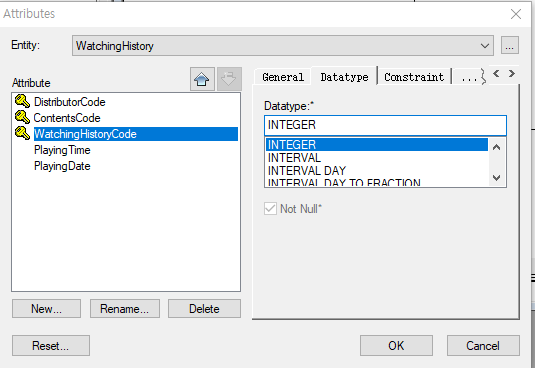


- DistributorCode, ContentsCode (foreign key)

WatchingHistory entity가 contents entity와 relation이 되어있고 그 contents entity는 distributor entity와 relation이 되어 있으므로 contets와 distributor의 primary key가 watchinghistory의 foreign key가 된다.

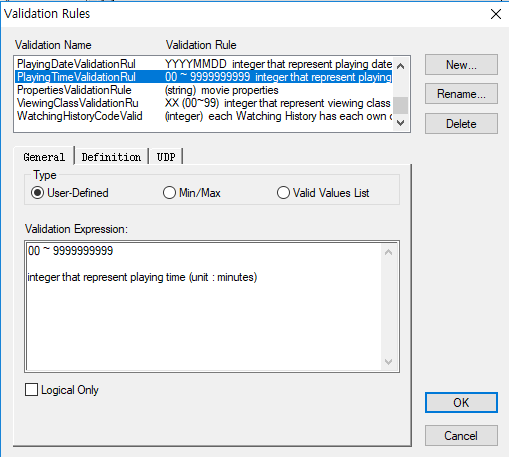
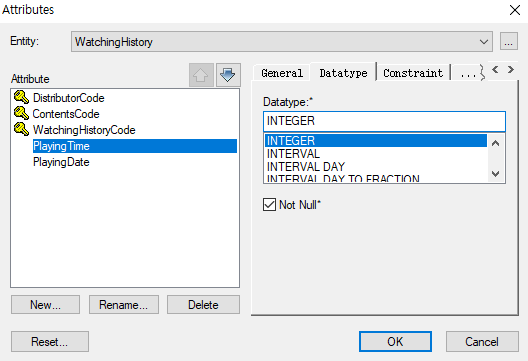
- ViewingHistoryCode attribute (primary key)

고객의 시청내역의 코드에 관한 정보로 integer형식으로 이루어져 있고 이는 시청 내역마다 integer형식의 고유 번호를 부여 받는다고 가정을 했기 때문에 primary key의 기능을 수행할 수 있다.



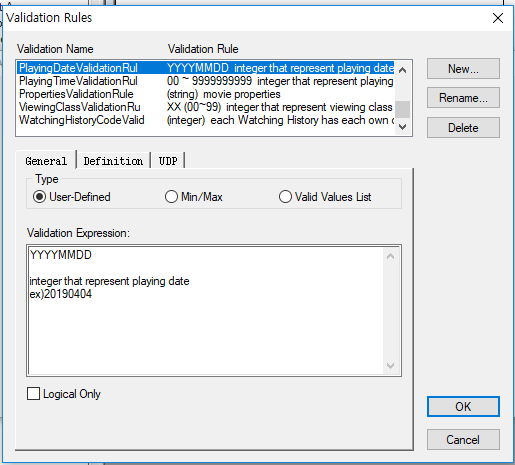
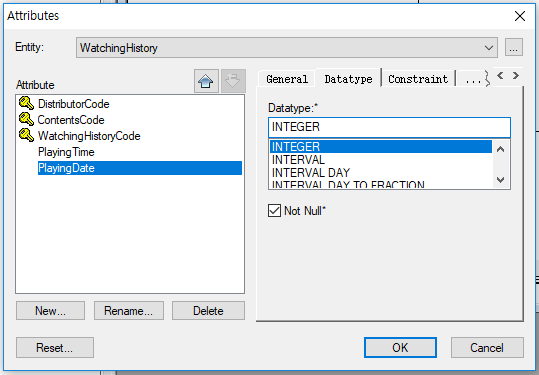
* PlayingTime attribute

컨텐츠를 얼마나 시청을 했는지에 관한 정보로 integer형식으로 이루어져 있고 단위는 분으로 integer를 저장한다.

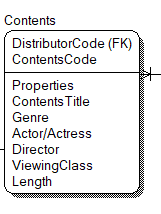


* 시청 일시 attribute

언제 시청을 했는지를 저장해주는 정보로 integer형식으로 이루어져 있고 YYYYMMDD형식의 integer가 저장이 된다.



5) Contents entity

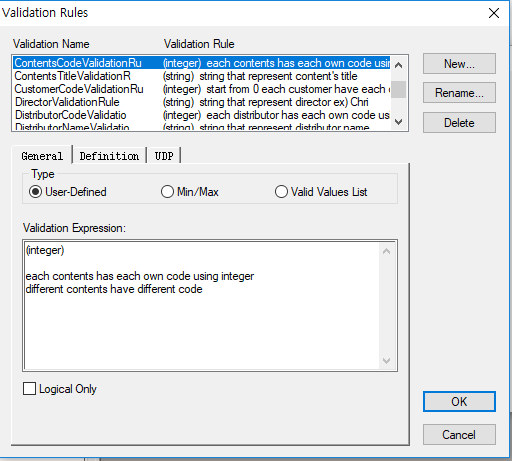
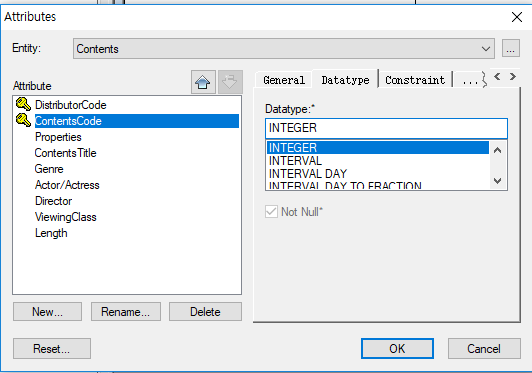


- DistributorCode (foreign key)

Contents entity가 distributor entity와 relation이 되어 있으므로 distributor entity의 primary key인 DistributorCode가 contents entity의 foreign key가 된다.

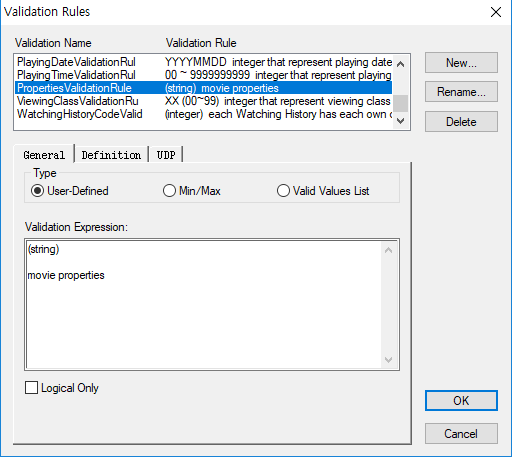
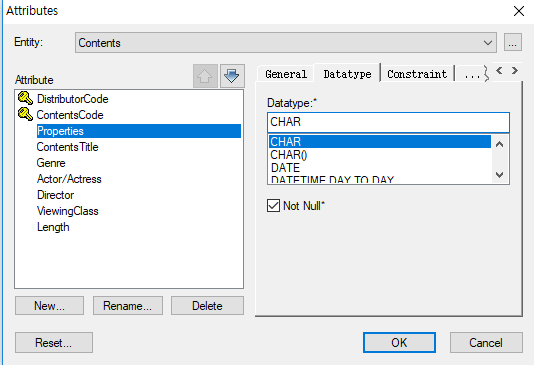
- ContentsCode attribute (primary key)

컨텐츠의 코드에 관한 정보로 integer형식으로 이루어져 있고 컨텐츠 하나당 integer형식의 고유 코드를 부여 받아서 primary key의 기능을 수행할 수 있다.



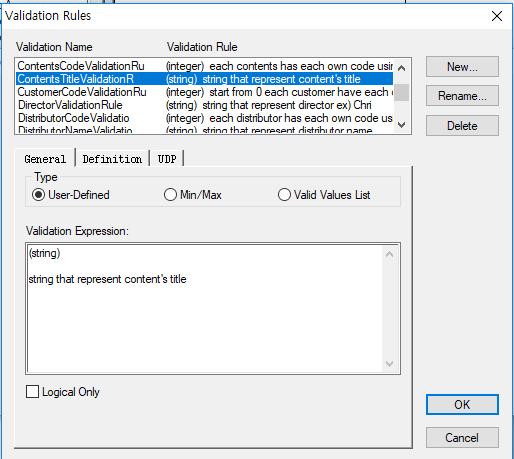
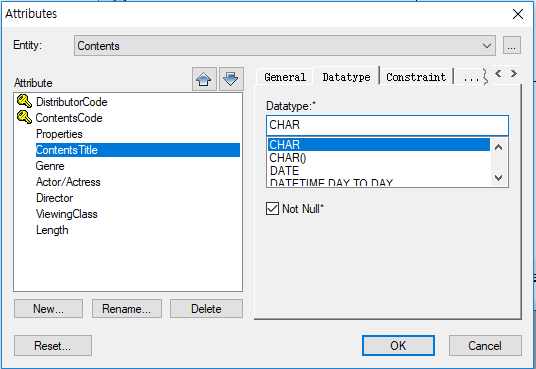
* Properties attribute

영화 특징에 관한 정보로 char형식으로 이루어져 있고 영화의 특징을 저장한다.



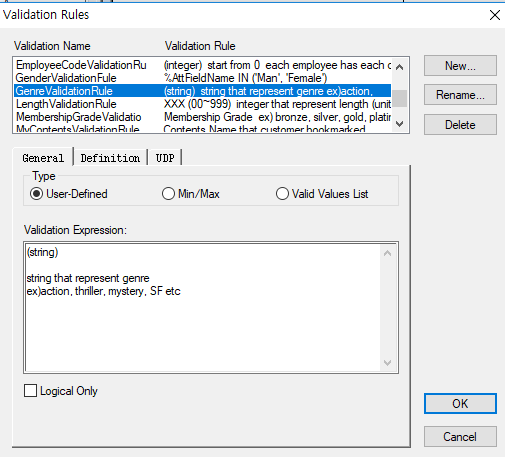
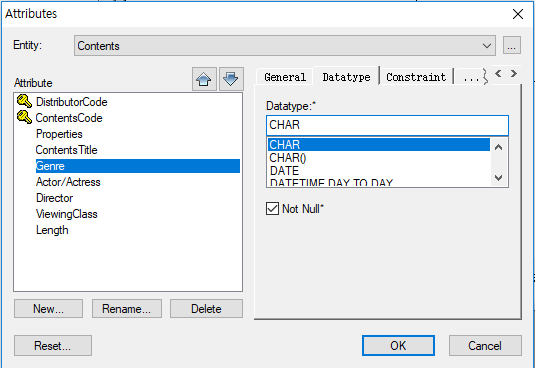
* ContentsTitle attribute

영화 제목에 관한 정보로 char형식으로 이루어져 있고 영화의 제목을 저장한다.



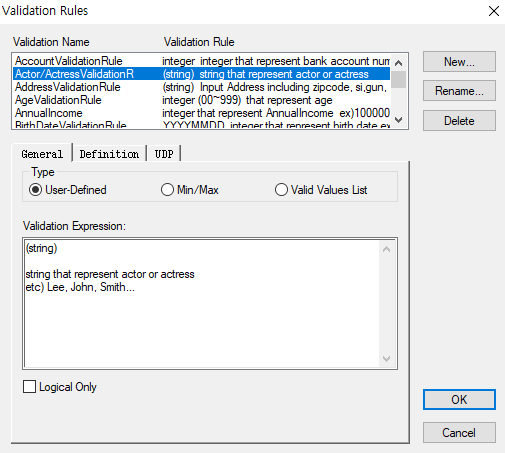
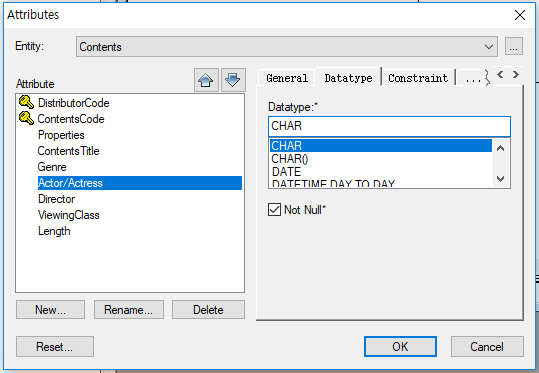
* Genre attribute

영화 장르에 관한 정보로 char형식으로 이루어져 있고 영화의 장르를 저장한다. 한 영화에 여러 장르가 있을 수 있으므로 여러 개를 입력할 수 있다.



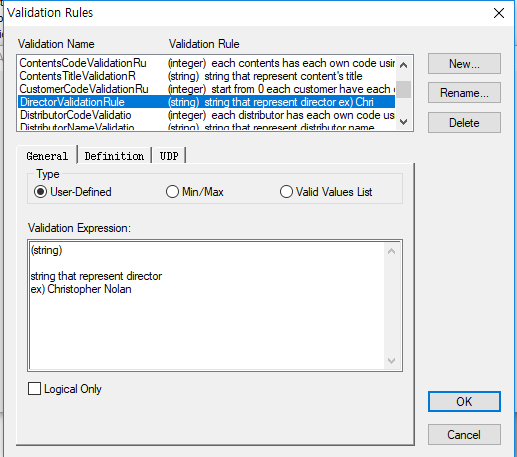
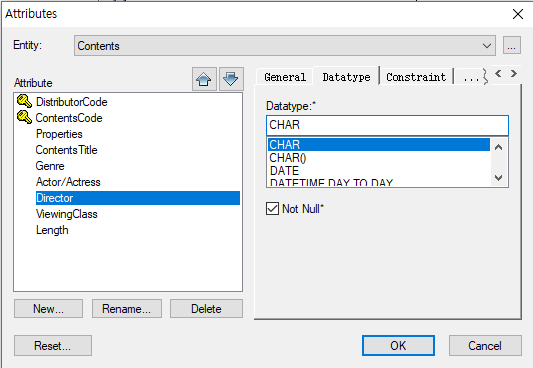
* Actor/Actress attribute

영화에 출연하는 사람들에 관한 정보로 char형식으로 이루어져 있고 어떠한 배우가 출연을 하는지 배우 이름을 저장을 한다. 한 컨텐츠에 여러 명의 배우가 출연할 수 있으므로 여러 개를 입력할 수 있다.



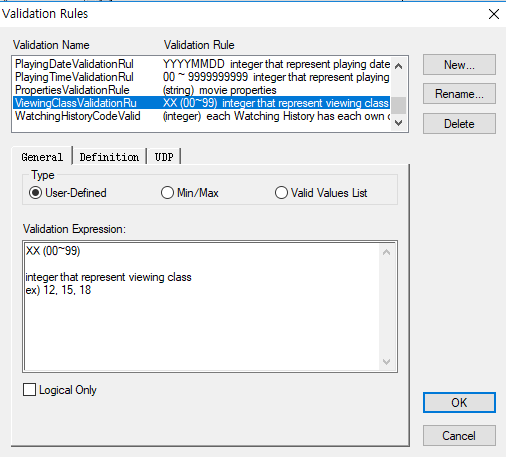
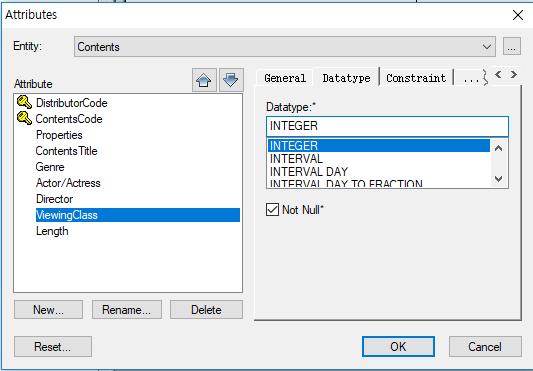
* Director attribute

컨텐츠 감독에 관한 정보로 char 형식으로 이루어져 있고 감독의 이름을 저장한다.



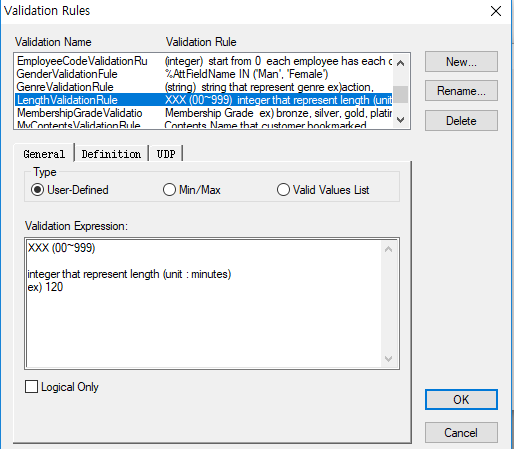
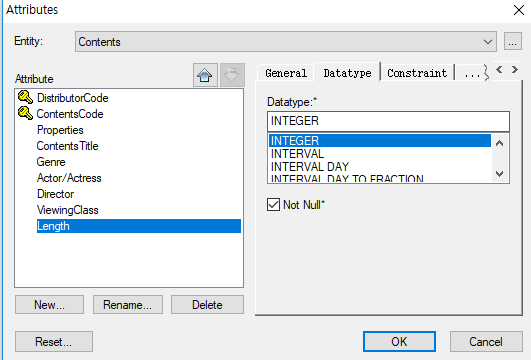
* ViewingClass attribute

컨텐츠의 관람등급에 관한 정보로 integer형식으로 이루어져 있고 몇 세 관람가인지를 저장을 할 수 있도록 integer로 저장을 한다.



* Length attribute

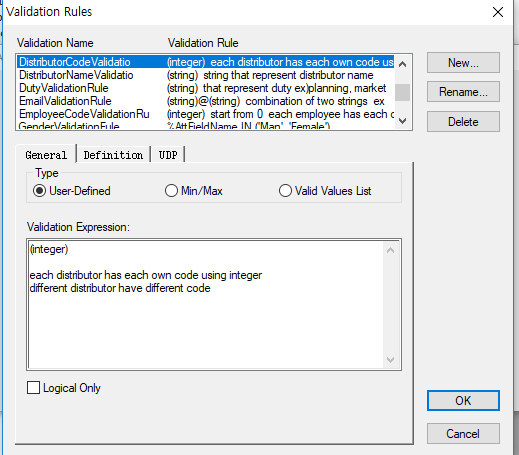
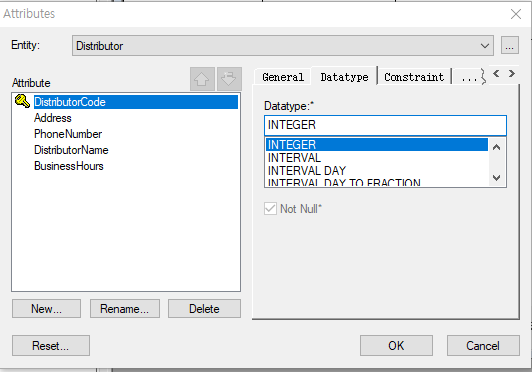
컨텐츠의 총 길이에 관한 정보로 integer형식으로 이루어져 있고 단위는 분으로 integer를 저장한다.



6) Distributor entity

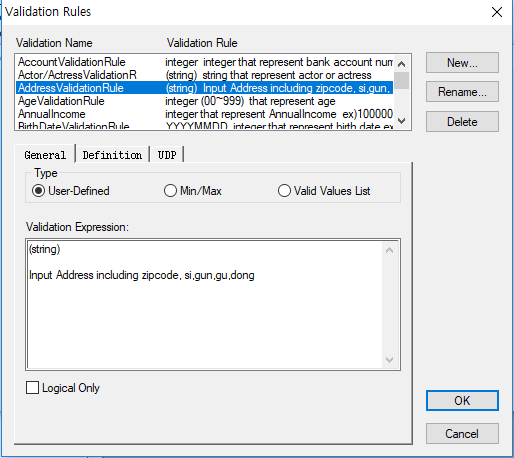
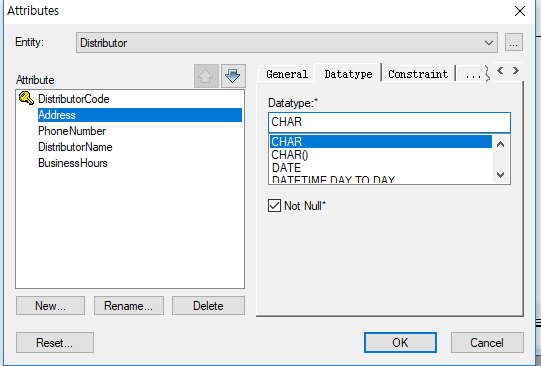
- DistributorCode attribute (primary key)

배급사의 코드에 관한 정보로 하나의 배급사 마다 integer형식의 고유 번호를 부여받는다고 가정을 하고 이는 primary key가 될 수 있다. integer형식으로 배급사의 코드를 저장한다.



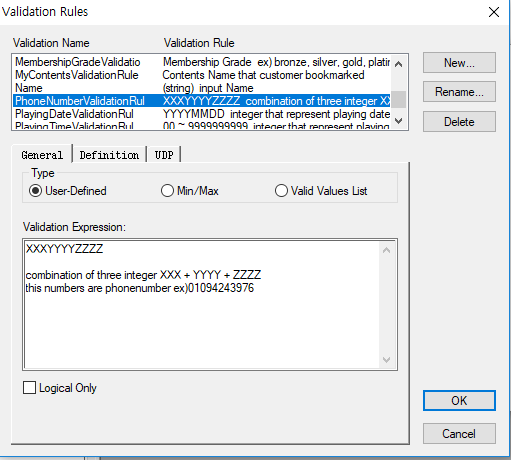
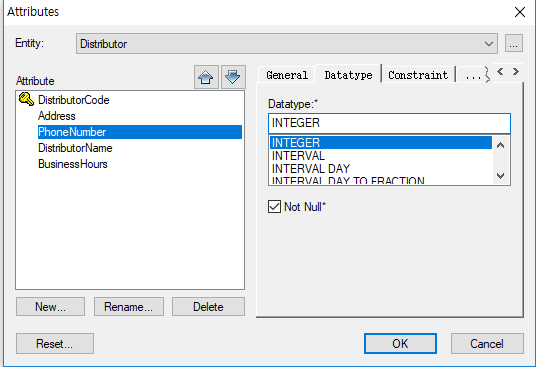
* Address attribute

배급사 주소에 관한 정보로 char형식으로 이루어져 있고 이 주소는 시, 군, 구, 동, 우편번호 정보로 이루어져 있다. 배급사가 여러 군데 있을 수 있으므로 여러 개를 입력할 수 있다.



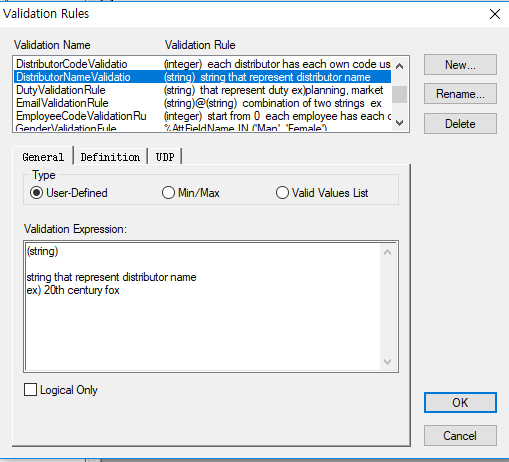
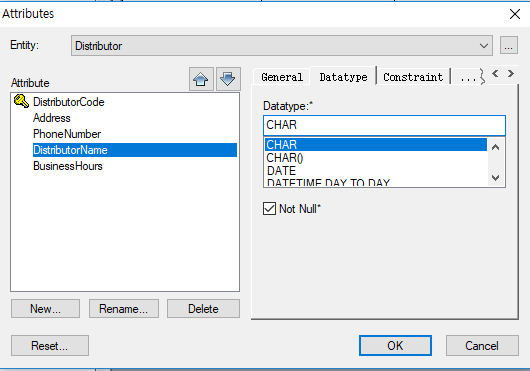
* PhoneNumber attribute

배급사 전화번호에 관한 정보로 integer형식으로 이루어져있고 배급사마다 핸드폰 번호는 하나로 가정을 했고 이는 primary key가 될 수 있다. XXXYYYYZZZZ형식으로 전화번호가 이루어져 있다고 판단했다. 여러 군데에 배급사가 있을 수 있고 담당 부서 마다 전화번호가 다를 수 있으므로 여러 개의 값을 입력할 수 있다.



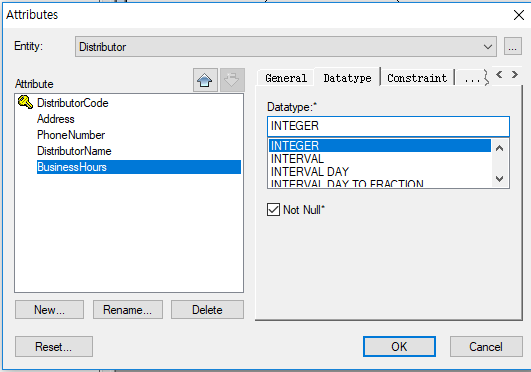
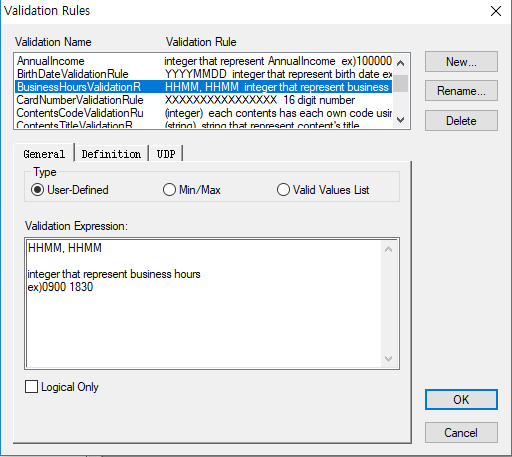
* DistributorName attribute

배급사의 이름에 관한 정보로 char형식으로 이루어져 있고 배급사의 이름을 저장한다.



* BusinessHours attribute

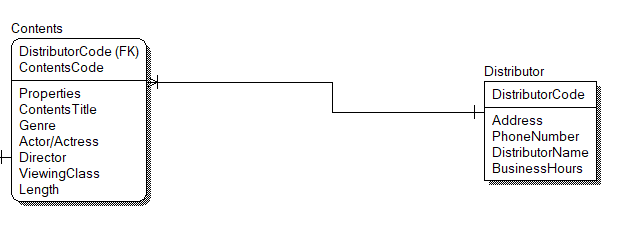
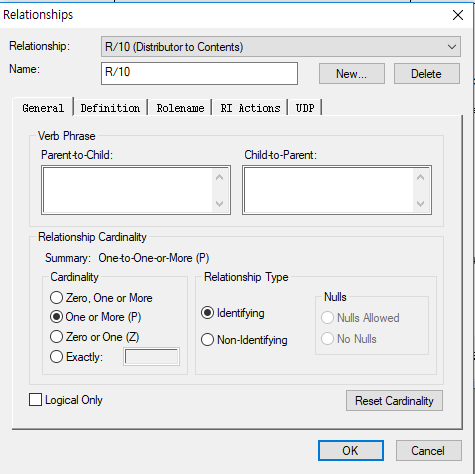
배급사의 영업시간에 관한 정보로 integer형식으로 이루어져 있고 HHMM, HHMM형식으로 영업시간을 입력할 수 있다.

.

**- Mapping Cardinality**

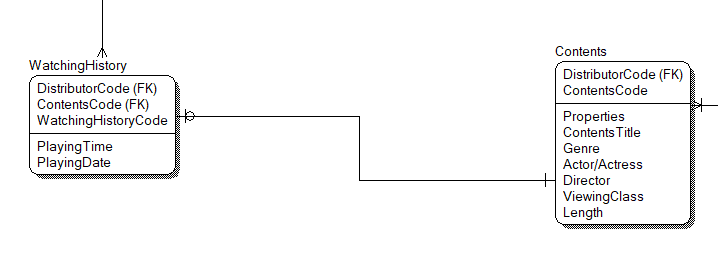
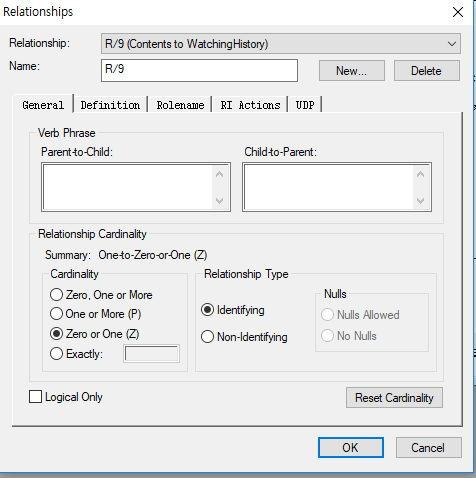
1) Contents와 Distributor의 Mapping Cardinality

컨텐츠와 배급사는 N:1의 관계로 하나의 배급사에는 여러 개의 컨텐츠가 있을 수 있고 하나의 컨텐츠는 하나의 배급사로부터 나오기 때문이다.



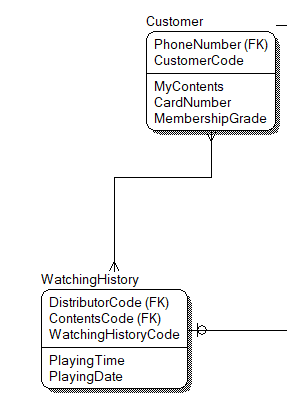
2) Contents와 ViewingHistory의 Mapping Cardinality

컨텐츠와 시청내역은 1:1(or zero)의 관계로 하나의 컨텐츠는 하나의 시청내역이 고객이 보는 거에 따라 있을 수 있고 없을 수 있고 하나의 시청내역에는 하나의 컨텐츠가 있기 때문이다.



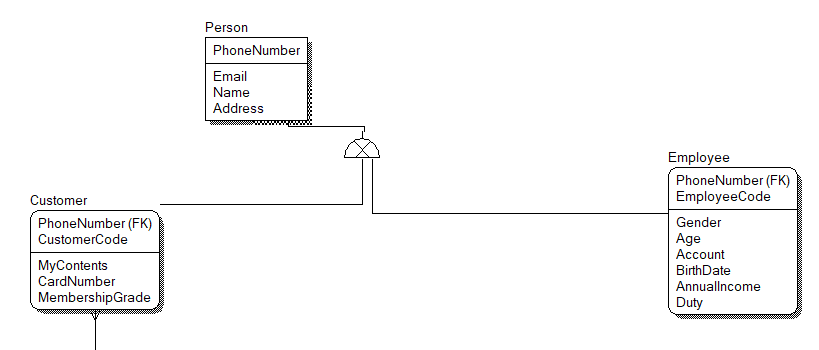
3) Customer와 ViewingHistory의 Mapping Cardinality

회원 정보와 시청내역은 M:N의 관계로 하나의 회원은 여러 개의 시청을 하기 때문에 여러 개의 시청내역을 가질 수 있고 하나의 시청내역에는 여러 명의 회원이 볼 수 있기 때문에 M:N의 관계를 갖는다.



4) Person Entity와 Customer/Employee의 mapping cardinality

고객과 직원은 사람의 상위 class로 묶을 수 있으므로 이는 ISA를 통해서 연결할 수 있다.



**5. 효율성**

1) 직원과 회원의 정보는 사람이라는 상위 entity로 묶을 수 있고 사람 entity를 통해 직원과 회원 사이의 공통된 정보를 한번에 처리할 수 있고 직원과 회원에서 여러 attribute를 중복되도록 저장을 할 필요가 없기 때문에 효율성을 높일 수 있었다.

2) 요구사항에서 보면 컨텐츠 entity에는 배급사 attribute가 있을 수 있다. 그런데 실제로 er win을 모델링 할 때 배급사 attribute를 설정하지 않았다. 왜냐하면 컨텐츠와 배급사를 relation을 통해서 이어주었기 때문에 배급사 코드를 이용해서 배급사의 데이터에 접근을 할 수 있기 때문이다. 컨텐츠 entity에서 배급사코드가 foreign key이기 때문에 이 foreign key로 배급사 entity를 접근할 수 있다. 그래서 컨텐츠 entity에 배급사 attribute를 하나 더 입력을 안 해줘도 배급사 데이터에 접근을 할 수 있기 때문에 효율적이다.

3) 요구사항에서 보면 시청내역 entity에서 컨텐츠 정보의 attribute가 있을 수 있다. 그런데 실제로 er win을 모델링 할 때 컨텐츠 정보의 attribute를 설정하지 않았다. 왜냐하면 시청내역과 컨텐츠 사이를 relation을 통해서 이어주었기 때문에 컨텐츠 코드를 이용해서 컨텐츠의 데이터에 접근을 할 수 있기 때문이다. 시청내역에서 컨텐츠 코드가 foreign key이기 때문에 이 foreign key로 컨텐츠를 접근할 수 있다. 그래서 컨텐츠 정보의 attribute를 하나 더 안 써줘도 충분히 접근이 가능하기 때문에 효율적이라 할 수 있다. 그리고 시청내역에서 배급사 코드까지 알 수 있고 컨텐츠 정보와 배급사 정보까지 한번에 알 수 있어서 한번의 접근으로 컨텐츠와 배급사의 정보까지 얻을 수 있어서 효율적이다.

4) 요구사항에서 배급사에서 컨텐츠 리스트의 attribute가 있을 수 있다. 그런데 실제로 모델링 할 때는 배급사에서 컨텐츠 리스트를 설정하지 않았다. 왜냐하면 배급사와 컨텐츠는 “제공”이라는 relation을 통해서 1:N관계로 되어있어서 컨텐츠에 대한 속성을 또 안 만들어주고 관계를 통해 접근을 할 수 있어서 중복된 저장을 방지할 수 있기 때문에 효율적이다.

5) 요구사항에서 시청내역에서 고객에 대한 정보가 attribute가 될 수 있다. 그런데 실제로 모델링 할 때는 시청내역에서 고객에 대한 attribute를 설정하지 않았다. 왜냐하면 시청내역과 고객은 N:M의 “시청”이라는 relation을 통해서 관계되어 있어서 고객이라는 속성을 안 만들어주고 관계를 통해 접근을 해서 중복된 저장을 방지할 수 있기 때문에 효율적이다.