Homework1 보고서

ER Diagram Design

|  |  |
| --- | --- |
| **과목명** | 데이터베이스시스템및응용 |
| **담당 교수님** | 차재혁 교수 |
| **제출일** | 2021년 9월 30일(목요일) |
| **소속** | 한양대학교 공과대학 |
| 컴퓨터소프트웨어학부 |
| **학번** | **이름** |
| 2019009261 | 최가온(CHOI GA ON) |

Ⅰ. Entity Set

1. **Sellers** – 판매자(가게)

|  |  |  |
| --- | --- | --- |
| **Attributes** | **데이터 타입** | **설명** |
| seller\_id | serial4 | Seller entity를 식별하기 위한 고유 ID이다. |
| owner\_id | serial4 | 가게는 반드시 1명의 점주가 있어야 한다.  점주의 ID를 의미한다. |
| address | character(10) | 가게의 우편번호(5자리)를 의미한다.  정수로 표현 시, 0으로 시작하는 경우에 대해서는 부적절하므로 character로 설정하였다. |
| brand\_name | character(100) | 가게의 상호명을 의미한다. |
| open\_time | time[(p)][without timezone] | 가게의 개장시간을 의미한다. |
| close\_time | time[(p)][without timezone] | 가게의 폐장시간을 의미한다. |
| build\_date | timestamp[(p)]with timezone | 가게의 개업일/시각을 의미한다. |
| open\_days | Boolean[7] | 가게가 영업하는 요일을 의미한다.  월요일부터 일요일까지 총 7개의 Boolean 배열로 구성하였다. |
| location\_city | character(20) | 가게가 속하는 “시”를 의미한다. |
| location\_county | character(20) | 가게가 속하는 “군”을 의미한다. |
| location\_district | character(20) | 가게가 속하는 “구”를 의미한다. |
| type | character(20) | 가게의 업종을 의미한다. |
| latitude | integer | 가게 위치의 위도를 의미한다. |
| longitude | integer | 가게 위치의 경도를 의미한다. |

2. **Owners** – 점주

점주는 하나 이상의 가게를 소유할 수 있다. 초기에는 점주에 대한 정보를 판매자 entity의 attribute로 넣으려고 하였으나, 1개 이상의 가게를 소유할 가능성도 있다는 점에 착안하여 점주는 별도의 entity set로 구성하였다.

|  |  |  |
| --- | --- | --- |
| **Attributes** | **데이터 타입** | **설명** |
| owner\_id | serial4 | Owner entity를 식별하기 위한 고유 ID이다. |
| id | character(100) | HYDelivery 앱에 접속하기 위한 ID이다. |
| e-mail | character(100) | HYDelivery 앱에 접속하기 위한 이메일이다. |
| password | character(100) | HYDelivery 앱에 접속하기 위한 패스워드이다. |

3. **Services** – 서비스

하나의 가게는 여러 가지의 서비스들을 제공할 수 있다. 하나의 서비스가 아니라, 다수의 서비스들을 소유하는 관계에 있어야 하므로, 서비스를 별도의 entity set로 구성하였다.

|  |  |  |
| --- | --- | --- |
| **Attributes** | **데이터 타입** | **설명** |
| service\_id | serial4 | Services entity를 식별하기 위한 고유 ID이다. |
| name | character(50) | 서비스의 이름을 의미한다. |
| price | integer | 서비스의 가격을 의미한다. |
| discount\_rate | decimal | 서비스의 할인율을 의미한다. 0부터 100까지의 정수를 기본 전제로 하였다. |
| seller\_id | serial4 | 소속된 Seller의 ID이다. |

4. **Tags** – 태그

판매자의 서비스는 구매자가 검색할 수 있는 다수의 태그를 가지고 있다. 이를 위해 태그 정보를 하나의 Entity Set로 별도로 구성하게 되었다.

|  |  |  |
| --- | --- | --- |
| **Attributes** | **데이터 타입** | **설명** |
| tag\_id | serial4 | Tags entity를 식별하기 위한 고유 ID이다. |
| data | character(100) | 태그를 구성하는 데이터를 의미한다. |
| service\_id | serial4 | 태그가 속한 서비스의 ID이다. |

5. **Buyers** – 구매자

|  |  |  |
| --- | --- | --- |
| **Attributes** | **데이터 타입** | **설명** |
| buyer\_id | serial4 | Buyers entity를 식별하기 위한 고유 ID이다. |
| phone\_number | character(50) | 구매자의 전화번호를 의미한다. (🡪 1개) |
| id | character(100) | HYDelivery 앱에 접속하기 위한 ID이다. |
| e-mail | character(100) | HYDelivery 앱에 접속하기 위한 이메일이다. |
| password | character(100) | HYDelivery 앱에 접속하기 위한 패스워드이다. |

6. **Payments** – 결제수단

한 명의 구매자는 여러 개의 결제수단을 가질 수 있다. 이를 위해 결제수단을 하나의 Entity Set로 별도로 구성하게 되었다.

|  |  |  |
| --- | --- | --- |
| **Attributes** | **데이터 타입** | **설명** |
| payment\_id | serial4 | Payments entity를 식별하기 위한 고유 ID이다. |
| type | integer | 결제수단의 타입을 의미한다.  결제수단의 타입은 총 3종류이다.  1 – 카드, 2 – 계좌, 3 – 포인트 |
| buyer\_id | character(100) | 결제수단이 소속된 구매자의 ID를 의미한다. |
| address | character(100) | 결제수단의 계좌번호 또는 카드번호 또는 포인트 잔액을 의미한다. |

7. **Destinations** – 배송지

한 명의 구매자는 여러 개의 배송지를 가질 수 있다. 이를 구현하기 위해 배송지를 하나의 Entity Set로 구성하였다.

|  |  |  |
| --- | --- | --- |
| **Attributes** | **데이터 타입** | **설명** |
| dest\_id | serial4 | Payments entity를 식별하기 위한 고유 ID이다. |
| name | character(100) | 목적지의 이름을 의미한다. |
| favorite | boolean | 목적지의 선호여부를 의미한다. |
| location\_city | character(20) | 목적지가 속하는 “시”를 의미한다. |
| location\_county | character(20) | 목적지가 속하는 “군”을 의미한다. |
| location\_district | character(20) | 목적지가 속하는 “구”를 의미한다. |
| buyer\_id | serial4 | 목적지가 소속된 구매자의 ID를 의미한다. |

8. **Delivery Agents** – 배달 대행자

|  |  |  |
| --- | --- | --- |
| **Attributes** | **데이터 타입** | **설명** |
| agent\_id | serial4 | Delivery Agents entity를 식별하기 위한 고유 ID이다. |
| id | character(100) | HYDelivery 앱에 접속하기 위한 ID이다. |
| e-mail | character(100) | HYDelivery 앱에 접속하기 위한 이메일이다. |
| password | character(100) | HYDelivery 앱에 접속하기 위한 패스워드이다. |
| latitude | integer | 배달 대행자의 위도를 의미한다. |
| longitude | integer | 배달 대행자의 경도를 의미한다. |
| timestamp | timestamp[(p)]with timezone | 가장 최근에 위치가 파악된 시간을 의미한다. |
| active\_point | character(100) | 활동지점을 우편번호 형식으로 나타낸 것을 의미한다. |
| is\_available | boolean | 배송 가능 여부를 의미한다. |
| left\_service | integer | 현재 남아있는 배달 건수를 의미한다. |
| delivery\_fee | integer | 배송 수수료에 대한 정보를 의미한다. |

9. **Contacts** – 연락처

배달 대행자는 여러 개의 연락처를 소유할 수 있다. 이를 위해 배달 대행자의 하나의 attribute로 연락처를 설정하는 것이 아닌, 새롭게 하나의 연락처 Entity Set를 구성하였다.

|  |  |  |
| --- | --- | --- |
| **Attributes** | **데이터 타입** | **설명** |
| con\_id | serial4 | Contacts entity를 식별하기 위한 고유 ID이다. |
| own\_id | serial4 | 연락처가 소속된 배달 대행자의 ID이다. |
| phone\_number | character(50) | 연락처 데이터이다. |

10. **Accounts** – 계정

배달의한양 앱에 접속할 수 있는 대상은 판매자, 구매자, 배달대행자 3그룹이다. 초기에 이를 디자인할 때에는 각각의 attribute로 구성을 하려고 하였다. 그러나, 배달앱 전체에서 이 계정들을 하나로 통합하여 관리하는 것이 소프트웨어의 유지보수 관련 측면에서 더 효율적일 것이라고 판단하게 되었다.

아이디는 겹칠 수 없으므로 이를 이용하여 계정 Entity Set를 독립적으로 구성하였다.

|  |  |  |
| --- | --- | --- |
| **Attributes** | **데이터 타입** | **설명** |
| acc\_id | serial4 | Accounts entity를 식별하기 위한 고유 ID이다. |
| id | character(100) | HYDelivery 앱에 접속하기 위한 ID이다. |
| e-mail | character(100) | HYDelivery 앱에 접속하기 위한 이메일이다. |
| password | character(100) | HYDelivery 앱에 접속하기 위한 패스워드이다. |

11. **Orders** – 주문

배달의한양 앱 E-R 다이어그램에서 핵심적인 Entity Set에 해당한다.

|  |  |  |
| --- | --- | --- |
| **Attributes** | **데이터 타입** | **설명** |
| order\_id | serial4 | Orders entity를 식별하기 위한 고유 ID이다. |
| buyer\_id | serial4 | 주문을 한 구매자의 ID이다. |
| seller\_id | serial4 | 주문을 받은 판매자의 ID이다. |
| service\_id | serial4 | 주문받은 서비스의 ID이다. |
| deliver\_id | serial4 | 주문을 처리할 배달대행자의 ID이다. |
| number | integer | 주문을 받은 서비스의 수량이다. |
| pay\_type | integer | 결제 수단이다.  1 – 카드, 2 – 계좌, 3 – 포인트 |
| dest\_id | serial4 | 주문이 배달될 목적지의 ID이다. |
| buy\_time | timestamp[(p)] with timezone | 주문이 시작된 시각이다. |
| estimated\_time | interval [fields] [(p)] | 주문에 소요되는 예상시간이다. |

Ⅱ. Relations

배달의한양 데이터베이스 구조를 디자인하며 고려한 사항은 아래와 같다.

- 만약 A에 속한 요소 B가 여러 개가 존재해야 하는 상황이라면, B를 A의 attribute가 아닌, 별도의 Entity Set으로 정의하는 것이 적절한 디자인일 것이다.

- Entity Set으로 연결한다면 어떤 Entity Set의 attributes로 구성할 때보다 자유도가 커진다. 하나의 독립적인 개체로 작동할 수 있기 때문에 다수의 또 다른 Entity Set과도 Relations를 정의할 수 있다는 점도 있다.

- Entity Set으로 연결한다면 주로 “A have B”의 구조를 갖게 되는데, 이때 B가 A와의 관계에 대해 Weak Entity인지를 확인해야 한다.

- 검색 기능과 관련하여 이후에 SQL문 등으로 Query를 작성할 때 간결한 쿼리문으로 원하는 검색 기능을 구현할지에 대해 고민하였다.

- 관계(Relation)의 구조를 사용해야 한다면, 일대일/일대다/다대일/다대다 구조 중 어떤 것이 적절한 선택인지를 항상 고민하였다.

1.

하나의 점주는 여러 개의 가게를 소유할 수 있다. 🡪 일대다 관계



2.

하나의 가게는 여러 개의 서비스를 제공할 수 있다. 🡪 다대다 관계

만약 가게가 사라진다면, 그 가게가 제공했던 서비스에 대한 정보도 모두 사라지게 되는 구조이다. 따라서 서비스를 Weak Entity Set으로간주하였다.



3.

하나의 서비스에 대해 여러 개의 태그가 존재할 수 있다. 🡪 다대다 관계

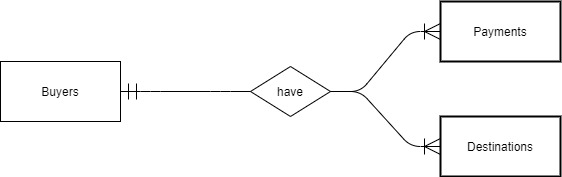
마찬가지로 서비스가 사라진다면, 그 서비스에 연결되어 있던 모든 태그 정보도 사라져야 한다. 따라서 태그를 Weak Entity Set으로 간주하였다.



4.

하나의 구매자는 여러 개의 결제 수단을 가질 수 있다. 🡪 다대다 관계

하나의 구매자는 여러 개의 배송지를 설정할 수 있다. 🡪 다대다 관계



5.

하나의 배달대행자는 여러 개의 연락처 정보를 가질 수 있다. 🡪 다대다 관계

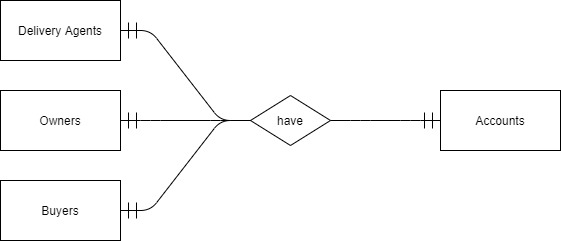


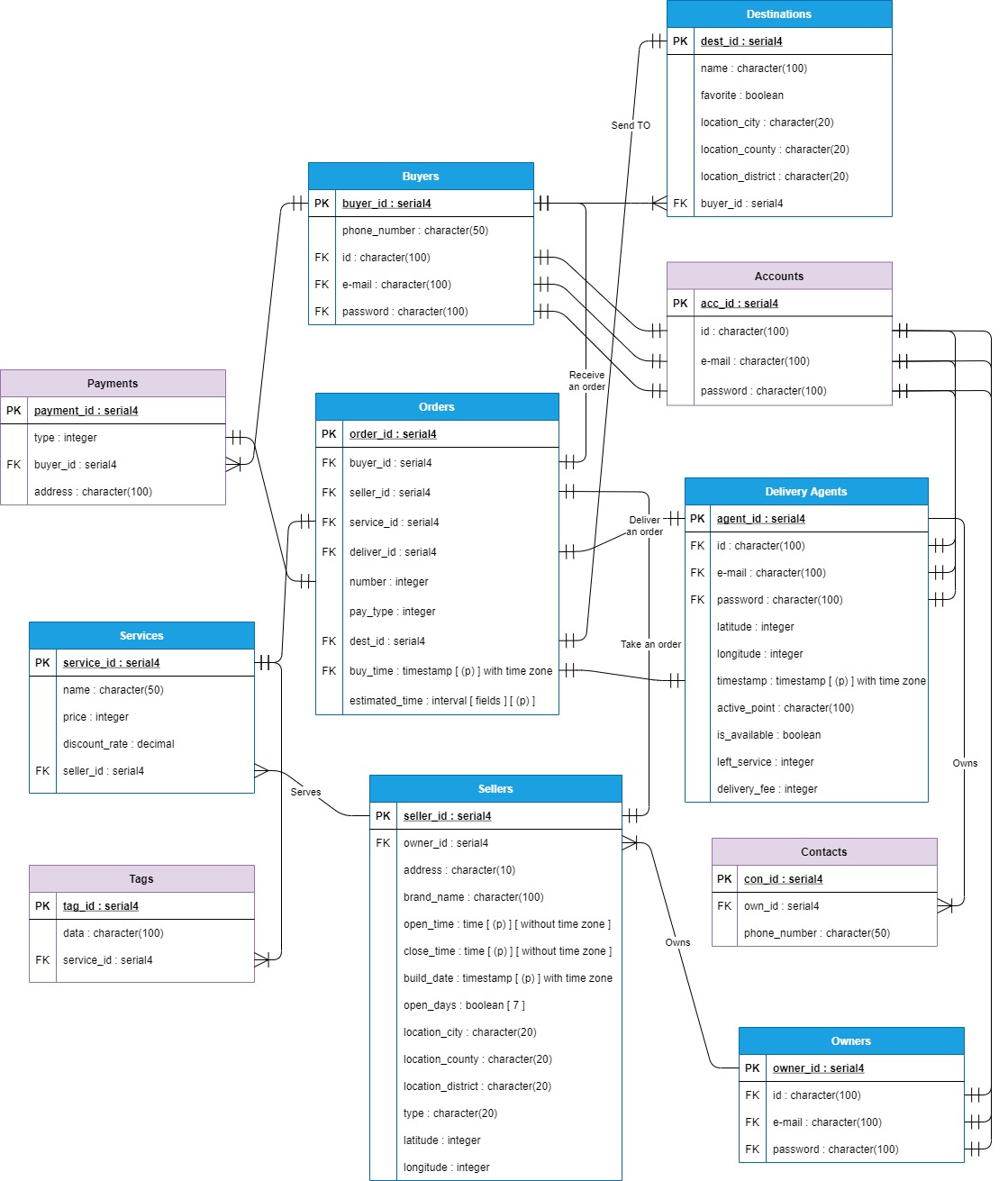
6.

하나의 구매자는 하나의 계정정보를 가지고 앱에 로그인할 수 있다. 🡪 일대일 관계

하나의 점주는 하나의 계정정보를 가지고 앱에 로그인할 수 있다. 🡪 일대일 관계

하나의 배달 대행자는 하나의 계정정보를 가지고 앱에 로그인할 수 있다. 🡪 일대일 관계



ER DIAGRAM for HYDelivery APP