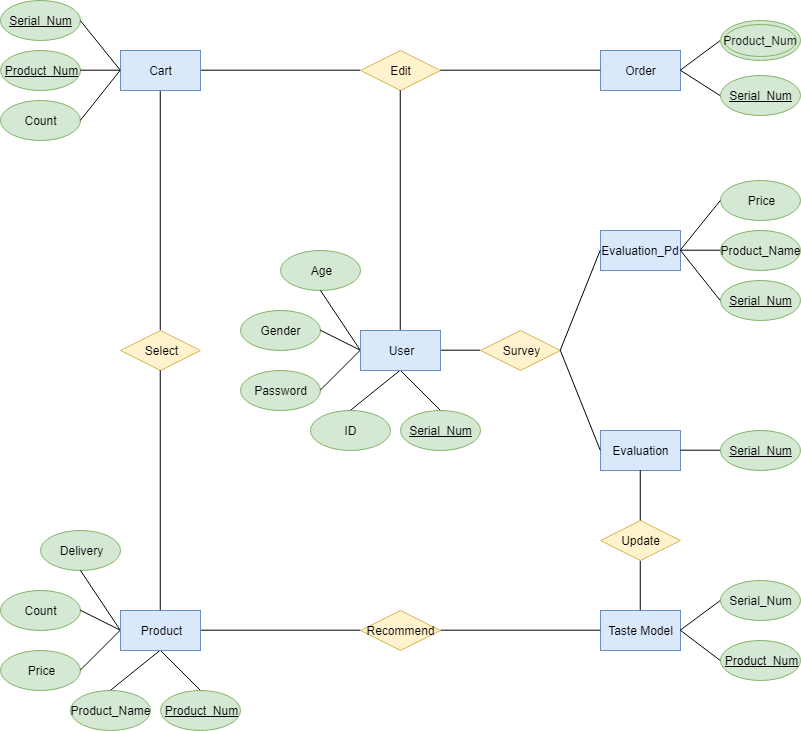
**1. Database Design**

**1.1 Objectives**

Database Design은 요구사항 명세서에서 기술한 데이터베이스 요구사항을 바탕으로 작성하였다. 요구사항을 바탕으로 ER Diagram을 작성하고, 이를 통해 Relational Schema와 SQL DDL을 작성한다.

**1.2 ER Diagram**

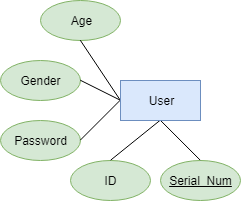
ER Diagram에 대한 설명은 다음과 같다. Entity는 직사각형으로 표현하며, Entity들 간의 관계를 나타내는 Relation은 마름모 모양으로 표현한다. 또한, Entity와 Relation은 Attribute를 가질 수 있는데, 이는 타원형으로 표현한다. 마지막으로 각 Entity의 Primary key는 아래에 밑줄을 치고, Multi-valued Attribute는 이중 타원형으로 표현한다.



**Diagram 1. Overall ER Diagram**

* + 1. **Entity**

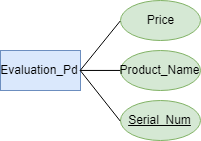
**A.1 User**

****

**Figure 1. User Entity**

User는 회원의 정보를 나타낸다. Primary key는 Serial number(Serial\_Num)이고, Serial\_Num, Age, Gender, Password, ID의 속성을 가지고 있다.

**A.2 Evaluation\_Pd**

****

**Figure 2. Evaluation\_Pd Entity**

Evaluation\_Pd는 사용자가 처음 로그인 했을 때 진행하는 평가에 사용되는 상품 정보이다. Primary key는 Serial number(Serial\_Num)이고, Serial\_Num, Price, Product\_Name의 속성을 가지고 있다.

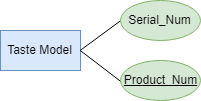
**A.3 Evaluation**

****

**Figure 3. Evaluation Entity**

Evaluation은 사용자가 첫 로그인 시에 진행되는 평가의 데이터를 갖고 있다. Primary key는 Serial number(Serial\_Num)이고, Serial\_Num의 속성을 가지고 있다.

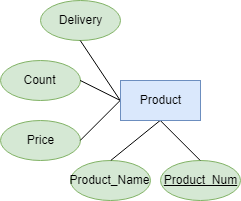
**A.4 Taste Model**

****

**Figure 4. Taste Model Entity**

Taste Model은 사용자의 취향 데이터를 가지고 있다. Primary key는 Product number(Product\_Num)이고, Product\_Num, Serial\_Num의 속성을 가지고 있다.

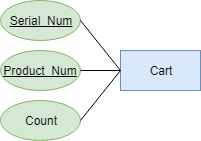
**A.5 Product**

****

**Figure 5. Product Entity**

Product은 상품의 정보를 나타낸다. Primary key는 Product number(Product\_Num)이고, Product\_Num, Product\_Name, Price, Count, Delivery의 속성을 가지고 있다.

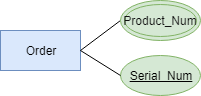
**A.6 Cart**

****

**Figure 6. Cart Entity**

Cart는 사용자가 선택한 상품이 들어간 장바구니를 나타낸다. Primary key는 Serial number(Serial\_Num)와 Product number(Product\_Num)이고, Serial\_Num, Product\_Num, count의 속성을 가지고 있다.

**A.7 Order**

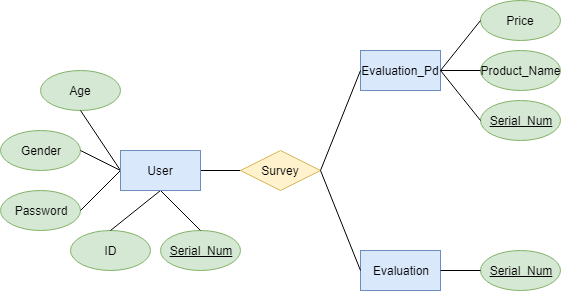
****

**Figure 7. Order Entity**

Order는 사용자에 구매에 대해 나타낸다. Primary key는 Serial number(Serial\_Num)이고, Serial\_Num, Product\_Num의 속성을 가지고 있다.

**B Relationship**

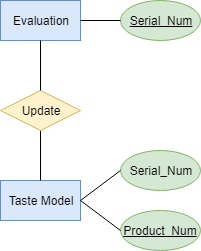
**B.1 Survey**

****

**Figure 8. Survey Relationship**

Survey Relationship은 survey를 통해서 사용자의 취향 데이터를 얻는다. 한 명의 사용자는 하나의 취향을 가진다.

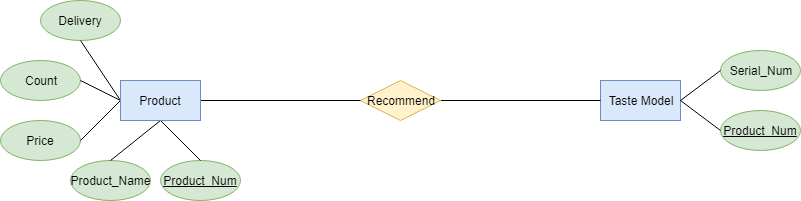
**B.2 Update**

****

**Figure 9. Update Relationship**

Update relationship은 사용자의 취향을 토대로 taste model을 만들기 때문에, 하나의 취향이 하나의 모델을 만든다. 또한, 하나의 모델은 여러 취향에 담길 수도 있다.

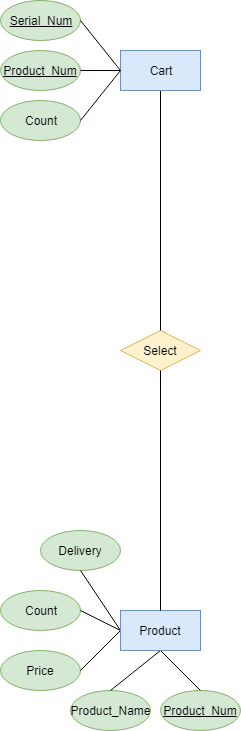
**B.3 Recommend**

****

**Figure 10. Recommend Relationship**

Recommend relationship은 taste model을 바탕으로 사용자의 취향에 맞는 상품을 상품 DB 중에서 추천해주는 관계로, 하나의 모델이 여러 개의 상품을 추천 받을 수 있다.

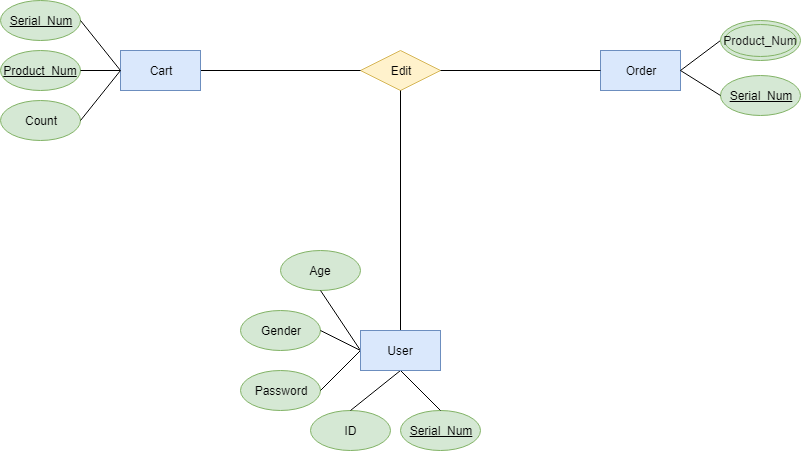
**B.4 Select**



**Figure 11. Select Relationship**

Select relationship은 상품 중에서 선택 받은 것만이 카트에 담기게 되는 관계로, 하나의 상품이 카트에 담길 수도 있고, 또 여러 개의 상품이 카트에 담길 수 있다.

**B.5 Edit**



**Figure 12. Edit Relationship**

Edit relationship은 사용자가 카트를 편집할 수 있으며, 사용자의 선택에 따라 주문을 정할 수 있다. 한 명의 사용자는 자신에 해당된 하나의 카트를 편집할 수 있고, 여러 개의 상품을 주문 및 취소할 수 있다.

**1.3 Relational Schema**

**A. User**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial\_Num | Age | Gender | Password | ID |

Primary Key (PK): Serial\_Num

Foreign Key (FK): 없음

Function Dependency (FD):

Serial\_Num → { ID, Password, Gender, Age };

Description: Entity User에 관한 테이블이다. 모든 속성에 Null을 허용하지 않는다.

* + 1. **Evaluation\_Pd**

|  |  |  |
| --- | --- | --- |
| Serial\_Num | Product\_Name | Price |

Primary Key (PK): Serial\_Num

Foreign Key (FK): Serial\_Num

Function Dependency (FD):

Serial\_Num → { Product\_Name, Price };

Description: Entity Evaluation\_Pd에 관한 테이블이다. 모든 속성에 Null을 허용하지 않으며, Entity User의 PK를 FK로 갖는다.

* + 1. **Evaluation**

|  |
| --- |
| Serial\_Num |

Primary Key (PK): Serial\_Num

Foreign Key (FK): Serial\_Num

Function Dependency (FD):

Serial\_Num → { };

Description: Entity Evaluation에 관한 테이블이다. 모든 속성에 Null을 허용하지 않으며, Entity User의 PK를 FK로 갖는다.

* + 1. **Taste Model**

|  |  |
| --- | --- |
| Product\_Num | Serial\_Num |

Primary Key (PK): Product\_Num

Foreign Key (FK): Product\_Num

Function Dependency (FD):

Product\_Num → { Serial\_Num };

Description: Entity Taste Model에 관한 테이블이다. 모든 속성에 Null을 허용하지 않으며, Entity Product의 PK를 FK로 갖는다.

* + 1. **Product**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product\_Num | Product\_Name | Price | Count | Delivery |

Primary Key (PK): Product\_Num

Foreign Key (FK): 없음

Function Dependency (FD):

Product\_Num → { Product\_Name, Price, Count, Delivery };

Description: Entity Product에 관한 테이블이다. 모든 속성에 Null을 허용하지 않는다.

* + 1. **Cart**

|  |  |  |
| --- | --- | --- |
| Serial\_Num | Product\_Num | Count |

Primary Key (PK): Serial\_Num, Product\_Num

Foreign Key (FK): Serial\_Num, Product\_Num

Function Dependency (FD):

Serial\_Num → { Product\_Num, Count };

Description: Entity Cart에 관한 테이블이다. 모든 속성에 Null을 허용하지 않으며, Entity User의 PK와 Entity Product의 PK를 FK로 갖는다.

* + 1. **Order**

|  |  |
| --- | --- |
| Serial\_Num | Product\_Num |

Primary Key (PK): Serial\_Num

Foreign Key (FK): Serial\_Num

Function Dependency (FD):

Serial\_Num → { Product\_Num };

Description: Entity Order에 관한 테이블이다. 모든 속성에 Null을 허용하지 않으며, Entity User의 PK 를 FK로 갖는다. Product\_Num은 Multi-valued Attribute로 여러 값을 가진다.

**1.4 SQL DDL**

**A. User**

CREATE TABLE User (

‘Serial\_Num’ int NOT NULL AUTO\_INCREMENT,

‘ID’ varchar(45) NOT NULL,

‘Password’ varchar(45) NOT NULL,

‘Gender’ varchar(45) NOT NULL,

‘Age’ INT NOT NULL

PRIMARY KEY (‘Serial\_Num’),

};

**B. Evaluation\_Pd**

CREATE TABLE Evaluation\_Pd (

‘Serial\_Num’ int NOT NULL AUTO\_INCREMENT,

‘Product\_Name’ varchar(45) NOT NULL,

‘Price’ int NOT NULL,

PRIMARY KEY (‘Serial\_Num’),

KEY ‘user\_idx’ (‘Serial\_Num’),

CONSTRAINT ‘user\_idx’ FOREIGN KEY(‘Serial\_Num’) REFERENCES ‘User’(‘Serial\_Num’’) ON DELETE NO ACTION ON UPDATE NO ACTION,

};

**C. Evaluation**

CREATE TABLE Evaluation (

‘Serial\_Num’ int NOT NULL AUTO\_INCREMENT,

PRIMARY KEY (‘Serial\_Num’),

KEY ‘user\_idx’ (‘Serial\_Num’),

CONSTRAINT ‘user\_idx’ FOREIGN KEY(‘Serial\_Num’) REFERENCES ‘User’(‘Serial\_Num’’) ON DELETE NO ACTION ON UPDATE NO ACTION,

};

**D. Taste Model**

CREATE TABLE Taste Model (

‘Product\_Num’ int NOT NULL AUTO\_INCREMENT,

‘Serial\_Num’ int NOT NULL,

PRIMARY KEY (‘Product\_Num’),

KEY ‘product\_idx’ (‘Product\_Num’),

CONSTRAINT ‘product\_idx’ FOREIGN KEY(‘Product\_Num’) REFERENCES ‘Product’(‘Product\_Num’’) ON DELETE NO ACTION ON UPDATE NO ACTION,

};

**E. Product**

CREATE TABLE Product (

‘Product\_Num’ int NOT NULL AUTO\_INCREMENT,

‘Product\_Name’ varchar(45) NOT NULL,

Price int NOT NULL,

Count int NOT NULL,

Delivery varchar(45) NOT NULL,

PRIMARY KEY (‘Product\_Num’),

};

**F. Cart**

CREATE TABLE Cart (

‘Serial\_Num’ int NOT NULL AUTO\_INCREMENT,

‘Product\_Num’ int NOT NULL AUTO\_INCREMENT,

Count int NOT NULL,

PRIMARY KEY (‘Serial\_Num’, ‘Product\_Num’),

KEY ‘user\_idx’ (‘Serial\_Num’),

KEY ‘product\_idx’ (‘Product\_Num’),

CONSTRAINT ‘user\_idx’ FOREIGN KEY(‘Serial\_Num’) REFERENCES ‘User’(‘Serial\_Num’’) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT ‘product\_idx’ FOREIGN KEY(‘Product\_Num’) REFERENCES ‘Product’(‘Product\_Num’’) ON DELETE NO ACTION ON UPDATE NO ACTION,

};

**G. Order**

CREATE TABLE Evaluation (

‘Serial\_Num’ int NOT NULL AUTO\_INCREMENT,

‘Product\_Num’ int NOT NULL,

PRIMARY KEY (‘Serial\_Num’),

KEY ‘user\_idx’ (‘Serial\_Num’),

CONSTRAINT ‘user\_idx’ FOREIGN KEY(‘Serial\_Num’) REFERENCES ‘User’(‘Serial\_Num’’) ON DELETE NO ACTION ON UPDATE NO ACTION,

};