

# 기초 서버 세팅

<https://github.com/codingspecialist/kakao-shop-api-setting>

## 데이터베이스 모델링

데이터베이스를 모델링할 때에는 가장 바탕이 되는 부분을 먼저 하는 것이 좋습니다.

- 예를들어 회원, 제품, 주문이 필요한 데이터베이스를 모델링한다면, 관계도 중에서 가장 바탕이 되는 회원, 제품을 먼저 모델링합니다.

테이블을 작성할 때 해당 테이블에 필드에는 반드시 PK(Primary Key)가 필요합니다. 사용하는 주요 이유는 다음과 같습니다.

- 중복된 값을 허용하지 않기 때문에 데이터 무결성을 유지할 수 있습니다.
- PK를 기준으로 행을 검색하거나 조인할 때 데이터베이스의 성능이 향상됩니다.
- 다른 테이블과의 관계를 설정할 때 PK를 사용하여 FK(Foreign Key)를 참조할 수 있습니다.
- 애플리케이션에서 특정 Row을 참조할 때 PK를 사용하여 빠르게 찾을 수 있습니다.

## User 테이블

- 먼저 회원가입의 응답 데이터의 형태를 확인합니다.

### 회원가입 Response Body



- 응답 데이터에서 필요한 필드를 확인합니다. 위의 경우에는 username, email, password가 필요합니다.
- 그리고 앞서 말했듯 테이블을 작성할 때에는 PK가 필요합니다. (흔히 이 값의 필드명은 ID(Identity)를 사용합니다.)
- 유저 테이블에는 추가적으로 어떤 필드가 필요할까요?
- 이제 테이블 필드를 채우는 것이 완료되었다면, 제약조건을 확인하는 것이 좋습니다. 어떤 제약조건이 더 필요할까요?

응답 데이터를 토대로 테이블을 설계해봅니다.

```
CREATE TABLE `USER`
(
    `id`          bigint(20) NOT NULL auto_increment,
    `username`    varchar(255) NOT NULL,
    `email`       varchar(255) NOT NULL,
    `password`    varchar(255) NOT NULL,
    `created_at`  datetime NOT NULL,
    `updated_at`  datetime NOT NULL,
    PRIMARY KEY (`id`),
)
```

## Product 테이블

- 상품 테이블을 모델링하기 위해서 상품 상세 보기의 데이터 형태를 확인합니다.

### 상품 상세 Response Body

```
{
  "success" : true,
  "response" :
  {
    "id" : 1,
    "productName" : "기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용품 특  
가전",
    "description" : "",
    "image" : "/images/1.jpg",
    "price" : 1000,
    "starCount" : 5,
    "options" : [
      {
        "id" : 1,
        "optionName" : "01. 슬라이딩 지퍼백 크리스마스에디션 4종",
        "price" : 10000
      },
      {
        "id" : 2,
        "optionName" : "02. 슬라이딩 지퍼백 플라워에디션 5종",
        "price" : 10900
      },
      {
        "id" : 3,
        "optionName" : "고무장갑 베이지 S(소형) 6팩",
        "price" : 9900
      }
    ]
  }
}
```

```

    },
    {
      "id" : 4,
      "optionName" : "뽑아쓰는 키친타올 130매 12팩",
      "price" : 16900
    },
    {
      "id" : 5,
      "optionName" : "2겹 식빵수세미 6매",
      "price" : 8900
    }
  ]
},
"error" : null
}

```

- 응답 데이터를 토대로 테이블을 설계해봅니다.

```

CREATE TABLE `product`
(
  `id`          BIGINT(20) NOT NULL auto_increment,
  `user_id`     BIGINT(20) DEFAULT NULL,
  `product_name` VARCHAR(255) NOT NULL,
  `description`  VARCHAR(255) NOT NULL,
  `price`       BIGINT(20) NOT NULL,
  `created_at`   DATETIME NOT NULL,
  `updated_at`   DATETIME NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (user_id) REFERENCES user(id)
);

```

## Option 테이블

- 다시한번 제품 상세 보기의 데이터 형식을 확인해봅니다.

```

{
  "success" : true,
  "response" :
  {
    "id" : 1,
    "productName" : "기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용품 특  
가전",
    "description" : "",
    "image" : "/images/1.jpg",
    "price" : 1000,
    "starCount" : 5,
    "options" : [
      {

```

```

        "id" : 1,
        "optionName" : "01. 슬라이딩 지퍼백 크리스마스에디션 4종",
        "price" : 10000
    },
    {
        "id" : 2,
        "optionName" : "02. 슬라이딩 지퍼백 플라워에디션 5종",
        "price" : 10900
    },
    {
        "id" : 3,
        "optionName" : "고무장갑 베이지 S(소형) 6팩",
        "price" : 9900
    },
    {
        "id" : 4,
        "optionName" : "뿔아쓰는 키친타올 130매 12팩",
        "price" : 16900
    },
    {
        "id" : 5,
        "optionName" : "2겹 식빵수세미 6매",
        "price" : 8900
    }
]
},
"error" : null
}

```

해당 응답 데이터에서는 제품에 대한 정보(제품명, 제품 설명, 제품 사진, 제품 가격)가 있습니다. 그리고 해당 제품에서 선택할수 있는 옵션에 대한 정보가 있습니다.

그렇다면 제품과 옵션을 따로 테이블을 생성하는 이유가 무엇일까요?

응답 데이터를 토대로 테이블을 설계해봅시다.

```

CREATE TABLE `option`
(
    `id`          BIGINT(20) NOT NULL auto_increment,
    `product_id` BIGINT(20) DEFAULT NULL,
    `option`      VARCHAR(255) NOT NULL,
    `price`       BIGINT(20) NOT NULL,
    `created_at`  DATETIME NOT NULL,
    `updated_at`  DATETIME NOT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (product_id) REFERENCES product(id)
);

```

- 장바구니 조회의 응답 데이터를 확인합니다.

```
{
  "success" : true,
  "response" :
  {
    "products" : [
      {
        "id" : 1,
        "productName" : "기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용
품 특가전",
        "carts" : [
          {
            "id" : 1,
            "option" :
            {
              "id" : 5,
              "optionName" : "2겹 식빵수세미 6매",
              "price" : 8900
            },
            "quantity" : 3,
            "price" : 26700
          }
        ]
      },
      {
        "id" : 3,
        "productName" : "삼성전자 JBL JR310 외 어린이용/성인용 헤드셋 3종!",
        "carts" : [
          {
            "id" : 2,
            "option" : {
              "id" : 10,
              "optionName" : "JR310BT (무선 전용) - 레드",
              "price" : 49900
            },
            "quantity" : 4,
            "price" : 199600
          },
          {
            "id" : 3,
            "option" : {
              "id" : 11,
              "optionName" : "JR310BT (무선 전용) - 그린",
              "price" : 49900
            },
            "quantity" : 5,
            "price" : 249500
          }
        ]
      }
    ]
  }
}
```

```

    ]
  }
],
"totalPrice" : 475800
},
"error" : null
}

```

응답 데이터를 토대로 테이블을 설계해봅니다.

```

CREATE TABLE `cart`
(
  `id`          BIGINT(20) NOT NULL auto_increment,
  `option_id`   BIGINT(20) DEFAULT NULL,
  `user_id`     BIGINT(20) DEFAULT NULL,
  `quantity`    BIGINT(20) NOT NULL,
  `option`      VARCHAR(255) NOT NULL,
  `price`       BIGINT(20) NOT NULL,
  `created_at`  DATETIME NOT NULL,
  `updated_at`  DATETIME NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (option_id) REFERENCES OPTION(id),
  FOREIGN KEY (user_id) REFERENCES user(id)
);

```

해당 테이블에는 총 두가지의 다른 테이블 정보가 필요합니다. 어떤 유저의 장바구니인지(`user_id`), 어떤 옵션을 선택했는지(`option_id`) 입니다.

## Order 테이블

- 주문 조회의 응답 데이터를 확인합니다.

```

{
  "success" : true,
  "response" :
  {
    "id" : 1,
    "products" : [
      {
        "productName" : "기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용품 특가전",
        "items" : [
          {
            "id" : 1,
            "optionName" : "01. 슬라이딩 지퍼백 크리스마스에디션 4종",

```

```

        "quantity" : 5,
        "price" : 50000
    },
    {
        "id" : 2,
        "optionName" : "02. 슬라이딩 지퍼백 플라워에디션 5종",
        "quantity" : 5,
        "price" : 54500
    }
]
},
{
    "productName" : "삼성전자 JBL JR310 외 어린이용/성인용 헤드셋 3종!",
    "items" : [
        {
            "id" : 3,
            "optionName" : "JR310BT (무선 전용) - 레드",
            "quantity" : 5,
            "price" : 249500
        }
    ]
}
],
"totalPrice" : 354000
},
"error" : null
}

```

- 응답 데이터를 토대로 테이블을 설계해봅니다.

```

CREATE TABLE `cart`
(
    `id`          BIGINT(20) NOT NULL auto_increment,
    `user_id`     BIGINT(20) DEFAULT NULL,
    `created_at`  DATETIME NOT NULL,
    `updated_at`  DATETIME NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (user_id) REFERENCES user(id)
);

```

## Item 테이블

- 주문 조회의 응답 데이터를 다시 확인합니다.

```

{
  "success" : true,
  "response" :
  {
    "id" : 1,
    "products" : [
      {
        "productName" : "기본에 슬라이딩 지퍼백 크리스마스/플라워에디션 에디션 외 주방용품
특가전",
        "items" : [
          {
            "id" : 1,
            "optionName" : "01. 슬라이딩 지퍼백 크리스마스에디션 4종",
            "quantity" : 5,
            "price" : 50000
          },
          {
            "id" : 2,
            "optionName" : "02. 슬라이딩 지퍼백 플라워에디션 5종",
            "quantity" : 5,
            "price" : 54500
          }
        ]
      },
      {
        "productName" : "삼성전자 JBL JR310 외 어린이용/성인용 헤드셋 3종!",
        "items" : [
          {
            "id" : 3,
            "optionName" : "JR310BT (무선 전용) - 레드",
            "quantity" : 5,
            "price" : 249500
          }
        ]
      }
    ],
    "totalPrice" : 354000
  },
  "error" : null
}

```

- 응답 데이터를 토대로 테이블을 설계해봅니다.



```
CREATE TABLE `item`  
(  
  `id`          BIGINT(20) NOT NULL auto_increment,  
  `option_id`   BIGINT(20) DEFAULT NULL,  
  `order_id`    BIGINT(20) DEFAULT NULL,  
  `price`       BIGINT(20) NOT NULL,  
  `quantity`    BIGINT(20) NOT NULL,  
  `created_at`  DATETIME NOT NULL,  
  `updated_at`  DATETIME NOT NULL,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (option_id) REFERENCES OPTION(id),  
  FOREIGN KEY (order_id) REFERENCES order(id)  
);
```