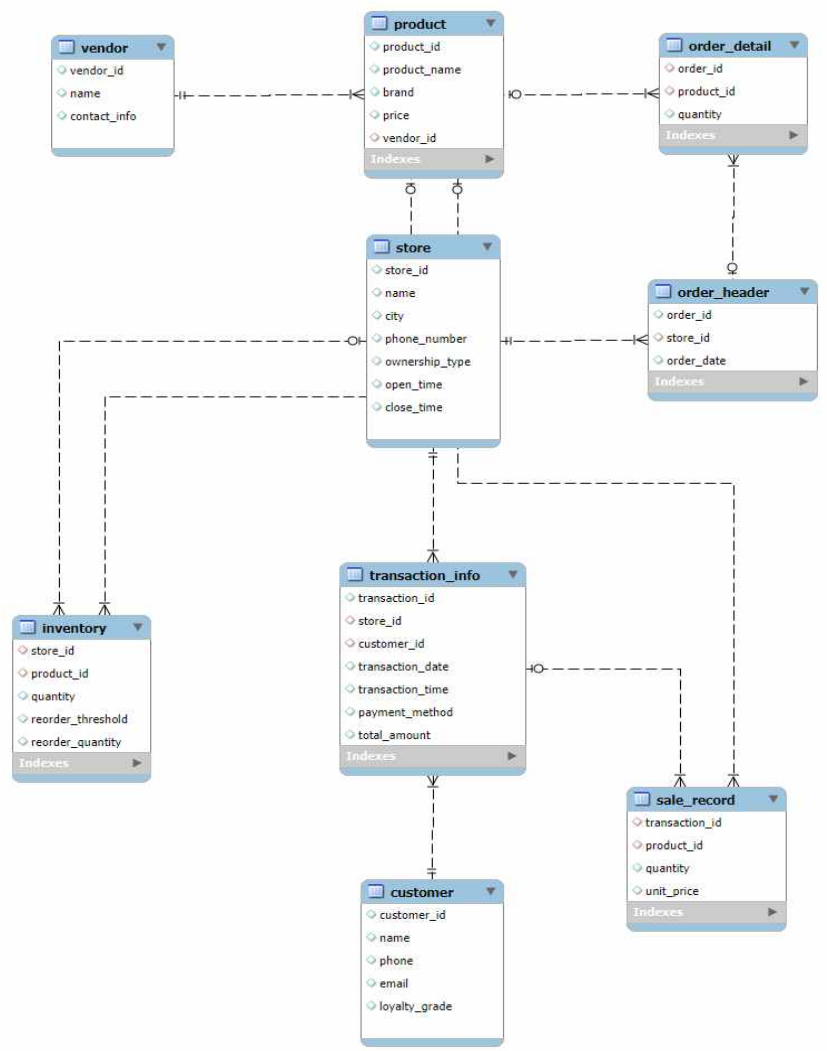
# DB Project 2

**2023078 신현주**

## 1. Logical Schema Design




9개의 엔터티(Store, Vendor, Product, Customer, Order_Header, Order_Detail, Transaction_Info, Sale_Record, Inventory)를 1:1 관계형 테이블로 생성하였다.
발주 헤더 정보는 Order_Header에, 개별 품목 수량은 Order_Detail에 분리하여 데이터 중복과 이상을 방지하였다. 또한, 고객 거래 정보는 Transaction_Info로 기록하고, 판매 품목은 Sale_Record에 기록하여 회계 집계와 품목 집계를 분리하였다.

모든 외래키는 ON DELETE CASCADE 로 설정하여 상위 레코드 삭제 시 연관 데이터를 자동 정리하게 하여 관리 비용을 줄였으며 ownership_type 과 loyalty_grade 는 ENUM으로 도메인을 제한하여 입력 오류를 차단하였다. 시간 필드는 TIME/DATE 자료형을 사용하여 조회·집계를 용이하게 하고, 전체 스키마가 BCNF를 만족하도록 설계하였다.

## 2. Normalization Analysis

**(1) 함수적 종속성 분석**

- Store 테이블: store_id가 단일 후보키이며, 이 키가 나머지 정보를 전부 결정한다
- Vendor 테이블: vendor_id → name, contact_info 가 유일 FD이고, vendor_id가 후보키이기 때문에 부분·이행 종속이 존재하지 않는다
- Product 테이블: product_id → product_name, brand, price, vendor_id가 성립하며 product_id는 후보키이다.
- 초기 설계 단계에서 존재했던 Order_Item을 제외하면, 나머지 테이블도 동일 원리로 부분, 이행 종속이 발견되지 않았다.

**(2) BCNF 분해**

예외적으로, 초기 설계 단계에서 존재했던 Order_Item(order_id, product_id, vendor_id, quantity) 테이블은 product_id → vendor_id가 성립하지만 product_id는 후보키가 아니었기 때문에 BCNF를 위반하였다.

1. vendor_id 제거: 발주 내역에서 공급사를 직접 저장하지 않고, Product가 이미 공급사를 참조하므로 중복을 없앴다.
2. 테이블 분리:

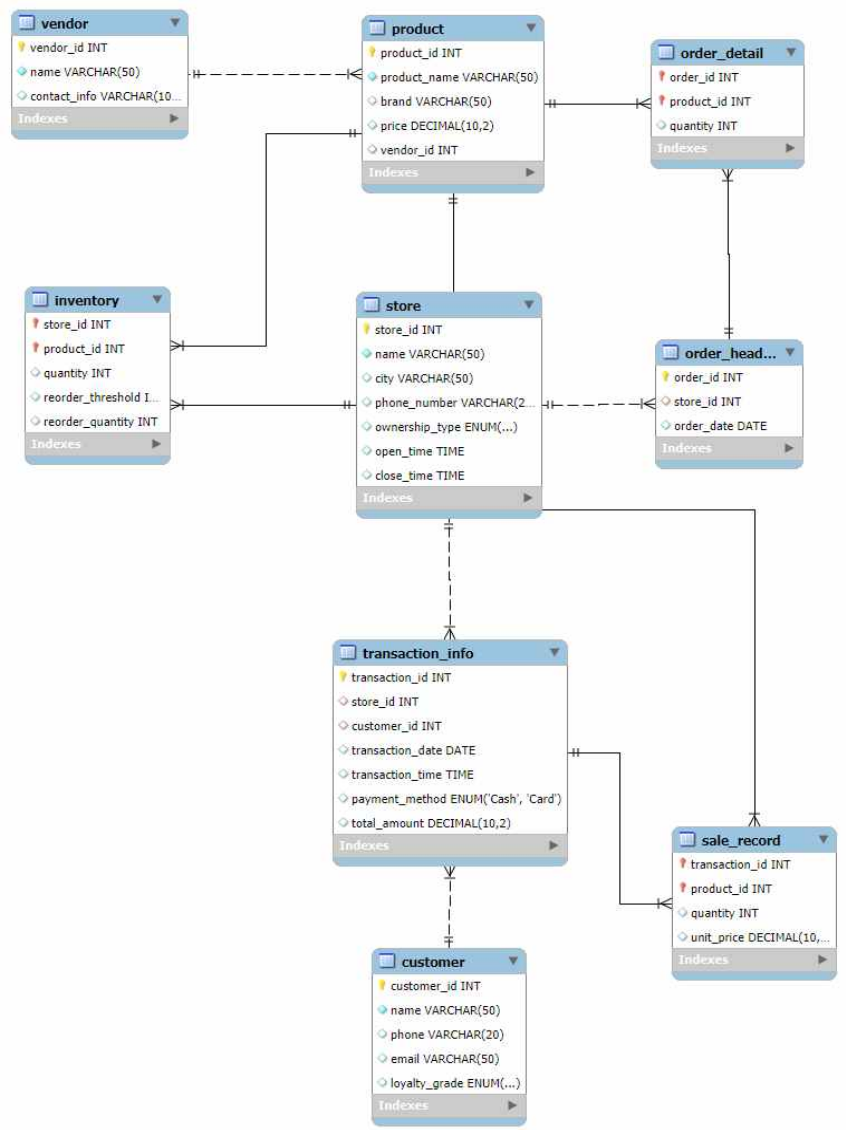
Order_Header(order_id, store_id, order_date) – 발주서 일반 정보 저장
Order_Detail(order_id, product_id, quantity) – 발주 품목, 수량 저장
두 테이블은 order_id로 1:N 관계를 이룬다.

새로 생성된 두 릴레이션의 모든 FD 결정자는 각각 order_id 또는 (order_id, product_id)라는 후보키가 되었으므로 BCNF 조건을 만족한다.

**(3) 최종 BCNF 충족 증명**

| 테이블 | 주요 FD | BCNF 여부 |
|---|---|---|
| Store | store_id → name, city, … | O |
| Vendor | vendor_id → name, contact_info | O |
| Product | product_id → product_name, price, vendor_id | O |
| Customer | customer_id → name, phone, email, loyalty_grade | O |
| Order_Header | order_id → store_id, order_date | O |
| Order_Detail | (order_id, product_id) → quantity | O |
| Transaction_Info | transaction_id → store_id, customer_id, … | O |
| Sale_Record | (transaction_id, product_id) → quantity, unit_price | O |
| Inventory | (store_id, product_id) → quantity, reorder_ | O |

## 3. Physical Implementation




**(1) 데이터 타입 선정 이유**

부동소수 오차 방지를 위해 가격은 DECIMAL(10,2)를 사용하였으며, 전화번호는 지역번호와 하이픈을 고려하여 VARCHAR(20)을 사용하였다. 또한, 열거형 컬럼을 활용하여 도메인이 제한되는 값을 단일 필드로 관리했다. Franchise/Corporate 두 값만 허용하는 Store.ownership_type과 VIP/Regular 두 값만 허용하는 Customer.loyalty_grade를 그 예로 들 수 있다.

**(2) 제약 및 비즈니스 규칙**

1. 단일 공급사 체계

편의점에서 취급하는 특정 상품은 계약된 한 벤더에게서만 공급받는 것이 일반적이기 때문에 Product 테이블에만 공급사 ID를 두고, 발주·판매 테이블에는 같은 정보가 중복 저장되지 않도록 하였다.

2. 안전재고 기반 자동 발주
편의점은 품목마다 '재주문 임계치'를 설정해 두고 재고가 그보다 떨어지면 자동으로 발주를 생성하기 때문에 Inventory 테이블에 reorder_threshold와 reorder_quantity 칼럼을 두고, quantity < reorder_threshold 조건을 만족할 때 Type 5 쿼리로 부족 품목을 즉시 확인할 수 있게 하였다.

3. 가맹점과 직영점 구분
본사 기준으로 운영 방식이 다른 두 매장군을 구분하고, 품목 다양성·매출 등을 따로 분석할 수 있도록 Store 테이블에 ownership_type ENUM을 두었다.

4. 연쇄 삭제 (CASCADE)
매장이 폐점되거나 상품이 단종될 때 관련 재고, 발주, 판매 기록까지 일괄 정리해야하므로 모든 외래키에 ON DELETE CASCADE를 적용하여 상위 레코드 삭제 시 하위 레코드가 자동으로 삭제되도록 하였다.

## 4. Application Development

**(1) DB 연결 구현**

- MySQL C API 사용
- 접속 정보는 main.cpp 상단에 기재하였다

**(2) 주요 쿼리 구현**

TYPE 1: 재고 테이블과 매장·상품 테이블을 JOIN한 뒤, 사용자가 입력한 키워드를 LIKE '%키워드%'로 검색하여 매장별 잔여 수량을 보여준다.
TYPE 2: 최근 1개월 판매량을 매장/상품별로 집계한 서브쿼리에서 MAX(qty)를 뽑아, 각 매장당 최다 판매 상품만 외부쿼리로 추린다.
TYPE 3: 분기, 연도 조건으로 거래 금액을 GROUP BY store_id로 합산하고, ORDER BY SUM(total_amount) DESC로 정렬하여 1위 매장만 반환한다.
TYPE 4: 발주량, 판매량을 벤더 기준으로 각각 집계한 두 서브쿼리를 LEFT JOIN하여 공급량 기준 1위 벤더와 판매량을 동시에 보여준다.
TYPE 5: 재고 테이블에서 quantity < reorder_threshold 조건으로 부족 품목을 걸러 매장, 상품, 임계치 정보를 출력한다.
TYPE 6: VIP 거래 중 기준 상품이 포함된 transaction_id 목록을 IN (SELECT DISTINCT ...)로 추린 뒤, 같은 거래의 다른 상품을 합계해 상위 3개를 찾는다.
TYPE 7: 매장별 품목 종류 수를 계산한 서브쿼리, 유형별 MAX(variety)를 구한 서브쿼리를 다시 JOIN하여 가맹/직영 각각 품목 수가 가장 많은 매장을 출력한다.

**(3) 오류 처리 및 UI**

프로그램은 사용자가 메뉴 번호를 입력하면 해당 쿼리 함수를 호출하는 단순 콘솔 메뉴 구조로 동작한다. 각 함수에서 SQL을 실행할 때 mysql_query가 실패하면 mysql_error()의 메시지를 화면에 출력해 원인을 알 수 있도록 하였으며, 성공/실패 여부를 판단해 불필요한

후속 처리를 막도록 하였다.

## 5. Testing and Validation

### (1) 테스트

```
------------ SELECT QUERY TYPES ------------
 1. TYPE 1
 2. TYPE 2
 3. TYPE 3
 4. TYPE 4
 5. TYPE 5
 6. TYPE 6
 7. TYPE 7
 0. QUIT
Select: 1

-------- TYPE 1 --------
** Which stores currently carry a certain product (by UPC, name, or brand), and how much inventory do they have? **
Enter product identifier (UPC, name, or brand): ORION
store_name      product_name    quantity
CU Gangnam station      Choco Pie       20
GS Shinchon     Choco Pie       6
CU Daejeon      Choco Pie       10
CU Gangnam station      Sun Chip        12
GS Itaewon      Sun Chip        8
```

> TYPE 1에 ORION 입력

```
Select: 2

-------- TYPE 2 --------
** Which products have the highest sales volume in each store over the past month? **
store_name      product_name    qty
CU Gangnam station      Maxim Coffee    3
GS Shinchon     T.O.P   3
CU City Hall    Maxim Coffee    2
GS Itaewon      Dutch Black Coffee 390ml        3
7-Eleven Busan  Hot Chicken Flavor Ramen        4
CU Daejeon      Banana Flavored Milk    2
```

> TYPE 2

```
Select: 3

-------- TYPE 3 --------
** Which store has generated the highest overall revenue this quarter? **
store_name      revenue
GS Shinchon     11100.00
```

> TYPE 3

```
Select: 4

-------- TYPE 4 --------
** Which vendor supplies the most products across the chain, and how many total units have been sold? **
vendor  total_supplied  total_sold
Maxim   340     8
```

> TYPE 4

```
Select: 5

-------- TYPE 5 --------
** Which products in each store are below the reorder threshold and need restocking? **
store_name      product_name    quantity        reorder_threshold
CU Gangnam station      Coke 500ml      8       10
GS Shinchon     Choco Pie       6       10
7-Eleven Hongdae        Maxim Coffee    0       10
CU Daegu station        Coke 500ml      5       10
CU Daegu station        Pocari Sweat 500ml      3       10
GS Jeju Papico Choco Ice Cream  5       10
CU City Hall    Papico Choco Ice Cream  9       10
GS Itaewon      Sun Chip        8       10
GS Gwangju      Coke 500ml      6       10
GS Gwangju      Dutch Black Coffee 390ml        5       10
```

> TYPE 5

```
Select: 6

-------- TYPE 6 --------
** List the top 3 items that loyalty program customers typically purchase with coffee. **
Enter a product name: Maxim Coffee
product_name    total_qty
Choco Pie       1
Papico Choco Ice Cream  1

Select: 7

-------- TYPE 7 --------
** Among franchise-owned stores, which one offers the widest variety of products, and how does that compare to corporate-owned stores? **
ownership_type  store_name      variety
Franchise       CU Gangnam station      4
Corporate       GS Shinchon     4
```

> TYPE 6에 Maxim Coffe 입력

```
Select: 7

-------- TYPE 7 --------
** Among franchise-owned stores, which one offers the widest variety of products, and how does that compare to corporate-owned stores? **
ownership_type  store_name      variety
Franchise       CU Gangnam station      4
Corporate       GS Shinchon     4
```

> TYPE 7

**(2) 비즈니스 규칙 검증**

먼저 연쇄 삭제(CASCADE) 규칙 점검을 위해 DELETE FROM Store WHERE store_id = 1;을 실행하였고, 해당 매장이 갖고 있던 재고/발주/판매 레코드가 모두 제거되어 참조 무결성이 유지됨을 확인했다. 두 번째로, 재고 알림 규칙을 검증하기위해 Store 3의 상품 1010 재고를 0으로 만든 뒤 Type 5 쿼리를 실행하였고, 해당 품목이 부족 목록에 표시되어 quantity < reorder_threshold 조건이 정상 작동함을 확인하였다. 마지막으로 VIP 고객 장바구니 분석을 테스트하기 위해 Coffee가 포함된 VIP 거래들을 조회한 뒤 Type 6 쿼리를 실행하였고, Coffee와 함께 가장 많이 팔린 품목으로 Choco Pie 1개, Papico Choco Ice Cream 1개가 집계됨을 확인하였다.