

# 데이터베이스 스키마 및 테이블 관계 문서

작성일: 2025-12-29

목적: 데이터베이스 테이블 구조, 관계, 모델 정보를 정리한 참조 문서

데이터베이스: MySQL (소셜 로그인 지원 버전)

작성자: 김택권

## ■ 데이터베이스 관계도

branch (지점)

  └ staff (직원) – 1:N

user (고객)

  └ user\_auth\_identities (인증 정보) – 1:N  
  └ purchase\_item (구매 내역) – 1:N  
  └ pickup (수령) – 1:N  
  └ refund (반품) – 1:N

maker (제조사)

  └ product (제품) – 1:N  
  └ receive (입고) – 1:N  
  └ request (발주) – 1:N

kind\_category (종류 카테고리)

  └ product (제품) – 1:N

color\_category (색상 카테고리)

  └ product (제품) – 1:N

size\_category (사이즈 카테고리)

  └ product (제품) – 1:N

gender\_category (성별 카테고리)

  └ product (제품) – 1:N

refund\_reason\_category (반품 사유 카테고리)

  └ refund (반품) – 1:N

product (제품)

  └ purchase\_item (구매 내역) – 1:N  
  └ receive (입고) – 1:N  
  └ request (발주) – 1:N

purchase\_item (구매 내역)

  └ pickup (수령) – 1:N

pickup (수령)

  └ refund (반품) – 1:N

**staff** (직원)

- └ refund (반품 처리) – 1:N
- └ receive (입고 처리) – 1:N
- └ request (발주 요청) – 1:N
- └ chatting (채팅 담당) – 1:N (논리적 참조)

**user** (고객)

- └ chatting (채팅 세션) – 1:N (논리적 참조)

▷ **시각화 ERD:** [dbdiagram.io에서 보기](#)

▣ **원본 ERD:** [Miro에서 보기](#)

## ▣ 테이블 상세 정보

### 1. branch (지점)

**설명:** 오프라인 지점 정보를 저장하는 테이블입니다.

컬럼명	타입	설명	제약조건
br_seq	INT	지점 고유 ID	PRIMARY KEY, AUTO_INCREMENT
br_name	VARCHAR(100)	지점명	NOT NULL, UNIQUE
br_phone	VARCHAR(30)	지점 전화번호	
br_address	VARCHAR(255)	지점 주소	
br_lat	DECIMAL(10,7)	지점 위도	
br_lng	DECIMAL(10,7)	지점 경도	

**관계:**

- **staff.br\_seq → branch.br\_seq** (N:1)

**인덱스:**

- **idx\_branch\_name:** 지점명 UNIQUE 인덱스

### 2. user (고객)

**설명:** 고객 계정 정보를 저장하는 테이블입니다. 소셜 로그인을 지원합니다.

컬럼명	타입	설명	제약조건
u_seq	INT	고객 고유 ID	PRIMARY KEY, AUTO_INCREMENT
u_email	VARCHAR(255)	고객 이메일	NOT NULL, UNIQUE
u_name	VARCHAR(255)	고객 이름 (선택 사항)	NULL 허용

컬럼명	타입	설명	제약조건
u_phone	VARCHAR(30)	고객 전화번호 (선택 사항)	NULL 허용, UNIQUE
u_image	MEDIUMBLOB	고객 프로필 이미지	
u_address	VARCHAR(255)	고객 주소	
created_at	DATETIME	고객 가입일자	NOT NULL, DEFAULT CURRENT_TIMESTAMP
u_quit_date	DATETIME	고객 탈퇴일자	

관계:

- user\_auth\_identities.u\_seq → user.u\_seq (N:1)
- purchase\_item.u\_seq → user.u\_seq (N:1)
- pickup.u\_seq → user.u\_seq (N:1)
- refund.u\_seq → user.u\_seq (N:1)

인덱스:

- idx\_user\_email: 이메일 UNIQUE 인덱스
- idx\_user\_phone: 전화번호 UNIQUE 인덱스
- idx\_user\_created\_at: 가입일자 인덱스
- idx\_user\_quit\_date: 탈퇴일자 인덱스

주의사항:

- u\_id, u\_password 필드는 제거되었습니다 (소셜 로그인 지원)
- 인증 정보는 user\_auth\_identities 테이블에서 관리합니다

### 3. user\_auth\_identities (사용자 로그인 수단별 인증 정보)

설명: 사용자의 로그인 수단별 인증 정보를 저장하는 테이블입니다. 로컬 로그인, 구글, 카카오 등을 지원합니다.

컬럼명	타입	설명	제약조건
auth_seq	INT	인증 수단 고유 ID	PRIMARY KEY, AUTO_INCREMENT
u_seq	INT	고객 번호	NOT NULL, FOREIGN KEY → user.u_seq
provider	VARCHAR(50)	로그인 제공자	NOT NULL (local, google, kakao 등)
provider_subject	VARCHAR(255)	제공자 고유 식별자	NOT NULL (로컬: 이메일, 구글: sub, 카카오: id)
provider_issuer	VARCHAR(255)	제공자 발급자	(구글 iss 등)
email_at_provider	VARCHAR(255)	제공자에서 받은 이메일	
password	VARCHAR(255)	로컬 로그인 비밀번호	(로컬만 사용)

컬럼명	타입	설명	제약조건
created_at	DATETIME	생성일자	NOT NULL, DEFAULT CURRENT_TIMESTAMP
last_login_at	DATETIME	마지막 로그인 일시	

관계:

- `user_auth_identities.u_seq` → `user.u_seq` (N:1)

인덱스:

- `idx_provider_subject`: (provider, provider\_subject) UNIQUE 인덱스
- `idx_user_auth_u_seq`: 고객별 인증 정보 조회
- `idx_user_auth_provider`: 제공자별 조회

제약조건:

- FOREIGN KEY: ON DELETE CASCADE ON UPDATE CASCADE

#### 4. staff (직원)

설명: 지점 직원 정보를 저장하는 테이블입니다.

컬럼명	타입	설명	제약조건
s_seq	INT	직원 고유 ID	PRIMARY KEY, AUTO_INCREMENT
s_id	VARCHAR(50)	직원 로그인 ID	NOT NULL, UNIQUE
br_seq	INT	소속 지점 ID	NOT NULL, FOREIGN KEY → branch.br_seq
s_password	VARCHAR(255)	직원 비밀번호(해시)	NOT NULL
s_name	VARCHAR(255)	직원명	NOT NULL
s_phone	VARCHAR(30)	직원 전화번호	NOT NULL, UNIQUE
s_rank	VARCHAR(100)	직원 직급	
s_image	MEDIUMBLOB	직원 프로필 이미지	
s_superseq	INT	상급자 직원 ID	(논리적 참조, FK 없음)
created_at	DATETIME	생성일자	NOT NULL, DEFAULT CURRENT_TIMESTAMP
s_quit_date	DATETIME	직원 탈퇴 일자	

관계:

- `staff.br_seq` → `branch.br_seq` (N:1)
- `refund.s_seq` → `staff.s_seq` (N:1)
- `receive.s_seq` → `staff.s_seq` (N:1)
- `request.s_seq` → `staff.s_seq` (N:1)

**인덱스:**

- `idx_staff_br_seq`: 지점별 직원 조회
- `idx_staff_id`: 로그인 ID UNIQUE 인덱스
- `idx_staff_phone`: 전화번호 UNIQUE 인덱스
- `idx_staff_created_at`: 생성일자 인덱스
- `idx_staff_quit_date`: 탈퇴일자 인덱스

**제약조건:**

- FOREIGN KEY: ON DELETE RESTRICT ON UPDATE CASCADE
- 

**5. maker (제조사)**

**설명:** 신발 제조사 정보를 저장하는 테이블입니다.

컬럼명	타입	설명	제약조건
<code>m_seq</code>	INT	제조사 고유 ID	PRIMARY KEY, AUTO_INCREMENT
<code>m_name</code>	VARCHAR(255)	제조사명	NOT NULL, UNIQUE
<code>m_phone</code>	VARCHAR(30)	제조사 전화번호	
<code>m_address</code>	VARCHAR(255)	제조사 주소	

**관계:**

- `product.m_seq → maker.m_seq` (N:1)
- `receive.m_seq → maker.m_seq` (N:1)
- `request.m_seq → maker.m_seq` (N:1)

**인덱스:**

- `idx_maker_name`: 제조사명 UNIQUE 인덱스
- 

**6. kind\_category (종류 카테고리)**

**설명:** 제품 종류 카테고리를 저장하는 테이블입니다.

컬럼명	타입	설명	제약조건
<code>kc_seq</code>	INT	종류 카테고리 ID	PRIMARY KEY, AUTO_INCREMENT
<code>kc_name</code>	VARCHAR(100)	종류명	NOT NULL, UNIQUE (러닝화, 스니커즈, 부츠 등)

**관계:**

- `product.kc_seq → kind_category.kc_seq` (N:1)

**인덱스:**

- idx\_kind\_category\_name: 종류명 UNIQUE 인덱스
- 

## 7. color\_category (색상 카테고리)

**설명:** 제품 색상 카테고리를 저장하는 테이블입니다.

컬럼명	타입	설명	제약조건
cc_seq	INT	색상 카테고리 ID	PRIMARY KEY, AUTO_INCREMENT
cc_name	VARCHAR(100)	색상명	NOT NULL, UNIQUE

**관계:**

- product.cc\_seq → color\_category.cc\_seq (N:1)

**인덱스:**

- idx\_color\_category\_name: 색상명 UNIQUE 인덱스
- 

## 8. size\_category (사이즈 카테고리)

**설명:** 제품 사이즈 카테고리를 저장하는 테이블입니다.

컬럼명	타입	설명	제약조건
sc_seq	INT	사이즈 카테고리 ID	PRIMARY KEY, AUTO_INCREMENT
sc_name	VARCHAR(100)	사이즈 값	NOT NULL, UNIQUE

**관계:**

- product.sc\_seq → size\_category.sc\_seq (N:1)

**인덱스:**

- idx\_size\_category\_name: 사이즈값 UNIQUE 인덱스
- 

## 9. gender\_category (성별 카테고리)

**설명:** 제품 성별 카테고리를 저장하는 테이블입니다.

컬럼명	타입	설명	제약조건
gc_seq	INT	성별 카테고리 ID	PRIMARY KEY, AUTO_INCREMENT
gc_name	VARCHAR(100)	성별 구분	NOT NULL, UNIQUE

**관계:**

- product.gc\_seq → gender\_category.gc\_seq (N:1)

**인덱스:**

- **idx\_gender\_category\_name:** 성별명 UNIQUE 인덱스
- 

**10. refund\_reason\_category (반품 사유 카테고리)**

**설명:** 반품 사유 카테고리를 저장하는 테이블입니다.

컬럼명	타입	설명	제약조건
ref_re_seq	INT	반품 사유 번호	PRIMARY KEY, AUTO_INCREMENT
ref_re_name	VARCHAR(100)	반품 사유명	NOT NULL, UNIQUE

**관계:**

- **refund.ref\_re\_seq → refund\_reason\_category.ref\_re\_seq (N:1)**

**인덱스:**

- **idx\_refund\_reason\_name:** 반품 사유명 UNIQUE 인덱스
- 

**11. product (제품)**

**설명:** 판매 상품(SKU) 정보를 저장하는 테이블입니다.

컬럼명	타입	설명	제약조건
p_seq	INT	제품 고유 ID	PRIMARY KEY, AUTO_INCREMENT
kc_seq	INT	제품 종류 카테고리 ID	NOT NULL, FOREIGN KEY → kind_category.kc_seq
cc_seq	INT	제품 색상 카테고리 ID	NOT NULL, FOREIGN KEY → color_category.cc_seq
sc_seq	INT	제품 사이즈 카테고리 ID	NOT NULL, FOREIGN KEY → size_category.sc_seq
gc_seq	INT	제품 성별 카테고리 ID	NOT NULL, FOREIGN KEY → gender_category.gc_seq
m_seq	INT	제조사 ID	NOT NULL, FOREIGN KEY → maker.m_seq
p_name	VARCHAR(255)	제품명	
p_price	INT	제품 가격	DEFAULT 0
p_stock	INT	중앙 재고 수량	NOT NULL, DEFAULT 0
p_image	VARCHAR(255)	제품 이미지 경로	
p_description	TEXT	제품 설명	

컬럼명	타입	설명	제약조건
created_at	DATETIME	제품 등록일자	NOT NULL, DEFAULT CURRENT_TIMESTAMP
<b>관계:</b>			
<ul style="list-style-type: none"> <li>product.kc_seq → kind_category.kc_seq (N:1)</li> <li>product.cc_seq → color_category.cc_seq (N:1)</li> <li>product.sc_seq → size_category.sc_seq (N:1)</li> <li>product.gc_seq → gender_category.gc_seq (N:1)</li> <li>product.m_seq → maker.m_seq (N:1)</li> <li>purchase_item.p_seq → product.p_seq (N:1)</li> <li>receive.p_seq → product.p_seq (N:1)</li> <li>request.p_seq → product.p_seq (N:1)</li> </ul>			
<b>인덱스:</b>			
<ul style="list-style-type: none"> <li>idx_product_p_name: 제품명 인덱스</li> <li>idx_product_m_seq: 제조사별 제품 조회</li> <li>idx_product_kc_seq: 종류별 제품 조회</li> <li>idx_product_cc_seq: 색상별 제품 조회</li> <li>idx_product_sc_seq: 사이즈별 제품 조회</li> <li>idx_product_gc_seq: 성별별 제품 조회</li> <li>idx_product_created_at: 등록일자 인덱스</li> </ul>			
<b>제약조건:</b>			
<ul style="list-style-type: none"> <li>UNIQUE: uq_product_color_size_maker_name (cc_seq, sc_seq, m_seq, p_name) <ul style="list-style-type: none"> <li>같은 색상, 사이즈, 제조사, 제품명 조합은 중복 불가</li> <li>동일한 제품명이라도 색상, 사이즈, 제조사가 다르면 별도 제품으로 등록 가능</li> <li>동일한 색상, 사이즈, 제조사라도 제품명이 다르면 별도 제품으로 등록 가능</li> </ul> </li> <li>FOREIGN KEY: 모든 카테고리 및 제조사는 ON DELETE RESTRICT ON UPDATE CASCADE</li> </ul>			
<b>검수 결과 (2026-01-03):</b>			
<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> UNIQUE 제약조건이 실제 DB에 정확히 반영되어 있음</li> <li><input checked="" type="checkbox"/> 중복 데이터 없음 확인</li> </ul>			
<b>12. purchase_item (구매 내역)</b>			
<b>설명:</b> 고객의 구매 내역을 저장하는 테이블입니다.			
컬럼명	타입	설명	제약조건
b_seq	INT	구매 고유 ID	PRIMARY KEY, AUTO_INCREMENT
br_seq	INT	수령 지점 ID	NOT NULL, FOREIGN KEY → branch.br_seq
u_seq	INT	구매 고객 ID	NOT NULL, FOREIGN KEY → user.u_seq
p_seq	INT	구매 제품 ID	NOT NULL, FOREIGN KEY → product.p_seq

컬럼명	타입	설명	제약조건
b_price	INT	구매 당시 가격	DEFAULT 0
b_quantity	INT	구매 수량	DEFAULT 1
b_date	DATETIME	구매 일시	NOT NULL
b_tnum	VARCHAR(100)	결제 트랜잭션 번호	
b_status	VARCHAR(50)	상품주문상태	

#### 상태값 (b\_status):

- '0': 준비중
- '1': 준비완료
- '2': 수령완료
- '3': 반품완료
- NULL: 상태 미정

참고: b\_status는 숫자 문자열로 저장되며, Flutter 애플리케이션에서 숫자로 파싱하여 사용합니다.

#### 검수 결과 (2026-01-03):

- 모든 b\_status 값이 숫자 문자열 ('0', '1', '2', '3') 또는 NULL로 저장되어 있음
- 실제 DB 데이터 분포 확인: NULL(5개), '0'(1개), '1'(3개), '2'(2개), '3'(6개)

#### 관계:

- purchase\_item.br\_seq → branch.br\_seq (N:1)
- purchase\_item.u\_seq → user.u\_seq (N:1)
- purchase\_item.p\_seq → product.p\_seq (N:1)
- pickup.b\_seq → purchase\_item.b\_seq (N:1)

#### 인덱스:

- idx\_purchase\_item\_b\_tnum: 결제 트랜잭션 번호 인덱스
- idx\_purchase\_item\_b\_date: 구매 일시 인덱스
- idx\_purchase\_item\_u\_seq: 고객별 구매 내역 조회
- idx\_purchase\_item\_br\_seq: 지점별 구매 내역 조회
- idx\_purchase\_item\_p\_seq: 제품별 구매 내역 조회
- idx\_purchase\_item\_b\_status: 상태별 조회

#### 제약조건:

- FOREIGN KEY: 모든 참조는 ON DELETE RESTRICT ON UPDATE CASCADE

### 13. pickup (수령)

설명: 오프라인 수령 기록을 저장하는 테이블입니다.

컬럼명	타입	설명	제약조건
-----	----	----	------

컬럼명	타입	설명	제약조건
pic_seq	INT	수령 고유 ID	PRIMARY KEY, AUTO_INCREMENT
b_seq	INT	구매 ID	NOT NULL, FOREIGN KEY → purchase_item.b_seq
u_seq	INT	고객 번호	NOT NULL, FOREIGN KEY → user.u_seq
created_at	DATETIME	수령 완료 일시	NOT NULL, DEFAULT CURRENT_TIMESTAMP

관계:

- pickup.b\_seq → purchase\_item.b\_seq (N:1)
- pickup.u\_seq → user.u\_seq (N:1)
- refund.pic\_seq → pickup.pic\_seq (N:1)

인덱스:

- idx\_pickup\_b\_seq: 구매별 수령 조회
- idx\_pickup\_u\_seq: 고객별 수령 조회
- idx\_pickup\_created\_at: 수령 일시 인덱스

제약조건:

- FOREIGN KEY: 모든 참조는 ON DELETE RESTRICT ON UPDATE CASCADE

## 14. refund (반품/환불)

설명: 오프라인 반품/환불 기록을 저장하는 테이블입니다.

컬럼명	타입	설명	제약조건
ref_seq	INT	반품 고유 ID	PRIMARY KEY, AUTO_INCREMENT
ref_date	DATETIME	반품 처리 일시	
ref_reason	VARCHAR(255)	반품 사유	
ref_re_seq	INT	반품 사유 번호	FOREIGN KEY → refund_reason_category.ref_re_seq
ref_re_content	VARCHAR(255)	반품 사유 내용	
u_seq	INT	반품 요청 고객 ID	NOT NULL, FOREIGN KEY → user.u_seq
s_seq	INT	반품 처리 직원 ID	NOT NULL, FOREIGN KEY → staff.s_seq
pic_seq	INT	수령 ID	NOT NULL, FOREIGN KEY → pickup.pic_seq

관계:

- refund.ref\_re\_seq → refund\_reason\_category.ref\_re\_seq (N:1)

- refund.u\_seq → user.u\_seq (N:1)
- refund.s\_seq → staff.s\_seq (N:1)
- refund.pic\_seq → pickup.pic\_seq (N:1)

**인덱스:**

- idx\_refund\_u\_seq: 고객별 반품 조회
- idx\_refund\_s\_seq: 직원별 반품 처리 조회
- idx\_refund\_pic\_seq: 수령별 반품 조회
- idx\_refund\_ref\_date: 반품 처리 일시 인덱스
- idx\_refund\_ref\_re\_seq: 반품 사유별 조회

**제약조건:**

- FOREIGN KEY: 모든 참조는 ON DELETE RESTRICT ON UPDATE CASCADE
- 

**15. receive (입고)**

**설명:** 제조사로부터의 입고(수주) 처리 기록을 저장하는 테이블입니다.

컬럼명	타입	설명	제약조건
rec_seq	INT	입고 고유 ID	PRIMARY KEY, AUTO_INCREMENT
rec_quantity	INT	입고 수량	DEFAULT 0
rec_date	DATETIME	입고 처리 일시	
s_seq	INT	입고 처리 직원 ID	NOT NULL, FOREIGN KEY → staff.s_seq
p_seq	INT	입고 제품 ID	NOT NULL, FOREIGN KEY → product.p_seq
m_seq	INT	제조사 ID	NOT NULL, FOREIGN KEY → maker.m_seq

**관계:**

- receive.s\_seq → staff.s\_seq (N:1)
- receive.p\_seq → product.p\_seq (N:1)
- receive.m\_seq → maker.m\_seq (N:1)

**인덱스:**

- idx\_receive\_s\_seq: 직원별 입고 처리 조회
- idx\_receive\_p\_seq: 제품별 입고 조회
- idx\_receive\_m\_seq: 제조사별 입고 조회
- idx\_receive\_rec\_date: 입고 일시 인덱스

**제약조건:**

- FOREIGN KEY: 모든 참조는 ON DELETE RESTRICT ON UPDATE CASCADE
- 

**16. request (발주/품의)**

**설명:** 재고 부족 시 발주/품의 기록을 저장하는 테이블입니다.

컬럼명	타입	설명	제약조건
req_seq	INT	발주/품의 고유 ID	PRIMARY KEY, AUTO_INCREMENT
req_date	DATETIME	발주 요청 일시	
req_content	TEXT	발주 내용	
req_quantity	INT	발주 수량	DEFAULT 0
req_manappdate	DATETIME	팀장 결재 일시	
req_dirappdate	DATETIME	이사 결재 일시	
s_seq	INT	발주 요청 직원 ID	NOT NULL, FOREIGN KEY → staff.s_seq
p_seq	INT	발주 제품 ID	NOT NULL, FOREIGN KEY → product.p_seq
m_seq	INT	제조사 ID	NOT NULL, FOREIGN KEY → maker.m_seq
s_superseq	INT	승인자 직원 ID	(논리적 참조, FK 없음)

**관계:**

- request.s\_seq → staff.s\_seq (N:1)
- request.p\_seq → product.p\_seq (N:1)
- request.m\_seq → maker.m\_seq (N:1)

**인덱스:**

- idx\_request\_s\_seq: 직원별 발주 요청 조회
- idx\_request\_p\_seq: 제품별 발주 조회
- idx\_request\_m\_seq: 제조사별 발주 조회
- idx\_request\_req\_date: 발주 요청 일시 인덱스
- idx\_request\_req\_manappdate: 팀장 결재 일시 인덱스
- idx\_request\_req\_dirappdate: 이사 결재 일시 인덱스

**제약조건:**

- FOREIGN KEY: 모든 참조는 ON DELETE RESTRICT ON UPDATE CASCADE

## 17. chatting (고객-직원 채팅 세션)

**설명:** 고객-직원 채팅 세션을 저장하는 테이블입니다. Firebase Firestore와 연동하여 실시간 채팅을 지원합니다.

컬럼명	타입	설명	제약조건
chatting_seq	INT	채팅 세션 고유 ID	PRIMARY KEY, AUTO_INCREMENT
u_seq	INT	고객 번호	NOT NULL (논리적 참조 → user.u_seq)

컬럼명	타입	설명	제약조건
fb_doc_id	VARCHAR(100)	Firebase Firestore 문서 ID	
s_seq	INT	담당 직원 번호	(논리적 참조 → staff.s_seq, 선택 사항)
created_at	TIMESTAMP	채팅 세션 생성 일시	NOT NULL, DEFAULT CURRENT_TIMESTAMP
is_closed	TINYINT(1)	채팅 종료 여부	NOT NULL, DEFAULT 0 (0: 진행중, 1: 종료)

관계:

- chatting.u\_seq → user.u\_seq (논리적 참조, FK 없음)
- chatting.s\_seq → staff.s\_seq (논리적 참조, FK 없음)

인덱스:

- idx\_chatting\_u\_seq: 고객별 채팅 세션 조회
- idx\_chatting\_s\_seq: 직원별 채팅 세션 조회
- idx\_chatting\_created\_at: 생성 일시 인덱스
- idx\_chatting\_is\_closed: 종료 여부 인덱스

제약조건:

- 외래 키 없음: 실시간 채팅 성능 최적화 및 Firebase Firestore 연동을 위해 논리적 참조만 사용
- 외래 키를 사용하지 않는 이유:
  - 실시간 채팅 성능 최적화
  - Firebase Firestore와의 외부 시스템 연동 용이성
  - 탈퇴한 user의 chatting 기록 보존 필요
  - 애플리케이션 레벨에서 JOIN으로 검증 가능

## 🔗 주요 조인 패턴

### 1. 제품 상세 정보 조회 (모든 카테고리 + 제조사)

```
SELECT
  p.p_seq,
  p.p_name,
  p.p_price,
  p.p_stock,
  p.p_image,
  kc.kc_seq,
  kc.kc_name,
  cc.cc_seq,
  cc.cc_name,
  sc.sc_seq,
  sc.sc_name,
  gc.gc_seq,
  gc.gc_name,
```

```

m.m_seq,
m.m_name,
m.m_phone,
m.m_address
FROM product p
JOIN kind_category kc ON p.kc_seq = kc.kc_seq
JOIN color_category cc ON p.cc_seq = cc.cc_seq
JOIN size_category sc ON p.sc_seq = sc.sc_seq
JOIN gender_category gc ON p.gc_seq = gc.gc_seq
JOIN maker m ON p.m_seq = m.m_seq
WHERE p.p_seq = ?

```

## 2. 고객별 구매 내역 조회

```

SELECT
pi.b_seq,
pi.b_price,
pi.b_quantity,
pi.b_date,
pi.b_status,
p.p_name,
p.p_image,
br.br_name,
br.br_address
FROM purchase_item pi
JOIN product p ON pi.p_seq = p.p_seq
JOIN branch br ON pi.br_seq = br.br_seq
WHERE pi.u_seq = ?
ORDER BY pi.b_date DESC

```

## 3. 수령 상세 정보 조회 (구매 + 고객 + 제품 + 지점)

```

SELECT
pic.pic_seq,
pic.created_at,
pi.b_price,
pi.b_quantity,
pi.b_date,
u.u_name,
u.u_phone,
p.p_name,
p.p_image,
br.br_name,
br.br_address
FROM pickup pic
JOIN purchase_item pi ON pic.b_seq = pi.b_seq
JOIN user u ON pic.u_seq = u.u_seq
JOIN product p ON pi.p_seq = p.p_seq

```

```
JOIN branch br ON pi.br_seq = br.br_seq
WHERE pic.pic_seq = ?
```

#### 4. 반품 상세 정보 조회 (수령 + 고객 + 직원 + 제품 + 지점)

```
SELECT
    ref.ref_seq,
    ref.ref_date,
    ref.ref_reason,
    ref.ref_re_content,
    u.u_name,
    u.u_phone,
    s.s_name,
    s.s_rank,
    pic.created_at,
    pi.b_price,
    pi.b_quantity,
    p.p_name,
    p.p_image,
    br.br_name,
    br.br_address
FROM refund ref
JOIN user u ON ref.u_seq = u.u_seq
JOIN staff s ON ref.s_seq = s.s_seq
JOIN pickup pic ON ref.pic_seq = pic.pic_seq
JOIN purchase_item pi ON pic.b_seq = pi.b_seq
JOIN product p ON pi.p_seq = p.p_seq
JOIN branch br ON pi.br_seq = br.br_seq
WHERE ref.ref_seq = ?
```

## 📌 주요 특징

### 소셜 로그인 지원

- **user 테이블:** `u_id`, `u_password` 필드 제거, `u_email` 필수화
- **user\_auth\_identities 테이블:** 로그인 수단별 인증 정보 분리 저장
  - 로컬 로그인: `provider='local'`, `provider_subject`에 이메일 저장
  - 구글 로그인: `provider='google'`, `provider_subject`에 구글 sub 저장
  - 카카오 로그인: `provider='kakao'`, `provider_subject`에 카카오 id 저장

### 재고 관리

- 중앙 재고 관리: `product.p_stock`으로 본사가 중앙 관리
- 대리점별 재고: 현재 미구현 (미래 확장 가능)

### 주문 그룹화

- **purchase\_item:** 분 단위로 주문 그룹화 (`b_date` 기준)

- 같은 분, 같은 고객, 같은 지점의 구매는 하나의 주문으로 처리

## 반품 처리

- **pickup 필수:** 반품은 반드시 수령(pickup) 후에만 가능
- **반품 사유:** `refund_reason_category` 테이블로 관리
- **직원 처리:** 모든 반품은 직원이 처리 (`s_seq` 필수)

## 발주/입고 프로세스

- **발주(request):** 직원이 발주 요청 → 팀장 결재 → 이사 결재
- **입고(receive):** 발주된 제품이 입고되면 `receive` 테이블에 기록
- **재고 반영:** 입고 시 `product.p_stock` 업데이트 (애플리케이션 로직)

## 🔍 인덱스 활용

모든 인덱스는 조인 쿼리 및 필터링 성능 향상을 위해 설계되었습니다:

- **고객 관련:** `idx_user_email`, `idx_user_phone`, `idx_purchase_item_u_seq`
- **지점/직원:** `idx_staff_br_seq`, `idx_purchase_item_br_seq`
- **제품 관련:** `idx_product_m_seq`, `idx_product_kc_seq`, `idx_product_cc_seq`,  
`idx_product_sc_seq`, `idx_product_gc_seq`
- **주문 상태:** `idx_purchase_item_b_status`, `idx_purchase_item_b_date`
- **수령/반품:** `idx_pickup_u_seq`, `idx_refund_u_seq`, `idx_refund_s_seq`



## 변경 이력

2025-12-30 김택권

- **ERD 1차 최종 반영:** 새로운 ERD 구조에 맞게 전체 스키마 재구성
  - `refund_reason_category` 테이블 추가
  - `user`, `staff`, `product` 테이블에 `created_at` 추가
  - `purchase_item`에 `b_status` 추가
  - `pickup`에 `u_seq` 추가
  - `refund`에 `ref_re_seq`, `ref_re_content` 추가

2025-12-31 김택권

- **소셜 로그인 지원:**
  - `user` 테이블 구조 변경 (`u_id`, `u_password` 제거, `u_email` 추가)
  - `user_auth_identities` 테이블 추가 (로그인 수단별 인증 정보)
- **카테고리 구조 변경:**
  - `ProductBase` → `kind_category`, `color_category`, `size_category`, `gender_category`로 분리
  - `Manufacturer` → `maker`로 변경
- **주문 구조 변경:**
  - `Purchase` 테이블 제거, `purchase_item` 단일 테이블로 통합
  - 분 단위 주문 그룹화 로직 추가

2026-01-01 김택권

- **ERD 링크 추가:** 데이터베이스 관계도 섹션에 DBML ERD 및 원본 Miro ERD 링크 추가

2026-01-03 김택권

- **product 테이블 UNIQUE 제약조건 수정:** `(cc_seq, sc_seq, m_seq) → (cc_seq, sc_seq, m_seq, p_name)`
  - 제품명을 포함하여 동일한 색상, 사이즈, 제조사라도 제품명이 다르면 별도 제품으로 등록 가능
- **purchase\_item.b\_status 값 형식 변경:** 한국어 문자열 → 숫자 문자열
  - '`0`' : 준비중
  - '`1`' : 준비완료
  - '`2`' : 수령완료
  - '`3`' : 반품완료
  - `NULL` : 상태 미정
- **Flutter 애플리케이션 호환성:** `b_status`는 숫자 문자열로 저장되며, Flutter에서 숫자로 파싱하여 사용
- **DB 검수 결과 반영:**
  - 실제 DB에 접속하여 스키마 및 데이터 검수 수행
  - product 테이블 UNIQUE 제약조건 확인: `uq_product_color_size_maker_name` 정확히 반영됨
  - purchase\_item.b\_status 값 확인: 모든 값이 숫자 문자열로 저장됨
  - refund 테이블 구조 확인: 모든 필드 존재 확인
  - 사이즈 카테고리 확인: 8개 생성, 7개 사용 (290 미사용)
  - 제품 개수 확인: 84개 정확히 생성됨
  - 반품 사유 카테고리 확인: 8개 정확히 생성됨

2026-01-04 김택권

- **chatting 테이블 추가:** 고객-직원 채팅 세션 관리
  - Firebase Firestore 연동 지원 (`fb_doc_id` 필드)
  - 외래 키 없이 논리적 참조만 사용 (성능 및 유연성 고려)
  - 실시간 채팅 성능 최적화를 위해 외래 키 제약 없음
  - 탈퇴한 user의 chatting 기록 보존 가능

---

문서 버전: 2.3

최종 수정일: 2026-01-04

최종 수정자: 김택권