

주문 → 결제 → KDS 데이터 흐름 예시 문서

이 문서는 손님이 한 테이블에서 메뉴 3개(커피 3잔)를 주문하고, 결제를 완료한 뒤 KDS로 전달되기까지의 전 과정을 더미 데이터 + 테이블 단위 설명으로 한 번에 이해할 수 있도록 정리한 예시 문서다.

이 문서 한 장만 보면,

- 주문 단위가 무엇인지
- 메뉴/옵션이 어디에 저장되는지
- 결제는 어떤 기준으로 묶이는지
- KDS에는 어떤 데이터가 전달되는지 를 전부 파악할 수 있다.

0. 주문 시나리오 (현실 예시)

- 테이블: **3번 테이블**
- 주문 메뉴 (총 3잔)
 1. 아메리카노 × 1잔
 - 옵션: 샷 추가
 2. 카페라떼 × 1잔
 - 옵션: 얼음 적게
 3. 바닐라라떼 × 2잔
 - 옵션: 샷 추가 + 얼음 많이

1. store_orders (주문 묶음 생성)

손님이 "주문하기" 버튼을 누르는 순간 생성되는 최상위 주문 컨테이너

```
store_orders
-----
store_order_id | table_no | total_price | status
-----
S0_1001       | 3        | 14500       | ORDERED
```

설명

- 메뉴/옵션 정보는 없음
- "3번 테이블의 이번 주문 전체"를 대표
- 이후 결제·주방·정산의 기준 단위

2. order_details (메뉴 단위 저장)

메뉴 1개 = order_details 1행

order_details

order_detail_id	store_order_id	menu_name	quantity	price_snapshot
OD_001	S0_1001	아메리카노	1	4500
OD_002	S0_1001	카페라떼	1	4500
OD_003	S0_1001	바닐라라떼	2	11000

설명

- 이 시점에서 이미 **메뉴 가격 + 옵션 가격이 합산된 최종 금액**이 고정됨
- 이후 메뉴 가격이나 옵션 가격이 바뀌어도 이 주문 금액은 변하지 않음
- KDS에서 “메뉴 단위”로 조리되는 기준

3. order_detail_options (옵션 스냅샷 저장)

선택된 옵션 1개당 1행

order_detail_options

order_detail_option_id	order_detail_id	option_name	price_snapshot
ODO_001	OD_001	샷 추가	500
ODO_002	OD_002	얼음 적게	0
ODO_003	OD_003	샷 추가	500
ODO_004	OD_003	얼음 많이	003
많이	0		얼음

설명

- 옵션은 **정의 테이블과 완전히 분리된 스냅샷 데이터**
- 옵션 삭제/이름 변경/가격 변경이 있어도 기존 주문에는 영향 없음
- KDS에는 이 데이터를 그대로 출력

4. 결제 생성 (payments)

손님이 결제를 완료하면 **store_orders** 기준으로 결제 1건 생성

payments

payment_id	store_order_id	method	total_amount	status	approved_at
------------	----------------	--------	--------------	--------	-------------

PAY_9001		SO_1001		CARD		14500		DONE		2026-01-12
----------	--	---------	--	------	--	-------	--	------	--	------------

설명

- 결제는 항상 주문 묶음(**store_orders**) 기준
- 부분 결제 / 추가 결제 확장 가능
- PG사 연동 정보는 이 테이블에만 존재

5. KDS로 전달되는 데이터 구조

KDS는 결제 완료된 주문(**store_orders**) 기준으로 **order_details + order_detail_options**를 조회한다.

KDS에 전달되는 개념 구조

[3번 테이블 주문]

- 아메리카노 (1잔)
 - 샷 추가
- 카페라떼 (1잔)
 - 얼음 적게
- 바닐라라떼 (2잔)
 - 샷 추가
 - 얼음 많이

중요한 포인트

- KDS는 옵션 정의 테이블을 볼 필요가 없음
- 주문 당시 확정된 데이터만 사용

6. 전체 흐름 요약 (한 눈에)

```
메뉴/옵션 선택
↓
store_orders 생성 (주문 묶음)
↓
order_details 생성 (메뉴 단위)
↓
order_detail_options 생성 (옵션 스냅샷)
↓
결제(payments)
↓
KDS 전달 및 조리
```

7. 이 구조가 실무에서 안전한 이유

- 주문과 결제가 분리되어 유연한 결제 정책 가능
- 가격/옵션 변경으로 인한 과거 주문 데이터 훼손 없음
- KDS는 단순 조회 구조로 안정성 높음
- 테이블 주문 / 포장 / 배달 구조로 확장 가능

이 문서는 **팀 내부 공유용 기준 예시 문서**로 사용하면 되며, ERD · API · KDS · 결제 로직을 설명할 때 동일한 참조 자료로 활용 가능하다.