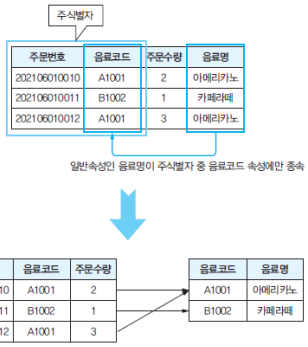
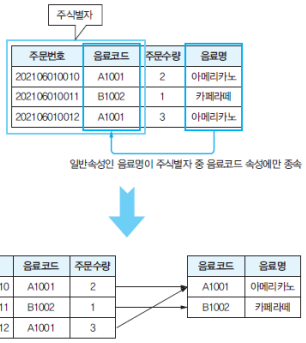
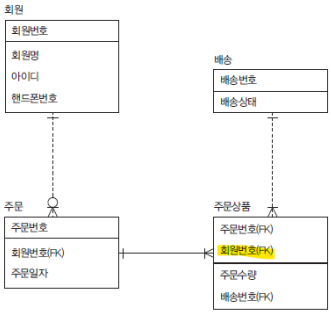
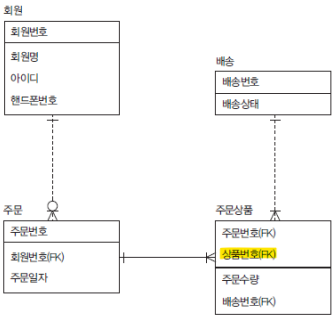
위치	오류유형	수정 전	수정 후
46p	개념,공식-설명	**(2) 제2정규형** 엔티티의 모든 일반속성은 반드시 모든 주식별자에 종속되어야 한다. 주식별자가 단일식별자가 아닌 복합식별자인 경우 일반속성이 주식별자의 일부에만 종속될 수 있는데 이런 경우 어떤 문제점이 발생하는지 알아보도록 하자.  그렇다면 실제 테이블로 2차 정규화를 시켜보자.	**(2) 제2정규형** 엔티티의 모든 일반속성은 반드시 모든 주식별자에 종속되어야 한다. 주식별자가 단일식별자가 아닌 복합식별자인 경우 일반속성이 주식별자의 일부에만 종속될 수 있는데 이런 경우 어떤 문제점이 발생하는지 알아보도록 하자.  이해를 돕기 위한 설명 추가 위 그림에서 음료코드와 음료명이 하나의 엔티티에 존재할 경우 데이터 이상 현상이 발생할 수 있다. 주문이 되지 않은 음료는 입력을 할 수 없는 입력 이상 현상과 음료명이 변경될 경우 해당 음료에 대한 주문 데이터가 모두 변경되어야 하는 수정 이상 현상이 그것이다. 그렇다면 실제 테이블로 2차 정규화를 시켜보자.
50p	문제-본문	**(4) 주의사항** 과유불급이란 말이 있듯이 적절한 정규화는 성능상 이롭지만 지나친 정규화는 오히려 성능 저하를 일으킬 수 있다. ① 회원의 배송상태를 조회하려면 여러 번의 JOIN을 해야 함  	**(4) 주의사항** 과유불급이란 말이 있듯이 적절한 정규화는 성능상 이롭지만 지나친 정규화는 오히려 성능 저하를 일으킬 수 있다. ① 회원의 배송상태를 조회하려면 여러 번의 JOIN을 해야 함 

<table>
<tr><th>위치</th><th>오류유형</th><th colspan="2">수정 전</th><th>수정 후</th></tr>
<tr><td rowspan="2">107p</td><td rowspan="2">문제-본문</td><td>수정 전</td><td colspan="2">
다음 쿼리는 COMPANY가 NULL인 행을 조회한다.

```
1 SELECT FIRST_NAME, LAST_NAME, EMAIL
2   FROM MEMBER
3  WHERE FIRST_NAME = Mark;
```

SQL Error [1722] [42000]: ORA-01722: 수치가 부적합합니다

(4) 부정 SQL 연산자

연산자	의미	예시
NOT BETWEEN A AND B	A와 B의 사이가 아님(A, B 미포함)	where col not between 1 and 10
NOT IN (LIST)	LIST 중 일치하는 것이 없음	where col not in (1, 3, 5)
IS NOT NULL	NULL 값이 아님	where col is not null
</td></tr>
<tr><td>수정 후</td><td colspan="2">
다음 쿼리는 COMPANY가 NULL인 행을 조회한다.

```
1 SELECT FIRST_NAME, LAST_NAME, COMPANY
2   FROM MEMBER
3  WHERE COMPANY IS NULL;
```

	FIRST_NAME	LAST_NAME	COMPANY
1	Leonie	Köhler	[NULL]
2	François	Tremblay	[NULL]
3	Bjørn	Hansen	[NULL]
4	Helena	Holý	[NULL]
5	Astrid	Gruber	[NULL]
6	Daan	Peeters	[NULL]
7	Kara	Nielsen	[NULL]
8	Fernanda	Ramos	[NULL]
9	Michelle	Brooks	[NULL]
10	Dan	Miller	[NULL]

(4) 부정 SQL 연산자

연산자	의미	예시
NOT BETWEEN A AND B	A와 B의 사이가 아님(A, B 미포함)	where col not between 1 and 10
NOT IN (LIST)	LIST 중 일치하는 것이 없음	where col not in (1, 3, 5)
</td></tr>
<tr><td>199p
마지막 줄</td><td>개념,공식-설명</td><td colspan="2">날짜별, 주문음료별, 판매사원별로 그룹핑 + 날짜별, 주문음료별로 그룹핑 + 판매사원별로 그룹핑 + 총합계의 결과가 나왔다. GROUP BY ROLLUP((ORDER_DT, ORDER_ITEM), REG_NAME)과 비교했을 때 날짜별, 주문음료별로 그룹핑이 더해진 것을 알 수 있다.</td><td>날짜별, 주문음료별, 판매사원별로 그룹핑 + 날짜별, 주문음료별로 그룹핑 + 판매사원별로 그룹핑 + 총합계의 결과가 나왔다. GROUP BY ROLLUP((ORDER_DT, ORDER_ITEM), REG_NAME)과 비교했을 때 판매사원별로 그룹핑이 더해진 것을 알 수 있다.</td></tr>
</table>

도서의 오류로 학습에 불편드린 점 진심으로 사과드립니다.
더 나은 도서를 만들기 위해 노력하는 시대교육그룹이 되겠습니다.