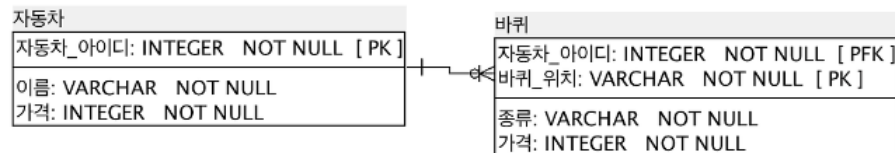


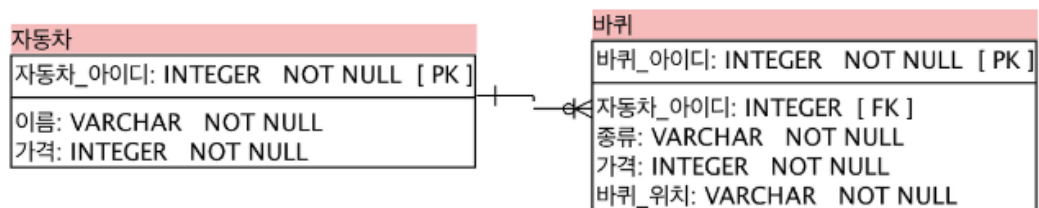
1. ERD

식별 관계 VS 비식별 관계

식별 관계



비식별 관계



식별관계

- 식별관계는 부모 엔티티의 기본키 또는 유니크 키를 자식 테이블이 자신의 **기본키**로 사용한다. 즉 반드시 부모 엔티티에 데이터가 존재 해야한다. 위 그림에서는 바퀴 엔티티의 데이터가 존재하려면, 자동차 엔티티 데이터가 반드시 존재해야 한다.
- 장점:
 - 식별 관계는 요구사항이 변하지 않을 때 강력한 데이터 정합성을 보장할 수 있다.
- 단점
 - 주 식별자 영역이 점점 비대해지며(식별관계는 부모 실체의 주 식별자를 자식 실체의 주 식별자로 계속하여 전파시키는 특성이 있기 때문에), 결과적으로 무의미한 외래 식별자들이 주 식별자 영역에 들어가는 경우도 많이 있음.
 - 이로 인해 요구사항이 변경됐을 때, 수정이 어려워진다.

비식별관계

비식별 관계는 부모 엔티티의 기본키 또는 유니크 키를 자식 테이블이 자신의 **외래키**로 사용한다. 따라서 부모 데이터가 없어도 자식 데이터는 독립적으로 생성될 수 있다. (외래키가 NULL 값이 됨)

- 장점
 - 변경되는 요구사항을 유동적으로 수용할 수 있다
 - 부모 데이터와 독립적인 자식 데이터를 생성할 수 있다
- 단점
 - 데이터 정합성을 지키기 위해서는 별도의 비즈니스 로직이 필요하다.
 - 즉, 데이터 무결성을 보장하지 않는다.

식별 관계보단 비식별 관계를 사용하자!

데이터베이스 설계 관점에서 보면 식별 관계보다는 비식별 관계를 선호합니다 👍

- 식별 관계는 부모 테이블의 기본 키를 자식 테이블로 전파하면서 자식 테이블의 기본 키 칼럼이 점점 늘어납니다. 이는 SQL 의 성능을 저하시킵니다.
- 식별 관계는 일반적으로 복합 기본 키를 만들어야 하는 경우가 많습니다.
- 식별 관계를 사용할 때 기본 키로 **자연 키** 칼럼을 조합하는 경우가 많습니다.
비즈니스 요구사항은 시간이 지남에 따라 언젠가는 변합니다. 따라서 비즈니스와 관련 없는 **대리 키**를 사용하는 것이 좋습니다.
- 식별 관계에서 자식 테이블은 부모 테이블의 기본 키를 자신의 기본 키로 사용함으로써 테이블 구조가 유연하지 못합니다..
- 비식별 관계의 기본 키는 주로 대리 키를 사용합니다. JPA 에서는 `@GeneratedValue` 와 같이 대리 키를 생성하는 편리한 방법을 제공합니다.
- 비식별 관계를 사용시 식별자 칼럼이 하나이기에 매핑하기 편리합니다.

하지만 식별 관계에도 상황에 따른 장점은 존재합니다.

먼저 기본 키 인덱스를 사용하기 편리하며 상위 테이블의 기본 키 칼럼을 자식 테이블들이 가지고 있으므로 특정 상황에 조인이 필요가 없습니다.

결론은 가능한 비식별 관계를 사용하고 기본 키는 Long 타입의 대리 키를 사용하는 것입니다 또한 비식별 관계에서도 Null 값을 허용하지 않는 필수적 비식별 관계를 사용하는 것이 좋은데 이는 내부 조인을 사용할 수 있기 때문입니다. !!