

# 정규화

정규화(Normalization)는 데이터베이스 설계 시 중복을 최소화하고 데이터 의존성을 관리하기 위한 과정. 주로 제1, 제2, 제3 정규화가 사용되며, 각 정규화는 특정 종류의 데이터 의존성 문제를 해결할 수 있음.

## 1. 제1 정규화 (1NF - First Normal Form):

- 각 테이블의 모든 속성 값은 **원자적(Atomic)**이어야 한다.
- 즉, **각 셀에는 하나의 값만 들어가야 함**을 의미.

(비정규화)

ID	학생	과목
1	Steve	Java, Python
2	Tom	C
3	Alvin	Python

(제 1 정규화)

ID	학생	과목
1	Steve	Java
1	Steve	Python
2	Tom	C
3	Alvin	Python

## 2. 제2 정규화 (2NF - Second Normal Form):

- 제1 정규화를 만족하면서, **모든 비주요 속성이 기본 키에 대해 완전 함수 종속이어야 한다**.
- Composite primary key(학생 + 과목)의 종속이어야 한다.

(제 1 정규화)

ID	학생	과목	강사	성적
1	Steve	Java	Kim	A
1	Steve	Python	Lee	B
2	Tom	C	Choi	C+

3	Alvin	Python	Lee	A+
---	-------	--------	-----	----

(제 2 정규화)

ID	학생	과목ID	성적
1	Steve	100	A
1	Steve	102	B
2	Tom	101	C+
3	Alvin	102	A+

과목ID	과목	강사
100	Java	Kim
101	C	Choi
102	Python	Lee

### 3. 제3 정규화 (3NF - Third Normal Form):

- 제2 정규화를 만족하면서, 이행적 종속이 없어야 한다.
- 이행종속이란?  $A > B, B > C$  이므로  $A > C$  가 되는 관계

(제 2 정규화)

과목ID	과목	강사	이메일
100	Java	Kim	<a href="mailto:kim@gmail.com">kim@gmail.com</a>
101	C	Choi	<a href="mailto:choi@naver.com">choi@naver.com</a>
102	Python	Lee	<a href="mailto:lee@daum.net">lee@daum.net</a>
103	C++	Choi	<a href="mailto:choi@naver.com">choi@naver.com</a>
104	C#	Choi	<a href="mailto:choi@naver.com">choi@naver.com</a>

(제 3 정규화)

과목ID	과목	강사
100	Java	Kim
101	C	Choi
102	Python	Lee
103	C++	Choi
104	C#	Choi

강사ID	강사	이메일
------	----	-----

1	Kim	<u>kim@gmail.com</u>
2	Choi	<u>choi@naver.com</u>
3	Lee	<u>lee@daum.net</u>

1. 제1정규화를 수행하기 위해서는 제공된 데이터가 중복되거나 다중 값 속성을 가지고 있어야 함. 이후 중복된 데이터를 분리하여 각 값에 하나의 행이 있는 형태로 구조화.
2. 제2정규화를 수행하기 위해서는 제1정규화가 이미 완료된 상태여야 함. 제1정규화를 통해 모든 열이 원자적인 값을 가지게 되었기 때문에 각 열이 기본 키에 완전히 종속되도록 데이터를 구조화할 수 있음.
3. 마찬가지로 제3정규화를 수행하기 위해서는 제2정규화가 이미 완료된 상태여야 함. 제2정규화를 통해 부분 함수 종속성을 제거하여 각 열이 기본 키에 완전히 종속되도록 했기 때문에 제3정규화를 통해 이행적 종속문제를 해결 할 수 있음.

**정규화는 데이터 모델의 일관성을 유지하고 데이터베이스의 성능을 향상시키는데 중요한 역할을 함. 따라서 각 단계별로 순차적으로 진행되어야 하며, 그 전 단계가 완료된 상태에서 다음 단계로 넘어가야 함.**