

11. 다음으로

#0.강의/2.데이터베이스로드맵/4.설계2

전체 학습 목차

1. 강의 소개와 수업 자료

2. 공통 코드 설계

- /공통 코드가 필요한 이유
- /공통 코드 테이블 설계
- /공통 코드를 더 범용성 있게 - 그룹화 설계
- /공통 코드와 추가 속성
- /공통 코드의 단점
- /공통 코드의 단점 해결 방안1
- /공통 코드의 단점 해결 방안2
- /공통 코드 vs 애플리케이션 ENUM 1
- /공통 코드 vs 애플리케이션 ENUM 2
- /공통 코드 vs 애플리케이션 ENUM 3
- /공통 코드 설계와 비즈니스 설계의 차이
- /정리

3. 계층 구조 설계

- /계층 구조 설계가 필요한 이유
- /인접 리스트 모델
- /계층 구조 조회의 어려움
- /CTE와 재귀 쿼리 1
- /CTE와 재귀 쿼리 2
- /폐쇄 테이블 모델 1
- /폐쇄 테이블 모델 2
- /정리

4. 데이터 변경 이력 설계1

- /데이터 변경 이력 설계가 필요한 이유
- /변경 추적 컬럼 - 기본
- /변경 추적 컬럼 - 변경 사유

- /변경 추적 컬럼 - 감사(Audit) 컬럼
- /정리

5. 데이터 변경 이력 설계2

- /컬럼에 이전 값 보관 방식
- /현재 테이블로 이력 관리 - 시작
- /현재 테이블로 이력 관리 - 단점 1
- /현재 테이블로 이력 관리 - 단점 2
- /현재 테이블로 이력 관리 - 유효 기간
- /전체 행 스냅샷 이력 테이블 - 시작
- /전체 행 스냅샷 이력 테이블 - 주의점
- /전체 행 스냅샷 이력 테이블 - 유효 기간
- /전체 행 스냅샷 이력 테이블 - 한계
- /컬럼 단위 변경 로그 테이블
- /공통 이력 테이블
- /정리

6. SOFT DELETE

- /SOFT DELETE가 필요한 이유
- /SOFT DELETE - is_deleted 컬럼 방식
- /SOFT DELETE - deleted_at 컬럼 방식 1
- /SOFT DELETE - deleted_at 컬럼 방식 2
- /SOFT DELETE vs HARD DELETE
- /SOFT vs HARD vs STATUS
- /SOFT DELETE와 이력 테이블
- /SOFT DELETE와 인덱스 설계
- /정리

7. 통계 테이블 설계

- /통계 데이터와 성능 문제
- /통계 테이블 설계
- /주간, 월간 통계의 효율적인 처리
- /실시간 통계와 하이브리드 설계
- /떡등성 설계
- /마이크로 배치

- /UPSERT 최적화
- /정리

8. 상속 관계 설계

- /상속 관계 설계 - 문제 상황
- /구현 클래스마다 테이블 전략
- /구현 클래스마다 테이블 전략의 장단점
- /단일 테이블 전략
- /단일 테이블 전략의 장단점
- /조인 전략
- /조인 전략의 장단점
- /정리

9. 동적 설계 - EAV

- /EAV 설계 - 기존 방식의 한계와 새로운 접근법
- /EAV 패턴 소개
- /EAV 실습 - 쇼핑몰 상품 속성 관리
- /EAV 패턴 개선 - 속성 정의 테이블
- /EAV의 장단점과 사용 시 주의사항
- /EAV 실무 활용 사례
- /정리

10. JSON 설계

- /EAV의 한계와 JSON의 필요성
- /JSON 문법
- /MySQL에서 JSON 사용하기 1
- /MySQL에서 JSON 사용하기 2
- /JSON 활용 - 다양한 실무 사례 1
- /JSON 활용 - 다양한 실무 사례 2
- /JSON 인덱스와 성능 최적화 1
- /JSON 인덱스와 성능 최적화 2
- /JSON 설계의 장단점과 한계
- /JSON 사용 가이드라인
- /관계형 데이터베이스 vs NoSQL
- /정리

로드맵 소개

김영한의 실전 데이터베이스 로드맵

1. 김영한의 실전 데이터베이스 입문 - SQL부터 차근차근 (오픈)
2. 김영한의 실전 데이터베이스 - 기본편 (오픈)
3. 김영한의 실전 데이터베이스 - 설계 1편, 현대적 데이터 모델링 완전 정복 (오픈)
4. 김영한의 실전 데이터베이스 - 설계 2편, 실무에서 반드시 마주치는 9가지 설계 패턴 (오픈)
5. 김영한의 실전 데이터베이스 - 성능 최적화와 고급 기능편 (2026년 예정)

백엔드 개발자 로드맵 소개



김영한 백엔드 개발자 자바 스프링 JPA 실무 로드맵

백엔드 개발자 로드맵 소개 영상 링크: <https://youtu.be/ZgtvcyH58ys>

김영한의 실전 자바 로드맵

- 자바를 완전히 마스터 할 수 있는 로드맵
- URL: <https://www.inflern.com/roadmaps/744>

스프링 완전 정복 로드맵

- 스프링을 완전히 마스터 할 수 있는 로드맵
- URL: <https://www.inflern.com/roadmaps/373>

스프링 부트와 JPA 실무 완전 정복 로드맵

- 최신 실무 기술로 웹 애플리케이션을 만들어보면서 학습

- URL: <https://www.infllearn.com/roadmaps/149>

하고 싶은 이야기

패턴을 알면 설계가 보인다

- 1편에서 데이터 모델링의 기본기를 배웠다면, 2편에서는 실무에서 반복되는 문제들의 해결 패턴을 배웠다.
- 공통 코드, 계층 구조, 이력 관리, 상속 관계... 이런 문제들은 거의 모든 프로젝트에서 만난다.
- 패턴을 모르면 매번 처음부터 고민하고, 시행착오를 겪는다.
- 패턴을 알면 검증된 해결책으로 빠르게 설계하고, **더 중요한 비즈니스 로직에 집중**할 수 있다.

정답은 없다, 트레이드오프가 있을 뿐

- 모든 설계 패턴에는 장점과 단점이 있다.
- 단일 테이블 전략 vs 조인 전략, EAV vs JSON, SOFT DELETE vs HARD DELETE
- "이게 정답이다"라고 말하는 사람을 조심하라.
- 우리 서비스의 특성, 데이터의 양, 팀의 역량, 변경 가능성을 고려해서 선택해야 한다.
- **트레이드오프를 이해하고 근거 있는 선택을 하는 것이 진짜 설계 능력이다.**

설계는 미래를 예측하는 것이 아니다

- 완벽한 설계를 하려고 너무 오래 고민하지 마라.
- 미래의 요구사항을 100% 예측하는 것은 불가능하다.
- 중요한 것은 변경에 유연하게 대응할 수 있는 구조를 만드는 것이다.
- **과도한 범용성은 오히려 복잡성만 증가시킨다.**
- 현재의 요구사항을 명확히 해결하고, 확장 가능성을 열어두는 **균형이 중요하다.**

관계형 데이터베이스의 힘을 믿어라

- NoSQL, JSON... 새로운 것들이 계속 나온다.
- 하지만 관계형 데이터베이스는 50년 넘게 검증된 기술이다.
- 대부분의 비즈니스 문제는 관계형 모델로 충분히 해결할 수 있다.
- 새로운 기술은 관계형 모델의 한계가 명확할 때 선택하라.
- 기본기가 탄탄하면 어떤 기술이 나와도 빠르게 적응할 수 있다.

끝으로

- 설계 능력은 하루아침에 생기지 않는다.
- 직접 설계하고, 실패하고, 개선하는 과정을 반복해야 한다.
- 이 강의에서 배운 패턴들을 실무에 적용해 보라.
- 처음에는 어색하겠지만, 몇 번 해보면 자연스럽게 된다.

- 여러분의 설계 능력이 한 단계 성장하기를 진심으로 응원한다.

하고 싶은 이야기 정리 링크

개발 인생 전반의 이야기

EO 인터뷰 영상

- 한국 개발자 최고 1타강사 김영한의 인생 [1부]: https://youtu.be/_HTj5b59Em0
- 한국 개발자 최고 1타강사 김영한의 인생 [2부]: <https://youtu.be/MNyNRraMU8Y>

개발바닥 - 시골 청년 개발왕 되다

- 1편: <https://youtu.be/Pb69UQ6f8n0>
- 2편: <https://youtu.be/b4QP5RsuJts>
- 3편: <https://youtu.be/l0h1pQ96u2g>

취업과 이직에 대한 고민

인프콘 - 어느 날 고민 많은 주니어 개발자가 찾아왔다, 성장과 취업, 이직 이야기

- <https://youtu.be/QHlyr8soUDM>

인프런 최초 20만 명 달성 기념 QA

- <https://youtu.be/psXdWq008DA>

인프런 최초 30만 명 달성 기념 QA

- <https://inf.run/81ogv>