예상 질문)

1. 서버와 스키마는 통합하였지만, 테이블은 통합하지 않은 이유

(왜냐하면, 테이블 또한 정보시스템의 하위 요소이기 때문)

첫째, 이들은 같은 정보시스템 내의 구성요소인 테이블에 비해 중요성이 상대적으로 덜하여 통합을 고려하게 되었습니다. (상대적인 중요도)

\* 상대적인 중요도 (서버, 스키마 VS 테이블 – 테이블을 따로 관리하겠다고 요구에서도 존재)

둘째, 정보시스템구성 통합을 하게 된 이유는 서버와 스키마가 정보시스템구성요소로 동일한 성질을 가지고 있다고 생각했으며, 개별 엔터티의 공통속성의 존재와 적은 속성 수 때문에 통합을 고려했다. 통합을 통해 전과 달리 테이블로 상속되는 식별자의 수 또한 줄일 수 있었다.

2) 정보시스템구성이 순환관계를 가지지 않고 내부에 서버와 스키마가 직접 관계를 가지는 이유?

( 순환관계가 계층적 구조 (순환 VS 직접 관계) )

-- 순환을 하게 되면 서버끼리도 순환을 or 스키마끼리도 순환을 돌 수 있다고 생각하여 명확성을 위해 서버와 스키마를 관계를 직접 만들어 주었다.

3) 통합테이블이력에 관해.

테이블을 기준으로 하여 순번에 따라 테이블의 외적인 부분이라고 볼 수 있는 [테이블] 엔터티랑 관련된 것일 수도 있고, 테이블 내적인 부분인 [구성컬럼]엔터티랑 관련 되어있는 것일 수도 있다.

테이블변경요소구분에서 테이블(테이블의 외적인 정보가 바뀌는 경우) & 구성컬럼(테이블 내의 내적인 요소 컬럼 순서 PK순서 NOTNULL여부를 관리하기 위해) =>

case) 만약 1번 테이블의 외적인 부분이 변경되는 경우에는,

통합테이블이력순번 - 1

변경이유구분 -> 변경

테이블변경요소구분 -> 테이블

변경하고자 하는 속성 작성

case) 1번 테이블의 컬럼이 변경되면

통합테이블이력순번 - 2

테이블변경이유구분 -> 변경

테이블변경요소구분 -> 구성컬럼

『컬럼에 대해서 추가 / 삭제 / 변경될 때마다

삭제되는 경우에는 테이블 변경 요소 구분 내의 **구성컬럼의 속성을 다 기존 값으로 채우고** 삭제를 하는지 or [Default 값으로 채워서 넣는지]

=> 복구?? 로그?? 』

4) 컬럼과 컬럼그룹 분리, 테이블 구성 컬럼으로 아크 관계.

컬럼과 컬럼그룹을 통합했을 때 엔터티를 대표하는 식별자를 명확하게 정의할 수 없었고,

통합하는데 적절한 속성 명을 정의하는데 어려움이 있었습니다..

속성도 차이가 나고 컬럼들은 단순히 모아놓은 것이라고 생각해서.??

아크 관계를 사용한 이유는 컬럼 단독으로 있는 테이블에 소속되는 것과 컬럼그룹에

속해 있는 컬럼이 테이블에 속해지는 것을 표현하기 위해서이다.

5) 목표테이블 선후 관계만 이력이 같이 있는 이유

목표테이블 선후관계는 다른 엔터티들과 달리 관계를 가지는 자식이 없어서

이력을 같이 관리해도 효율적으로 관리된다고 생각했다.

7) 매핑변환일 때 신구 코드그룹 매핑정보로부터 관계가 이어지는 이유

상위 단계에 있는 엔터티인 신구코드그룹매핑정보를 참조하면 그 하위에 있는 상세한 신구코드매핑정보를 조인하여 알 수 있다고 생각했다.

8) 스크립트의 테이블매핑정보와 컬럼매핑정보로부터 오는 관계 설명

스크립트는 테이블 매핑정보를 기준으로, 때에 따라 컬럼매핑정보를 참조하여 작성한다.

그러므로 테이블매핑정보와는 식별관계를, 컬럼매핑정보와는 비식별관계를 갖는다.

테이블 매핑정보만을 필요로 할 때는 테이블 매핑정보만을 참조해서 스크립트를 작성하고, 테이블 매핑정보 아래의 컬럼매핑정보까지 필요로 할 때는 컬럼매핑정보까지 참조.

테이블 매핑정보를 기준으로 하여 그 테이블매핑정보의 하위에 있는 컬럼 매핑정보라는 제약을 둚..?

9) Job 실행결과 종합에 관한 설명.

(스크립트와 유사) 두 개의 실행결과는 JOB의 결과를 결정하는 엔터티들이기 때문에 개별 관리보다 통합함으로써 JOB실행결과종합 엔터티로 나타냈다.

JOB실행결과 엔터티에서 JOB엔터티를 기준으로 실행단계가 JOB구성스크립트이면 JOB구성스크립트의 엔터티를 참조하며, 전체 JOB일 경우에는 참조하지 않는다.

그래서 스크립트ID 속성은 선택속성으로 가져오게 된다.

JOB실행결과종합 엔터티는 JOB실행 단계에 따라 JOB구성스크립트를 실행하여 실행결과를 추출해내고 JOB내의 모든 스크립트를 실행한 후 마지막 단계로써 JOB의 결과를 추출한다.

이 엔터티를 통해 JOB내의 JOB구성스크립트가 하나 이상 실패하게 되면 JOB구성스크립트 실행 후 마지막 단계로 JOB실행결과를 실패하는 것으로 나타낸다.

예를들어 , JOB에 구성된 스크립트가 5개 있다고 하면

1번 JOB에 1째실행의 1단계 2단계 3단계 4단계 5단계는 각각의 구성된

스크립트에 관한 결과로서 JOB구성스크립트를 참조하여 나타내고,

6단계는 JOB구성스크립트를 참조하지 않고 JOB만을 참조하여 JOB전체의 실행결과를 나타낸다.

- 실체 검토 : 실체 정의의 명확성, 식별자 정의의 정확성 등

- 관계 검토 : 관계 형태의 정확성, 관계명 정의의 정확성 등

- 속성 검토 : 속성 정의의 정확성, 중복값 존재 여부, 추출값 정의의 정확성 등

- 정규화 : 제 1,2,3 정규화 적용

- 논리 DB 설계서의 실체, 관계선 등의 재배치 및 정리