

후기

고연수

과제를 받았을 때는 EMPLOYEES 테이블의 데이터를 옮기는 것이 가장 어려울 거라 생각했습니다. 하지만 막상 시작하니, 107개의 데이터를 옮기는 것은 그리 어려운 일이 아니었습니다. 그보다 테이블의 구조를 파악하고 각 제약조건의 특징을 고려해 테이블을 생성하는 것이 더 큰 문제였습니다.

특히 DEPARTMENTS 테이블과 EMPLOYEES 테이블에 FK 제약조건을 적용하려 할 때 문제가 발생해, 이를 해결하는데 시간이 소요되었습니다. FK가 참조할 컬럼을 다시 선정하고 테이블 간 상호 참조로 인한 테이블 생성 오류를 해결하는 것이 쉽지는 않았지만, 이 과정이 제약조건의 특징에 대해 다시 한 번 고민하는 계기가 되어 주었습니다.

자바 세미 프로젝트를 할 때도 느꼈지만, 내용을 쌓아올리는 것보다도 그 토대를 마련하는 것이 무엇보다 중요하다는 것을 실감할 수 있었습니다. 그리고 이를 위해서는 기본적인 내용을 철저하게 공부하고 활용할 수 있어야 하니 공부를 더 열심히 해야겠다는 교훈도 얻었습니다.

김인교

과제 실습 시작 당시엔 단순히 테이블 7개만 생성하는 간단한 실습이라 생각했습니다. 하지만 직접 만들어보니 테이블이 쉽게 생성되지 않았고 부모-자식 테이블 관계가 형성되면서 여러 가지 오류가 발생해 문제에 대한 의도 파악을 할 수 있었습니다. 마치 자바 프로젝트를 할 때 객체지향 클래스 구성하던 것 같은 느낌을 받았습니다.

특히 가장 어려웠던 문제는 DEPARTMENTS 테이블과 EMPLOYEES 테이블을 만들 때 참조관계를 설정하다보니 서로가 서로를 상호참조하게 되는 상황이 만들어졌고 서로가 서로의 부모테이블이 되어 먼저 생성해야 할 테이블을 정할 수 없게 되는 상황이 되었습니다. 그래서 DEPARTMENTS 테이블의 FK 제약조건 부여를 테이블 생성 이후로 미루어 해당 문제를 해결 하였습니다

결과적으로 과제 수행하다보니 난해했던 참조관계에 대해 공부한 내용을 토대로 팀원들과 적용해보며 공부할 수 있어서 유익했습니다. 혼자 했으면 막막했을 과제를 팀원들과 함께 수행하니 굉장히 수월했고 또한, 여러가지를 배울 수 있던 시간이었습니다.

정미경

팀과제 진행하기 전 데이터베이스 언어는 이전 자바와 같은 프로그래밍 언어보다 직관적인 언어로 코드를 작성하는 것 자체도 훨씬 쉽게 느껴졌고, 작성하는 구문도 어떤 기능이 들어있는 무언가를 만들기 보다는 데이터가 들어갈 표를 만드는 것이기 때문에 주어진 과제를 해결하는 데 시간이 오래 걸릴 거라는 것도 그 안에서 해매게 되리란 것도 시작하기 전엔 전부 생각지도 못한 일들이었습니다.

처음에는 각자 테이블을 나눠 맡아 테이블을 생성하는 구문과 insert 구문을 작성한 후 그것을 취합하여 실행만 하면 된다고 여겼는데 막상 팀원들이 작성한 코드를 모두 취합한 후 실행 하기 시작했을 땐 생각지도 못한 에러문구들로 인해 한참을 고민해야 했습니다.

테이블을 전부 생성한 후 데이터를 삽입하는 과정에서 일단 테이블 자체도 제약조건에 맞지 않은 순서로 생성을 시도하면

참조하고 있는 테이블이 없다는 에러와 함께 테이블 생성이 되지 않았고, 그렇다고 다른 테이블을 먼저 만들기엔 그 테이블도 또 다른 참조하고 있는 테이블에 의해 생성할 수 없는 상태였습니다.

이러한 오류들의 중심에 있던건 외래키였는데 이 덕분에 이번 과제에서 외래키에 대해 전보다 많이 알게 된 것 같습니다. 결국 에러들을 해결하고 팀과제를 무사히 완성시키긴 하였지만 사실 어떻게 완성된 건지는 테이블이 모두 완성되고 나서야 이해할 수 있었습니다.

나중에 든 생각은 초반엔 과제를 좀 더 빨리 끝내기 위해선 길게 생각하기 보다 일단 만들면서 에러가 나오면 그때 그때 해결하면서 하는게 낫다라고 생각했는데 데이터베이스의 경우엔 그렇게 하면 오히려 더 멀리 돌아가게 된다는 사실이었 습니다.

추가로 각 테이블간의 관계에 대해 즉, 어떤 테이블이 다른 테이블을 참조하고 있다면 논리적으로 왜 그 테이블을 참조해야 하는지와 왜 참조하고 있는지에 대해 테이블을 만들기 전 먼저 생각해보고 정리하는데 시간을 많이 들였다면 더 수월 하게 끝낼 수 있는 과제였겠다라는 생각을 하였습니다.

최나윤

원본 테이블과 동일한 구조, 동일한 데이터를 가지는 새 테이블을 만들려고 하니 생각보다 신경 쓸 게 많았습니다.

테이블 생성 전, 원본 테이블 조회를 통해 데이터 구조 파악, 뷰를 통해 제약조건 확인, 디폴트값 확인을 먼저 했습니다. 또한, 추가 데이터 입력 시 부여하는 번호(순서)를 편의성을 위해 시퀀스가 있는지 확인한 후, 시퀀스가 있는 원본 테이블에 한해 시퀀스도 생성을 했습니다. 원본 테이블의 제약조건을 확인한 후에 새 테이블의 제약조건을 동일하게 걸어주고 생성을 했지만 DEPARTMENTS1 테이블의 제약조건에서 에러가 발생했습니다.

고민하다보니 두 테이블의 공통 컬럼명인 MANAGER_ID 의 제약조건이 둘 다 포린키(FK)였습니다. 포린키가 있으면 참조하는 컬럼에 프라이머리키나 유니크 제약조건이 걸려있어야하는데 이상했습니다. 깊게 생각해보니 DEPARTMENTS1 테이블의 MANAGER_ID 컬럼은 EMPLOYEE_ID 컬럼을 갖는 EMPLOYEES1 테이블이 생성되고 데이터가 입력된 후에 참조를 해야 에러가 발생하지 않을 것 같았습니다. EMPLOYEE_ID 라는 직원번호가 존재해야 상관번호인 MANAGER_ID 에 해당하는 직원번호를 다시 조회할 수 있기 때문입니다. 따라서 EMPLOYEES1 테이블의 EMPLOYEE_ID 컬럼을 참조하는 DEPARTMENTS1 테이블의 MANAGER_ID 에 포린키라는 제약조건을 테이블 생성과 데이터 입력이 끝난 후에 추가해줘야 에러가 해결될거라고 생각했고, 이렇게 하니 에러가 해결되었습니다.

이번 과제 수행을 통해 꼭 같은 이름의 컬럼을 통해서만 참조할 수 있는 게 아니라는 걸 다시 한 번 알았고, 관계형 데이터베이스에 대해 더 잘 이해할 수 있는 시간이었습니다. 또한, 어려웠지만 해결하면서 데이터베이스에 재미를 느꼈고 더 자세히 공부해보고 싶다는 생각을 하는 계기가 되었습니다.

.....

.....