

1조 후기

고연수

프로젝트에 앞서 가장 걱정스러웠던 것은 프로젝트를 모두 비대면으로 진행해야 한다는 점이었습니다. 모두의 환경이 일정한 것이 아니어서 팀원들의 상황을 확인할 때 어려움이 있었습니다. 하지만 한 번 함께 했던 팀원들이라 그런지 자바 세미 프로젝트 때보다는 수월하게 의견 교환을 할 수 있었던 것 같습니다. 덕분에 우려했던 것보다는 편안하게 프로젝트를 진행했습니다.

프로젝트를 진행하면서는 직전에 배운 것이 자바라 그런지 자연스럽게 자바와 비교하게 되었습니다. 자바는 기능을 구현하기 위해 논리 구조를 쌓아올리는 것이 어려웠는데, DB는 사용자의 편의를 고려하여 다양한 상황을 가정해야 한다는 점이 어려웠습니다. 특히 요구 분석서에 나와 있지는 않지만, 우리가 가정하여 조건을 걸어야 할 때 어디까지 해야 좋을지 판단하는 것이 힘들었습니다. 팀원들의 의견이 옳고 그른 것이 아니라, 다 나름의 장단점이 있어서 해당 부분을 논의할 때 시간이 많이 들었습니다. 이 부분이 유독 어려웠던 것은 DB는 잘못 구성하면 처음부터 다시 테이블을 구성하고 제약조건을 걸어야 한다는 것이 부담스러워서 었던 것 같습니다. INSA 테이블을 다 같이 재구성한 경험을 바탕으로 비교해 가며 진행했지만, 경험이 더 필요할 것 같다는 생각을 했습니다.

이번 세미 프로젝트는 요청서를 선생님이 1차로 분석해 주셨습니다. 덕분에 키워드를 파악하는 것이 수월하여 테이블을 구성할 때 많은 도움을 받았습니다. 그런데 실무에서는 클라이언트로부터 날 것의 요청서를 받게 될 테니, 이를 분석하는 연습이 필요해 보입니다. 조금 더 연습할 수 있는 기회가 있으면 좋겠습니다.

김인교

프로젝트를 진행하며 요구분석서를 통해 업무파악이 얼마나 중요한지 깨달았고, 데이터의 흐름, ERD 구축 방법 등 실질적인 데이터베이스 구축 방법을 배울 수 있어서 개인적으로 얻은 점이 많은 프로젝트 였다고 생각합니다

팀단위로서도 얻은 점이 많았습니다. 1차 때는 다소 개인플레이를 했던 터라 팀 프로젝트의 어려움을 느낄 수 있었다면 이번엔 2차 팀프로젝트라 그런지 팀원끼리 부족한 점은 이끌어주고 좋은 점은 배워가는 훌륭한 협동심을 느낄 수 있었습니다.

결과적으로 이러한 협동은 혼자서는 생각하기 힘들 수 있는 점을 5명이 같이 고민하니 쉽게 풀리기도 하고 기상천외한 의견도 나와 코드적으로도 더 좋은 결과물을 내놓을 수 있었습니다.

이러한 경험을 토대로 앞으로도 좋은 팀프로젝트를 진행할 수 있을 거 같습니다.

민찬우

오라클 데이터베이스를 공부하고나서 세미 프로젝트를 진행할때 걱정되는게 많았다. 자바와 달리 조금 더 조심하게 다뤄야 하기 때문에 조금더 세심하게 프로젝트를 진행했다. 그리고 수업을 들을 땐 이해됐지만 전체적인 쿼리문을 작성 할 때 오라클 내부 구조에 대한 이해가 부족해서 내가 쿼리문을 작성해서 제대로 구현할수 있을지에 대한 생각을 많이 했다.

하지만 오라클을 처음 배울때 부터 들었던 설계 단계의 중요함을 생각하고 2~3일 ERD 설계를 하면서 구조가 잡히기 시작했고 어렵고 이해가 안 갔던 것들이 차츰 풀리기 시작했다. 전체적인 틀을 먼저 잡고 그에 따라 차츰 차츰 구체화 되는 과정이 신기했다. 연결지어진 테이블 끼리의 관계가 어떻게 되고 이것들이 쿼리문을 통해 어떤식으로 구현되는지를 직접 눈으로 보니 설계 단계에서 흐트러지면 다시 처음부터 시작해야된다는 말의 뜻을 이해하게 됐다.

그리고 내부적으로 식별 관계의 중요성과 프로시저, 커서, 트리거가 어떤식으로 제대로 구현이 되는지를 프로젝트 하는 도중 팀원들과의 회의 및 실습에서 많이 배웠다고 생각한다.

팀적으로는 두번의 팀 프로젝트 동안 모두 서로 배려해주는 분위기였고 모두가 열심히 하는 분위기가 조성돼서 나도 자극받아 열심히 할 수 있었고 서로가 서로를 보안해주고 팀 내부적으로 배울수 있는 팀을 만나게돼서 좋았던거 같다.

정미경

이번 세미 프로젝트를 진행하면서 느꼈던 점 중 하나는 사용자가 요구하는 사항들에 대한 DB설계는 또 다시 새롭게 고민해 될 부분 중 하나라는 사실입니다. 처음에 요구분석서를 통해 DB 설계를 고민했을 땐 사용자가 요구하는 부분들 그대로 테이블을 구성하면 끝나는 거 아닌가 라고 쉽게 생각했었습니다. 하지만 그렇게 설계하니 각 테이블마다 겹치는 컬럼이 생겨 더 세부적으로 테이블을 나눠야 했습니다.

추가로 사용자가 요구했던 건 성적처리 시스템이었고, 그 요구 분석서안에는 수강신청이라는 기능이 존재하지 않았지만 저희가 데이터베이스를 설계할때는 과정정보와 학생정보를 이어주는 테이블이 필요했습니다. 이 테이블이 저희에겐 수강신청 테이블이 되었고, 이는 사용자가 구체적으로 요구했던 내용중엔 없던 부분이었습니다.

이 외에도 과목과 과정을 이어줄때 데이터가 1:1로 매칭될 수 있게끔 그 사이에 매칭 테이블을 만들어 주었어야 했는데 이로 인해 결과적으로는 처음엔 생각지도 못했던 테이블들이 꽤 많이 생성 되었습니다. 그 외에도 PK 제약조건을 설정하는 것에 대해 고민했었는데 이는 각 테이블의 정보를 다른 테이블에서 가지고 가서 사용하려면 묶여있는 각 데이터를 특정할 수 있는 무언가가 필요하다는 것에서 해결할 수 있었습니다.

ER-다이아그램 설계가 끝나고 물리적 설계에 들어갔을 때 어려웠던 점은 부모 테이블과 자식테이블간의 관계에 대한 문제였습니다. 자식테이블은 부모테이블에서 정보를 받아다가 사용하고, 그 외에도 자식테이블이 다른 자식 테이블의 정보를 받아다가 사용하면 나중에 삽입, 수정, 삭제 기능을 구현할 때 연결되어있는 자식테이블의 부모테이블까지도 고려해야 되는 부분이 생겨 예외조건문을 생성하는데 꽤 많은 시간이 소요되었습니다.

프로시저 구성 시 코드를 생성하면서 많은 고민이 필요했던 부분은 NULL 값에 대한 조건문 구성이었습니다. 처음엔 사용자에게 입력받은 값이 실제로 데이터 베이스 안에 존재하고 있는지에 대해 값이 없는 경우 조건문 구성을 IS NULL 혹은 IS NOT NULL 의 형태로 구성하였는데 그렇게 하니 프로시저 구문에서는 데이터가 존재하지 않는 경우를 NULL 값이라고 인지하지 못하는 건지 원했던 방향으로 기능이 제대로 구현되지 않는 현상이 발생하였습니다. 이 부분은 추후 다른 조원들이 처리 했던 방법 중 하나인 NVL 함수를 사용하여 NULL 값을 숫자 0 혹은 문자 '0' 으로 바꾸어 조건문을 구성는 것과 COUNT 함수를 사용하는 방법으로 해결할 수 있었습니다.

처음에 생각했던 것보다 많은 고민이 필요했고, 시간도 꽤 많이 들어갔던 프로젝트였습니다. 이전에 수업 중 과제로 임직원, 직업, 부서 등등 다양한 테이블을 똑같이 만들어서 제약조건을 걸었던 과제를 했었기에 그나마 방향을 잡고 나아갈 수 있었던 것 같습니다. 이번 프로젝트를 하면서 생겼던 문제 중 각 테이블마다 데이터 삽입, 수정, 삭제 기능을 권한이 있는 사용자만 할 수 있도록 걸러내는 작업을 제대로 구현하지 못했는데 이 부분이 과연 이전에 조원들끼리 얘기한 것처럼 자바에서만 처리를 해도 문제가 없는 건지 확인해 봐야 될 것 같고 그 외에도 데이터 베이스에서 하는 작업과 자바에서 할 수 있는 작업에 대해 한 번 생각해봐야 될 것 같습니다. (주민번호 유효성 검사를 수업 중엔 자바에서 처리하였지만 검색을 통해 찾아보니 db에서하는 경우도 있었음)

최나운

데이터베이스 설계부터 프로시저 작성까지 스스로 하는 게 낫설기도 하고 어려웠지만 팀원들과 같이 수정을 거듭하며 ERD를 완성시킬 수 있었다. ERD 생성에 시간을 많이 쏟았는데 그럴만한 가치가 있었다고 생각한다. 큰 틀을 잡고 테이블 구조를 만들며 ERD를 생성함과 동시에 제약조건이나 시퀀스 생성해서 부여할 부분들도 자연스럽게 구상할 수 있었기 때문이다.

프로시저를 작성하면서 어려웠던 점은 우선 어떤 프로시저들이 필요한지를 파악하고 그 프로시저 안에서 구현되어야하는 기능들이 무엇인지를 확인하기까지 좀 오래 걸렸다. 또한, 트리거를 만들어야할 부분은 어디이며 이 트리거가 무슨 역할을 하게 만들 것인지에 대해서도 오랜 시간 고민했다. 내가 맡은 부분은 과정테이블, 강의실테이블, 개설과정테이블이었는데 이 중에서는 개설과정이 제일 어려웠다. 다른 테이블들과 엮여 있는 컬럼이 많아서 그랬던 거 같다.

예외처리 해줘야 하는 부분들에 대한 조건문을 만드는 게 쉽지 않았고 개설과정등록 같은 경우에는 그냥 SELECT 했더니 나온 결과들을 IF문에서 한 번에 비교할 수 있는 방법이 있는지를 몰라서 커서를 이용해서 한 줄씩 조건문에 집어넣고 판별해주는 형식으로 처리했는데 생각해보니 그룹함수 MAX를 이용했어도 됐을 거 같다는 생각이 든다

이번 데이터베이스 프로젝트를 통해 배운 것들에 대해 더 깊이 찾아보면서 개념을 복기하기 좋았다. 그리고 자바 프로젝트 때보다 팀 합이 더 잘 맞는 거 같아서 그 부분에서도 만족스러웠다. 내가 모르거나 헛갈리는 부분을 공유할 수 있는 팀이 있다는 게 정말 다행이고 혼자 찾기 힘든 예러들이나 경우의 수를 제시해준 팀원들에게도 정말 고마움을 느꼈다. 이번 프로젝트 덕분에 파이널 프로젝트를 할 때는 과제수행을 조금은 더 수월하고 매끄럽게 할 수 있을 것 같다.
