

후기 취합

🕒 생성일	@2021년 4월 19일 오전 9:42
👤 참여자	 희주 이 <small>언니</small>  혜진 심 <small>혜림</small>  혜림 전 <small>민지</small>  박 민지  Harim Lee

▼ 목차

[박민지](#)

[심혜진](#)

[이하림](#)

[이희주](#)

[전혜림](#)

박민지

이번 오라클 세미 프로젝트를 통해 사용자 요구분석의 중요성에 대해 많이 느꼈습니다. 개념적/논리적 erd를 구성하면서 충분히 반영을 했다고 생각하고 그걸로 물리적 erd를 만들었지만 그럼에도 불구하고 사용자의 요구사항대로 모델링을 하는 것이 불편했습니다. 자바도 그렇지만 오라클은 테이블을 만드는 순서가 중요하고 사용하는 계정이라던지 데이터를 넣었는지 아닌지 등 염두해야는 것이 많아서 팀원들간의 소통이 매우 중요하다고 생각했습니다. 마지막 제출을 앞둔 주말에 개인적인 사정으로 프로젝트에 많이 참여하지 못했는데, 팀원들이 저의 상황을 이해해주고 배려해줘서 너무 고마웠습니다. 이번 오라클 세미프로젝트와 지난 자바 세미프로젝트에서 배운 것들을 잘 정리하고 복습해놓아서 파이널 프로젝트에서 잘 활용할 수 있도록 해야겠다고 느꼈습니다.

심혜진

팀 프로젝트로는 세번째 오라클로는 두번째 프로젝트를 하게 되습니다.

초반엔 개발도 시작한지 얼마 안됐는데 라는 마음과 함께

생각보다 자주 이뤄지는 팀 프로젝트가 부담으로 느껴지기도 했지만,

시간이 지날수록 이러한 팀 프로젝트에 대한 요령이 생기는 것을 느낍니다.

세번째 프로젝트인 오라클 팀 프로젝트는 2조 팀원들이 적극적으로 의견을 내고 참여해주어서 팀 프로젝트를 진행하는데 수월했습니다.

좋았던 점은 서로 안되는 부분, 헷갈리는 부분들을 공유하며 배우게 되는 것들이 많았던 것입니다.

SQL에 대해 사용도 많이 해보며 실제 DB가 어떻게 구축될지 상상하며 쿼리를 짜는 것들이

흥미로웠습니다.

이번 프로젝트 또한 어쩌면 이 과정의 한가지 목표일 수 있는 파이널 프로젝트에 자양분이 될 것이라고 생각합니다.

이런 팀 프로젝트를 지속적으로 하게 해주시는 김호진 선생님께 감사의 마음을 전하고 싶습니다.

이하림

오라클 세미 프로젝트를 시작하고 가장 막막했던 점은 요구분석이었다.

요구분석서라는 것을 처음 보기도 했거니와 이 문서를 바탕으로 사용자의 요구사항을 도출해내서 다이어그램을 만드는 것도 막막했다.

어떤 테이블이 필요한지, 어떤 값을 가져야 하는지 파악하는 데도 오래 걸렸지만 그 테이블들의 관계성을 파악하는데에 가장 많은 시간을 소요한 것 같다.

두 번째로 힘들었던 점은 '가정' 이었다.

이번 프로젝트는 다른 프로그램과의 연동 없이 sql developer만을 이용해야 했다.

그 과정에서 수많은 가정이 필요했다. 로그인을 한다고 가정했고, 교수/관리자/학생을 선택한다고 가정했고, 화면이 전환되었다고 가정했다.

이 모든걸 가정만 한 상태로 코드로 구현해내야 한다는게 추상적으로만 느껴졌다.

그래서 코드를 짜면서도 이게 맞는걸까 라고 계속 생각했고, 결과를 보면서도 제대로 나온게 맞나 여러 번 확인해야 했다.

또, 요구분석서에서 요구한 것이 제대로 구현된 게 맞는지도 제대로 파악해내기가 힘들었다.

그래도 핵심 기능들을 중심으로 하나 둘 씩 코드를 짜다보니 가정을 하고 구현한다는게 어떤 건지 깨달을 수 있었고, 요구분석서를 바탕으로 요구사항들을 구현해낸게 뿌듯하게 느껴졌다.

비록 비대면이라 팀원들과 zoom으로만 소통해야 했지만, 다들 열의를 가지고 참여해줘서 어려움이 없었고 팀장님이 프로젝트를 잘 이끌어 줘서 제 시간안에 잘 끝낼 수 있었다.

다음 프로젝트가 어떤 것이 될지는 모르겠지만, 지금까지 배운 내용들을 합치면 더 재밌는 프로젝트를 할 수 있을 것 같다.

이희주

처음 요구분석서를 보고 어떠한 테이블을 구성하고 어떠한 연결고리를 가지고서 테이블을 연결해야할지 구현하는 단계가 제일 어렵고 막막했던 것 같다.

특히 물리적설계 단계에서 어떠한 테이블이 참조를 받고, 어떠한 테이블이 참조를 하는지 관계를 따지는게 어려웠다. 이러한 문제점들을 팀원들과 공유하고, 여러 의견들을 통해 문제점들을 쉽게 해결할 수 있었다.

그리고 단순히 DB로 구현을 하는 단계이기 때문에 눈에 보이지 않는 로그인 페이지를 구현하는데에서도 어려움을 겪었고, 특히 실제로 데이터를 삽입, 수정, 삭제하는 프로시저 구현 과정에서 예외처리하는 부분이 제일 까다롭고 시간이 오래 걸렸다.

처음 ORACLE 프로젝트를 통해 DB 설계부터 구현까지 해보면서 많은 어려움이 있었지만 성실하고 열정적인 팀원들 덕분에 성공적으로 프로젝트를 끝마칠 수 있었던 것 같다.

전혜림

처음 프로젝트를 시작하면서, 요구분석서를 읽고 정규화된 테이블을 논리적으로 설계하는 것부터 난감했다. 수업시간에 배운대로 정규화 규칙을 거쳐 한 테이블을 계속해서 분리했지만, 완전히 이해되지 않는 부분도 있었고 특히 다대다 관계를 일대다 관계로 쪼개는 경우 어떤 의미의 테이블을 생성해서 연결해야 할지 고민이 많았다. 예를 들어, 학생이 개설 과정을 수강 신청하는 경우 이를 학생 테이블과 개설 과정 테이블로만 분리하면 정규화가 완료되지 않은 상태가 된다. 따라서 수강신청 테이블을 생성하여 두 테이블을 수강신청코드와 같은 새로운 연결고리로 관계지어야 했다.

또, 교수가 성적을 입력할 때나 학생 관련 출력물을 확인할 때 보여주어야 하는 중도탈락 내용은 어떻게 연결지어야 할지 특히 고민이 많았다. 처음엔 학생이나 개설 과목 테이블 혹은 성적 테이블에 연결지어야 하는지 고민했는데, 결국 선생님의 피드백을 받고 수강신청 테이블과 연결지어 후에 쿼리문을 작성할 때 잘 활용할 수 있었다. 논리 설계를 할 때는 왜 이렇게 연결지어야 하는지 감이 잘 오지 않았지만, 쿼리문을 작성하고 지금 생각해보니 왜 수강 신청 테이블과 연결시켜야 했는지 이해가 간다.

중도탈락 내용은 중도탈락 시기에 따라 학생이 수강하는 과목별로 영향을 미치면서, 교수가 자신의 과목을 '수강'하는 학생들의 목록을 볼 때, 또 성적을 확인할 때, 학생 목록을 확인할 때 나와야 하므로

이 모든 테이블을 연결하는 수강신청 테이블과 연결이 필요했다. 물리 설계 ER 다이어그램을 완성하기까지 선생님의 피드백이 3번 있었는데, 만약 이를 거치지 않고 우리끼리 논의한 뒤 쿼리문을 짜기 시작했다면 작성 도중 테이블을 수정하는 일을 계속해서 반복했을 것이다.

쿼리문을 작성하면서는 각 테이블의 PRIMARY KEY로 활용되는 코드를 어떻게 구성할지 더 논의했어야 한다는 생각이 들었다. 교수, 학생, 책, 강의실은 VARCHAR2 타입으로 'S0001'처럼 다른 테이블과 구별되는 코드를 만들었지만 수강신청코드, 성적코드, 개설과목 코드, 개설과정코드와 같은 코드들은

1001처럼 다른 테이블과 구별되지 않는 코드를 사용하여 데이터를 입력할 때 혼란이 많았다. 또, 해당 코드들은 시퀀스를 사용하여 자동으로 생성하도록 만들었는데 논의할 때는 자동으로 번호를 생성해주기 때문에 편할 줄 알았던 시퀀스가 실제로 사용해보니 매우 불편했

다. 데이터를 입력하고 수정하고 삭제하는 과정에서 한 번 생성된 번호는 다시 생성할 수 없기 때문에 의미 없이 사라지는 번호들이 많았기 때문이다.

또한, 프로시저를 생성할 때도 예외처리와 관련하여 어려움을 겪었는데, 여러 테이블을 조인하여 출력하는 부분도 생각보다 어려웠다. 10개의 테이블을 조인하더라도, 테이블의 모든 데이터들에 NULL값이 없다면 조인이 어렵지 않지만 여러 테이블에서 각각 NULL값이 있는 데이터도 보여주고자 하는 경우 LEFT, RIGHT 조인을 사용해야하기 때문에 까다로웠다. 예를 들어, 학생 목록과 함께 수강 과정, 과목, 성적을 출력하면서 동시에 등록 과목과 성적이 없는 학생목록도 나오게하고 싶었으나 계속해서 성적이 등록되어 있는 학생만 나오거나, 성적이 없는 학생도 나오는 경우에는 성적이 있는 학생들이 CARTESIAN PRODUCT로 모든 경우의 수가 중복해서 나왔다. 결국 한 줄씩 코드를 수정해가며 계속 테스트해서 학생 테이블과 성적 테이블의 모든 값이 나오도록 조인하여 해결할 수 있었다.

이번 프로젝트는 데이터베이스를 설계하고, 쿼리문을 작성한 뒤 다른 프로그램과의 연결 없이 오라클 내에서만 테스트했지만 실제로 해당 테이블이나 프로시저, 뷰가 다른 프로그램과 연결되었을 때 어떻게 활용될지 계속해서 생각하려 노력했다. 예를 들어, 처음에는 학생 데이터를 삭제하는 문제를 단순하게 생각하여 학생과 관련된 수강 정보가 있다면, 참조키로 인해 학생 테이블의 해당 학생 정보는 자동으로 삭제할 수 없게 만들었다. 그런데 현실적으로 생각했을 때, 사이트를 탈퇴하거나 학생의 요청으로 인해 데이터 베이스 안의 학생 개인 정보를 삭제해야 하는 경우, 관리자도 학생 데이터를 바로 삭제할 수 없는 것은 이상하다는 의견이 있었다. 따라서 학생 데이터를 삭제하는 경우, 수강신청이나 점수와 같은 관련 정보는 유지하면서 학생 정보만 NULL이 되도록 변경할 수 있었다. 이처럼 실제 데이터베이스를 설계하고 필요한 쿼리문을 작성할 때 현실에서 발생할 수 있는 여러 상황을 고려하는 것이 필요함을 느꼈다.

마지막으로, 요구분석서를 바탕으로 디테일한 내용을 정하기 위해 다양한 의견을 공유하고 쿼리문을 작성할 때도 여러가지 상황을 고려한 예외처리를 꼼꼼히 하면서 끝까지 수고한 팀원들에게 감사하다. 서로 작성한 코드에 문제가 생겼을 때, 먼저 나서서 확인하고 함께 오류를 해결해나가는 과정이 있었기에 프로젝트를 마무리 할 수 있었다.