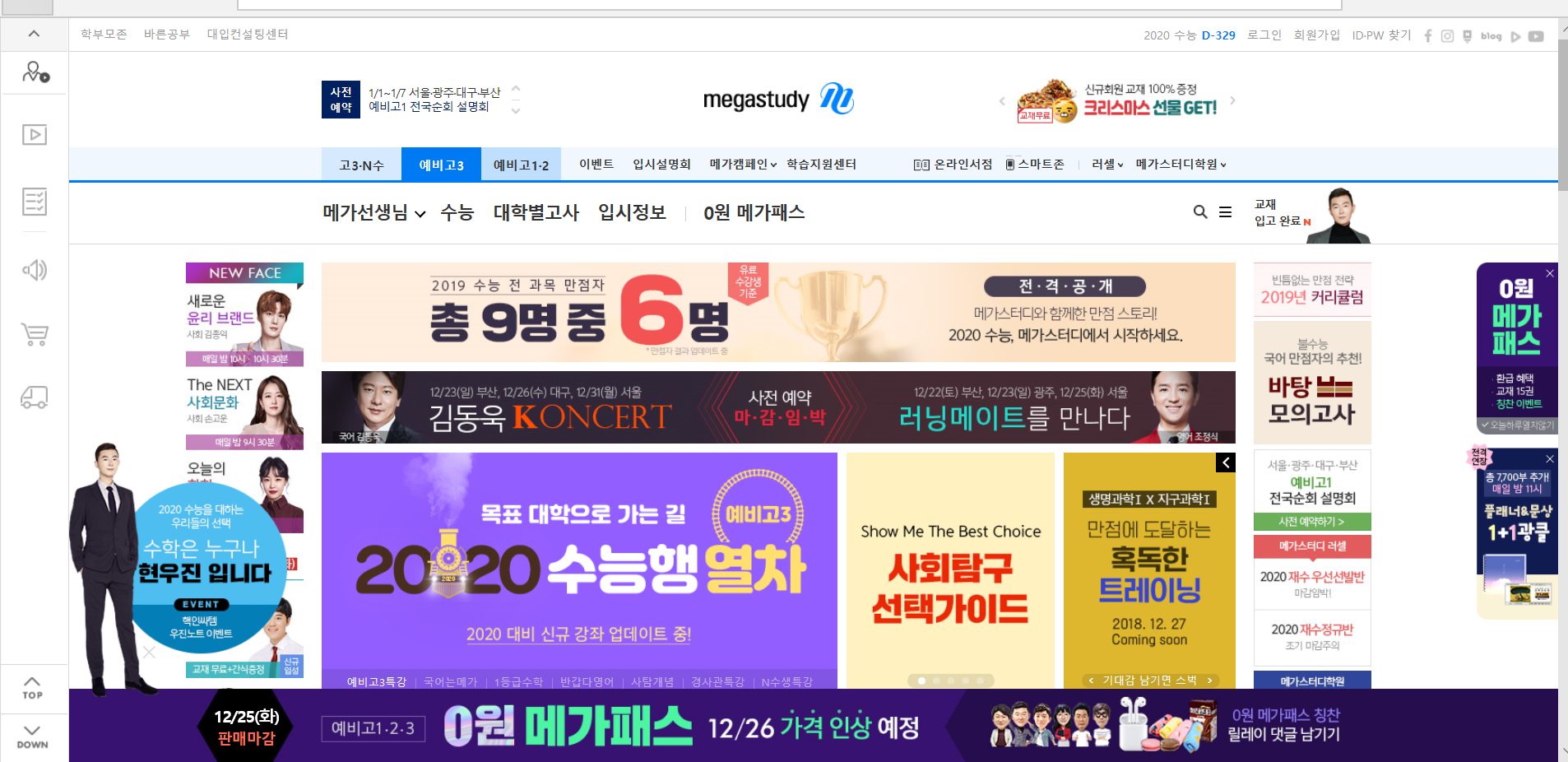
**인하스터디 강의사이트 구현을 위한 DB**

12161697 김민지,12161735배윤진,12161785전수연 12조

***1. 응용에 대한 간단한 소개 (주제 선정 이유)***

**M사**

**인하스터디**

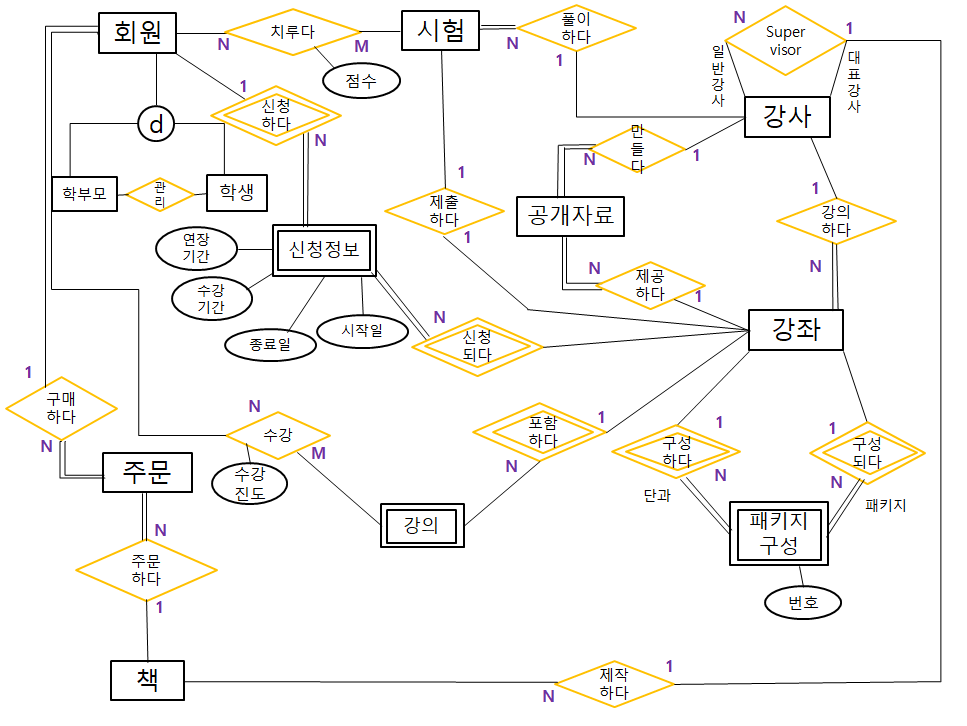


🡪 유명한 ‘ㅁ’사 인터넷 강의 사이트에는 처음 이용하는 고객이나, 학부모님들은 어떤 기능이 있는지 한눈에 파악이 불가능합니다. 많은 배너들로인해 꼭 필요한 주요기능을 이용하는 것 있어 불편함을 겪어, 보다 간단하고, 한눈에 메뉴들과 기능을 확인할수 있는 ‘인하스터디’를 구현했습니다.  
🡪 많은 학생들이 한번쯤 이용해 보았던 인터넷 강의사이트를 이번 데이터베이스 설계를 수강하면서 방대한 데이터를 효율적으로 관리 할수있도록 도와주는 최고의 DB administer가 되기위해 한걸음 다가가기 위한 적합한 주제라고 생각하여 구현하였습니다.

***2. 현실감 있는 요구 분석서***

* 김인하는 인터넷강의 사이트 인하스터디를 운영하기 위해 데이터베이스를 설계하려고 한다. 회원가입을 한 회원들만 인터넷강의를 수강할 수 있다. 인하스터디는 회원의 id, 이름, 생년월일, 이메일, 전화번호, 비밀번호를 관리한다. 또한 회원은 여러 개의 배송관리주소를 가진다.
* 회원은 학부모회원과 학생회원으로 나뉘며, 회원가입시 학부모회원은 직업, 학생회원은 학교와 학년에 대한 정보를 입력해야 하고 졸업여부를 통해 N수생인지 아닌지를 구분한다.
* 강사는 고유의 강사번호, 이름, 성별, 이메일, 과목, 연봉 정보를 가지며 한 과목만을 담당할 수 있다. 또한 강사의 학력 및 경력 정보를 프로필을 통해 회원들은 볼 수 있다. 또한 한 과목당 한 명의 관리 강사가 있다. 또한 강사는 공개자료를 제공하며 고유의 자료번호를 가지며 제목, 등록일을 알 수 있다.
* 강사는 여러 개의 강좌를 강의하며 강좌에는 단과 강좌와 패키지 강좌가 있는데, 패키지 강좌는 여러 개의 단과 강좌로 구성된다. 패키지이던 단과이던 모든 강좌는 고유한 강좌번호, 학년, 권장진도, 가격, 유형, 강좌명을 가진다.
* 강좌에는 여러 강의가 있으며 최소 chapter5 에서 chapter7까지 이루어져 있다. 강의에는 고유의 강의번호, 강의명이 있다. 이 강의를 구성하는 강좌를 신청한 회원들만 강의를 수강할 수 있고 수강진도가 기록에 남는다.
* 회원은 여러 개의 강좌를 신청할 수 있고 시작일, 종료일, 연장기간, 수강기간을 기록한다. 강좌를 연장하면 그만큼 종료일도 늘어나게 되나, 수강기간은 늘어나지 않는다.
* 회원은 강좌에 대한 시험을 치루고 점수가 기록에 남는다. 시험은 시험번호, 시험명, 해설범위, 듣기파일, 학년으로 구성되어 있으며 과목별로 시험을 검색하여 원하는 시험을 칠 수 있다. 한 강좌당 시험은 한 개 뿐이다. 시험은 강좌를 신청하지 않은 회원도 치룰 수 있다.
* 회원은 참고할 책들을 주문할 수 있으며 주문 상태는 주문정보에서 알 수 있다. 주문정보에는 주문번호, 번호, 주문일, 결제금액, 진행상태, 수량, 주소, 주문을 한 회원의 id, 주문한 책의 번호가 있다.
* 해당 책은 고유한 책번호를 가지며 발행일, 목차, 책이름, 판매가, 발행처, 사이즈(가로cm,세로cm)를 알 수 있다. 일부 책들 중에는 인하스터디의 강사가 만든 책들도 있다.

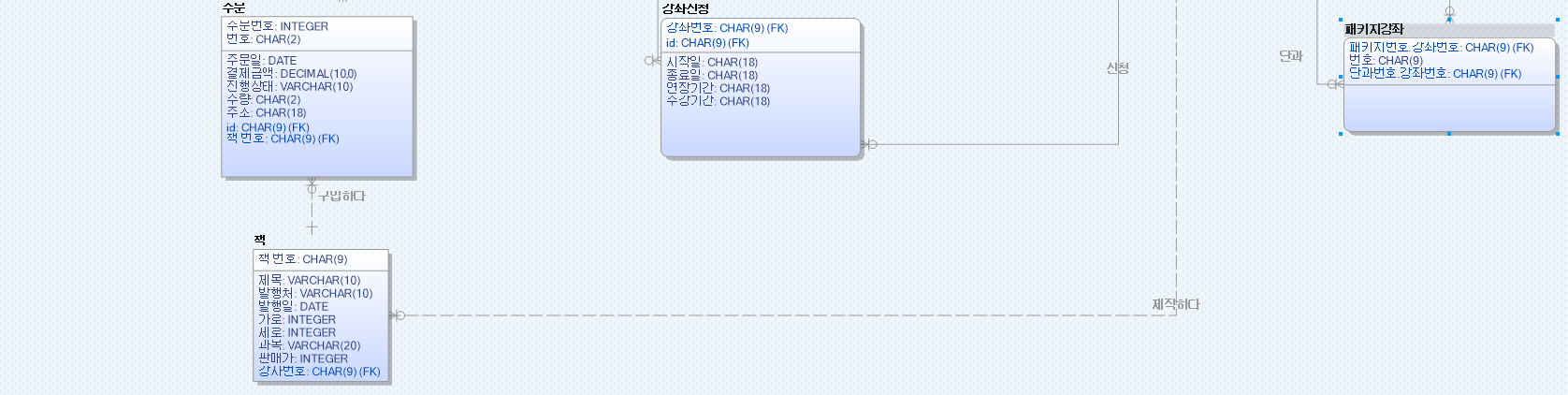
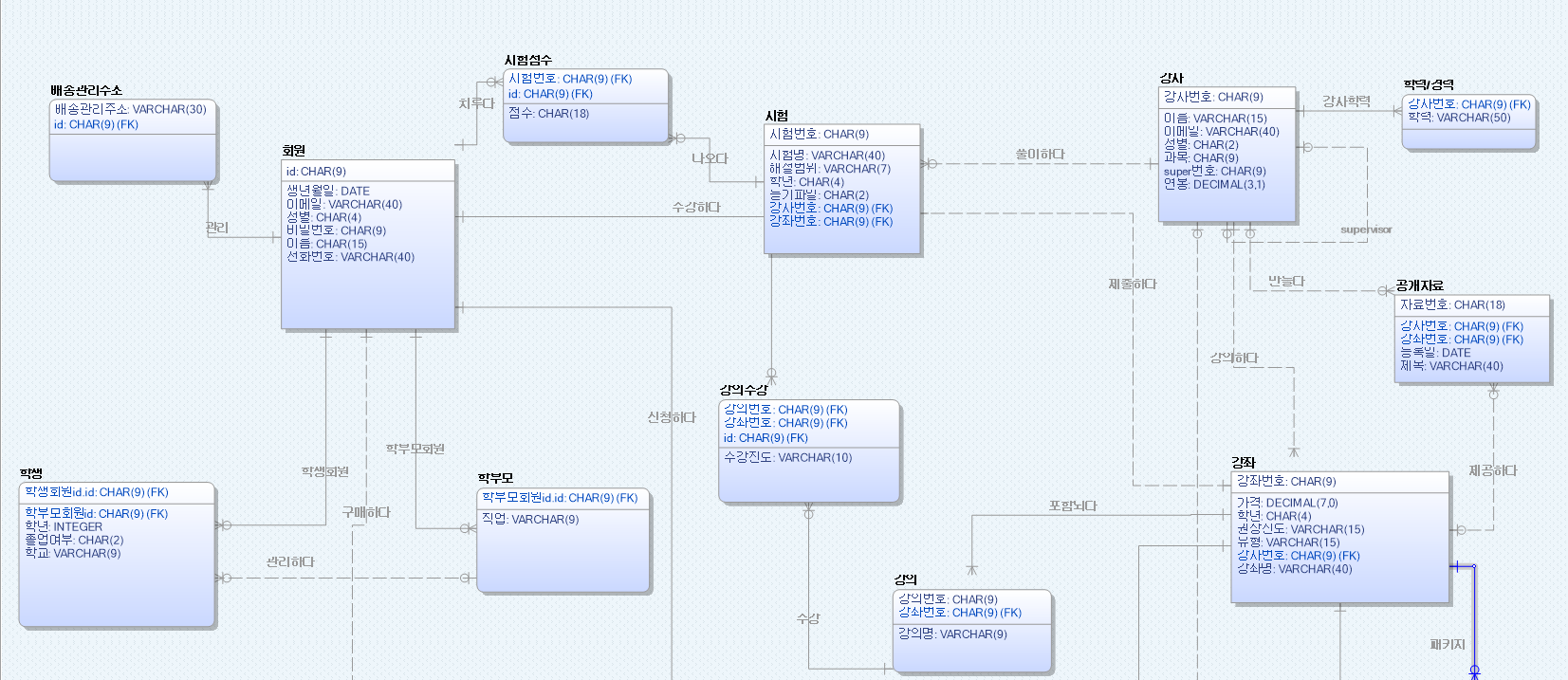
***3. 개념적 (E)ER Diagram***



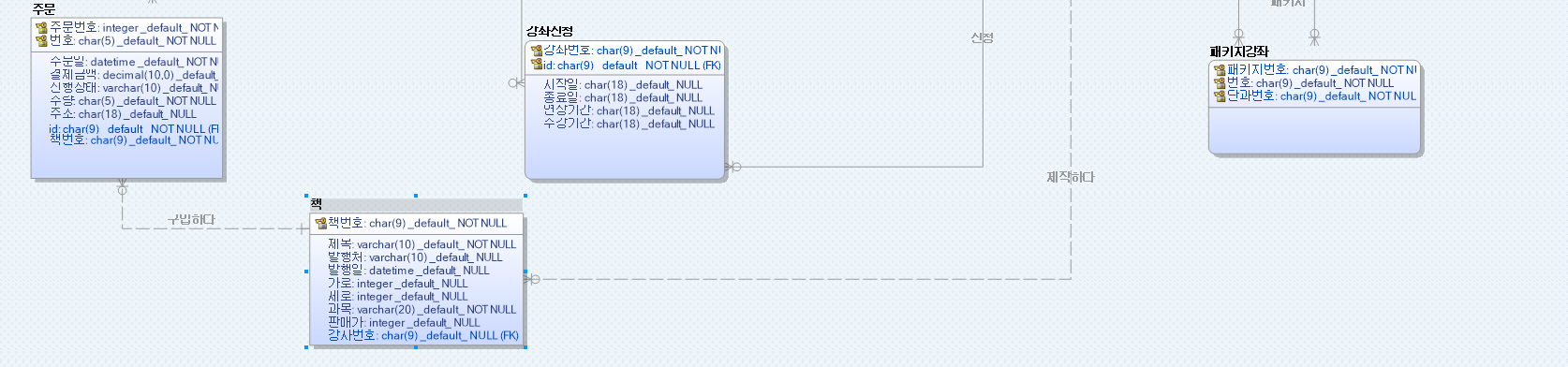
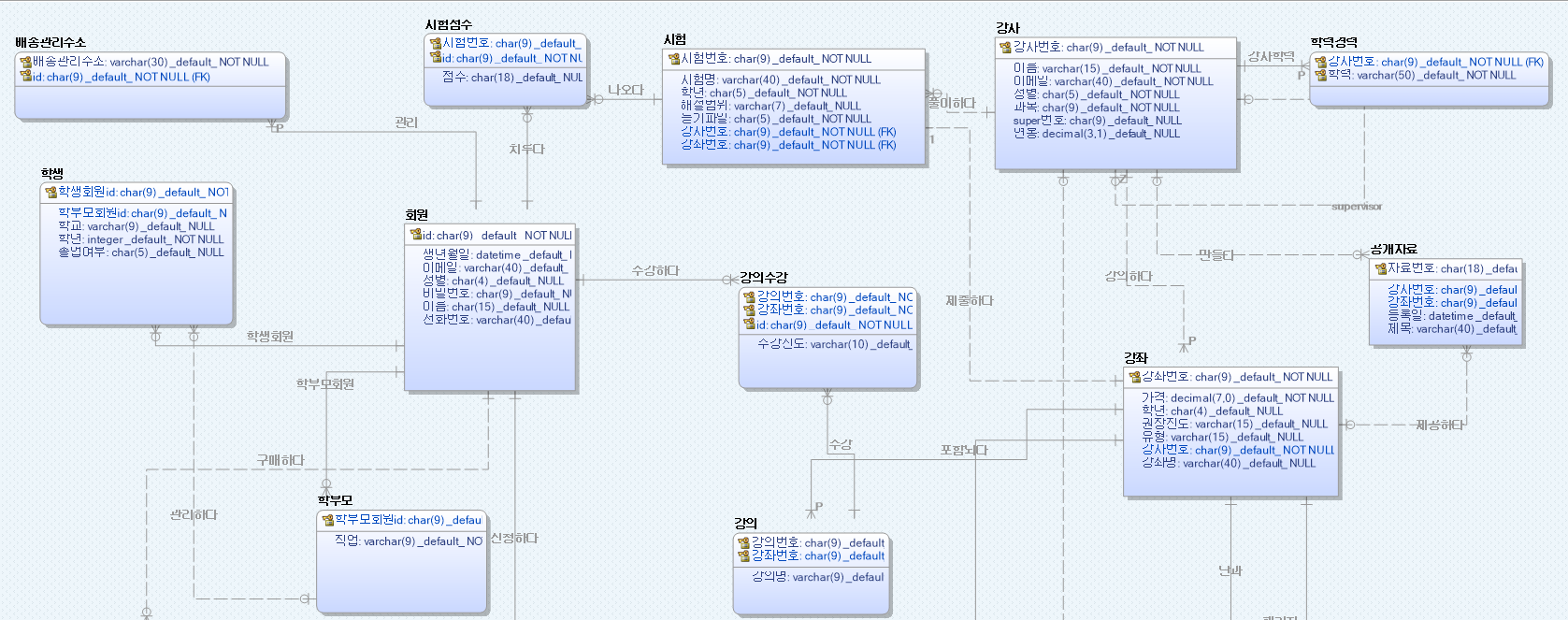
🡪위의 개념 er diagram 에서 해당 강좌에 강의가 있는데 살펴보면 신청하지 않은 강좌의 강의도 수강할수 있다는 관계로 해석할수 있습니다. 이 부분에서는 SQL에서 데이터를 넣어줄 때 강좌를 신청하고 강좌안에 있는 해당 강의만을 수강할수 있도록 넣어주었습니다. asp에서도 마찬가지로 구현해 주었습니다.

🡪주문에서도 주문번호와, 번호(구매한 물품의번호) 만으로도 주문한 상품에대한 정보를 알 수 있음으로, 책과 회원사이의 (책 primary key, 회원, primary key)를 받아오지 않고 인조키(주문번호,글번호) 사용을 통해서 strong entity로 만들어 주었습니다.

🡪패키지 구성이라는 weak entity를 보면 강좌의 self relationship 을 사용해주었습니다. 그 이유는 강좌의 여러 개의 단과 강좌는 결국 패키지 강좌가 되기때문이고, 패키지구성을 보시면 강좌에서 받아온 FK로 단과강좌번호, 패키지강좌번호, 번호를 복합키로 사용했습니다. 그 이유는 패키지 강좌안에서 몇번째 단과강좌인지 패키지강좌내의 구성을 알기위하여 번호도 한번 복합키로 사용했습니다.

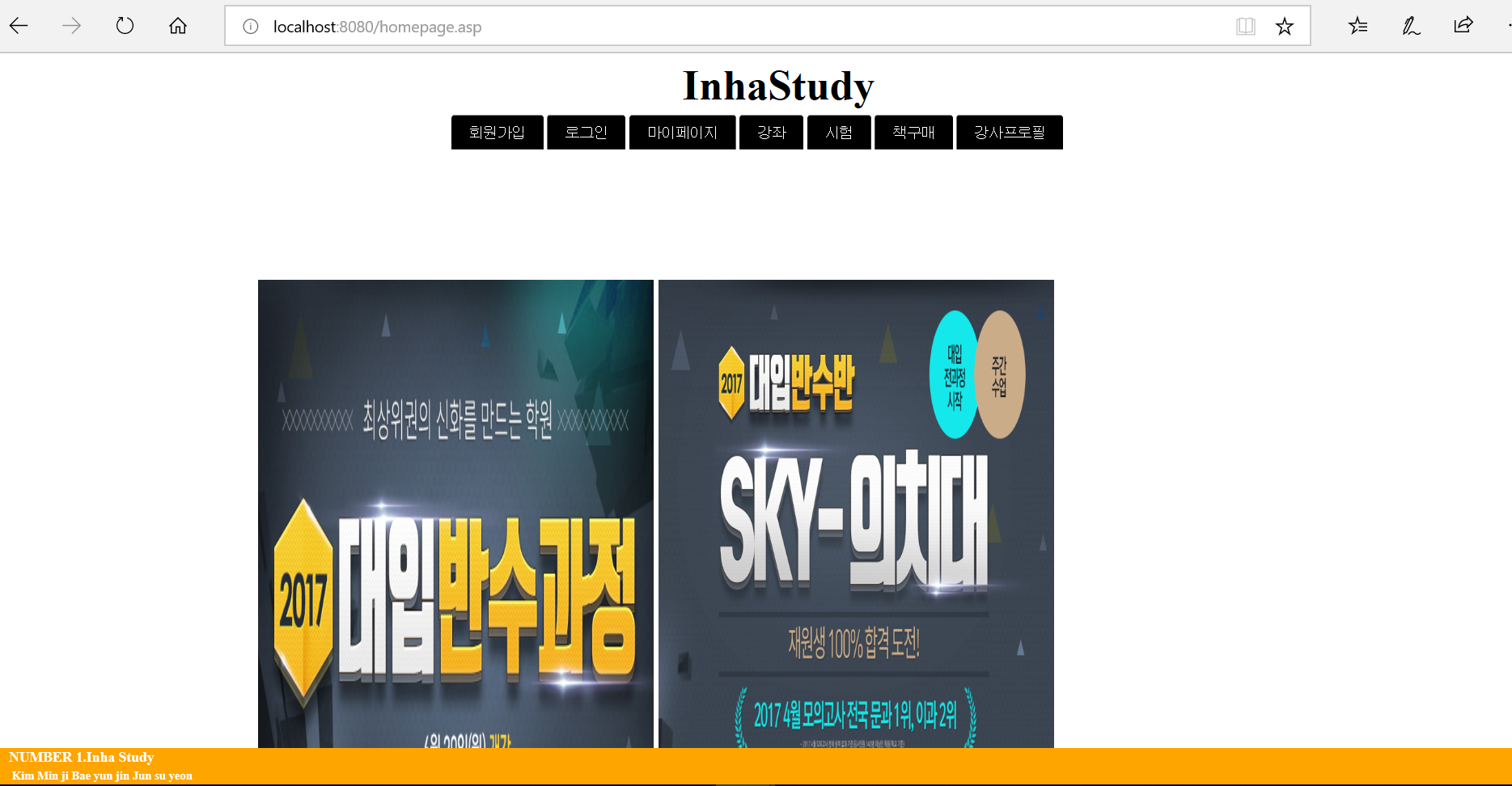
-> 강사table 안의 super번호는 과목별로 대표강사를 표현해주기 위해 임의로 만든 부분이며 과목당 한명의 대표강사를 두도록 했습니다.

***4. ER-Win (Physical) Diagram***



🡪Physical design으로 바꾼 모습입니다. domain, null 속성을 정해주었습니다.

***5. ASP 프로그램***



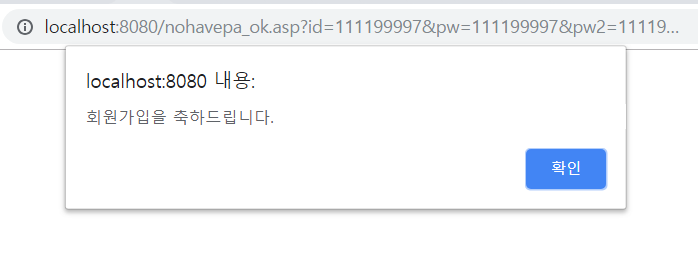
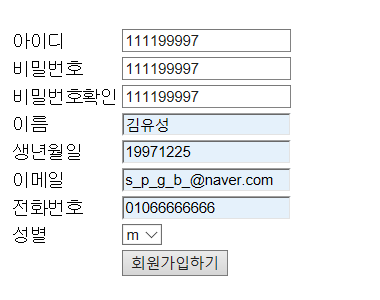
🡪첫 홈페이지 화면입니다.



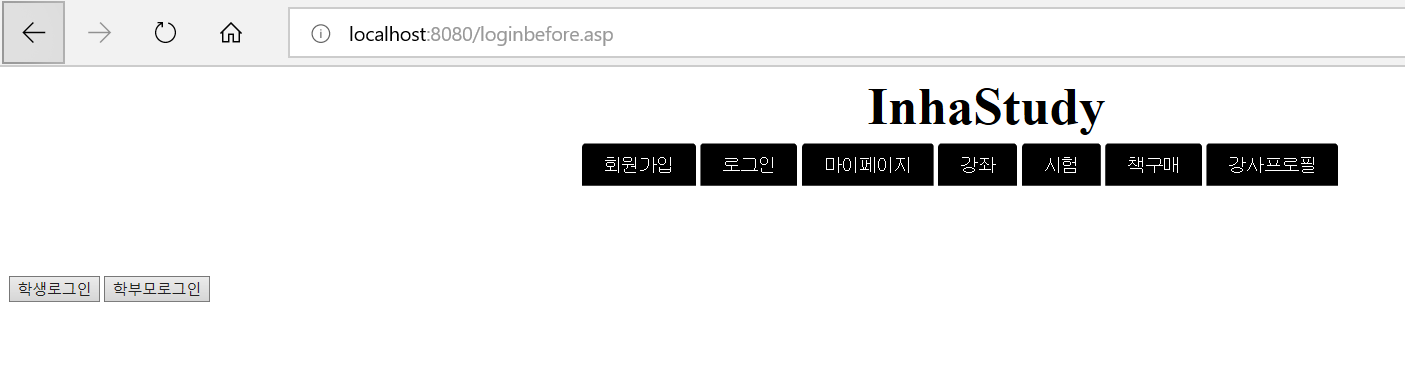
🡪 아직 로그인을 하지 않았을 의 header입니다 /header.asp



🡪기본적으로 회원가입을 해 학생회원,학부모회원이 가입 할 수 있으며, 예외처리를 통해서 이미 가입한 회원의 아이디를 입력하면 팝업창이 뜨고, 다시 아이디를 입력하도록 해주었습니다.

🡪 정보를 력하고 ‘회원가입하기’버튼을 누르면 해당정보를 입력하고, 예외처리를 통과하면 학생회원,학부모회원 버튼이 뜨며 해당버튼을 누르고 추가사항을 입력해주면 됩니다. 로그인 후 header는 위와 같은 모습입니다 / header2.asp





🡪로그인 뉴를 누르면 학생로그인,학부모로그인이 나오며,

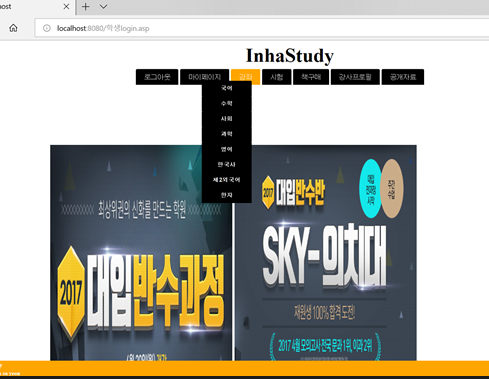
🡪버튼(학생로그인or학부모로그인)을 누르면 로그인창이 나타나며, 아이디와 pw를 칩니다. 예외처리를 이용하여 없는 아이디를 입력시 로그인이 되지않습니다.

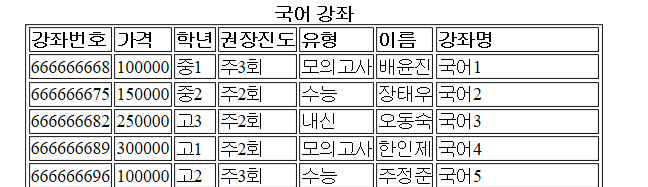


🡪로그인 직후 화면입니다.(학생로그인)



🡪강좌를 누르면 과목별 이름이 뜨고 해당 이름을 누르면 과목에 대한 강좌에 대한 정보를 알 있습니다. 마찬가지로 시험을 누르면 강좌에서 만들어진 시험에 대한 정보를 한눈에 확인할 있습니다.

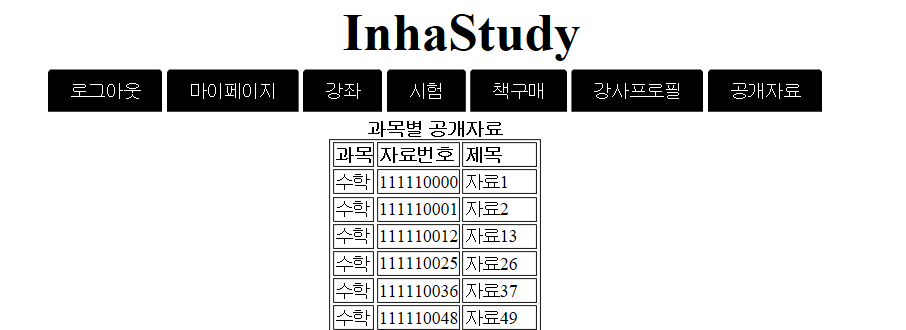




🡪로그인 후 메뉴의 강좌를 누르면 각 과목에 대한 정보가 나오며, 강좌에 대한 정보를 과목별로 한눈에 알수 있습니다. 시험도 마찬가지로 과목별로 시험에 대한 정보를 확인 할 수 있습니다.

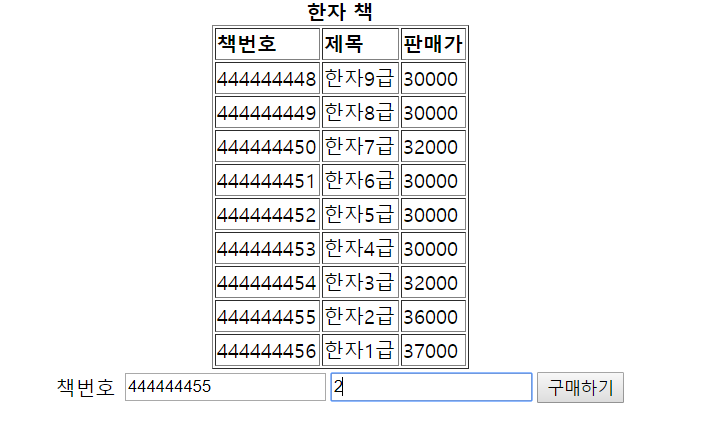


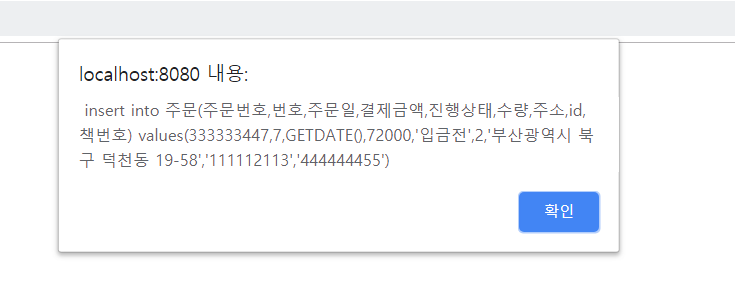
🡪로그인 후, 공개자료 tab에 들어가면 과목별로 공개자료에 대한 정보를 알 수 있습니다.

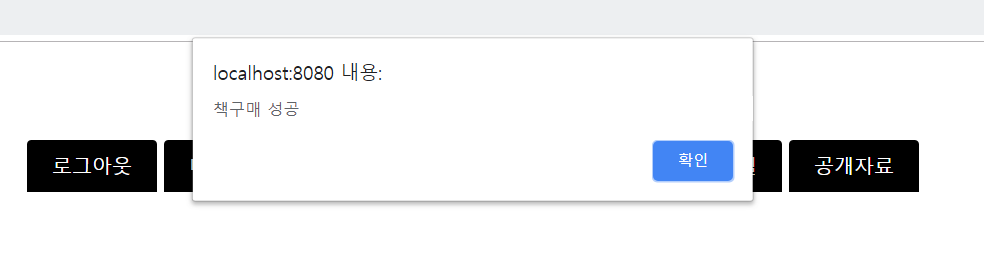
🡪

🡪과목에 ‘수학’을 입력했을 때 의 결과입니다. 수학과목에 대한 공개자료의 자료번호와 제목을 확인할 수 있습니다.







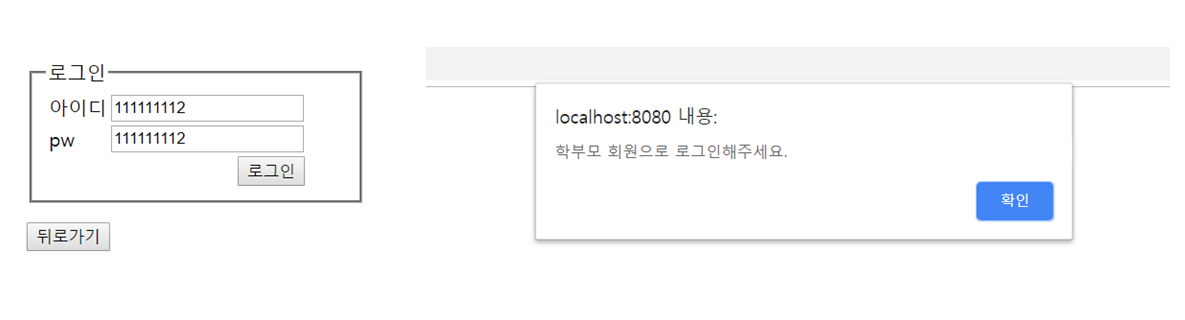


🡪로그인 전, 책구매를 누르면 구매 할 수 있는 책 목록을 볼 수 있고 구매를 할 수 있습니다. getdate() 함수를 통해 주문일자도 실시간으로 넣어주었습니다.

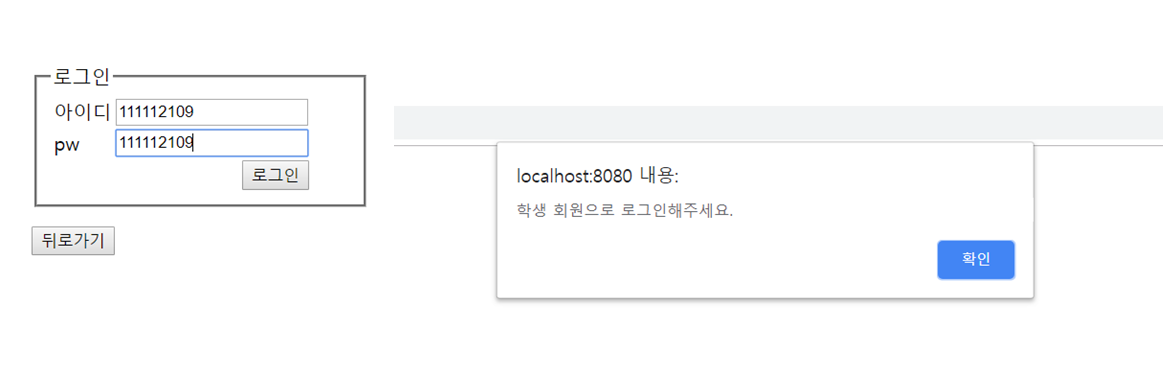


🡪로그인전 책구매를 메뉴를 누르면 책 구매할 수 있는 목록이 뜹니다.

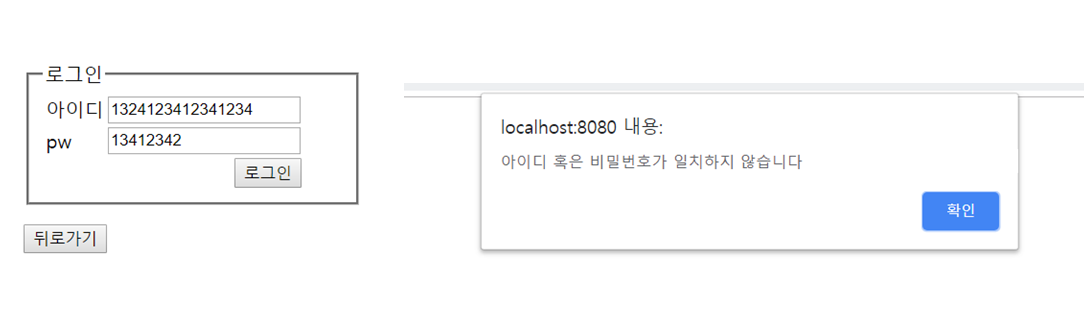
**[예외처리]**



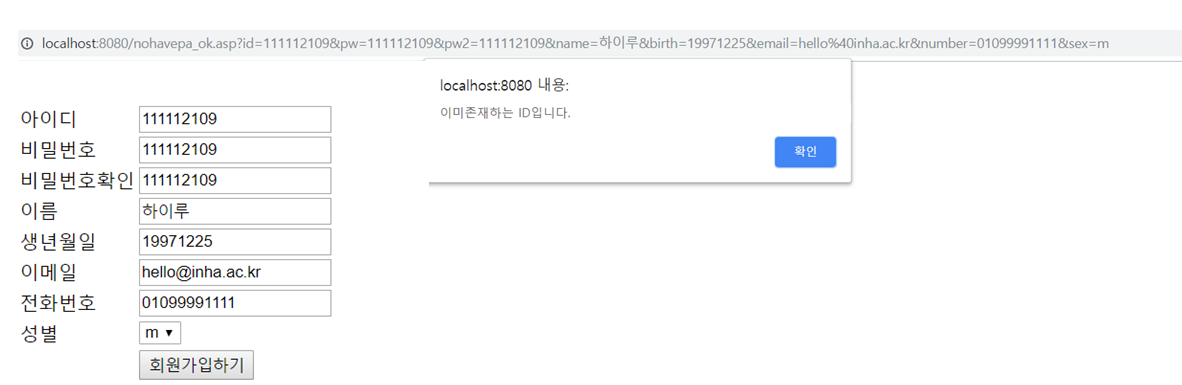
**-> 아이디가 111111112 학부모인데 학부모가 학생로그인을 이용할시 팝업창을 뜨게했습니다.**



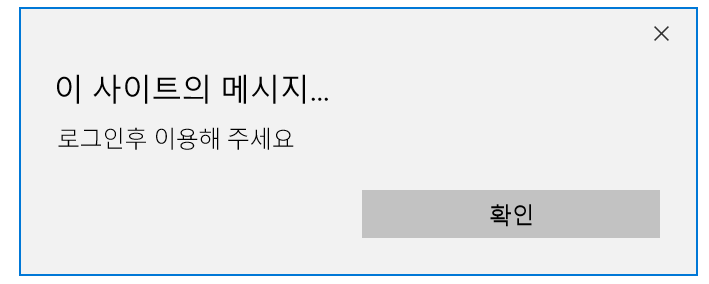
**-> 아이디가 111112109 학생인데 학생이 학부모 로그인을 할시 팝업창을 뜨게 했습니다.**



**-> 회원가입 하지 않은 아이디로 로그인시 팝업창을 뜨게 했습니다.**



**->이미 존재하는 id로 회원가입 할시엔 해당 아이디로 회원가입할수 없다는 경고메세지를 출력하고 아이디를 다시 입력하도록 했습니다.**



🡪 로그인하지 않고 마이페이지를 누르면 로그인후 이용해달라는 팝업창을 띄우도록 했습니다.

***6.주어진 응용의 대포 SQL 질의 및 실행 결과***

1. 인하스터디는 1등급 대박기원 이벤트를 지원한다. 생일이 11월 1일~11일 사이의 중학생~고등학생에게 패키지강좌의 30% 할인된 가격으로 이벤트를 진행하려고한다. 각 패키지명과 할인된 가격, 본래의 가격을 표시한다.

select distinct 강좌.강좌명, 강좌.가격\*0.7 as 할인가격 , 강좌.가격 as 원래가격

from 회원, 강좌, 강좌신청

where (YEAR(회원.생년월일) between 2002 and 2004 and month(회원.생년월일) like '11' and day(회원.생년월일) between 1 and 11)

AND 회원.id=강좌신청.id AND 강좌신청.강좌번호 = 강좌.강좌번호

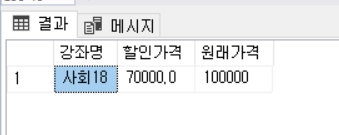
UNION

select distinct 강좌.강좌명, 강좌.가격\*0.7 as 할인가격, 강좌.가격 as 원래가격

from 회원,강좌,강좌신청

where (YEAR(회원.생년월일) between 1999 and 2001 and month(회원.생년월일) like '11' and day(회원.생년월일) between 1 and 11)

AND 회원.id=강좌신청.id AND 강좌신청.강좌번호 = 강좌.강좌번호



**2. 모든 사용자들을 연령대별, 수강하는 패키지강좌를 종류별로 셈하여 각 강좌들에대한 모든 인원들의합을, 각 연령대별에 대한 총 합도 같이 세는 질의**

select

case GROUPING(TEMP.학부모) when 1 then '전체 사용자' else TEMP.학부모 end as '연령대',

count(TEMP.학부모) "구입한 인원수",

case GROUPING(TEMP.제2외국어) when 1 then '전체 단과 강좌 수' else TEMP.제2외국어 end as '단과명'

from ( select

case when 2018-YEAR(회.생년월일) between 14 and 16 then '중학생'

when 2018-YEAR(회.생년월일) between 17 and 19 then '고등학생'

when 2018-YEAR(회.생년월일) between 40 and 50 then '학부모' end as '학부모',

case when 강좌명 like '수학%' then '수학' when 강좌명 like '국어%' then '국어'

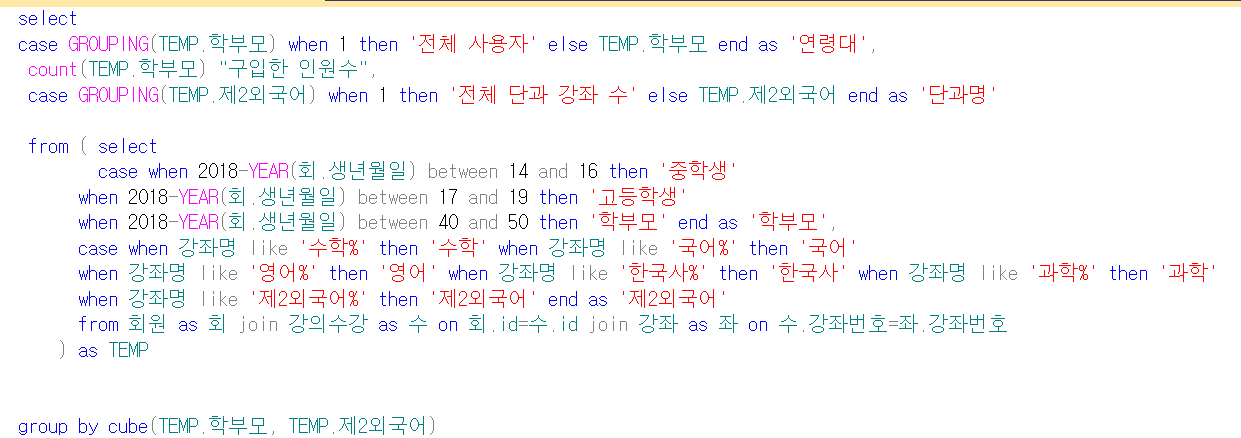
when 강좌명 like '영어%' then '영어' when 강좌명 like '한국사%' then '한국사' when 강좌명 like '과학%' then '과학'

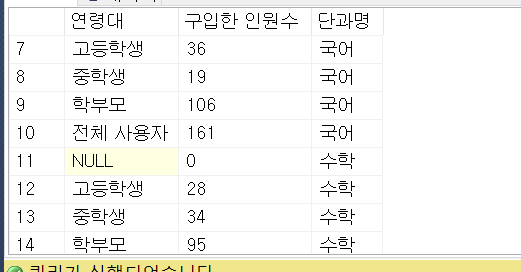
when 강좌명 like '제2외국어%' then '제2외국어' end as '제2외국어'

from 회원 as 회 join 강의수강 as 수 on 회.id=수.id join 강좌 as 좌 on 수.강좌번호=좌.강좌번호

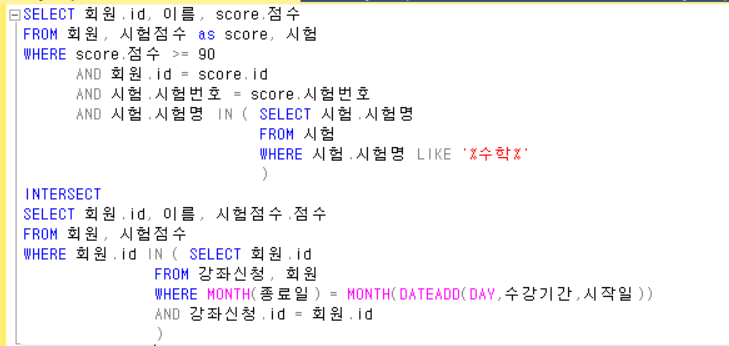
) as TEMP

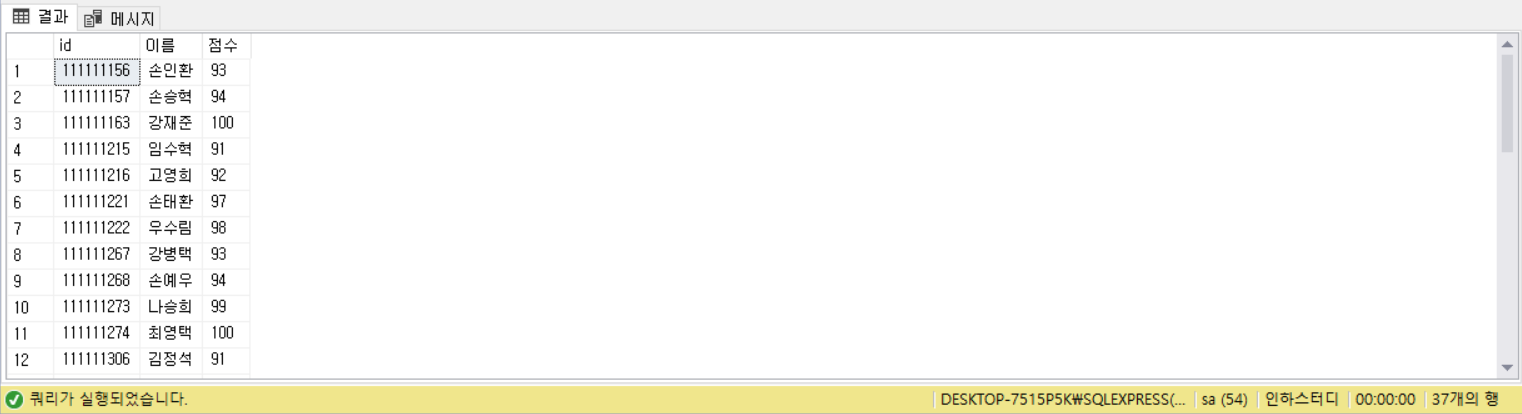
group by cube(TEMP.학부모, TEMP.제2외국어)





**3. 강좌를 신청한 학생 중, 신청했던 강좌를 연장한 학생을 구한다. 그 학생들 중 수학과목 시험을 쳤는데 90점 이상이 나온 학생의 id, 이름, 점수를 구한다.**





**4. 성별, 과목별로 강사를 구하여 과목별 인원수와 평균 월급을 구한다.**

select

case GROUPING(table1.성별) when 1 then '전체성별' else table1.성별 end as '성별',

case GROUPING(table1.과목) when 1 then '전체과목' else table1.과목 end as '과목',

count(\*) as "인원 수",

avg(table1.월급) as "평균 월급"

from ( select 강사.과목 as 'subject', 강사.이름 as 'name',

case when 강사.과목 like '국어' then '국어강사' when 강사.과목 like '수학' then '수학강사'

when 강사.과목 like '사회' then '사회강사' when 강사.과목 like '과학' then '과학강사'

when 강사.과목 like '영어' then '영어강사' when 강사.과목 like '제2외국어' then '제2외국어'

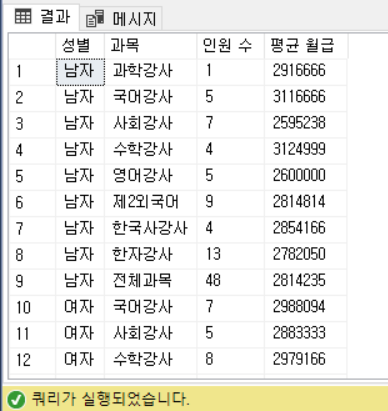
when 강사.과목 like '한국사' then '한국사강사' else '한자강사' end as '과목',

case when 강사.성별 like 'm' then '남자' else '여자' end as '성별',

case when 강사.super번호 is null then 강사.연봉/12 else 강사.연봉/12 end as '월급'

from dbo.강사) as table1

group by ROLLUP(table1.성별, table1.과목);





**5. 아직 졸업하지 않은 학생 회원을 년도, 학년, 성별로 나누어서 인원수를 구한다.**

select table1.년생 as '년생', table1.학년, table1.성별, count(\*) "인원수"

from ( select 회원.생년월일 as '생년월일', 학생.학교 as '학교명',

CASE WHEN 회원.생년월일 <= '19500101' THEN '1950년생 미만'

WHEN 회원.생년월일 between '19600101' and '19691231' THEN '1960년생'

WHEN 회원.생년월일 between '19700101' and '19791231' THEN '1970년생'

WHEN 회원.생년월일 between '19800101' and '19891231' THEN '1980년생'

WHEN 회원.생년월일 between '19900101' and '19991231' THEN '1990년생'

WHEN 회원.생년월일 between '20000101' and '20091231' THEN '2000년생'

WHEN 회원.생년월일 between '20100101' and '20191231' THEN '2010년생' END AS '년생',

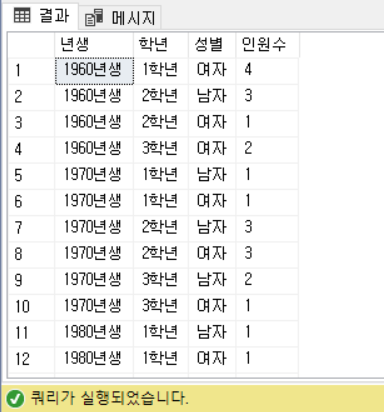
CASE WHEN 학생.학년= 1 THEN '1학년' WHEN 학생.학년=2 THEN'2학년' ELSE '3학년' END AS '학년',

CASE WHEN 회원.성별 like 'm' THEN '남자' ELSE '여자' END AS '성별'

from dbo.회원, dbo.학생

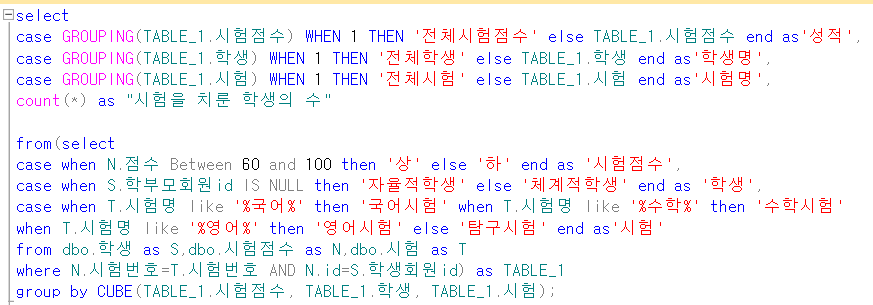
where 회원.id=학생.학생회원id and 학생.졸업여부='n') as table1

group by table1.년생, table1.학년, table1.성별





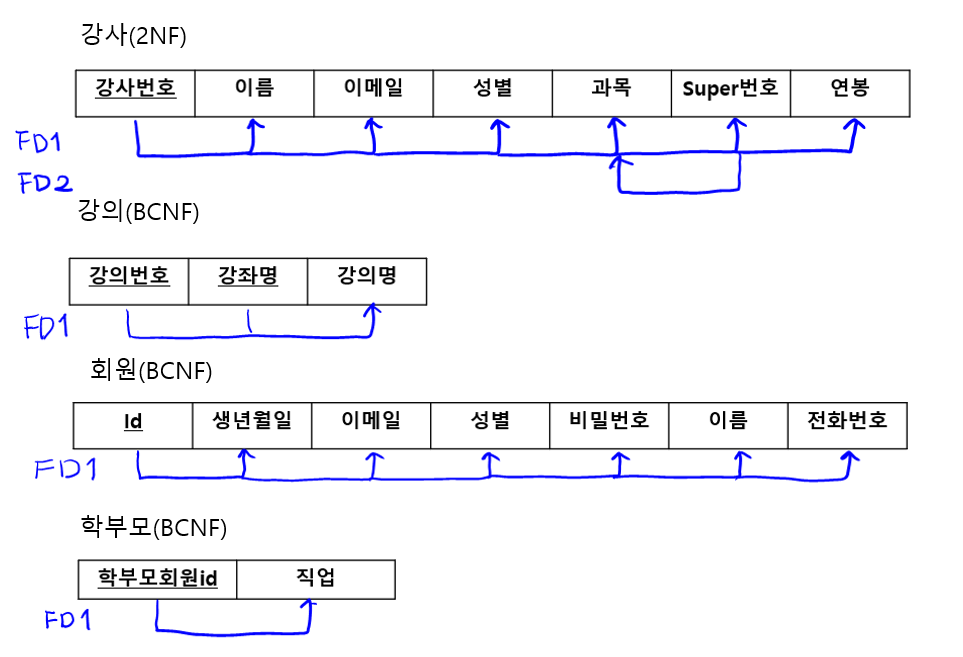
**6. 모든 회원들 중 시험을 본 학생들에 한하여 각 과목에 관한 시험을 분류한 다음, 관리하는 학부모가 있는지 또한 분류한다. 그리고 시험 점수에 따라 성적을 상, 하로 나누고 이렇게 나누어진 시험을 본 학생의 수를 세는 질의**



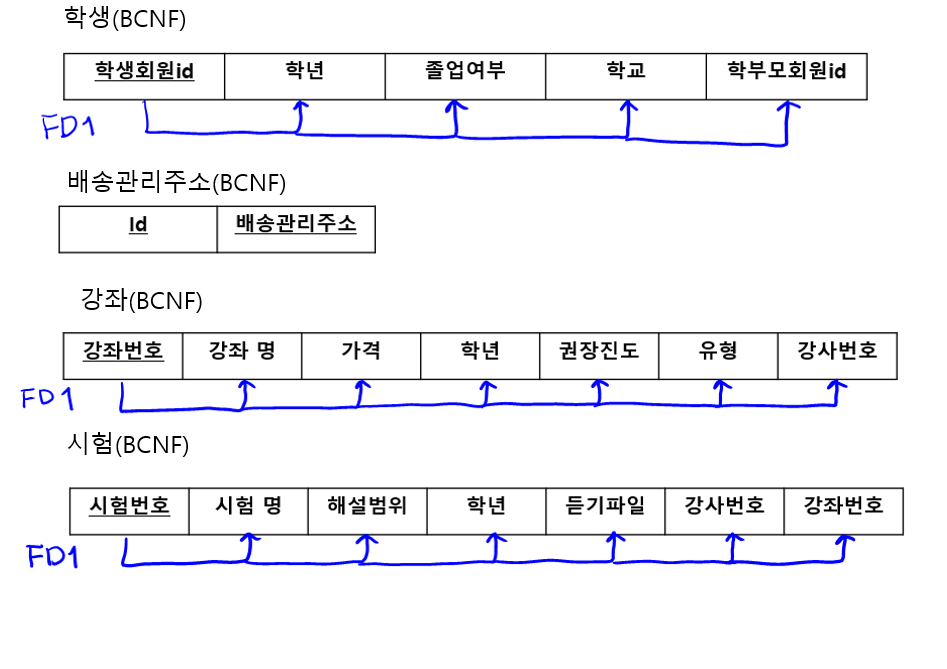
 

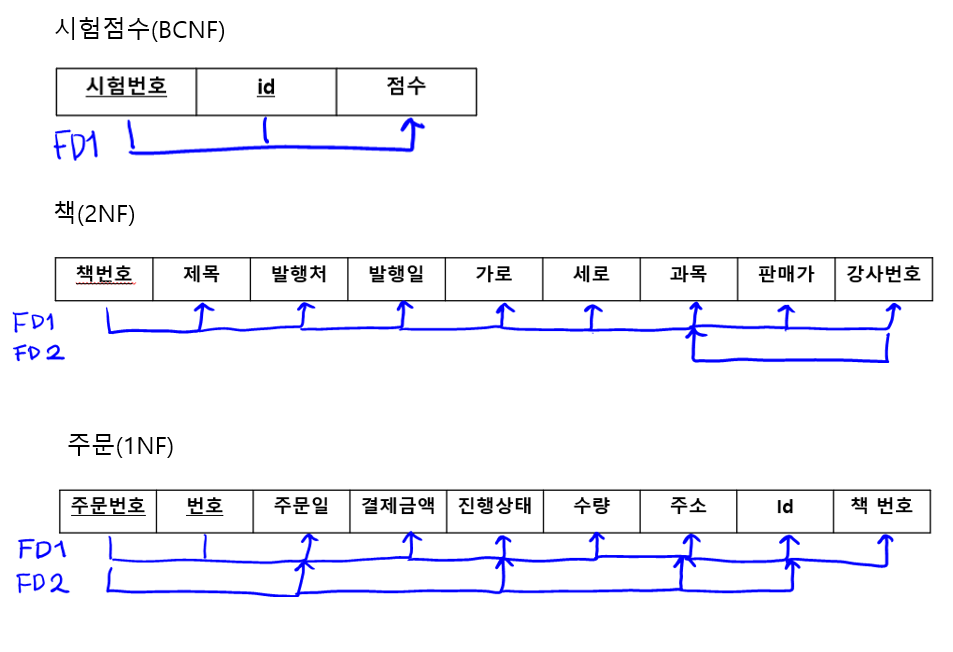
***8. 정규화 레벨 추가***

* 각 entity에서 보여지는 함수 종속성을 나타내고 정규레벨이 무엇인지 구하였습니다.



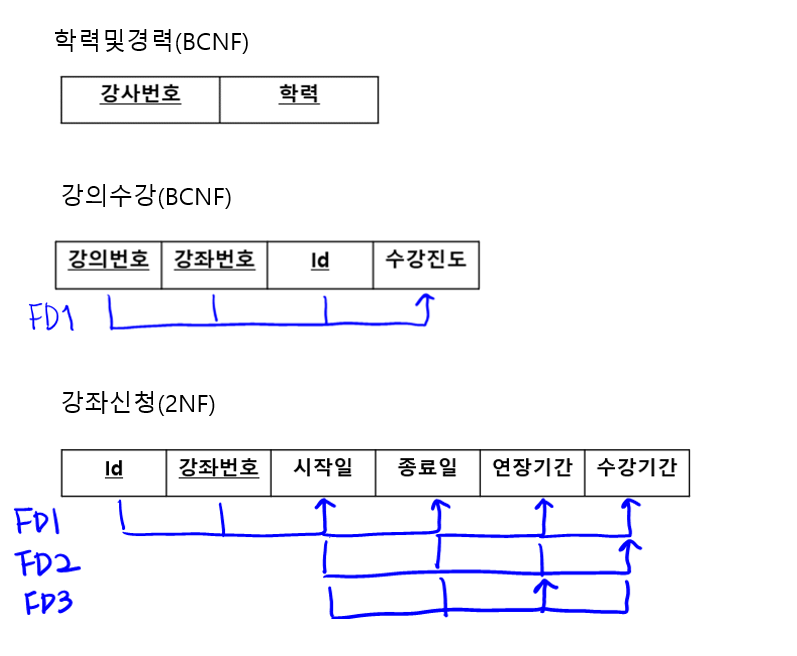
🡪강사에서 super 번호로 과목을 정할 수 있지만 정규화 하지 않은 이유는 강사를 과목별로 분류해서 ASP를 구현하기 위해서 입니다.

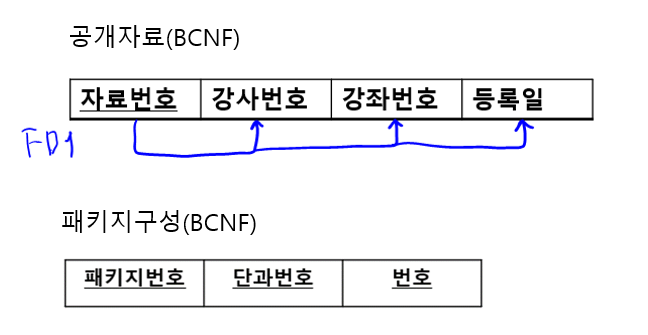




🡪책 entity에서 과목과 강사번호가 3NF를 위반하지만 정규화 하지 않은 이유는 책의 과목별로 구분해서 회원들이 책을 구매할 수 있게 하기위해서 입니다.

🡪주문 entity는 FD1,FD2에 따라서 정규화를 진행하면 원하는 정보를 얻기 위해 JOIN 연산이 많이 사용될 것이라고 예상하여 저렇게 두었습니다.





🡪강좌신청에서 FD2,FD3가 3NF를 위반하지만 마찬가지로 join연산을 줄이기 위해서 정규화를 하지 않았습니다.