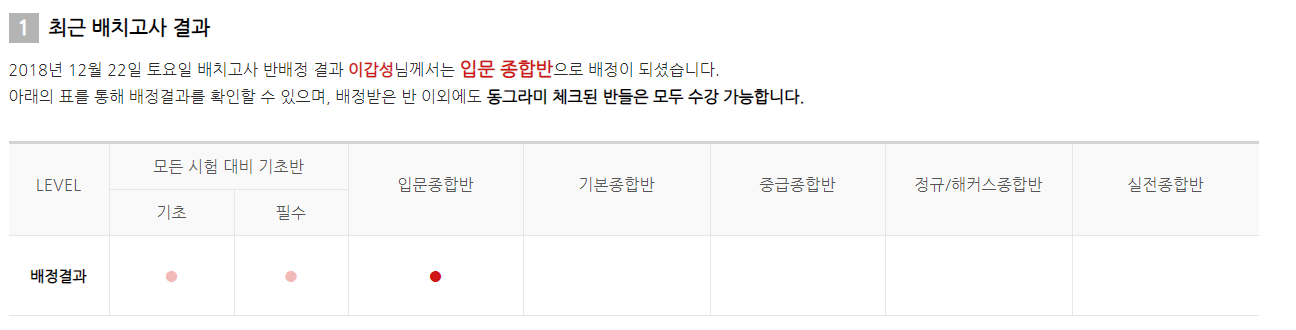
**데이터베이스 기말 프로젝트 보고서**

**학번: 20135151**

**컴퓨터공학과, 이갑성**

1. **프로젝트 개요**

요즘 대학생들이나 취준생들한테 운전면허증과 같을 정도로 흔히 가지고 있는 토익 성적이다. 또한 성인이 된 후 토익 학원 웹사이트를 단 한 번도 접속하지 않은 사람은 거의 없을 것이다. 본인 또한 이번 여름에 대학졸업 기준 요건을 충족하기 위해 토익을 공부하였다. 강의 정보를 찾아보기 위해 해커스어학원 웹사이트를 이용하면서 중간중간 참 편리한 기능들을 잘 만들었다고 생각했다. 그 이유는 수많은 강좌 중 어떤 강좌를 들을지 모를 때 토익배치고사 결과를 통한 맞춤형 강좌를 자동으로 사용자에게 보여주는 것과 자신의 현재 상황에 맞는 조건을 입력하면 그 조건에 따른 강의가 필터링 되어 보여주고 듣고자 하는 강의를 장바구니에 추가하여 결제하는 것이 매우 편리했기 때문이다. 이번 프로젝트를 통해 이런 편리한 시스템을 한번 DB를 통해 대략 구현해보기로 하였다. 





1. **응용프로그램 기능**

[응용프로그램: 해커스어학원 웹사이트]

1) 배치고사 결과에 따른 추천 강의 제공

토익이란 공부에 처음 접하거나 오랜만에 영어공부를 하여 자신의 수준이 어느 정도인지 모르는 고객들에게 배치고사라는 시스템을 통하여 현재 자신의 수준(레벨)을 알려주어 그 수준(레벨)에 알맞은 강의를 찾아 제공하는 기능입니다.

2) 나에게 맞는 맞춤 강의 찾기

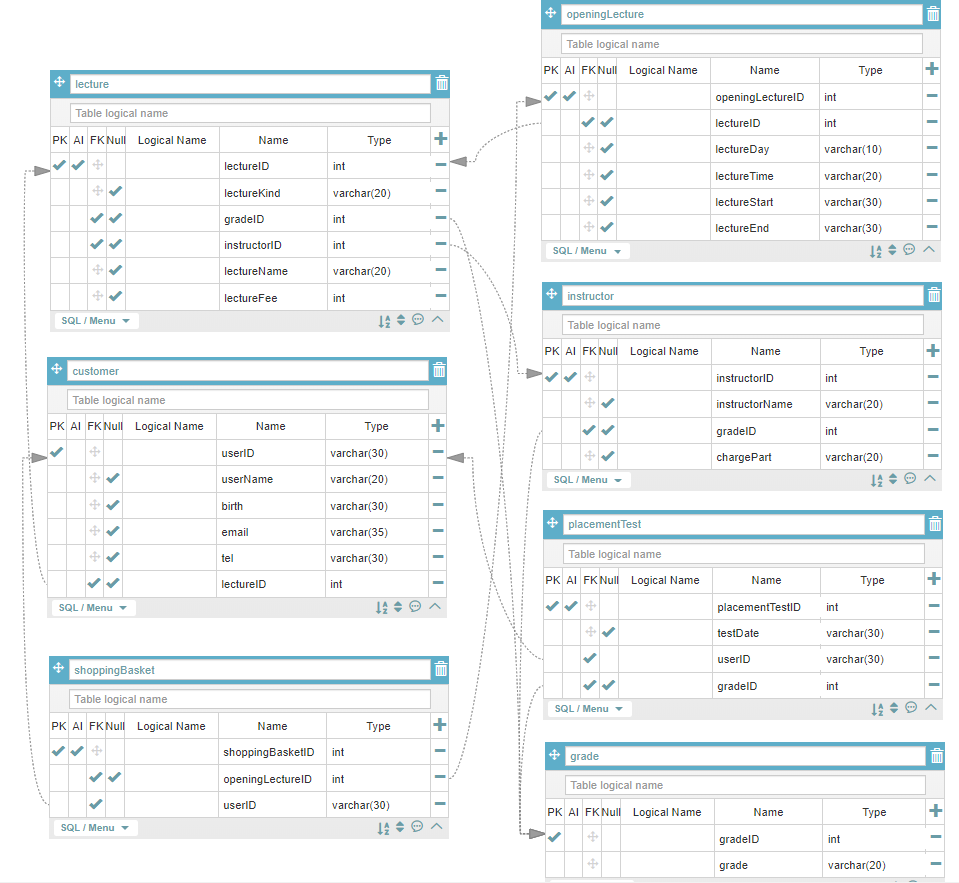
고객마다 수준, 가능한 시간대, 하고자 하는 공부 종류 등이 모두 다르기에 그에 맞는 조건을 입력받아 그 조건에 해당하는 강의를 선별하여 고객들에게 맞춤 강의 정보를 제공하는 기능입니다.

3) 장바구니

고객이 듣고자 하는 강의를 임시로 담아두는 장바구니 기능

1. **데이터베이스 설계: Create table문, 예제 데이터 삽입**

[데이터베이스 설계]



/\*Toeic 데이터 베이스 생성\*/

create database Toeic

go

/\*Toeic 데이터베이스 사용\*/

use Toeic

go

/\*

lecture 테이블은 강의에 대한 정보를 담는 테이블이다.

테이블 구조는 강의ID 를 저장하고 null을 허용하지않는 정수형 lectureID

(강의ID는 primary key로 설정하여 강의를 식별할 수 있게 하고

강의 ID생성은 identity(1000, 1)로 설정하여 1001부터 자동으로 값을 넣어준다.)

수업종류를 저장하는 크기가 20인 가변문자열 lectureKind

수준(레벨)의 ID를 저장하는 정수형 gradeID

강사ID를 저장하는 정수형 gradeID (null을 허용하지 않음)

강의명을 저장하는 크기가 20인 가변문자열 lectureName

수강료를 저장하는 정수형 lectureFee

\*/

create table lecture(

lectureID int identity(1001, 1) not null primary key,

lectureKind varchar(20),

gradeID int,

instructorID int not null,

lectureName varchar(20),

lectureFee int

)

/\*

customer 테이블은 고객의 정보를 담는 테이블이다.

테이블 구조는 고객의 ID를 저장하는 크기가 30인 가변문자열 userID

(고객을 식별하는 기준으로 지정하기 위해 primary key로 지정한다.

primary key 특성을 맞추기 위해 null을 허용하지 않음)

고객의 이름을 저장하는 크기가 20인 가변문자열 userName

고객의 생년월일을 저장하는 크기가 30인 가변문자열 birth

고객의 휴대전화번호를 저장하는 크기가 30인 가변문자열 tel

고객이 수강중인 강의ID를 저장하는 정수형 lectureID

\*/

create table customer(

userID varchar(30) not null primary key,

userName varchar(20),

birth varchar(30),

email varchar(35),

tel varchar(30),

lectureID int

)

/\*

shoppingBasket 테이블은 고객의 장바구니로 넣어둔 강의 정보를 담는 테이블이다.

테이블 구조는 장바구니의ID를 저장하는 정수형 shoppingBasketID

(장바구니ID는 primary key로 설정하여 장바구니를 식별할 수 있게 하고

장바구니ID생성은 identity(3000, 1)로 설정하여 3001부터 자동으로 값을 넣어준다.

primary key 특성을 맞추기 위해 null을 허용하지 않음)

고객의ID를 저장하는 크기가 30인 가변문자열 userID (null을 허용하지 않음)

장바구니 담은 개설한 강의ID를 저장하는 정수형 openingLectureID (null을 허용하지 않음)

\*/

create table shoppingBasket(

shoppingBasketID int identity(3001, 1) not null primary key,

userID varchar(30) not null,

openingLectureID int not null

)

/\*

openingLecture 테이블은 개설한 강의의 정보를 담는 테이블이다.

테이블 구조는 개설한 강의ID를 저장하는 정수형 openingLectureID

(개설한 강의ID는 primary key로 설정하여 개설한 강의를 식별할 수 있게 하고

개설한 강의ID생성은 identity(4000, 1)로 설정하여 4001부터 자동으로 값을 넣어준다.

primary key 특성을 맞추기 위해 null을 허용하지 않음)

강의ID를 저장하는 정수형 lectureID (null을 허용하지 않음)

개설한 강의 요일을 저장하는 크기가 10인 가변문자열 lectureDay

개설한 강의 수업 시간을 저장하는 크기가 20인 가변문자열 lectureTime

개설한 강의가 시작하는 날짜를 저장하는 크기가 20인 가변문자열 lectureStart

개설한 강의가 끝나는 날짜를 저장하는 크기가 20인 가변문자열 lectureEnd

\*/

create table openingLecture(

openingLectureID int identity(4001, 1) not null primary key,

lectureID int not null,

lectureDay varchar(10),

lectureTime varchar(20),

lectureStart varchar(20),

lectureEnd varchar(20),

)

/\*

instructor 테이블은 강사의 정보를 담는 테이블이다.

테이블 구조는 강사ID를 저장하는 정수형 instructorID

(강사ID는 primary key로 설정하여 강사를 식별할수있게 하고

강사ID생성은 identity(5000, 1)로 설정하여 5001부터 자동으로 값을 넣어준다.

primary key 특성을 맞추기 위해 null을 허용하지 않음)

강사명을 저장하는 크기가 20인 가변문자열 instructorName

강사가 맡는 강의의 수준(레벨)을 저장하는 정수형 gradeID

강사가 맡는 수업 종류를 저장하는 크기가 20인 가변문자열 chargePart

\*/

create table instructor(

instructorID int identity(5001, 1) not null primary key,

instructorName varchar(20),

gradeID int,

chargePart varchar(20)

)

/\*

placementTest 테이블은 배치고사에 대한 정보를 담는 테이블이다.

테이블 구조는 배치고사ID를 저장하는 정수형 placementTestID

(배치고사ID는 primary key로 설정하여 배치고사를 식별할수있게 하고

배치고사ID생성은 identity(6000, 1)로 설정하여 6001부터 자동으로 값을 넣어준다.

primary key 특성을 맞추기 위해 null을 허용하지 않음)

배치고사를 본 유저ID를 저장하는 크기가 30인 가변문자열 userID (null을 허용하지 않음)

배치고사를 본 날짜가 들어있는 크기가 30인 가변문자열 testDate

배치고사를 통한 수준(레벨)을 저장하는 정수형 gradeID

\*/

create table placementTest(

placementTestID int identity(6001, 1) not null primary key,

userID varchar(30) not null,

testDate varchar(30),

gradeID int

)

/\*

grade 테이블은 영어 수준(레벨)에 대한 정보를 담는 테이블이다.

테이블 구조는 각각의 수준(레벨)의 ID를 저장하는 정수형 gradeID

(primary key로 설정하여 수준(레벨)을 식별하게 해준다.

primary key 특성을 맞추기 위해 null을 허용하지 않음)

수준(레벨)을 저장하는 크기가 20인 가변문자열 grade

\*/

create table grade(

gradeID int not null primary key,

grade varchar(20)

)

/\*foreign key 설정\*/

-- lecture테이블의 instructorID 컬럼을 instructor테이블의 instructorID를 참조하게 foreign key 설정한다.

alter table lecture

add foreign key(instructorID) references instructor(instructorID)

-- lecture테이블의 gradeID 컬럼을 grade테이블의 gradeID를 참조하게 foreign key 설정한다.

alter table lecture

add foreign key(gradeID) references grade(gradeID)

-- customer테이블의 lectureID 컬럼을 lecture테이블의 lectureID를 참조하게 foreign key 설정한다.

alter table customer

add foreign key(lectureID) references lecture(lectureID)

-- shoppingBasket테이블의 openingLectureID 컬럼을 openingLecture테이블의 openingLectureID를 참조하게 foreign key 설정한다.

alter table shoppingBasket

add foreign key(openingLectureID) references openingLecture(openingLectureID)

-- shoppingBasket테이블의 userID 컬럼을 customer테이블의 userID를 참조하게 foreign key 설정한다.

alter table shoppingBasket

add foreign key(userID) references customer(userID)

-- openingLecture테이블의 lectureID 컬럼을 lecture테이블의 lectureID를 참조하게 foreign key 설정한다.

alter table openingLecture

add foreign key(lectureID) references lecture(lectureID)

-- instructor테이블의 gradeID 컬럼을 grade테이블의 gradeID를 참조하게 foreign key 설정한다.

alter table instructor

add foreign key(gradeID) references grade(gradeID)

-- placementTest테이블의 gradeID 컬럼을 grade테이블의 gradeID를 참조하게 foreign key 설정한다.

alter table placementTest

add foreign key(gradeID) references grade(gradeID)

-- placementTest테이블의 userID 컬럼을 customer테이블의 userID를 참조하게 foreign key 설정한다.

alter table placementTest

add foreign key(userID) references customer(userID)

/\*grade 데이터 입력\*/

insert into grade values(10, '기초')

insert into grade values(20, '필수')

insert into grade values(30, '입문')

insert into grade values(40, '기본')

insert into grade values(50, '중급')

insert into grade values(60, '정규')

insert into grade values(70, '실전')

/\*instructor 데이터 입력\*/

insert into instructor values('고희정', 30, 'LC')

insert into instructor values('김현서', 50, 'LC')

insert into instructor values('윤인아', 40, 'LC')

insert into instructor values('김진태', 30, 'RC')

insert into instructor values('강소영', 50, 'RC')

insert into instructor values('조성재', 70, 'RC')

insert into instructor values('심동혁', 40, 'Part7')

insert into instructor values('이은빈', 50, 'Part7')

insert into instructor values('박영선', 70, 'Part7')

/\*lecture 데이터 입력\*/

insert into lecture values('LC', 30, 5001, '토익 입문 LC', 70000)

insert into lecture values('LC', 50, 5002, '토익 중급 LC', 95000)

insert into lecture values('RC', 30, 5004, '토익 입문 RC', 78000)

insert into lecture values('RC', 70, 5006, '토익 입문 RC', 88000)

insert into lecture values('Part7', 50, 5008, '토익 입문 Part7', 138000)

insert into lecture values('Part7', 70, 5008, '토익 실전 Part7', 138000)

/\*openingLecture 데이터 입력\*/

insert into openingLecture values(1001, '주중', '10:00~10:50', '2019-12-02', '2019-12-30')

insert into openingLecture values(1002, '주말', '13:00~13:50', '2019-12-02', '2019-12-30')

insert into openingLecture values(1003, '월수금', '09:00~09:50', '2019-12-02', '2019-12-30')

insert into openingLecture values(1004, '화목금', '14:00~14:50', '2019-12-02', '2019-12-30')

insert into openingLecture values(1005, '주중', '11:00~11:50', '2019-12-02', '2019-12-30')

insert into openingLecture values(1006, '주말', '15:00~15:50', '2019-12-02', '2019-12-30')

/\*customer 데이터 입력\*/

insert into customer values('sonsational','손흥민', '1992-07-08', 'son@naver.com', '010-1234-5678', null)

insert into customer values('goalblocker', '김승규', '1990-09-30', 'block@daum.net', '010-1472-5893', 1004)

insert into customer values('youngboy' ,'이강인', '2001-02-19', 'valencia@naver.com', '010-3256-8745', 1006)

insert into customer values('bullstriker', '황희찬', '1996-01-26', 'bull@google.com', '010-4265-9624', null)

insert into customer values('zulatan', '김신욱', '1988-04-14', 'tallGuy@daum.net', '010-7865-5214', null)

insert into customer values('shine', '황의조', '1992-03-27', 'hawng@daum.net', '010-5369-1478', 1002)

/\*placementTest 데이터 입력\*/

insert into placementTest values('sonsational', '2019-11-13', 70)

insert into placementTest values('goalblocker', '2019-11-07', 30)

insert into placementTest values('youngboy', '2019-11-11', 50)

insert into placementTest values('bullstriker', '2019-10-30', 20)

insert into placementTest values('zulatan', '2019-11-01', 50)

insert into placementTest values('shine', '2019-11-09', 30)

1. **응용 프로그램 기능 구현**

기능1. 배치고사를 통한 맞춤 강의 추천

손흥민이라는 고객이 자신의 영어 수준이 어느 정도 인지 잘 몰라서 배치고사를 보고 난 후에 결과가 실전반이라는 결과가 나왔다. 그 결과를 배치고사(placementTest) 테이블에

손흥민 ID와 배치고사 본 날짜, 수준(레벨)을 저장해준다. 손흥민이 “배치고사 결과보기” 라는 페이지를 들어갔을 때 배치고사 결과에 따른 수준(레벨)을 DB에서 가져와 손흥민에게 보여주고 강의(lecture)테이블과 개설강의(openingLecture) 테이블을 조인하고 강의 (lecture) 테이블과 강사(instructor) 테이블을 조인하여 수준(레벨)에 따른 강의 정보를 보여 준다.

예) 손흥민의 배치고사 결과등급을 가져와 맞춤 강의 추천해주기

[SQL문]

select l.lectureKind "수업종류", l.lectureName "강의명", i.instructorName "강사명", o.lectureDay "수강요일", o.lectureTime "수강시간", l.lectureFee "수강료"

from lecture l, instructor i, openingLecture o

where l.instructorID = i.instructorID and l.lectureID = o.lectureID

and l.gradeID = (select gradeID

from placementTest

where userID = (

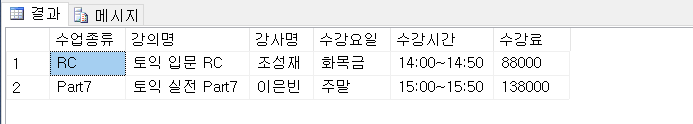
select userID

from customer

where userName = '손흥민'

))

[결과]



기능2. 나에게 맞는 맞춤 강의 찾기

수업종류, 레벨, 수업요일 등 조건을 선택할 수 있는 맞춤 강의 페이지를 고객이 들어가 본인에 적합한 조건을 선택하여 검색을 하면 해당 조건들을 가져와 강의, 강사, 개설한 강의 테이블을 테이블 조인을 통해서 합친 후에 고객의 찾고자 하는 조건을 where 절에 넣어서 필터링해주고 필터링 된 강의 정보를 고객에게 보여준다.

예) 고객이 찾고자 하는 강의의 조건( 1)수업종류: LC, 2)레벨: 입문, 3)수업요일: 주중)에 맞춰서 맞춤 강의 찾아주기

[SQL문]

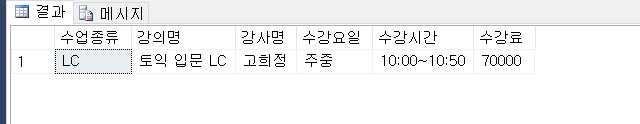
select l.lectureKind "수업종류", l.lectureName "강의명", i.instructorName "강사명", o.lectureDay "수강요일", o.lectureTime "수강시간", l.lectureFee "수강료"

from lecture l, instructor i, openingLecture o

where l.instructorID = i.instructorID and l.lectureID = o.lectureID

and l.lectureKind = 'LC' and i.gradeID = 30 and o.lectureDay = '주중'

[결과]



기능3. 장바구니 추가하는 기능

고객이 찾은 개설한 강의를 장바구니에 담고자 할 때 장바구니 넣기를 클릭하면 해당 개설한 강의 ID와 고객의ID를 insert문을 통해서 장바구니(shoppingBasket) 테이블에 넣어준다.

예) 김신욱라는 고객이 주말을 이용해서 영어공부를 하려고 한다. 주말에 가능한 강의를 모두 장바구니에 담아 두려고한다.

[SQL문]

insert into shoppingBasket

(c.userID, o.openingLectureID)

select c.userID, o.openingLectureID

from customer c, openingLecture o

where c.userID = 'zulatan' and o.openingLectureID in (select o.openingLectureID from openingLecture o where o.lectureDay = '주말' )

[결과]

