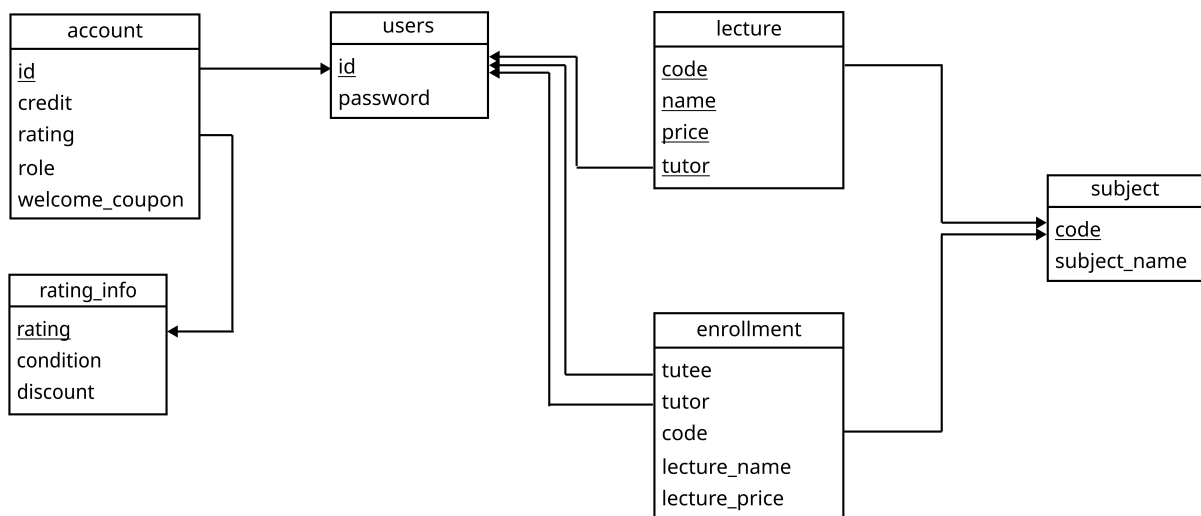


COSE371 Term Project

2019320137 컴퓨터학과 황상민

1. Schema diagram



추가 기능 구현을 위해 account 테이블에 welcome_coupon이라는 속성을 추가하였습니다.

2. 추가 기능 구현

1. Credit 선물 기능

원하는 사용자에게 credit을 선물할 수 있습니다. 메인 페이지에서 'Credit gift'라는 버튼을 누르면, 선물하기 페이지로 이동합니다. 이 페이지에서 선물 받는 사람의 아이디와 주고자 하는 credit의 양을 입력하면 자신의 credit을 원하는 사람에게 선물할 수 있습니다.

2. 신규 가입 쿠폰 기능

새로 회원가입하는 회원마다 welcome coupon을 발급합니다. 이는 account 테이블에 boolean 값으로 명시되어 있습니다. 이 웰컴 쿠폰을 사용하면 1회에 한해 강의 하나를 무료로 들을 수 있습니다.

3. 함수별 기능 설명

login()

'/login' 경로로 가는 라우팅을 담당하는 함수입니다.

- GET으로 접근시 로그인 페이지인 login.html을 출력합니다.
- POST로 접근시 form data를 참조하여, id, password 일치 여부를 확인합니다.

```
SELECT id, password FROM users WHERE id=(%s) and password=(%s)
```

일치하는 유저 정보가 없을 경우 login_fail.html을 보여주고, 로그인에 성공했을 경우 세션의 user_id라는 키로 유저 아이디를 저장한 뒤 '/' 경로로 redirect합니다.

signup()

'/signup' 경로로 가는 라우팅을 담당하는 함수입니다.

- GET으로 접근시 회원가입 페이지인 signup.html을 출력합니다.
- POST로 접근시 form data를 참조하여, id, password, role을 바탕으로 새로운 유저를 생성합니다. 이때 users 테이블과 account 테이블 모두에 새로운 행을 추가합니다.

```
INSERT INTO users VALUES (%s, %s)
```

```
INSERT INTO account(id, credit, rating, role) VALUES (%s, %s, %s, %s)
```

이때 credit=10000, rating=welcome으로 설정합니다.

중복 ID 등으로 인해 회원가입에 실패할 경우 signup_fail.html을 보여줍니다. 회원가입에 성공했을 경우 signup_succes.html을 보여줍니다.

logout()

세션에 저장한 user_id 값을 삭제하고, '/' 경로로 redirect합니다. '/'에 접근하면 세션 정보가 없으므로 자연스럽게 '/login' 경로로 redirect되게 됩니다.

index()

‘/’ 경로로 가는 라우팅을 담당하는 함수입니다.

- 세션을 참조하여 로그인이 되어 있지 않으면 ‘/login’으로 redirect합니다.
- 로그인이 되어 있으면 유저 정보, 강의 목록 등이 포함된 메인 페이지를 보여줍니다. 세션에는 user_id라는 변수에 로그인한 사용자의 id가 저장되어 있습니다.
- 로그인이 되어 있을 경우, account 테이블을 참조하여 사용자의 정보를 가져와 출력합니다.

```
SELECT id, credit, rating, role, welcome_coupon FROM account WHERE id=(%s)
```

- 또한 lecture 테이블을 참조하여 현재 개설된 강의 목록을 표시합니다.

```
SELECT code, name, price, tutor FROM lecture
```

- 그리고 enrollment 테이블과 subject 테이블을 참조하여 수강 등록 횟수가 가장 많은 subject, lecture, tutor를 표시합니다.

```
SELECT tutor FROM enrollment  
GROUP BY tutor  
ORDER BY count(tutor) DESC  
LIMIT 1
```

```
SELECT lecture_name FROM enrollment  
GROUP BY lecture_name  
ORDER BY count(lecture_name) DESC  
LIMIT 1
```

```
SELECT subject_name FROM subject  
where code = (  
    SELECT code FROM enrollment  
    GROUP BY code  
    ORDER BY count(code) DESC  
    LIMIT 1  
)
```

index() 함수에서 가져온 정보들은 index.html에 전달되어 렌더링됩니다.

get_users()

‘/user’로 가는 라우팅을 담당하는 함수입니다. admin function 중 user info를 출력하는 기능을 수행합니다. admin만 접근이 가능하며, admin이 아닌 사용자가 접근할 경우 abort합니다. admin이 접근하면 users 테이블을 참조하여 모든 정보를 가져옵니다.

```
SELECT id, password FROM users
```

여기서 가져온 정보들은 print_user.html에 전달되어 렌더링됩니다.

enrollment()

‘/enrollment’로 가는 라우팅을 담당하는 함수입니다.

- GET으로 접근시 admin function 중 enrollment 정보를 출력하는 기능을 수행합니다. enrollment 테이블을 참조하여 모든 정보를 가져옵니다.

```
SELECT tutee, tutor, code, lecture_name, lecture_price FROM enrollment
```

여기서 가져온 정보들은 print_enrollments.html에 전달되어 렌더링됩니다.

- POST로 접근시 새로운 enrollment 정보를 추가하는 기능을 수행합니다.
 - form data를 참조하여 tutee, tutor, code, name, price 및 discount(할인되는 가격)를 가져옵니다.
 - 만약 tutee와 tutor가 같은 경우일 경우 결제에 실패하며 enroll_fail.html을 렌더링합니다.
 - account 테이블을 참조하여 tutee와 tutor 정보를 가져옵니다.

```
SELECT id, credit, rating, role FROM account WHERE id=(%s) # tutee id  
SELECT id, credit, rating, role FROM account WHERE id=(%s) # tutor id
```

- 만약 tutee가 결제할 금액만큼의 credit을 가지고 있지 않은 경우 결제에 실패합니다.
 - enrollment 테이블을 참조하여 중복된 등록 정보가 있는지 확인합니다. 이미 결제했던 강의라면 결제에 실패합니다.

```
SELECT tutee, tutor, code, lecture_name, lecture_price FROM enrollment  
WHERE tutee=(%s) AND tutor=(%s) AND code=(%s) AND lecture_name=(%s) and lecture_price=(%s)
```

- 결제 실패 조건에 해당되지 않는 경우 enrollment 테이블에 새로운 등록 정보를 추가합니다.

```
INSERT INTO enrollment VALUES (%s, %s, %s, %s, %s)
```

또한 tutor과 tutee의 credit을 업데이트합니다. (welcome coupon을 사용한 경우 tutee의 credit은 바뀌지 않습니다.)

```
UPDATE account SET credit=(%s) WHERE id=(%s) # tutee credit
UPDATE account SET credit=(%s) WHERE id=(%s) # tutor credit
```

rating_info 테이블을 참조하여 변경된 credit에 따른 rating 변화 또한 반영합니다.

```
SELECT rating, condition, discount FROM rating_info ORDER BY condition DESC
```

```
UPDATE account SET rating=(%s) WHERE id=(%s) # tutee rating
UPDATE account SET rating=(%s) WHERE id=(%s) # tutor rating
```

welcome coupon을 사용한 경우라면 account 테이블을 참조하여 쿠폰 사용 여부 또한 업데이트합니다.

```
UPDATE account SET welcome_coupon=false WHERE id=(%s)
```

- 마지막으로, DB 업데이트가 끝난 뒤엔 '/enrollment/my'로 redirect하여 자신의 전체 등록 정보를 볼 수 있도록 합니다.

my_enrollment()

'/enrollment/my'로 가는 라우팅을 담당하는 함수입니다. 세션을 참조하여 로그인된 사용자의 id를 기반으로, 사용자의 현재 등록 정보를 출력합니다.

```
SELECT tutee, tutor, subject_name, lecture_name, lecture_price
FROM enrollment NATURAL JOIN subject
WHERE tutee = (%s)
```

여기서 참조한 정보는 my_enrollment.html에 전달되어 렌더링됩니다.

my_lecture()

'/lecture/my'로 가는 라우팅을 담당하는 함수입니다. 세션을 참조하여 로그인된 사용자의 id를 가져온 뒤, id를 기반으로 자신과 관련된 강의 정보를 출력합니다.

- 먼저 사용자의 id를 기반으로 account 테이블을 참조하여 계정 정보를 가져옵니다.

```
SELECT id, credit, rating, role FROM account WHERE id=(%s)
```

- 사용자의 role이 tutor라면 자신이 강의하는 강의 목록과, 자신이 수강하는 강의 목록을 모두 보여줍니다.

```
SELECT subject_name, lecture_name, tutee, lecture_price  
FROM enrollment NATURAL JOIN subject WHERE tutor=(%s)
```

```
SELECT subject_name, lecture_name, tutor, lecture_price  
FROM enrollment NATURAL JOIN subject WHERE tutee=(%s)
```

- tutee라면 자신이 수강하는 강의 목록만 보여줍니다.

```
SELECT subject_name, lecture_name, tutor, lecture_price  
FROM enrollment NATURAL JOIN subject WHERE tutee=(%s)
```

여기서 가져온 정보들은 my_lecture.html에 전달되어 렌더링됩니다.

lecture_form()

'/lecture/new'로 가는 라우팅을 담당하는 함수입니다. 새로운 강의를 추가하는 폼 화면을 출력하는 기능을 수행합니다. 세션을 참조하여 로그인되어 있지 않으면 접근을 차단합니다. 로그인되어 있다면 subject 테이블을 참조하여 subject 목록을 가져온 뒤, lecture_form.html에 렌더링합니다.

```
SELECT code, subject_name FROM subject
```

add_lecture()

'/lecture'로 가는 라우팅을 담당하는 함수입니다. 새로운 강의를 추가하는 기능을 수행합니다. form data를 참조하여 code, name, price를 가져오고, 세션을 참조하여 강의를 추가하려는 사용자의 아이디를 가져옵니다.

가져온 정보를 바탕으로 lecture 테이블에 추가합니다.

```
INSERT INTO lecture VALUES (%s, %s, %s, %s)
```

중복 강의 등의 문제로 추가에 실패할 경우 `add_lecture_fail.html`을 렌더링합니다. 추가에 성공하면 `/` 경로로 redirect 합니다.

register_lecture()

`/lecture/register`로 가는 라우팅을 담당하는 함수입니다. 메인 페이지에서 register를 누르면 강의를 결제하는 화면을 출력하는 기능을 수행합니다.

form data를 참조하여 등록하려는 강의에 관한 정보를 가져옵니다.

세션을 참조하여 로그인한 사용자의 id를 기반으로 계정 정보를 가져옵니다.

```
SELECT id, credit, rating, role, welcome_coupon FROM account WHERE id=(%s)
```

그리고 계정의 rating을 바탕으로 rating_info를 참조하여 rating 관련 정보를 가져옵니다.

```
SELECT rating, condition, discount FROM rating_info WHERE rating=(%s)
```

이렇게 가져온 정보를 바탕으로 `register_lecture.html`에 등록하려는 강의 정보 및 계정 정보, 결제 금액을 렌더링합니다.

credit_gift()

`/credit/gift`로 가는 라우팅을 담당하는 함수입니다.

- GET으로 접근시 Credit을 선물하는 화면을 출력합니다. 메인 페이지에서 Credit gift 버튼을 누르면 접근할 수 있습니다.
- POST로 접근시 원하는 사람에게 Credit을 선물하는 기능을 수행합니다.
 - form data를 참조하여 선물받는 사람의 id와 선물하고자 하는 credit 양을 가져옵니다. 또한 세션을 참조하여 보내고자 하는 사용자의 id를 가져옵니다.
 - account 테이블을 참조하여 보내는 사람과 받는 사람의 계정 정보를 가져옵니다.

```
SELECT id, credit FROM account WHERE id=(%s) # sender account  
SELECT id, credit FROM account WHERE id=(%s) # reciever account
```

- 이때 1) 보내고자 하는 사람과 받는 사람이 같거나 2) 받는 사람이 존재하지 않거나 3) 선물하는 credit이 음수이면 선물에 실패합니다.
- 그러한 경우가 아니라면 보내는 사람과 받는 사람의 credit 값을 업데이트합니다.

```
UPDATE account SET credit=(%) WHERE id=(%) # sender update
UPDATE account SET credit=(%) WHERE id=(%) # reciever update
```

이때 만약 sender가 기존에 가진 credit보다 많은 credit을 선물하는 경우 등이 발생하면 선물에 실패합니다.

- 선물 후 credit의 변화에 따라 rating 또한 업데이트합니다.

```
SELECT rating, condition, discount FROM rating_info ORDER BY condition DESC
```

```
UPDATE account SET rating=(%) WHERE id=(%) # sender rating
UPDATE account SET rating=(%) WHERE id=(%) # reciever rating
```

- 선물에 성공했을 경우 credit_gift_success.html을 렌더링하고, 실패했을 경우 credit_gift_fail.html을 렌더링합니다.

4. HTML 파일 설명

- **index.html**: 로그인 후 나오는 메인 페이지입니다. 유저 정보, 자신과 관련된 강의 정보, 현재 개설된 강의 목록 등을 조회할 수 있습니다. tutor인 경우 강의를 추가할 수 있는 버튼이 나타나며 admin인 경우 admin function도 같이 출력됩니다.
- **login.html**: 로그인 화면입니다. ID와 패스워드를 입력하고 login을 누르면 로그인되며 signup을 누르면 회원가입 화면으로 넘어갑니다.
- **login_fail.html**: 로그인 실패 화면입니다. 패스워드가 일치하지 않거나 존재하지 않는 계정으로 로그인할 경우에 출력됩니다. return을 누르면 메인 페이지로 돌아갑니다.
- **signup.html**: 회원가입 화면입니다. ID, 패스워드, role을 입력하고 sign up을 누르면 회원가입됩니다. cancel을 누르면 메인 페이지로 돌아갑니다.
- **signup_success.html**: 회원가입 성공 화면입니다. login 버튼을 누르면 로그인 화면으로 돌아갑니다.
- **signup_fail.html**: 회원가입 실패 화면입니다. 중복 아이디가 있거나 하는 경우에 표시되며 return을 누르면 회원가입 화면으로 돌아갑니다.

- **print_users.html**: admin function 중 사용자 정보를 출력하는 화면입니다. users 테이블의 모든 정보가 출력됩니다.
- **print_enrollments.html**: admin function 중 강의 등록 정보를 출력하는 화면입니다. enrollment 테이블의 모든 정보가 출력됩니다.
- **my_lecture.html**: 자신과 관련된 강의 정보를 출력하는 화면입니다. tutor인 경우 자신이 가르치는 강의와 수강하는 강의를 모두 출력되며, tutee인 경우엔 수강하는 강의만 출력됩니다.
- **lecture_form.html**: 메인 페이지에서 add 버튼을 누르면 출력되는 화면입니다. code, name, price를 입력하고 add를 누르면 강의를 추가됩니다. cancel을 누르면 메인 페이지로 돌아갑니다.
- **add_lecture_fail.html**: 중복되는 강의 등으로 강의 추가에 실패했을 때 출력되는 화면입니다. retry를 두르면 다시 강의 추가 화면으로 돌아갑니다.
- **register_lecture.html**: 메인 페이지에서 register 버튼을 누르면 출력되는 화면입니다. 등록하려는 강의 정보 및 현재 사용자 계정의 상태, 그리고 결제해야 하는 금액 등이 표시되며 confirm을 누르면 강의를 등록됩니다. welcome coupon이 남아있을 경우 쿠폰을 사용 여부를 선택할 수 있는 란이 추가로 표시됩니다. cancel을 누르면 메인 페이지로 돌아갑니다.
- **enroll_fail.html**: 강의 등록이 실패했을 경우 보이는 화면입니다. 실패 이유가 출력되며 return을 누르면 메인 페이지로 돌아갑니다.
- **my_enrollment.html**: 강의 등록이 성공했을 경우 보이는 화면입니다. 자신이 tutee로 등록한 모든 강의를 보여줍니다. return을 누르면 메인 페이지로 돌아갑니다.
- **credit_gift.html**: 메인 페이지에서 Credit gfit를 누르면 보이는 화면입니다. credit을 선물할 사람의 아이디와 선물할 credit 양을 입력할 수 있습니다. confirm을 누르면 선물되고, cancel을 누르면 메인 페이지로 돌아갑니다.
- **credit_gift_success.html**: credit 선물이 성공하면 보이는 화면입니다. return을 누르면 메인 페이지로 돌아갑니다.
- **credit_gift_success.html**: credit 선물이 실패하면 보이는 화면입니다. return을 누르면 메인 페이지로 돌아갑니다.