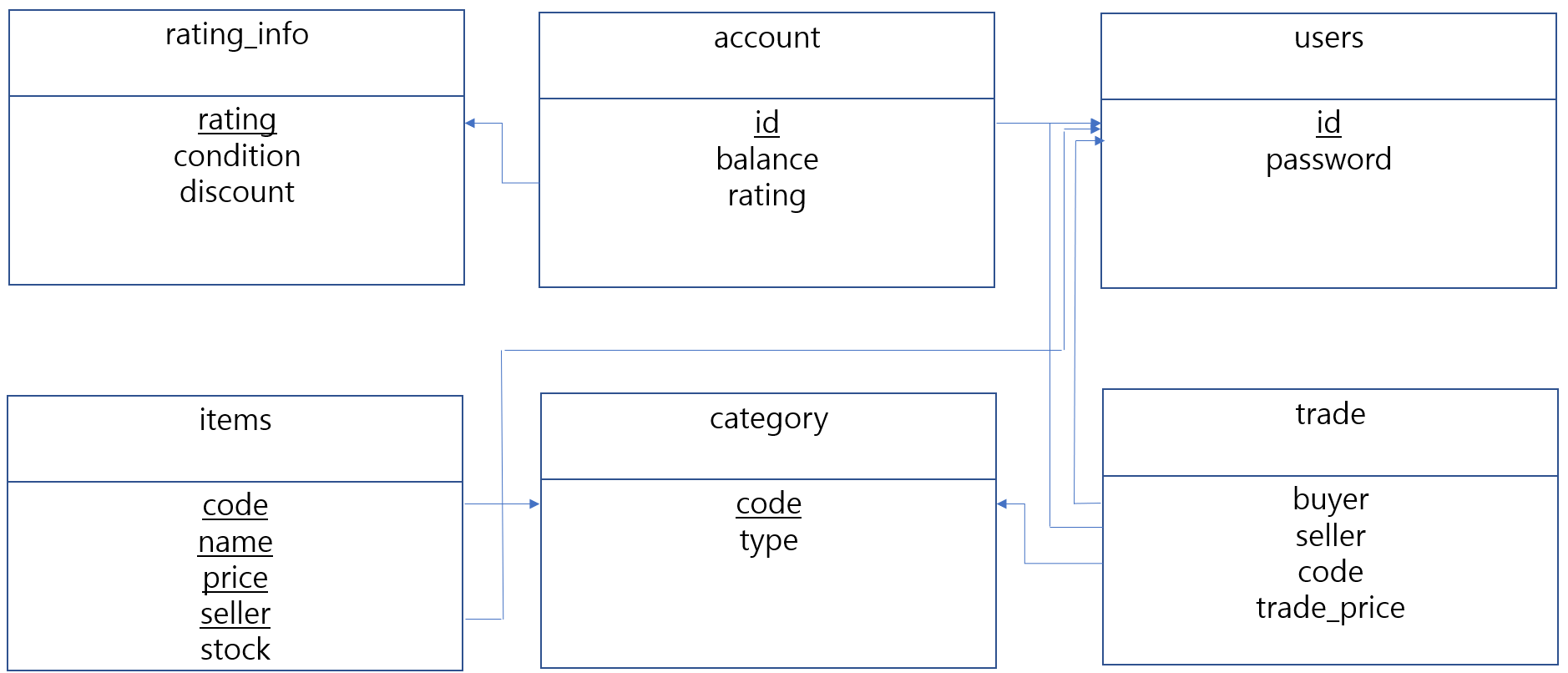
2018320151 조영민

Schema diagram



제가 이번 과제에서 추가한 기능은 1. 마이페이지와 2. 광고입니다.

마이페이지에서는 자신의 정보와 다음 등급의 정보를 확인할 수 있고 잔고를 채울 수 있습니다. 또한 자신의 사고 파는 거래 내역을 확인할 수 있고 각각에 대해 총 거래 횟수와 총 거래량이 표시됩니다. 거래상대로서 가장 많이 거래한 상대방도 표시됩니다.

광고는 메인 페이지에서 최하단에 보이며 현재 올라와 있는 물건 중 가장 적게 팔린 물건에 대해 추가 할인을 하여 팝니다. 이 때 추가 할인금액은 웹사이트에서 부담하고 여러 사람이 물건을 올린 경우 가장 판매량이 적은 사람의 물건이 광고됩니다.

함수 설명

login()

로그인 페이지를 불러옵니다. 최초 시작 지점입니다.

return\_login()

로그아웃을 했을 때 로그인 페이지로 돌아가도록 하는 함수입니다.

find\_advertisement()

추가기능의 하나인 광고를 위해 광고할 물건을 정하는 함수입니다. trade에서 code와 seller로 group by하여 개수에 따라 정렬하여 가장 위에 거래량이 가장 적은 code와 seller가 위치하도록 하는 sql을 사용하고, code는 반환 값에서 0번 인덱스 값을 추출하여 최소거래량의 code를 추출합니다. 이후 판매량이 적은 순으로 seller가 해당 code의 물건을 파는지 검사하여 찾으면 그 정보를 반환합니다.

register()

로그인 페이지에서 회원가입이나 로그인을 했을 때 동작하는 함수입니다.

로그인이라면 select \* from users where id = '' and password = '' 형식으로 정보를 검사해 정보가 맞으면 해당 유저의 메인 페이지로 넘어가고 아니면 로그인 실패 창을 띄웁니다. 이 때 메인 페이지로 넘어갈 때는 select \* from account where id = '{}'로 유저의 계좌정보를 받고 find\_advertisement()로 현재 광고정보를 받고, 아래서 설명할 get\_trade\_info()와 get\_items\_info()로 trade정보와 items정보를 받아 메인 페이지에 넘겨주며 종료합니다.

회원가입이라면 아이디가 0자리 이하면 에러용 html을 띄우고, 같은 아이디가 있는지 검사하고 없다면 users와 account에 유저 초기정보를 입력해주고 다시 로그인 화면으로 돌아갑니다. 같은 아이디가 있다면 에러용 html을 띄웁니다.

show\_info()

admin으로 접속했을 때 활용하는 기능을 위해 만든 것입니다. 현재 users와 trade의 정보를 모두 받아 이를 보여주는 show\_info.html로 이동시킵니다.

return\_home()

방금 나온 show\_info()나 아래에서 나올 물건 추가나 구매 이후 다시 메인 페이지로 돌아갈 때, 혹은 구매나 물건추가를 취소하고 싶을 때 메인 페이지로 돌아가게 해주는 함수입니다. register()에서 로그인이 성공했을 때의 기능과 동일합니다.

get\_trade\_info()

메인 페이지에서 가장 인기있는 카테고리, 최고의 구매자와 판매자를 출력하기 위해 이를 구하는 함수입니다. 가장 인기있는 카테고리는 아래의 sql을 사용하여 구하였습니다.

select type from category

where code in (with count\_code as (select code, count(code) from trade group by code)

select count\_code.code

from count\_code

where count\_code.count = (select max(count\_code.count) from count\_code))

최고의 구매자는 아래의 sql을 이용하였고, 판매자는 여기서 buyer를 seller로 바꾸었습니다.

with sum\_buyer as (select buyer, sum(trade\_price) from trade group by buyer)

select sum\_buyer.buyer from sum\_buyer

where sum\_buyer.sum = (select max(sum\_buyer.sum) from sum\_buyer)

이렇게 해서 구한 3개의 값을 이중리스트로 묶어 반환합니다.

get\_items\_info()

단순히 select \* from items로 item 정보 전체를 반환합니다.

show\_add\_page()

물건 추가를 위한 창을 띄워줍니다. 이때 가능한 물건 코드와 물건을 추가하려는 유저를 알기 위해 코드정보와 유저id정보를 html파일에 넘겨줍니다.

show\_buy\_page()

물건을 사기 위한 창을 띄워줍니다. 이때는 선택된 물건의 정보가 필요하기 때문에 선택된 물건의 정보와 사려는 유저정보, 그리고 extra라는 값을 넘겨주는데, extra는 위에서 구현하려 했던 광고를 통해 구매를 할 경우 추가되는 할인율에 대한 값입니다.

add\_item()

물건 추가 페이지에서 입력한 추가하려는 물건의 정보들을 받아 DB에 저장합니다. 이때 카테고리를 잘못 입력하거나 물건의 이름이 없거나 가격, 수량이 잘못된 경우 fail.html이라는 실패창을 출력합니다. 문제가 없을 경우 추가하려는 아이템이 이미 있는지 확인하고 없으면 items에 insert를 이용해 추가하고 있다면 이미 존재하는 물건의 정보에서 stock만을 update를 이용해 수정하였습니다.

confirm()

물건 구매가 실행되기 전 확인을 위한 창을 실행시키는 함수입니다. 선택된 물건의 정보들과 구매하려는 수량, 구매자 id와 extra를 받아 실행합니다. 우선 구매자의 rating을 확인하여 이에 대한 할인율과 extra를 기반으로 구매가격을 계산합니다. 광고하는 물품이 아닌 경우 extra는 0.0으로 설정돼 있습니다. 이후 구매수량이 물건의 수량보다 많거나, 음수이거나, 판매자가 본인이거나, 금액이 부족할 경우 fail.html창을 띄웁니다. 아니라면 여기서 받고 계산한 모든 정보를 그대로 confirm.html이라는 구매확인 창으로 넘겨줍니다.

buy\_item()

confirm창에서 확인을 누를 경우 실제로 구매를 진행하는 함수입니다. 구매자와 판매자의 잔고를 update하고, 이에 따라 변한 balance를 바탕으로 select rating from rating\_info where condition <= {} order by condition desc라는 sql을 이용해 변화된 rating 정보를 가져옵니다. 이 변한 rating 정보를 다시 account에 업데이트 시켜주고 판매된 물건의 수량은 update를 통해 구매된 만큼 감소시켜주는데, 구매된 수량이 물품의 총량과 같다면 delete를 통해 판매된 물건을 items에서 삭제합니다. 위 모든 과정이 끝나면 connect.commit()으로 한방에 저장합니다. 이후 구매 성공 후 메인 페이지로 돌아가기 전 success.html이라는 창을 띄워줍니다.

my\_page()

추가 기능인 마이페이지를 구현하기 위한 함수입니다. 유저의 현재 정보와 다음 rating을 알기 위해 select sql을 사용했으며, 각각을 user\_info와 next\_rating에 저장합니다. 이 때 유저가 이미 최고 rating 단계라면 next\_rating은 (‘max level', '0', '2.50')로 저장됩니다. 각각은 원래라면 다음 rating, 잔고, 할인율입니다. 또한 다음 단계까지 얼만큼의 잔고가 더 있어야 하는지를 위해 need\_money도 계산합니다. 아래에서는 자신의 판매 기록과 구매 기록을 보여주기 위해 필요한 것들을 가져옵니다. 자신이 구매한 모든 정보들, 총 구매 횟수와 구매량, 그리고 자신과 가장 많이 거래한 유저를 자신이 판매자일 때와 구매자 일 때를 나누어서 구했습니다. 이 때 아직 구매나 판매내역이 없을 경우 가장 많이 거래한 유저는 None에 그 유저와 거래한 횟수는 0으로 설정했습니다. 가장 많이 거래한 유저는 자신이 구매자일 때 “select seller, count(seller) from trade where buyer='{}' group by (buyer, seller) order by count desc".format(user) 라는 sql을 사용했습니다. buyer와 seller를 기준으로 묶어 count에 따라 정렬하여 가장 위에 가장 많이 거래한 유저가 오도록 하였습니다.

여기서 계산한 값들 전부를 my\_page.html에 넘겨주고 이를 실행시키면서 종료합니다.

to\_fill\_balance()

마이페이지에서 잔고를 충전하기 위한 창으로 넘어갈 때 실행되는 함수입니다. fill\_page.html을 실행시킵니다.

fill\_balance()

입력 받은 충전할 잔고를 업데이트 시킵니다. 이 때 충전하는 금액이 0 이하면 fail.html을 실행시키고 종료합니다. 아니라면 account의 해당 유저 balance를 업데이트 시키고 rating도 해당하는 rating으로 업데이트 시켜줍니다.

html파일 설명

login.html

로그인을 위한 시작 페이지로 아이디와 비밀번호를 입력 받고 로그인 또는 회원가입을 진행할 수 있습니다.

error\_login.html

로그인 시 에러상황을 처리하기 위한 페이지입니다. message와 announcement 변수에 실패 원인을 입력 받아 띄워줍니다.

main.html

맨 위 부분에서 로그인한 유저가 admin일 경우만 submit타입의 버튼 2개를 보여줍니다. 이는 admin function인 users info와 trade info입니다. 바로 아래부터는 모든 유저가 볼 수 있는 창으로 가장 인기있는 카테고리와 최고의 구매, 판매자가 표로 출력됩니다. 그 아래에는 현재 접속중인 유저의 정보가 나오고, 마이페이지를 위한 버튼과 로그아웃을 위한 버튼 2개가 존재합니다. 각각은 마이페이지와 로그인 창으로 이동시켜줍니다.

더 아래에는 items관련 정보로 add를 위한 버튼이 있습니다. 이 버튼을 누르면 구매 관련 페이지로 넘어갑니다. 아래에는 현재 trade에 올라와 있는 물건들의 표가 나오고, 표의 가장 오른쪽 열에는 각각의 물건에 대한 buy 버튼이 있습니다. 버튼을 누르면 구매를 위한 창으로 넘어갑니다.

마지막으로 제일 아래에는 광고에 대한 창입니다. 광고 대상으로 뽑힌 물건의 정보를 표로 출력합니다. 이 때 같은 물건이라도 광고 표에서 사야 5% 추가 할인이 적용됩니다.

show\_info.html

admin function을 보여주는 페이지입니다. type 인자의 값이 무엇인지에 따라 user\_info를 보여주거나 trade\_info를 보여줍니다. 각각은 users 테이블 혹은 trade 테이블 안의 모든 원소를 표로 출력해줍니다. 리턴 버튼을 눌러 메인 페이지로 돌아갈 수 있습니다.

my\_page.html

제일 상단에는 유저의 정보들이 표로 나열됩니다. 이때 현재 유저의 정보와 다음 rating에 대한 정보가 같이 나열되고, 가장 오른쪽에는 충전을 위한 버튼이 있어 이를 누르면 충전을 위한 페이지로 넘어갑니다. 아래에는 자신의 거래내역이 출력되는데 구매내역이 먼저 출력됩니다. 본인이 물건을 가장 많이 산 유저와 그 횟수가 먼저 출력되고 총 구매횟수, 구매액이 표로 출력됩니다. 아래에는 모든 구매내역이 표로 출력됩니다. 판매내역은 구매내역과 구성이 같습니다. 리턴 버튼을 누르면 메인 페이지로 돌아갑니다.

fill\_page.html

충전을 위한 페이지로 단순히 충전할 금액을 입력 받기 위한 text input과 submit input이 있습니다. 금액을 입력하고 submit input을 누르면 올바르게 입력했을 시 success.html, 실패 시 fail.html을 보여줍니다. cancel버튼을 눌러 취소하고 메인 페이지로 돌아갈 수 있습니다.

success.html

충전, 구매, 물건추가 등의 작업이 성공했을 때 이것이 성공했음을 보여주고 메인 페이지로 되돌아가는 버튼을 통해 메인 페이지로 돌아갑니다.

fail.html

충전, 구매, 물건추가 등의 작업이 실패했을 때 이를 보여주고 tip이라는 변수에 원인을 담아 출력해줍니다. 이 역시 버튼을 눌러 메인 페이지로 돌아가게 합니다.

add\_page.html

물건 추가를 위한 페이지입니다. 현재 추가 가능한 물건들의 카테고리를 표로 보여줍니다. 아래에는 text type의 input들이 존재하고, 이는 추가할 물건의 code, name, price, stock을 입력 받기 위함입니다. 엔터를 누르거나 add버튼을 누르면 이 정보들이 올바를 시 success.html이 출력되며 업데이트가 되고, 아니면 fail.html이 출력되며 업데이트되지 않습니다. cancel버튼을 눌러 메인 페이지로 돌아갈 수 있습니다.

buy\_page.html

구매하는 물건의 정보를 표로 보여줍니다. 본인의 잔고와 등급도 아래에 보여줍니다. 구매할 수량을 입력하고 buy를 누르면 확인을 위한 confirm.html로 넘어갑니다. 이 때 올바르지 않은 수량이 입력되면 fail.html을 출력합니다. cancel버튼을 누르면 메인 페이지로 돌아갑니다.

confirm.html

본인의 잔고와 등급, 구매액과 할인액, 최종금액이 출력됩니다. confirm버튼을 누르면 실제로 구매가 진행되고 cancel을 누르면 취소하고 메인 페이지로 돌아갑니다.