<SQL 테이블 구성>

drop table if exists update\_date;  
create table update\_date(  
     match\_date date primary key  
     );

drop table if exists User;  
create table User(  
     Id varchar(20) primary key,  
     Name varchar(20),  
     Age int,  
     Money dec(10) );  
  
drop table if exists HomeGround;  
create table HomeGround(  
     Number int primary key,  
     Name varchar(20),  
     Location varchar(20) );  
  
drop table if exists Team;  
create table Team(  
     Number int primary key,  
     Name varchar(20),  
     HomeGround\_number int,  
     foreign key(HomeGround\_number) references HomeGround(Number)  
);

drop table if exists Matches;  
create table Matches(  
     Number int primary key,  
     Date date,  
     Home\_Rate float,  
     Away\_Rate float,  
     Home\_Team\_Number int,  
     Away\_Team\_Number int,  
     Recommend\_Team\_Number int,  
  
     foreign key(Home\_Team\_Number) references Team(Number),  
     foreign key(Away\_Team\_Number) references Team(Number),  
     foreign key(Recommend\_Team\_Number) references Team(Number)  
);  
  
drop table if exists MatchHistory;  
create table MatchHistory(  
     Number int primary key,  
     Match\_Number int,  
     Win\_Team\_Number int,  
     Loss\_Team\_Number int,      
      
     foreign key(Match\_Number) references Matches(Number),      
     foreign key(Win\_Team\_Number) references Team(Number),  
     foreign key(Loss\_Team\_Number) references Team(Number)      
);

drop table if exists Bet;  
create table Bet(  
     Match\_Number int,  
     User\_Id varchar(20),  
     User\_Choice\_Team\_Number int,  
     Money Dec(10),

         foreign key(Match\_number) references  Matches(Number),

     foreign key(User\_Id) references User(Id),

     foreign key(User\_Choice\_Team\_Number) references Team(Number),

     primary key(Match\_Number, User\_Id)  
);

<SQL 쿼리문 및 프로시져문 정리>

##날짜별로 경기결과 업데이트(배팅 결과 업데이트)

DROP PROCEDURE IF EXISTS SP\_UPDATE;  
DELIMITER $$  
CREATE PROCEDURE SP\_UPDATE(IN cdate varchar(20))  
     BEGIN  
          DECLARE chk\_count int;  
          set chk\_count = 0;  
      
     select count(\*) into chk\_count from update\_date where match\_date = cdate;  
      
     if chk\_count = 0 then  
     start transaction;  
          insert into update\_date values (cdate);  
          update user as A inner join  
( select user.id, user.money as tot\_money, ttt.away\_rate, ttt.money from  
(select tt.match\_number, tt.user\_id,tt.money, tt.away\_team\_number, tt.away\_rate  from (( select t.match\_number, t.user\_id, t.money, m.away\_team\_number, m.away\_rate from  
( select b1.match\_number,b1.user\_id, b1.money, b1.user\_choice\_team\_number,d1.win\_team\_number from  
(select m1.date, m1.number, mh1.win\_team\_number from (select \* from matches where date = cdate) as m1  
left join matchhistory as mh1 on m1.number = mh1.number ) as d1  
left join bet as b1 on b1.match\_number = d1.number where user\_choice\_team\_number = win\_team\_number ) as t  
left join matches as m on m.number = t.match\_number and t.win\_team\_number=m.away\_team\_number ) as tt ) where tt.away\_rate is not  NULL) as ttt  
left join user on user.id = ttt.user\_id) as B on  A.id =B.id set A.money = A.money + B.money\*B.away\_rate;  
  
  
update user as A inner join  
( select user.id, user.money as tot\_money, ttt.home\_rate, ttt.money from  
(select tt.match\_number, tt.user\_id,tt.money, tt.home\_team\_number, tt.home\_rate  from (( select t.match\_number, t.user\_id, t.money, m.home\_team\_number, m.home\_rate from  
( select b1.match\_number,b1.user\_id, b1.money, b1.user\_choice\_team\_number,d1.win\_team\_number from  
(select m1.date, m1.number, mh1.win\_team\_number from (select \* from matches where date = cdate) as m1  
left join matchhistory as mh1 on m1.number = mh1.number ) as d1  
left join bet as b1 on b1.match\_number = d1.number where user\_choice\_team\_number = win\_team\_number ) as t  
left join matches as m on m.number = t.match\_number and t.win\_team\_number=m.home\_team\_number ) as tt ) where tt.home\_rate is not  NULL) as ttt  
left join user on user.id = ttt.user\_id) as B on  A.id =B.id set A.money = A.money + B.money\*B.home\_rate;  
     commit;  
     end if;  
     select chk\_count;  
     END $$  
DELIMITER ;

설명 :

해당 날짜의 경기를 찾고  
matchhistory에서 해당경기의 이긴 팀을 찾음  
date number win\_team\_number 가 나오면  
경기번호가 같고, 승리팀을 맞춘 유저의정보를 뽑음  
match\_number, user\_id,money-user\_choice\_team\_number, win\_team\_number 뽑음  
승리팀이 어웨이 팀(홈팀)인 경우 나눠서  
match\_number, user\_id, money, away\_team\_number, away\_rate를 뽑아옴  
다시 조인으로 업데이트 해야할 user의  
id tot\_money,away\_rate, 배팅한money 불러옴  
그리고 업데이트

<사용자가 배팅을 하는 기능>

DROP PROCEDURE IF EXISTS SP\_BET;  
DELIMITER $$  
CREATE PROCEDURE SP\_BET(IN mnumber int, IN uid varchar(20), IN cteam varchar(20), IN bmoney dec(10) )  
     BEGIN  
          DECLARE chkmoney dec(10);  
            DECLARE team\_name varchar(20);  
            DECLARE team\_number int;  
            DECLARE avail int;  
            DECLARE chk\_count int;  
             
            set chk\_count = 0;  
            set avail = 0;  
            set team\_name = cteam;  
             
          start transaction;  
           
            update user set money = money - bmoney where id = uid;               
          select money into chkmoney from user where id = uid;  
            select number into team\_number from team where name = cteam;  
             
            select count(\*) into chk\_count from bet where match\_number=mnumber and user\_id=uid;  
           
            if  chk\_count = 0 then  
               insert into Bet(Match\_number, user\_id, user\_choice\_Team\_Number, Money) values                          (mnumber,uid, team\_number, bmoney);  
            end if;  
          
          if chkmoney >= 0 and chk\_count = 0 then  
                    set avail = 1;  
               commit;  
          elseif chk\_count > 0 then  
                    set avail = 2;  
               rollback;  
            elseif chkmoney < 0 then  
                    set avail = 3;  
                    rollback;  
          end if;  
            select team\_number,avail;  
     end $$  
delimiter ;

<회원가입>

DROP PROCEDURE IF EXISTS SP\_JOIN;

DELIMITER $$

CREATE PROCEDURE SP\_JOIN( IN inputid varchar(20), IN inputname varchar(20), IN inputage int, IN inputpwd varchar(20) )

     BEGIN

          DECLARE count int;

          select count(\*) into count from user where id = inputid;

          IF count = 0 THEN

                insert into User(Id, Name, Age, Passwd) values (inputid, inputname, inputage, md5(inputpwd));

          END IF;

     END $$

DELIMITER ;

<로그인>

DROP PROCEDURE IF EXISTS SP\_LOGIN;

DELIMITER $$  
CREATE PROCEDURE SP\_LOGIN(IN inputid varchar(20), inputpasswd char(32) )  
     BEGIN  
          DECLARE chk int;  
          DECLARE user\_name varchar(20);      
              select count(\*),user\_name into chk,user\_name from user where id = inputid and passwd = md5(inputpasswd);  
           
          select chk,user\_name;  
end $$  
delimiter ;

<현재 세션의 사용자 정보를 보여줌>

DROP PROCEDURE IF EXISTS SP\_MY\_INFO;  
DELIMITER $$  
CREATE PROCEDURE SP\_MY\_INFO( IN inputid varchar(20) )  
     BEGIN  
     DECLARE user\_name VARCHAR(20);  
     DECLARE user\_age int;  
     DECLARE user\_money int;  
     DECLARE chk int;  
  
     select count(\*) into chk from user where id = inputid;  
     if chk = 1 then  
          select name,age,money into user\_name,user\_age,user\_money from user where id = inputid;  
     end if;  
     select chk,user\_name,user\_age,user\_money;  
     END $$  
DELIMITER ;

<특정 경기에 대한 추천팀 선정>

DROP PROCEDURE IF EXISTS SP\_RECOMMEND;  
DELIMITER $$  
CREATE PROCEDURE SP\_RECOMMEND(IN matchnumber INT)  
     BEGIN  
          DECLARE home\_win\_count INT;          
          DECLARE away\_win\_count INT;  
          DECLARE home\_tot\_win\_count INT;          
          DECLARE away\_tot\_win\_count INT;           
          DECLARE HOME INT;  
          DECLARE AWAY INT;  
          DECLARE recommend\_team\_name varchar(20);  
          DECLARE win\_count INT;  
          DECLARE loss\_count INT;  
          DECLARE tot\_win\_count INT;  
          DECLARE tot\_loss\_count INT;  
          declare team\_number int;  
          select home\_team\_number into HOME from matches where number = matchnumber;  
          select away\_team\_number into AWAY from matches where number = matchnumber;  
  
          select count(\*) into home\_win\_count from matchhistory where win\_team\_number = HOME and loss\_team\_number = AWAY;  
          select count(\*) into away\_win\_count from matchhistory where win\_team\_number = AWAY and loss\_team\_number = HOME;  
  
          select count(\*) into home\_tot\_win\_count from matchhistory where win\_team\_number = HOME;  
          select count(\*) into away\_tot\_win\_count from matchhistory where win\_team\_number = AWAY;  
          
          IF home\_win\_count > away\_win\_count then  
            select name into recommend\_team\_name from team where number = HOME;  
            set win\_count = home\_win\_count;  
            set loss\_count = away\_win\_count;

           set tot\_win\_count = home\_tot\_win\_count;

           select count(\*) into tot\_loss\_count from matchhistory where loss\_team\_number = HOME;  
            SELECT RECOMMEND\_TEAM\_number into team\_number from matches where number = matchnumber;  
            IF team\_number IS NULL then  
              update matches set recommend\_team\_number = HOME where number = matchnumber;  
         end if;  
       ELSEIF home\_win\_count < away\_win\_count then  
               select name into recommend\_team\_name from team where number = AWAY;  
               set win\_count = away\_win\_count;  
               set loss\_count = home\_win\_count;

               set tot\_win\_count = away\_tot\_win\_count;

               select count(\*) into tot\_loss\_count from matchhistory where loss\_team\_number = AWAY;

          SELECT RECOMMEND\_TEAM\_number into team\_number from matches where number = matchnumber;  
          IF team\_number IS NULL then      
                    update matches set recommend\_team\_number = AWAY where number = matchnumber ;      
          end if;  
        ELSE     
               IF home\_tot\_win\_count > away\_tot\_win\_count then  
                   select name into recommend\_team\_name from team where number = HOME;      
                    SELECT RECOMMEND\_TEAM\_number into team\_number from matches where number = matchnumber;  
              if team\_number IS NULL then  
               update matches set recommend\_team\_number = HOME where number = matchnumber;  
                 end if;  
                    set tot\_win\_count = home\_tot\_win\_count;

                    select count(\*) into tot\_loss\_count from matchhistory where loss\_team\_number = HOME;

                    set win\_count = home\_win\_count;

                    set loss\_count = away\_win\_count;

               else  
                    select name into recommend\_team\_name from team where number = AWAY;      
              SELECT RECOMMEND\_TEAM\_number into team\_number from matches where number = matchnumber;  
              if team\_number IS NULL then  
               update matches set recommend\_team\_number = AWAY where number = matchnumber;  
                 end if;  
                    set tot\_win\_count = away\_tot\_win\_count;

               select count(\*) into tot\_loss\_count from matchhistory where loss\_team\_number = HOME;

                set win\_count = home\_win\_count;

                    set loss\_count = away\_win\_count;

               end if;  
       END IF;  
          select win\_count, loss\_count, tot\_win\_count, tot\_loss\_count,RECOMMEND\_TEAM\_name;         
          
     end $$  
delimiter ;

DROP PROCEDURE IF EXISTS SP\_SHOW\_TEAM;  
DELIMITER $$  
CREATE PROCEDURESP\_SHOW\_TEAM( IN inputid varchar(20) )

     BEGIN  
     DECLARE user\_name VARCHAR(20);  
     DECLARE user\_age int;  
     DECLARE user\_money int;  
     DECLARE chk int;  
  
     select count(\*) into chk from user where id = inputid;  
     if chk = 1 then  
          select name,age,money into user\_name,user\_age,user\_money from user where id = inputid;  
     end if;  
     select chk,user\_name,user\_age,user\_money;  
     END $$  
DELIMITER ;

<특정 팀의 정보 보기>

select t.name as team\_name, g.name as ground\_name, g.location as location from (select \* from team where name

= '특정팀') as t

inner join homeground as g on t.homeground\_number = g.number;

<오늘의 경기 정보 보기>

( select M1.number, M1.date,t1.name As Home,t2.name As Away, M1.home\_rate, M1.away\_rate from (select number, date, Home\_Team\_Number, Away\_Team\_Number, home\_rate,away\_rate from matches where date = '오늘 날짜') As M1 inner join team as t1 on M1.Home\_Team\_Number = t1.number inner join team as t2 on M1.Away\_Team\_Number = t2.number) order by M1.number"

<성능 개선 사항>

matches  테이블은 date로 index

bet 테이블은 win\_team\_number 로 index

user 테이블은 id로 index

date 테이블은 match\_date로 index(바로 primarykey 설정)

쿼리문의 효율성 늘림

잘게 쪼갤 수 있는 쿼리문은 잘게 쪼개서 성능을 높임

<dummy data 파일 생성 프로시져>

DROP PROCEDURE IF EXISTS SP\_DUMMY2;

DELIMITER $$

CREATE PROCEDURE SP\_DUMMY2()

BEGIN

declare k int;

set k = 1;

WHILE k < 10000 DO

INSERT INTO matches VALUES(k, date\_sub('2014-06-22', interval k day), 2, 2,floor(rand()\*100), floor(rand()\*900) ,NULL);

INSERT INTO matches VALUES(k+1, date\_sub('2014-06-22', interval k day), 2, 2,floor(rand()\*100), floor(rand()\*900) ,NULL);

INSERT INTO matches VALUES(k+2, date\_sub('2014-06-22', interval k day), 2, 2,floor(rand()\*100), floor(rand()\*900) ,NULL);

INSERT INTO matches VALUES(k+3, date\_sub('2014-06-22', interval k day), 2, 2,floor(rand()\*100), floor(rand()\*900) ,NULL);

SET k = k + 4;

end while;

END $$

DELIMITER ;

DROP PROCEDURE IF EXISTS SP\_DUMMY;

DELIMITER $$

CREATE PROCEDURE SP\_DUMMY()

BEGIN

DECLARE i int;

SET i = 1;

WHILE i < 10000 DO

INSERT INTO team VALUES(i, CONCAT('test-team#',i), 1);

SET i = i + 1;

END WHILE;

END $$

DELIMITER ;

DROP PROCEDURE IF EXISTS SP\_DUMMY3;

DELIMITER $$

CREATE PROCEDURE SP\_DUMMY3()

BEGIN

declare k int;

set k = 1;

WHILE k < 10000 DO

INSERT INTO user VALUES(k, concat('name#',k), k, 10000,md5('1234') );

SET k = k + 1;

end while;

END $$

DELIMITER ;

DROP PROCEDURE IF EXISTS SP\_DUMMY4;

DELIMITER $$

CREATE PROCEDURE SP\_DUMMY4()

BEGIN

declare k int;

declare home int;

declare away int;

set k = 1;

WHILE k < 10000 DO

select home\_team\_number, away\_team\_number into home,away from matches where number = k;

if (k mod 2) = 1 then

INSERT INTO matchhistory VALUES(k, k, home, away,NULL );

else

INSERT INTO matchhistory VALUES(k, k, away, home,NULL );

end if;

SET k = k + 1;

end while;

END $$

DELIMITER ;