

## SQL programming HW2

Xiaolin Li (2791913)

### 1. Enforce constraint

```
ALTER TABLE made_money
```

```
Modify Category SET DEFAULT 'Action'
```

```
ALTER TABLE log_data
```

```
Modify Category SET DEFAULT 'Action'
```

```
ALTER TABLE made_money
```

```
add Constraint cat1 check (category = 'Romantic' or category = 'Comedy' or category = 'Drama'  
or category = 'Action')
```

```
ALTER TABLE log_data
```

```
add Constraint cat2 check (category = 'Romantic' or category = 'Comedy' or category = 'Drama'  
or category = 'Action')
```

```
delimiter //
```

```
CREATE TRIGGER check_made_money_insert
```

```
BEFORE INSERT ON made_money
```

```
FOR each row
```

```
Begin
```

```
    if NEW.category not in ('Romantic', 'Comedy', 'Drama', 'Action') Then
```

```
        set new.category = 'Action';
```

```
    end if;
```

```
end;
```

```
//
```

```
delimiter //
```

```
CREATE TRIGGER check_made_money_update
```

```
BEFORE UPDATE ON made_money
```

```
FOR each row
```

```
Begin
```

```
    if NEW.category not in ('Romantic', 'Comedy', 'Drama', 'Action') Then
```

```
        set new.category = 'Action';
```

```
    end if;
```

```
end;
```

```
//
```

```
delimiter //
```

```
CREATE TRIGGER check_log_data_insert
```

```
BEFORE INSERT ON log_data
```

```
FOR each row
```

```
Begin
```

```
    if NEW.category not in ('Romantic', 'Comedy', 'Drama', 'Action') Then
```

```
        set new.category = 'Action';
```

```
    end if;
```

```
end;
```

```
//
```

```
delimiter //
```

```
CREATE TRIGGER check_log_data_update
```

```
BEFORE update ON log_data
```

FOR each row

Begin

if NEW.category not in ('Romantic', 'Comedy', 'Drama', 'Action') Then

set new.category = 'Action';

end if;

end;

//

delimiter ;

INSERT INTO made\_money VALUE('IRON MAN','1000000','2008-05-02','Action');

#select \* from made\_money where how\_much = '1000000';

UPDATE made\_money SET made\_money.category = 'Horror' where made\_money.movie = 'Fight Club';

select \* from made\_money;

drop trigger check\_made\_money\_insert;

drop trigger check\_made\_money\_update;

drop trigger check\_log\_data\_insert;

drop trigger check\_log\_data\_update;

## **2. Check if a star can be a part of a Comedy**

delimiter //

create trigger check\_comedy

before insert on appeared\_in

for each row

```

begin
    if exists
        (select * from appeared_in a1, made_money m1, made_money m2
         where new.movie = m1.movie
         and a1.movie = m2.movie
         and new.star = a1.star
         and new.movie <> a1.movie
         and m2.category not in ('Drama', 'Romantic', 'Comedy')
         and m2.day_opened < m1.day_opened
        ) then
        update made_money mm
        set mm.category = 'Drama'
        where mm.movie = new.movie;
    end if;
end;

```

delimiter ;

```

insert into appeared_in value ('Matt Damon', 'Bruce Almighty');

select * from appeared_in, made_money where star = 'Matt Damon' and made_money.movie =
appeared_in.movie;

drop trigger check_comedy

```

### **3. Cannot be married to multiple stars simultaneously**

```

#delete from married where couple_num=1;

#drop trigger c3;

delimiter //

```

```

create trigger c3
before insert on married
for each row
begin
    if exists ( select * from in_couple i1, in_couple i2, married m1
    where i1.couple_num = new.couple_num
    and i1.star = i2.star and i1.couple_num <> i2.couple_num #in other couple number
    and m1.couple_num = i2.couple_num
    and m1.day < new.day
    and (exists (select * from divorced d1 where d1.couple_num = i2.couple_num and
d1.day>new.day)
    or (not exists(select * from divorced d2 where d2.couple_num = i2.couple_num)))
    ) then
    signal sqlstate '45001'
    set message_text = 'Married check failed';
    end if;
end;
//

delimiter ;
insert into married value (1,'2015-06-26');
select * from married;

```

```
drop trigger c3;
```

#### **4. Check box office constraint**

```

delimiter //
create trigger check_total_box

```

```

before insert on made_money
for each row
begin
    if (new.how_much <1000) then
        signal sqlstate '45001'
        set message_text = 'Error: box office less than 1,000';
    end if;

    if (new.how_much > 3000000000) then
        signal sqlstate '45002'
        set message_text = 'Error: box office is too much';
    end if;

    if (new.how_much < 10000 and new.category = 'Action') then
        signal sqlstate '45003'
        set message_text = 'Error: An action movie has to have at least $10000 office box';
    end if;

    if (new.how_much > 1000000000 and new.category = 'Comedy') then
        signal sqlstate '45004'
        set message_text = 'Error: A comedy movie''s box office could not exceed 1e9';
    end if;
end;
//

delimiter ;

```

```
insert into made_money values('Most Welcome', 8000, '2012-07-07', 'Action');
insert into made_money values('Speed', 9000, '2012-03-28', 'Comedy');
insert into made_money values('Hangover', 1500000000, '2011-03-05', 'Comedy');
```

```
drop trigger check_total_box
```

## **5. Check divorce date**

```
delimiter //
create trigger check_divorce
before insert on divorced
for each row
begin
    if exists (select * from married m1
               where m1.couple_num = new.couple_num and new.day <
m1.day) then
        set new.day = (select m2.day from married m2
                       where m2.couple_num = new.couple_num);
    end if;
end;
//
```

```
delimiter ;
```

```
insert into divorced value(6, '2004-01-01');
drop trigger check_divorce;
select * from divorced
```

**6. To evaluate the final state of log\_data table, we should put all the triggers together. Note that the version of MySQL I used does not support multiple triggers with the same action on the same table, I have to merge the triggers in Q4 into Q1. The final SQL script is (The script for Q6 is the last trigger):**

```
ALTER TABLE made_money
```

```
Modify Category SET DEFAULT 'Action'
```

```
ALTER TABLE log_data
```

```
Modify Category SET DEFAULT 'Action'
```

```
ALTER TABLE made_money
```

```
add Constraint cat1 check (category = 'Romantic' or category = 'Comedy' or category = 'Drama'  
or category = 'Action')
```

```
ALTER TABLE log_data
```

```
add Constraint cat2 check (category = 'Romantic' or category = 'Comedy' or category = 'Drama'  
or category = 'Action')
```

```
delimiter //
```

```
CREATE TRIGGER check_log_data_insert
```

```
BEFORE INSERT ON log_data
```

```
FOR each row
```

```
Begin
```

```
    if NEW.category not in ('Romantic', 'Comedy', 'Drama', 'Action') Then
```

```
        set new.category = 'Action';
```

```
    end if;
```

```
end;
```

```
//
```

```
delimiter //
```



```

CREATE TRIGGER check_log_data_update
BEFORE update ON log_data
FOR each row
Begin
    if NEW.category not in ('Romantic', 'Comedy', 'Drama', 'Action') Then
        set new.category = 'Action';
    end if;
end;
//

```

delimiter //

```

CREATE TRIGGER check_made_money_insert
BEFORE INSERT ON made_money
FOR each row
Begin
    if NEW.category not in ('Romantic', 'Comedy', 'Drama', 'Action') Then
        set new.category = 'Action';
    end if;

    if (new.how_much < 1000) then
        signal sqlstate '45001'
        set message_text = 'Error: box office less than 1,000';
    end if;

    if (new.how_much > 3000000000) then
        signal sqlstate '45002'
        set message_text = 'Error: box office is too much';
    end if;
end;

```

```

        end if;

if (new.how_much < 10000 and new.category = 'Action') then
    signal sqlstate '45003'
    set message_text = 'Error: An action movie has to have at least $10000 office box';
    end if;

if (new.how_much > 1000000000 and new.category = 'Comedy') then
    signal sqlstate '45004'
    set message_text = 'Error: A comedy movie's box office could not exceed 1e9';
    end if;
end;
//

delimiter //
CREATE TRIGGER check_made_money_update
BEFORE UPDATE ON made_money
FOR each row
Begin
    if NEW.category not in ('Romantic', 'Comedy', 'Drama', 'Action') Then
        set new.category = 'Action';
    end if;
end;
//

delimiter //

```

```
create trigger check_comedy
before insert on appeared_in
for each row
begin
    if exists
        (select * from appeared_in a1, made_money m1, made_money m2
         where new.movie = m1.movie
         and a1.movie = m2.movie
         and new.star = a1.star
         and new.movie <> a1.movie
         and m2.category not in ('Drama', 'Romantic', 'Comedy')
         and m2.day_opened < m1.day_opened
        ) then
        update made_money mm
        set mm.category = 'Drama'
        where mm.movie = new.movie;
    end if;
end;
```

```
delimiter ;
```

```
delimiter //
```

```
create trigger c3
before insert on married
for each row
begin
    if exists ( select * from in_couple i1, in_couple i2, married m1
```

```

where i1.couple_num = new.couple_num
and i1.star = i2.star and i1.couple_num <> i2.couple_num #in other couple number
and m1.couple_num = i2.couple_num
and m1.day < new.day

and (exists (select * from divorced d1 where d1.couple_num = i2.couple_num and
d1.day>new.day)

or (not exists(select * from divorced d2 where d2.couple_num = i2.couple_num)))
) then
signal sqlstate '45001'
set message_text = 'Married check failed';
end if;
end;
//

delimiter ;

delimiter //
create trigger check_divorce
before insert on divorced
for each row
begin
    if exists (select * from married m1
                where m1.couple_num = new.couple_num and new.day <
m1.day) then
        set new.day = (select m2.day from married m2
                        where m2.couple_num = new.couple_num);
    end if;
end;

```

```
//
```

```
# The part for question 6 =====
```

```
delimiter ;
```

```
delimiter //
```

```
create trigger log_track
```

```
after insert on made_money
```

```
for each row
```

```
begin
```

```
    insert into log_data values(new.movie, new.category);
```

```
end;
```

```
//
```

```
delimiter ;
```

```
#=====
```

```
INSERT INTO made_money VALUE('IRON MAN','1000000','2008-05-02','Action');
```

```
UPDATE made_money SET made_money.category = 'Horror' where made_money.movie =  
'Fight Club';
```

```
insert into appeared_in value ('Matt Damon', 'Bruce Almighty');
```

```
#insert into married value (1,'2015-06-26');
```

```
#insert into made_money values('Most Welcome', 8000, '2012-07-07', 'Action');
```

```
insert into made_money values('Speed', 9000, '2012-03-28', 'Comedy');
```

```
#insert into made_money values('Hangover', 1500000000, '2011-03-05', 'Comedy');
```

```
insert into divorced value(6, '2004-01-01');
```

```
select * from log_data
```

**The final state of log\_data is:**

# Movie, Category

'IRON MAN', 'Action'

'Speed', 'Comedy'