

ICSI311: Өгөгдлийн сангийн систем

Г. Саруул (20B1NUM2095)

Лабораторийн ажил №4: Trigger

Сонгон авсан өгөгдлийн сан дээрээ update, delete, insert гурван төрлийн триггерүүд бичнэ. Нэмэлт нэг триггер бичнэ тэр нь хоёр хүснэгт дамнасан байх шаардлагатай. Үндсэндээ 2 дэд триггэр бичиж байж хоорондоо холбогдоно.

Сонгон авсан өгөгдлийн сан:

Ашиглах гэж байгаа триггерүүдээ тохируулан жишээ өгөгдлийн сан үүсгэв.

```
CREATE DATABASE DVDStore ;

USE DVDStore ;

CREATE TABLE Studio (
    studio_id int unsigned not null,
    studio_name varchar(32) not null,
    description varchar(256) not null,
    primary key (studio_id)
) ;

CREATE TABLE game_orders (
    order_id int unsigned not null,
    user_id int unsigned not null,
    product_id int unsigned not null,
    quantity int unsigned not null,
    Subtotal int unsigned not null
);

CREATE TABLE games (
    product_id int unsigned not null,
    name varchar(24) not null,
    brand_id int unsigned not null,
    price int unsigned not null,
    category varchar(64) not null,
    stock int unsigned not null,
    saruul_bonus int unsigned not null
) ;

CREATE TABLE users (
    user_id int unsigned not null,
```

```
name varchar(24) not null,  
email varchar (64) not null,  
phone int unsigned not null,  
password varchar(128) not null,  
saruul_bonus int unsigned not null,  
primary key (user_id)  
) ;
```

Үүсгэсэн өгөгдлийн сан дээрээ жишээ өгөгдлүүд оруулав:

```
INSERT INTO Studio VALUES ('1001', 'Valve', 'created counter strike game on  
1999') ;  
INSERT INTO Studio VALUES ('1002', 'Riot Games', 'created valorant on  
2017') ;  
  
INSERT INTO game_orders VALUES ('2001', '3001', '4001', 2, 103800) ;  
INSERT INTO game_orders VALUES ('2002', '3002', '4003', 1, 50000) ;  
INSERT INTO game_orders VALUES ('2003', '3003', '4004', 3, 150000) ;  
  
INSERT INTO game_orders VALUES ('2001', '3001', '4001', 2, 103800) ;  
INSERT INTO game_orders VALUES ('2002', '3002', '4003', 1, 50000) ;  
INSERT INTO game_orders VALUES ('2003', '3003', '4004', 3, 150000) ;  
  
INSERT INTO users VALUES ('001', 'Saruul', 'saruul@admin.com', '90330000',  
'DatabaseSystem123', 500) ;  
INSERT INTO users VALUES ('002', 'Undrakh', 'undrakh@admin.com',  
'91102091', 'DatabaseSystem321', 300) ;
```

Insert Trigger:

Хэрвээ бүтээгдэхүүн захиалганд орсон бол тухайн бүтээгдэхүүн дээр байх мэдээлэл өөрчлөгдөх буюу stock - бэлэн байгаа нөөцийн тоо ширхгийг нь өөрчилж багасгадаг триггер. Ингэснээр хэрвээ бүтээгдэхүүний нөөц 0 болсон бол дахин тэр бүтээгдэхүүнээс захиалга авахгүй байх, эсвэл бүтээгдэхүүний татан авалтыг даруй хийх зэргийг мэдэх боломжтой болно.

```
DELIMITER &&  
CREATE TRIGGER after_orders_added  
AFTER INSERT ON game_orders  
FOR EACH ROW
```

```
BEGIN
    UPDATE games SET stock = stock - 1
    WHERE product_id = new.product_id and stock != 0 ;
END &&
```

#	product_id	name	brand_id	price	category	stock	saruul_bonu
	4003	Dota2	1001	5000	action	999	300
	4001	Counter Strike	1001	19500	action	999	100
3	4004	Valorant	1002	19500	action	999	100

Бүтээгдэхүүний захиалга нэмэгдсэн бөгөөд анх 999 нөөцтэй байж байгаад триггер ажилсны дараа нөөц нь нэгээр хорогдсон байна.

```
INSERT INTO game_orders VALUES ( '2004', '001', '4001', 1, 150000 ) ;
```

#	product_id	name	brand_id	price	category	stock	saruul_bonu
	4001	Counter Strike	1001	19500	action	998	100
	4003	Dota2	1001	5000	action	999	300
	4004	Valorant	1002	19500	action	999	100

Delete Trigger:

```
DELIMITER &&
CREATE TRIGGER after_game_deleted
AFTER DELETE ON Studio
FOR EACH ROW
BEGIN
    DELETE FROM Studio WHERE studio_id = old.studio_id ;
    INSERT INTO removed_studio VALUE (old.studio_id, old.studio_name) ;
END &&
```

DVDStore ямар нэг студитэй хамтран ажиллах гэрээгээ цуцлах эсвэл цаашид тухайн студийн тоглоомыг худалдаалахгүй болвол манай дэлгүүр бүтээгдэхүүнийг худалдаж буй бүтээгдэхүүнээсээ хасах ёстой. Энэ триггер нь хэрвээ тоглоомын мэдээлэл уствал тухайн тоглоомыг үндсэн хүснэгтээс устгаж removed_studio буюу хамтарч ажиллахаа больсон тоглоомын студийн мэдээллийг хадгалах хүснэгтэд нэмж оруулна.

#	studio_id	studio_name	description
	1001	Valve	created counter strike game on 1999
	1002	Riot Games	created valorant on 2017
	NULL	NULL	NULL

Триггер ажилсны дараа:

Studio хүснэгт:

studio_id	studio_name	description
1001	Valve	created counter strike game on 1999
NULL	NULL	NULL

Removed_studio хүснэгт:

#	studio_id	studio_name
	1002	Riot Games

Update Trigger:

```
DELIMITER &&
CREATE TRIGGER after_product_changed
BEFORE UPDATE
ON games
FOR EACH ROW
BEGIN
    if (old.price <> new.price)
    then
        set new.saruul_bonus = new.price / 100 ;
    end if ;
END &&
```

Тоглоомын үнэ нэмэгдвэл худалдан авалтанд дагалдах saruul_bonus оноо нь мөн адил зохих хэмжээгээр нэмэгдэх ёстой.

UPDATE үйлдэл:

```
UPDATE games SET price = 400000 where product_id = '4004' ;
```

Триггер ажиллахаас өмнө:

#	product_id	name	brand_id	price	category	stock	saruul_bonu
	4001	Counter Strike	1001	19500	action	998	100
	4003	Dota2	1001	5000	action	999	300
	4004	Valorant	1002	19500	action	999	100

Триггер ажилласны дараа:

#	product_id	name	brand_id	price	category	stock	saruul_bonu
	4001	Counter Strike	1001	19500	action	998	100
	4003	Dota2	1001	5000	action	999	300
	4004	Valorant	1002	400000	action	999	4000

4004 дугаартай тоглоомын үнийг өөрчлөхөд оноо нь давхар өөрчлөгдөж байна.

Дамнасан триггер:

```
DELIMITER &&
CREATE TRIGGER after_orders_changed
AFTER UPDATE
ON game_orders
FOR EACH ROW
BEGIN
    INSERT INTO extra_order
    VALUES (old.order_id, old.user_id, old.product_id, new.Quantity -
old.Quantity, new.SubTotal - old.subTotal) ;
END &&
```

```
DELIMITER &&
CREATE TRIGGER after_added_extra_order
AFTER INSERT
ON extra_order
FOR EACH ROW
BEGIN
    UPDATE users JOIN game_orders ON game_orders.user_id = users.user_id
join games on games.product_id = game_orders.product_id
SET users.saruul_bonus = users.saruul_bonus + (new.quantity *
games.saruul_bonus) ;
END &&
```

Үйлчлүүлэгч захиалсан тоглоом дээрээ нэмж тоглоом захиалж болно, захиалгын мэдээлэл өөрчлөгдөнө. Тэгвэл энэ мэдээллийг extra_order буюу нэмэлт захиалгын хүснэгтэд хадгална. Харин нэмэлт захиалга хүснэгтэд мэдээлэл нэмэгдвэл тухайн захиалгыг хийсэн хэрэглэгчийн saruul_bonus оноог нэмдэг байх гэсэн энэ ажил дээр триггерийг бичиж хэрэгжүүлсэн.

```
UPDATE game_orders JOIN games ON game_orders.product_id = games.product_id
SET game_orders.quantity = 5, game_orders.Subtotal = (5 * games.price)
WHERE game_orders.order_id = '2001' ;
```

Хүснэгт дээр өөрчлөлт орохоос өмнө:

Game_orders хүснэгт

order_id	user_id	product_id	quantity	Subtotal
2001	3001	4001	2	103800
2002	3002	4003	1	50000
2003	3003	4004	3	150000
2004	1	4001	3	150000

Users хүснэгт

user_id	name	email	phone	password	saruul_bonus
1	Saruul	saruul@admin.com	90330000	DatabaseSystem123	500
2	Undrakh	undrakh@admin.com	91102091	DatabaseSystem321	300
NULL	NULL	NULL	NULL	NULL	NULL

Хүснэгт дээр өөрчлөлт орсны дараа:

Game_orders хүснэгт

order_id	user_id	product_id	quantity	Subtotal
2001	3001	4001	5	97500
2002	3002	4003	1	50000
2003	3003	4004	3	150000
2004	1	4001	3	150000

Users хүснэгт

user_id	name	email	phone	password	saruul_bonu
1	Saruul	saruul@admin.com	90330000	DatabaseSystem123	600
2	Undrakh	undrakh@admin.com	91102091	DatabaseSystem321	300
NULL	NULL	NULL	NULL	NULL	NULL

Дүгнэлт

Триггерийг зөв зохион байгуулж чадвал хийх ажлаа олон дахин хөнгөвчилж, хялбаршуулах боломжтой. Мөн энэхүү лабораторийн ажлаар триггерийн талаар илүү ихийг мэдэж, өгөгдлийн сан зохион байгуулах мэдлэгээ нэмэгдүүлж чадлаа.