

16 트리거



01 트리거

트리거 기본

- 테이블 or 뷰에 부착되는 프로그램 코드
- 테이블에 특정 명령어 수행시(입력/수정/삭제) 자동으로 실행되는 코드

종류	설명
BEFORE 트리거	이벤트 발생 이전에 작동되는 트리거 INSERT, UPDATE, DELETE 세가지 이벤트로 작동
AFTER 트리거	INSERT,UPDATE,DELETE 등의 작업이 일어났을 때 작동하는 트리거

Part 2.

AFTER 트리거



01 AFTER트리거

AFTER 트리거

- 회원 테이블에 update 나 delete 를 시도하면 수정 또는 삭제된 데이터를
- 별도의 테이블에 보관하고 변경된 일자와 변경한 사람을 기록

트리거 대상 테이블(usertbl2) 복사

```
use sqldb;  
drop table if exists usertbl2;  
create table usertbl2(select * from usertbl);
```

01 AFTER트리거

백업 table 생성

```
create table backup_usertbl
(
userid char(8) not null primary key,
name varchar(10) not null,
birthYear int not null,
addr char(2) not null,
mobile1 char(3),
mobile2 char(8),
height smallint,
mdate date,
modtype char(2),-- 변경된 타입 , 수정 or 삭제
moddate date,      -- 변경된 날짜
moduser varchar(256)-- 변경한 사용자
);
```

01 AFTER트리거

수정 트리거 지정

```
-- 수정 트리거
drop trigger if exists backuserTbl_UpdateTrg;
DELIMITER //
create trigger backUserTbl_UpdateTrg
    after update
    on usertbl2
    for each row
begin
    insert into backup_usertbl values(old.userid,old.name,old.birthyear,
    old.addr,old.mobile1,old.mobile2, old.height, old.mdate, '수정',curdate(),current_user());
end //
DELIMITER ;
```

01 AFTER트리거

삭제 트리거 지정

```
-- 삭제 트리거
drop trigger if exists backUserTbl_DeleteTrg;
delimiter //
create trigger backUserTbl_deleteTrg
after delete
on usertbl2
for each row
begin
insert into backup_usertbl values(old.userid,old.name,old.birthyear,
old.addr,old.mobile1,old.mobile2,old.height,old.mdate,'삭제',curdate(),current_user());
end //
delimiter ;
```

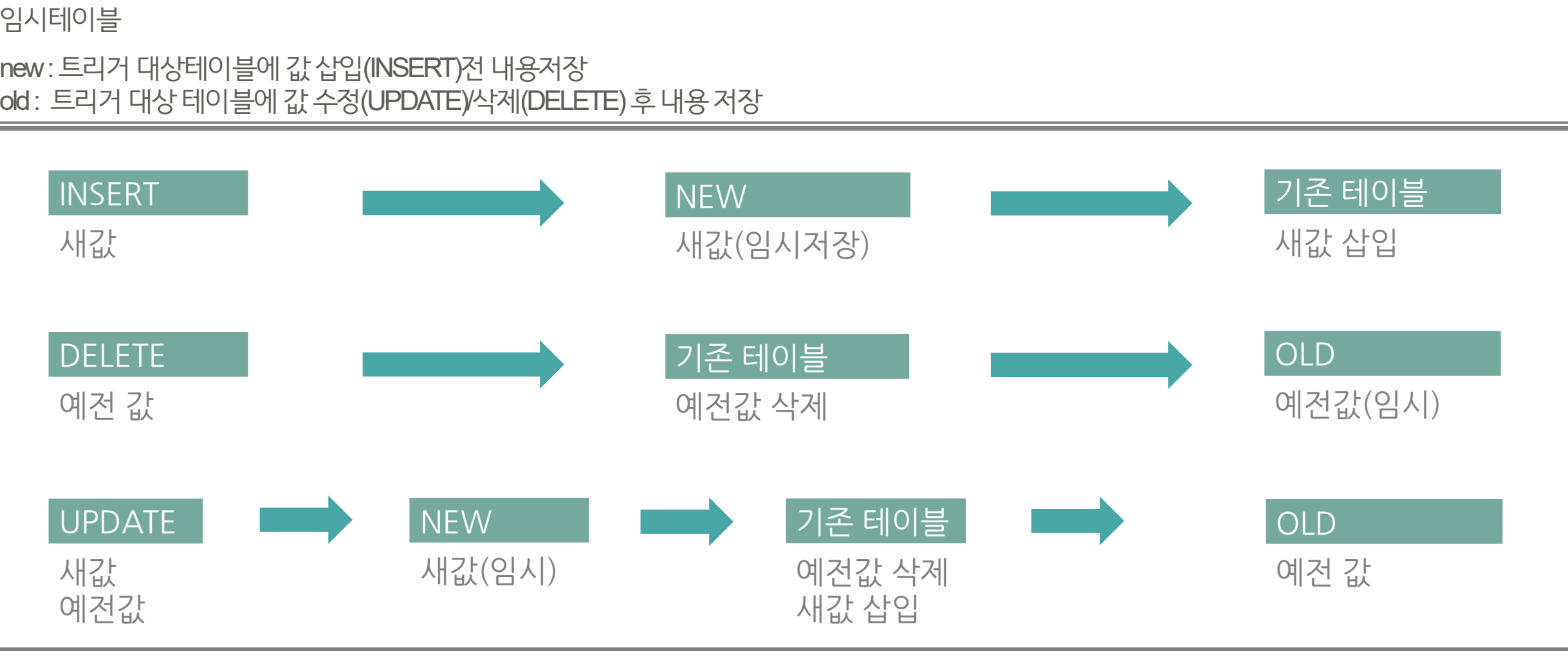
01 AFTER트리거

확인

```
update usertbl2 set addr='몽고' where userid='JKW';  
delete from usertbl2 where height>=177;
```

```
select * from backup_usertbl;
```


01 AFTER트리거



01 문제

오늘 생산제품(makeprodtbl)

CODE	prodName	amount
111	세탁기	10
222	냉장고	5
333	노트북	10

재고제품(saveprodtbl)

CODE	prodName	amount
111	세탁기	50
222	냉장고	10
333	노트북	30

1 생산제품 테이블에 값 추가(INSERT) 시 재고 제품에도 추가될수 있도록 트리거 설정하세요

2 생산제품 테이블에 기존 제품의 생산량(amount) 가 수정될때 재고제품의 amount 값에 더해질수 있도록 트리거 설정하세요

01 문제

환경설정(테이블)

```
use tempdb;  
drop table makeprodtbl;  
create table makeprodtbl  
(  
    code int not null,  
    prodname char(10) not null,  
    amount int  
);  
drop table saveprodtbl;  
create table saveprodtbl  
(  
    code int not null,  
    prodname char(10) not null,  
    amount int  
);
```

01 문제

환경설정(테이블)

```
insert into makeprodtbl values(111,'세탁기',10);  
insert into makeprodtbl values(222,'냉장고',5);  
insert into makeprodtbl values(333,'노트북',10);
```

```
insert into saveprodtbl values(111,'세탁기',50);  
insert into saveprodtbl values(222,'냉장고',10);  
insert into saveprodtbl values(333,'노트북',30);
```

```
select * from makeprodtbl;  
select * from saveprodtbl;
```

01 문제

문제 1

1 생산제품 테이블에 값 추가(INSERT) 시 재고 제품에도 추가될수 있도록 트리거 설정하세요

```
drop trigger if exists insertTrg;
delimiter //
create trigger insertTrg
after insert
on makeprodtbl
for each row
begin
    insert into saveprodtbl values(NEW.code , NEW.prodname , NEW.amount);
end//
delimiter ;
```

– 데이터 삽입

```
insert into makeprodtbl values(444,'TV',5);
```

– 확인

```
select * from makeprodtbl;
```

```
select * from saveprodtbl;
```

01 문제

문제 2

2 생산제품 테이블에 기존 제품의 생산량(amount)가 수정될때 제고제품의 amount 값에 더해질수 있도록 트리거 설정하세요

```
delimiter //  
create trigger updateTrg  
after update  
on makeprodtbl  
for each row  
begin  
  
    declare num int;  
    set num = new.amount-old.amount;  
    update saveprodtbl set amount=amount+num where code=new.code;  
end //  
delimiter ;
```

Part 3.

BEFORE 트리거



01 BEFORE 트리거

예제 1

```
use sqldb;
drop trigger if exists usertbl_beforeInsertTrg;
delimiter //
create trigger usertbl_beforeInsertTrg
before insert
on usertbl2
for each row
Begin
    if new.birthYear < 1900 then
        set new.birthYear = 0;
    elseif new.birthYear > year(curdate()) then
        set new.birthYear = year(curdate());
    end if;
end //
delimiter ;
```


01 BEFORE 트리거

예제 1

-- 확인

```
insert into usertbl2 values('aaa','에이',1877,'서울','011','1112222',181,'2022-12-15');
```

```
insert into usertbl2 values('bbb','비이',2977,'경기','011','1113333',171,'2019-03-25');
```

```
select * from usertbl2;
```

01 설계

회원테이블(memberTBL) - 개체Tbl

열이름	영문이름	데이터 형식	길이	NULL허용
아이디(PK)	memberID	문자(CHAR)	8글자(영문)	X
회원이름	memberName	문자(CHAR)	5글자(한글)	X
주소	memberAddress	문자(CHAR)	20글자(한글)	O

제품 테이블(productTBL) - 개체Tbl

열이름	영문이름	데이터 형식	길이	NULL허용
제품이름(PK)	productName	문자(CHAR)	4글자(한글)	X
가격	cost	숫자(INT)	정수	X
제조일자	makeDate	날짜(DATE)	날짜형	O
제조회사	company	문자(CHAR)	5글자(한글)	O
남은수량	amount	숫자(INT)	정수	X

01 설계

구매테이블(buyTBL) - 관계Tbl

열이름	영문이름	데이터 형식	길이	NULL허용
주문번호(PK)	buyID	숫자(INT)	정수	X
아이디(FK)	memberID	문자(CHAR)	8글자(영문)	X
제품이름(FK)	productName	문자(CHAR)	4글자(한글)	X
구매수량	amount	숫자(INT)	정수	X

01 문제

[계약조건 문제]

- 1 각 테이블들은 제약 조건 없이 만들어 주세요
- 2 회원테이블의 아이디(PK)와 구매 테이블의 아이디(FK)를 관계 설정하세요
- 3 제품테이블의 제품이름(PK)와 구매테이블의 제품이름(FK)를 관계 설정하세요
- 4 구매테이블의 주문번호에 PK 설정하세요

01 문제

환경설정(테이블+제약조건)

-- DB 생성

```
create database tmpdb;
```

```
use tmpdb;
```

-- 테이블 생성

```
create table memberTbl
```

```
(
```

```
memberID char(8) not null,
```

```
memberName char(5) not null,
```

```
memberaddress nchar(20) null
```

```
);
```

01 문제

환경설정(테이블+제약조건)

```
create table productTbl
(
  productName char(4) not null,
  cost int not null,
  makedate date null,
  company nchar(5) null,
  amount int not null
);
create table buytbl
(
  buyID int not null,
  memberID char(8) not null,
  productName char(4) not null,
  amount int not null
);
```

01 문제

환경설정(테이블+제약조건)

```
-- 구조 확인  
desc membertbl;  
desc productTbl;  
desc buytbl;
```

01 문제

관계 설정

-- 관계 설정

```
alter table membertbl add primary key(memberID);  
alter table productTbl add primary key(productName);  
alter table buytbl add primary key (buyID);  
alter table buytbl add constraint FK_memTbl_buyTbl  
foreign key(memberID) references membertbl(memberID);  
alter table buytbl add constraint FK_prodTbl_buyTbl  
foreign key(productName) references productTbl(productName);
```

```
desc membertbl;  
desc producttbl;  
desc buytbl;
```


01 문제

[BEFORE 트리거 문제]

- 1 주문 테이블에 제품주문시(INSERT) 제품 테이블의 재고량이 주문수량보다 작으면 INSERT 불가 설정
- 2 주문 테이블의 주문 수량 변경(UPDATE) 시 제품 테이블에도 수량에 맞게 amount 가 조절되도록 트리거 설정
- 3 주문 테이블의 주문 취소시(DELETE) 제품 수량도 맞게 amount 가 조절되도록 트리거 설정

01 문제

1 답

-- 1 주문 테이블에 제품주문시(INSERT) 제품 테이블의 재고량이 주문수량보다 작으면 INSERT 불가 설정

```
drop trigger if exists InsertbuytblTrg;
```

```
delimiter //
create trigger InsertbuytblTrg
before insert
on buytbl
for each row
begin
    declare pamount int;
    set pamount = (select amount from producttbl where productname=new.productname);
    if (new.amount -pamount <= 0) then
        update producttbl set amount = amount-new.amount where productname=new.productname;
    else
        delete from new;
    end if;
end //
delimiter ;
```

01 문제

2 답

-- 2 주문 테이블의 주문 수량 변경(UPDATE) 시 제품 테이블에도 수량에 맞게 amount 가 조절되도록 트리거 설정

```
drop trigger if exists UpdatebuytblTrg;
delimiter //
create trigger UpdatebuytblTrg
before update
on buytbl
for each row
begin
    declare sub int;
    set sub = (select amount from buytbl where productName=new.productName) - new.amount ;
    if sub >= 0 then
        update producttbl set amount=amount+sub where productname = new.productname;
    else
        delete from new;
    end if;
-- sql 코드
end //
delimiter ;
```

01 문제

2 답

-- 3 주문 테이블의 주문 취소시(DELETE) 제품 수량도 맞게 amount 가 조절되도록 트리거 설정

```
drop trigger if exists deletebuytblTrg;
delimiter //
create trigger deletebuytblTrg
before delete
on buytbl
for each row
begin

    update producttbl set amount=amount+old.amount where productname=old.productname;

end //
delimiter ;
```

END.

고생하셨습니다

