Triggers

==========

1)Triggers are similar to stored procedure which are

stored in the DBMS system.

2)Triggers are associated with a certain table.

3)when any DML operations related to the table is performed,

a trigger associated with that table for that DML operations

is executed automatically.

There are two types of triggers

1)row level trigger

2)statement level trigger

mysql supports only row level trigger

syntax:

delimiter //

create trigger trigger\_name{before/after}

{insert/update/delete} on tablename

for each row

BEGIN

trigger body

END //

USE:

====

1)Database auditing

=> In database auditing, we have the records of transaction

such as insert,delete and update similar to the

transaction of credit and debit in the bank.

so, we need to create a different table to store the

records of insert,delete and update transactions on

another table associated with the trigger.

2)Integrity constraint

---------------------------------------

If you need to store values of the column of the

record on which the DML operation is performed, we use

new and old keywords

DML new old

----------------------------

insert Yes No

update Yes Yes

delete No Yes

create database db;

use db;

create table employee(id int primary key auto\_increment,name varchar(50), dept varchar(50),sal float,doj date);

insert into employee(name,dept,sal,doj)

values('harry','HR',45000,'2022-09-09'),

('mac','IT',60000,'2023-09-08');

insert into employee(name,dept,sal,doj)

values('hari','production',40000,'2021-09-09'),

('shree','IT',65000,'2023-09-01');

select \* from employee where id=3;

delimiter //

create procedure abc()

begin

select \* from employee;

end//

show procedure status where db="db";

call abc();

drop procedure abc;

delimiter //

create procedure abc1(IN x int,IN y int)

begin

declare z int;

set z=x+y;

select z as result;

end //

delimiter ;

call abc1(10,20);

SET @y=20;

select @y;

call abc(10,@y);

delimiter //

create procedure emp\_grade(IN x int)

begin

declare temp float;

select sal into temp from employee where id=x;

if temp>=50000 then

select 'platinum employee grade' as grade;

else

select 'gold employee grade' as grade;

end if;

end //

call emp\_grade(2);

drop procedure emp\_grade;

delimiter //

create procedure emp\_grade(IN x int)

BEGIN

DECLARE temp float;

select sal INTO temp from employee where id=x;

IF temp>=60000 THEN

select 'Platinum employee' as grade;

ELSEIF temp>=45000 and temp<60000 THEN

select 'GOLD employee' as grade;

ELSE

select 'silver employee' as grade;

END IF;

END //

call emp\_grade(4);

delimiter //

create procedure loop1(IN x int)

begin

declare i int;

set i=1;

while i<=x DO

select i;

set i=i+1;

end while;

end //

call loop1(5);

select \* from employee;

delimiter //

create procedure loop2()

begin

declare i int;

declare temp float;

set i=1;

while i<=4 do

select sal into temp from employee where id=i;

if temp>=60000 then

select 'platinum employee as grade';

else

select 'gold employee as grade ';

end if;

set i=i+1;

end while;

end //

call loop2();

delimiter //

create function my\_grade(x float)

RETURNS varchar(50)

reads sql data

deterministic

BEGIN

DECLARE res varchar(50);

IF x>=60000 THEN

SET res='Platinum';

ELSE

SET res='silver';

END IF;

RETURN (res);

END //

delimiter //

create function myfun(x float)

returns varchar(20)

reads sql data

deterministic

begin

declare res varchar(20);

if x>=60000 then

set res='paltinum';

else

set res='gold';

end if;

return (res);

end //

select name,dept,sal,myfun(sal) from employee;

delimiter //

create procedure emp\_grade2()

begin

declare s float;

declare n varchar(20);

declare cur cursor for select name,sal from employee;

open cur ;

myloop :Loop

fetch cur into n,s ;

if s>=6000 then

select n as name,s as salary,'platinum' as grade;

else

select n as name, s as salary ,'gold' as grade;

end if;

end loop;

close cur;

end //

call emp\_grade2();

delimiter //

create procedure emp\_grade6()

begin

declare s float;

declare done int default false;

declare cur cursor for select sal from employee;

declare continue handler for not found set done=true;

open cur ;

myloop :Loop

fetch cur into s ;

if done then

leave myloop;

end if;

if s>=6000 then

select s as salary,'platinum' as grade;

else

select s as salary ,'gold' as grade;

end if;

end loop;

close cur;

end //

call emp\_grade6();

select \* from employee;

alter table employee drop column doj;

select \* from employee;

create table logtable (id int primary key auto\_increment,action varchar(50),datetime datetime);

desc logtable;

select \* from logtable;

delimiter //

create trigger insert\_trigger after insert on employee

for each row

begin

insert into logtable(action,datetime)values('insert',now());

end //

delimiter //

select \* from employee;

insert into employee(name,dept,sal)values('pqr','HR',56000);

select \* from employee;

select \* from logtable;

insert into employee(name,dept,sal)values('aaa','IT',70000),('bbb','Production',45000);

select \* from employee;

select \* from logtable;

delimiter //

create trigger update\_trigger after update on employee

for each row

begin

insert into logtable(action,datetime)values('update',now());

end //

update employee set dept='CS' where id=3;

select \* from employee;

select \* from logtable

delimiter //

create trigger delete\_trigger after delete on employee

for each row

BEGIN

insert into logtable(action,datetime) values('delete',now());

END //

delimiter ;

select \* from employee;

select \* from logtable;

delete from employee where id=4;

select \* from logtable;

delete from employee where id=3 or id=6;

select \* from employee;

select \* from logtable;

alter table logtable add lvalue varchar(20);

alter table logtable drop column value;

alter table logtable add nvalue varchar(20);

desc logtable;

drop trigger insert\_trigger;

drop trigger delete\_trigger;

drop trigger update\_trigger;

select \* from logtable;

delimiter //

create trigger insert\_trigger after insert on employee

for each row

begin

insert into logtable(action,datetime,nvalue)values

('insert',now(),concat(new.name,'\_',new.dept,'\_',new.sal));

end //

delete from employee where name="Pooja";

insert into employee(name,dept,sal)values('pooja','HR',34000);

select \* from employee;

select \* from logtable;

delimiter //

create trigger update\_trigger after update on employee

for each row

BEGIN

insert into logtable(action,datetime,lvalue,nvalue)

values('update',now(),CONCAT(old.name,'-',old.dept,'-',old.sal),

CONCAT(new.name,'-',new.dept,'-',new.sal));

END //

delimiter ;

update employee SET dept='IT',sal=56666 where id=5;

select \* from employee;

select \* from logtable;

drop trigger insert\_trigger;

delimiter //

create trigger insert\_trigger before insert on employee

for each row

BEGIN

IF new.sal<=10000 THEN

signal sqlstate '45000' SET message\_text="Salary must be grea

ter than 10000";

END IF ;

END //

delimiter ;