DBMSL Mock Test

Aditya Sawant

31302

**Problem Statement:**

**7. Triggers:**

Consider a Employee database and write row-level triggers for the same. Implement both before & after triggers for the relevant database tables at the time of insertion, update and deletion. Implement using following schema:

**EMPLOYEE**(Emp\_Id, First\_Name, Last\_Name, Email, Phone\_No, Hire\_Date, Job\_Profile, Salary, HRA)

**COMPANY\_INFO**(Emp\_Count, Salary\_Expenses)

**EMP\_LOG**(Emp\_Id, Old\_Salary, New\_Salary, Edit\_Time, Job\_Status)

* **Before insert:** Check the column value of FIRST\_NAME, LAST\_NAME, JOB\_ID for following criteria:
* If there are any spaces before or after the FIRST\_NAME, LAST\_NAME, use TRIM() function to remove them.
* The value of the JOB\_PROFILE will be converted to upper cases by UPPER() function.
* **After insert:** Every time an INSERT happens into EMPLOYEE table, insert relevant information into the EMP\_LOG table. Also update the COMPANY\_INFO table
* **Before update:** Each time the HRA is updated for the EMPLOYEE table, convert it into decimal value (i.e. for 10%, store 0.1)
* **After Update:** Each time the HRA is updated, accordingly update the salary in EMPLOYEE table & keep track of updated salary in EMP\_LOG table.
* **Before Delete:** Every time a DELETE happens on EMPLOYEE table, accordingly change the JOB\_STATUS in EMP\_LOG table from ACTIVE to DELETED & keep track of EDIT\_TIME
* **After Delete:** Keep the COMPANY\_INFO table updated

**Queries:**

delimiter //

create trigger t1

before insert

on employee for each row

begin

set new.first\_name = trim(new.first\_name);

set new.last\_name = trim(new.last\_name);

set new.job\_profile = upper(new.job\_profile);

end;

//

delimiter //

create trigger t2

after insert

on employee for each row

begin

declare emp\_cnt integer;

declare sal\_exp integer;

declare ifempty integer;

select count(emp\_count) into ifempty from company\_info;

if ifempty != 0 then

select emp\_count, salary\_expenses into emp\_cnt, sal\_exp from company\_info where emp\_count = ifempty;

insert into company\_info values(emp\_cnt+1, sal\_exp+new.salary);

else

insert into company\_info values(1, new.salary);

end if;

insert into emp\_log values(new.emp\_id, 0, new.salary, current\_time(), 'Active');

end;

//

delimiter //

create trigger t3

before update

on employee for each row

begin

declare hhra integer;

if old.hra > 1 then

set new.hra = new.hra/100;

elseif new.hra > 1 then

set new.hra = new.hra/100;

end if;

end;

//

delimiter //

create trigger t4

after update

on employee for each row

begin

if old.hra != new.hra then

insert into emp\_log values(new.emp\_id, old.salary, (old.salary\*new.hra)+old.salary, current\_time(), 'Active');

end if;

end;

//

delimiter //

create trigger t5

before delete

on employee for each row

begin

insert into emp\_log values(old.emp\_id, old.salary, old.salary, current\_time(), 'Deleted');

end;

//

delimiter //

create trigger t6

after delete

on employee for each row

begin

declare emp\_cnt integer;

declare sal\_exp integer;

declare maxcount integer;

select count(emp\_count) into maxcount from company\_info;

if ifempty != 0 then

select emp\_count, salary\_expenses into emp\_cnt, sal\_exp from company\_info where emp\_count = maxcount;

insert into company\_info values(emp\_cnt-1, sal\_exp-old.salary);

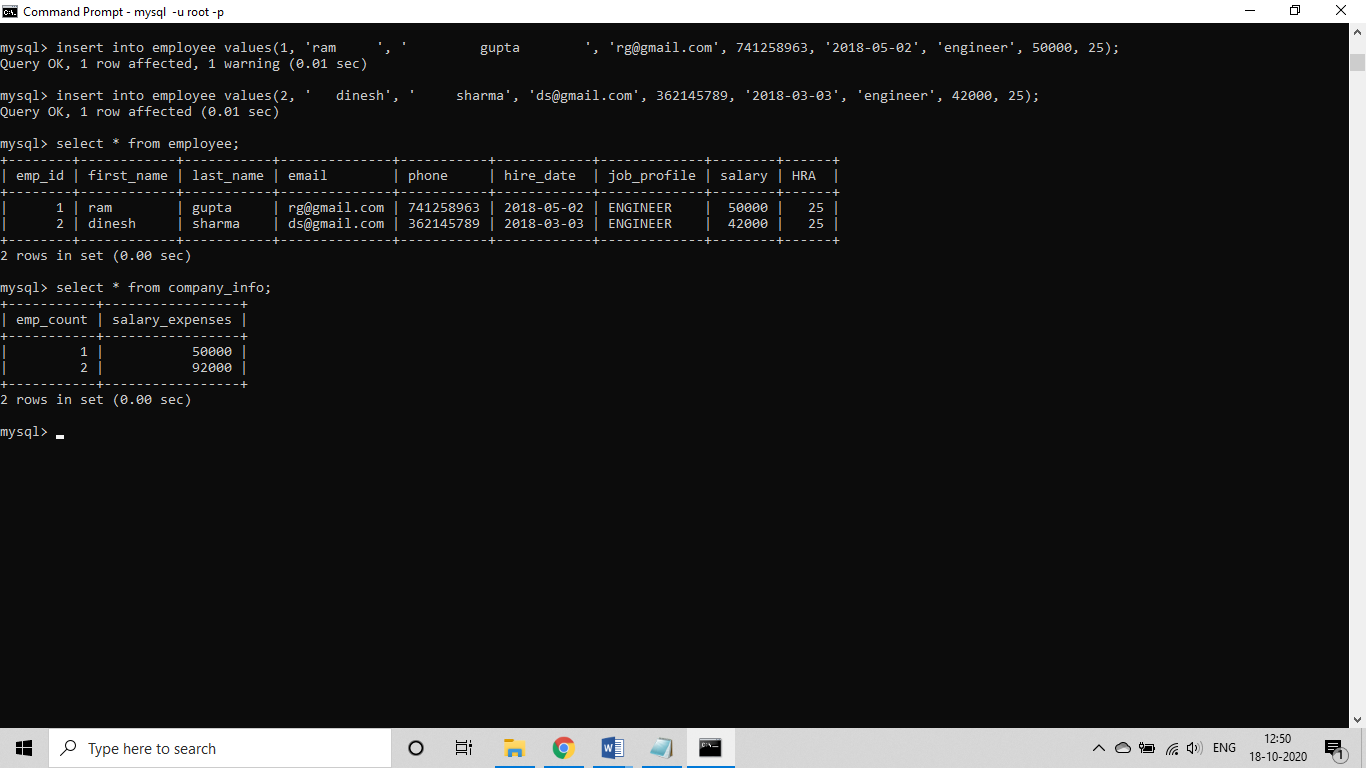
end if;

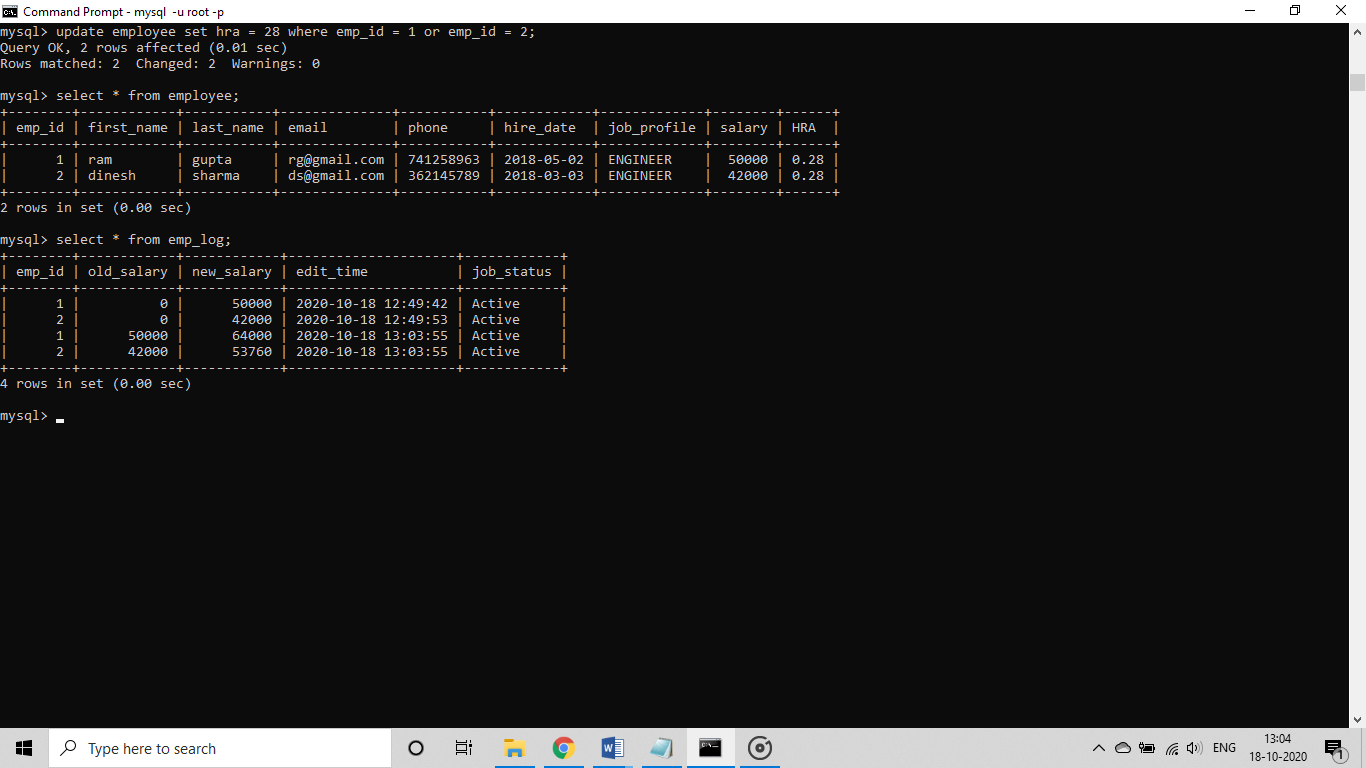
end;

//

**Output:**

Insert triggers:



Update triggers: 

Delete triggers:

