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:

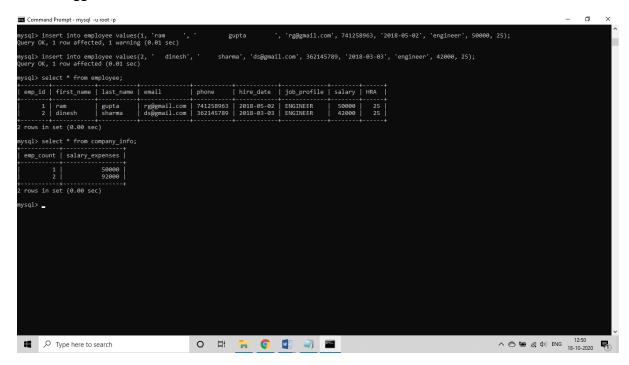
```
delimiter //
create trigger t1
before insert
on employee for each row
begin
set new.first_name = trim(new.first_name);
set new.last_name = trim(new.last_name);
set new.job_profile = upper(new.job_profile);
end;
//
delimiter //
create trigger t2
after insert
on employee for each row
begin
declare 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;
//
```

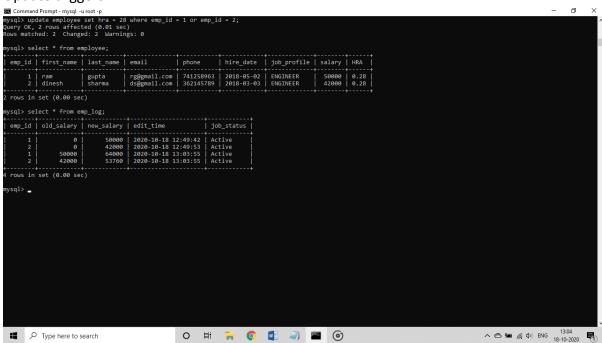
```
delimiter //
create trigger t5
before delete
on employee for each row
begin
insert into emp_log values(old.emp_id, old.salary, old.salary, current_time(), 'Deleted');
end;
//
delimiter //
create trigger t6
after delete
on employee for each row
begin
declare emp_cnt integer;
declare sal_exp integer;
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:

