Answers to Trigger Problem Statements

1. **Create before insert trigger to increase salary of instructors by 10 dollars**

create trigger before\_trigger\_example

before insert

on instructor

for each row

set new.salary = new.salary + 10;

insert into instructor (ID, name, dept\_name, salary)

values (101, 'John', 'Comp. Sci.', 200000);

select \* from instructor

where name = 'John';

1. **Create after trigger to add id and salary of employees into a new table after insert**

drop table id\_and\_salary;

create table id\_and\_salary (

id\_of\_instructor int,

salary\_of\_instructor int);

describe id\_and\_salary;

drop trigger before\_trigger\_example;

create trigger after\_trigger\_example

after insert

on instructor

for each row

insert into id\_and\_salary values (new.ID, new.salary);

insert into instructor (ID, name, dept\_name, salary)

values (102, 'Jim', 'Comp. Sci.', 300000);

select \* from id\_and\_salary;

select \* from instructor

where ID = 102;

1. **Create a trigger which will convert all letters of instructor names to upper case before updating the table**

create trigger conv\_ins\_name\_to\_uppercase

before update

on instructor

for each row

set new.name = upper(new.name);

select \* from instructor;

select \*

from instructor

where name = 'Jim';

update instructor

set name = 'Jazz'

where name = 'Jim';

select \*

from instructor

where name like 'J%';

1. **Create a trigger which will create a new table to add id and names of instructors after increasing salary of all instructors whose name starts with A**

drop table id\_name\_of\_ins;

create table id\_name\_of\_ins (

name\_of\_ins varchar(30),

sal\_of\_ins int);

select name, salary

from instructor

where name like 'A%';

create trigger after\_update\_trigger

after update

on instructor

for each row

insert into id\_name\_of\_ins values (new.name, new.salary);

update instructor

set salary = salary + 100

where name like 'A%';

select \* from id\_name\_of\_ins;

truncate table id\_name\_of\_ins;

drop trigger after\_update\_trigger;

1. **Create trigger which will store all the data of instructors which we will delete, instructors to be deleted must have ‘i’ as 2nd and ‘e’ as 3rd letter in their name.**

describe instructor;

create table deleted\_data\_of\_ins (

ins\_id varchar(5) primary key,

i\_name varchar(20),

ins\_dept varchar(20),

i\_sal int);

describe deleted\_data\_of\_ins;

DELIMITER $$

CREATE TRIGGER before\_data\_ofins\_delete

BEFORE DELETE

ON instructor

FOR EACH ROW

BEGIN

INSERT INTO deleted\_data\_of\_ins(ins\_id, i\_name, ins\_dept, i\_sal)

VALUES(OLD.ID, OLD.name, OLD.dept\_name, OLD.salary);

END$$

DELIMITER ;

select name

from instructor

where name like '\_ie%';

delete from instructor

where name like '\_ie%';

select \* from deleted\_data\_of\_ins;