1.Create a Trigger for employe table it will update another table salary while updating values.To develop and execute a Trigger for After update/Delete/Insert operations on a table.

sql>

CREATE TABLE `employe` (

`emp\_id` int(11) NOT NULL,

`emp\_name` varchar(45) DEFAULT NULL,

`dob` date DEFAULT NULL,

`address` varchar(45) DEFAULT NULL,

`designation` varchar(45) DEFAULT NULL,

`mobile\_no` int(11) DEFAULT NULL,

`dept\_no` int(11) DEFAULT NULL,

`salary` int(11) DEFAULT NULL,

PRIMARY KEY (`emp\_id`)

);

Sql>

CREATE TABLE `salary` (

`employee\_id` int(11) NOT NULL,

`old\_sal` int(11) DEFAULT NULL,

`new\_sal` int(11) DEFAULT NULL,

`rev\_date` date DEFAULT NULL,

PRIMARY KEY (`employee\_id`)

);

sql>

CREATE DEFINER=`root`@`localhost` TRIGGER `db1`.`personal\_updations\_AFTER\_UPDATE\_1`

AFTER UPDATE ON `employe`

FOR EACH ROW

BEGIN

if(new.salary != old.salary)

then

INSERT INTO salary (employee\_id,old\_sal,new\_sal,rev\_date) values (new.emp\_id,old.salary,new.salary,sysdate());

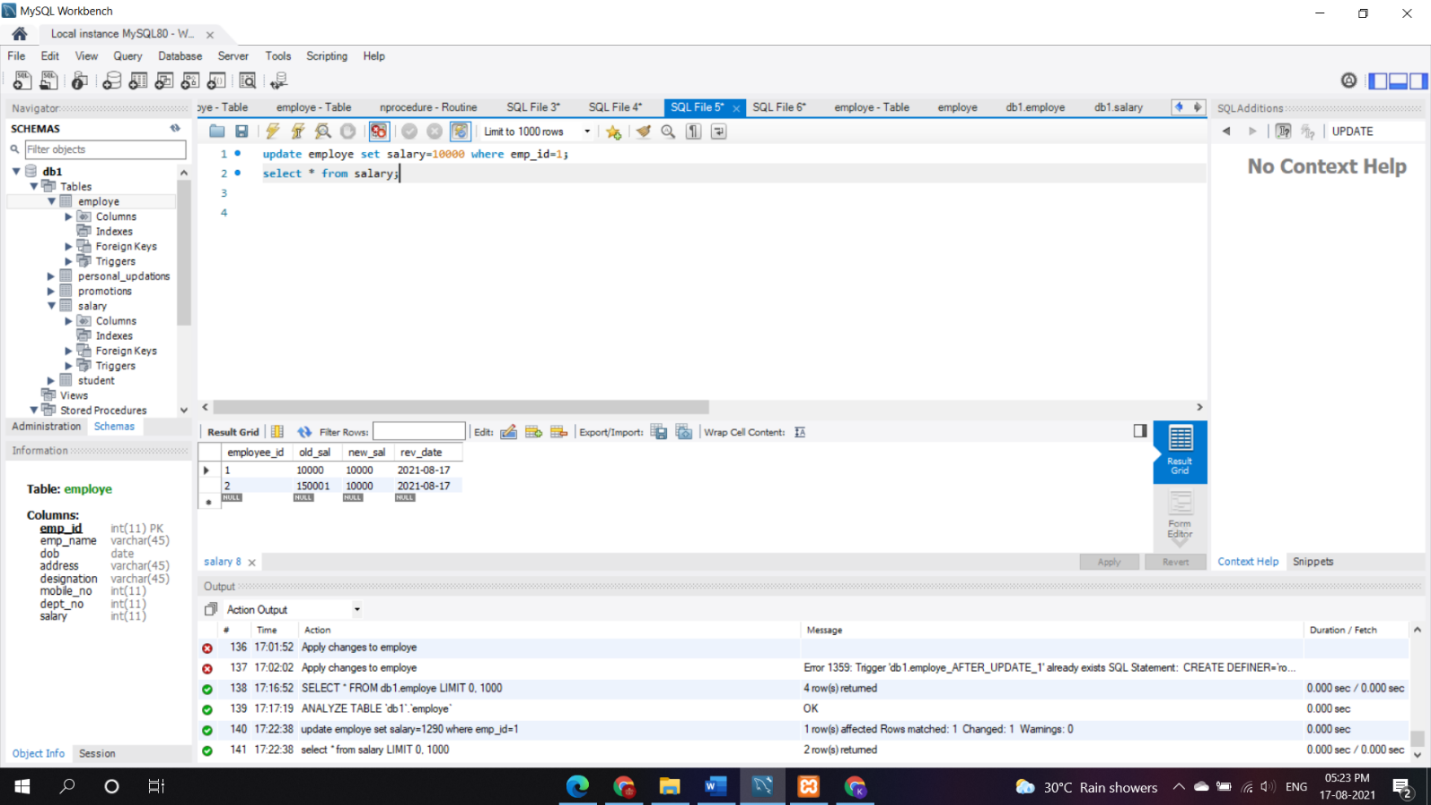
END if;

end;

sql>

update employe set salary=234569 where emp\_id=1;

select \* from salary;



2.Create a Trigger for employe table it will update another table personal\_updations while updating values. To develop and execute a Trigger for Before and After update/Delete/Insert operations on a table.

Sql>

CREATE TABLE `employe` (

`emp\_id` int(11) NOT NULL,

`emp\_name` varchar(45) DEFAULT NULL,

`dob` date DEFAULT NULL,

`address` varchar(45) DEFAULT NULL,

`designation` varchar(45) DEFAULT NULL,

`mobile\_no` int(11) DEFAULT NULL,

`dept\_no` int(11) DEFAULT NULL,

`salary` int(11) DEFAULT NULL,

PRIMARY KEY (`emp\_id`)

);

Sql>

CREATE TABLE `personal\_updations` (

`emp\_id` int(11) NOT NULL,

`old\_phoneno` int(11) DEFAULT NULL,

`new\_phoneno` int(11) DEFAULT NULL,

`rev\_date` date DEFAULT NULL,

PRIMARY KEY (`emp\_id`)

);

sql>

CREATE DEFINER=`root`@`localhost` TRIGGER `db1`.`personal\_updations\_AFTER\_UPDATE`

AFTER UPDATE ON `employe`

FOR EACH ROW

BEGIN

if(new.mobile\_no != old.mobile\_no)

then

INSERT INTO personal\_updations (emp\_id,old\_phoneno,new\_phoneno,rev\_date) values (new.emp\_id,new.mobile\_no,old.mobile\_no,sysdate());

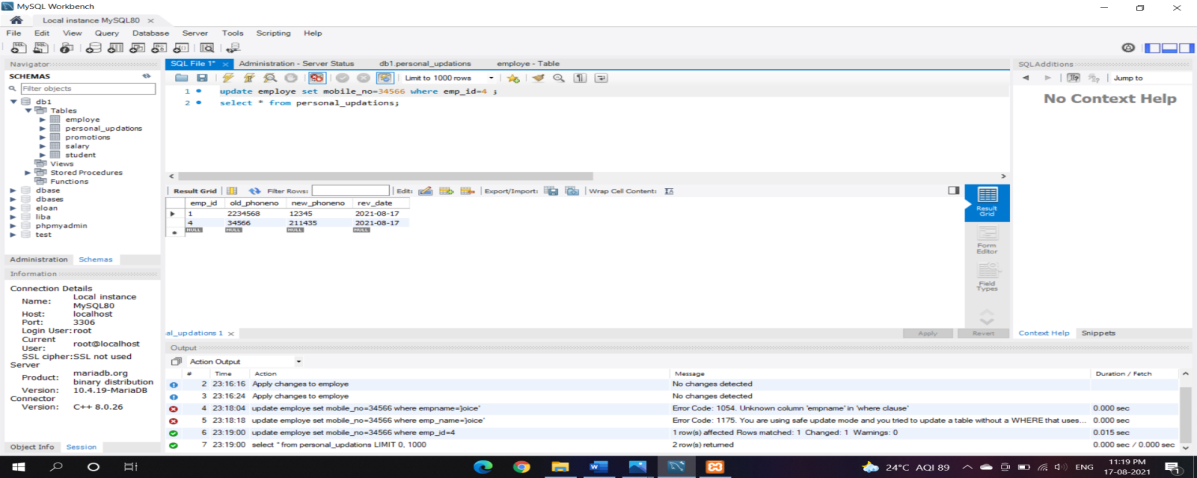
END if;

end;

sql>

update employe set mobile\_no=34566 where emp\_id=4 ;

select \* from personal\_updations;



3.Create a Trigger for employe table it will update another table promotions while updating values. To develop and execute a Trigger for Before and After update/Delete/Insert operations on a table.

sql>

CREATE TABLE `employe` (

`emp\_id` int(11) NOT NULL,

`emp\_name` varchar(45) DEFAULT NULL,

`dob` date DEFAULT NULL,

`address` varchar(45) DEFAULT NULL,

`designation` varchar(45) DEFAULT NULL,

`mobile\_no` int(11) DEFAULT NULL,

`dept\_no` int(11) DEFAULT NULL,

`salary` int(11) DEFAULT NULL,

PRIMARY KEY (`emp\_id`)

);

Sql>

CREATE TABLE `personal\_updations` (

`emp\_id` int(11) NOT NULL,

`old\_phoneno` int(11) DEFAULT NULL,

`new\_phoneno` int(11) DEFAULT NULL,

`rev\_date` date DEFAULT NULL,

PRIMARY KEY (`emp\_id`)

);

sql>

CREATE DEFINER=`root`@`localhost` TRIGGER`db1`.`employe\_AFTER\_UPDATE\_1`

AFTER UPDATE ON `employe`

FOR EACH ROW

BEGIN

if(new.designation != old.designation)

then

INSERT INTO promotions (emp\_id,old\_designation,new\_designation,rev\_date) values (new.emp\_id,new.designation,old.designation,sysdate());

END if;

end;

sql>

update employe set designation='clk' where emp\_id=4 ;

select \* from promotions;

