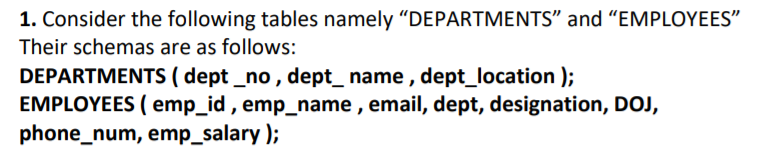


|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | CSLR 51 : DBMS LAB-6 | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Roll no. : 106119100Name : Rajneesh PandeySection : CSE-B |  | | | |
|  | | |  | | |  | | |

PROBLEM 1



**ER-Diagram**



CREATE DATABASE CompanyDB;

USE CompanyDB;

CREATE TABLE Departments  (

    dept\_id VARCHAR (20) NOT NULL PRIMARY KEY,

    dept\_name VARCHAR (20),

    dept\_location VARCHAR (20)

);

CREATE TABLE Employees (

  emp\_id INT NOT NULL PRIMARY KEY,

  emp\_name VARCHAR(100),

  email VARCHAR(100),

  dept\_id  INT,

  designation VARCHAR(100),

  doj DATE,

  phone\_num VARCHAR(45),

  emp\_salary DECIMAL(12,3)

);

create table messages

(

    id int(11) not null auto\_increment primary key,

    message varchar(255) default null,

    time timestamp null default current\_timestamp on update current\_timestamp

);

/\*\* ----------------------------- Queries -------------------------------\*\*/

-- Create a trigger for:-

-- a) Calculating the number of rows inserted so far.

delimiter //

create trigger trig1A

 before insert on departments

  for each row

  set @dept\_rows = @dept\_rows+1//

set @dept\_rows = 0//

delimiter ;

insert into departments

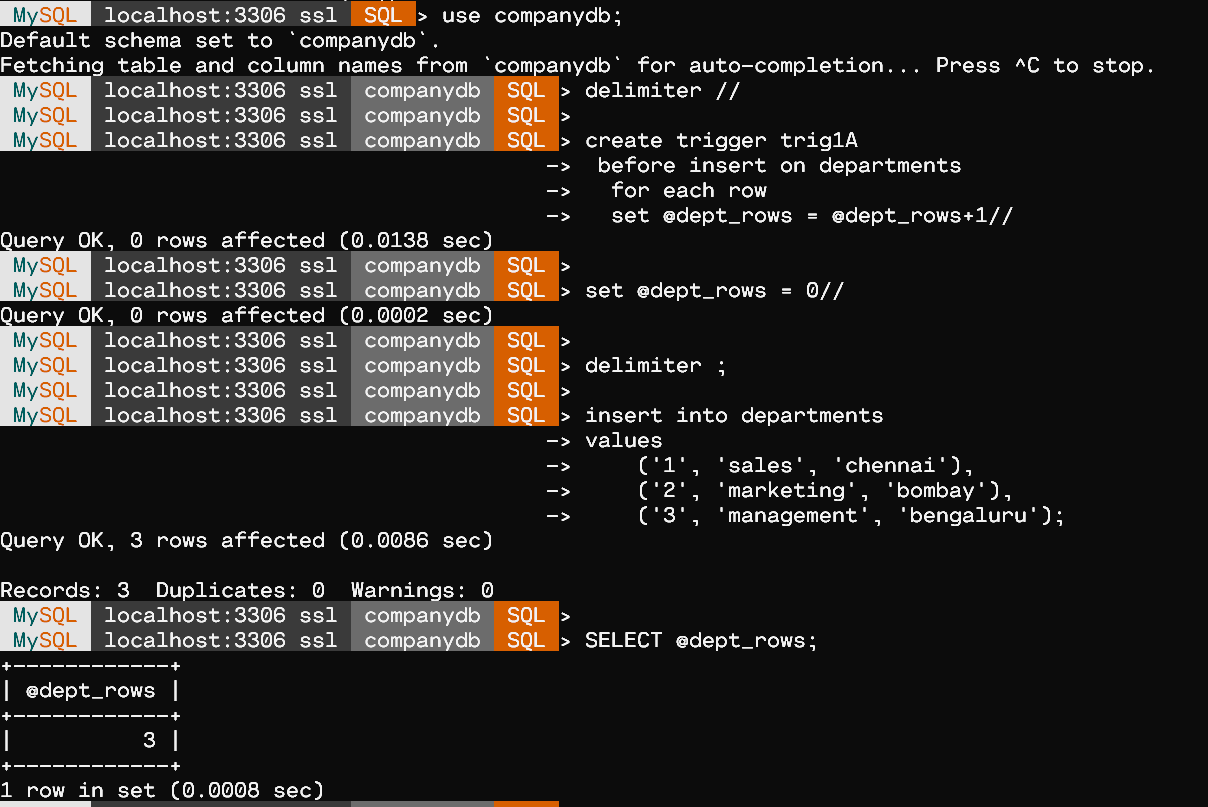
values

    ('1', 'sales', 'chennai'),

    ('2', 'marketing', 'bombay'),

    ('3', 'management', 'bengaluru');

SELECT @dept\_rows;



-- b) Displays message prior to insert operation on Employee Table

delimiter //

create trigger display\_message before insert on employees

    for each row

    begin

    insert into messages (message) values('Inserted Row in Employees');

    end;//

delimiter ;

insert into employees

values

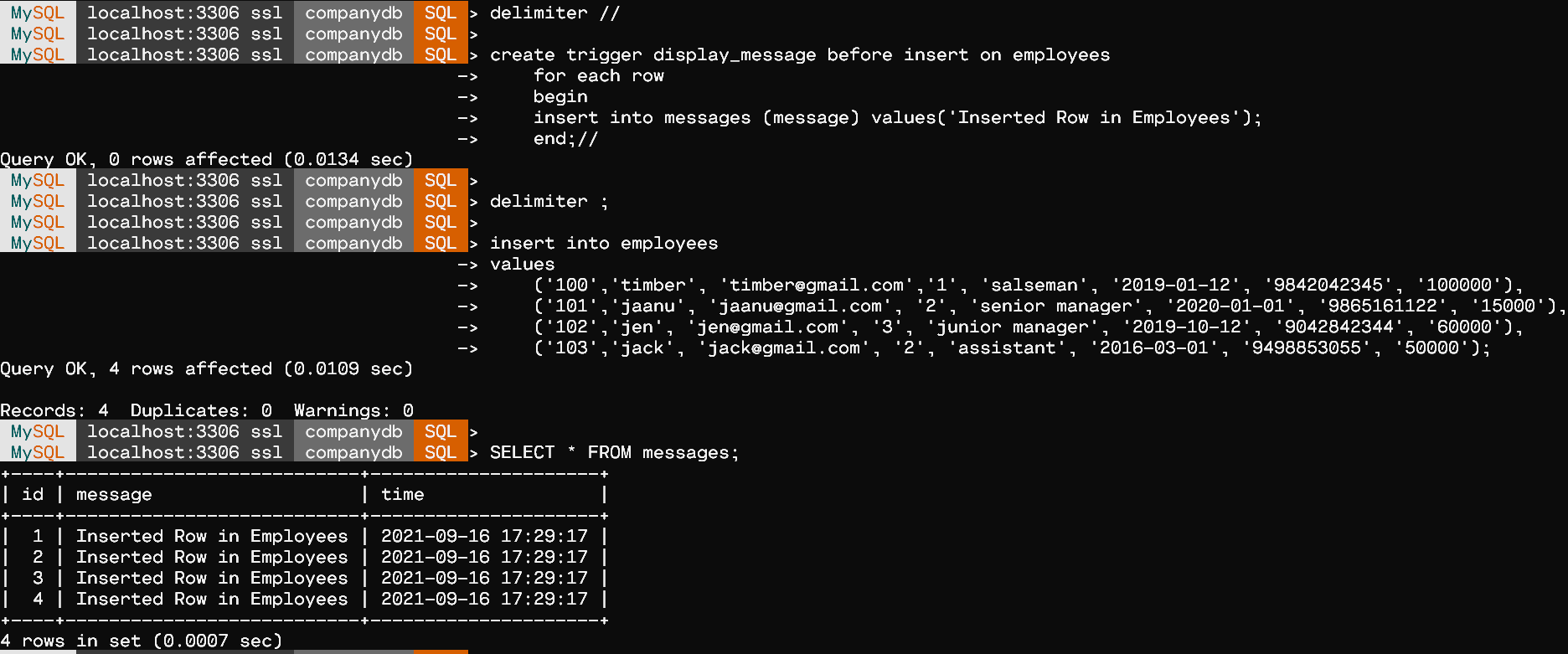
    ('100','timber', 'timber@gmail.com','1', 'salseman', '2019-01-12', '9842042345', '100000'),

    ('101','jaanu', 'jaanu@gmail.com', '2', 'senior manager', '2020-01-01', '9865161122', '15000'),

    ('102','jen', 'jen@gmail.com', '3', 'junior manager', '2019-10-12', '9042842344', '60000'),

    ('103','jack', 'jack@gmail.com', '2', 'assistant', '2016-03-01', '9498853055', '50000');

SELECT \* FROM messages;



-- c) Ensuring that the Employee Table does not contain duplicates or null value in the Name column

delimiter //

create trigger NULL\_val\_trigger before insert on employees

for each row

begin

    if (exists(select 1 from employees where emp\_name = new.emp\_name or new.emp\_name is null))

    then

    signal sqlstate '45001' set message\_text = "INSERT failed due to duplicate or null emp\_name";

    end if;

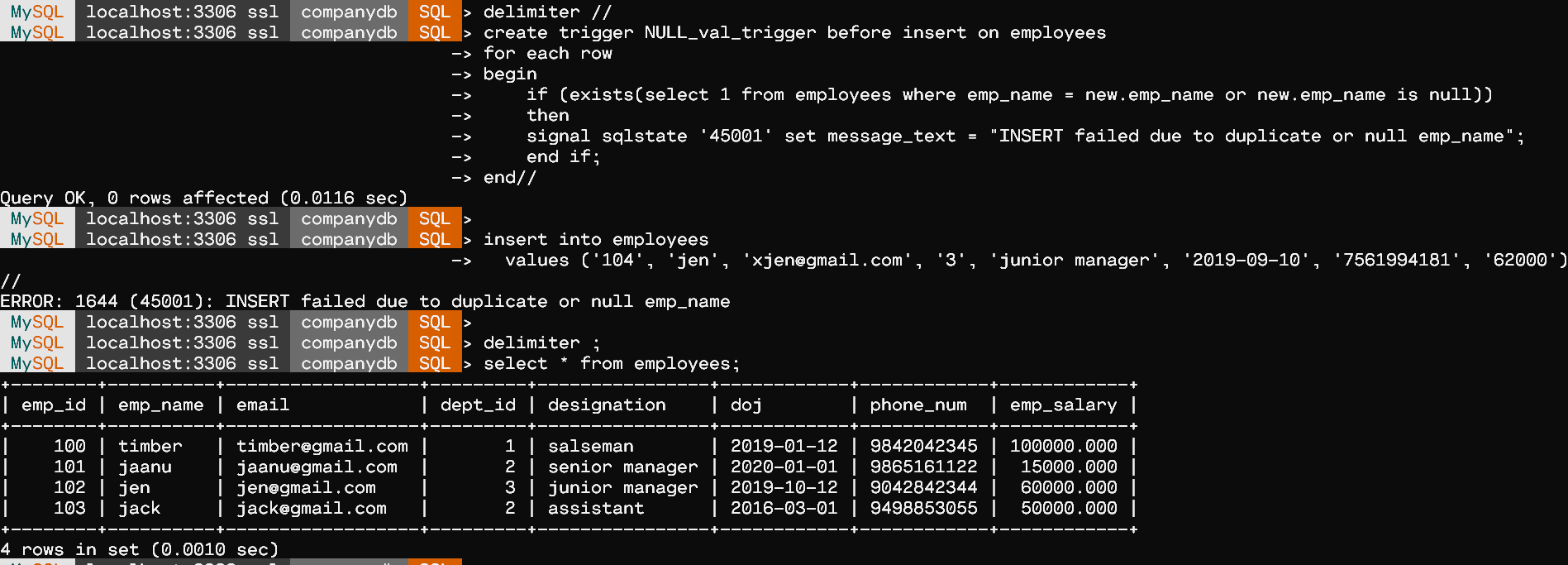
end//

insert into employees

  values ('104', 'jen', 'xjen@gmail.com', '3', 'junior manager', '2019-09-10', '7561994181', '62000')//

delimiter ;

select \* from employees;



-- d) To insert Employee details into Employee Table only of DOJ > 2018

DELIMITER //

CREATE TRIGGER NOT\_BELOW\_2018

    BEFORE INSERT

    ON EMPLOYEES

    FOR EACH ROW

BEGIN

    IF(NEW.doj < '2018-01-01') THEN

        SIGNAL SQLSTATE VALUE '45000' SET MESSAGE\_TEXT = 'EMPLOYEE BELOW 2018 DOJ CANNOT BE INSERTED';

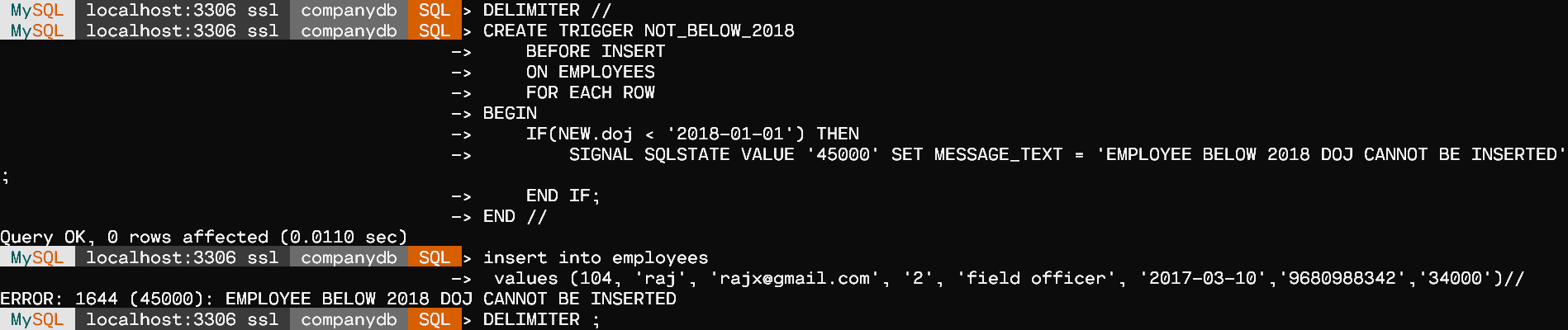
    END IF;

END //

insert into employees

 values (104, 'raj', 'rajx@gmail.com', '2', 'field officer', '2017-03-10','9680988342','34000')//

DELIMITER ;



-- e) To Prevent any Employee Named Tom to be inserted into the table

DELIMITER //

create trigger prevent\_name before insert on employees

    for each row

    begin

    if new.emp\_name = "tom" then signal sqlstate '45000' set message\_text= "tom is prohibited from insertion !!!";

    end if;

    end//

DELIMITER ;

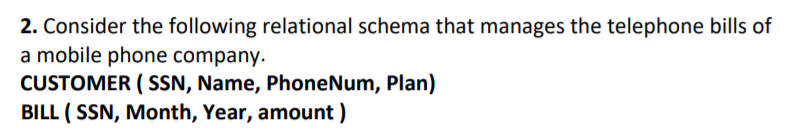
insert into employees values

  ('107', 'tom', 'tommy@gmail.com', '6', 'assistant doctor', '2020-02-22', '7373558197','25000');

select \* from employees;



PROBLEM 2

****

****

CREATE DATABASE customerDB;

USE customerDB;

create table customer

(

    ssn int primary key,

    name varchar(30),

    phone\_no varchar(30),

    plan varchar(30)

);

create table bill

(

    ssn int not null,

    foreign key (ssn) references customer(ssn),

    month int not null,

    year int not null,

    amount int not null

);

create table calls

(

    from\_ssn int,

    to\_ssn int,

    month int,

    year int,

    amount int,

    foreign key(from\_ssn) references customer(ssn),

    foreign key(to\_ssn) references customer(ssn)

);

-- a. That Adds +91 to all phone numbers as a prefix in the Customer Table.

 delimiter //

 create trigger trig2A

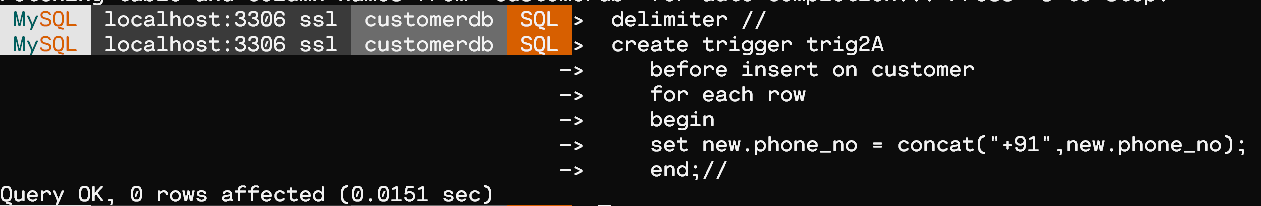
    before insert on customer

    for each row

    begin

    set new.phone\_no = concat("+91",new.phone\_no);

    end;//



-- b. Write a trigger that after each phone call updates the customer's bill.

delimiter //

create trigger trig2B after insert on calls

    for each row

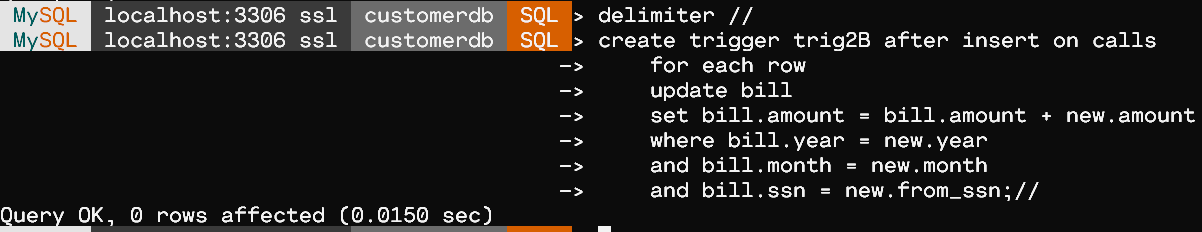
    update bill

    set bill.amount = bill.amount + new.amount

    where bill.year = new.year

    and bill.month = new.month

    and bill.ssn = new.from\_ssn;//



 -- c. Write a trigger to not insert bill older than 2020 (year > 2020).

 delimiter //

create trigger trig2C before insert on bill

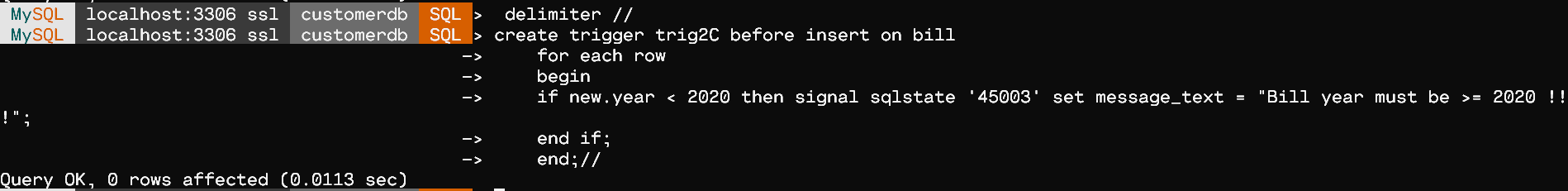
    for each row

    begin

    if new.year < 2020 then signal sqlstate '45003' set message\_text = "Bill year must be >= 2020 !!!";

    end if;

    end;//

****