Ejercicios 51-60

BASES DE DATOS PARA APLICACIONES

Julio Alejandro Tejada Nava

2018

|  |  |  |  |
| --- | --- | --- | --- |
| # | Captura | Query | |
| 51 |  | CREATE TRIGGER `bajas`  AFTER DELETE ON `empleado`  FOR EACH ROW  BEGIN  insert into empleados\_baja values (old.no\_empleado,old.nombre,old.edad,old.oficio,.old.direccion,old.fecha\_alta,old.salario,old.comision,old.no\_depto,user(),now());  END  delete from empleado where no\_empleado=7; |
| 52 |  | CREATETRIGGER `aumentoSalario`  BEFORE UPDATE ON `empleado`  FOR EACH ROW  BEGIN  if new.salario > (old.salario\*1.20) THEN  Signal sqlstate '02000' set message\_text ='Nose puede aumentar el salario mas de un 20%'; end IF; END  update empleado set salario = salario \*1.21 where no\_empleado=1; |
| 53 |  | CREATE TRIGGER `oficios`  BEFORE INSERT ON `empleado`  FOR EACH ROW  BEGIN  set @oficioCount = (select count(\*) from empleado where oficio = new.oficio);  if @oficioCount>=3 then  Signal sqlstate '02000' set message\_text ='No puede haber mas de 3 empleados con el mismo oficio';  end if;  END  insert into empleado values (7,'Juan',34,'Vendedor','Universidad','1970-02-13',10000,0,20); |
| 54 |  | CREATE TRIGGER `fecha\_baja`  BEFORE DELETE ON `empleado`  FOR EACH ROW  BEGIN set @dias = (select datediff(now(),fecha\_alta) from empleado where no\_empleado = old.no\_empleado);  if @dias<60 THEN  SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'NO ALTA PARA IMSS';  end if;  END  insert into empleado values (6,'Juan',15,'Contador','Universidad',now(),12000,0,20);  delete from empleado where no\_empleado = 6; |
| 55 |  | create trigger salario\_tope  before insert on empleado  for each row  begin  set @salario = (select sum(salario) from empleado where no\_depto=new.no\_depto);  if @salario + new.salario > 50000 then  Signal sqlstate '02000' set message\_text ='No puede haber un salario superior a 50000 entre todos los empleados';  end if;  end  CREATE TRIGGER `salario\_tope1`  BEFORE UPDATE ON `empleado`  FOR EACH ROW  begin set @salario = (select sum(salario) from empleado where no\_depto=new.no\_depto);  if (@salario-old.salario + new.salario) > 50000 then  Signal sqlstate '02000' set message\_text ='No puede haber un salario superior a 50000 entre todos los empleados';  end if;  END |
| 56 |  | CREATE TRIGGER `validar\_saldo`  BEFORE UPDATE ON `cuenta`  FOR EACH ROW  BEGIN  set @saldo = (select saldo from cuenta where no\_cliente = old.no\_cliente or no\_cuenta=old.no\_cuenta);  if (new.saldo <0 ) THEN  Signal sqlstate '02000' set message\_text ='Saldo negativo, es insuficiente';  elseif (@saldo - (-1\*new.saldo) <0 ) THEN  Signal sqlstate '02000' set message\_text ='Saldo negativo, es insuficiente';  end if;  END |
| 57 |  | CREATE TRIGGER `tipo\_cuenta`  BEFORE INSERT ON `cuenta`  FOR EACH ROW  BEGIN  set @totalCuentas = (select count(\*) from cuenta where tipo\_cuenta = new.tipo\_cuenta and no\_cliente = new.no\_cliente);  if (@totalCuentas > 0) then  Signal sqlstate '02000' set message\_text ='No puede haber un tipo de cuenta repetido para este cliente';  end if;  END |
| 58 |  | CREATE TRIGGER `tope\_tipo\_cuenta`  BEFORE UPDATE ON `cuenta`  FOR EACH ROW  BEGIN  SET @saldo = (select saldo from cuenta where no\_cuenta = old.no\_cuenta or no\_cliente = old.no\_cliente);  SET @message = CONCAT('Pronto llegaras al tope, te sugerimos un abrir una cuenta del tipo: ',old.tipo\_cuenta);  SET @messageMax = CONCAT('La cuenta de tipo ',old.tipo\_cuenta,' llego a su tope maximo');  if old.tipo\_cuenta = 'Nomina' THEN  if new.saldo >= 50000 THEN  Signal sqlstate '02000' set message\_text = @messageMax;  elseif @saldo + new.saldo >= 50000\*0.80 THEN  Signal sqlstate '02000' set message\_text = @message;  end if;  elseif old.tipo\_cuenta = 'Ahorro' THEN  if new.saldo >= 500000 THEN  Signal sqlstate '02000' set message\_text = @messageMax;  elseif @saldo + new.saldo >= 500000\*0.80 THEN  Signal sqlstate '02000' set message\_text = @message;  end if;  end if;  END |
| 59 |  | CREATE TRIGGER `autoriza\_credito`  AFTER UPDATE ON `cuenta`  FOR EACH ROW BEGIN  SET @monto = new.saldo;  IF @monto < 5000 THEN  Signal sqlstate '02000' set message\_text ='No hay credito';  ELSEIF @monto >=5000 and @monto <=10000 THEN  Signal sqlstate '02000' set message\_text ='Tienes 10% de Credito';  ELSEIF @monto >=10000 and @monto <=20000 THEN  Signal sqlstate '02000' set message\_text ='Tienes 20% de Credito';  ELSE  Signal sqlstate '02000' set message\_text ='Tienes 30% de Credito';  END IF;  END |
| 60 |  | CREATE TRIGGER `bitacora\_auditoria\_cliente` AFTER INSERT ON `cliente`  FOR EACH ROW  BEGIN  INSERT INTO bitacora\_auditoria\_table VALUES (user(),'INSERT','Cliente',now());  END  CREATE TRIGGER `bitacora\_auditoria\_cliente\_update`  AFTER UPDATE ON `cliente`  FOR EACH ROW  BEGIN  INSERT INTO bitacora\_auditoria\_table VALUES (user(),'UPDATE','Cliente',now());  END  CREATE TRIGGER `bitacora\_auditoria\_cliente\_delete`  AFTER DELETE ON `cliente`  FOR EACH ROW  BEGIN  INSERT INTO bitacora\_auditoria\_table VALUES (user(),'DELETE','Cliente',now());  END  CREATE TRIGGER `bitacora\_auditoria\_cuenta` AFTER INSERT ON `cuenta`  FOR EACH ROW  BEGIN  INSERT INTO bitacora\_auditoria\_table VALUES (user(),'INSERT','Cuenta',now());  END  CREATE TRIGGER `bitacora\_auditoria\_cuenta\_update`  AFTER UPDATE ON `cuenta`  FOR EACH ROW  BEGIN  INSERT INTO bitacora\_auditoria\_table VALUES (user(),'UPDATE','Cuenta',now());  END  CREATE TRIGGER `bitacora\_auditoria\_cuenta\_delete`  AFTER DELETE ON `cuenta`  FOR EACH ROW  BEGIN  INSERT INTO bitacora\_auditoria\_table VALUES (user(),'DELETE','Cuenta',now());  END  CREATE TRIGGER `bitacora\_auditoria\_movimientos`  AFTER INSERT ON `movimientos`  FOR EACH ROW  BEGIN  INSERT INTO bitacora\_auditoria\_table VALUES (user(),'INSERT','Movimientos',now());  END  CREATE TRIGGER `bitacora\_auditoria\_movimientos\_update`  AFTER UPDATE ON `movimientos`  FOR EACH ROW  BEGIN  INSERT INTO bitacora\_auditoria\_table VALUES (user(),'UPDATE','Movimientos',now());  END  CREATE TRIGGER `bitacora\_auditoria\_movimientos\_delete`  AFTER DELETE ON `movimientos`  FOR EACH ROW  BEGIN  INSERT INTO bitacora\_auditoria\_table VALUES (user(),'DELETE','Movimientos',now());  END |