

Ad:Özgür Sadık Soyad:Utku Numara: 181816072 Lab Numarası:Lab Föy 5

1.a) CREATE TABLE Staff(

staffID INT GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT by 1),
staffName VARCHAR(20),
staffSurname VARCHAR(20),
DOB Date NOT NULL,
staffBranchNo,
salary DECIMAL(7,2),
startDateOfWork Date,
PRIMARY KEY (staffID),
FOREIGN KEY (staffBranchNo) REFERENCES Branch
);

b) CREATE TABLE Branch(

branchNo INT GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT by 1),
branchCity VARCHAR(20),
branchTotalStaffNumber INT,
PRIMARY KEY (branchNo)
);

c) CREATE TABLE Sales(

saleID INT GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT by 1),
orderID INT,
sellerID INT,
orderTotal DECIMAL(7,2),
PRIMARY KEY (saleID),
FOREIGN KEY (sellerID) REFERENCES LOGS
);

d) CREATE TABLE Logs(
logID INT GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT by 1),
logType VARCHAR(10) NOT NULL,
relatedTable VARCHAR(10) NOT NULL,
relatedID INT NOT NULL,
logTime TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (logID)
);

2) CREATE OR REPLACE Trigger Branch_Guncelleme AFTER INSERT or UPDATE or DELETE ON Staff

BEGIN

Update Branch Set branchTotalStaffNumber = (SELECT Count(staffBranchNo) FROM Staff WHERE staffBranchNo=branchNo GROUP BY staffBranchNo);

END;

3) CREATE OR REPLACE PROCEDURE maas_arttir (staff_ID INT,year INT)

AS

BEGIN

IF year<3 THEN

UPDATE Staff SET salary=salary+salary*0.04

WHERE staffID=staff_ID;

END IF;

IF year>=3 and year<6 THEN

UPDATE Staff SET salary=salary+salary*0.06

WHERE staffID=staff_ID;

END IF;

IF year>=6 THEN

UPDATE Staff SET salary=salary+salary*0.08

WHERE staffID=staff_ID;

END IF;

END;

4)

---STAFF TABLOSU İÇİN LOGLAMA---

CREATE OR REPLACE Trigger log_staff AFTER INSERT or UPDATE or DELETE on Staff

FOR EACH ROW

Begin

if Inserting Then

Insert into Logs(logType,relatedTable,relatedID) VALUES ('insert', 'staff', :new.staffId);

end if;

if Updating Then

Insert into Logs(logType,relatedTable,relatedID) VALUES ('update', 'staff', :new.staffId);

end if;

if Deleting Then

Insert into Logs(logType,relatedTable,relatedID) VALUES ('delete', 'staff', :new.staffId);

end if;

End;

---SALES TABLOSU İÇİN LOGLAMA---

CREATE OR REPLACE Trigger log_sales AFTER INSERT or UPDATE or DELETE on Sales

FOR EACH ROW

Begin

if Inserting Then

Insert into Logs(logType,relatedTable,relatedID) VALUES ('insert', 'sales', :new.saleID);

end if;

if Updating Then

Insert into Logs(logType,relatedTable,relatedID) VALUES ('update', 'sales', :new.saleID);

end if;

```
    if Deleting Then
        Insert into Logs(logType,relatedTable,relatedID) VALUES ('delete', 'sales',
:new.saleID);
    end if;
end;
```

---BRANCH TABLOSU İÇİN LOGLAMA---

CREATE OR REPLACE Trigger log_branch AFTER INSERT or UPDATE or DELETE on
Branch

FOR EACH ROW

Begin

```
    if Inserting Then
        Insert INTO Logs(logType,relatedTable,relatedID) VALUES ('insert', 'branch',
:new.branchNo);
    end if;
    if Updating Then
        Insert INTO Logs(logType,relatedTable,relatedID) VALUES ('update', 'branch',
:new.branchNo);
    end if;
    if Deleting Then
        Insert INTO Logs(logType,relatedTable,relatedID) VALUES ('delete', 'branch',
:new.branchNo);
    end if;
end;
```

5) REVOKE all on Logs FROM PUBLIC;