CS4350 Database Systems HW#5

Name: Loc Nguyen

1. Rewrite the trigger AvgNetWorthTrigger in the lecture notes “Constraints and Triggers” for delete and for insert.

Insert trigger as follows:

CREATE TRIGGER AvgNetWorthTrigger

AFTER INSERT ON MovieExec

REFERENCING NEW TABLE AS NewStuff

FOR EACH STATEMENT

WHEN (500000 > (SELECT AVG(netWorth) FROM MovieExec))

BEGIN

DELETE FROM MovieExec

WHERE (name, address, cer#, netWorth) IN NewStuff;

INSERT INTO MovieExec

(SELECT \* FROM NewStuff);

END;

Delete trigger as follows:

CREATE TRIGGER AvgNetWorthTrigger

AFTER DELETE ON MovieExec

REFERENCING OLD TABLE AS OldStuff

FOR EACH STATEMENT

WHEN (500000 > (SELECT AVG(netWorth) FROM MovieExec))

BEGIN

DELETE FROM MovieExec

WHERE (name, address, cer#, netWorth) IN OldStuff;

INSERT INTO MovieExec

(SELECT \* FROM OldStuff);

END;

1. Consider the database schema :

EMPLOYEE( EmployeeID, EmployeeName, SupervisorID, DepartmentID)

PROJECT (ProjectID, EmployeeID)

DEPARTMENT( Department ID, DepartmentName)

Write a row trigger after insert that allows inserting a new row into the table

PROJECT and if the EmployeeID is not exist in the table EMPOLOYEE , the trigger inserts the new EmployeeID into the table EMPLOYEE and sets the other attributes EmployeeName. SupervisorID, DepartmentID to NULL.

CREATE TRIGGER AddEmployee

AFTER INSERT ON PROJECT

REFERENCING NEW ROW AS NewEmployee

FOR EACH ROW

WHEN (NewEmployee.EmployeeID NOT IN EMPLOYEE)

INSERT INTO EMPLOYEE

VALUES(NewEmployee.ID, NULL, NULL, NULL);

CREATE TRIGGER trigger-name  
{BEFORE | AFTER}  
{INSERT | DELETE | UPDATE [ OF column-name-list ]}  
ON table-name  
[ REFERENCING [ OLD AS var-to-refer-to-old-tuple ]  
[ NEW AS var-to-refer-to-new-tuple ] ]  
[ OLD TABLE AS name-to-refer-to-old-table ] ]  
[ NEW TABLE AS name-to-refer-to-new-table ] ]  
[ FOR EACH { ROW | STATEMENT } ]  
[ WHEN (precondition) ]  
statement-list