CS 240
Tianyu Zhang
UID: 805332081

1.
a.
starburst:

```
CREATE RULE deletedept ON Department
WHEN DELETED
    THEN UPDATE Employee
    SET Department = null
    WHERE Department IN (SELECT Dept-No FROM DELETED)
```

Oracle:

```
CREATE TRIGGER deletedept
AFTER DELETE ON Department
FOR EACH ROW
     BEGIN
        UPDATE Employee
        SET Department = NULL
        WHERE Department = OLD.Dept-No
     END;
```

DB2:

```
CREATE TRIGGER deletedept
AFTER DELETE ON Department
FOR EACH ROW
UPDATE Employee
SET Department = NULL
WHERE Department = OLD.Dept-No
```

b.
starburst:

```
CREATE RULE deleteemp ON Department
WHEN DELETED
    THEN DELETE FROM Employee WHERE Department IN (SELECT Dept_No FROM DELETED)
```

Oracle:

```
CREATE TRIGGER deleteemp
AFTER DELETE ON Department
FOR EACH ROW
     BEGIN
        DELETE FROM Employee WHERE Department = OLD.Dept-No
     END;
```

DB2:

```
CREATE TRIGGER deleteemp
AFTER DELETE ON Department
```

```
FOR EACH ROW
DELETE FROM Employee WHEREDepartment = OLD.Dept-No
```

c.

starburst:

```
CREATE RULE equalsalary ON Employee
WHEN INSERTED,UPDATED(Salary)
THEN UPDATED Employee E_Employee
IF Salary > (SELECT E_Manager.salary FROM Employee E_Manager and Department D WHERE
  E_Manager.name = Department.manager AND E_Employee.Department =
Department.Dept_No)
THEN SET Salary = (SELECT E_Manager.salary FROM Employee E_Manager and Department
WHERE
  Employee.name = Department.manager AND E_Employee.Department = Department.Dept_No)
```

Oracle:

```
CREATE TRIGGER equalsalary
AFTER UPDATE OF Salary ON Employee E
FOR EACH ROW
DECLARE NUMBER Sal
    BEGIN
       SELECT E_Manager.Salary FROM Employee E_Manager, Department D WHERE
D.Manager = E_Manager.Name
       AND D.Dept-No = E.Department INTO Sal
       UPDATE Employee
       IF Salary > Sal
       THEN
         SET Salary = Sal
       ENDIF;
    END;
```

DB2:

```
CREATE TRIGGER equalsalary
AFTER UPDATE OF Salary ON Employee E_Employee
FOR EACH ROW
    BEGIN
        UPDATE Employee
        WHEN Salary > (SELECT E_Manager.Salary FROM Employee E_Manager,
                       Department D WHERE D.Manager = E_Manager.Name AND
                       D.Dept-No = E_Employee.Department)
        SET Salary = (SELECT E_Manager.Salary FROM Employee E_Manager,
                       Department D WHERE D.Manager = E_Manager.Name AND
                       D.Dept-No = E_Employee.Department)
    END;
```

2

a.

```
CREATE TRIGGER Stopreordering
```

```
BEFORE INSERT ON PendingOrders
FOR EACH ROW
    WHEN (EXISTS (SELECT * FROM PendingOrders WHERE Part = New.Part))
    SIGNAL SQLSTATE '45000' (‘pending order exists ’)

CREATE TRIGGER Reorder
AFTER UPDATE OF PartOnHand ON Inventory
FOR EACH ROW
WHEN (NEW.PartOnHand < NEW.ReorderPoint)
    BEGIN
       INSERT INTO PendingOrders
       VALUES (NEW.Part, NEW.ReorderQuantity, SYSDATE)
    END;
```

b.

```
CREATE TRIGGER HalfQuantity
BEFORE INSERT ON PendingOrders
FOR EACH ROW
UPDATE NEW
SET ReorderQuantity = 1/2 * NEW.ReorderQuantity
WHERE EXISTS (SELECT * FROM PendingOrders WHERE Part = NEW.Part)

CREATE TRIGGER Reorder
AFTER UPDATE OF PartOnHand ON Inventory
FOR EACH ROW
WHEN (NEW.PartOnHand < NEW.ReorderPoint)
    BEGIN
       INSERT INTO PendingOrders
       VALUES (NEW.Part, NEW.ReorderQuantity, SYSDATE)
    END;
```