

ITEM\_DESCRIPTION) INTO reports FROM Rental\_History  
 \_Num = customernum AND  
 g that there exist entries in rental\_history that are NOT problem rentals,  
 y those with entries in the problem\_description would be problem rentals.  
 not the case, the following line can be removed:  
 ascription IS NOT NULL AND  
 ate BETWEEN '2016-01-01' AND '2016-12-31';  
 s > 5 THEN  
 ntalrisk = 'High-Risk';  
 reports > 2 AND reports < 6 THEN  
 rentalrisk = 'Medium-Risk';  
 AT rentalrisk = 'Low-Risk';  
 D IF;  
 \$\$  
 MITER ;  
 ALL 'prRiskAssessment'('3',@risklevel);  
 SELECT @risklevel;  
 CALL 'prRiskAssessment'('1',@risklevel);  
 SELECT @risklevel;  
 #3 Modify the Equipment\_Type table to include a new Boolean attribute called  
 # Reorder\_necessary that will be set to true when an item needs to be reordered. The  
 # attribute should default to false.  
 # Since the default value may not be valid for all the data in your table, write a stored  
 # procedure to set the value of the new Reorder\_necessary attribute to its correct value.  
 # If the Inventory\_count is less than or equal to the Reorder\_qty, then set the  
 # Reorder\_necessary value to true.  
 # Otherwise, set the Reorder\_necessary value to false.  
 # Execute the procedure to reset the value.  
 ALTER TABLE Equipment\_Type  
 ADD COLUMN Reorder\_Necessary bool default false; #Because it is a boolean, values are 0 or 1  
 DELIMITER \$\$  
 CREATE PROCEDURE isReorderNecessary () #(IN rowID int)  
 BEGIN #We assume that Reorder\_qty is a trigger quantity and that equality triggers a reorder.  
 UPDATE Equipment\_Type SET Reorder\_Necessary = (Inventory\_count <= Reorder\_qty);  
 #To update a specific row add the following:  
 #WHERE Equipment\_type\_code = rowID AND Inventory\_count < Reorder\_qty;  
 END \$\$  
 DELIMITER ;  
 CALL 'isReorderNecessary';  
 SELECT \* FROM Equipment\_Type;  
 UPDATE Equipment\_Type SET Reorder\_qty = 2  
 WHERE Equipment\_type\_code = 1;  
 CALL 'isReorderNecessary';  
 SELECT \* FROM Equipment\_Type;