DELIMITER //

CREATE PROCEDURE end\_disaster(

IN dID INT

)

BEGIN

UPDATE Disasters

SET Active = 0

WHERE (DisasterID = dID);

END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE goods\_categories()

BEGIN

SELECT Category

FROM PossibleGoods

GROUP BY Category;

END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE make\_match(

IN requestID INT, IN offerID INT

)

BEGIN

SELECT @claimAmount := LEAST(o.QuantityNeeded - o.QuantityReceived, r.QuantityOffered - r.QuantityClaimed)

FROM Offers as o, Requests as r

WHERE o.OfferID = offerID AND r.RequestID = requestID;

UPDATE Offers

SET QuantityClaimed = QuantityClaimed + @claimAmount

WHERE OfferID = offerID;

UPDATE Requests

SET QuantityReceived = QuantityReceived + @claimAmount

WHERE RequestID = requestID;

INSERT INTO Matches (OfferID, RequestID, Status)

VALUES (requestID, offerID, "pending");

END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE requests\_for\_offer(

IN userID INT, IN disasterID INT, IN goodsID INT

)

BEGIN

SELECT RequestID, QuantityNeeded, QuantityReceived, DatePosted, Duration

FROM Requests

WHERE (userID != UserID) AND (QuantityNeeded < QuantityReceived) AND (DisasterID = disasterID) AND (GoodsID = goodsID) AND (DATE\_ADD(DatePosted, INTERVAL Duration DAY) > NOW());

END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE offer\_for\_request(

IN userID INT, IN disasterID INT, IN goodsID INT

)

BEGIN

SELECT OfferID, QuantityOffered, QuantityClaimed, DatePosted, Duration

FROM Requests

WHERE (userID != UserID) AND (QuantityClaimed < QuantityOffered) AND (DisasterID = disasterID) AND (GoodsID = goodsID) AND (DATE\_ADD(DatePosted, INTERVAL Duration DAY) > NOW());

END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE search\_goods(

IN searchTerm varchar(45)

)

BEGIN

SELECT \* FROM PossibleGoods

WHERE Name LIKE searchTerm

Order By Name;

END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE select\_disaster(

IN dID INT

)

BEGIN

SELECT \* FROM Disasters

WHERE (DisasterID = dID);

END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE select\_goods(

)

BEGIN

SELECT \* FROM PossibleGoods

Order By Name;

END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE short\_disasters(

IN activeCheck INT

)

BEGIN

SELECT DisasterID, Name, DisasterLocation, StartDate, Active FROM Disasters

WHERE (Active >= activeCheck);

END //

DELIMITER ;

**SAMPLE CODE TO RUN:**

We use a python class Connection to manage connections to the database. Externally, each stored procedure has an equivalent python method that calls it. Internally, it looks like:

cursor.callproc(procedure, params)

result = next(cursor.stored\_results())

return result.fetchall()

For example, to render the HTML for the disasters page:

disasters = connection.short\_disasters()

return render\_template('./disasters.html', disasters=disasters)