

MySQL code:

Tables:

```
mysql> CREATE TABLE relief_centers (
    ->     center_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     center_name VARCHAR(100),
    ->     location VARCHAR(100),
    ->     capacity INT
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE victims (
    ->     victim_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     victim_name VARCHAR(100),
    ->     age INT,
    ->     incident_type VARCHAR(100),
    ->     region VARCHAR(100),
    ->     center_id INT,
    ->     FOREIGN KEY (center_id) REFERENCES relief_centers(center_id)
    -> );

```

```
mysql> CREATE TABLE supplies (
    ->     supply_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     item_name VARCHAR(100),
    ->     quantity INT,
    ->     center_id INT,
    ->     FOREIGN KEY (center_id) REFERENCES relief_centers(center_id)
    -> );
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> CREATE TABLE donations (
    ->     donation_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     donor_name VARCHAR(100),
    ->     item_donated VARCHAR(100),
    ->     amount INT,
    ->     donation_date DATE
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE TABLE demands (
    ->     demand_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     region VARCHAR(100),
    ->     food_required INT,
    ->     water_required INT,
    ->     medical_kits_required INT
    -> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> CREATE TABLE feedback (
    ->     feedback_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     person_name VARCHAR(100),
    ->     message TEXT,
    ->     date_submitted DATE
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> DELIMITER ;
mysql> CREATE TABLE IF NOT EXISTS inventory_low (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     item_name VARCHAR(100),
    ->     center_id INT,
    ->     quantity INT,
    ->     status VARCHAR(50),
    ->     checked_at DATETIME
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> CREATE TABLE low_inventory_log (
    ->     log_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     supply_id INT,
    ->     item_name VARCHAR(100),
    ->     center_id INT,
    ->     quantity INT,
    ->     log_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql>
mysql> CREATE TABLE allocation_log (
    ->     log_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     region VARCHAR(100),
    ->     item_name VARCHAR(100),
    ->     quantity_allocated INT,
    ->     allocation_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE TABLE disaster_report_log (
    ->     report_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     region VARCHAR(100),
    ->     victims INT,
    ->     total_demand INT,
    ->     supplies_available INT,
    ->     report_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> INSERT INTO relief_centers (center_name, location, capacity) VALUES
-> ('Central Relief Camp', 'Delhi', 300),
-> ('North Flood Shelter', 'Punjab', 220),
-> ('South Quake Support Center', 'Gujarat', 280),
-> ('East Cyclone Relief Point', 'Odisha', 250),
-> ('West Landslide Camp', 'Himachal Pradesh', 150),
-> ('Metro Emergency Hub', 'Mumbai', 320),
-> ('Valley Relief Zone', 'Kashmir', 180),
-> ('Riverbank Evacuation Camp', 'Assam', 210),
-> ('Highway First-Aid Station', 'Rajasthan', 140),
-> ('Urban Multi-Relief Unit', 'Bangalore', 260);
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0   Warnings: 0
```

```
mysql> INSERT INTO victims (victim_name, age, incident_type, region, center_id) VALUES
-> ('Amit Kumar', 32, 'Flood', 'Punjab', 2),
-> ('Riya Sharma', 28, 'Earthquake', 'Gujarat', 3),
-> ('Karan Verma', 45, 'Storm', 'Delhi', 1),
-> ('Meena Devi', 38, 'Cyclone', 'Odisha', 4),
-> ('Sohan Lal', 50, 'Landslide', 'Himachal Pradesh', 5),
-> ('Priya Gupta', 22, 'Building Collapse', 'Delhi', 1),
-> ('Mohit Singh', 29, 'Flood', 'Assam', 8),
-> ('Sunita Reddy', 41, 'Heatwave', 'Rajasthan', 9),
-> ('Jatin Arora', 34, 'Earthquake', 'Gujarat', 3),
-> ('Rohit Das', 19, 'Cyclone', 'Odisha', 4),
-> ('Ananya Mehta', 26, 'Flood', 'Punjab', 2),
-> ('Lakshmi Nair', 31, 'Storm', 'Bangalore', 10),
-> ('Harpreet Kaur', 47, 'Flood', 'Punjab', 2),
-> ('Arjun Singh', 30, 'Landslide', 'Himachal Pradesh', 5),
-> ('Shweta Rawat', 27, 'Snowstorm', 'Kashmir', 7);
Query OK, 15 rows affected (0.02 sec)
Records: 15  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO supplies (item_name, quantity, center_id) VALUES
-> ('Food Packets', 120, 1),
-> ('Water Bottles', 250, 1),
-> ('Medical Kits', 70, 3),
-> ('Blankets', 90, 2),
-> ('Sanitary Pads', 150, 4),
-> ('Baby Food', 60, 1),
-> ('Torch Lights', 40, 5),
-> ('Tents', 25, 7),
-> ('Painkillers', 80, 3),
-> ('Dry Rations', 200, 2),
-> ('Winter Jackets', 50, 7),
-> ('ORS Packets', 120, 9),
-> ('Masks', 300, 10),
-> ('Gloves', 180, 8),
-> ('Energy Bars', 140, 4);
Query OK, 15 rows affected (0.01 sec)
Records: 15  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO donations (donor_name, item_donated, amount, donation_date) VALUES
-> ('Rahul Mehta', 'Food Packets', 40, '2025-11-10'),
-> ('Simran Kohli', 'Water Bottles', 30, '2025-11-12'),
-> ('Global Aid Org', 'Medical Kits', 20, '2025-11-08'),
-> ('Priya Foundation', 'Blankets', 50, '2025-11-14'),
-> ('Arun Kumar', 'Cash', 5000, '2025-11-11'),
-> ('Hope Trust', 'Tents', 10, '2025-11-09'),
-> ('Nisha Patel', 'Sanitary Kits', 35, '2025-11-13'),
-> ('Vikas Sharma', 'Dry Rations', 45, '2025-11-10'),
-> ('Rotary Club', 'Winter Jackets', 25, '2025-11-07'),
-> ('Ritu Arora', 'Baby Food', 15, '2025-11-12'),
-> ('SNS Group', 'Medicines', 80, '2025-11-05'),
-> ('Harsh Gupta', 'Energy Bars', 60, '2025-11-13'),
-> ('Metro Charitable Trust', 'ORS Packets', 90, '2025-11-14'),
-> ('Vcare NGO', 'Masks', 100, '2025-11-11'),
-> ('People United', 'Gloves', 70, '2025-11-10');
Query OK, 15 rows affected (0.00 sec)
Records: 15  Duplicates: 0  Warnings: 0
```

```

mysql> INSERT INTO demands (region, food_required, water_required, medical_kits_required) VALUES
-> ('Delhi', 60, 40, 20),
-> ('Punjab', 75, 55, 30),
-> ('Gujarat', 90, 45, 25),
-> ('Odisha', 110, 70, 50),
-> ('Himachal Pradesh', 45, 30, 15),
-> ('Assam', 80, 55, 35),
-> ('Rajasthan', 40, 20, 10),
-> ('Kashmir', 65, 40, 20),
-> ('Mumbai', 30, 15, 10),
-> ('Bangalore', 50, 25, 15),
-> ('Uttarakhand', 55, 35, 25),
-> ('Chhattisgarh', 70, 45, 30);
Query OK, 12 rows affected (0.01 sec)
Records: 12  Duplicates: 0  Warnings: 0

```

```

mysql> INSERT INTO feedback (person_name, message, date_submitted) VALUES
-> ('Ravi Singh', 'The relief center helped us greatly during the floods.', '2025-11-13'),
-> ('Meena Devi', 'Food distribution was well organized.', '2025-11-14'),
-> ('Amit Verma', 'Medical support was timely and effective.', '2025-11-12'),
-> ('Rohini Sharma', 'Volunteers were helpful and kind.', '2025-11-15'),
-> ('Suresh Thapa', 'Need more blankets at the center.', '2025-11-11'),
-> ('Neha Gupta', 'Clean drinking water was insufficient.', '2025-11-10'),
-> ('Pooja Sethi', 'Great coordination by the Delhi team.', '2025-11-09'),
-> ('Jaspreet Kaur', 'Happy with the assistance provided.', '2025-11-08'),
-> ('Zahid Khan', 'Transport arrangements needed improvement.', '2025-11-07'),
-> ('Anil Kumar', 'Appreciated the volunteers.', '2025-11-06'),
-> ('Shaila Rani', 'Medical team very supportive.', '2025-11-05'),
-> ('Rahul Sharma', 'Supplies were delivered on time.', '2025-11-04');
Query OK, 12 rows affected (0.02 sec)
Records: 12  Duplicates: 0  Warnings: 0

```

Triggers:

```

mysql> DELIMITER $$ 
mysql>
mysql> CREATE TRIGGER trg_low_inventory
-> AFTER UPDATE ON supplies
-> FOR EACH ROW
-> BEGIN
->     IF NEW.quantity < 30 THEN
->         INSERT INTO low_inventory_log (supply_id, item_name, center_id, quantity)
->             VALUES (NEW.supply_id, NEW.item_name, NEW.center_id, NEW.quantity);
->     END IF;
-> END$$
Query OK, 0 rows affected (0.03 sec)

```

```

mysql>
mysql> create trigger trg_supplies_check
-> before insert on supplies
-> for each row
-> begin
->     if new.quantity is null or new.quantity < 0 then
->         set new.quantity = 0;
->     end if;
-> end$$
Query OK, 0 rows affected (0.02 sec)

```

```
mysql> create trigger trg_supplies_low_update
-> after update on supplies
-> for each row
-> begin
->     if new.quantity < 30 then
->         insert into inventory_low(item_name, center_id, quantity, status, checked_at)
->             values (new.item_name, new.center_id, new.quantity, 'low stock', now());
->     end if;
-> end$$
Query OK, 0 rows affected (0.01 sec)
```

```
mysql>
mysql> create trigger trg_supplies_low_insert
-> after insert on supplies
-> for each row
-> begin
->     if new.quantity < 30 then
->         insert into inventory_low(item_name, center_id, quantity, status, checked_at)
->             values (new.item_name, new.center_id, new.quantity, 'low stock', now());
->     end if;
-> end$$
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> create trigger trg_victims_age_check
-> before insert on victims
-> for each row
-> begin
->     if new.age < 0 or new.age > 120 then
->         set new.age = null;
->     end if;
-> end$$
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> create trigger trg_feedback_check
-> before insert on feedback
-> for each row
-> begin
->     if new.message = '' then
->         set new.message = 'no feedback provided';
->     end if;
-> end$$
Query OK, 0 rows affected (0.02 sec)
```

Procedures:

```
mysql> DELIMITER $$  
mysql>  
mysql> CREATE PROCEDURE allocate_resources_weighted()  
--> BEGIN  
-->     DECLARE region_name VARCHAR(100);  
-->     DECLARE food INT;  
-->     DECLARE water INT;  
-->     DECLARE kits INT;  
-->     DECLARE demand_weight FLOAT;  
-->  
-->     DECLARE done INT DEFAULT 0;  
-->  
-->     DECLARE cur CURSOR FOR  
-->         SELECT region, food_required, water_required, medical_kits_required  
-->         FROM demands;  
-->  
-->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;  
-->  
-->     OPEN cur;  
-->  
-->     read_loop: LOOP  
-->  
-->         FETCH cur INTO region_name, food, water, kits;  
-->         IF done = 1 THEN  
-->             LEAVE read_loop;  
-->         END IF;  
-->  
-->         SET demand_weight = (0.5 * food) + (0.3 * water) + (0.2 * kits);  
-->  
-->         INSERT INTO allocation_log (region, item_name, quantity_allocated)  
-->             VALUES (region_name, 'Weighted Score', demand_weight);  
-->  
-->     END LOOP;  
-->  
-->     CLOSE cur;  
--> END$$  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DELIMITER $$  
mysql>  
mysql> CREATE PROCEDURE distribute_supplies()  
--> BEGIN  
-->     INSERT INTO allocation_log (region, item_name, quantity_allocated)  
-->         SELECT demands.region, supplies.item_name,  
-->             LEAST(supplies.quantity, demands.food_required)  
-->         FROM demands  
-->         JOIN supplies  
-->             ON supplies.center_id IN (SELECT center_id FROM relief_centers);  
--> END$$  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> DELIMITER $$  
mysql>  
mysql> CREATE PROCEDURE get_center_utilization()  
--> BEGIN  
-->     SELECT rc.center_id, rc.center_name, rc.capacity,  
-->             COUNT(v.victim_id) AS victims_present,  
-->             ROUND((COUNT(v.victim_id) / rc.capacity) * 100, 2) AS utilization_percent  
-->     FROM relief_centers rc  
-->     LEFT JOIN victims v ON rc.center_id = v.center_id  
-->     GROUP BY rc.center_id;  
--> END$$  
Query OK, 0 rows affected (0.01 sec)
```

```

mysql> CREATE PROCEDURE check_low_inventory()
-> BEGIN
->     -- clear old alerts
->     DELETE FROM inventory_low;
->
->     -- insert new low-stock items
->     INSERT INTO inventory_low (item_name, center_id, quantity, status, checked_at)
->     SELECT item_name, center_id, quantity, 'LOW STOCK', NOW()
->     FROM supplies
->     WHERE quantity < 30;
-> END$$
Query OK, 0 rows affected (0.02 sec)

```

```

mysql> DELIMITER $$

mysql>
mysql> CREATE PROCEDURE generate_disaster_report()
-> BEGIN
->     INSERT INTO disaster_report_log (region, victims, total_demand, supplies_available)
->     SELECT
->         d.region,
->         COUNT(v.victim_id),
->         (SUM(d.food_required) + SUM(d.water_required) + SUM(d.medical_kits_required))
->
->         IFNULL(SUM(s.quantity), 0)
->     FROM demands d
->     LEFT JOIN victims v ON d.region = v.region
->     LEFT JOIN relief_centers rc ON v.center_id = rc.center_id
->     LEFT JOIN supplies s ON rc.center_id = s.center_id
->     GROUP BY d.region;
-> END$$
Query OK, 0 rows affected (0.01 sec)

```

Tables:

```

mysql> select * from relief_centers;
+-----+-----+-----+-----+
| center_id | center_name | location | capacity |
+-----+-----+-----+-----+
| 1 | Central Relief Camp | Delhi | 300 |
| 2 | North Flood Shelter | Punjab | 220 |
| 3 | South Quake Support Center | Gujarat | 280 |
| 4 | East Cyclone Relief Point | Odisha | 250 |
| 5 | West Landslide Camp | Himachal Pradesh | 150 |
| 6 | Metro Emergency Hub | Mumbai | 320 |
| 7 | Valley Relief Zone | Kashmir | 180 |
| 8 | Riverbank Evacuation Camp | Assam | 210 |
| 9 | Highway First-Aid Station | Rajasthan | 140 |
| 10 | Urban Multi-Relief Unit | Bangalore | 260 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

```

```
mysql> select * from victims;
+-----+-----+-----+-----+-----+-----+
| victim_id | victim_name | age | incident_type | region | center_id |
+-----+-----+-----+-----+-----+-----+
| 1 | Amit Kumar | 32 | Flood | Punjab | 2 |
| 2 | Riya Sharma | 28 | Earthquake | Gujarat | 3 |
| 3 | Karan Verma | 45 | Storm | Delhi | 1 |
| 4 | Meena Devi | 38 | Cyclone | Odisha | 4 |
| 5 | Sohan Lal | 50 | Landslide | Himachal Pradesh | 5 |
| 6 | Priya Gupta | 22 | Building Collapse | Delhi | 1 |
| 7 | Mohit Singh | 29 | Flood | Assam | 8 |
| 8 | Sunita Reddy | 41 | Heatwave | Rajasthan | 9 |
| 9 | Jatin Arora | 34 | Earthquake | Gujarat | 3 |
| 10 | Rohit Das | 19 | Cyclone | Odisha | 4 |
| 11 | Ananya Mehta | 26 | Flood | Punjab | 2 |
| 12 | Lakshmi Nair | 31 | Storm | Bangalore | 10 |
| 13 | Harpreet Kaur | 47 | Flood | Punjab | 2 |
| 14 | Arjun Singh | 30 | Landslide | Himachal Pradesh | 5 |
| 15 | Shweta Rawat | 27 | Snowstorm | Kashmir | 7 |
+-----+-----+-----+-----+-----+-----+
```

```
mysql> select * from supplies;
+-----+-----+-----+-----+
| supply_id | item_name | quantity | center_id |
+-----+-----+-----+-----+
| 1 | Food Packets | 120 | 1 |
| 2 | Water Bottles | 250 | 1 |
| 3 | Medical Kits | 70 | 3 |
| 4 | Blankets | 90 | 2 |
| 5 | Sanitary Pads | 150 | 4 |
| 6 | Baby Food | 60 | 1 |
| 7 | Torch Lights | 40 | 5 |
| 8 | Tents | 25 | 7 |
| 9 | Painkillers | 80 | 3 |
| 10 | Dry Rations | 200 | 2 |
| 11 | Winter Jackets | 50 | 7 |
| 12 | ORS Packets | 120 | 9 |
| 13 | Masks | 300 | 10 |
| 14 | Gloves | 180 | 8 |
| 15 | Energy Bars | 140 | 4 |
+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

```
mysql> select * from donations;
+-----+-----+-----+-----+-----+-----+
| donation_id | donor_name | item_donated | amount | donation_date |
+-----+-----+-----+-----+-----+-----+
| 1 | Rahul Mehta | Food Packets | 40 | 2025-11-10 |
| 2 | Simran Kohli | Water Bottles | 30 | 2025-11-12 |
| 3 | Global Aid Org | Medical Kits | 20 | 2025-11-08 |
| 4 | Priya Foundation | Blankets | 50 | 2025-11-14 |
| 5 | Arun Kumar | Cash | 5000 | 2025-11-11 |
| 6 | Hope Trust | Tents | 10 | 2025-11-09 |
| 7 | Nisha Patel | Sanitary Kits | 35 | 2025-11-13 |
| 8 | Vikas Sharma | Dry Rations | 45 | 2025-11-10 |
| 9 | Rotary Club | Winter Jackets | 25 | 2025-11-07 |
| 10 | Ritu Arora | Baby Food | 15 | 2025-11-12 |
| 11 | SNS Group | Medicines | 80 | 2025-11-05 |
| 12 | Harsh Gupta | Energy Bars | 60 | 2025-11-13 |
| 13 | Metro Charitable Trust | ORS Packets | 90 | 2025-11-14 |
| 14 | Vcare NGO | Masks | 100 | 2025-11-11 |
| 15 | People United | Gloves | 70 | 2025-11-10 |
+-----+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

```
mysql> select * from demands;
+-----+-----+-----+-----+-----+
| demand_id | region | food_required | water_required | medical_kits_required |
+-----+-----+-----+-----+-----+
| 1 | Delhi | 60 | 40 | 20 |
| 2 | Punjab | 75 | 55 | 30 |
| 3 | Gujarat | 90 | 45 | 25 |
| 4 | Odisha | 110 | 70 | 50 |
| 5 | Himachal Pradesh | 45 | 30 | 15 |
| 6 | Assam | 80 | 55 | 35 |
| 7 | Rajasthan | 40 | 20 | 10 |
| 8 | Kashmir | 65 | 40 | 20 |
| 9 | Mumbai | 30 | 15 | 10 |
| 10 | Bangalore | 50 | 25 | 15 |
| 11 | Uttarakhand | 55 | 35 | 25 |
| 12 | Chhattisgarh | 70 | 45 | 30 |
+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

```
mysql> select * from feedback;
+-----+-----+-----+-----+
| feedback_id | person_name | message | date_submitted |
+-----+-----+-----+-----+
| 1 | Ravi Singh | The relief center helped us greatly during the floods. | 2025-11-13 |
| 2 | Meena Devi | Food distribution was well organized. | 2025-11-14 |
| 3 | Amit Verma | Medical support was timely and effective. | 2025-11-12 |
| 4 | Rohini Sharma | Volunteers were helpful and kind. | 2025-11-15 |
| 5 | Suresh Thapa | Need more blankets at the center. | 2025-11-11 |
| 6 | Neha Gupta | Clean drinking water was insufficient. | 2025-11-10 |
| 7 | Pooja Sethi | Great coordination by the Delhi team. | 2025-11-09 |
| 8 | Jaspreet Kaur | Happy with the assistance provided. | 2025-11-08 |
| 9 | Zahid Khan | Transport arrangements needed improvement. | 2025-11-07 |
| 10 | Anil Kumar | Appreciated the volunteers. | 2025-11-06 |
| 11 | Shaila Rani | Medical team very supportive. | 2025-11-05 |
| 12 | Rahul Sharma | Supplies were delivered on time. | 2025-11-04 |
+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```