2018

# Project Report: Database Management

## DISASTER MANAGEMENT SYSTEM

JOHN ONGKO, NEETHU NARAYANAN, ANAND MURALEEDHARAN

|NAVEEN JINDAL SCHOOL OF MANAGEMENT | UNIVERSITY OF TEXAS AT DALLAS

# TABLE OF CONTENTS


Phenomenon ..... 4

Information Systems Scope ..... 4

Users ..... 4

Business Processes ..... 5

Business Requirements ..... 6

Data Requirements ..... 7

Conceptual Modelling ..... 9

Entity Relationship Diagram ..... 10

Relationship Diagram and Design Considerations ..... 11

- Relationship Diagram ..... 11
- Design Considerations ..... 11

Tables – Keys and Constraints ..... 12

Normalized Table Design – Table Definition ..... 13

- Climate_Forecast Table ..... 13
- Disaster_Alert Table ..... 13
- Disaster_Expenditure Table ..... 13
- Disaster_Funding Table ..... 13
- Disaster_Volunteer Table ..... 14
- Disasters Table ..... 14
- Disasters_Prediction Table ..... 14
- Volunteer_Organization Table ..... 14
- Volunteer_Schedule Table ..... 15
- Volunteers Table ..... 15

Queries ..... 16

- 1. Volunteers Serving More Than 2 Hours ..... 16
  - Query ..... 16
  - Query Result ..... 16
- 2. Total Fund for each Disaster ..... 16
  - Query ..... 16
  - Query Result ..... 17
- 3. Organization with maximum volunteers serving ..... 17
  - Query ..... 17

Query Result ..... 17
4. Climate Forecast and Disaster Prediction ..... 18
QUery ..... 18
Parameter Insertion ..... 18
Query Result ..... 18
5. Fund and Expenditure Report ..... 18
Query ..... 18
Query Result ..... 19
6. Volunteers for each Disaster ..... 19
Query ..... 19
Parameter Insertion ..... 19
Query Result ..... 19
7. Predicted to Real ..... 20
Query ..... 20
Query Result ..... 20
8. Predicted Disasters with Severity ..... 20
Query ..... 20
Query Result ..... 20
9. Volunteer Count in Organizations ..... 21
Query ..... 21
Query Result ..... 21
10. Number of Alerts per Disaster ..... 21
Query ..... 21
Query Result ..... 21
Appendix – Journal of User Group Interaction ..... 22
Phase 1: Business Requirements and Data Requirements ..... 22
Review Request ..... 22
Review Comments ..... 22
Summary ..... 22
PHASE 2: Conceptual Design – ERD ..... 23
Review Request ..... 23
Review Comments ..... 23
Summary ..... 23

PHASE 3: Normalized Table Design ..... 24
Review Request ..... 24
Review Comments ..... 24
Summary ..... 24
PHASE 4: Queries and Results ..... 25
Review Request ..... 25
Review Comments ..... 25
Summary ..... 25
Tools Used ..... 26
References ..... 26

## PHENOMENON

Natural disasters that happen in Texas

- Tornado
- Hurricane
- Flooding
- Drought

## INFORMATION SYSTEMS SCOPE

- Disaster Management System: The system is aimed at supporting all the disaster relief activities that happens in the state of Texas.

## USERS

- Government Officials
- Local Volunteer Organizations
- Residents
- Volunteers

## BUSINESS PROCESSES

BP1. Predict Disasters and Alert People

- A1. Analyze Past Disasters and Predict potential disasters in Texas
- A2. Alert system Maintenance, Verification and Alerting

BP2. Manage Volunteers

- A1. Create list of volunteer organizations
- A2. Create guidelines and prep docs for volunteers

BP3. Manage Relief funds

- A1. Collect Relief Funds
- A2. Distribute Relief Funds

## BUSINESS REQUIREMENTS

BP1.A1 [Analyze Past Disasters and Predict potential disasters in Texas]

- Govt. User should be able to analyze the trends from historical data.
- Govt. User should be able to monitor the current climate forecasts.
- Govt. User should be able to view a dashboard with maps of affected areas.

BP1.A2 [Alert system Maintenance, Verification and Alerting People]

- Govt. User wants to broadcast alerts timely and effectively.
- Govt. should be able to test the Alert Broadcast systems.
- Residents wants to be alerted at least a week before the disaster.

BP2.A1 [Create list of volunteer organizations]

- Govt. User wants to allocate appropriate number of volunteers to match the severity of the disaster.
- Govt. Users need a mechanism/platform to vet each volunteer organization based on decided set of criteria.
- Volunteers should receive a certificate of gratefulness from the state of Texas.

BP2.A2 [Create guidelines and prep docs for volunteers]

- Volunteers want to monitor their schedule.
- Volunteers should be trained on basic disaster management skills and first aid skills.

BP3.A1 [Collect Relief Funds]

- Govt User needs to track the source of relief funds for future needs.
- Government should be able to have a detailed tally on the funding

BP3.A2 [Manage Relief funds]

- Govt. User needs to allocate an amount to the disaster areas according to the severity and affected population.
- Local Organizations should be able to provide funds for disasters.

## DATA REQUIREMENTS

BP1.A1 [Analyze Past Disasters and Predict potential disasters in Texas]

- R1. Database must contain the Disaster table which includes all the data of all the disasters that occurred in various federal states.
- R2. Database must contain the Climate_Forecast table which has the climate forecast details. User will be provided with the latest record as per the timestamp.
- R3. Disaster table must contain all the data with respect to the disaster along with the area details.

BP1.A2 [Alert system Maintenance, Verification and Alerting People]

- R4. Database must contain an Alert table which provides the system with alert timing as per the predicted disaster and disaster severity.
- R5. Database must contain an Alert_IND table which can consist only of 'Y/N/T' value. On 'T' value test alerts will send out.
- R6. Alert table contains the alert timings.

BP2.A1 [Create list of volunteer organizations]

- R7. Database must contain a Volunteer_Org table which stores the details of the Organization including number of volunteers. Disaster severity can be obtained from Disaster_Pred table.
- R8. Database must contain a Disaster_Volunteer table which has the number of hours served by a volunteer for each disaster.
- R9. Disaster_Volunteer table gives the detail of hours served by each volunteer.

BP2.A2 [Create guidelines and prep docs for volunteers]

- R10. Database must contain a Volunteer_Schedule table providing the schedule details for participating volunteers.
- R11. Volunteer table indicates whether the volunteer is trained on disaster management skills and first aid skills.

BP3.A1 [Collect Relief Funds]

- R12. Database must contain a Funding table which contains the details of relief funds obtained for every disaster.
- R13. Database must contain an Expenditure table which stores the details of the various expenditures along with the expenditure request ID and Disaster ID.

BP3.A2 [Manage Relief funds]

- R14. Disaster table contains the severity of each disaster.
- R15. Database must contain a Funding Table which should identify the funding type as provided by an organization.

## CONCEPTUAL MODELLING

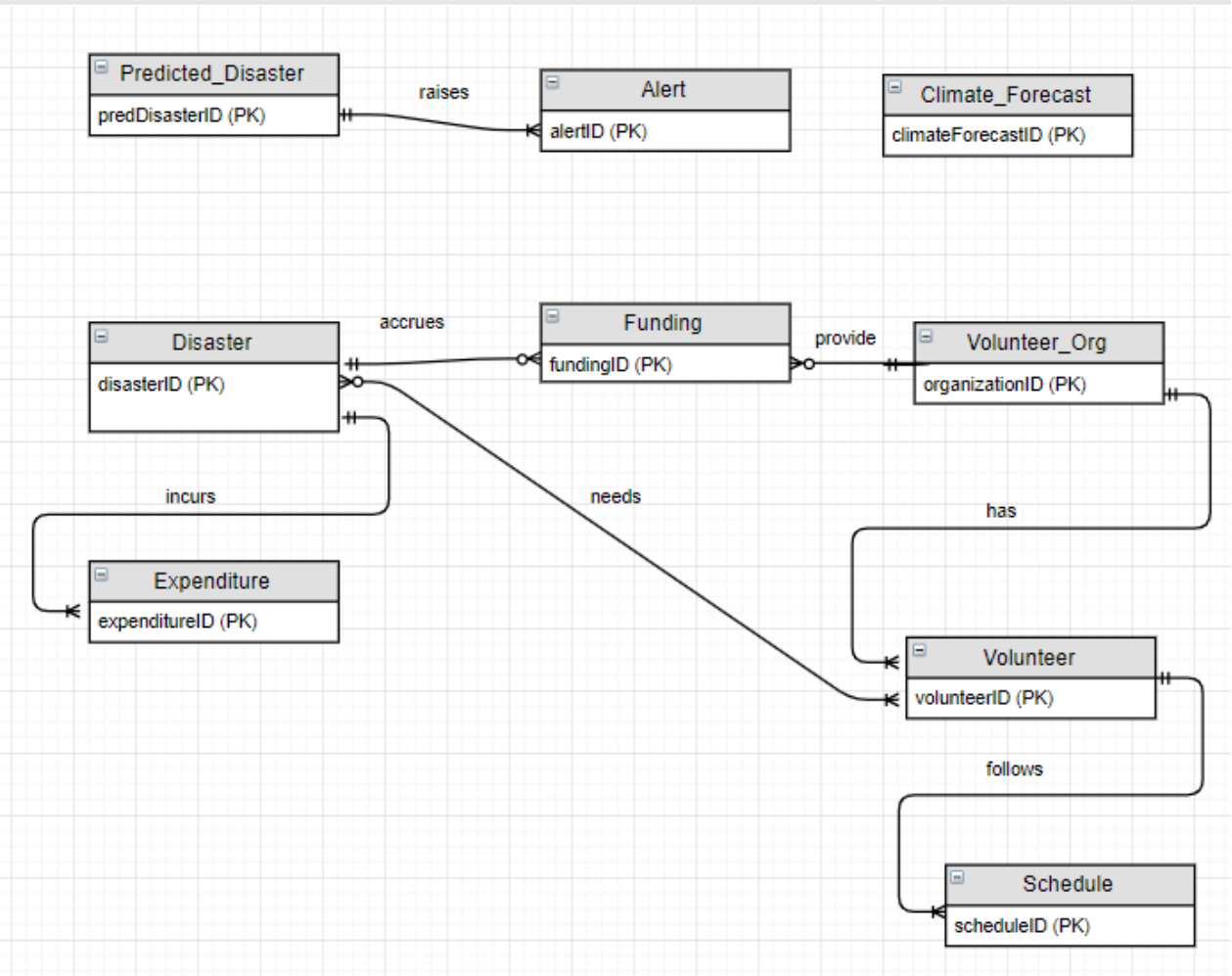
From the Data Requirements, the following entities can be identified.

- Predicted Disaster
- Alert
- Climate Forecast
- Disaster
- Funding
- Volunteer Organization
- Expenditure
- Volunteer
- Schedule

Each entity is related to each other as follows:

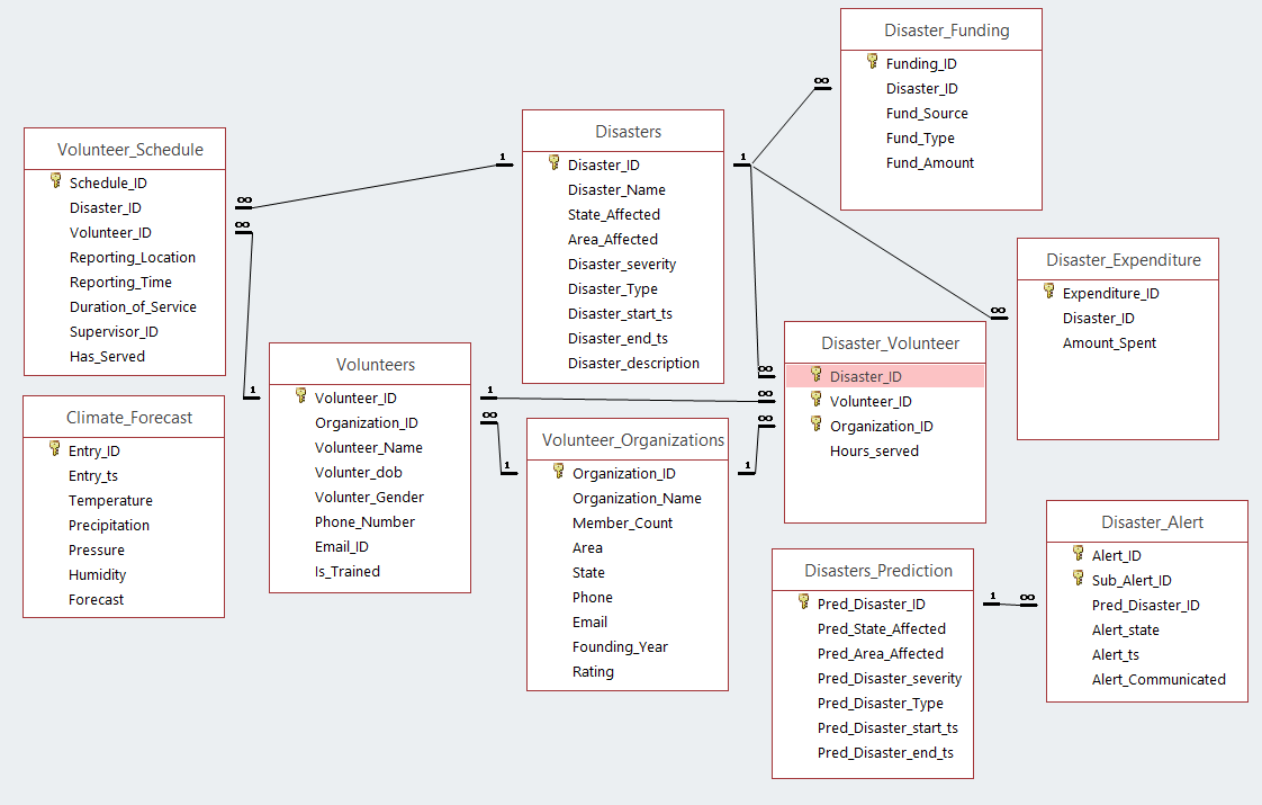
- A predicted disaster should lead to the **raising** of one/more alerts. Every alert corresponds to one and only one predicted disaster.
- Climate Forecast is an **independent** entity.
- Every disaster **accrues** one/more funds. Every fund is related to one disaster alone.
- Every Volunteer Organization may **provide** funding multiple times and each fund is related to one organization alone.
- Every disaster **needs** the service of multiple volunteers. A volunteer may serve during multiple disasters.
- Every volunteer **follows** one/more schedules to serve for a disaster. Every schedule is associated with a particular volunteer and disaster.
- Every disaster might **lead** to multiple expenditures. Every expenditure corresponds to a disaster.

## ENTITY RELATIONSHIP DIAGRAM



# RELATIONSHIP DIAGRAM AND DESIGN CONSIDERATIONS

## RELATIONSHIP DIAGRAM




## DESIGN CONSIDERATIONS

- Every volunteer is restricted to one organization by design.
- Climate Forecast is designed as an independent entity and will be queried using entry_ts parameters.
- Each disaster may have multiple alerts associated. So, Alert_ID represents an alert group that contains multiple sub alerts.
- The Volunteer training is depicted by the is_trained Boolean field in Volunteers tables.
- Volunteer and Disasters are in a many-to-many relationship. They are split by using an associative entity called Volunteer_Schedule.
- Volunteer_Schedule is a weak entity. The foreign keys Disaster_ID and Volunteer_ID forms the composite key. This composite key is substituted by a surrogate key, schedule_ID.
- The vetting of organization is based on the fields rating and Founding_Year of the Volunteer_Organizations table.
- Alert_state depicts the status of the alert broadcasting system.

## TABLES – KEYS AND CONSTRAINTS

| TABLE NAME | FIELD NAME | KEY/CONSTRAINT |
|---|---|---|
| Climate_Forecast | Entry_ID | Primary Key |
| Climate_Forecast | Entry_ts | Not Null, Unique |
| Disaster_Alert | Alert_ID | Primary Key |
| Disaster_Alert | Sub_alert_ID | Primary Key |
| Disaster_Alert | Pred_Disaster_ID | Foreign Key, Not Null |
| Disaster_Alert | Alert_ts | Not Null |
| Disaster_Expenditure | Expenditure_ID | Primary Key |
| Disaster_Expenditure | Disaster_ID | Foreign Key, Not Null |
| Disaster_Funding | Funding_ID | Primary Key |
| Disaster_Funding | Disaster_ID | Foreign Key, Not Null |
| Disaster_Volunteer | Disaster_ID | Primary Key |
| Disaster_Volunteer | Volunteer_ID | Primary Key |
| Disaster_Volunteer | Organization_ID | Primary Key |
| Disasters | Disaster_ID | Primary Key |
| Disasters | Disaster_name | Not Null |
| Disasters_prediction | Pred_disaster_ID | Primary Key |
| Volunteer_Organizations | Organization_ID | Primary Key |
| Volunteer_Organizations | Organization_name | Not Null |
| Volunteer_schedule | Schedule_ID | Primary Key |
| Volunteer_schedule | Disaster_ID | Foreign Key, Not Null |
| Volunteer_schedule | Volunteer_ID | Foreign Key, Not Null |
| Volunteers | Volunteer_ID | Primary Key |
| Volunteers | Organization_ID | Foreign Key, Not Null |
| Volunteers | Volunteer_name | Not Null |

## NORMALIZED TABLE DESIGN – TABLE DEFINITION

### CLIMATE_FORECAST TABLE

| Field Name | Data Type | Description (Optional) |
|---|---|---|
| Entry_ID | AutoNumber | ID of the entry |
| Entry_ts | Long Text | Date and Time of the measurement |
| Temperature | Number | Temperature observed |
| Precipitation | Number | Precipitation observed |
| Pressure | Number | Pressure observed in inches |
| Humidity | Number | Humidity observed in percent |
| Forecast | Long Text | Forecast made with the observation set |

### DISASTER_ALERT TABLE

Disaster_Alert

| Field Name | Data Type | Description (Optional) |
|---|---|---|
| Alert_ID | AutoNumber | Alert ID |
| Sub_Alert_ID | Number | Sub Alert ID |
| Disaster_Pred_ID | Number | Disaster Prediction ID |
| Alert_state | Yes/No | State of the alert broadcast system |
| Alert_ts | Date/Time | Date/Time of the Alert |
| Alert_Communicated | Yes/No | Boolean flag whether the alert has been broadcasted |

### DISASTER_EXPENDITURE TABLE

Disaster_Expenditure

| Field Name | Data Type | Description (Optional) |
|---|---|---|
| Expenditure_ID | AutoNumber | ID of the Expenditure |
| Disaster_ID | Number | ID of the disaster for which the expenditure was made |
| Amount_Spent | Currency | Amount spent |

### DISASTER_FUNDING TABLE

Disaster_Funding

| Field Name | Data Type | Description (Optional) |
|---|---|---|
| Funding_ID | AutoNumber | ID of the received fund |
| Disaster_ID | Number | ID of the Disaster |
| Fund_Source | Short Text | Organization or Person who contributed the fund |
| Fund_Type | Short Text | Type of the fund |
| Fund_Amount | Currency | Amount Received |

## DISASTER_VOLUNTEER TABLE

**Disaster_Volunteer**

| Field Name | Data Type | Description (Optional) |
|---|---|---|
| Disaster_ID | Number | Disaster ID |
| Volunteer_ID | Number | Volunteer ID |
| Organization_ID | Number | Organization ID |
| Hours_served | Number | Hours served by a volunteer for a particular disaster |

## DISASTERS TABLE

**Disasters**

| Field Name | Data Type | Description (Optional) |
|---|---|---|
| Disaster_ID | AutoNumber | The Unique ID that determines a disaster |
| Disaster_Name | Short Text | The name of the disaster |
| State_Affected | Short Text | The state affected by the disaster |
| Area_Affected | Short Text | The area of the state affected by the disaster |
| Disaster_severity | Short Text | Severity of the disaster - High, Medium, Low |
| Disaster_Type | Short Text | Type of the Disaster - Flood, Drought... |
| Disaster_start_ts | Date/Time | Start Date/Time of the disaster |
| Disaster_end_ts | Date/Time | End Date/Time of the disaster |
| Disaster_description | Long Text | Details of the Disaster |

## DISASTERS_PREDICTION TABLE

**Disasters_Prediction**

| Field Name | Data Type | Description (Optional) |
|---|---|---|
| Pred_Disaster_ID | AutoNumber | The Unique ID that determines a predicted disaster |
| Pred_State_Affected | Short Text | The state affected by the disaster |
| Pred_Area_Affected | Short Text | The area of the state affected by the disaster |
| Pred_Disaster_severity | Short Text | Severity of the disaster - High, Medium, Low |
| Pred_Disaster_Type | Short Text | Type of the Disaster - Flood, Drought... |
| Pred_Disaster_start_ts | Date/Time | Start Date/Time of the disaster |
| Pred_Disaster_end_ts | Date/Time | End Date/Time of the disaster |

## VOLUNTEER_ORGANIZATION TABLE

**Volunteer_Organizations**

| Field Name | Data Type | Description (Optional) |
|---|---|---|
| Organization_ID | AutoNumber | ID of the Volunteer Organization |
| Organization_Name | Long Text | Name of the Volunteer Organization |
| Member_Count | Number | Number of Volunteers present in the organization |
| Area | Short Text | Area of the Volunteer Organization |
| State | Short Text | State of the Volunteer Organization |
| Phone | Large Number | Phone Number of the Volunteer Organization |
| Email | Short Text | Email ID of the Volunteer Organization |
| Founding_Year | Date/Time | Year in which the Organization was founded |
| Rating | Number | The rating of the organization based on previous disaster relief activities |

## VOLUNTEER_SCHEDULE TABLE

Volunteer_Schedule

| Field Name | Data Type | Description (Optional) |
|---|---|---|
| Schedule_ID | AutoNumber | Unique ID for the schedules |
| Disaster_ID | Number | ID of the Disaster |
| Volunteer_ID | Number | ID of the Volunteer |
| Reporting_Location | Short Text | Place where the volunteer has to report at |
| Reporting_Time | Date/Time | Time when the volunteer has to reach the reporting_location |
| Duration_of_Service | Number | Number of hours that volunteer has to serve for |
| Supervisor_ID | Number | ID of the supervising Volunteer |
| Has_Served | Short Text | The Boolean flag which tells us if the volunteer has served his term |

## VOLUNTEERS TABLE

Volunteers

| Field Name | Data Type | Description (Optional) |
|---|---|---|
| Volunteer_ID | AutoNumber | The unique identifier of volunteer |
| Organization_ID | Number | The Organization which the volunteer is registered with |
| Volunteer_Name | Short Text | Name of the Volunteer |
| Volunter_dob | Date/Time | Volunteer date of birth |
| Volunter_Gender | Short Text | Gender of the volunteer |
| Phone_Number | Large Number | Phone Number of the Volunteer |
| Email_ID | Short Text | Email ID of the Volunteer |
| Is_Trained | Yes/No | Boolean flag for training |

## QUERIES

### 1. VOLUNTEERS SERVING MORE THAN 2 HOURS

#### QUERY

```
SELECT dis.disaster_name,
           vol.volunteer_id,
           vol.volunteer_name,
           Sum(volsh.Duration_of_Service) AS total_hour
FROM Disasters AS dis,
         Volunteers AS vol,
         Volunteer_Schedule AS volsh
WHERE dis.disaster_id = volsh.disaster_id
    and vol.volunteer_id=volsh.volunteer_id
GROUP BY vol.volunteer_id, vol.volunteer_name, dis.disaster_name
HAVING (((Sum(volsh.Duration_of_Service))>2));
```

#### QUERY RESULT

| disaster_name | volunteer_id | volunteer_name | total_hour |
|---|---|---|---|
| Harvey | 1 | Tom Cruise | 5 |
| Harvey | 4 | Jack Black | 7 |
| El Nino | 5 | Rihanna Ella | 8 |
| El Nino | 6 | Beyonce Ring | 8 |
| El Nino | 7 | Hillary Trump | 7 |
| Twister | 9 | Kobe O'neal | 7 |
| Twister | 10 | Ellen Degeneres | 8 |

### 2. TOTAL FUND FOR EACH DISASTER

#### QUERY

```
SELECT dis.disaster_name,
           sum(dis_fund.fund_amount) AS total_amount
FROM disasters AS dis,
         disaster_funding AS dis_fund
WHERE dis.disaster_id = dis_fund.disaster_id
GROUP BY dis.disaster_name;
```

QUERY RESULT

| disaster_name | total_amoun |
|---|---|
| El Nino | $250,000.00 |
| Harvey | $400,000.00 |
| Twister | $300,000.00 |

## 3. ORGANIZATION WITH MAXIMUM VOLUNTEERS SERVING

QUERY

```
SELECT dis.disaster_name AS disaster,
          org.organization_name,
          count(volsh.volunteer_id) AS number_of_volunteers
FROM disasters AS dis,
         volunteer_organizations AS org,
         volunteer_schedule AS volsh,
         volunteers AS vol
WHERE dis.disaster_id = volsh.disaster_id
    and volsh.volunteer_id = vol.volunteer_id
    and vol.organization_id = org.organization_id
GROUP BY dis.disaster_name,
               org.organization_name;
```

QUERY RESULT

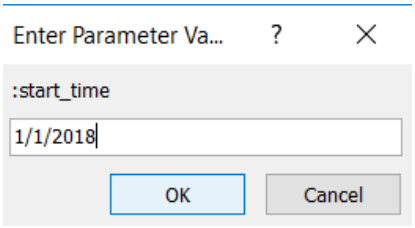
| disaster | organization_name | number_of_volunteers |
|---|---|---|
| El Nino | Austin Disaster Relief | 3 |
| Harvey | Texas Volunteers of Hous | 4 |
| Twister | St.Paul's church at Dallas | 3 |

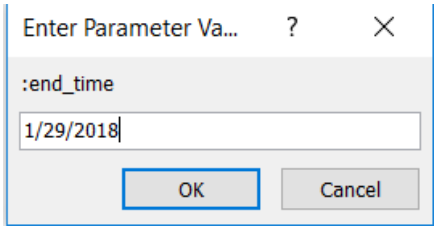
## 4. CLIMATE FORECAST AND DISASTER PREDICTION

### QUERY

```
SELECT cf.temperature,
       cf.precipitation,
       cf.pressure,
       cf.humidity,
       dispred.pred_disaster_type
FROM climate_forecast AS cf,
     disasters_prediction AS dispred
WHERE cf.entry_ts Between [:start_time] And [:end_time]
  And dispred.pred_disaster_start_ts>=[:start_time]
  And dispred.pred_disaster_end_ts<=[:end_time];
```

### PARAMETER INSERTION




### QUERY RESULT

| temperature | precipitation | pressure | humidity | pred_disaster_type |
|---|---|---|---|---|
| 68 | 34 | 29 | 30 | Hurricane |

## 5. FUND AND EXPENDITURE REPORT

### QUERY

```
SELECT dis.disaster_name,
       Sum(disfund.fund_amount) AS total_fund,
       Sum(disexp.amount_spent) AS total_expenditure,
       (Sum(disfund.fund_amount)-Sum(disexp.amount_spent)) AS balance_amount
FROM disasters AS dis,
     disaster_funding AS disfund,
     disaster_expenditure AS disexp
WHERE (((dis.disaster_id)=[disfund].[disaster_id]
  And (dis.disaster_id)=[disexp].[disaster_id]))
GROUP BY dis.disaster_name;
```

QUERY RESULT

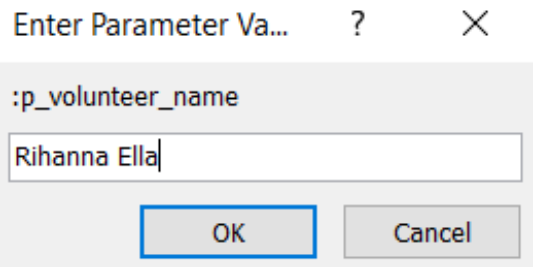
| disaster_name | total_fund | total_expend | balance_amount |
|---|---|---|---|
| El Nino | $250,000.00 | $780,000.00 | ($530,000.00) |
| Harvey | $400,000.00 | $1,005,000.00 | ($605,000.00) |
| Twister | $300,000.00 | $500,000.00 | ($200,000.00) |

## 6. VOLUNTEERS FOR EACH DISASTER

QUERY

```
SELECT dis.disaster_name,
          vol.volunteer_name,
          sum(volsh.duration_of_service) AS hours_served,
          switch(
                  sum(volsh.duration_of_service) = 0, "NO",
                  sum(volsh.duration_of_service) > 0, "YES"
                ) AS status
FROM disasters AS dis,
        volunteers AS vol,
        volunteer_schedule AS volsh
WHERE dis.disaster_id=volsh.disaster_id
    And vol.volunteer_id=volsh.volunteer_id
   And vol.volunteer_name=[:p_volunteer_name]
GROUP BY dis.disaster_name,
             vol.volunteer_name;
```

PARAMETER INSERTION




QUERY RESULT

| disaster_name | volunteer_name | hours_servec | status |
|---|---|---|---|
| El Nino | Rihanna Ella | 8 | YES |

## 7. PREDICTED TO REAL

QUERY

```
SELECT dis.disaster_name,
       dis.disaster_start_ts AS start_time,
       dis.disaster_end_ts AS end_time
FROM disasters AS dis,
       disasters_prediction AS dispred
WHERE dis.disaster_id = dispred.pred_disaster_id;
```

QUERY RESULT

| disaster_name | start_time | end_time |
|---|---|---|
| Twister | 4/20/2018 | 4/25/2018 |
| El Nino | 7/1/2014 | 9/25/2015 |
| Harvey | 9/14/2018 | 9/19/2018 |

## 8. PREDICTED DISASTERS WITH SEVERITY

QUERY

```
SELECT dispred.pred_disaster_id,
       dispred.pred_disaster_start_ts,
       dispred.pred_disaster_end_ts,
       dispred.pred_disaster_severity
FROM disasters_prediction AS dispred
GROUP BY dispred.pred_disaster_id,
         dispred.pred_disaster_start_ts,
         dispred.pred_disaster_end_ts,
         dispred.pred_disaster_severity;
```

QUERY RESULT

| pred_disaste | pred_disaster_start_ts | pred_disaster_end_ts | pred_disaster_sev |
|---|---|---|---|
| 1 | 7/1/2014 | 2/25/2015 | High |
| 2 | 1/21/2018 | 1/24/2018 | High |
| 3 | 9/10/2018 | 9/11/2018 | Low |

## 9. VOLUNTEER COUNT IN ORGANIZATIONS

QUERY

```
SELECT org.organization_name,
       org.member_count
FROM volunteer_organizations AS org;
```

QUERY RESULT

| organization_name | member_count |
|---|---|
| Texas Volunteers of Houston | 40 |
| Austin Disaster Relief | 36 |
| St.Paul's church at Dallas | 22 |

## 10. NUMBER OF ALERTS PER DISASTER

QUERY

```
SELECT alert.disaster_pred_id,
       count(alert.alert_id) AS Alert_count
FROM disaster_alert AS alert
GROUP BY alert.disaster_pred_id;
```

QUERY RESULT

| disaster_pred_id | Alert_count |
|---|---|
| 1 | 3 |

# APPENDIX – JOURNAL OF USER GROUP INTERACTION

## PHASE 1: BUSINESS REQUIREMENTS AND DATA REQUIREMENTS

### REVIEW REQUEST

Muraleedharan, Anand
Mon 26/11, 20:24
Mathur, Shruti; Rajput, Vishakha; Drozdowski, Kacper; Narayanan, Neethu; Ongko, John Paul

Project Write Up 2.1.docx
44 KB

Download Save to OneDrive - The University of Texas at Dallas

Hi Friends,

Welcome back to school after the Thanksgiving Break! Hope you all had a wonderful holiday week.

We have completed the first phase of the Database Management Project. Phase 1 includes:

1. Introduction - Phenomenon, Scope, Users
2. Business Processes with Activities
3. Business Requirements for each Activity
4. Data Requirement corresponding to Business requirements.

Please find the document attached. Please review and let us know how to improve.

Thanks and Regards,
Anand Muraleedharan 😊

### REVIEW COMMENTS

MS

Mathur, Shruti
Sat 01/12, 06:10

Hey,
Three of us have gone through the document that you have sent. It was well elaborated and documented no suggestion from our end, please go ahead with designing the ER diagram.

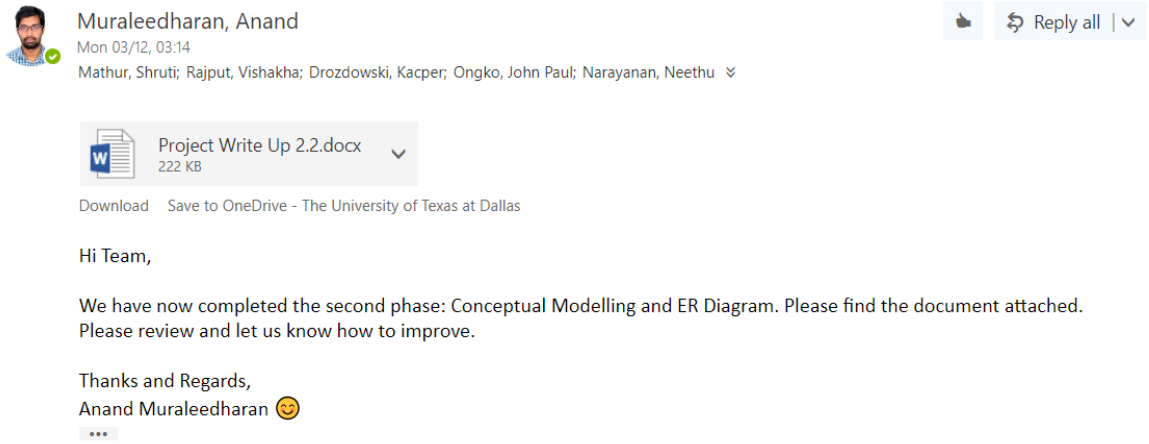
With Regards,
Shruti Mathur

### SUMMARY

User Group is happy with the Business Processes, Business Requirements and Data Requirements – No comments or Action Items.

## PHASE 2: CONCEPTUAL DESIGN – ERD

### REVIEW REQUEST



Muraleedharan, Anand
Mon 03/12, 03:14
Mathur, Shruti; Rajput, Vishakha; Drozdowski, Kacper; Ongko, John Paul; Narayanan, Neethu

Reply all

Project Write Up 2.2.docx
222 KB

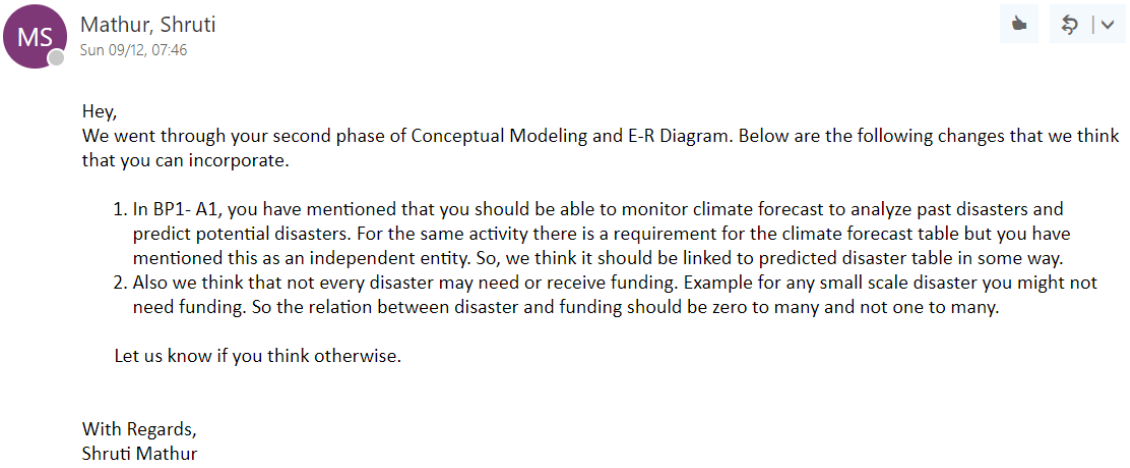
Download Save to OneDrive - The University of Texas at Dallas

Hi Team,

We have now completed the second phase: Conceptual Modelling and ER Diagram. Please find the document attached. Please review and let us know how to improve.

Thanks and Regards,
Anand Muraleedharan 😊

### REVIEW COMMENTS



Mathur, Shruti
Sun 09/12, 07:46

Hey,
We went through your second phase of Conceptual Modeling and E-R Diagram. Below are the following changes that we think that you can incorporate.

1. In BP1- A1, you have mentioned that you should be able to monitor climate forecast to analyze past disasters and predict potential disasters. For the same activity there is a requirement for the climate forecast table but you have mentioned this as an independent entity. So, we think it should be linked to predicted disaster table in some way.
2. Also we think that not every disaster may need or receive funding. Example for any small scale disaster you might not need funding. So the relation between disaster and funding should be zero to many and not one to many.

Let us know if you think otherwise.

With Regards,
Shruti Mathur

### SUMMARY

- **Relation between climate_forecast and disaster table:** We intend to keep it as an independent entity and connect it to predicted disaster using queries.
- **Relation between Disaster and Funding:** As suggested the minimum cardinality is adjusted to zero.

## PHASE 3: NORMALIZED TABLE DESIGN

### REVIEW REQUEST

Database Managment [MIS6326]: Review for Phase 1 of the Project

3

Muraleedharan, Anand
Today, 09:23
Mathur, Shruti; Drozdowski, Kacper; Rajput, Vishakha; Ongko, John Paul; Narayanan, Neethu

Reply all

Project Phase 3.pdf
885 KB

Download Save to OneDrive - The University of Texas at Dallas

Hi Team,

As far as the previous suggestions are concerned:

1. We choose to move forward with climate_forecast as an independent entity. In queries, we intend to connect them with predicted_disaster table by means of timestamp window matching.
2. We have adjusted the minimum cardinality of this relation.

Please find the project document till the 3rd phase attached. It includes the Normalized Table Structure, Design Considerations, and Relationship diagram. Please review and let us know.

Thanks and Regards,
Anand Muraleedharan 😊

### REVIEW COMMENTS

DK

Drozdowski, Kacper
Today, 09:42
Muraleedharan, Anand; Mathur, Shruti; Rajput, Vishakha; Ongko, John Paul; Narayanan, Neethu

Reply all

Hi Anand,
answering in the name of my group as they are currently travelling home for the break. I have the following questions:
1) In the Disaster_Alert table, you have Disaster_Pred_ID. Then, in Disaster_Prediction table you have Pred_Disaster_ID. Is it the same attribute? It either needs a name change if it is one attribute, or an explanation how it is different if it's two separate attributes.
2) In Disaster_Volunteer table, the description of Volunteer_ID is "Organization ID". I assume it's a misspelling, and a simple correction.
3) In the Volunteer_Schedule table, you have a "Reporting_location" attribute which tells where a volunteer should report before the action. Then, you have a "has_served" attribute, which should be filled out after the action to check if a volunteer served his term. Does it mean you will be coming to the table twice, first filling some of the data before the emergency, then completing it after the emergency?
it is probably correct, just a bit more work, but I think it might need a clarification.

We will be waiting for your part 4 - queries to complete the cooperation between our groups.

Best regards,
Kacper Drozdowski

### SUMMARY

- The mistakes pointed out in point 1 and point 2 are corrected.
- Volunteer Schedule table will be hit twice. Once to INSERT the record and then to UPDATE the 'has_served' attribute.

## PHASE 4: QUERIES AND RESULTS

### REVIEW REQUEST

Muraleedharan, Anand
Today, 01:36
Drozdowski, Kacper; Mathur, Shruti; Rajput, Vishakha; Ongko, John Paul; Narayanan, Neethu

Reply all

Phase 4.pdf
1 MB

Download Save to OneDrive - The University of Texas at Dallas

Hello Team,

Thanks for a prompt response. We have addressed issues 1 & 2. The 3rd one is designed as multiple hits on the database table. Once to insert a schedule record and then to update whether he turned up for the schedule.

Please find the 4th phase attached. It contains the queries mainly. Please review and let us know.

Thanks and Regards,
Anand Muraleedharan 😊

### REVIEW COMMENTS

DK

Drozdowski, Kacper
Today, 04:34

Hi Anand,
the queries look good. Just a few small comments:
1) Query 3 is called "Organization with maximum volunteers serving". Then the results are sorted: 3,4,3. I would sort it ascending by number of volunteers, not by organization name.
2) In Query 7, end time for Twister is earlier than start time. Is it intended, or is it an error in entered data?
3) In Query 8, I would add disaster name so someone who sees this table doesn't have to go to another table to find the disaster name.

Best regards

### SUMMARY

Muraleedharan, Anand
Today, 04:55
Drozdowski, Kacper; Rajput, Vishakha; Mathur, Shruti; Narayanan, Neethu; Ongko, John Paul

Reply all

Hi Kacper,

1. Each record shows the organization that sent the maximum number of volunteers to a particular disaster. Each record represents a particular disaster. So, sorting would not really make any sense.
2. This is a mistake in data population. we will correct that.
3. The query is about disaster prediction. So we name disasters only when they happen and not in the prediction phase. So they really do not have a name at that point in time.

Thanks a lot, team.

Thanks and Regards,
Anand Muraleedharan 😊

## TOOLS USED

- MS Access – Database
- Visual Paradigm – Functional Dependency diagram
- Draw.io – Entity Relationship Diagram

## REFERENCES


- Database Design, Application Development, and Administration, Eleventh Edition
- https://www.udemy.com/microsoftaccess/learn/v4/overview
- https://en.wikipedia.org/wiki/Category:Natural_disasters_in_Texas
- https://www.nbcdfw.com/weather/stories/Changing-Climate-Intensifying-Disasters-Texas-Report-502732902.html