

PHASE 3

Logical Data Model Design

In the design below, we clearly represent entities, their attributes, and relationships between them, including:

- **Entities and Attributes:** Each key entity has specific attributes (fields) and an identified primary key.
- **Relationships:** Define the logical connections between entities with foreign keys to maintain referential integrity.
- **Constraints:** Outline constraints like primary keys, foreign keys, and unique or not-null constraints where applicable.

2. Detailed Entity Descriptions

Based on Phase 1, here are the main entities and their attributes:

1. Disaster

- **Attributes:**
 - Disaster_ID (Primary Key)
 - Disaster_Type

- Date
- Magnitude
- Location_ID (Foreign Key referencing Location)
- **Constraints:**
 - Disaster_ID: Unique, Not Null
 - Location_ID: Foreign Key referencing Location(Location_ID)

CREATE TABLE Disaster (

Disaster_ID INT PRIMARY KEY,

Disaster_Type VARCHAR2(50) NOT NULL,

Disaster_Date DATE NOT NULL,

Magnitude NUMBER(5, 2),

Location_ID INT NOT NULL,

CONSTRAINT fk_Location_ID FOREIGN KEY (Location_ID) REFERENCES Location(Location_ID)

);

2. Location

- **Attributes:**

- Location_ID (Primary Key)
- Country
- State
- City

```
CREATE TABLE Location (
```

```
    Location_ID INT PRIMARY KEY,
```

```
    Country VARCHAR(50) NOT NULL,
```

```
    State VARCHAR(50),
```

City VARCHAR(50),

);

3. Prediction

- **Attributes:**

- Prediction_ID (Primary Key)
- Disaster_Type
- Predicted_Date
- Risk_Level
- Predicted_Location_ID (Foreign Key referencing Location)
- Prediction_Method

- **Constraints:**

- Predicted_Location_ID: Foreign Key referencing Location(Location_ID)

```
CREATE TABLE Prediction (  
  
    Prediction_ID INT PRIMARY KEY,  
  
    Disaster_Type VARCHAR2(50) NOT NULL,  
  
    Predicted_Date DATE NOT NULL,  
  
    Risk_Level VARCHAR2(20),  
  
    Predicted_Location_ID INT NOT NULL,  
  
    Prediction_Method VARCHAR2(100),  
  
    CONSTRAINT fk_Predicted_Location_ID FOREIGN KEY (Predicted_Location_ID) REFERENCES Location(Location_ID)  
  
);
```

4. **Weather_Condition**

- **Attributes:**

- Condition_ID (Primary Key)
- Location_ID (Foreign Key referencing Location)

- Temperature
- Rainfall
- Date
- **Constraints:**
 - Location_ID: Foreign Key referencing Location(Location_ID)

```
CREATE TABLE Weather_Condition (
```

```
    Condition_ID INT PRIMARY KEY,
```

```
    Location_ID INT NOT NULL,
```

```
    Temperature NUMBER(5, 2),
```

```
    Rainfall NUMBER(5, 2),
```

```
    Weather_Date DATE NOT NULL,
```

```
    CONSTRAINT fk_Weather_Location_ID FOREIGN KEY (Location_ID) REFERENCES Location(Location_ID)
```

```
);
```

5. Preparedness_Measure

- **Attributes:**

- Measure_ID (Primary Key)
- Disaster_ID (Foreign Key referencing Disaster)
- Measure_Description
- Measure_Type (e.g., Evacuation, Flood Barriers)
- Start_Date
- End_Date

- **Constraints:**

- Disaster_ID: Foreign Key referencing Disaster(Disaster_ID)

```
CREATE TABLE Preparedness_Measure (
```

```
Measure_ID INT PRIMARY KEY,
```

```
Disaster_ID INT NOT NULL,
```

```
Measure_Description VARCHAR2(4000), -- Use VARCHAR2 for text-like data
```

```
Measure_Type VARCHAR2(50),
```

Start_Date DATE,

End_Date DATE,

CONSTRAINT fk_Preparedness_Disaster_ID FOREIGN KEY (Disaster_ID) REFERENCES Disaster(Disaster_ID)

);

6. **Historical_Disaster_Data**

- **Attributes:**

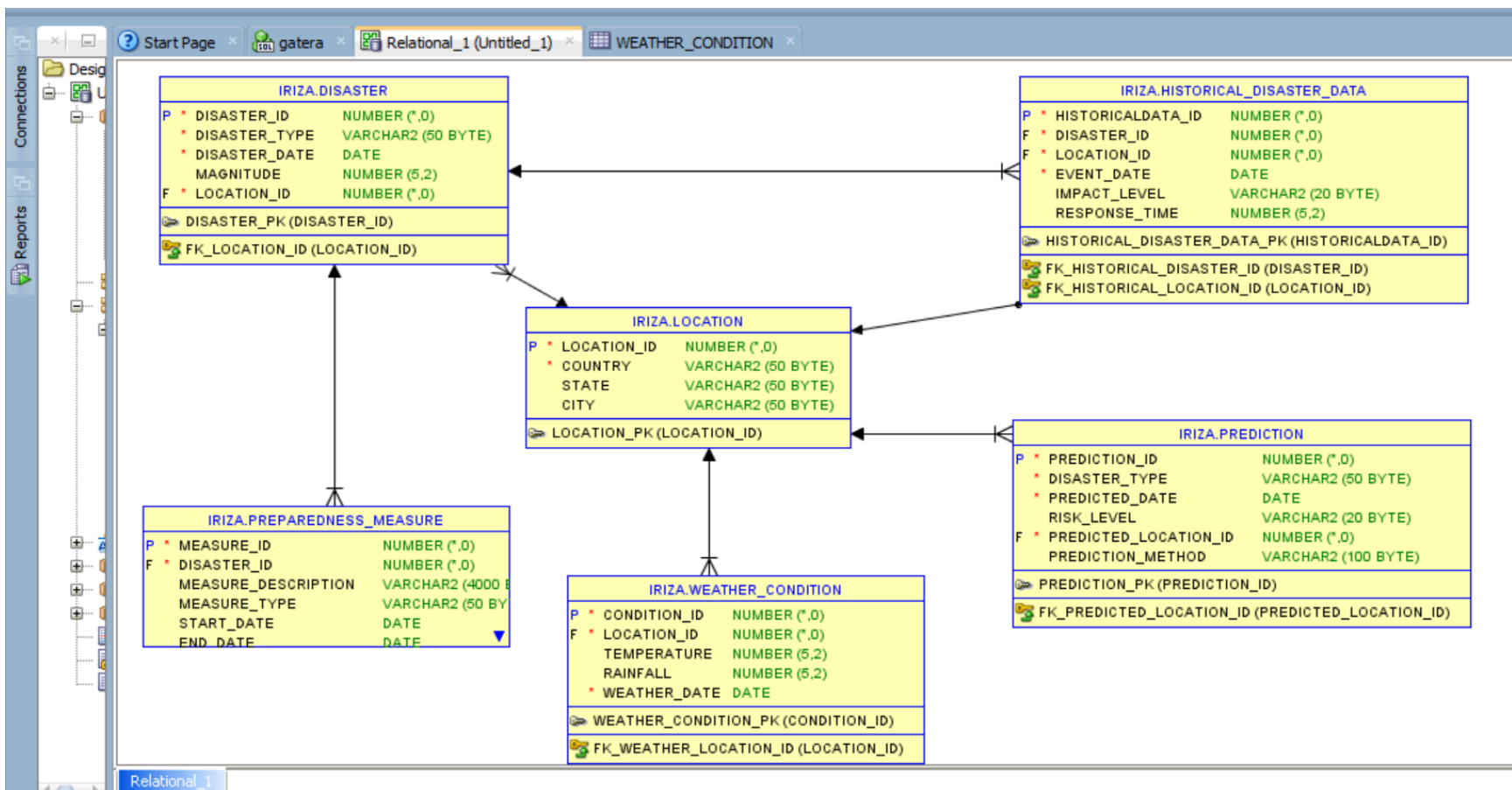
- HistoricalData_ID (Primary Key)
- Disaster_ID (Foreign Key referencing Disaster)
- Location_ID (Foreign Key referencing Location)
- Date
- Impact_Level (Scale of destruction)
- Response_Time

- **Constraints:**

- HistoricalData_ID: Unique, Not Null
- Disaster_ID: Foreign Key referencing Disaster(Disaster_ID)

- Location_ID: Foreign Key referencing Location(Location_ID)

```
CREATE TABLE Historical_Disaster_Data (  
  
    HistoricalData_ID INT PRIMARY KEY,  
  
    Disaster_ID INT NOT NULL,  
  
    Location_ID INT NOT NULL,  
  
    Date DATE NOT NULL,  
  
    Impact_Level VARCHAR(20),  
  
    Response_Time DECIMAL(5, 2),  
  
    CONSTRAINT HistoricalData_ID_Unique UNIQUE (HistoricalData_ID),  
  
    CONSTRAINT fk_Historical_Disaster_ID FOREIGN KEY (Disaster_ID) REFERENCES Disaster(Disaster_ID),  
  
    CONSTRAINT fk_Historical_Location_ID FOREIGN KEY (Location_ID) REFERENCES Location(Location_ID)  
  
);
```



3. Entity-Relationship Diagram (ERD)

1. **Disaster to Location:** Disaster is linked to Location with a one-to-many relationship, as multiple disasters can occur in the same location.
2. **Prediction to Location:** Prediction is linked to Location with a many-to-one relationship, as predictions are made for specific locations.
3. **Weather_Condition to Location:** Weather_Condition has a many-to-one relationship with Location, as weather data is specific to each location.
4. **Preparedness_Measure to Disaster:** Preparedness_Measure has a one-to-many relationship with Disaster, as multiple preparedness actions can be associated with one disaster.
5. **Historical_Disaster_Data to Disaster and Location:** Historical_Disaster_Data is linked to both Disaster and Location in a many-to-many relationship, as historical data could pertain to various disaster events across multiple locations.

4. Handling Data Scenarios

This logical model can handle scenarios such as:

- **Real-time disaster predictions** by storing recent weather data and linking predictions to specific locations.
- **Historical data tracking** by maintaining disaster records and past response effectiveness.

- **Decision-making support** by centralising data on locations, preparedness actions, and disaster severity for effective emergency planning.