

# INDICE

<b>1.</b>	<b>RACCOLTA E ANALISI DEI REQUISITI</b>	<b>2</b>
<b>2.</b>	<b>SCHEMA CONCETTUALE</b>	<b>4</b>
2.1.	SCELTE PROGETTUALI	4
<b>3.</b>	<b>SCHEMA CONCETTUALE RISTRUTTURATO</b>	<b>6</b>
3.1.	SCELTE PROGETTUALI	7
<b>4.</b>	<b>SCHEMA LOGICO</b>	<b>7</b>
4.1.	TRADUZIONE	7
<b>5.</b>	<b>PROGETTAZIONE PRATICA</b>	<b>9</b>
5.1.	CREAZIONE TABELLE	9
5.2.	QUERY DI INSERIMENTO	11
5.3.	QUERY DI SELEZIONE	12

# Raccolta e analisi dei requisiti

I dati relativi alle aree di interesse dovranno essere salvati nel database e comprendono:

- GeonameID
- Nome del paese in UNICODE
- Nome del paese in codice ASCII
- Codice dello stato
- Nome dello stato
- Latitudine
- Longitudine

Inoltre l'applicazione *ClimateMonitoring* prevede la registrazione di operatori climatici, i dati di questi ultimi dovranno essere salvati nell'applicazione e riguardano:

- UserID
- Nome
- Cognome
- Codice fiscale
- Email
- Password
- Nome del centro di appartenenza

Ogni operatore registrato correttamente può registrare un centro di monitoraggio oppure sceglierlo, i dati relativi ai centri sono:

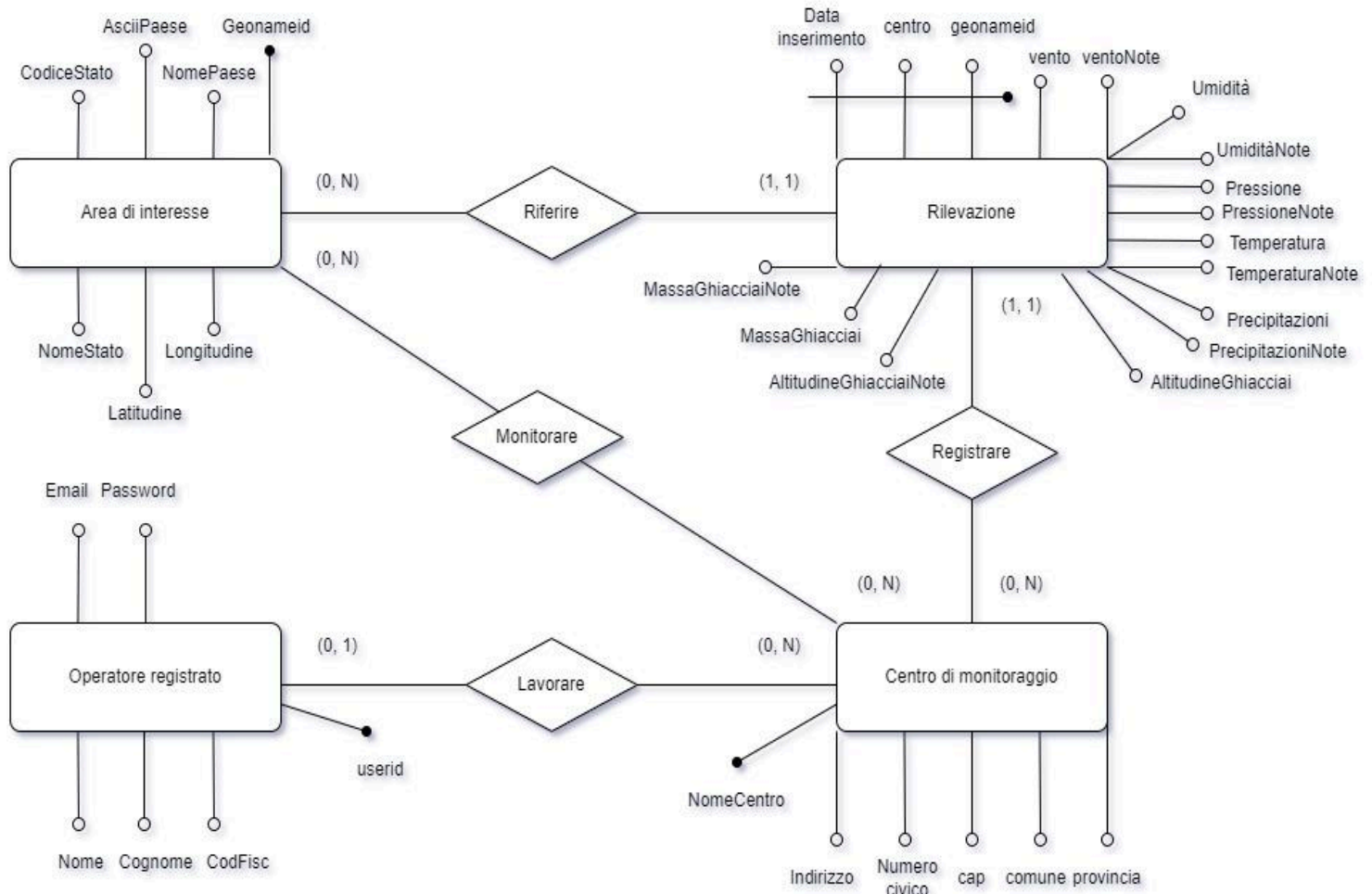
- Nome
- Indirizzo
- Numero civico
- CAP

- Comune
- Provincia

Infine ogni operatore con un centro di appartenenza può fare delle rilevazioni su delle aree di interesse monitorate dal centro, queste rilevazioni andranno salvate e dovranno contenere i seguenti dati:

- Nome del centro
- Area monitorata
- Data di inserimento
- Tutti i parametri climatici e le relative note:
  - Vento
  - Umidità
  - Pressione
  - Temperatura
  - Precipitazioni
  - Altitudine dei ghiacciai
  - Massa dei ghiacciai

# Schema concettuale



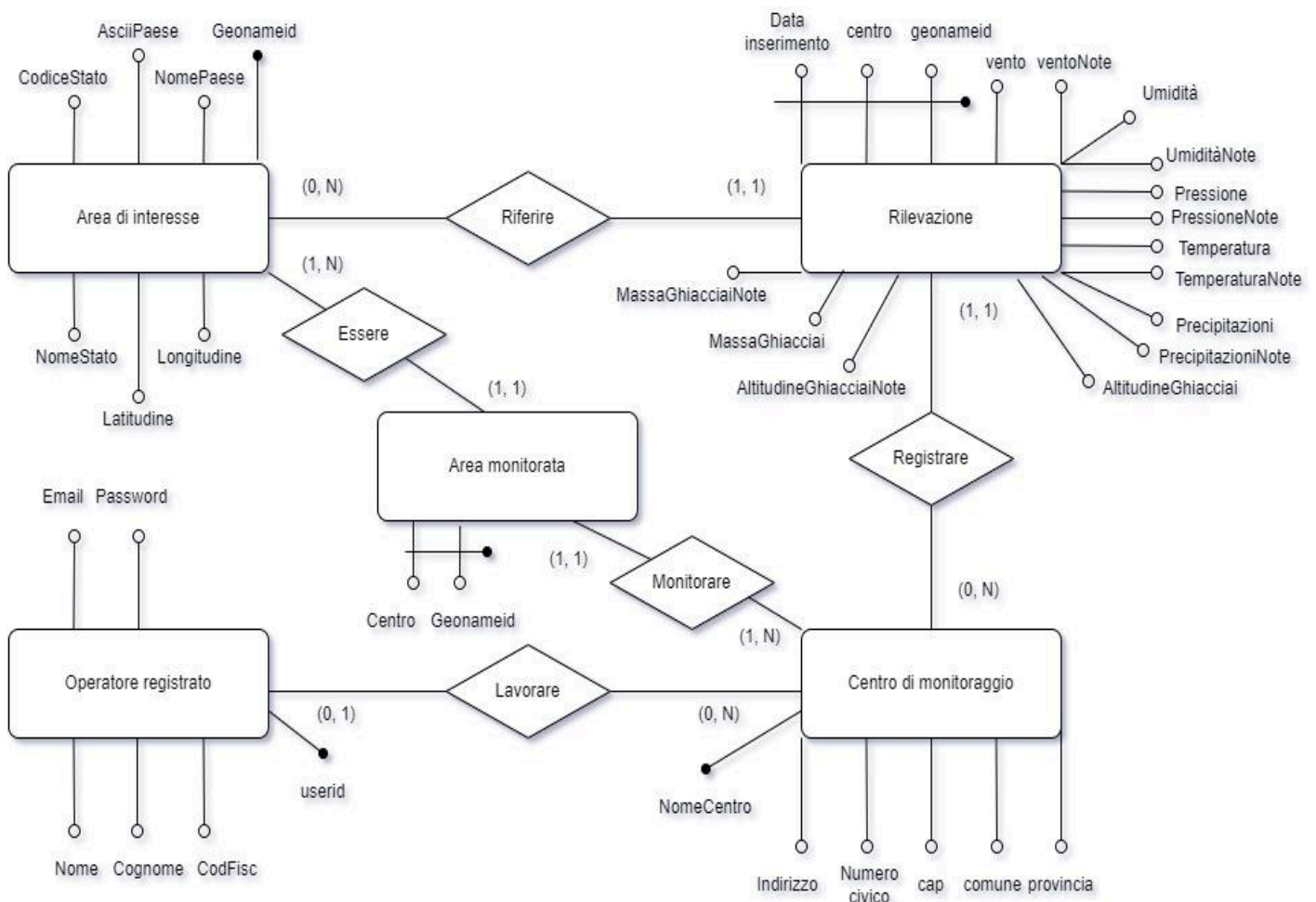
## Scelte progettuali

- Creazione dell'entità *Operatore registrato* con i relativi attributi che un operatore climatico deve poter avere. Come chiave primaria e quindi attributo univoco si utilizzano l'intero userID e come chiave esterna il nome del centro di appartenenza dell'operatore rappresentato dall'attributo *NomeCentro*.
- Creazione dell'entità *Centro di monitoraggio* che ha come attributi i dati di un centro di monitoraggio e come chiave primaria il nome univoco del centro identificato dall'attributo *NomeCentro*.

- Creazione dell'entità *Rilevazione* che ha una chiave primaria composta, infatti gli attributi *GeonameID*, *NomeCentro* e *DataInserimento* formano la *PRIMARY KEY*. Come chiavi esterne si utilizzano *GeonameID* e *NomeCentro*.
- Creazione dell'entità *CoordinateMonitoraggio* che ha come chiave primaria *GeonameID*, ovvero un codice univoco che rappresenta un'area di interesse.
- Inserire i vincoli *NOT NULL* per fare in modo che non vengano lasciati vuoti attributi importanti.
- Inserire i vincoli *ON UPDATE CASCADE* e *ON DELETE CASCADE*:
  - *ON UPDATE CASCADE*: questa opzione viene utilizzata per garantire che quando il valore di una colonna chiave primaria in una tabella viene aggiornato, anche i valori corrispondenti nelle colonne di chiave esterna nelle tabelle collegate vengano aggiornati automaticamente.
  - *ON DELETE CASCADE*: questa opzione viene utilizzata per garantire che quando una riga viene eliminata nella tabella con la chiave primaria, tutte le righe correlate nelle tabelle collegate vengano eliminate automaticamente.
- Si è scelto anche di mettere dei vincoli aggiuntivi come *CHECK(Latitudine >= -90 AND Latitudine <= 90)* per verificare che il valore della latitudine sia compreso in quel range e sia dunque un valore accettabile. Lo stesso ragionamento si può fare per la longitudine aggiungendo il seguente vincolo *CHECK(Longitudine >= -180 AND Longitudine <= 180)*. Infine per tutti i parametri climatici c'è un vincolo che implica l'inserimento di un valore maggiore o uguale a zero e minore o uguale a 5, ad esempio *CHECK(Vento >= 0 AND Vento <= 5)* per il vento.
- Le entità *Operatore registrato* e *Centro di monitoraggio* sono collegate tramite chiave esterna e rispettano il vincolo di *INTEGRITA' REFERENZIALE*. Lo

stesso vale per le coppie di entità *Centro di monitoraggio* e *Rilevazione*, per *Area di interesse* e *Rilevazione* e per *Area di interesse* e *Centro di monitoraggio*.

## Schema concettuale ristrutturato



## Scelte progettuali

- Si è scelto di sviluppare la relazione N:N tra le entità *Area di interesse* e l'entità *Centro di monitoraggio* per semplificare la gestione dei dati aggiungendo la nuova entità *Area monitorata* che contiene una chiave primaria ed esterna formata dalla coppia *GeonameID* e *Centro* che sono attributi chiave primaria di *Area di interesse* e *Centro di monitoraggio*.

## Schema logico

### Traduzione

Si traducono le relazioni al plurale come richiesto nelle linee guida del progetto. La traduzione avviene nel seguente modo:

- *Area di interesse* in *CoordinateMonitoraggio*
- *Operatore registrato* in *OperatoriRegistrati*
- *Centro di monitoraggio* in *CentriMonitoraggio*
- *Rilevazione* in *ParametriClimatici*
- *Area monitorata* in *AreeMonitorateDaCentri*

CoordinateMonitoraggio(

GeonameID,

NomePaese,

AsciiPaese,

CodiceStato,

NomeStato,

Latitudine,

Longitudine

)

OperatoriRegistrati(

UserID,

Nome,

Cognome,

CodFisc,

Email,

Password,

NomeCentro <sup>CentriMonitoraggio</sup>

)

CentriMonitoraggio(

NomeCentro,

Indirizzo,

NumeroCivico,

cap,

Comune,

Provincia

)

ParametriClimatici(

GeonameID <sup>CoordinateMonitoraggio</sup>,

NomeCentro <sup>CentriMonitoraggio</sup>,

DataInserimento,

Vento,

ventoNote,

Umidita,

UmiditaNote,

Pressione,

PressioneNote,

Temperatura,

TemperaturaNote,



```

        AltitudineGhiacciai,
        AltitudineGhiacciaiNote,
        MassaGhiacciai,
        MassaGhiacciaiNote
    )

AreeMonitorateDaCentri(
    CentriMonitoraggio
    NomeCentro ,
    CoordinateMonitoraggio
    GeonameID
)

```

# Progettazione pratica

## Creazione tabelle

### Area di interesse

```

CREATE TABLE CoordinateMonitoraggio (
    GeonameID char(10) PRIMARY KEY,
    NomePaese varchar(70) NOT NULL,
    AsciiPaese varchar(70) NOT NULL,
    CodiceStato char(2) NOT NULL,
    NomeStato varchar(70) NOT NULL,
    Latitudine float NOT NULL CHECK (Latitudine >= -90 AND Latitudine <= 90),
    Longitudine float NOT NULL CHECK (Longitudine >= -180 AND Longitudine <= 180)
);

```

### Operatore registrato

```

CREATE TABLE OperatoriRegistrati (
    UserID int PRIMARY KEY,
    Nome varchar(30) NOT NULL,

```

```

    Cognome varchar(30) NOT NULL,
    CodFisc char(16) NOT NULL,
    Email varchar(30) NOT NULL,
    Password varchar(60) NOT NULL,
    NomeCentro varchar(20) REFERENCES CentriMonitoraggio(NomeCentro)
    ON UPDATE CASCADE
    ON DELETE SET NULL
);

```

### **Centro di monitoraggio**

```

CREATE TABLE CentriMonitoraggio (
    NomeCentro varchar(20) PRIMARY KEY,
    Indirizzo varchar(50) NOT NULL,
    numeroCivico int NOT NULL,
    cap char(5) NOT NULL,
    Comune varchar(50) NOT NULL,
    Provincia char(2) NOT NULL
);

```

### **Rilvazione**

```

CREATE TABLE ParametriClimatici (
    GeonameID char(7) REFERENCES CoordinateMonitoraggio(GeonameID)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    NomeCentro varchar(20) REFERENCES CentriMonitoraggio(NomeCentro)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    DataInserimento timestamp NOT NULL,
    Vento int NOT NULL CHECK (Vento >= 0 AND Vento <= 5),
    VentoNote varchar(256),
    Umidita int NOT NULL CHECK (Umidita >= 0 AND Umidita <= 5),
    UmiditaNote varchar(256),

```

```

Pressione int NOT NULL CHECK (Pressione >= 0 AND Pressione <= 5),
PressioneNote varchar(256),
Temperatura int NOT NULL CHECK (Temperatura >= 0 AND Temperatura <= 5),
TemperaturaNote varchar(256),
Precipitazioni int NOT NULL CHECK (Precipitazioni >= 0 AND Precipitazioni <= 5),
PrecipitazioniNote varchar(256),
AltitudineGhiacciai int NOT NULL CHECK (AltitudineGhiacciai >= 0 AND AltitudineGhiacciai <= 5),
AltitudineGhiacciaiNote varchar(256),
MassaGhiacciai int NOT NULL CHECK (MassaGhiacciai >= 0 AND MassaGhiacciai <= 5),
MassaGhiacciaiNote varchar(256),
PRIMARY KEY (GeonameID, NomeCentro, DataInserimento)
);

```

### **Area monitorata**

```

CREATE TABLE AreeMonitorateDaCentri (
    NomeCentro varchar(20) REFERENCES CentriMonitoraggio(NomeCentro)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    GeonameID char(7) REFERENCES CoordinateMonitoraggio(GeonameID)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    PRIMARY KEY (NomeCentro, GeonameID)
);

```

## **Query di inserimento**

### **Aggiungi area di interesse**

```

INSERT INTO coordinateMonitoraggio (GeonameID, NomePaese, AsciiPaese, CodiceStato, NomeStato,
Latitudine, Longitudine);

```

### **Aggiungere operatore**

```
INSERT INTO OperatoriRegistrati (userid, nome, cognome, codfisc, email, password, nomecentro)
VALUES (?, ?, ?, ?, ?, ?, ?);
```

### **Aggiungere centro di monitoraggio**

```
INSERT INTO CentriMonitoraggio (nomecentro, indirizzo, numerocivico, cap, comune, provincia) VALUES
(?, ?, ?, ?, ?, ?);
```

### **Aggiungere una rilevazione**

```
INSERT INTO ParametriClimatici (geonameid,nomecentro, datainserimento, vento, ventonote, umidita,
umiditanote, pressione, pressionenote, temperatura, temperaturanote, precipitazioni, precipitazioninote,
altitudineghiacciai, altitudineghiacciainote, massaghiacciai, massaghiacciainote, ) VALUES (?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);
```

### **Aggiungere area monitorata da un centro**

```
INSERT INTO AreeMonitorateDaCentri (geonameid, nomecentro) VALUES (?, ?);
```

## **Query di selezione**

### **Ottenere tutti paesi**

```
SELECT *
FROM CoordinateMonitoraggio
```

### **Ottenere tutti paesi dato il nome**

```
SELECT *
FROM CoordinateMonitoraggio
WHERE asciipaese ILIKE ?
```

### **Ottenere tutti paesi dato il codice dello stato**

```
SELECT *
FROM CoordinateMonitoraggio
WHERE CodiceStato = ?
```

### **Ottenere tutti paesi dato il nome e il codice dello stato**

```
SELECT *  
FROM CoordinateMonitoraggio  
WHERE UPPER(asciipaese) = ? AND CodiceStato = ?
```

### **Ottenere un operatore dato l'userID**

```
SELECT *  
FROM OperatoriRegistrati  
WHERE userid = ?
```

### **Ottenere i punti di interesse associati ad un centro**

```
SELECT CoordinateMonitoraggio.*  
FROM CoordinateMonitoraggio NATURAL JOIN AreeMonitorateDaCentri NATURAL JOIN  
CentriMonitoraggio  
WHERE NomeCentro = ?
```

### **Aggiornare il centro di monitoraggio associato ad un operatore**

```
UPDATE OperatoriRegistrati  
SET NomeCentro = ?  
WHERE UserID = ?
```

### **Ottenere un paese associato a un centro di monitoraggio**

```
SELECT CoordinateMonitoraggio.*  
FROM AreeMonitorateDaCentri NATURAL JOIN CoordinateMonitoraggio  
WHERE AreeMonitorateDaCentri.NomeCentro = ? AND UPPER(CoordinateMonitoraggio.AsciiPaese) = ?  
AND CoordinateMonitoraggio.CodiceStato = ?
```

### **Ottenere i centri di monitoraggio**

```
SELECT *  
FROM CentriMonitoraggio
```

**Contare i centri di monitoraggio presenti con lo stesso nome (massimo 1)**

```
SELECT COUNT(*)  
FROM CentriMonitoraggio  
WHERE NomeCentro = ?
```

**Ottenere tutte le rilevazioni su un'area di interesse**

```
SELECT *  
FROM ParametriClimatici NATURAL JOIN AreeMonitorateDaCentri  
WHERE GeonameID = ?
```