

Założenia:

Kto jest klientem?

- **Urząd miasta**
 - **Rząd Państwowy**
-

Kto jest użytkownikiem?

- **User**
Osoba (najczęściej *migrant*), która potrzebuje pomocy od *obywateli*.
 - **Helper**
Osoba (najczęściej *obywatel*), która jest chętna do pomocy i działalności charytatywnej.
-

Cel bazy?

- Zarządzanie systemem *profilu użytkowników* do uproszczenia połączenia osób, potrzebujących pomocy, oraz osób, chętnych pomóc. Możliwe jest tworzenie *zadań* (*tasks*) przez *użytkowników standardowych* (*persons*) oraz przyjęcie tych zadań przez *pomocników* (*helpers*), które też są użytkownikami.
-

Scenariusze użycia

- Utworzenie konta *użytkownika*
- Utworzenie konta *pomocnika*
- Utworzenie *zadania*
- Utworzenie *zadania grupowego* (kiedy pomocy potrzebuje grupa osób)
- Przyjęcie *zadania*
- Wyszukiwanie *zadań* przy pomocy *filtrów*
- Zostawienie *opinii* (cel: *użytkownik*)
- Zostawienie *opinii* (cel: *pomocnik*)

- Dodanie innej osoby do *black-listu*
 - Usunięcie innej osoby z *black-listu*
 - Przypisanie *konta pomocnika* do *organizacji charytatywnej*
 - Zarządzanie *keyword-ami* (dodawanie oraz usuwanie)
-

Przykładowe zapytania

Podaj...

1. *zablokowanych* przeze mnie *użytkowników*
 2. *zadania*, w których *uczestniczę*
 3. *zadania*, które *zaakceptowałem*
 4. takie *zadania*, *odległość* których od mojego miasta jest *mniejsza od n*
 5. takie *zadania*, które wymagają *znany przeze mnie język*
-

Co nie będzie wchodziło w skład bazy?

- Szczegóły dotyczące *organizacji charytatywnych*
 - Szczegóły dotyczące *regionów i państw*
-

Entity Relationship Diagram:

Image

- **Attributes:**
 - `region_id_A` (*Foreign Key*): The first region in the connection.
 - `region_id_B` (*Foreign Key*): The second region in the connection.
 - `border_side_id` (**Primary Key**): Identifier for the border side connecting the regions.

5. Cities

- **Attributes:**
 - `city_id` (**Primary Key**): Unique identifier for the city.
 - `name`: Name of the city.
 - `population`: Population of the city.

6. Tasks

- **Attributes:**
 - `task_id` (**Primary Key**): Unique identifier for the task.
 - `date_created`: Date the task was created.
 - `date_deadline`: Deadline for the task.
 - `creator_id` (*Foreign Key*): Identifier of the person who created the task.
 - `km_to_closest_city`: Distance to the nearest city.
 - `language_required` (*Foreign Key*): Name of the language required for the task.

7. Persons

- **Attributes:**
 - `person_id` (**Primary Key**): Unique identifier for the person.
 - `name`: First name of the person.
 - `s_name`: Surname of the person.
 - `nationality`: Nationality of the person.
 - `b_date`: Birthdate of the person.
 - `rating`: Rating of the person.
 - `citizenship`: Citizenship of the person.
 - `status`: Current status of the person (for example refugee, student and so on).

8. Blocks

- **Attributes:**
 - `block_id` (**Primary Key**): Unique identifier for the block.
 - `person_id_blocking` (*Foreign Key*): Identifier of the person initiating the block.
 - `person_id_blocked` (*Foreign Key*): Identifier of the person being blocked.
 - `reason`: Reason for the block.
 - `date_start`: Start date of the block.

- `date_end` : End date of the block.
- `details` : Additional details about the block.

9. Volunteer_organizations

- **Attributes:**
 - `organization_id` (**Primary Key**): Unique identifier for the organization.
 - `name` : Name of the organization.
 - `number_of_employees` : Number of employees in the organization.
 - `contact_number` : Contact number of the organization.

10. Helpers

- **Attributes:**
 - `person_id` (**Primary Key**): Identifier of the person associated with the helper role.
 - `PESEL` : Unique identifier for the helper.
 - `organization_id` (*Foreign Key*): Identifier of the organization the helper belongs to. It can be empty if the helper is not a volunteer.

11. Participations

- **Attributes:**
 - `participation_id` (**Primary Key**): Unique identifier for the participation record.
 - `task_id` (*Foreign Key*): Identifier of the task the person participates in.
 - `person_id` (*Foreign Key*): Identifier of the person participating in the task.
 - `date_of_acceptance` : Date the participation was accepted.
 - `details` : Additional details about the participation.

12. Acceptances

- **Attributes:**
 - `acceptance_id` (**Primary Key**): Unique identifier for the acceptance record.
 - `task_id` (*Foreign Key*): Identifier of the task accepted.
 - `person_id` (*Foreign Key*): Identifier of the person who accepted.
 - `date_of_acceptance` : Date of acceptance.
 - `details` : Additional details about the acceptance.

13. Opinions

- **Attributes:**
 - `opinion_id` (**Primary Key**): Unique identifier for the opinion.
 - `person_id` (*Foreign Key*): Identifier of the person giving the opinion.
 - `acceptance_id` (*Foreign Key*): Identifier of the associated acceptance record.
 - `date` : Date the opinion was given.
 - `details` : Additional details about the opinion.

Relationships

1. Languages - Keywords

- **Relationship:** IS_A
- **Description:** Indicates that a language is a specific keyword.

2. Tasks - Languages

- **Relationship:** Requires
 - Tasks : 0...n
 - Languages : 1
- **Description:** A specific task always requires one language. The rest of them are optional, therefore they are added as keywords.

3. Tasks - Keywords

- **Relationship:** Has
 - Tasks : 0...n
 - Keywords : 0...n
- **Description:** A task can have some additional keywords.

4. Tasks - Cities

- **Relationship:** Belongs_to
 - Tasks : 0...n
 - Cities : 1
- **Description:** A task is always associated with a specific city. The "association area" is implemented with the help of km_to_closest_city parameter

5. Cities - Regions

- **Relationship:** Belongs_to
 - Tasks : 1...n
 - Cities : 1
- **Description:** Associates a city with the region it is located in.

6. Regions - Regions_connection

- **Relationship:** is_connected_to
 - Regions : 1
 - Regions_connection : 1
- **Description:** Shows a connection between two regions.

7. Persons - Keywords

- **Relationship:** Has
 - Person : 0...n
 - Keywords : 0...n

- **Description:** A `person` can have some additional `keywords`, which helps us to find the best fitting `task`.

8. Persons - Cities

- **Relationship:** `Lives_in`
 - `Persons`: `1...n`
 - `Cities`: `1`
- **Description:** A `person` lives in a specific `city`, so they are associated.

9. Persons - Blocks

- **Relationship:** `Has_blocked`
 - `Persons`: `1`
 - `Blocks`: `0...n`
- **Description:** One `person` added another to the black list. This relationship helps us to save history of `blocks`.

10. Persons - Participations

- **Relationship:** `Participates_in`
 - `Persons`: `1`
 - `Participations`: `0...n`
- **Description:** Additional info about `participation` in the `task`. There is always at least one `person` participating in the `task` (a creator of it).

11. Participations - Tasks

- **Relationship:** `Refers_to`
 - `Participations`: `1...n`
 - `Tasks`: `1`
- **Description:** Reference to the `task` `user` participates in.

12. Participations - Opinions

- **Relationship:** `Has_left`
 - `Participations`: `1`
 - `Opinions`: `0...1`
- **Description:** After the `task` has been finished, a `person` can leave the `opinion` referring to the `helper`'s `acceptance`.

13. Helpers - Persons

- **Relationship:** `IS_A`
- **Description:** A `helper` is a "subtype" of `person`.

14. Helpers - Volunteer_organizations

- **Relationship:** `Volunteers_in`
 - `Helpers` : `1...n`
 - `Volunteer_organizations` : `0...1`
- **Description:** A `helper` can be a part of the `volunteer_organization` (it is optional).

15. Helpers - Acceptances

- **Relationship:** `Accepted`
 - `Helpers` : `1`
 - `Acceptances` : `0...n`
- **Description:** Additional info about `helper` accepting the `task`.

16. Acceptances - Opinions

- **Relationship:** `Has_left`
 - `Acceptances` : `1`
 - `Opinions` : `0...1`
- **Description:** After finishing accepted `task`, `helper` can left the `opinion` referring to the specific `person`'s `participation` in the task.

17. Acceptances - Tasks

- **Relationship:** `Refers_to`
 - `Acceptances` : `0...n`
 - `Tasks` : `1`
- **Description:** Reference to the `task` `helper` accepted.

Relational Data Base

1. Languages:

```
CREATE TABLE Languages (  
    name VARCHAR(255) PRIMARY KEY,  
    `group` VARCHAR(255)  
);
```

2. Keywords:


```
CREATE TABLE Keywords (  
    name VARCHAR(255) PRIMARY KEY,  
    category VARCHAR(255)  
);
```

3. Regions:

```
CREATE TABLE Regions (  
    region_id INT PRIMARY KEY,  
    name VARCHAR(255),  
    country_name VARCHAR(255),  
    population INT  
);
```

4. Regions_connection:

```
CREATE TABLE Regions_connection (  
    region_id_A INT REFERENCES Regions(region_id),  
    region_id_B INT REFERENCES Regions(region_id),  
    border_side_id INT PRIMARY KEY  
);
```

5. Cities:

```
CREATE TABLE Cities (  
    city_id INT PRIMARY KEY,  
    name VARCHAR(255),  
    population INT,  
    region_id INT REFERENCES Regions(region_id)  
);
```

6. Tasks:

```
CREATE TABLE Tasks (  
    task_id INT PRIMARY KEY,  
    date_created DATETIME,  
    date_deadline DATETIME,  
    creator_id INT REFERENCES Persons(person_id),  
    km_to_closest_city FLOAT,  
    language_required VARCHAR(255) REFERENCES Languages(name),  
    city_id INT REFERENCES Cities(city_id)  
);
```

7. Persons:

```
CREATE TABLE Persons (  
    person_id INT PRIMARY KEY,
```

```
name VARCHAR(255),  
s_name VARCHAR(255),  
nationality VARCHAR(255),  
b_date DATETIME,  
rating FLOAT,  
citizenship VARCHAR(255),  
status VARCHAR(255),  
city_id INT REFERENCES Cities(city_id)  
);
```

8. Blocks:

```
CREATE TABLE Blocks (  
    block_id INT PRIMARY KEY,  
    person_id_blocking INT REFERENCES Persons(person_id),  
    person_id_blocked INT REFERENCES Persons(person_id),  
    reason TEXT,  
    date_start DATETIME,  
    date_end DATETIME,  
    details TEXT  
);
```

9. Volunteer_organizations:

```
CREATE TABLE Volunteer_organizations (  
    organization_id INT PRIMARY KEY,  
    name VARCHAR(255),  
    number_of_employees INT,  
    contact_number VARCHAR(15)  
);
```

10. Helpers:

```
CREATE TABLE Helpers (  
    person_id INT PRIMARY KEY REFERENCES Persons(person_id),  
    PESEL CHAR(11),  
    organization_id INT REFERENCE  
Volunteer_organizations(organization_id)  
);
```

11. Participations:

```
CREATE TABLE Participations (  
    participation_id INT PRIMARY KEY,  
    task_id INT REFERENCES Tasks(task_id),  
    person_id INT REFERENCES Persons(person_id),  
    date_of_acceptance DATETIME,
```

```
        details TEXT
    );
```

12. Acceptances:

```
CREATE TABLE Acceptances (
    acceptance_id INT PRIMARY KEY,
    task_id INT REFERENCES Tasks(task_id),
    person_id INT REFERENCES Helpers(person_id),
    date_of_acceptance DATETIME,
    details TEXT
);
```

13. Opinions:

```
CREATE TABLE Opinions (
    opinion_id INT PRIMARY KEY,
    person_id INT REFERENCES Persons(person_id),
    acceptance_id INT REFERENCES Acceptances(acceptance_id),
    participation_id INT REFERENCES Participations(participation_id),
    date_of_creation DATETIME,
    details TEXT,
    rate INT
);
```
