

Prepare a PHP project that uses PropelORM to import data from CSV files into a database. Use the imported data to generate several different outputs. The assignment contains 4 tasks.

The code of the project should be source-controlled with git. You can submit your final product through a public git repository (like GitHub), or by using `git bundle`.

ASSIGNMENT DATA

To perform this assignment, you will need data that is stored in 4 CSV files (folder 'data'). The columns named "lat" and "lng" represent latitude and longitude.

users.csv:

```
id,name
1,Janez
2,Miha
3,Kristof
4,Matic
```

restaurants.csv

```
id,name,lat,lng
1,"Dobra kuhinja",46.043590,14.492127
2,"Hitra Pizza",46.053479,14.511117
```

addresses.csv

```
id,address,lat,lng
1,"Devova ulica 5",46.086199,14.477493
2,"Kongresni trg 10",46.049509,14.504687
3,"Jaksiceva ulica 6",46.062311,14.506616
4,"Maistrova ulica 14",46.056501,14.517692
```

orders.csv:

```
id,user_id,address_id,restaurant_id,value,date,status
1,1,1,1,12.2,2018-08-02,2
2,3,2,2,22.3,2018-08-05,2
3,2,4,1,7.8,2018-08-07,3
4,4,3,2,5,2018-08-01,2
5,2,1,2,6.8,2018-08-01,2
6,2,4,2,12.1,2018-08-07,2
```

TASK 1A - Define the PropelORM XML schema

The data from the CSV files needs to be stored into a database. We'll be using PropelORM to interact with the database.

PropelORM is ready to use in this project. Before you begin your work, you need to execute the following in the root folder of this project:

- `php composer.phar install` - Installs PropelORM
- `./vendor/bin/propel sql:build` - Generates the SQL statements that will setup an empty database according to `schema.xml`
- `./vendor/bin/propel sql:insert` - Sets up the database with the SQL statements generated in the previous step
- `./vendor/bin/propel build` - Generates the PHP classes used to interact with the database
- `php composer.phar dump-autoload` - Generates `autoload.php`, which you use to access the classes generated in the previous step

Define the PropelORM XML schema (file `schema.xml`), that includes 4 tables:

1. user, with the fields: id, name
2. restaurant, with the fields: id, name, lat, lng
3. address, with the fields: id, address, lat, lng
4. orders, that connects tables 1-3 and has the following fields: id, user_id, address_id, restaurant_id, value, date, status

And follow these specifications:

- All fields named `id` should be a primary key (auto increment)
- Fields `user_id`, `address_id` and `restaurant_id` in the table `order` need to be foreign keys to relevant tables
- Other fields should use the most appropriate types, according to the data in the CSV files

To help you get started, the table `user` has already been added to `schema.xml`. An example of a script, that is properly configured to work with PropelORM can be found in `example.php`.

Hints:

- All PropelORM instructions can be found at <http://propelorm.org/documentation/>
- When expanding `schema.xml`, make sure you don't use SQL reserved keywords (like "order", "group") in table names
- No visual user interface is required (and this goes for all tasks in this assignment)

TASK 1B - import data

Write a script that imports CSV data from the supplied files and uses PropelORM to store them into the database.

Hint:

- import/reading of CSV files is a common issue, no need to re-invent the wheel, use existing PHP functions/libraries

TASK 1C - filtering and outputting data

Write a script that performs lookups in the database (the data needs to be imported - Task 1B). The script should output:

1. all orders on 1.8.2018
2. all orders by the user with the ID 2
3. all orders with status 3 and user ID 2
4. all orders with the value greater than 10
5. a sum of order values per day (a total sum of values for each date)

TASK 1D - extend the ORM classes

Let's assume that every restaurant delivers only inside the 2km radius from their location.

1. Extend the ORM class Restaurant, which represents restaurants, with a function that takes an Address object as an argument and checks if the passed address is inside the restaurant's radius.
2. Output all invalid orders - orders, with an address that is outside the restaurants delivery radius

Hint:

- Use the distance calculation function (below)

HELP

A function that takes 2 coordinates and returns the distance between them:

<php

```
/**
 * Calculate the distance between two points.
 * @param $lat1 float Latitude of point 1.
 * @param $lng1 float Longitude of point 1.
 * @param $lat2 float Latitude of point 2.
 * @param $lng2 float Longitude of point 2.
 * @return float Distance in meters
 */
function latLngToDistance($lat1, $lng1, $lat2, $lng2)
{
    $north = ($lat2 - $lat1) * 110574;
    $mean_lat = ($lat2 + $lat1) / 2.0;
    $east = ($lng2 - $lng1) * 111320 * cos($mean_lat / 180.0 * pi());
    return sqrt(pow($north, 2) + pow($east, 2));
}
```