

V okviru te naloge se pripravi PHP projekt, ki s pomočjo knjižnice PropelORM v bazo uvozi podatke iz CSV datotek. Glede na uvožene podatke pa kasneje pripravi določene izpise. Naloga vsebuje 4 sklope.

Koda projekta naj bo pod GIT source control-om. Končen izdelek lahko pošlješ preko javnega repository-ja (npr. GitHub) ali s pomočjo ukaza `git bundle`.

PODATKI ZA NALOGO

Za izvedbo te naloge potrebuješ podatke, ki so shranjeni v 4 CSV datotekah (mapa 'data').

Stolpci "lat" in "lng" predstavljajo zemljepisno širino in dolžino.

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

NALOGA 1A - Definiraj PropelORM XML schema

Podatke iz CSV datotek bo potrebno shraniti v bazo. Za interakcijo z bazo bomo uporabili PropelORM.

PropelORM je v tem projektu že pripravljen za uporabo. Pred začetkom dela moraš v root mapi tega projekta ročno zagnati še:

- `php composer.phar install` - za instalacijo PropelORM knjižnice (samo 1x)
- `./vendor/bin/propel sql:build` - Zgenerira ukaze, ki bojo postavili novo prazno bazo glede na `schema.xml`
- `./vendor/bin/propel sql:insert` - Postavi novo bazo glede na zgoraj zgenerirane ukaze
- `./vendor/bin/propel build` - Zgenerira PHP Class-e, ki se uporabljajo za interakcijo z bazo
- `php composer.phar dump-autoload` - Zgenerira `autoload.php`, ki omogoča dostop do ORM PHP Class-ov znotraj skript

Definiraj PropelORM XML schema (datoteka `schema.xml`), ki vsebuje 4 tabele:

1. tabela user, ki ima definirani polji: id, name
2. tabela restaurant, ki ima definirana polja: id, name, lat, lng
3. tabela address, ki ima definirana polja: id, address, lat, lng
4. tabela order, ki povezuje tabele 1-3 in ima definirana polja: id, user_id, address_id, restaurant_id, value, date, status

In ustreza naslednjim specifikacijam:

- Vsa polja `id` naj bodo primary key (auto increment).

- Polja `user_id`, `address_id` in `restaurant_id` v tabeli `order` morajo biti foreign key-ji na ustrezne tabele
- Za ostala polja uporabi najustreznejše tip-e, glede na podatke zgoraj (csv datoteke)

Za lažji začetek dela, je tabela `user` že dodana v `schema.xml`. Primer skripte, ki je skonfigurirana, da dela s PropelORM pa najdeš v `example.php`.

Namigi:

- Vsa navodila glede uporabe PropelORM knjižnice lahko najdeš na <http://propelorm.org/documentation/>
- Ko pišeš `schema.xml` za imena tabel ne uporablaj "SQL reserved keyword"-om kot so "order", "group", ...
- Vizualnega uporabniškega vmesnika ti ni potrebno pripravljati (velja za vse sklope te naloge)

NALOGA 1B - uvozi podatke

Napiši skripto, ki podatke iz vhodnih CSV datotek prebere in jih s pomočjo PropelORM-ja shrani v podatkovno bazo.

Namigi:

- uvoz/branje CSV datotek je pogost problem; ne izumljaj tople vode, uporabi obstoječe funkcije/knjižnice

NALOGA 1C - filtriranje in izpis podatkov

Napiši skripto, ki izvede poizvedbe na bazo (podatki morajo biti že uvoženi - Naloga 1b). Skripta naj izpiše:

1. vsa naročila na dan 1.8.2018
2. vsa naročila uporabnika z ID 2
3. vsa naročila s statusom 3 in uporabnikom 2
4. vsa naročila z vrednostjo nad 10
5. vsoto vrednosti vseh naročil, po datumu (za vsak datum seštevek vrednosti)

NALOGA 1D - razširitev ORM Class-ov

Predpostavimo, da vsaka restavracija dostavlja le v radiju 2km od svoje lokacije. V okviru te pod-naloge:

1. Razširi ORM class Restaurant, ki predstavlja restavracije, z metodo, ki kot argument sprejme objekt Address in preveri, če je ta naslov veljaven za dostavo s te restavracije (ali je znotraj radija dostave).

2. Izpiši vsa neveljavna naročila - torej naročila, ki imajo naslov izven radija dostave restavracije

Namigi:

- Pomagaj si s funkcijo za preračunavanje razdalj (glej spodaj)

POMOČ

Funkcija, ki pretvori 2 koordinati v razdaljo med njima:

```
<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));
}
```