

Tp Doctrine

Sara Mannasaf

1_: symfony new resataumange_MannasafSara -webapp

2_:fichier .env :

```
DATABASE_URL="mysql://root2:root2@127.0.0.1:3306/restau_db?  
serverVersion=5.7&charset=utf8mb4"
```

3_:php bin/console doctrine:database:create

4_:php bin/console make:entity

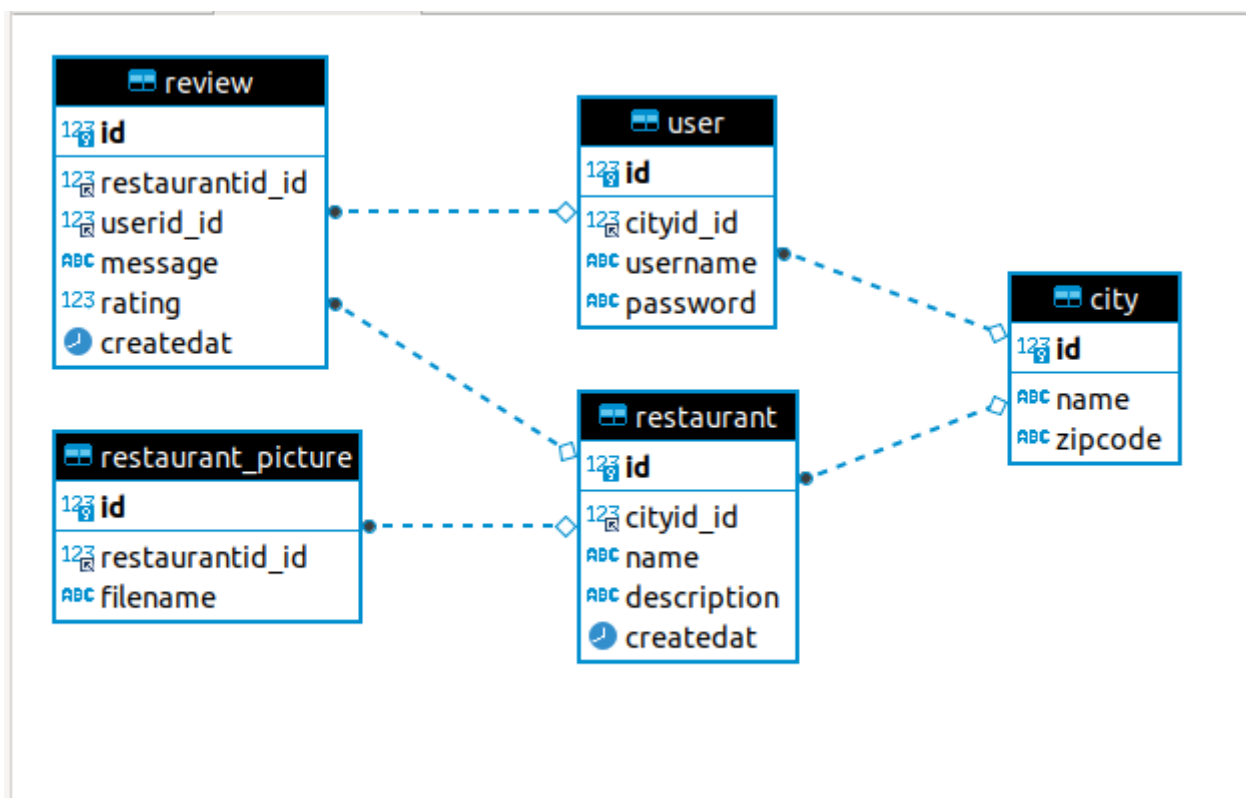
```
6_:public function __construct()  
{
```

```
$this->createdAt= new \DateTime('now');
```

```
}
```

7_: créer une migration : php bin/console make:migration

run la migration: php bin/console doctrine:migrations:migrate



8_ : php bin/console make:controller RestaurantController

9_ : On ajoute cette ligne dans le fichier de migration pour chaque script:
\$this->addSql(file_get_contents("scripts/city.sql"));

10 :

1_:

```
/**
 * @Route("/restaurants", name="app_restaurant")
 */
public function restaurants()
{
    $restaurants = $this->getDoctrine()->getRepository(Restaurant::class)->findAll();

    return $this->render('restaurant/index.html.twig', [
        'restaurants' => $restaurants,
    ]);
}
```

2_:

```
/**
 * @Route("/restaurant/new", name="app_restaurant2")
 */
public function new()
{
    $res= $this->getDoctrine()->getRepository(Restaurant::class)->findOneBy([],['createdat' => 'desc']);

    return $this->render('restaurant/new.html.twig', [
        'restaurant' => $res,
    ]);
}
```

3_:

```
/**
 * @Route("/restaurant", name="app_restaurant3")
 */
public function restadd()
{
    return $this->render('restaurant/add.html.twig');
}

/**
 * @Route("/restaurant/res", name="app_restaurant4")
 */
public function create(Request $request, ManagerRegistry $doctrine)
{
    $restaurant = new Restaurant();
    $entityManager = $this->getDoctrine()->getManager();

    $idcity = $request->get('id');
    $name = $request->get('name');
    $description = $request->get('desc');
    $createdat = $request->get('datecreat');

    $restaurant->setName($name);
    $restaurant->setDescription($description);
    $restaurant->setCreatedAt(\DateTime::createFromFormat('Y-m-d', $createdat));
    $city = $doctrine->getRepository(City::class)->find($idcity);
    $restaurant->setCityid($city);

    // Enregistrement //
    $entityManager->persist($restaurant);
    // Execution des enregistrement //
    $entityManager->flush();

    return $this->render('restaurant/add.html.twig');
}
```

4_:

```
/**
 * @Route("/restaurant/{restaurant}", name="app_restaurant5")
 */
public function restaurant(Restaurant $restaurant, ManagerRegistry $doctrine)
{
    $res = $doctrine->getRepository(Restaurant::class)->find($restaurant);

    return $this->render('restaurant/search.html.twig',
    [
        'restaurant' => $res,
    ]);
}
```

5_:

```

* @Route("/review", name="app_review")
*/
public function revadd()
{
    return $this->render('review/index.html.twig');
}
/**
* @Route("/review/rev", name="app_review2")
*/
public function addReview(Request $request, ManagerRegistry $doctrine)
{
    $review = new Review();
    $entityManager = $this->getDoctrine()->getManager();

    $idrestau=$request->get('idr');
    $iduser=$request->get('idu');
    $message=$request->get('message');
    $rating=$request->get('rating');
    $createdat=$request->get('datecreat');

    $review->setMessage($message);
    $review->setRating($rating);
    $review->setCreatedat(\DateTime::createFromFormat('Y-m-d', $createdat));
    $restaurant = $doctrine->getRepository(Restaurant::class)->find($idrestau);
    $User = $doctrine->getRepository(User::class)->find($iduser);

    $review->setRestaurantid($restaurant);
    $review->setUserid($User);

    // Enregistrement //
    $entityManager->persist($review);
    // Execution des enregistrement //
    $entityManager->flush();

    return $this->render('review/add.html.twig');
}

```

6_:

```

/**
* @Route("/user", name="app_user")
*/
public function users()
{
    $users = $this->getDoctrine()->getRepository(User::class)->findAll();

    return $this->render('user/index.html.twig', [
        'users' => $users,
    ]);
}

```


11_: Afficher les 6 derniers restaurants créés:

```
public function findbydate($limit): array
{
    $entityManager = $this->getEntityManager();

    $query = $entityManager->createQuery(
        'SELECT R
        FROM App\Entity\Restaurant R
        ORDER BY R.createdat DESC
        '
    )->setMaxResults($limit);

    // returns an array of Product objects
    return $query->getResult();
}
```

12_:Afficher la valeur moyenne de la note d'un restaurant:

```
public function restaurantrating($id): array
{
    $entityManager = $this->getEntityManager();

    $query = $entityManager->createQuery(
        'SELECT avg(r.rating)
        FROM App\Entity\Restaurant re ,
        App\Entity\review r where r.restaurantid = re.id
        AND re.id = :id GROUP BY re.id'
    )->setParameter('id', $id);

    // returns an array of Product objects
    return $query->getResult();
}
```

13_:Afficher les 3 top meilleurs restaurants :

```
public function topthree($limit): array
{
    $entityManager = $this->getEntityManager();

    $query = $entityManager->createQuery(
        'SELECT re.name
        FROM App\Entity\Restaurant re ,
        App\Entity\review r where r.restaurantid = re.id
        GROUP BY re.id
        ORDER BY avg(r.rating) DESC '
    )->setMaxResults($limit);

    // returns an array of Product objects
    return $query->getResult();
}
```

15- Lister les restaurants et leurs détails (review, city..)

```
public function restaudetaillls(): array
{
    $entityManager = $this->getEntityManager();

    $query = $entityManager->createQuery(
        'SELECT re
        FROM App\Entity\Restaurant re ,App\Entity\City c,
        App\Entity\review r where r.restaurantid = re.id
        AND c.id = re.cityid
        '
    );

    // returns an array of Product objects
    return $query->getResult();
}
```

16_ classer les restaurant pas vot

```
public function top(): array
{
    $entityManager = $this->getEntityManager();

    $query = $entityManager->createQuery(
        'SELECT re.name
        FROM App\Entity\Restaurant re ,
        App\Entity\review r where r.restaurantid = re.id
        GROUP BY re.id
        ORDER BY avg(r.rating) DESC '
    );

    // returns an array of Product objects
    return $query->getResult();
}
```