

2023/2024	Ap-Itération 5
BTS SIO	Auteur : Guihard Tristan
2SLAM	Date de rédaction :13/11/2023

Compte-rendu Ap

État de la base de donnée et de la table préférer qui lie les utilisateurs à leurs types de cuisine préféré.

On a créer une table qui va lier les utilisateurs et leurs types de cuisines préférer car un utilisateur peut avoir plusieurs type de cuisines préférer et un type de cuisine peut être préférer par plusieurs personne.

```
CREATE TABLE prefere(  
idU bigint not null,  
idTC bigint not null);  
  
ALTER TABLE prefere  
add PRIMARY KEY(idU,idTC),  
ADD KEY(idTC);  
  
ALTER TABLE prefere  
ADD CONSTRAINT prefere_ibfk_1 FOREIGN KEY(idU) REFERENCES utilisateur(idU),  
ADD CONSTRAINT prefere_ibfk_2 FOREIGN key(idTC) REFERENCES type_cuisines(idTC)
```

	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1	idU	bigint			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	2	idTC	bigint			Non	Aucun(e)			Modifier Supprimer Plus
idU		idTC								
	1	1								
	1	3								
	6	4								
	6	10								
	6	11								

L'utilisateur 6 est l'utilisateur SIO.

Tout d'abord, on veut que les types de restaurants qu'il aime soit afficher, ainsi que ceux qu'il n'a pas encore sélectionner comme aimé donc dans le controleur updProfil.php ont insigne des valeurs correspondantes contenant les listes.

2023/2024	Ap-Itération 5
BTS SIO	Auteur : Guihard Tristan
2SLAM	Date de rédaction :13/11/2023

Création d'un nouveau modèle DAO\metier
TypeCuisine pour les typeCuisine

```
class TypeCuisine {

    private int $idTC;

    private string $libelleTC;

    public function getIdTC(): int {
        return $this->idTC;
    }

    public function getLibelleTC(): string {
        return $this->libelleTC;
    }

    public function setIdTC(int $idTC): void {
        $this->idTC = $idTC;
    }

    public function setLibelleTC(string $libelleTC): void {
        $this->libelleTC = $libelleTC;
    }

    public function __construct(int $idTC, string $libelleTC) {
        $this->idTC = $idTC;
        $this->libelleTC = $libelleTC;
    }

}
```

Modification du modele\metier Utilisateur pour qu'il est une liste de TypesAimes en attribut.

```
function __construct(int $idU, string $mailU, ?string $mdpU, ?string $pseudoU) {
    $this->idU = $idU;
    $this->mailU = $mailU;
    $this->mdpU = $mdpU;
    $this->pseudoU = $pseudoU;

    $this->lesRestosAimes = array();

    $this->lesTypesAimes = array();
}

public function getLesTypesAimes(): array {
    return $this->LesTypesAimes;
}
```

2023/2024	Ap-Itération 5
BTS SIO	Auteur : Guihard Tristan
2SLAM	Date de rédaction :13/11/2023

```

public function setLesTypesAimes(array $LesTypesAimes): void {
    $this->LesTypesAimes = $LesTypesAimes;
}

```

Et changement des méthodes pour permettre de récupérer la liste des type de cuisines qu'il préfère

```

public static function getOneById(int $idU): ?Utilisateur {
    $leUser = null;
    try {
        $requete = "SELECT * FROM utilisateur WHERE idU = :idU";
        $stmt = Bdd::connecter()->prepare($requete);
        $stmt->bindParam(':idU', $idU, PDO::PARAM_INT);
        $ok = $stmt->execute();
        // attention, $ok = true pour un select ne retournant aucune ligne
        if ($ok && $stmt->rowCount() > 0) {
            $enreg = $stmt->fetch(PDO::FETCH_ASSOC);
            $lesRestosAimes = RestoDAO::getAimesByIdU($idU);
            $lesTypesAimes = TypeCuisinesDAO::getPrefererByIdU($idU);

            $leUser = new Utilisateur($idU, $enreg['mailU'], $enreg['mdpU'], $enreg['pseudoU']);

            $leUser->setLesRestosAimes($lesRestosAimes);
            $leUser->setLesTypesAimes($lesTypesAimes);
        }
    } catch (PDOException $e) {
        throw new Exception("Erreur dans la méthode " . get_called_class() . " : getOneById : <br/>" . $e->getMessage());
        throw new Exception("Erreur dans la méthode " . get_called_class() );
    }
    return $leUser;
}

```

```

// Sinon, on revient sur le formulaire
// Recharger les données
$util = UtilisateurDAO::getOneById($idU);
$mesRestosAimes = $util->getLesRestosAimes();
$mesTypesPreferer = $util->getLesTypesAimes();
$TypeCuisine= TypeCuisinesDAO::getAllSaufUtilisateur($idU);

```

\$TypeCuisine prends tous les types de cuisine que l'utilisateur n'a pas encore sélectionner.
Pour cela la méthode de TypeCuisinesDAO::getAllSaufUtilisateur est appelé.

2023/2024	Ap-Itération 5
BTS SIO	Auteur : Guihard Tristan
2SLAM	Date de rédaction :13/11/2023

```

public static function getAllSaufUtilisateur(int $idU): array{
    $lesObjets = array();
    try {
        $requete = "SELECT tc.* FROM type_cuisines tc
                    WHERE tc.idTC NOT IN (SELECT idTC FROM prefere WHERE idU =:idU)";
        $stmt = Bdd::getConnexion()->prepare($requete);
        $stmt->bindParam(':idU', $idU, PDO::PARAM_INT);
        $ok = $stmt->execute();
        if ($ok) {
            while($enreg = $stmt->fetch(PDO::FETCH_ASSOC)){
                $lesObjets[] = self::enregistrementVersObjet($enreg);
            }
        }
    } catch (Exception $e) {
        throw new Exception("Erreur dans la méthode " . get_called_class() . "::getAllType : <br/>" . $e->getMessage());
    }
    return $lesObjets;
}

```

Code complémentaire pour enregistrementVersObjet.

```

private static function enregistrementVersObjet(array $enreg): TypeCuisine {
    // Instanciation sans les associations
    $leTypeCuisine = new TypeCuisine(
        $enreg['idTC'], $enreg['libelleTC']
    );

    return $leTypeCuisine;
}

```

Ici, un requête sql est effectué sur la base de données pour récupérer les données.

On affiche les types que l'utilisateur préfère dans le profil simple grâce à la vueMonProfil.php

```

les types de restaurants que j'aime :
<?php
foreach ($mesTypesPreferer as $unType) {
    ??
    <div class = "cadre"><span style="color: red;">#</span><?= $unType->getLibelleTC() ?></div>
    <?php
}
??
<hr>
<a href = "../?action=deconnexion">se deconnecter</a>
<style>
    .cadre {
        border: 2px solid #000; /* Couleur de la bordure et épaisseur */
        padding: 10px; /* Espace intérieur du cadre */
        border-radius: 5px; /* Coins arrondis (facultatif) */
    }
</style>

```

La boucle foreach permet de parcourir la listes Des typesPreferer de la personne et afficher le libelle de chacun un par un grâce au getter LibelleTC.

Pour plus d'information sur les préférence culinaire de l'utilisateur si on ouvre les données complémentaire dans la profil avancé grâce à la vueUpdProfil.php

2023/2024	Ap-Itération 5
BTS SIO	Auteur : Guihard Tristan
2SLAM	Date de rédaction :13/11/2023

```

<br /><br />
Supprimer les types de cuisine que je préfère : <br />
<?php
for ($i = 0; $i < count($mesTypesPreferer); $i++) {
    $unType = $mesTypesPreferer[$i];
    ??
    <input type="checkbox" name="lstidTC[]" id="Gout<?= $i ?>" value="<?= $unType->getIdTC() ?>" >
    <label for="type_resto<?= $i ?>"><?= $unType->getLibelleTC() ?></label><br />
<?php
} ??
<input type="submit" value="Supprimer" />

<br /><br />
Ajouter les types de cuisine que je préfère : <br />
<?php
for ($i = 0; $i < count($TypeCuisine); $i++) {
    $toutLesTypes = $TypeCuisine[$i];
    ??
    <input type="checkbox" name="lstidGouts[]" id="Gouts<?= $i ?>" value="<?= $toutLesTypes->getIdTC() ?>" >
    <label for="type_Gouts<?= $i ?>"><?= $toutLesTypes->getLibelleTC() ?></label><br />
<?php
} ??
<input type="submit" value="Ajouter" />

```

Des checkbox sont créer et affiche les valeurs des typesPreferer de la personne ainsi que ceux non sélectionner pour l’instant.

Ici, l'utilisateur peut submit soit ceux qui veut supprimer soit ceux qui veut rajouter dans chacune des listes.

Toutes les données soumise sont récupérés dans le controleur updProfil.php pour permettre de supprimer ou de rajouter les préférence.

```

}
// Goûts à supprimer de la liste
if (isset($_POST["lstidTC"])) {
    $lstidTC = $_POST["lstidTC"];
    for ($i = 0; $i < count($lstidTC); $i++) {
        PrefererDAO::delete($idU, $lstidTC[$i]);
    }
}
// Lister les goûts
if (isset($_POST["lstidGouts"])) {
    $lstidGouts = $_POST["lstidGouts"];
    for ($i = 0; $i < count($lstidGouts); $i++) {
        PrefererDAO::insert($idU, $lstidGouts[$i]);
    }
}

```

Encore, une fois des méthodes sont appelées pour permettre de modifier la base de données.

2023/2024	Ap-Itération 5
BTS SIO	Auteur : Guihard Tristan
2SLAM	Date de rédaction :13/11/2023

Ici, `PrefererDAO::delete` et `PrefererDAO::insert` sont appelées.


PrefererDAO::delete


```
public static function delete(int $idU, int $idTC): bool {
    $resultat = false;
    try {
        $stmt = Bdd::getConnexion()->prepare("DELETE FROM prefere WHERE idTC=:idTC and idU=:idU");
        $stmt->bindParam(':idTC', $idTC, PDO::PARAM_INT);
        $stmt->bindParam(':idU', $idU, PDO::PARAM_INT);
        $resultat = $stmt->execute();
    } catch (PDOException $e) {
        throw new Exception("Erreur dans la méthode " . get_called_class() . "::delete : <br/>" . $e->getMessage());
    }
    return $resultat;
}
```

PrefererDAO::insert

```
public static function insert(int $idU, int $idTC): bool {
    $resultat = false;
    try {
        $stmt = Bdd::getConnexion()->prepare("INSERT INTO prefere(idU, idTC) VALUES(:idU,:idTC)");
        $stmt->bindParam(':idU', $idU, PDO::PARAM_INT);
        $stmt->bindParam(':idTC', $idTC, PDO::PARAM_INT);
        $resultat = $stmt->execute();
    } catch (PDOException $e) {
        throw new Exception("Erreur dans la méthode " . get_called_class() . "::insert : <br/>" . $e->getMessage());
    }
    return $resultat;
}
```

Vu du profil





Mon profil

Mon adresse électronique : test@bts.sio
Mon pseudo : test@bts.sio

les restaurants que j'aime :

[le bar du charcutier](#)
[Agadir](#)
[Le Bistrot Sainte Cluque](#)
[la petite auberge](#)

les types de restaurants que j'aime :

#fastfood

#viandes

#grillade

[se deconnecter](#)

Mise à jour effectuée par l'équipe

2023/2024	Ap-Itération 5
BTS SIO 2SLAM	Auteur : Guihard Tristan
	Date de rédaction :13/11/2023

Vu de la base de donnée.

idU	idTC
1	1
1	3
6	4
6	10
6	11

Vu du profil avancé.

Supprimer les types de cuisine que je préfère :

- ☐ fastfood
- ☐ viandes
- ☐ grillade

Supprimer

Ajouter les types de cuisine que je préfère :

- ☐ sud ouest
- ☐ japonaise
- ☐ orientale
- ☐ vegetarienne
- ☐ vegan
- ☐ crepe
- ☐ sandwich
- ☐ tartes

Ajouter

2023/2024	Ap-Itération 5
BTS SIO 2SLAM	Auteur : Guihard Tristan
	Date de rédaction :13/11/2023

Nous Allons supprimer fast food et viandes

Supprimer les types de cuisine que je préfère :

☒ fastfood

☒ viandes

☐ grillade

Supprimer

Ajouter les types de cuisine que je préfère :

☐ sud ouest

☐ japonaise

☐ orientale

☐ vegetarienne

☐ vegan

☐ crepe

☐ sandwich

☐ tartes

Ajouter

Résultat point de vue profil.

Supprimer les types de cuisine que je préfère :

☐ grillade

Supprimer

Ajouter les types de cuisine que je préfère :

☐ sud ouest

☐ japonaise

☐ orientale

☐ fastfood

☐ vegetarienne

☐ vegan

☐ crepe

☐ sandwich

☐ tartes

☐ viandes

Ajouter

2023/2024	Ap-Itération 5
BTS SIO 2SLAM	Auteur : Guihard Tristan
	Date de rédaction :13/11/2023

Résultat base de donnée.

idU	idTC
1	1
1	3
6	11

Ajout de type Cuisine.

Supprimer les types de cuisine que je préfère :

☐ grillade

Supprimer

Ajouter les types de cuisine que je préfère :

☒ sud ouest

☒ japonaise

☒ orientale

☒ fastfood

☒ vegetarienne

☐ vegan

☐ crepe

☐ sandwich

☐ tartes

☐ viandes

Ajouter

Point de vue base de donnée.

2023/2024	Ap-Itération 5
BTS SIO 2SLAM	Auteur : Guihard Tristan
	Date de rédaction :13/11/2023

idU	idTC
1	1
1	3
6	1
6	2
6	3
6	4
6	5
6	11