

Compétence 3 : Mettre en place une base de données

Objectif pédagogique

Réaliser les 7 étapes permettant de mettre en place une base de données avec un jeu de test fonctionnel et de réaliser sa sauvegarde.

Contexte

Le développeur web peut être amené à mettre en place une base de données à partir d'un MPD et d'insérer un jeu de données pour faire fonctionner un site internet, une application, un ERP ou un logiciel métier.

Intention

Identifier et mettre en œuvre les différentes étapes de mise en place d'une base de données avec un exemple concret.

Livable.s

1. Un fichier pdf contenant :

- les tableaux des étapes 2a et 2b
- la liste des données erronées, incomplètes ou à vérifier de l'étape 2c

2. le backup de la base de données contenant les données

3. toutes les requêtes sql et/ou scripts fonctionnels qui ont permis l'insertion des données

4. le fichier nommé `donnee_a_revoir.csv`

Les fichiers sont déposés sur votre Github, le lien sera déposé dans Simplonline

Durée

7H

Critères de réussite

- La base de données est conforme au schéma physique
- L'intégrité des données est assurée
- La base de données de test peut être restaurée en cas d'incident

Énoncé

Le zoo de la Flèche a mis en place une base de données pour répartir la charge de travail concernant l'alimentation des animaux dans le zoo.

Grâce à la base de données et l'interface qui vont être créées, chaque employé peut savoir quels animaux sont dans son secteur et leur apporter la nourriture appropriée.

Le MPD a été réalisé, la base de données sera réalisée avec PostgreSQL.

Étape 1 : Mettre en œuvre un script de création de base de données

A partir du script sql (MPD) suivant, d'une interface de base de données ou d'un terminal, créer une base de données PostgreSQL puis exécuter le script pour créer les tables et les contraintes.

Vérifier la conformité de la base de données avec le MPD.

```
-----  
--      Script Postgre  
-----
```

```
-----  
-- Table: employe  
-----
```

```
CREATE TABLE public.employe(  
    id_pe SERIAL NOT NULL ,  
    nom  VARCHAR (25) ,  
    prenom VARCHAR (35) NOT NULL ,  
    CONSTRAINT prk_constraint_employe PRIMARY KEY (id_pe)  
)WITHOUT OIDS;
```

```
-----  
-- Table: secteur  
-----
```

```
CREATE TABLE public.secteur(  
    id_an SERIAL NOT NULL ,  
    secteur VARCHAR (45) NOT NULL ,  
    CONSTRAINT prk_constraint_secteur PRIMARY KEY (id_an)  
)WITHOUT OIDS;
```

```
-----  
-- Table: animaux  
-----
```

```
CREATE TABLE public.animaux(  
    id_ani SERIAL NOT NULL ,  
    nom_com VARCHAR (40) NOT NULL ,  
    nom_scien VARCHAR (50) ,  
    nb_indi INT NOT NULL ,  
    id_ali INT ,  
    CONSTRAINT prk_constraint_animaux PRIMARY KEY (id_ani)  
)WITHOUT OIDS;
```

-- Table: alimentation

```
CREATE TABLE public.alimentation(  
    id_ali SERIAL NOT NULL ,  
    type VARCHAR (25) NOT NULL ,  
    CONSTRAINT prk_constraint_alimentation PRIMARY KEY (id_ali)  
)WITHOUT OIDS;
```

-- Table: travailler

```
CREATE TABLE public.travailler(  
    id_an INT NOT NULL ,  
    id_pe INT NOT NULL ,  
    CONSTRAINT prk_constraint_travailler PRIMARY KEY (id_an,id_pe)  
)WITHOUT OIDS;
```

-- Table: se situer

```
CREATE TABLE public.se_situer(  
    id_an INT NOT NULL ,  
    id_an1 INT NOT NULL ,  
    CONSTRAINT prk_constraint_se_situer PRIMARY KEY (id_an,id_an1)  
)WITHOUT OIDS;
```

```
ALTER TABLE public.animaux ADD CONSTRAINT FK_animaux_id_ali FOREIGN KEY (id_ali)  
REFERENCES public.alimentation(id_ali);  
ALTER TABLE public.travailler ADD CONSTRAINT FK_travailler_id_an FOREIGN KEY  
(id_an) REFERENCES public.secteur(id_an);  
ALTER TABLE public.travailler ADD CONSTRAINT FK_travailler_id_pe FOREIGN KEY  
(id_pe) REFERENCES public.employe(id_pe);  
ALTER TABLE public.se_situer ADD CONSTRAINT FK_se_situer_id_an FOREIGN KEY  
(id_an) REFERENCES public.secteur(id_an);  
ALTER TABLE public.se_situer ADD CONSTRAINT FK_se_situer_id_an1 FOREIGN KEY  
(id_an1) REFERENCES public.animaux(id_an1);
```

Étape 2 : Vérifier un jeu de données test pour l'insérer dans une base de données

Étape 2a – Recherche des correspondances entre champs (table) et colonnes (fichier)

A partir du jeu de test fourni : employe.csv et zoo.csv, compléter le tableau ci-dessous en suivant les instructions suivantes :

- identifier les différentes colonnes des 2 fichiers : associer les colonnes des fichiers avec les champs de la base de données en complétant le tableau suivant.
- Ensuite compléter le tableau en vérifiant la correspondance entre le type et la longueur des données dans les tables et dans les fichiers et préciser au besoin les modifications à réaliser

Nom de la colonne du fichier	Nom du champ de la bdd	Correspondance type/longueur	modification
<i>Nom des employés</i>	<i>empl_nom</i>	<i>oui</i>	
<i>Salaire</i>	<i>empl_sal</i>	<i>Non</i>	<i>Transformer les nombres à virgule en nombre entier</i>

Étape 2b : Recherche des champs manquants

A partir du jeu de test fourni : employe.csv et zoo.csv, les tableaux remplis précédemment, compléter le tableau ci-dessous en suivant les instructions suivantes :

- lister les champs dont la donnée est à créer
- indiquer le type et la longueur spécifiés dans la base de données
- indiquer comment obtenir la donnée

Nom du champ de la bdd	Type (longueur)	Obtenir la donnée
<i>fk_an</i>	<i>Int(25)</i>	<i>Récupérer le champs id_an après remplissage de la table 'animation'</i>
<i>Arbre</i>	<i>Varchar(50)</i>	<i>Demander la donnée au client</i>

Étape 2c : Lister les données erronées, incomplètes ou à vérifier

A partir du jeu de test fourni : employe.csv et zoo.csv, les tableaux remplis précédemment, réaliser 3 listes en expliquant le problème avec la donnée :

- liste de données erronées (ex : Ligne 7 : il y a un chiffre dans la colonne situation familiale)
- liste de données incomplètes (ex : Ligne 44 : absence d'un prénom)
- liste de données à vérifier (ex : Ligne 67 : asticot → toutes les autres entrées sont notés asticots)

Étape 3 : Préparer un jeu de données à importer dans les tables courantes de la base de données

A partir des fichiers employe.csv et zoo.csv, organiser, trier, sélectionner les données dans un ou des fichiers pour les importer dans les tables non associatives (dites courantes).

Mettre dans un fichier nommé donnee_a_revoir.csv les données des listes de l'étape 2c.

Étape 4 : Importer les données dans la base de données

A partir des fichiers créés, de vos connaissances et des ressources, importer vos données dans les tables courantes de la base de données créées précédemment en utilisant des requêtes SQL, des scripts et/ou un outil d'importation de données.

Toutes les données, à l'exception des données erronées, incomplètes ou à vérifier sont à importer dans la base de données.

Étape 5 : Créer et insérer les données des tables associatives de la base de données

A partir des fichiers créés et fournis, de vos connaissances, des tables de la base de données et des ressources, créer et insérer les données des tables associatives, en utilisant, un tableur, des requêtes SQL, des scripts et/ou un outil d'importation de données et en respectant l'intégrité des données.

Toutes les données, à l'exception des données erronées, incomplètes ou à vérifier sont à importer dans la base de données.

Étape 6 : Vérifier les données

A partir de requêtes SQL, des fichiers fournis et d'une interface de visualisation de données, vérifier l'exactitude des données.

Étape 7 : Réaliser une sauvegarde de la base de données

A partir de requêtes SQL et/ou de l'interface de gestion de la base de données, réaliser une sauvegarde de la base de données avec son jeu de test intégré.

Le choix du format de sauvegarde est libre.