

DOSSIER PROFESSIONNEL (DP)

Nom de
naissance

▶ WAROUX

Nom d'usage

▶

Prénom

▶ GUILLAUME

Adresse

▶ 31 D RUE DE LAMORLAYE 60270
GOUVIEUX

Titre professionnel visé

Concepteur Développeur d'Applications

MODALITÉ D'ACCÈS :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel. **Ce titre est délivré par le Ministère chargé de l'emploi.**

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.

<http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité p.5-7

- ▶ Développer une interface utilisateur de type desktopp..5:6..
- ▶ Développer des composants d'accès aux donnéesp..6:7...

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité p.8-10

- ▶ Concevoir une base de donnéesp.8:9...
- ▶ Mettre en place une base de donnéesp.10....

Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité p.11-12

- ▶ Concevoir une applicationp..11...
- ▶ Préparer et exécuter les plans de tests d'une applicationp..12....

Titres, diplômes, CQP, attestations de formation *(facultatif)* p. 13

Déclaration sur l'honneur p. 14

Documents illustrant la pratique professionnelle *(facultatif)* p. 15

Annexes *(Si le RC le prévoit)* p. 16

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité- type 1

Exemple n°1 ▶

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Développer une interface utilisateur de type desktop

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pour maîtriser cette compétence, un petit projet a été mis en place qui consiste à la gestion d'une banque. Juste pour situer le contexte, une banque contiendra des clients, qui eux même posséderont des comptes.

Cette compétence a été abordé en utilisant la bibliothèque Swing basée sur le langage JAVA. Après avoir appris les principes essentiels de cette bibliothèque, j'ai commencé à coder l'application dans le but d'obtenir la liste des comptes et clients. Pour cela j'ai codé les classes héritant des classes de la bibliothèque Swing JFRAME, JMENU, JTABLE, etc.

Dès que cela fût fonctionnel en passant par le codage des différentes classes ,(voir annexe 1) ... j'ai pu passer à l'implémentation de l'interface desktop pour pouvoir manipuler les données (suppression, mise à jour, ajout). Pour cela, j'ai codé des classes supplémentaires héritant des classes de la bibliothèque Swing JDIALOG, etc.

L'application JAVA « client riche » est finalement complètement fonctionnelle permettant de lister, ajouter, supprimer et modifier les clients et les comptes en utilisant la bibliothèque Swing. (voir annexe 2).

2. Précisez les moyens utilisés :

Le langage JAVA, l'IDE Eclipse, la bibliothèque Swing, la base de données MySQL, les différents diagrammes UML.

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec le formateur Soupramanien, ainsi qu'avec les différents camarades de la formation.

DOSSIER PROFESSIONNEL (DP)



4. Contexte

Nom de l'entreprise, organisme ou association ▶

Greta de Compiègne

5. Informations complémentaires (facultatif)

Activité- type 1

Exemple n°2 ▶

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Développer des composants d'accès aux données

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai appris à développer des composants d'accès aux données avec le langage SQL. Pour comprendre et maîtriser le langage SQL, j'ai commencé par les requêtes simples de LID (Langage d'Interrogation des Données) qui appartient au LMD (Langage de Manipulation des données). Pour pratiquer, le schéma Northwind très réputé a été utilisé. L'une des premières requêtes s'est faite via un SELECT pour obtenir les lignes d'une table.

Ensuite, j'ai commencé le LMD en utilisant les UPDATE, INSERT, DELETE. Après cela j'étais capable d'interroger une base de données mais aussi d'en manipuler les données.

Pour le LMD et le LID, j'ai appris à maîtriser le langage SQL en faisant des requêtes précises. Ces requêtes ont été faites avec les mots-clés WHERE, GROUP BY, HAVING, ORDER BY. Pour avoir la meilleure précision de nos requêtes, j'ai appris à utiliser les différentes options telles que DISTINCT, AS, LIKE, etc ainsi que différentes fonctions telles que MIN, MAX, COUNT, SUM, ROUND, etc. J'ai aussi appris à interroger plusieurs tables via des jointures, tout en utilisant les connaissances précédemment acquises. (voir annexe 3).

Pour maîtriser les requêtes SQL avancées, j'ai appris grâce aux opérateurs ensemblistes UNION, etc les sous-requêtes ainsi que les sous-requêtes synchronisées.

J'ai appris aussi AJAX qui permet le chargement de parties de page à l'interaction d'utilisateur. Pour faire cela, j'ai manipulé l'objet XMLHttpRequest. Cet Objet se manipule en plusieurs étapes indispensables, et s'analyse en fonction de plusieurs états. (voir annexe 4).

DOSSIER PROFESSIONNEL (DP)



2. Précisez les moyens utilisés :

MySQL, l'outil Workbench, le schéma NORTHWIND, AJAX.

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec le formateur Lesly, ainsi qu'avec les différents camarades de la formation

4. Contexte

Nom de l'entreprise, organisme ou association ►

Greta de Compiègne

5. Informations complémentaires *(facultatif)*

Activité- type 2

Exemple n°1

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

Concevoir une base de données

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Concernant la conception d'une base de données, j'ai commencé par l'approche Merise, pour comprendre le principe d'entité/relation afin de concevoir un modèle de données. J'ai travaillé sur un projet interne pour concevoir la base de données et ensuite la mettre en place.

Dans un premier temps, j'ai appris le modèle conceptuel de données. Cela va nous permettre de comprendre les différentes entités existantes et connaître les différentes relations entre-elles. J'ai compris aussi l'importance des cardinalités des relations. A ce stade, je connais les entités avec les attributs, et l'identifiant de l'entité, les relations, et les cardinalités.

Après l'analyse conceptuelle, j'ai pu ressortir les données utilisables pour un SI et ainsi d'en déduire le modèle logique de données. J'ai aussi appris à établir un dictionnaire de données pour avoir une vue précise et complète des données à utiliser. J'ai aussi appris les notions avancées telles que l'association porteuse, l'association réflexive. (voir annexe 5).

Ce modèle logique de données nous permet d'établir le modèle physique de données. Celui-ci contient en plus du modèle logique l'aspect technique de la base de données utilisée. J'ai appris la logique des clés primaires qui généralement sont l'identifiant défini dans le modèle conceptuel de données. J'ai aussi appris les clés étrangères qui permettent de joindre les différentes entités. De plus dans ce modèle physique, nous précisons le type de donnée de chaque table.

DOSSIER PROFESSIONNEL (DP)



2. Précisez les moyens utilisés :

Les différents modèles de données, le dictionnaire de données.

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec le formateur Lesly, ainsi qu'avec les différents camarades de la formation

4. Contexte

Nom de l'entreprise, organisme ou association ►

Greta de Compiègne

5. Informations complémentaires *(facultatif)*

Activité- type 2

Exemple n°2 ▶

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

Mettre en place une base de données

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Après avoir conçu la base de données du projet interne, j'ai appris à la mettre en place. J'ai appris à la mettre en place via le langage SQL. Pour cela j'ai utilisé le LDD (Langage de définition de données).

Premièrement, il est nécessaire de créer la base de données à l'aide d'un `CREATE DATABASE` en précisant le type de caractère utilisé, les options de la base de données telles que la sensibilité à la casse, etc. (voir annexe 6)

Ensuite, je peux créer les différentes tables à l'aide d'un `CREATE TABLE` en précisant les différentes colonnes avec les types de données, et les options telles que l'auto-incrémentation, si ça peut être « null », etc. Après il est possible d'ajouter différentes contraintes telles que des clés étrangères, la validité des données insérées, etc. Il est nécessaire aussi de préciser le type de moteur que la table va utiliser. (voir annexe 7).

Pour la mise en place, j'ai pu apprendre à gérer les différentes options telles que `IF NOT EXISTS`, etc.

Durant toute la mise en place de la base de données, nous avons vu les différentes syntaxes permettant d'obtenir le même résultat. Même si certaines syntaxes sont conseillées, nous avons pu apprendre les différences entre-elles pour ainsi avoir une meilleure compatibilité entre les bases de données de type SQL.

2. Précisez les moyens utilisés :

Les différents modèles de données, MySQL, l'outil Workbench, le langage SQL.

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec le formateur Lesly, ainsi qu'avec les différents camarades de la formation.

4. Contexte

Nom de l'entreprise, organisme ou association ▶

Greta de Compiègne

5. Informations complémentaires (facultatif)

Activité- type 3

Exemple n°1 ▶

Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Concevoir une application

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pour apprendre à concevoir une application j'ai travaillé sur un projet de bibliothèque.

Le principe de ce projet est de maîtriser le mode de pensée orientée objet, et ainsi bien comprendre les principes de base. Après avoir compris les principes de ce qu'est un objet, j'ai appris comment se manipule un objet, et comment différents objets peuvent interagir dans une application.

Pour comprendre et maîtriser la conception d'une application, j'ai appris les bases de l'algorithmie dans un mode de pensée orientée objet. (voir annexe 8)

De plus, pour avoir un point de vue plus large, j'ai appris la conception d'une application à travers le langage UML (Unified Modeling Language).

Durant ces cours, j'ai appris les différents diagrammes qui composent UML tels que le diagramme de cas d'utilisation, des cas d'utilisation détaillés, de séquences, d'activités, etc. Ce langage m'a permis de mieux comprendre les mécanismes d'une application conçu en mode de pensée orientée objet.

J'ai appris aussi les principes et les règles de l'architecture MVC (Modèle Vue Contrôleur). Grâce aux cours sur cette architecture, j'ai mieux compris le fonctionnement d'une application multicouche répartie.

Suite à cela, j'ai pu concevoir d'autres applications dans le cadre de projets internes comme le projet gé- rant une banque, un site vitrine (portail), un site e-commerce. L'ensemble s'est fait sur des langages ou des bibliothèques différentes ce qui a permis de mettre en œuvre différemment la conception par rap- port à chaque projet.

2. Précisez les moyens utilisés :

Cahier des charges des différents projets, Visual Paradigm Online et Community pour l'élaboration des diagrammes UML, les principes de structure des langages.

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec le formateur Denis et Soupramanien, ainsi qu'avec les différents camarades de la formation.

4. Contexte

Nom de l'entreprise, organisme ou association ▶

Sesame IT

5. Informations complémentaires (facultatif)

Activité- type 3

Exemple n°2

Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Préparer et exécuter les plans de test d'une application

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Lors des différents projets menés pendant toute l'année, les différents formateurs nous ont tous sensibilisés à l'importance des différents tests à exécuter pour une application.

J'ai appris qu'il était nécessaire de préparer et exécuter différents type de tests dont chacun ont leur importance mais aussi leur utilité pour connaître les forces et les faiblesses de l'application développée.

Pour mettre en pratique les différentes phases de tests, j'ai pu pratiquer lors du développement du site e-commerce. Dans un premier temps, je devais définir par quel moyen les tests vont être menés. Dans le cadre de ce projet qui était programmé en JAVA, j'ai utilisé le framework de test JUNIT.

Les premiers tests réalisés sont des tests unitaires qui permettent de contrôler le comportement d'une méthode. En envoyant le paramètre attendu pour la méthode, le framework JUNIT détermine si le comportement est normal et renvoie le succès du test ou une erreur. J'ai exécuté aussi des tests fonctionnels en testant le scénario. (voir annexe 9).

A chaque nouvelle implémentation, il était indispensable de faire des tests de non régression pour savoir si du code est devenu non fonctionnel. Durant tout le développement du site e-commerce, des tests de non régression ont été exécutés.

2. Précisez les moyens utilisés :

Le framework de test JUNIT, l'IDE Eclipse.

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec le formateur Soupramanien, ainsi qu'avec les différents camarades de la formation.

4. Contexte

Nom de l'entreprise, organisme ou association

Sesame IT

DOSSIER PROFESSIONNEL (DP)



5. Informations complémentaires (facultatif)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

Déclaration sur l'honneur

Je soussigné(e) *Mr Guillaume WAROUX*

déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis l'auteur(e) des réalisations jointes.

Fait à *Gouvieux*

????

pour faire valoir ce que de droit.

Signature :

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé

Cliquez ici pour taper du texte.

Annexe 1 : Application Swing(modèle 'Banque')



Annexe 3 : Composants d'accès aux données

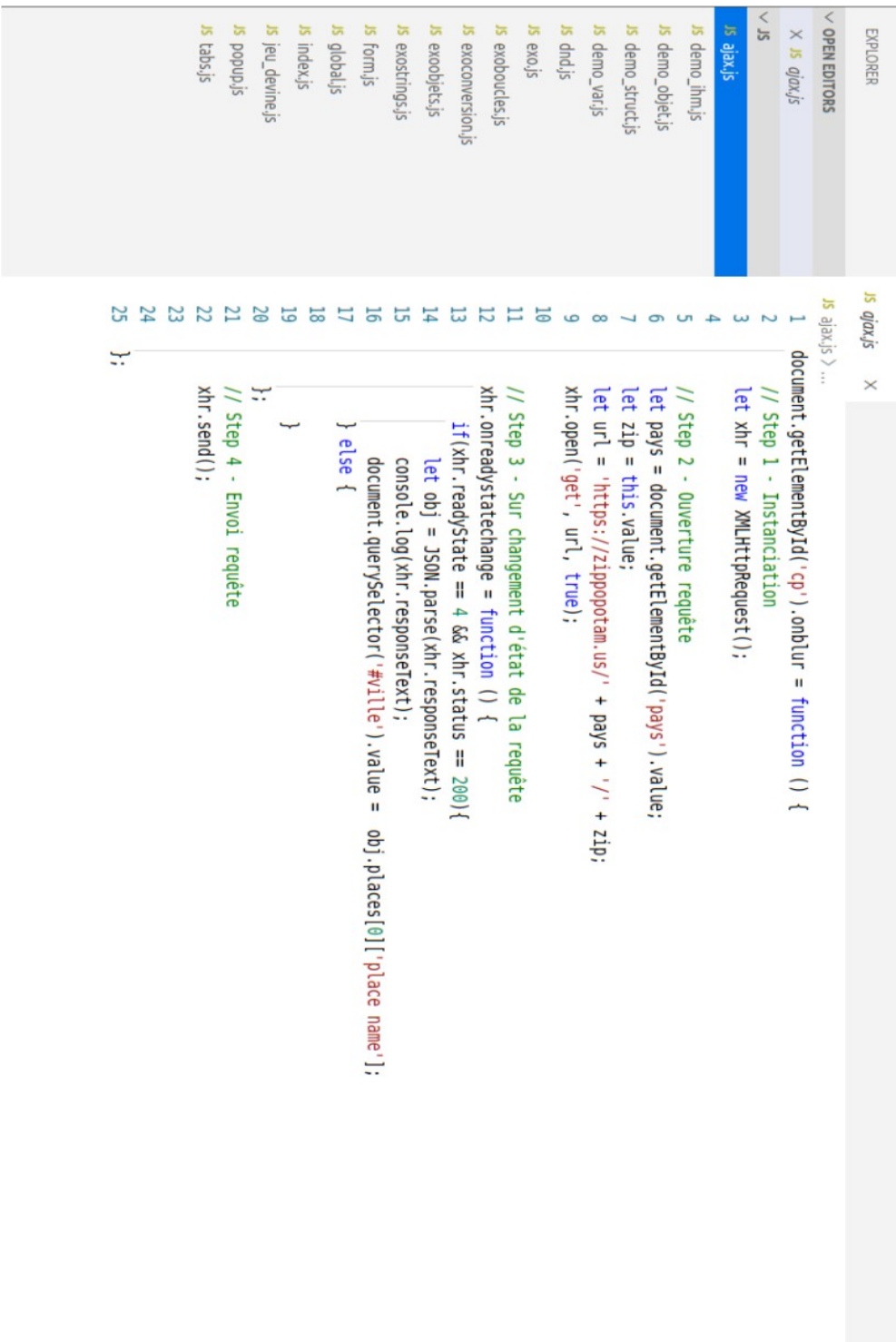
The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL query. The query is as follows:

```

217 ;
218 -- Atelier 2.5
219
220
221 • SELECT nom, prenom, fonction, salaire
222 FROM employees
223 WHERE salaire > 2500 AND salaire < 3500
224 ;
225
226 • SELECT p.nom, produit, f.societe, c.nom_categorie, p.quantite
227 FROM produits p
228 INNER JOIN fournisseurs f
229 ON p.no_fournisseur = f.no_fournisseur
230 INNER JOIN categories c
231 ON p.code_categorie = c.code_categorie
232 WHERE p.code_categorie NOT IN(1, 3, 5, 7) AND (p.quantite LIKE '%boites%' OR p.quantite LIKE '%cartons%')
233 ;
234
235 • SELECT p.nom, produit, f.societe, c.nom_categorie, p.quantite, c.code_categorie
236 FROM produits p
237 INNER JOIN fournisseurs f
238 ON p.no_fournisseur = f.no_fournisseur
239 INNER JOIN categories c
240 ON p.code_categorie = c.code_categorie
241 WHERE (f.no_fournisseur BETWEEN 1 AND 3 OR c.code_categorie BETWEEN 1 AND 3)
242 ;
243
244 • SELECT e.nom
245 FROM commandes a
246 INNER JOIN clients c
247 ON a.code_client = c.code_client
248 INNER JOIN employes e
249 ON a.no_employe = e.no_employe
250 WHERE UPPER(c.ville) = 'PARIS'
251 ;
  
```

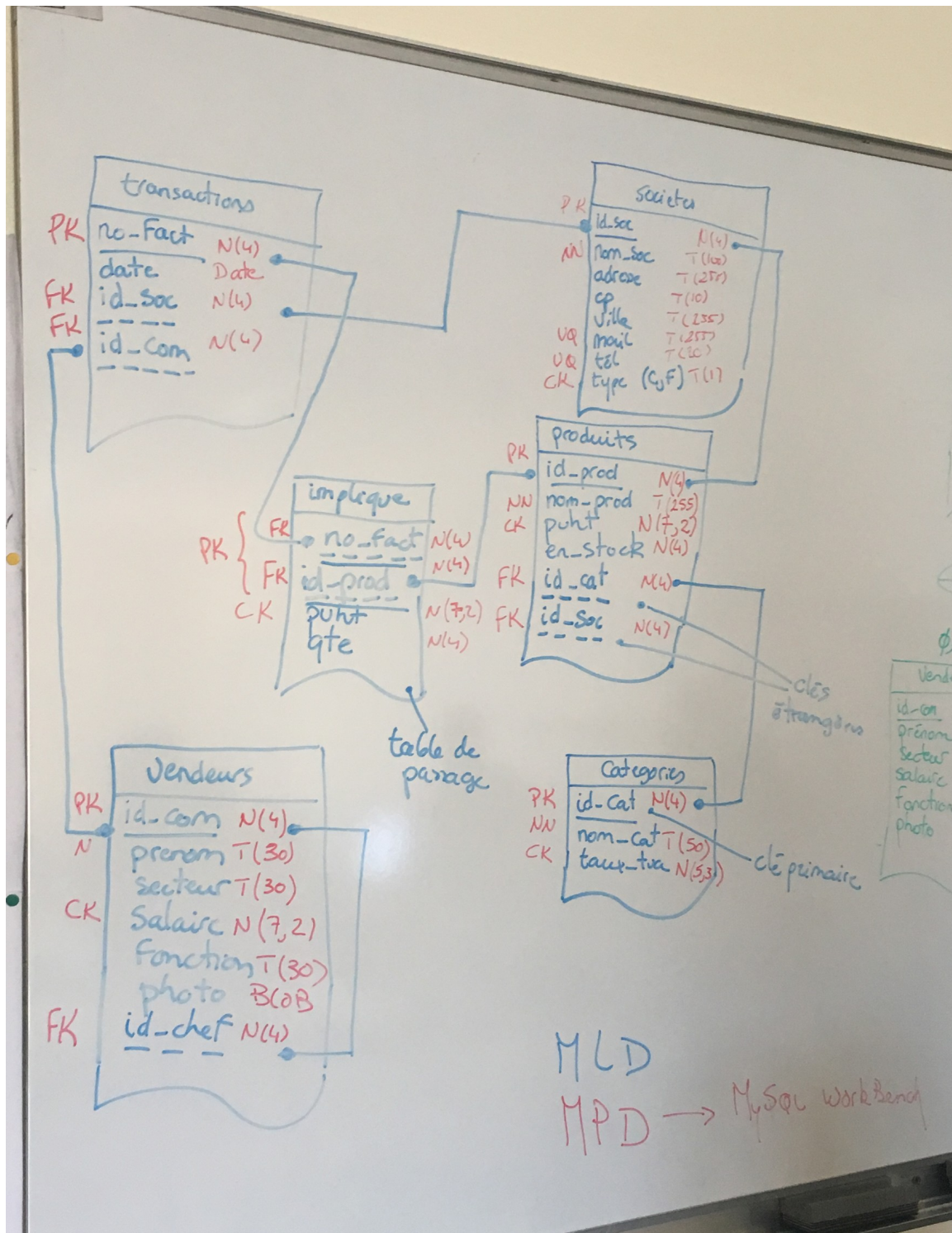
The interface also shows a sidebar on the left with a tree view of database objects (Tables, Views, Stored Procedures, Functions, etc.) and a top menu bar with options like File, Edit, View, Query, Database, Server, Tools, Scripting, and Help.

Annexe 4 : AJAX



```
1 document.getElementById('cp').onblur = function () {
2     // Step 1 - Instanciation
3     let xhr = new XMLHttpRequest();
4
5     // Step 2 - Ouverture requête
6     let pays = document.getElementById('pays').value;
7     let zip = this.value;
8     let url = 'https://zipopotam.us/' + pays + '/' + zip;
9     xhr.open('get', url, true);
10
11     // Step 3 - Sur changement d'état de la requête
12     xhr.onreadystatechange = function () {
13         if (xhr.readyState == 4 && xhr.status == 200) {
14             let obj = JSON.parse(xhr.responseText);
15             console.log(xhr.responseText);
16             document.querySelector('#ville').value = obj.places[0]['place name'];
17         } else {
18         }
19     };
20
21     // Step 4 - Envoi requête
22     xhr.send();
23
24
25 };
```

Annexe 5 : Conception d'une base de données



Annexe 6 : Mettre en place une base de données 1

```
Query 1 create_schema_greta
-- Suppression du schéma/database
2 • DROP SCHEMA IF EXISTS greta
3 ;
4
5 -- Création du schéma/database si le schéma n'existe pas
6 • CREATE SCHEMA IF NOT EXISTS greta
7   CHARACTER SET=utf8
8   COLLATE=utf8_general_ci
9 ;
10 -- Activation BDD
11 • use greta;
12 -- Suppression et création de la table ETUDIANT
13 • DROP TABLE IF EXISTS etudiant
14 ;
15
16 • CREATE TABLE IF NOT EXISTS etudiant(
17   code_etu INTEGER AUTO_INCREMENT PRIMARY KEY,
18   nom_etu VARCHAR(30) NOT NULL,
19   ddn_etu DATE,
20   sexe_etu ENUM('F', 'M', 'N')
21 )ENGINE=InnoDB;
22
23 • ALTER TABLE etudiant
24   MODIFY COLUMN nom_etu VARCHAR(35),
25   ADD COLUMN bourse_etu BOOLEAN,
26   ADD COLUMN chef_etu INTEGER,
27   ADD CONSTRAINT etudiant_chef_fk FOREIGN KEY (chef_etu) REFERENCES etudiant(code_etu)
28 ;
29
```

Annexe 7 : Mettre en place une base de données 2

```
Query 1 create_schema_greta
-- Création table MATIERE
40 CREATE TABLE matiere(
41     code_mat INTEGER AUTO_INCREMENT,
42     nom_mat VARCHAR(30) NOT NULL,
43     coeff_mat TINYINT DEFAULT 1,
44     code_ens INTEGER,
45     CONSTRAINT matiere_code_pk PRIMARY KEY (code_mat),
46     CONSTRAINT matiere_coef_ck CHECK (coeff_mat BETWEEN 1 AND 5), --> MySQL v8
47     CONSTRAINT matiere_code_ens_fk FOREIGN KEY (code_ens) REFERENCES enseignant(code_ens)
48 )ENGINE=InnoDB
49 ;
50
51 -- Le primary key à 2 arguments permet de créer une PK à partir de 2 FK
52 CREATE TABLE IF NOT EXISTS etudie(
53     code_etu INTEGER,
54     code_mat INTEGER,
55     note DECIMAL(4,2),
56     CONSTRAINT etudie_note_ck CHECK (note BETWEEN 0 AND 20),
57     CONSTRAINT etudie_code_etu_fk FOREIGN KEY (code_etu) REFERENCES etudiant(code_etu),
58     CONSTRAINT etudie_code_mat_fk FOREIGN KEY (code_mat) REFERENCES matiere(code_mat),
59     CONSTRAINT etudie_code_pk PRIMARY KEY(code_mat, code_etu)
60 )ENGINE=InnoDB
61 ;
62
63
64
```

Annexe 8 : Conception d'une application

Classe Livre :

Nom : Livre

Rôle : Permet de créer un objet Livre pour être intégré dans la bibliothèque.

Entrées : Les informations du livre à créer.

Sorties : L'objet instancié contenant les valeurs de l'utilisateur.

Variables d'instance : titre, auteur, isbn, dispo.

Début

```
public class Livre {  
    // Atributs  
    titre de type String ;  
    auteur de type String ;  
    isbn de type Integer ;  
    dispo de type booléen ;  
  
    // Constructeur sans paramètre  
    public Livre()  
    {  
        this(« », « », 0, false) ;  
    }  
  
    // Constructeur avec paramètres  
    public Livre(String titre, String auteur, int isbn, boolean dispo)  
    {  
        this.titre = titre ;  
        this.auteur = auteur ;  
        this.isbn = isbn ;  
        this.dispo = dispo ;  
    }  
  
    // Accesseurs : getters et setters  
    public qui renvoie un String getTitre()  
    {  
        return titre ;  
    }  
  
    public qui modifie la valeur setTitre(String titre)  
    {  
        this.titre = titre ;  
    }  
}
```

Annexe 9 : Préparer et exécuter les plans de tests de l'application

```
1 package fr.greta00.controller;
2
3 import static org.assertj.core.api.Assertions.assertThat;
4 @SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT,
5 classes = SpringSecurityWebTestConfig.class)
6
7 @AutoConfigureMockMvc
8 // @WebMvcTest(controllers = ProductController.class)
9 @TestPropertySource(locations = "classpath:applicationTest.properties")
10 public class ProductControllerTest {
11
12     @MockBean
13     ProductService productService;
14
15     @MockBean
16     CategoriesService catService;
17
18     @Autowired
19     private MockMvc mockMvc;
20
21     @Test
22     void produitListDoitRetourner200() throws Exception {
23         mockMvc.perform(get("/produit/list"))
24             .andExpect(status().isOk());
25     }
26
27     @Test
28     void produitListDoitRetournerProduitList() throws Exception {
29         MvcResult result = mockMvc.perform(get("/produit/list"))
30             .andExpect(status().isOk())
31             .andReturn();
32         ModelAndView modelAndView = result.getModelAndView();
33         String viewName = modelAndView.getViewName();
34         assertThat(viewName).isEqualTo("produit/list");
35     }
36
37     @WithAnonymousUser
38     @Test
39     void ajouterProduitDoitRetourner302() throws Exception {
40         mockMvc.perform(get("/admin/produit/add"))
41             .andExpect(status().is3xxRedirection());
42     }
43
44     @WithUserDetails("user")
45     @Test
46 }
```