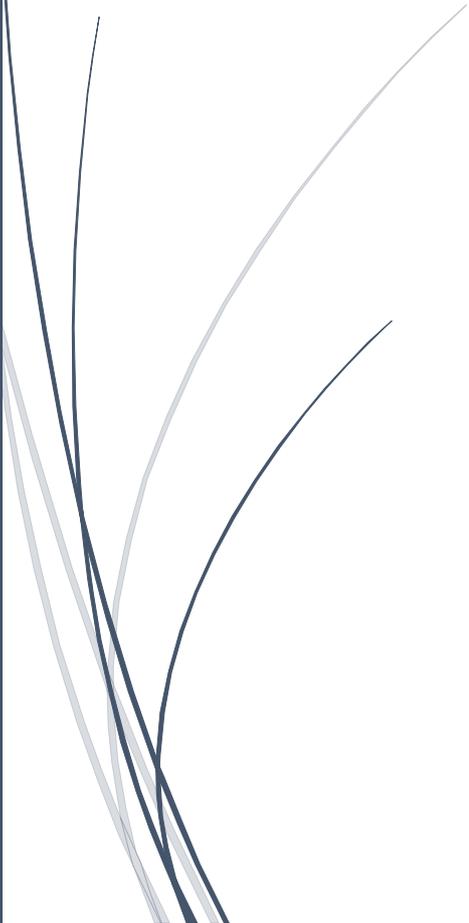




22/04/2025

SAE203

Mettre en place une solution
informatique pour l'entreprise -
Culture/Tourisme



Esteban Fernandez, Gaëtan Richaud
IUT DE BLOIS

Sommaire

Introduction.....	2
Partie 1 : Création des fichiers CSV pour chaque table	3
1.1 Introduction au Modèle Logique de Données (MLD)	3
1.2 Création des fichiers CSV pour chaque table	3
1.3 Détails de la création de chaque table.....	4
Table Site_Touristique	4
Table Langues	5
Table Visite.....	5
Table Point_Acces.....	6
Table Theme_Site	7
Table Services	7
Table Equipements	8
Table Visite_Langue.....	9
Table Equipements_par_services	9
Partie 2 : Création des Tables sur Oracle SQL à partir des fichiers CSV	10
Partie 3 : Description des tables avec la commande « DESCRIBE ».....	14
SITE_TOURISTIQUE	14
LANGUES.....	14
VISITE	15
POINT_ACCES	15
THEME_SITE	16
SERVICES.....	16
EQUIPEMENTS.....	16
VISITE_LANGUE	17
EQUIPEMENTS_PAR_SERVICES.....	17
Partie 4 : 5 requêtes pertinentes.....	18
Conclusion	20
Table des illustrations	21

Introduction

Dans ce projet, nous avons conçu et normalisé une base de données pour un système de gestion de visites touristiques, comprenant plusieurs entités interconnectées, telles que les sites touristiques, les visites, les langues parlées et les services. L'objectif était de structurer ces informations de manière cohérente et efficace à l'aide de la normalisation des données.

Le thème que nous avons choisi est le Culture/Tourisme. L'objectif de notre base de données est de donner un aperçu à l'utilisateur des différents sites touristique, en incluant les équipements et services de chacune, leurs caractéristiques, que cela soit au niveau de l'architecture ou de la langue parlé par exemple.

Partie 1 : Création des fichiers CSV pour chaque table

1.1 Introduction au Modèle Logique de Données (MLD)

Le Modèle Logique de Données (MLD) est une représentation de la structure de notre base de données, qui illustre les relations entre les différentes tables et comment elles interagissent entre elles. Le MLD est un outil essentiel pour visualiser les données et comprendre leur organisation dans une base de données relationnelle.

Dans ce projet, le MLD a joué un rôle crucial en nous permettant de définir les relations entre les entités (comme les sites touristiques, les visites, les services, et les langues), d'assurer l'intégrité des données, et de garantir que chaque table et chaque relation entre elles soit correctement représentée.

Voici l'image du MLD que nous avons conçu pour ce projet, illustrant les principales tables et leurs relations :

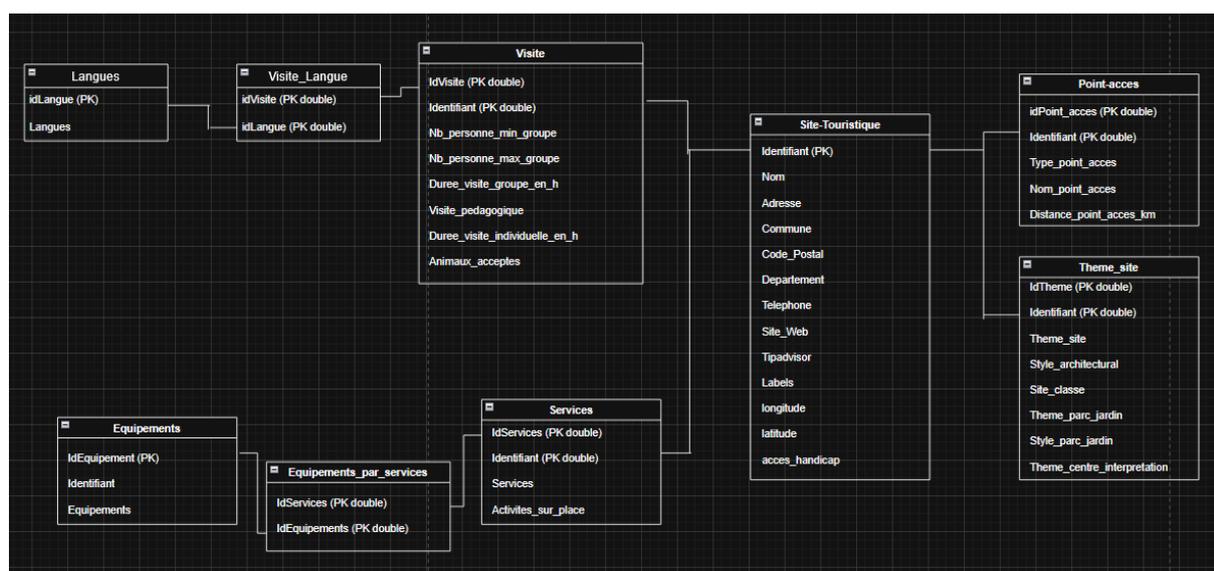


Figure 1 : Modèle logique de données

1.2 Création des fichiers CSV pour chaque table

Après avoir défini le MLD, nous avons créé les fichiers CSV pour chaque table afin de structurer et organiser les données avant leur insertion dans la base de données Oracle. Ces fichiers CSV ont été triés, nettoyés, et préparés pour garantir que seules les informations pertinentes et cohérentes soient insérées dans les tables.

Les fichiers CSV contiennent des données sur :

- Les sites touristiques (nom, adresse, localisation, etc.).
- Les langues associées aux visites.
- Les visites avec leurs détails spécifiques (durée, nombre de personnes, etc.).
- Les services et équipements disponibles pour chaque site.

1.3 Détails de la création de chaque table

Table Site_Touristique

La table Site_Touristique est la table principale de notre base de données, contenant des informations sur les sites touristiques, telles que le nom, l'adresse, les coordonnées géographiques, et d'autres informations pratiques comme le téléphone, le site web et l'accessibilité handicap. Cette table a été normalisée pour garantir que chaque site soit identifié de manière unique grâce à la clé primaire Identifiant.

La structure de la table est simple et bien organisée, et elle sert de base pour lier les autres tables grâce à des clés étrangères. Cette table est également utilisée pour connecter toutes les autres tables comme Point_Acces, Visite, Services, etc.

Voici le fichier CSV cliquable :

Adresse	Commune
Impasse du Clos Saint Georges	THESEE
10 Boulevard Carnot	BLOIS
73 Rue Nationale	SAINT-DYE-SUR-LOIRE
1 rue Henri Goyer	LECONTROIS-EN-SOLOGNE
18 rue du Château	TALCY
1 Hameau de Rochambeau	THORE-LA-ROCHETTE
	SARGE-SUR-BRAYE
	MONTHOU-SUR-CHER
Rampe du Château	VENDOME
La Possonnière	VALLEE-DE-RONSARD
La Prazerie	LUNAY
Avenue du Château	CHEVERNY
12 Chemin de la Fontaine	CELLETES

Table Langues

La table Langues contient les différentes langues disponibles pour les visites. Chaque langue est identifiée par un idLangue unique. Une particularité de cette table est l'ajout d'une entrée par défaut « Aucune Langue » avec idLangue = 1 pour les visites sans langue associée. Ce choix permet de traiter les visites sans langue définie, en garantissant la cohérence des données.

La normalisation de cette table a été réalisée en s'assurant qu'il n'y ait qu'une seule entrée pour chaque langue, permettant ainsi de minimiser la redondance. La table Langues est liée à la table Visite_Langue par la clé idLangue, facilitant la gestion des langues pour chaque visite.

Voici le fichier CSV cliquable :

idLangue	Langues				
1	Aucune Langues				
2	Allemand				
3	Anglais				
4	Espagnol				
5	Français				
6	Italien				
7	Russe				
8	Portugais				
9	Japonais				

Table Visite

La table Visite contient des informations sur chaque visite, comme le nombre de personnes minimum et maximum, la durée de la visite, la possibilité d'une visite pédagogique, et l'acceptation des animaux. Pour cette table, idVisite est une valeur unique afin de garantir l'intégrité des données. De plus, Identifiant fait référence à la table Site_Touristique, ce qui permet d'associer chaque visite à un site spécifique.

Une clé étrangère a été ajoutée pour faire le lien entre Identifiant dans Visite et Identifiant dans Site_Touristique, assurant ainsi une relation cohérente entre ces deux entités. La table Visite est liée à Visite_Langue via idVisite, permettant de gérer les langues associées à chaque visite.

Voici le fichier CSV cliquable :

idVisite	Identifiant	Nb_personne_min_groupe	Nb_personne_max_groupe
1	PCU41AASOR100001		200
2	PCU41AASOR100025		
3	PCU41AASOR100031	12	30
4	PCU41AASOR100039	20	
5	PCU41AASOR100040	20	45
6	PCU41AASOR100120	15	
7	PCU41AASOR100122	10	30
8	PCU41AASOR100123	20	
9	PCU41AASOR100125	15	

Table Point_Acces

La table Point_Acces est la première table où des défis de normalisation ont été rencontrés. Initialement, l'Identifiant seul n'était pas suffisant pour identifier de manière unique un point d'accès, car un site pouvait avoir plusieurs points d'accès de types différents. Pour résoudre ce problème, nous avons introduit une clé primaire composée de Identifiant et idPoint_acces, garantissant ainsi l'unicité des points d'accès pour chaque site.

De plus, une clé étrangère a été ajoutée pour relier la table Point_Acces à la table Site_Touristique via Identifiant, assurant l'intégrité des données entre ces deux tables.

Voici le fichier CSV cliquable :

idPoint_acces	Identifiant	Type_point_acces	Nom_point_acces
1	PCU41AASOR100001		
2	PCU41AASOR100025	Autoroute	A 10, sortie n° 17
3	PCU41AASOR100025	Gare Routière	Blois (environ)
4	PCU41AASOR100031	Autoroute	A 10, sortie n° 16
5	PCU41AASOR100031	Gare Routière	Mer
6	PCU41AASOR100039	Autoroute	A 10, sortie n° 17
7	PCU41AASOR100039	Gare Routière	Blois (environ)
8	PCU41AASOR100040		
9	PCU41AASOR100120	Gare Ferroviaire	Blois

Table Theme_Site

La table Theme_Site contient des informations sur les thèmes associés à chaque site touristique, tels que le style architectural, les jardins, et les centres d'interprétation. Cette table utilise une clé primaire composée de Identifiant et idTheme pour garantir que chaque thème soit unique pour chaque site touristique.

La normalisation de cette table a permis de gérer de manière efficace les multiples thèmes possibles pour un site. La table Theme_Site est liée à Site_Touristique par Identifiant, ce qui permet de lier chaque thème à un site spécifique.

Voici le fichier CSV cliquable :

12	PCU41AASOR100127
13	PCU41AASOR100128
14	PCU41AASOR100129
15	PCU41AASOR100130
16	PCU41AASOR100131
17	PCU41AASOR100132
18	PCU41AASOR100134
19	PCU41AASOR100135
20	PCU41AASOR100136
21	PCU41AASOR100137

Table Services

La table Services décrit les différents services offerts par les sites touristiques, tels que les visites guidées, la restauration, ou d'autres activités sur place. Chaque service est identifié de manière unique par la clé primaire composée de Identifiant et idServices.

Cette table permet d'associer un service à plusieurs sites touristiques et de gérer les activités disponibles sur chaque site. La normalisation a permis de minimiser la redondance des données et de rendre la table plus flexible. La table Services est liée à la table Site_Touristique par Identifiant.

Voici le fichier CSV cliquable :

idServices	Identifiant	Services
1	PCU41AASOR100001	
2	PCU41AASOR100025	Boutique
3	PCU41AASOR100031	Boutique
4	PCU41AASOR100031	Boutique
5	PCU41AASOR100039	Boutique
6	PCU41AASOR100040	Boutique
7	PCU41AASOR100120	Boutique
8	PCU41AASOR100122	Boutique
9	PCU41AASOR100123	Boutique

Table Equipements

La table Equipements contient les équipements disponibles dans chaque site touristique, comme les casques audio ou les tablettes interactives. Chaque équipement est identifié de manière unique par idEquipements.

Les équipements peuvent être associés à des services via la table de liaison Equipements_par_services, permettant de gérer les relations plusieurs-à-plusieurs entre les Services et les Equipements. La normalisation a permis d'éviter toute duplication des équipements et de garantir leur association correcte aux services.

Voici le fichier CSV cliquable :

idEquipements	Identifiant
1	PCU41AASOR100001
2	PCU41AASOR100025
3	PCU41AASOR100031
4	PCU41AASOR100031
5	PCU41AASOR100039
6	PCU41AASOR100040
7	PCU41AASOR100120
8	PCU41AASOR100122
9	PCU41AASOR100123

Table Visite_Langue

La table Visite_Langue est une table de liaison entre les Visite et les Langues, permettant de gérer les relations plusieurs-à-plusieurs entre ces deux entités. La clé primaire composée de idVisite et idLangue garantit que chaque combinaison de visite et langue est unique.

Deux clés étrangères ont été ajoutées : l'une pour faire référence à idVisite dans la table Visite, et l'autre pour faire référence à idLangue dans la table Langues. Cela permet de maintenir l'intégrité des données et de lier chaque visite à une ou plusieurs langues.

Voici le fichier CSV cliquable :

idVisite	idLangue				
1	1				
2	2				
2	3				
2	4				
2	5				
3	3				
3	4				
3	5				
4	3				

Table Equipements_par_services

La table Equipements_par_services gère la relation entre les Services et les Equipements, en permettant à un service d'être associé à plusieurs équipements. Cette table utilise une clé primaire composée de idServices et idEquipements pour garantir l'unicité de chaque relation.

Deux clés étrangères ont été définies : l'une pour relier idServices à la table Services et l'autre pour relier idEquipements à la table Equipements. Cela permet de maintenir l'intégrité des données et de gérer les équipements associés à chaque service sans duplication.

Voici le fichier CSV cliquable :

idServices	idEquipements				
1	1				
2	2				
3	3				
4	4				
5	5				
6	6				
7	7				
8	8				
9	9				

Partie 2 : Création des Tables sur Oracle SQL à partir des fichiers CSV

Grâce aux fichiers CSV préparés, nous avons pu établir la liaison entre les données et créer les tables relationnelles dans notre base de données Oracle. Chaque fichier CSV a été transformé en une table SQL, respectant les relations et contraintes définies dans notre Modèle Logique de Données (MLD). Les tables ont été construites en utilisant les codes SQL suivants.

Les requêtes `CREATE TABLE` suivantes ont permis de définir la structure des tables dans la base de données et de garantir l'intégrité des données par l'utilisation de clés primaires et clés étrangères pour maintenir les relations entre les entités.

```
CREATE TABLE SITE_TOURISTIQUE (
    Identifiant VARCHAR2(20),
    Nom VARCHAR2(100),
    Adresse VARCHAR2(60),
    COMMUNE VARCHAR2(50),
    Code_postal NUMBER(5),
    Departement VARCHAR2(15),
    Telephone VARCHAR2(50),
    Site_web VARCHAR2(250),
    Tripadvisor VARCHAR2(200),
    Labels VARCHAR2(60),
    Latitude VARCHAR2(12),
    Longitude VARCHAR2(12),
    acces_handicap VARCHAR2(3));

CREATE TABLE POINT_ACCES (
    idPoint_acces NUMBER(4),
    Identifiant VARCHAR2(20),
    Type_point_acces VARCHAR2(20),
    Nom_point_acces VARCHAR2(50),
    Distance_point_acces_km VARCHAR2(5),
    PRIMARY KEY (Identifiant, idPoint_acces),
    FOREIGN KEY (Identifiant) REFERENCES
    site_touristique(Identifiant));

CREATE TABLE THEME_SITE (
    idTheme NUMBER(4),
    Identifiant VARCHAR2(20),
    Theme_site VARCHAR2(150),
    Style_architectural VARCHAR2(75),
    Site_Classe VARCHAR2(175),
```

```

Theme_parc_jardin VARCHAR2(100),
Style_parc_jardin VARCHAR2(100),
Theme_centre_interpretation VARCHAR2(75),
PRIMARY KEY (Identifiant, idTheme),
FOREIGN KEY (Identifiant) REFERENCES
site_touristique(Identifiant));

```

```

CREATE TABLE SERVICES (
    idServices NUMBER(4),
    Identifiant VARCHAR2(20),
    Services VARCHAR2(125),
    Activites_sur_place VARCHAR2(75),
    PRIMARY KEY (Identifiant, idServices),
    FOREIGN KEY (Identifiant) REFERENCES
site_touristique(Identifiant));

```

```

CREATE TABLE EQUIPEMENTS (
    idEquipements NUMBER(4),
    Identifiant VARCHAR2(20),
    Equipements VARCHAR2(200),
    PRIMARY KEY (idEquipements));

```

```

CREATE TABLE Equipements_par_services (
    idServices NUMBER(4),
    idEquipements NUMBER(4),
    PRIMARY KEY (idServices, idEquipements),
    FOREIGN KEY (idServices) REFERENCES Services(idServices),
    FOREIGN KEY (idEquipements) REFERENCES
Equipements(idEquipements));

```

```

CREATE TABLE LANGUES (
    idLangue NUMBER(4) PRIMARY KEY,
    Langues VARCHAR2(25));

```

```

CREATE TABLE VISITE (
    idVisite NUMBER(4),
    Identifiant VARCHAR2(20),
    Nb_personne_min_groupe NUMBER(5),
    Nb_personne_max_groupe NUMBER(5),
    Duree_visite_groupe_en_h VARCHAR2(6),

```

```
Visite_pedagogique VARCHAR2(3),
Duree_visite_individuelle_en_h VARCHAR2(6),
Animaux_acceptes VARCHAR2(3),
PRIMARY KEY(idVisite, Identifiant),
FOREIGN KEY (Identifiant) REFERENCES
site_touristique(Identifiant));
```

```
CREATE TABLE VISITE_LANGUE (
    idVisite NUMBER(4),
    idLangue NUMBER(4),
    PRIMARY KEY (idVisite, idLangue),
    FOREIGN KEY (idVisite) REFERENCES Visite(idVisite),
    FOREIGN KEY (idLangue) REFERENCES Langues(idLangue));
```

Les PRIMARY KEY on servit à initier les clés primaires et FOREIGN KEY on servit à initier les clés étrangères.

Ensuite j'ai peuplé les données de chaque table comme ceci :

Tout d'abord je sélectionne la table et je vais sur importer les données :

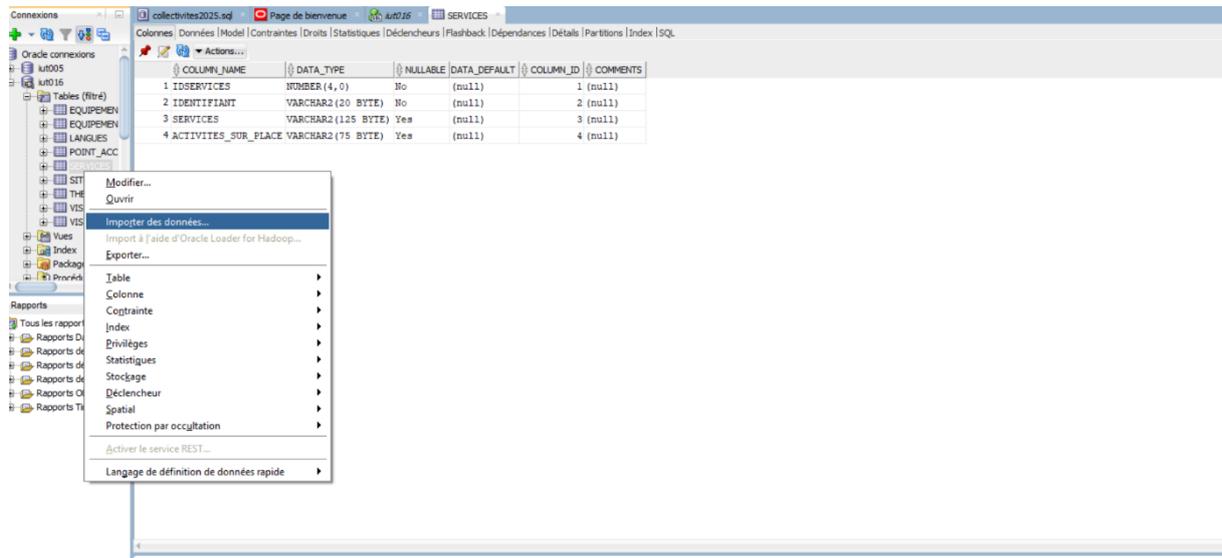


Figure 2 : Importer les données

Ensuite, on met en place le fichier, on choisit l'encodage UTF-8 et en délimiteur on met « ; » :

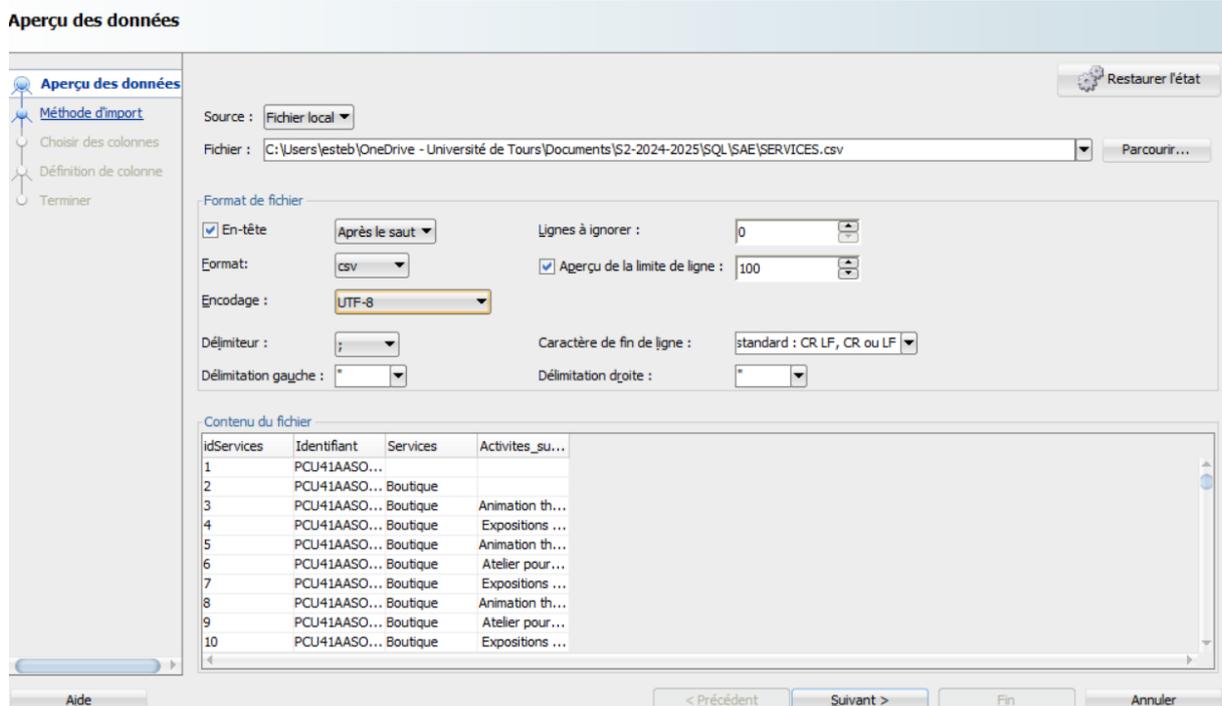


Figure 3 : Aperçu des données

Enfin on fait suivants jusqu'à arriver sur « fin » et les données sont bel et bien implémenté dans la table.

Partie 3 : Description des tables avec la commande « DESCRIBE »

La commande DESCRIBE en SQL sert à afficher la structure d'une table : elle montre les colonnes, leurs types de données, si elles peuvent être nulles, les clés, etc. C'est un moyen rapide de voir la définition d'une table.

SITE_TOURISTIQUE

On utilise la commande : DESCRIBE SITE_TOURISTIQUE;

Nom	NULL ?	Type
IDENTIFIANT	NOT NULL	VARCHAR2 (20)
NOM		VARCHAR2 (100)
ADRESSE		VARCHAR2 (60)
COMMUNE		VARCHAR2 (50)
CODE_POSTAL		NUMBER (5)
DEPARTEMENT		VARCHAR2 (15)
TELEPHONE		VARCHAR2 (50)
SITE_WEB		VARCHAR2 (250)
TRIPADVISOR		VARCHAR2 (200)
LABELS		VARCHAR2 (60)
LATITUDE		VARCHAR2 (12)
LONGITUDE		VARCHAR2 (12)
ACCES_HANDICAP		VARCHAR2 (3)

Figure 4 : Description de la table SITE_TOURISTIQUE

LANGUES

On utilise la commande : DESCRIBE LANGUES;

Nom	NULL ?	Type
IDLANGUE	NOT NULL	NUMBER (4)
LANGUES		VARCHAR2 (25)

Figure 5 : Description de la table LANGUES

VISITE

On utilise la commande : `DESCRIBE VISITE;`

Nom	NULL ?	Type
IDVISITE	NOT NULL	NUMBER(4)
IDENTIFIANT	NOT NULL	VARCHAR2(20)
NB_PERSONNE_MIN_GROUPE		NUMBER(5)
NB_PERSONNE_MAX_GROUPE		NUMBER(5)
DUREE_VISITE_GROUPE_EN_H		VARCHAR2(6)
VISITE_PEDAGOGIQUE		VARCHAR2(3)
DUREE_VISITE_INDIVIDUELLE_EN_H		VARCHAR2(6)
ANIMAUX_ACCEPTES		VARCHAR2(3)

Figure 6 : Description de la table VISITE

POINT_ACCES

On utilise la commande : `DESCRIBE POINT_ACCES;`

Nom	NULL ?	Type
IDPOINT_ACCES	NOT NULL	NUMBER(4)
IDENTIFIANT	NOT NULL	VARCHAR2(20)
TYPE_POINT_ACCES		VARCHAR2(20)
NOM_POINT_ACCES		VARCHAR2(50)
DISTANCE_POINT_ACCES_KM		VARCHAR2(5)

Figure 7 : Description de la table POINT_ACCES

THEME_SITE

On utilise la commande : `DESCRIBE THEME_SITE;`

Nom	NULL ?	Type
IDTHEME	NOT NULL	NUMBER(4)
IDENTIFIANT	NOT NULL	VARCHAR2(20)
THEME_SITE		VARCHAR2(150)
STYLE_ARCHITECTURAL		VARCHAR2(75)
SITE_CLASSE		VARCHAR2(175)
THEME_PARC_JARDIN		VARCHAR2(100)
STYLE_PARC_JARDIN		VARCHAR2(100)
THEME_CENTRE_INTERPRETATION		VARCHAR2(75)

Figure 8 : Description de la table *THEME_SITE*

SERVICES

On utilise la commande : `DESCRIBE SERVICES;`

Nom	NULL ?	Type
IDSERVICES	NOT NULL	NUMBER(4)
IDENTIFIANT	NOT NULL	VARCHAR2(20)
SERVICES		VARCHAR2(125)
ACTIVITES_SUR_PLACE		VARCHAR2(75)

Figure 9 : Description de la table *SERVICES*

EQUIPEMENTS

On utilise la commande : `DESCRIBE EQUIPEMENTS;`

Nom	NULL ?	Type
IDEQUIPEMENTS	NOT NULL	NUMBER(4)
IDENTIFIANT		VARCHAR2(20)
EQUIPEMENTS		VARCHAR2(200)

Figure 10 : Description de la table *EQUIPEMENTS*

VISITE_LANGUE

On utilise la commande : `DESCRIBE VISITE_LANGUE;`

```
Nom          NULL ?   Type
-----
IDVISITE NOT NULL NUMBER(4)
IDLANGUE NOT NULL NUMBER(4)
```

*Figure 11 : Description de la table
VISITE_LANGUE*

EQUIPEMENTS_PAR_SERVICES

On utilise la commande : `DESCRIBE EQUIPEMENTS_PAR_SERVICES;`

```
Nom          NULL ?   Type
-----
IDSERVICES  NOT NULL NUMBER(4)
IDEQUIPEMENTS NOT NULL NUMBER(4)
```

*Figure 12 : Description de la table
EQUIPEMENTS_PAR_SERVICES*

Partie 4 : 5 requêtes pertinentes

Requête numéro 1 :

```
SELECT V.IDVISITE, L.LANGUES
FROM VISITE V, VISITE_LANGUE VL, LANGUES L
WHERE V.IDVISITE = VL.IDVISITE
AND VL.IDLANGUE = L.IDLANGUE;
```

Cette requête permet de voir les langues disponibles dans chaque site touristique.

Requête numéro 2 :

```
SELECT NOM, ADRESSE, ACCES_HANDICAP
FROM Site_Touristique
WHERE ACCES_HANDICAP = 'Oui';
```

Cette requête permet de trouver les sites touristiques ayant un accès handicapé.

Requête numéro 3 :

```
SELECT ST.NOM, V.VISITE_PEDAGOGIQUE, V.ANIMAUX_ACCEPTES
FROM Site_Touristique ST, VISITE V
WHERE ST.IDENTIFIANT = V.IDENTIFIANT
AND V.VISITE_PEDAGOGIQUE = 'Oui'
OR V.ANIMAUX_ACCEPTES = 'Oui';
```

Cette requête permet de trouver les sites touristiques ayant des visites pédagogiques ou acceptant les animaux de compagnies.

Requête numéro 4 :

```
SELECT ST.NOM, COUNT(S.IDSERVICES) AS NB_SERVICES
FROM Site_Touristique ST, SERVICES S
WHERE ST.IDENTIFIANT = S.IDENTIFIANT
GROUP BY ST.NOM;
```

Cette requête permet d'afficher le nombre de services disponibles par établissement.

Requête numéro 5 :

```
SELECT L.LANGUES, COUNT(VL.IDVISITE) AS NB_VISITES
FROM LANGUES L, VISITE_LANGUE VL
WHERE L.IDLANGUE = VL.IDLANGUE
GROUP BY L.LANGUES
HAVING COUNT(VL.IDVISITE) > 1;
```

Cette requête permet d'afficher le nombre de visites disponibles par langues dans la région.

Conclusion

Ce projet nous a permis de plonger concrètement dans la conception d'une base de données complète autour du thème du tourisme et de la culture. De la normalisation des fichiers CSV à l'intégration dans Oracle SQL, en passant par le Modèle Logique de Données, chaque étape nous a permis de mieux comprendre comment structurer des informations pour les rendre utiles, claires et bien organisées.

On a rencontré quelques défis (notamment avec les clés primaires composées ou la normalisation), mais ça nous a justement permis de mieux maîtriser les bonnes pratiques en base de données relationnelle. Grâce à ce travail, on dispose maintenant d'un système solide, capable de répondre à différents besoins utilisateurs : consulter les langues disponibles, les services proposés, ou encore les sites accessibles aux personnes en situation de handicap. Dans la suite de cette SAE, nous allons pouvoir exploiter ces données grâce à un site web dynamique.

Table des illustrations

Figure 1 : Modèle logique de données	3
Figure 2 : Importer les données	13
Figure 3 : Aperçu des données	13
Figure 4 : Description de la table SITE_TOURISTIQUE	14
Figure 5 : Description de la table LANGUES.....	14
Figure 6 : Description de la table VISITE	15
Figure 7 : Description de la table POINT_ACCES	15
Figure 8 : Description de la table THEME_SITE	16
Figure 9 : Description de la table SERVICES.....	16
Figure 10 : Description de la table EQUIPEMENTS	16
Figure 11 : Description de la table VISITE_LANGUE	17
Figure 12 : Description de la table EQUIPEMENTS_PAR_SERVICES.....	17