

TABLE DES MATIERES

INTRODUCTION.....	2
1. DESCRIPTION DU MODÈLE E/A.....	3
2. CONCEPTION DE LA BASE DE DONNÉES.....	4
2.1 Élaboration du dictionnaire de données.....	4
2.2 Diagramme E/A.....	5
2.3 Transformation vers le schéma relationnel.....	7
3. ALIMENTATION DE LA BASE DE DONNÉES.....	8
4. DESCRIPTION DES SCENARIOS XML/DTD.....	9

INTRODUCTION

L'objectif général du développement du système est de permettre au personnel de bureau d'achever rapidement et facilement les tâches de gestion des hôtels.

Dans ce projet on s'intéresse à l'étude et analyse de conception de l'application proposée pour le traitement des différentes difficultés rencontrées lors des réservations aux hôtels . Les langages de modélisation engagé pour ce fait est le **XML** et **DTD** qui est consacré à faire une étude préliminaire de l'application à concevoir. Ce présent chapitre offre également une démonstration des diagrammes des différentes utilisations en décrivant les scénarios de chaque acteur, par conséquent, les diagrammes de séquence qui évoquent les interactions entre ces acteurs et les différents objets.

Nous nous sommes répartis les tâches pour effectuer ce projet. En effet, Mohamed Ben Yassine s'est occupé du diagramme E/A notamment pour la description du modèle, du dictionnaire de données et de la transformation vers le schéma relationnel. Massin TAMZAIT s'est occupé de la partie création de table et de l'insertion de données dans SQLITEMAN, mais aussi il s'est occupé de la description de scénario XML/DTD et de sa vérification à l'aide de la commande : **xmllint - - scenario1.xml** et **xmllint - - scenario2.xml** sur linux. Mais nous nous sommes tous les deux investis pour concevoir le diagramme E/A.

1. DESCRIPTION DU MODÈLE E/A

On souhaite réaliser une application pour la gestion des hôtels, chaque hôtel est caractérisé par son id, nom, adresse, tel et son nombre d'étoiles. Chaque hôtel possède différentes catégories de chambres.

Le client effectue une ou plusieurs réservations, il doit indiquer son nom, son email et son téléphone, le client exprime ensuite son besoin : catégorie de chambre, période de séjour.

Le prix de la catégorie d'une chambre dépend du nombre d'étoiles d'hôtel.

Le prix d'une chambre dépend de sa catégorie et nombre d'étoiles d'hôtel.

Une réservation ne concerne qu'une seule chambre.

2. CONCEPTION DE LA BASE DE DONNÉES

2.1 Élaboration du dictionnaire de données

Le dictionnaire de données permet, comme son nom l'indique, de lister l'ensemble des champs nécessaires à la création de la base de données, avec une explication sommaire du rôle de chacun des champs.

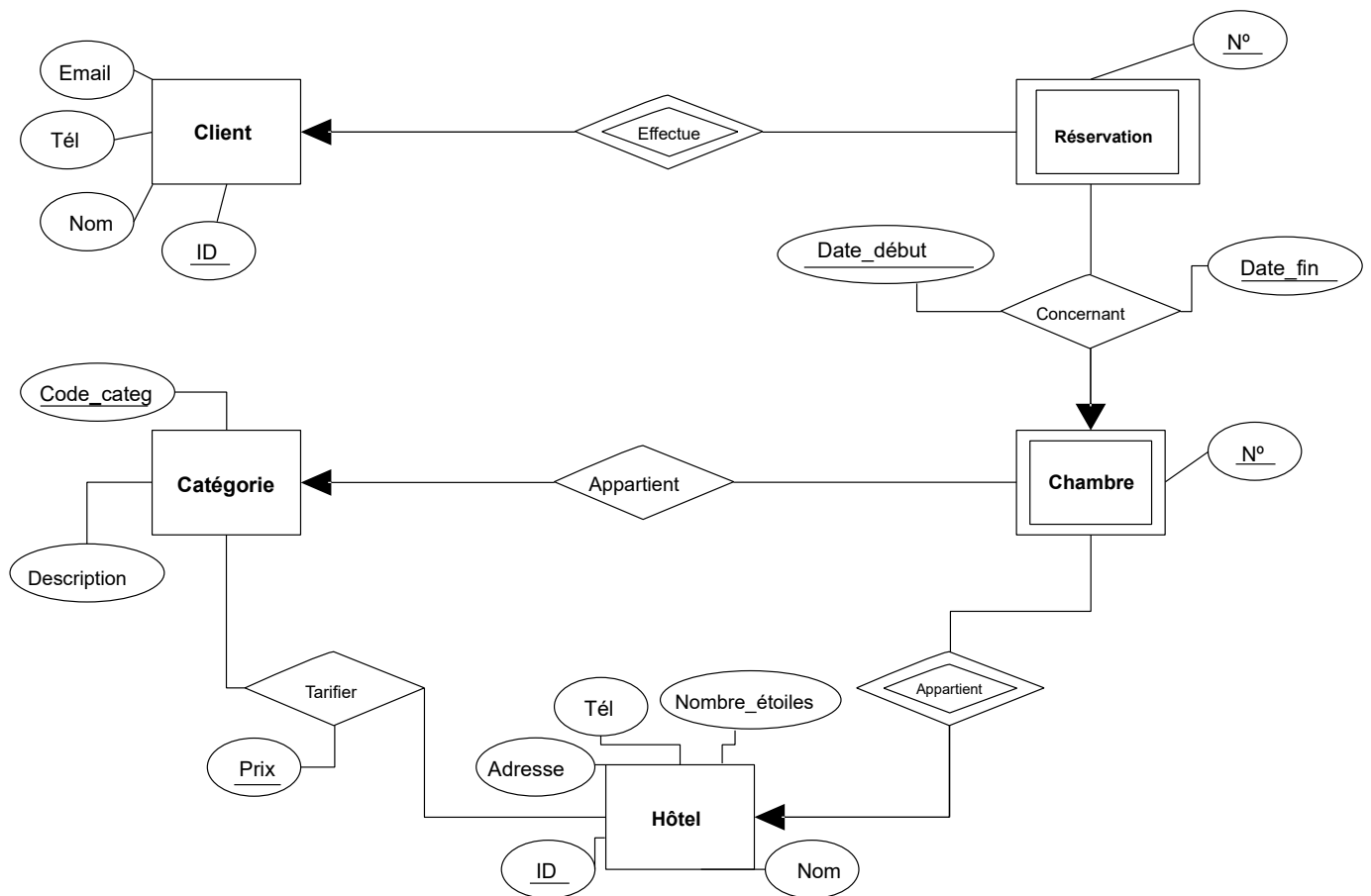
Nous avons décidé de mettre plusieurs contraintes sur différentes colonnes des tables, ainsi, si l'utilisateur tente de stocker des données dans une colonne en violation d'une contrainte définie, une erreur sera levée et indiquera à l'utilisateur d'effectuer une autre action.

Nous avons utilisé une contrainte de vérification au niveau de la table **Hotel** dans la colonne **Nombre_etoiles**, dans ce cas, nous voulons une classification pour un hôtel allant de 1 à 5 étoiles.

Les autres contraintes sont des contraintes de non-nullité, ainsi chaque colonne des tables ne peuvent pas prendre de valeurs NULL.

Champs	Description	Type	Nature
ID_Client	Identifiant du client	Integer	Élémentaire
Nom_Client	Nom du client	Texte	Élémentaire
Tel_Client	Téléphone du client	Texte	Élémentaire
Email_Client	Mail du client	Texte	Élémentaire
Code_Categorie	Identifiant de la catégorie de chambre	Integer	Élémentaire
Description_Categorie	Description de la catégorie de chambre	Texte	Élémentaire
Nom_Hotel	Nom de l'hôtel	Texte	Élémentaire
Adresse_Hotel	Adresse de l'hôtel	Char(80)	Élémentaire
Tel_Hotel	Téléphone de l'hôtel	Texte	Élémentaire
Nombre_etoiles_Hotel	Nombre d'étoiles de l'hôtel (réputation)	Integer CHECK (nombre_etoiles BETWEEN 1 AND 5)	Élémentaire
ID_Hotel	Identifiant de l'hôtel	Integer	Élémentaire
Numero_Chambre	Identifiant de la chambre	Integer	Élémentaire
Numero_Reservation	Identifiant de la réservation	Integer	Élémentaire
Date_Debut_Reservati on	Date de début de réservation	Date	Élémentaire
Date_Fin_Reservation	Date de fin de réservation	Date	Élémentaire
Prix_Tarification	Prix du tarif de la catégorie de chambre	Float	Élémentaire

2.2 Diagramme E/A



Description des flèches :

La mise en relation de tables permet de relier les données d'une table à celles d'une autre table et ainsi d'établir une base de données de type relationnelle.

Les relations entre des tables permettent d'utiliser les données d'une table dans une autre et d'éviter ainsi la saisie redondante d'informations. Elles permettent également de contrôler la saisie des données : si la saisie des données de la table reliée ne trouve aucune correspondance dans la table primaire, la saisie sera refusée.

En ce qui concerne le modèle, si on connaît la réservation, on connaît le client d'où la flèche vers le client, et il en est de même pour la chambre, si on connaît la réservation, on connaît la chambre.

Pour ce qui est de la catégorie de chambre, si on connaît la chambre et son numéro, on connaîtra sa catégorie auquel la chambre est affiliée, et même chose pour l'hôtel, si on connaît la chambre, on connaît l'hôtel d'où les choix de flèches ci-dessus.

Description des entités faible :

Une entité faible ne peut être utilisée indépendamment car elle dépend d'un type d'entité fort appelé entité propriétaire. En outre, la relation qui relie l'entité faible à son identité de propriétaire est appelée la relation d'identification.

Dans le diagramme ER, l'entité faible et la relation correspondante sont représentées par une double ligne.

Dans notre cas, on remarque pour les entités faible, qu'il existe pas des Réservations sans Client cela vaut dire que la table Réservation est dépendante de la table Client et même chose entre la table Chambre et Hôtel, on sait très bien qu'on peut jamais trouver une chambre sans l'hôtel auquel elle appartient.

Les entité Réservation et Chambre ont toujours une contrainte de participation totale en ce qui concerne leur relation d'identification car elles ne peuvent être identifiées indépendamment de leur identité de propriétaire.

Les entités faible « Réservation » et « Chambre » peuvent avoir une clé partielle, qui est une liste d'attributs qui identifient les entités faibles liées à la même entité propriétaire.

Dans le diagramme ER, l'entité faible et la relation correspondante sont représentées par une double ligne et la clé partielle par une ligne pointillée.

2.3 Transformation vers le schéma relationnel

→ ce symbole signifie que la colonne fait référence à une autre table (clé étrangère)

Client (ID-client, Nom, Tel, Email)

Hôtel (ID-hôtel, Nom, Adresse, Tel, Nombre-etoiles)

Catégorie (Code-categ, Description)

Chambre (Numero-chambre-reserv, ID-hôtel#, Code-categ#)

Réservation (Numero-reserv, date-debut, date-fin, ID-client#, Numero-chambre-reserv#)

Tarification (Prix, ID-hôtel#, Code-categ#)

Chambre [ID-hôtel] \subseteq Hôtel [ID]

Chambre [code-categ] \subseteq Catégorie [code-categ]

Réservation [ID-client] \subseteq Client [ID]

Réservation [numero-chambre-reserv] \subseteq Chambre [numero-chambre-reserv]

Tarification [Code-categ] \subseteq Catégorie [Code-categ]

Tarification [ID-hôtel] \subseteq Hôtel [ID]

3. ALIMENTATION DE LA BASE DE DONNÉES

La commande CREATE TABLE va permettre de créer nos tables du modèle E/A en langage SQL. De ce fait, nous pourrons, par la suite, alimenter notre base de données pour avoir un exemple concret. (**Voir fichier schema.sql et insert_data.sql + BD database.db**)

Table Client :

```
CREATE TABLE Client (  
id_client integer not null,  
nom text not null,  
tel text not null,  
email text not null,  
primary key (id_client));
```

Table Catégorie :

```
CREATE TABLE Categorie (  
code_categ integer not null,  
description text not null,  
primary key(code_categ));
```

Table Hotel :

```
CREATE TABLE hotel (  
id_hotel integer not null,  
nom text not null,  
adresse char(80) not null,  
tel text not null,  
nombre_etoiles integer CHECK  
(nombre_etoiles BETWEEN 1 AND 5) not null,  
primary key(id_hotel));
```

Table Tarification :

```
CREATE TABLE Tarification (  
prix float not null,  
code_categ integer not null,  
id_hotel integer not null,  
primary key(prix,id_hotel,code_categ),  
foreign key (code_categ) references  
Categorie(code_categ),  
foreign key (id_hotel) references  
hotel(id_hotel));
```

Table Chambre :

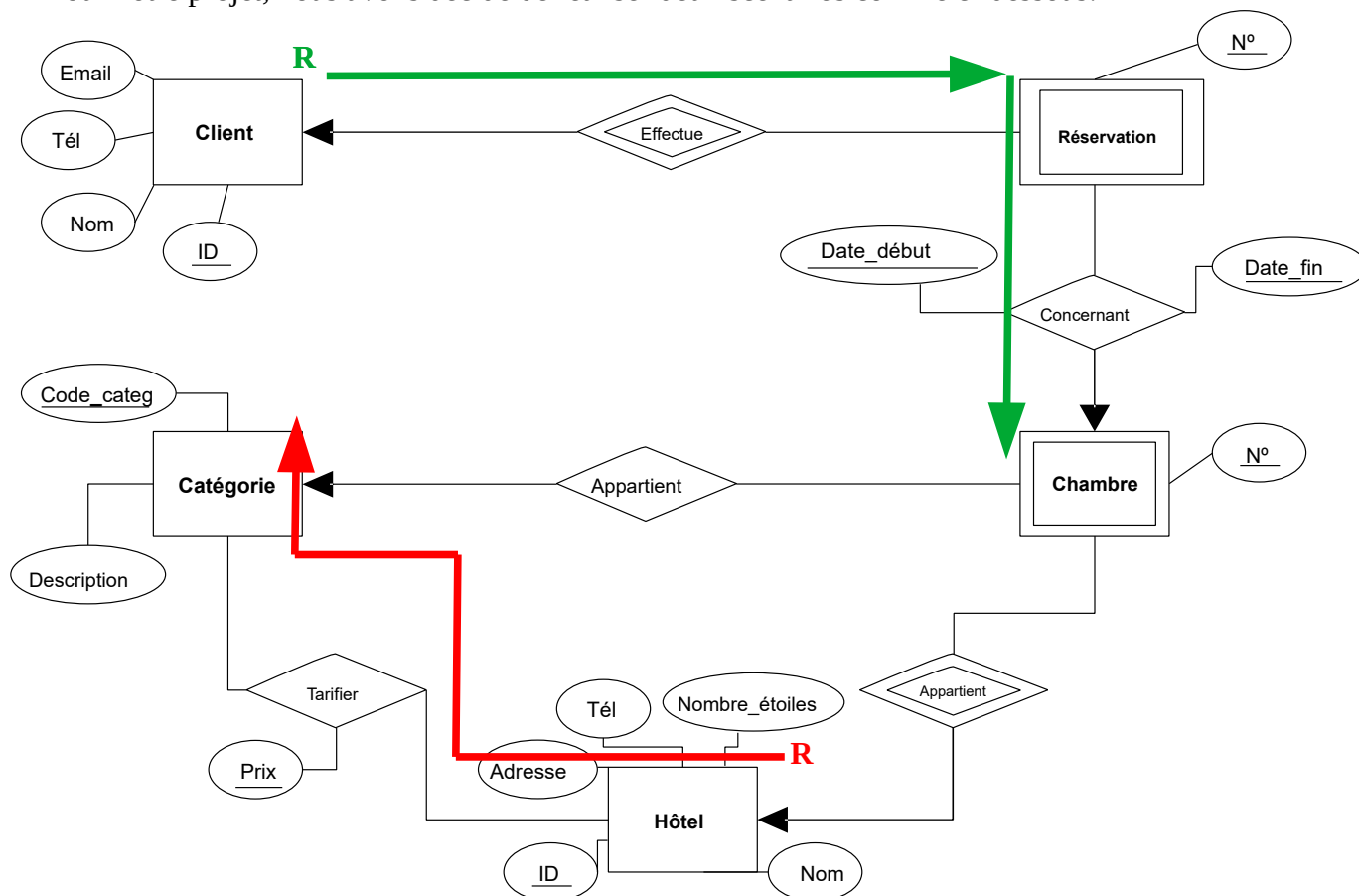
```
CREATE TABLE Chambre (  
numero_chambre_reserv integer not null,  
id_hotel integer not null,  
code_categ integer not null,  
primary key(numero_chambre_reserv, id_hotel),  
foreign key(id_hotel) references hotel(id_hotel),  
foreign key(code_categ) references Categorie(code_categ));
```

Table Réservation :

```
CREATE TABLE Reservation ( numero_reserv integer not null, id_client integer not null,  
date_debut date not null, date_fin date not null, numero_chambre_reserv integer not null, primary  
key(numero_reserv, id_client, date_debut, date_fin), foreign key(numero_chambre_reserv)  
references Chambre(numero_chambre), foreign key(id_client) references Client(id_client));
```

4. DESCRIPTION DES SCENARIOS XML/DTD

Pour notre projet, nous avons décidé de réaliser deux scénarios comme ci-dessous:



1^{er} scénario : Nous souhaitons avoir la liste des tarifs pour une catégorie de chambre et un hôtel spécifique, ainsi, à partir de l'hôtel (nombre d'étoiles) et sa catégorie de chambre, on obtient le prix associé. **(flèche rouge)**

2^e scénario : Nous souhaitons avoir le détail des réservations du client (période de séjour, catégorie chambre). Ainsi, à partir de l'ID du client, du numéro de chambre, de l'hôtel et catégorie de chambre, on aura la période de séjour du client. **(flèche verte)**.

(Voir fichier scenario1.xml, scenario1.dtd et voir fichier scenario2.xml, scenario2.dtd)