

UNIVERSITE AFRICAINE DE MANAGEMENT ET TECHNOLOGIE

UATM – Département Génie Electrique



Domaine : Système Informatique et Logiciel (SIL)

Niveau : L1 – Génie Electrique

PROJET DE MERISE

Thème du projet:

Conception d'une Base de Données de
Gestion de Réservations Hôtelières

Réalisé par :

PEDROS MARCOS Samuel

Encadré par :

Mr MONTCHO Pierre-Claver

Année universitaire : 2024 – 2025

Projet Merise - UATM

INTRODUCTION

Dans le cadre de la gestion hôtelière, l'optimisation des réservations et des services offerts aux clients est essentielle pour assurer une expérience satisfaisante et une administration efficace. Ce projet vise à concevoir une base de données relationnelle pour la gestion des réservations des hôtels situés dans les arrondissements du Bénin, en s'appuyant sur la méthode MERISE. L'objectif est de modéliser un système capable de gérer les chambres, les réservations, les clients, les paiements et les services supplémentaires, tout en respectant des règles de gestion strictes pour garantir la cohérence et la fiabilité des données. À travers ce travail, nous allons élaborer un dictionnaire de données, un modèle conceptuel, un schéma logique relationnel, et implémenter des requêtes SQL fonctionnelles pour répondre aux besoins opérationnels de VLAVONOU Měgnonkpon, gestionnaire d'hôtels.

Projet Merise - UATM

1. Règles de Gestion

- Une chambre ne peut être réservée que si elle est disponible pour les dates demandées.
- Une chambre ne peut pas être supprimée si elle est associée à des réservations en cours ou passées.
- Une réservation doit être associée à un client et à au moins une chambre.
- Une réservation ne peut pas chevaucher une autre réservation pour la même chambre.
- Le nombre de clients dans une réservation ne peut pas dépasser la capacité de la chambre.
- Un client ne peut pas être supprimé s'il a des réservations en cours ou passées.
- Un client peut avoir plusieurs réservations, mais chaque réservation est unique.
- Un paiement doit être associé à une réservation.
- Le montant total d'une réservation est calculé en fonction du tarif de la chambre et des services supplémentaires.
- Les services supplémentaires peuvent être ajoutés à une réservation et possèdent un tarif fixe.
- Chaque service a un tarif fixe qui s'ajoute au montant total de la réservation.

2. Dictionnaire de Données

CODE	SIGNIFICATION
numchbr	Numéro de la chambre
typchbr	Type de la chambre
capacite	Capacité de la chambre
tarifchbr	Prix journalier de la chambre
datpaie	Date de paiement
datdebu	Date du début de séjour
datfin	Date de fin de séjour
numcli	Numéro de client
nomcli	Nom du client
precli	Prénoms du client
telcli	Téléphone du client
mail	Email du client
numres	Numéro de réservation
nbpersonnes	Nombre de personnes
numser	Numéro de service
libser	Libellé du service demandé
tarifser	Tarif du service
quantite	Nombre de service
monttot	Montant total
codhot	Code de l'hôtel

Projet Merise - UATM

nomhot	Nom de l'hôtel
adrhot	Adresse de l'hôtel
arr	Nom de l'arrondissement
dep	Nom du département
com	Nom de la commune

3. Couverture Minimale

numres → numcli, monttot

numchbr → codhot

codhot → arr

arr → com

com → dep

4. Modèle Conceptuel de Données (MCD)

Créons d'abord le SLR:

- CLIENT (**numcli**, nomcli, precli, mail, telcli)
- CHAMBRE (**numchbr**, typchbr, capacite, tarifchbr, #codhot)
- RESERVATION (**numres**, #numcli, datdebu, datfin)
- HOTEL (**codhot**, nomhot, adrhot, #arr)
- ARRONDISSEMENT (**arr**, #com)
- COMMUNE (**com**, #dep)
- DEPARTEMENT (**dep**)
- PAIEMENT (**datpaie**, #numres, monttot)
- SERVICES (**numser**, libser, tarifser)
- CONTENIR (**#numres**, **#numchbr**, nbpersonnes)
- FOURNIR (**#numres**, **#numser**, quantite)

Créons à present le MCD.

Éclaircissement

Élément	Type
CLIENT	Entité
CHAMBRE	Entité
RESERVATION	Entité
PAIEMENT	Entité
HOTEL	Entité
SERVICE	Entité
ARRONDISSEMENT	Entité
COMMUNE	Entité
DEPARTEMENT	Entité

Projet Merise - UATM

CIF	Association
CIF	Association
CONTENIR	Association
FOURNIR	Association
CIF	Association
CIF	Association
CIF	Association



5. Schéma Relationnel (MLD)

[illegible]

Projet Merise - UATM

5	<u>datpaie</u>		*									
6	monttot		1									
7	<u>numres</u>			*								
8	datdebu			1								
9	datfin			1								
10	<u>numcli</u>				*							
11	nomcli				1							
12	precli				1							
13	telcli				1							
14	mail				1							
15	<u>numser</u>					*						
16	libser					1						
17	tarifser					1						
18	nbpersonnes										1	
19	quantite											1
20	<u>codhot</u>						*					
21	nomhot						1					
22	adrhot						1					
23	<u>arr</u>							*				
24	<u>dep</u>								*			
25	<u>com</u>									*		
26	<u>7 + 1</u>										*	
27	<u>7 + 15</u>											*

6. SQL – Création des Tables

create table CLIENT (numcli varchar (10) not null, nomcli varchar (50) not null, precli varchar (50) not null, mail varchar (100) not null, telcli varchar (20) not null, primary key(numcli) references CLIENT);

create table RESERVATION (numres varchar (10) not null, datdebu date not null, datfin date not null, monttot decimal (10,2) not null, numcli varchar (10) not null, primary key(numres), foreign key (numcli) references CLIENT), check (datfin > datdebu);

create table HOTEL (codhot varchar (10) not null, nomhot varchar (10) not null, adrhot varchar (10) not null, arr varchar (10) not null, primary key (codhot), foreign key (arr) references ARRONDISSEMENT);

create table PAIEMENT (numres varchar (10) not null, datpaie date not null, monttot decimal (10,2) not null, primary key(datpaie), foreign key (numres) references RESERVATION);

Projet Merise - UATM

create table SERVICES (numser varchar (10) not null, libser varchar (50) not null, tarifser decimal (10,2) not null, primary key (numser) references SERVICES);

create table DEPARTEMENT (dep varchar (50) not null, primary key(dep) references DEPARTEMENT);

create table COMMUNE (com varchar (50) not null, dep varchar (50) not null, primary key(com), foreign key (dep) references DEPARTEMENT);

create table ARRONDISSEMENT (arr varchar (50) not null, com varchar (50) not null, primary key(arr), foreign key (com) references COMMUNE);

create table CONTENIR (numres varchar (10) not null, numchbr varchar (10) not null, nbpersonnes INT not null, primary key (numres, numchbr), foreign key (numres) references RESERVATION, foreign key (numchbr) references CHAMBRE);

create table FOURNIR (numres varchar (10) not null, numser varchar (10) not null, quantite int (10) not null, primary key (numres, numser), foreign key (numres) references RESERVATION (numres), foreign key (numser) references SERVICES);

7. Requêtes SQL Fonctionnelles

1. Afficher les chambres disponibles entre deux dates données.
2. Lister les réservations effectuées par un client donné.
3. Afficher les services associés à une réservation donnée.
4. Calculer le montant total d'une réservation (chambre + services).
5. Lister les clients ayant effectué des réservations ce mois-ci.
6. Afficher les paiements effectués sur une période donnée.
7. Lister toutes les chambres d'un hôtel donné.
8. Afficher les revenus générés par chaque hôtel.
9. Afficher les réservations en cours aujourd'hui.
10. Afficher les hôtels situés dans une commune donnée.

CONCLUSION

Ce projet a permis de concevoir une base de données robuste et fonctionnelle pour la gestion des réservations hôtelières, en adéquation avec les exigences du contexte et les règles de gestion définies. Grâce à la modélisation MERISE, nous avons pu structurer les entités et leurs relations, traduites ensuite en un schéma logique relationnel et en requêtes SQL opérationnelles. Cette solution offre une gestion optimisée des chambres, des clients, des réservations et des services supplémentaires, tout en garantissant l'intégrité des données. Elle constitue un outil précieux pour améliorer l'expérience client et simplifier les processus internes de gestion hôtelière. Enfin, ce travail collaboratif a renforcé nos compétences en conception de bases de données et en utilisation des SGBDR, tout en répondant aux attentes du cahier.