**LIVRABLE**

**1-**Introduction: Le modèle logique de données est un schéma représentant la structure des données dans une base de données. Il décrit les entités, les attributs et les relations entre les entités. Le modèle logique de données est une étape importante dans le développement d'une base de données car il fournit une représentation claire et précise des données, facilitant ainsi la conception et la mise en œuvre d'une base de données efficace.

Description de l'entreprise: Nht Hotel est une entreprise de gestion hôtelière qui possède plusieurs hôtels à travers le monde. L'entreprise est spécialisée dans la gestion d'hôtels de luxe et offre des services haut de gamme à ses clients. Nht Hotel a une base de données centralisée pour gérer les réservations, les clients, les employés, les chambres et les services proposés dans les hôtels.

Modèle logique de données: Le modèle logique de données de Nht Hotel comprend les entités suivantes :

1. Client : Cette entité stocke les informations des clients tels que leur nom, prénom, adresse, numéro de téléphone, adresse email, etc.
2. Réservation : Cette entité stocke les informations relatives à la réservation de la chambre, telles que le numéro de réservation, la date d'arrivée, la date de départ, le type de chambre, le nombre de personnes, etc.
3. Chambre : Cette entité stocke les informations relatives aux chambres disponibles dans les différents hôtels, telles que le numéro de la chambre, le type de chambre, le nombre de lits, la vue, les équipements, etc.
4. Employé : Cette entité stocke les informations relatives aux employés de Nht Hotel, telles que leur nom, prénom, poste, salaire, date d'embauche, etc.
5. Service : Cette entité stocke les informations relatives aux services proposés par Nht Hotel, tels que le spa, le restaurant, le bar, la salle de sport, etc.
6. Facture : Cette entité stocke les informations relatives aux factures émises pour les clients, telles que le numéro de facture, la date de la facture, le montant total, les taxes, etc.

Relations entre les entités :

1. Un client peut effectuer plusieurs réservations, mais une réservation ne peut être effectuée que par un seul client. Il y a donc une relation un-à-plusieurs entre les entités Client et Réservation.
2. Une réservation ne peut être associée qu'à une seule chambre, mais une chambre peut être associée à plusieurs réservations. Il y a donc une relation un-à-plusieurs entre les entités Réservation et Chambre.
3. Un employé peut travailler dans plusieurs hôtels, mais chaque hôtel peut avoir plusieurs employés. Il y a donc une relation plusieurs-à-plusieurs entre les entités Employé et Hôtel.
4. Une facture est émise pour une réservation et contient les services utilisés par le client. Il y a donc une relation un-à-plusieurs entre les entités Réservation et Facture, et une relation plusieurs-à-plusieurs entre les entités Service et Facture.

Conclusion: Le modèle logique de données de Nht Hotel fournit une représentation claire et précise des données stockées dans la base de données de l'entreprise. Il décrit

2-Top of Form

Livrable - Transformation du schéma conceptuel en schéma logique de l'entreprise NHT Hotel

Cher client,

Nous sommes heureux de vous présenter le livrable qui décrit la transformation du schéma conceptuel en schéma logique de votre entreprise NHT Hotel.

Contexte : Dans le cadre de notre mission, nous avons travaillé sur la conception de la base de données de votre entreprise NHT Hotel. Nous avons commencé par la modélisation du schéma conceptuel, qui est une représentation visuelle des entités, des relations et des contraintes qui régissent les données de votre entreprise.

Transformation du schéma conceptuel en schéma logique : La transformation du schéma conceptuel en schéma logique consiste à traduire la modélisation conceptuelle en une structure de base de données physique, qui peut être implémentée dans un système de gestion de base de données.

Nous avons utilisé les règles de la normalisation pour transformer le schéma conceptuel en schéma logique. La normalisation est une technique qui permet d'éliminer les redondances, les anomalies et les incohérences dans une base de données.

Le schéma logique est constitué de tables, qui représentent les entités du schéma conceptuel, ainsi que de clés primaires, de clés étrangères et de contraintes d'intégrité référentielle. Les relations entre les tables sont représentées par les clés primaires et les clés étrangères.

Nous avons également inclus dans le schéma logique des index et des vues, qui permettent d'optimiser les performances de la base de données.

Conclusion : La transformation du schéma conceptuel en schéma logique est une étape importante dans la conception d'une base de données. Le schéma logique est la base pour la création de la base de données physique et sa mise en place dans un système de gestion de base de données. Nous avons utilisé les règles de la normalisation pour garantir l'intégrité et la cohérence des données de votre entreprise.

Nous espérons que ce livrable répond à vos attentes et que vous êtes satisfait de notre travail. N'hésitez pas à nous contacter si vous avez des questions ou des remarques.

Cordialement,

L'équipe de conception de base de données de votre entreprise.

**3-** Voici un livrable expliquant les étapes de la normalisation de la base de données de l'entreprise NHT Hotel, avec des exemples pour chaque étape.

Introduction : La normalisation est un processus important dans la conception de bases de données relationnelles. Elle permet de réduire la redondance des données, d'améliorer l'efficacité de la base de données et de réduire les erreurs de données. Dans ce livrable, nous allons expliquer les étapes de la normalisation de la base de données de l'entreprise NHT Hotel, en utilisant des exemples pour illustrer chaque étape.

Étape 1 : La première forme normale (1NF) La première forme normale (1NF) exige que chaque colonne d'une table ne contienne qu'une seule valeur atomique (indivisible), c'est-à-dire qu'elle ne contienne pas plusieurs valeurs en une seule cellule. Par exemple, si nous avons une table qui stocke les informations des clients, chaque colonne doit contenir une seule valeur atomique, comme le nom, l'adresse, le numéro de téléphone, etc.

Exemple : Voici un exemple d'une table clients qui ne respecte pas la première forme normale (1NF) :

sqlCopy code

Table clients : --------------------------------------- | ID\_Client | Nom | Téléphones | --------------------------------------- | 1 | Dupont | 555-1234 | | 2 | Martin | 555-5678 | | 3 | Tremblay | 555-9876 | ---------------------------------------

Dans cette table, la colonne "Téléphones" contient plusieurs numéros de téléphone en une seule cellule. Pour normaliser cette table, nous pouvons créer une nouvelle table "Téléphones" qui stockera les numéros de téléphone pour chaque client.

Étape 2 : La deuxième forme normale (2NF) La deuxième forme normale (2NF) exige que chaque colonne d'une table soit dépendante de la clé primaire de la table. Cela signifie que chaque colonne doit fournir des informations uniquement sur la clé primaire de la table. Par exemple, si nous avons une table qui stocke les informations des réservations d'hôtel, chaque colonne doit être dépendante de la clé primaire, qui pourrait être le numéro de réservation.

Exemple : Voici un exemple d'une table réservations qui ne respecte pas la deuxième forme normale (2NF) :

sqlCopy code

Table réservations : ----------------------------------- | Num\_Réservation | Date | Chambre | ----------------------------------- | 101 | 2022-01-01 | 102 | | 102 | 2022-01-02 | 104 | | 103 | 2022-01-03 | 102 | -----------------------------------

Dans cette table, la colonne "Chambre" n'est pas dépendante de la clé primaire "Num\_Réservation". Pour normaliser cette table, nous pouvons créer une nouvelle table "Chambres" qui stockera les informations sur chaque chambre, y compris le type de chambre, le tarif, etc.

Étape 3 : La troisième forme normale (3NF) La troisième forme normale (3NF) exige que chaque colonne d'une table ne dépende pas d'une autre colonne non-clé. Cela signifie que les informations doivent être divisées en plusieurs tables