

Agence de Location de Bateaux

Spécification, Analyse et Conception

Membres du Groupe

- **BELKZIZ**
- **BADDOU**
- **DRHIMER**

Description du Sujet

L'application « Agence de Location de Bateaux » est une plateforme dédiée à l'optimisation des opérations d'une agence de location de bateaux, englobant la gestion intégrée des bateaux, clients, réservations, paiements, et services additionnels. Conçue pour répondre aux besoins spécifiques de l'agence, cette solution centralise l'ensemble des processus opérationnels en une interface unique et ergonomique. Son objectif est de maximiser l'efficacité opérationnelle grâce à l'automatisation des flux de travail, la simplification des interactions utilisateurs, et une gestion rigoureuse des données essentielles, offrant ainsi un suivi précis et une flexibilité accrue dans la gestion des activités quotidiennes.

Structures et Services

- **Bateaux** : Contient toutes les informations essentielles sur chaque bateau disponible, telles que le nom, le type, la capacité d'accueil, le statut (disponible, loué, maintenance), et le tarif journalier.
- **Clients** : Stocke les détails des clients, notamment le nom, le prénom, l'adresse email, et le numéro de téléphone.
- **Réservations** : Enregistre les réservations en liant un client à un bateau pour une période définie, incluant toutes les informations pertinentes sur la location.
- **Paiements** : Consigne les transactions relatives aux réservations, incluant les détails de chaque paiement associé à une réservation.
- **Services** : Liste les options supplémentaires que les clients peuvent ajouter à leur réservation, comme un équipage ou des équipements spéciaux.

Fonctionnalités de Gestion

- **Gestion des Bateaux** : Permet d'ajouter, de modifier, de supprimer et de consulter les informations de chaque bateau.

- **Gestion des Clients** : Fournit les options d'ajout, de modification, de suppression et de consultation pour les profils clients.
- **Gestion des Réservations** : Gère la création, l'ajustement, l'annulation et la consultation des réservations.
- **Gestion des Paiements** : Offre des fonctionnalités pour enregistrer, modifier, supprimer et consulter les paiements liés aux réservations.
- **Gestion des Services Supplémentaires** : Permet d'ajouter, de supprimer et de consulter les services additionnels que les clients peuvent sélectionner.

Cette documentation décrit les requêtes SQL pour l'application « Agence de Location de Bateaux », détaillant les opérations d'insertion, de suppression, de mise à jour, et de consultation sur les différentes tables de données.

1. Requêtes d'Insertion

Table Client

- **Insérer un client** : Enregistre un nouveau client avec les informations suivantes : nom, prénom, email (unique), et numéro de téléphone.
Exemple : Insérer un client nommé "Jean Dupont" avec l'email jean.dupont@example.com et le numéro de téléphone 0601020304.

Table Bateau

- **Ajouter un bateau** : Insère un bateau avec un nom unique, un type (par ex., Voilier, Yacht), une capacité maximale, un statut (ex. : Disponible), et un tarif journalier.
Exemple : Ajouter un bateau nommé "Le Voyageur", de type "Voilier", capacité de 6 personnes, statut "Disponible", à 200€ par jour.

Table Réservation

- **Créer une réservation** : Associe un client et un bateau disponible pour une période déterminée.
Exemple : Créer une réservation pour le client ID 1 avec le bateau ID 1, du 01/11/2023 au 05/11/2023.

Table Paiement

- **Enregistrer un paiement** : Ajoute un paiement pour une réservation spécifique, incluant le montant, la date, et le mode de paiement (Carte, Virement, Espèces).
Exemple : Enregistrer un paiement de 800€ pour la réservation ID 1, effectué par carte le 30/10/2023.

Table Service

- **Ajouter un service supplémentaire** : Associe un service additionnel (ex. : Personnel de bord) à une réservation, avec un coût spécifique.
Exemple : Ajouter le service "Personnel de bord" pour 400€ à la réservation ID 1.
-

2. Requêtes de Suppression

Table Client

- **Supprimer un client** : Retire un client si aucune réservation en cours ou future n'est associée.
Exemple : Supprimer le client ID 5 si aucune réservation n'est en cours.

Table Bateau

- **Supprimer un bateau** : Supprime un bateau en l'absence de réservations futures.
Exemple : Supprimer le bateau ID 10 s'il est disponible pour toutes les dates futures.

Table Réservation

- **Annuler une réservation** : Annule une réservation avant la date de début, en actualisant le statut du bateau concerné.
Exemple : Annuler la réservation ID 3 et rendre le bateau disponible.

Table Paiement

- **Supprimer un paiement** : Efface un paiement en cas d'erreur, avec enregistrement de la raison pour audit.
Exemple : Supprimer le paiement ID 5 et enregistrer la raison pour le suivi.

Table Service

- **Retirer un service** : Supprime un service additionnel d'une réservation si le client annule avant le début.
Exemple : Retirer le service ID 2 de la réservation ID 1.
-

3. Requêtes de Mise à Jour

Requêtes sur une Table

1. **Mise à jour des coordonnées d'un client** : Met à jour le numéro de téléphone et l'email du client ID 1. L'email doit rester unique.

Exemple : Mettre à jour le client ID 1 avec 0612345678 et nouveau.email@example.com.

2. **Changement du statut d'un bateau** : Met à jour le statut d'un bateau à 'En Maintenance' ou autre.

Exemple : Mettre le statut du bateau ID 2 à 'En Maintenance'.

Requêtes sur Deux Tables

1. **Modification de dates et mise à jour de statut de réservation** : Change les dates de réservation et actualise le statut du bateau.

Exemple : Modifier la réservation ID 4 (du 10/12/2023 au 15/12/2023) et mettre à jour le statut du bateau.

2. **Réajustement de paiement après modification de réservation** : Ajuste le montant d'un paiement selon les nouvelles dates d'une réservation.

Exemple : Recalculer le paiement pour la réservation ID 4 en fonction des nouvelles dates.

Requêtes sur Plus de Deux Tables

1. **Transfert d'une réservation à un autre client** : Attribue une réservation à un autre client, avec ajustement des services et paiements.

Exemple : Transférer la réservation ID 5 au client ID 6 et ajuster les services et paiements.

2. **Mise à jour d'informations client avec propagation** : Change les informations d'un client et applique les modifications aux réservations futures.

Exemple : Mettre à jour les coordonnées du client ID 7 et propager aux réservations futures.

4. Requêtes de Consultation

Consultations Simples (1 Table)

1. **Lister les bateaux disponibles par nom** :

Exemple : Sélectionner tous les bateaux disponibles, triés par nom.

2. **Détails d'un client spécifique** :

Exemple : Afficher les informations pour le client ID 3.

3. **Nombre de bateaux par type** :

Exemple : Compter les bateaux pour chaque type (GROUP BY Type).

4. **Liste des services supplémentaires** :

Exemple : Afficher les services disponibles, sans doublons.

5. **Historique des paiements triés par date :**

Exemple : Lister tous les paiements, triés par date décroissante.

Consultations avec Jointures Internes (2 Tables)

1. **Réservations avec noms des clients :**

Exemple : Joindre Reservation et Client pour afficher les noms des clients et leurs réservations.

2. **Bateaux loués et dates de location :**

Exemple : Joindre Reservation et Bateau pour afficher les bateaux et leurs dates de location.

3. **Clients sans réservation (jointure externe) :**

Exemple : Joindre Client et Reservation pour lister tous les clients, y compris ceux sans réservation.

4. **Nombre de réservations par client :**

Exemple : Compter les réservations par client (GROUP BY ID_Client).

5. **Paiements supérieurs à 1000€, triés :**

Exemple : Sélectionner les paiements >1000€, triés par montant décroissant.

Consultations avec Jointures sur Plus de Deux Tables

1. **Lister clients, réservations et bateaux :**

Exemple : Joindre Client, Reservation, et Bateau pour afficher les clients, leurs réservations, et les bateaux associés.

2. **Paiements avec détails du client et bateau :**

Exemple : Joindre Paiement, Reservation, Client, et Bateau pour afficher des paiements avec informations complètes.

3. **Services additionnels par réservation et client :**

Exemple : Joindre Service, Reservation, et Client pour lister les services par réservation et client.

4. **Réservations par type de bateau :**

Exemple : Compter les réservations pour chaque type de bateau (GROUP BY Type).

5. **Clients et paiements, incluant ceux sans paiements :**

Exemple : Joindre Client et Paiement, avec jointure externe pour inclure les clients sans paiement, triés par nom.

Dictionnaire de Données MERISE

Entité Client

- **ID_Client**

- *Description* : Identifiant unique du client
- *Format* : NUMBER(10)
- *Type* : Clé primaire
- *Identifiant* : Oui
- *Contraintes* : NOT NULL

- **Nom**

- *Description* : Nom du client
- *Format* : VARCHAR2(50)
- *Type* : Attribut simple
- *Identifiant* : Non
- *Contraintes* : NOT NULL

- **Prénom**

- *Description* : Prénom du client
- *Format* : VARCHAR2(50)
- *Type* : Attribut simple
- *Identifiant* : Non
- *Contraintes* : NOT NULL

- **Email**

- *Description* : Adresse email du client
- *Format* : VARCHAR2(100)
- *Type* : Attribut unique
- *Identifiant* : Non
- *Contraintes* : NOT NULL, UNIQUE

- **Téléphone**

- *Description* : Numéro de téléphone du client

- *Format* : VARCHAR2(15)
- *Type* : Attribut simple
- *Identifiant* : Non
- *Contraintes* : NOT NULL

Entité Bateau

- **ID_Bateau**

- *Description* : Identifiant unique du bateau
- *Format* : NUMBER(10)
- *Type* : Clé primaire
- *Identifiant* : Oui
- *Contraintes* : NOT NULL

- **Nom**

- *Description* : Nom du bateau
- *Format* : VARCHAR2(100)
- *Type* : Attribut unique
- *Identifiant* : Non
- *Contraintes* : NOT NULL, UNIQUE

- **Type**

- *Description* : Type de bateau
- *Format* : VARCHAR2(50)
- *Type* : Attribut simple
- *Identifiant* : Non
- *Contraintes* : NOT NULL

- **Capacité**

- *Description* : Capacité maximale de passagers
- *Format* : NUMBER(3)
- *Type* : Attribut simple
- *Identifiant* : Non

- *Contraintes* : NOT NULL, CHECK (Capacité > 0)
- **Statut**
 - *Description* : Statut du bateau
 - *Format* : VARCHAR2(20)
 - *Type* : Attribut simple
 - *Identifiant* : Non
 - *Contraintes* : NOT NULL, CHECK (Statut IN ('Disponible', 'Loué', 'En Maintenance'))
- **Prix_Jour**
 - *Description* : Prix de location par jour
 - *Format* : NUMBER(10,2)
 - *Type* : Attribut simple
 - *Identifiant* : Non
 - *Contraintes* : NOT NULL, CHECK (Prix_Jour > 0)

Entité Reservation

- **ID_Reservation**
 - *Description* : Identifiant unique de la réservation
 - *Format* : NUMBER(10)
 - *Type* : Clé primaire
 - *Identifiant* : Oui
 - *Contraintes* : NOT NULL
- **ID_Client**
 - *Description* : Référence au client
 - *Format* : NUMBER(10)
 - *Type* : Clé étrangère vers Client
 - *Identifiant* : Non
 - *Contraintes* : NOT NULL
- **ID_Bateau**

- *Description* : Référence au bateau
- *Format* : NUMBER(10)
- *Type* : Clé étrangère vers Bateau
- *Identifiant* : Non
- *Contraintes* : NOT NULL
- **Date_Debut**
 - *Description* : Date de début de la réservation
 - *Format* : DATE
 - *Type* : Attribut simple
 - *Identifiant* : Non
 - *Contraintes* : NOT NULL
- **Date_Fin**
 - *Description* : Date de fin de la réservation
 - *Format* : DATE
 - *Type* : Attribut simple
 - *Identifiant* : Non
 - *Contraintes* : NOT NULL, CHECK (Date_Fin > Date_Debut)
- **Commentaire**
 - *Description* : Commentaires sur la réservation
 - *Format* : VARCHAR2(255)
 - *Type* : Attribut simple
 - *Identifiant* : Non
 - *Contraintes* : Aucune

Entité Paiement

- **ID_Paiement**
 - *Description* : Identifiant unique du paiement
 - *Format* : NUMBER(10)
 - *Type* : Clé primaire

- *Identifiant* : Oui
- *Contraintes* : NOT NULL
- **ID_Reservation**
 - *Description* : Référence à la réservation
 - *Format* : NUMBER(10)
 - *Type* : Clé étrangère vers Reservation
 - *Identifiant* : Non
 - *Contraintes* : NOT NULL, UNIQUE
- **Montant**
 - *Description* : Montant du paiement
 - *Format* : NUMBER(10,2)
 - *Type* : Attribut simple
 - *Identifiant* : Non
 - *Contraintes* : NOT NULL, CHECK (Montant > 0)
- **Date_Paiement**
 - *Description* : Date du paiement
 - *Format* : DATE
 - *Type* : Attribut simple
 - *Identifiant* : Non
 - *Contraintes* : NOT NULL
- **Mode_Paiement**
 - *Description* : Mode de paiement
 - *Format* : VARCHAR2(20)
 - *Type* : Attribut simple
 - *Identifiant* : Non
 - *Contraintes* : NOT NULL, CHECK (Mode_Paiement IN ('Carte', 'Virement', 'Espèces'))

Entité Service

- **ID_Service**

- *Description* : Identifiant unique du service
- *Format* : NUMBER(10)
- *Type* : Clé primaire
- *Identifiant* : Oui
- *Contraintes* : NOT NULL

- **ID_Reservation**

- *Description* : Référence à la réservation
- *Format* : NUMBER(10)
- *Type* : Clé étrangère vers Reservation
- *Identifiant* : Non
- *Contraintes* : NOT NULL

- **Type_Service**

- *Description* : Type de service
- *Format* : VARCHAR2(50)
- *Type* : Attribut simple
- *Identifiant* : Non
- *Contraintes* : NOT NULL

- **Coût**

- *Description* : Coût du service
- *Format* : NUMBER(10,2)
- *Type* : Attribut simple
- *Identifiant* : Non
- *Contraintes* : NOT NULL, CHECK (Coût >= 0)

Description des Associations

? Client (1,N) --- effectue --- (0,N) Réserve

- **Description** : Un client peut effectuer plusieurs réservations, et une réservation est toujours associée à un client unique.
- **Cardinalité** : (1, N) signifie qu'un client peut avoir de nombreuses réservations, tandis que (0, N) pour la réservation indique qu'elle peut exister ou non pour un client.
- **Exemple** : Jean Dupont, client de l'agence, peut effectuer des réservations pour différents bateaux sur plusieurs périodes, toutes ses réservations étant liées à son identifiant client.

? Bateau (1,N) --- est réservé dans --- (0,N) Réserve

- **Description** : Un bateau peut être réservé plusieurs fois au fil du temps par différents clients, mais chaque réservation ne concerne qu'un seul bateau à la fois.
- **Cardinalité** : (1, N) pour le bateau signifie qu'il peut être réservé un nombre illimité de fois, et (0, N) pour les réservations indique que chaque réservation est pour un bateau unique.
- **Exemple** : Le bateau "Le Voyageur" peut être réservé par plusieurs clients à différentes dates, mais chaque réservation unique ne concerne que ce bateau.

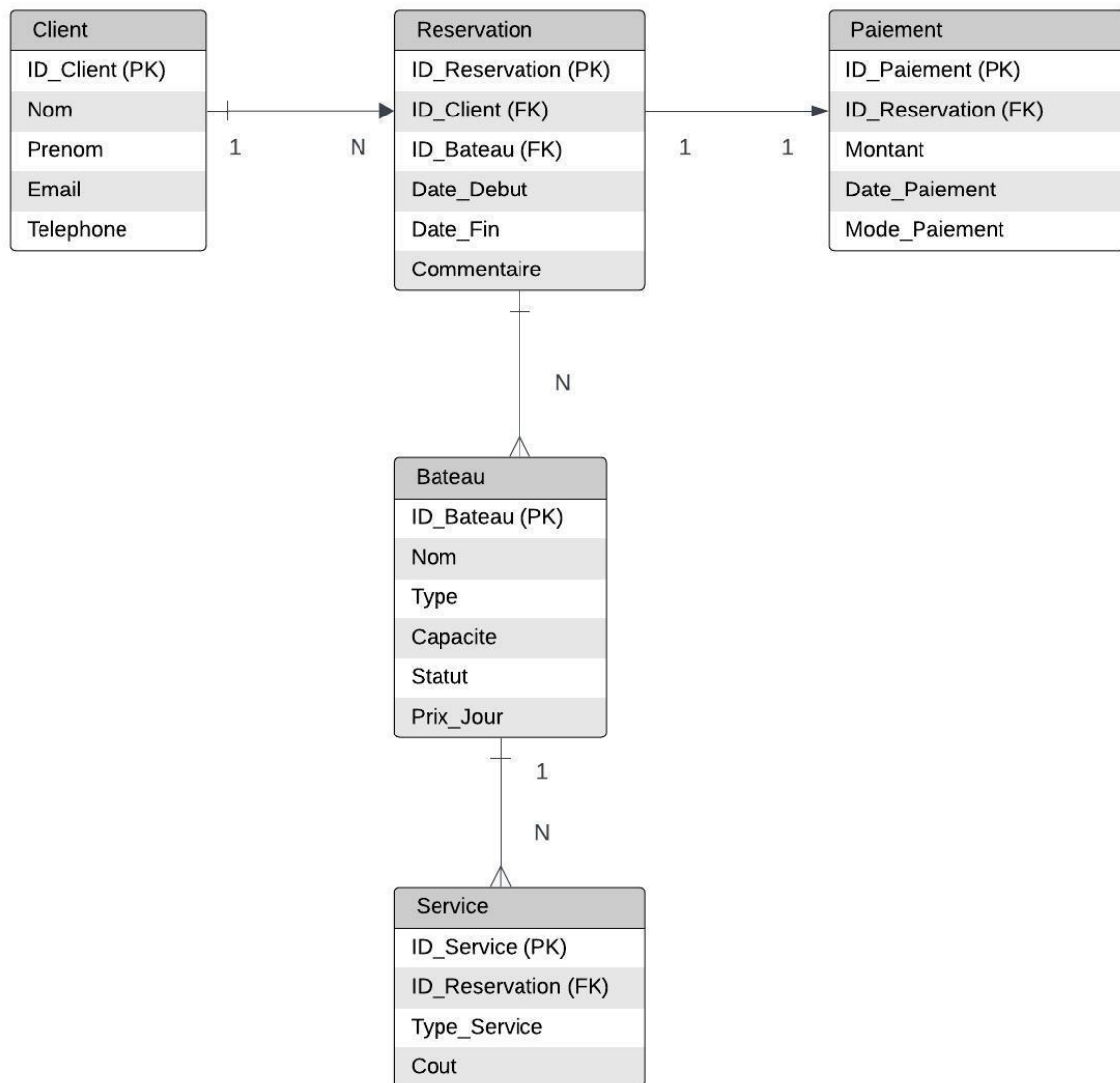
? Réserve (1,1) --- est payée par --- (1,1) Paiement

- **Description** : Chaque réservation génère un seul paiement, et chaque paiement est associé à une seule réservation. Cette association garantit que pour chaque réservation confirmée, il y a une transaction enregistrée dans le système.
- **Cardinalité** : (1,1) de part et d'autre implique une relation strictement un-à-un entre réservation et paiement.
- **Exemple** : La réservation ID 101 pour le bateau ID 3 a un paiement de 800€ enregistré sous un paiement ID spécifique.

? Réserve (1,N) --- comprend --- (0,N) Service

- **Description** : Une réservation peut inclure plusieurs services optionnels (comme le personnel de bord ou des équipements additionnels). Ces services sont ajoutés pour une réservation spécifique et ne sont pas partagés avec d'autres réservations.
 - **Cardinalité** : (1, N) pour la réservation signifie qu'elle peut inclure plusieurs services, alors que (0, N) pour les services indique qu'un service est attaché à une réservation unique.
 - **Exemple** : La réservation ID 102 inclut le service "Personnel de bord" et "Équipement de plongée", chacun lié uniquement à cette réservation.
-

Modèle Entité-Association MERISE (MCD)



Client-Reservation : Un client peut effectuer plusieurs réservations, mais chaque réservation est associée à un seul client.

Reservation-Bateau : Un bateau peut être réservé plusieurs fois, mais une réservation ne concerne qu'un bateau à la fois.

Reservation-Paiement : Chaque réservation est associée à un paiement unique, assurant une correspondance stricte.

Reservation-Service : Une réservation peut inclure plusieurs services, mais chaque service appartient à une réservation unique.

Modèle Logique de Données (MLD)

1. Table **Client**

Attribut	Type de Donnée	Contraintes
ID_Client	NUMBER(10)	PRIMARY KEY
Nom	VARCHAR2(50)	NOT NULL
Prenom	VARCHAR2(50)	NOT NULL
Email	VARCHAR2(100)	NOT NULL UNIQUE
Telephone	VARCHAR2(15)	NOT NULL

2. Table **Bateau**

Attribut	Type de Donnée	Contraintes
ID_Bateau	NUMBER(10)	PRIMARY KEY
Nom	VARCHAR2(50)	NOT NULL UNIQUE
Type	VARCHAR2(50)	NOT NULL
Capacite	NUMBER(3)	NOT NULL CHECK (Capacite > 0)
Statut	VARCHAR2(20)	NOT NULL CHECK (Statut IN ('Disponible', 'Loue', 'En Maintenance'))
Prix_Jour	NUMBER(10,2)	NOT NULL CHECK (Prix_Jour > 0)

3. Table **Reservation**

Attribut	Type de Donnée	Contraintes
ID_Reservation	NUMBER(10)	PRIMARY KEY
ID_Client	NUMBER(10)	NOT NULL, FOREIGN KEY (ID_Client) REFERENCES Client(ID_Client) ON DELETE CASCADE
ID_Bateau	NUMBER(10)	NOT NULL, FOREIGN KEY (ID_Bateau) REFERENCES Bateau(ID_Bateau) ON DELETE CASCADE
Date_Debut	DATE	NOT NULL
Date_Fin	DATE	NOT NULL
Commentaire	VARCHAR2(255)	

1. Table Client

Attribut	Type de Donnée	Contraintes
ID_Client	NUMBER(10)	PRIMARY KEY
Nom	VARCHAR2(50)	NOT NULL
Prenom	VARCHAR2(50)	NOT NULL
Email	VARCHAR2(100)	NOT NULL UNIQUE
Telephone	VARCHAR2(15)	NOT NULL

2. Table Bateau

Attribut	Type de Donnée	Contraintes
ID_Bateau	NUMBER(10)	PRIMARY KEY
Nom	VARCHAR2(50)	NOT NULL UNIQUE
Type	VARCHAR2(50)	NOT NULL
Capacite	NUMBER(3)	NOT NULL CHECK (Capacite > 0)
Statut	VARCHAR2(20)	NOT NULL CHECK (Statut IN ('Disponible', 'Loue', 'En Maintenance'))
Prix_Jour	NUMBER(10,2)	NOT NULL CHECK (Prix_Jour > 0)

3. Table Reservation

Attribut	Type de Donnée	Contraintes
ID_Reservation	NUMBER(10)	PRIMARY KEY
ID_Client	NUMBER(10)	NOT NULL, FOREIGN KEY (ID_Client) REFERENCES Client(ID_Client) ON DELETE CASCADE
ID_Bateau	NUMBER(10)	NOT NULL, FOREIGN KEY (ID_Bateau) REFERENCES Bateau(ID_Bateau) ON DELETE CASCADE
Date_Debut	DATE	NOT NULL
Date_Fin	DATE	NOT NULL
Commentaire	VARCHAR2(255)	

4. Table Paiement

Attribut	Type de Donnée	Contraintes
ID_Paiement	NUMBER(10)	PRIMARY KEY
ID_Reservation	NUMBER(10)	UNIQUE NOT NULL, FOREIGN KEY (ID_Reservation) REFERENCES Reservation(ID_Reservation) ON DELETE CASCADE
Montant	NUMBER(10,2)	NOT NULL CHECK (Montant > 0)
Date_Paiement	DATE	NOT NULL
Mode_Paiement	VARCHAR2(20)	NOT NULL CHECK (Mode_Paiement IN ('Carte', 'Virement', 'Espèces'))

5. Table Service

Attribut	Type de Donnée	Contraintes
ID_Service	NUMBER(10)	PRIMARY KEY
ID_Reservation	NUMBER(10)	NOT NULL, FOREIGN KEY (ID_Reservation) REFERENCES Reservation(ID_Reservation) ON DELETE CASCADE
Type_Service	VARCHAR2(50)	NOT NULL
Cout	NUMBER(10,2)	NOT NULL CHECK (Cout >= 0)

Spécification des Traitements avec des Packages PL/SQL

1. Package pkg_client

Ce package gère les opérations relatives aux clients dans la table client, y compris l'ajout, la modification, la suppression, la consultation et le comptage des clients.

Spécification

- **Variables globales** : v_total_clients pour stocker le nombre total de clients.
- **Exceptions personnalisées** :
 - e_client_not_found : Levée lorsque le client spécifié n'est pas trouvé.
 - e_email_duplicate : Levée en cas de doublon d'email.
- **Procédures et fonctions** :

- ajouter_client : Insère un nouveau client après vérification de l'unicité de l'email.
- modifier_client : Modifie l'email et le téléphone d'un client existant.
- supprimer_client : Supprime un client si aucune réservation future n'est enregistrée.
- consulter_client : Retourne les informations d'un client via un SYS_REFCURSOR.
- compter_clients : Calcule et affiche le nombre total de clients.

Implémentation

1. **ajouter_client** :
 - Vérifie l'unicité de l'email et génère un nouvel identifiant.
 2. **modifier_client** :
 - Vérifie l'existence du client avant la mise à jour des informations.
 3. **supprimer_client** :
 - Supprime le client après vérification de l'absence de réservations futures.
 4. **consulter_client** :
 - Utilise un curseur pour récupérer les informations du client.
 5. **compter_clients** :
 - Affiche le total des clients enregistrés dans la base.
-

2. Package pkg_bateau

Ce package gère les opérations sur la table Bateau, y compris l'ajout, la modification, la suppression, la consultation et le comptage des bateaux.

Spécification

- **Variables globales** : v_total_bateaux pour stocker le nombre de bateaux.
- **Exceptions personnalisées** :
 - e_bateau_not_found : Levée si le bateau n'existe pas.
 - e_nom_bateau_duplicate : Levée si le nom du bateau est déjà utilisé.
- **Procédures et fonctions** :
 - ajouter_bateau : Insère un nouveau bateau avec vérification de l'unicité du nom.
 - modifier_statut_bateau : Change le statut d'un bateau (ex. : En Maintenance).
 - supprimer_bateau : Supprime un bateau sans réservations futures.
 - consulter_bateau : Récupère les détails d'un bateau.
 - compter_bateaux : Compte le nombre total de bateaux.

Implémentation

1. **ajouter_bateau** :
 - Vérifie le nom et génère un nouvel identifiant pour le bateau.
2. **modifier_statut_bateau** :
 - Change le statut après validation de l'existence du bateau.
3. **supprimer_bateau** :

- Supprime le bateau s'il n'est pas lié à des réservations en cours.
 - 4. **consulter_bateau** :
 - Retourne un curseur sur les informations du bateau.
 - 5. **compter_bateaux** :
 - Affiche le nombre total de bateaux.
-

3. Package `pkg_reservation`

Ce package gère les réservations, comprenant la création, la modification, l'annulation et la consultation des réservations.

Spécification

- **Exceptions personnalisées** :
 - `e_reservation_not_found` : Levée si la réservation est introuvable.
 - `e_bateau_non_disponible` : Levée si le bateau est réservé.
 - `e_bateau_en_maintenance` : Levée si le bateau est en maintenance.
- **Procédures et fonctions** :
 - `creer_reservation` : Crée une réservation après vérification de la disponibilité du bateau.
 - `modifier_reservation` : Change les dates d'une réservation si le bateau est disponible.
 - `annuler_reservation` : Annule une réservation et met à jour le statut du bateau.
 - `consulter_reservation` : Retourne les informations de la réservation.

Implémentation

1. **creer_reservation** :
 - Vérifie le statut du bateau avant insertion.
 - Met à jour le statut du bateau à "Loué".
 2. **modifier_reservation** :
 - Change les dates de la réservation, avec vérification de disponibilité.
 3. **annuler_reservation** :
 - Supprime la réservation et remet le bateau en "Disponible" si nécessaire.
 4. **consulter_reservation** :
 - Fournit les détails de la réservation via un curseur.
-

4. Package `pkg_paiement`

Ce package gère les paiements, incluant l'enregistrement, la modification, la suppression et la consultation des transactions.

Spécification

- **Exceptions personnalisées :**
 - `e_paiement_not_found` : Levée lorsque le paiement est introuvable.
 - `e_paiement_already_exists` : Levée si un paiement pour la même réservation existe déjà.
- **Procédures et fonctions :**
 - `enregistrer_paiement` : Insère un paiement pour une réservation.
 - `modifier_paiement` : Modifie le montant d'un paiement existant.
 - `supprimer_paiement` : Supprime un paiement spécifique.
 - `consulter_paiement` : Affiche les détails d'un paiement.

Implémentation

1. **`enregistrer_paiement` :**
 - Insère un paiement pour une réservation unique.
 2. **`modifier_paiement` :**
 - Change le montant d'un paiement après vérification de son existence.
 3. **`supprimer_paiement` :**
 - Supprime un paiement après validation de son existence.
 4. **`consulter_paiement` :**
 - Renvoie un curseur avec les informations du paiement.
-

5. Package `pkg_service`

Ce package gère les services supplémentaires associés aux réservations.

Spécification

- **Exceptions personnalisées :**
 - `e_service_not_found` : Levée si le service n'existe pas.
- **Procédures et fonctions :**
 - `ajouter_service` : Ajoute un service pour une réservation.
 - `supprimer_service` : Supprime un service existant.
 - `consulter_service` : Retourne les informations du service.

Implémentation

1. **`ajouter_service` :**
 - Ajoute un service pour une réservation avec un coût défini.
 2. **`supprimer_service` :**
 - Supprime le service après vérification de son existence.
 3. **`consulter_service` :**
 - Utilise un curseur pour renvoyer les informations du service.
-

Tests des Packages

Chaque package est testé avec des blocs DECLARE et BEGIN/END, où les principales opérations sont effectuées pour vérifier le bon fonctionnement des procédures et fonctions, par exemple :

- **Test pkg_client** : Ajoute, consulte, compte, modifie et supprime un client.
 - **Test pkg_bateau** : Ajoute, consulte, compte, modifie le statut, et supprime un bateau.
 - **Test pkg_reservation** : Crée, consulte, modifie, et annule une réservation.
 - **Test pkg_paiement** : Enregistre, consulte, modifie, et supprime un paiement.
 - **Test pkg_service** : Ajoute, consulte et supprime un service.
-

Triggers

1. **Trigger trg_verif_statut_bateau** : Avant une insertion ou modification dans Reservation, il vérifie que le bateau est disponible et qu'il n'y a pas de conflit de dates.
 2. **Trigger trg_maj_statut_bateau_apres_suppression** : Après la suppression d'une réservation, ce trigger remet à jour le statut du bateau en "Disponible" s'il n'a pas de réservations futures.
-