En informatique, un objet d'accès aux données (DAO) est un objet qui fournit une interface abstraite vers un type de base de données. L’interface DAO sert à fournir certaines opérations spécifiques de données sans exposer les détails de la base de données.

Cet isolement sépare les préoccupations :

* Quelles données accèdent aux besoins de l'application, en termes de domaines spécifiques d'objets et types de données (l'interface publique de la DAO)
* Comment ces besoins peuvent être satisfaits avec un SGBD spécifique, le schéma de base de données, etc. (la mise en œuvre de la DAO).

Bien que ce modèle de conception soit également applicable à la plupart des langages de programmation, il est traditionnellement associé aux applications Java EE et bases de données relationnelles accessibles via l'API JDBC.

Voici nos DAO :

* **ClientDAO**
* Relation dans laquelle elle est impliquée : Lors de l’expédition un assemblage est envoyé à un client spécifié dans un bon d’expédition.
* Fonctionnalités :
* Ajouter/Créer un client dans la table Client (entrée : le client à ajouter, sortie : nombre de lignes ajoutées à la table)
* Supprimer un client de la table (entrée : l’ID du client à supprimer, sortie : nulle si aucun client ne correspond à l’ID)
* Trouver un client à partir de son ID (entrée : l’ID du client à trouver, sortie : nulle si aucun client ne correspond à l’ID)
* Trouver un client à partir de son nom (entrée : le nom du client à trouver, sortie : nulle si aucun client ne correspond au nom recherché)
* Récupérer la liste de tous les clients de la table (entrée : rien, sortie : la liste des clients)
* **FournisseurDAO**
* Relation dans laquelle elle est impliquée : le fournisseur livre les pièces.
* Fonctionnalités :
* Ajouter/Créer un fournisseur dans la table Fournisseur (entrée : le fournisseur à ajouter, sortie : nombre de lignes ajoutées à la table)
* Supprimer un fournisseur de la table (entrée : l’ID du fournisseur à supprimer, sortie : nulle si aucun fournisseur ne correspond à l’ID)
* Trouver un fournisseur à partir de son ID (entrée : l’ID du fournisseur à trouver, sortie : nulle si aucun fournisseur ne correspond à l’ID)
* Trouver un fournisseur à partir de son nom (entrée : le nom du fournisseur à trouver, sortie : nulle si aucun fournisseur ne correspond au nom recherché)
* Récupérer la liste de tous les fournisseurs de la table (entrée : rien, sortie : la liste des fournisseurs)
* **PieceDAO**
* Relation dans laquelle elle est impliquée : Une pièce est livrée par un fournisseur puis subie des contrôles et opérations pour être intégrée à un assemblage.
* Fonctionnalités :
* Ajouter une pièce dans la table Pièce (entrée : la pièce à ajouter, sortie : nombre de lignes ajoutées à la table)
* Supprimer une pièce de la table (entrée : l’ID de la pièce à supprimer, sortie : nulle si aucune pièce ne correspond à l’ID)
* Supprimer une pièce de la table (entrée : le code barre de la pièce à supprimer, sortie : nulle si aucune pièce ne correspond au code barre)
* Modifier le statut d’une pièce (entrée : l’ID de la pièce à modifier et le nouveau statut, sortie : nombre de lignes modifiées dans la table)
* Modifier le code barre d’une pièce (entrée : l’ID de la pièce à modifier et le nouveau code barre, sortie : nombre de lignes modifiées dans la table)
* Trouver une pièce à partir de son ID (entrée : l’ID de la pièce à trouver, sortie : nulle si aucun pièce ne correspond à l’ID)
* Trouver une pièce à partir de son code barre (entrée : le code barre de la pièce à trouver, sortie : nulle si aucun pièce ne correspond à l’ID)
* Récupérer la liste de toutes les pièces de la table (entrée : rien, sortie : la liste des pièces)
* **AssemblageDAO**
* Relation dans laquelle elle est impliquée : L’ordonnancement lance la production d’un assemblage, composé de pièces
* Fonctionnalités :
* Ajouter un assemblage dans la table Assemblage (entrée : l’assemblage à ajouter, sortie : nombre de lignes ajoutées à la table)
* Ajouter une pièce dans un assemblage (entrée : l’ID de l’assemblage concerné et l’ID de la pièce à ajouter, sortie : true si succès et false si echec)
* Supprimer une pièce dans un assemblage (entrée : l’ID de l’assemblage concerné et l’ID de la pièce à supprimer, sortie : true si succès et false si echec)
* Supprimer un assemblage de la table (entrée : l’ID de l’assemblage à supprimer, sortie : nulle si aucun assemblage ne correspond à l’ID)
* Supprimer un assemblage de la table (entrée : la référence de l’assemblage à supprimer, sortie : nulle si aucun assemblage ne correspond à cette référence)
* Trouver un assemblage à partir de son ID (entrée : l’ID de l’assemblage à trouver, sortie : nulle si aucun assemblage ne correspond à l’ID)
* Trouver un assemblage à partir de sa référence (entrée : la référence de l’assemblage à trouver, sortie : nulle si aucun assemblage ne correspond à cette référence)
* Récupérer la liste de tous les assemblages de la table (entrée : rien, sortie : la liste des assemblages)
* **OperateurDAO**
* Relation dans laquelle elle est impliquée : L’operateur réalise les opérations sur les pièces afin de constituer les assemblages
* Fonctionnalités :
* Ajouter/Créer un operateur dans la table Operateur (entrée : l’operateur à ajouter, sortie : nombre de lignes ajoutées à la table)
* Supprimer un operateur de la table (entrée : l’ID de l’operateur à supprimer, sortie : nulle si aucun operateur ne correspond à l’ID)
* Trouver un operateur à partir de son ID (entrée : l’ID de l’operateur à trouver, sortie : nulle si aucun operateur ne correspond à l’ID)
* Trouver un operateur à partir de son nom et son prénom (entrée : le nom et le prénom de l’opérateur à trouver, sortie : nulle si aucun operateur ne correspond au nom et au prénom recherchés)
* Récupérer la liste de tous les operateurs de la table (entrée : rien, sortie : la liste des operateurs)
* **OperationDAO**
* Relation dans laquelle elle est impliquée : Des opérations de type mesures, contrôles, intégrations sont réalisées par l’opérateur sur les pièces et assemblages
* Fonctionnalités :
* Créer une opération dans la table Opération (entrée : l’opération à ajouter, sortie : nombre de lignes ajoutées à la table)
* Modifier le libellé d’une opération (entrée : l’ID de l’opération et le nouveau libellé, sortie : nombre de lignes modifiées à la table)
* Modifier l’état d’une opération (entrée : l’ID de l’opération et le nouvel état, sortie : nombre de lignes modifiées à la table)
* Supprimer une opération de la table (entrée : l’ID de l’opération à supprimer, sortie : nulle si aucune opération ne correspond à l’ID)
* Supprimer une opération de la table (entrée : le libellé de l’opération à supprimer, sortie : nulle si aucune opération ne correspond à ce libellé)
* Trouver une opération à partir de son ID (entrée : l’ID de l’opération à trouver, sortie : nulle si aucune opération ne correspond à l’ID)
* Trouver une opération à partir de son libellé (entrée : le libellé de l’opération à trouver, sortie : nulle si aucune opération ne correspond au libellé)
* Récupérer la liste de toutes les opérations de la table (entrée : rien, sortie : la liste des opérations)
* **EtatOperationDAO**
* Relation dans laquelle elle est impliquée : Une opération est associée à un état (en cours, en attente, suspendue, réalisée, annulée, contrôle ok etc.)
* Fonctionnalités :
* Ajouter un état opération dans la table EtatOperation (entrée : l’état opération à ajouter, sortie : nombre de lignes ajoutées à la table)
* Supprimer un état opération de la table (entrée : l’ID de l’état opération à supprimer, sortie : nulle si aucun état opération ne correspond à l’ID)
* Trouver un état opération à partir de son ID (entrée : l’ID de l’état opération à trouver, sortie : nulle si aucun état opération ne correspond à l’ID)
* Récupérer la liste de tous les états opération de la table (entrée : rien, sortie : la liste des clients)
* **LieuDAO**
* Relation dans laquelle elle est impliquée : Les pièces et assemblages transitent ou sont stockées dans des lieux tout au long de leur séjour au sein de l’entreprise
* Fonctionnalités :
* Ajouter un lieu dans la table Lieu (entrée : le lieu à ajouter, sortie : nombre de lignes ajoutées à la table)
* Supprimer un lieu de la table (entrée : l’ID du lieu à supprimer, sortie : nulle si aucun lieu ne correspond à l’ID)
* Trouver un lieu à partir de son ID (entrée : l’ID du lieu à trouver, sortie : nulle si aucun lieu ne correspond à l’ID)
* Récupérer la liste de tous les lieux de la table (entrée : rien, sortie : la liste des lieux)
* **LocalisationPiece**
* Relation dans laquelle elle est impliquée : Contient la localisation d’une pièce (pièce, date, lieu, opérateur)
* Fonctionnalités :
* Ajouter une localisation pièce dans la table localisation pièce (entrée : la localisation pièce à ajouter, sortie : nombre de lignes ajoutées à la table)
* Supprimer une localisation pièce de la table (entrée : l’ID de la localisation pièce à supprimer, sortie : nulle si aucune localisation pièce ne correspond à l’ID)
* Trouver une localisation pièce à partir de son ID (entrée : l’ID de la localisation pièce à trouver, sortie : nulle si aucune localisation pièce ne correspond à l’ID)
* Trouver une localisation pièce à partir de l’ID de la pièce recherchée (entrée : l’ID de la pièce, sortie : nulle si aucune localisation pièce ne correspond à la pièce recherchée)
* Récupérer la liste de toutes les localisations pièce de la table (entrée : rien, sortie : la liste des localisations pièces)
* Récupérer la liste de toutes les localisations pièce dans un lieu précis (entrée : l’ID du lieu, sortie : la liste des localisations pièces)
* **LocalisationAssemblageDAO**
* Relation dans laquelle elle est impliquée : Contient la localisation d’un assemblage (assemblage, date, lieu, opérateur)
* Fonctionnalités :
* Ajouter une localisation assemblage dans la table LocalisationAssemblage (entrée : la localisation assemblage à ajouter, sortie : nombre de lignes ajoutées à la table)
* Supprimer une localisation assemblage de la table (entrée : l’ID de la localisation assemblage à supprimer, sortie : nulle si aucune localisation assemblage ne correspond à l’ID)
* Trouver une localisation assemblage à partir de son ID (entrée : l’ID de la localisation assemblage à trouver, sortie : nulle si aucune localisation assemblage ne correspond à l’ID)
* Trouver une localisation assemblage à partir de l’ID de l’assemblage recherchée (entrée : l’ID de l’assemblage, sortie : nulle si aucune localisation assemblage ne correspond à l’assemblage recherché)
* Récupérer la liste de toutes les localisations assemblage de la table (entrée : rien, sortie : la liste des localisations assemblages)
* Récupérer la liste de toutes les localisations assemblage dans un lieu précis (entrée : l’ID du lieu, sortie : la liste des localisations assemblage)
* **OrdonnancementDAO**
* Relation dans laquelle elle est impliquée : L’ordonnancement lance les opérations de production
* Fonctionnalités :
* Créer un ordonnancement dans la table Ordonnancement (entrée : l’ordonnancement à ajouter, sortie : nombre de lignes ajoutées à la table)
* Supprimer un ordonnancement de la table (entrée : l’ID de l’ordonnancement à supprimer, sortie : nulle si aucun ordonnancement ne correspond à l’ID)
* Modifier la quantité à fabriquer d’un assemblage (entrée : l’ID de l’ordonnancement et la nouvelle quantité, sortie : nulle si aucun ordonnancement ne correspond à l’ID)
* Trouver un ordonnancement à partir de son ID (entrée : l’ID de l’ordonnancement à trouver, sortie : nulle si aucun ordonnancement ne correspond à l’ID)
* Récupérer la liste de tous les ordonnancements de la table (entrée : rien, sortie : la liste des ordonnancements)