

# Projet BD 2025-2026

Équipe BD

## 1 Introduction

Le projet auquel vous allez participer a pour but de mettre en œuvre vos compétences en systèmes de gestion de bases de données relationnelles. De plus, vous vous essaieriez à la programmation d'application utilisant une base de données (à travers des transactions). Le développement du démonstrateur sera fait en Java en utilisant l'API JDBC. Le projet est à faire en équipes de cinq et donnera lieu à une réunion de suivi intermédiaire, ainsi qu'à une soutenance en fin de projet. La constitution des équipes et la remise des livrables (rapport final et code source) se fera sur l'application Teide. L'évaluation se fera uniquement sur les aspects bases de données du projet en l'état au moment de la soutenance.

## 2 Description de l'application

L'épicerie locale "Au Valaisan" souhaite moderniser son système d'information pour gérer son épicerie spécialisée dans la vente de produits en vrac et de produits locaux. Elle vous propose de mettre en place la base de données et l'application interagissant avec elle, ainsi que de tester certaines fonctionnalités critiques.

### 2.1 Les produits et les producteurs

Les produits proposés proviennent de producteurs locaux. Chaque producteur possède un identifiant unique (le choix du type d'identifiant est laissé à votre appréciation – justifiez votre choix dans le rapport). Un producteur est décrit par son nom, son adresse complète incluant la géolocalisation (latitude, longitude), ses coordonnées de contact (dont l'e-mail) et son type d'activité (agriculteur, éleveur, fromager, etc.).

Un producteur peut fournir plusieurs produits et avoir plusieurs activités.

Un produit possède un identifiant unique et est caractérisé par son nom, sa catégorie (fromage, bison, céréales, légumineuses, fruits secs, huiles, etc.), sa description, son producteur d'origine, et des caractéristiques spécifiques (bio, label, allergènes, origine géographique, etc.).

**Important :** Un produit provient d'un unique producteur. Si deux producteurs fournissent des produits analogues (deux fournisseurs de raclette par exemple), ils correspondent à deux entrées distinctes dans la base de données.

#### Saisonnalité

Certains produits ne sont disponibles que durant certaines périodes de l'année (par exemple, les abricots du Valais de mi-juin à septembre). Un produit peut avoir zéro, une ou plusieurs périodes de disponibilité annuelles. Ces périodes de disponibilité doivent être spécifiées et prises en compte pour la gestion des commandes.

#### Conditionnement

Les produits se déclinent selon un ou deux modes de conditionnement :

- **Vrac :** vendu au poids (prix au kg), le client choisit la quantité désirée

- **Pré-conditionné** : vendu en sachets pré-pesés de poids fixe (exemple : sachet de 500g de lentilles)

Un même produit peut être proposé selon les deux modes simultanément (exemple : lentilles disponibles en vrac ET en sachets pré-conditionnés).

Pour chaque produit, l'épicerie définit un prix d'achat producteur (ce que paie l'épicerie au producteur) et un prix de vente TTC au client. La marge réalisée doit pouvoir être calculée pour des besoins de comptabilité. Les prix peuvent varier selon le conditionnement.

## 2.2 Les contenants et accessoires

Pour le vrac, les clients peuvent acquérir des contenants : bocaux en verre réutilisables, sachets kraft, sachets en tissu, papier ciré, etc.

Ces contenants sont vendus séparément, avec leur propre référence, prix et stock. Un contenant est caractérisé par son type, sa capacité (en litres ou en grammes), le stock disponible et son caractère réutilisable ou jetable.

## 2.3 La gestion des stocks de produits

L'épicerie gère un stock pour chaque produit. Afin de gérer les différentes dates de péremptions d'un produit chaque stock est scindé en lots caractérisé par :

- La quantité disponible (en kg pour le vrac, en unités pour le pré-conditionné et les contenants)
- La date de réception
- La date de péremption (DLC - Date Limite de Consommation) ou la DLUO (Date Limite d'Utilisation Optimale)

Le stock total d'un produit correspond donc à la somme de quantité disponible dans chacun de ses lots. **Important** : Un lot de stock possède soit une DLC soit une DLUO, mais jamais les deux simultanément (les deux concepts sont mutuellement exclusifs). Cette contrainte doit être implémentée dans votre modèle.

**Hypothèse simplificatrice** : Pour un produit donné, on considère qu'il y a au maximum une livraison par jour.

Le système doit alerter automatiquement les gérants lorsqu'une date de péremption ou DLUO approche (par exemple, 7 jours avant). Dans ce cas, le prix du produit peut être automatiquement réduit pour écouler le stock.

Le système doit également gérer les pertes : vols constatés et casse (bocaux cassés, produits abîmés, etc.). Ces pertes doivent être enregistrées avec leur date, leur nature et la quantité concernée pour des besoins de comptabilité et d'analyse.

## 2.4 Les clients et leurs comptes

Les clients sont identifiés par leur adresse e-mail et décrits par leur nom, prénom, numéro de téléphone et adresse postale de livraison. Un client peut avoir plusieurs adresses de livraison enregistrées.

Le système doit respecter le droit à l'oubli : un client peut demander la suppression de ses données personnelles. Dans ce cas, ses données sont anonymisées tout en conservant l'historique des commandes pour la comptabilité (obligation légale de conservation pendant 10 ans).

## 2.5 Les commandes et modalités d'achat

Une commande est passée par un client à une date et heure données. Elle est identifiée de façon unique et possède un statut évolutif :

- **En préparation** : commande enregistrée, paiement validé

- **Prête** : commande préparée, en attente de retrait ou d'expédition
- **En livraison** : colis expédié (uniquement pour les livraisons)
- **Récupérée/Livrée** : commande finalisée
- **Annulée** : commande annulée par le client ou l'épicerie

Une commande se compose de lignes de commande, chacune concernant un produit spécifique avec :

- La quantité commandée (en kg pour le vrac, en unités pour le pré-conditionné)
- Le prix unitaire au moment de la commande
- Le sous-total de la ligne

### **Gestion du stock pendant le cycle de vie d'une commande**

Pour la gestion du stock lors du passage de commandes les contraintes suivantes sont imposées :

- Annulation par le client autorisée tant que la commande est "en préparation"
- Les articles restent disponibles à la vente pendant la préparation
- Lors du passage au statut "prête", les produits sortent du stock et le client ne peut plus annuler
- L'épicerie peut toujours annuler jusqu'au statut "expédié"

Vous devrez gérer correctement les problèmes de concurrence qui peuvent survenir avec cette stratégie.

### **Modes de paiement**

Le client peut choisir son mode de paiement :

- Paiement en ligne : lors de la commande (carte bancaire)
- Paiement en boutique : au moment du retrait

### **Modes de récupération**

Le client peut choisir son mode de récupération :

- Retrait en boutique : aucun frais supplémentaire, le client vient chercher sa commande
- Livraison à domicile : des frais de livraison s'appliquent. Ces frais varient selon :
  1. Le pays de livraison (France métropolitaine, DOM-TOM, international)
  2. Le poids total de la commande
  3. La distance entre la boutique et l'adresse de livraison
- Pour une commande avec livraison, une date de livraison estimée doit être calculée et communiquée au client.

**Note :** DOM-TOM = Départements et Territoires d'Outre-Mer. Les frais d'expédition y sont significativement plus élevés qu'en métropole.

## 2.6 Gestion de la disponibilité

La disponibilité d'un produit dépend de plusieurs facteurs :

- Pour les produits en stock :
  - Si la quantité en stock est suffisante : disponibilité immédiate
  - Si hors saison : indisponible avec indication de la prochaine période de disponibilité
- Pour les produits sur commande (certains produits rares ou spécifiques) :
  - Le délai de disponibilité doit être spécifié par produit (exemple : 72h, 1 semaine)
  - Le client est informé du délai lors de sa commande

Le nombre de produits disponibles en stock doit être affiché en temps réel sur la plateforme en ligne pour éviter les commandes impossibles à honorer.

## 3 Travail à réaliser (15h encadrées + travail personnel, le tout $\times 5$ )

Le travail se découpe en quatre étapes, décrites ci-dessous, dont seules les deux premières sont obligatoirement séquentielles. L'étape 3.3 peut être découpée et parallélisée, et l'étape 3.4 peut être commencée dès le début du projet.

### 3.1 Modélisation du problème

La modélisation se décompose en deux temps.

Dans un premier temps, vous aurez à analyser le problème posé pour en extraire les propriétés élémentaires, les dépendances fonctionnelles reliant ces propriétés, ainsi que tous les autres types de contraintes (contraintes de valeur, contraintes de multiplicité et contraintes contextuelles).

Vous devrez proposer ensuite un schéma Entités/Associations représentant les données nécessaires à l'application et leurs liens sémantiques (ce qui correspond à l'état cohérent de la base de données).

Le schéma Entités/Associations doit être totalement justifié par l'analyse. Expliquez vos choix de modélisation dans la documentation du projet, notamment :

- La gestion des deux modes de conditionnement (vrac/pré-conditionné)
- La modélisation de la saisonnalité
- La distinction entre produits et contenants
- La gestion des pertes (vols et casse)
- La mise en œuvre du droit à l'oubli
- La gestion des dates de péremption et DLUO
- Le calcul des frais de livraison
- La stratégie de gestion du stock pendant le cycle de vie d'une commande

## 3.2 Implantation de la base de données

Vous devrez ensuite traduire le schéma Entités/Associations en un schéma relationnel décrivant les noms des relations obtenues, les noms et types de leurs attributs, ainsi que les contraintes à vérifier pour chacune des relations. Vous devrez également justifier vos choix de traduction et expliquer les points difficiles.

Précisez et justifiez la forme normale de chacune des relations obtenues.

Vous implanterez ensuite ce schéma relationnel sur le SGBD Oracle disponible sur le serveur oracle1. Vous devrez insérer suffisamment de données pertinentes pour la suite du projet, en veillant à couvrir:

- Plusieurs producteurs de types différents
- Des produits en vrac et pré-conditionnés
- Des produits avec différentes saisonnalités
- Des contenants variés
- Des stocks avec dates de péremption diverses
- Des commandes dans différents états
- Des exemples de pertes (vols, casse)

## 3.3 Analyse des fonctionnalités

Vous devrez définir les requêtes SQL nécessaires pour réaliser les fonctionnalités suivantes en les regroupant en transactions, ce qui permettra d'assurer la cohérence globale de la base de données, même en cas d'accès concurrents :

### Fonctionnalité 1 : Passage d'une commande par un client

Cette transaction doit :

- Vérifier la disponibilité de chaque produit commandé (stock suffisant, respect de la saisonnalité)
- Calculer le montant total de la commande (produits + éventuels frais de livraison)
- Gérer le stock selon la stratégie choisie (voir section 2.5)
- Créer la commande avec le statut approprié
- Gérer les cas d'erreur (stock insuffisant, produit hors saison)

### Fonctionnalité 2 : Alertes de péremption et ajustement des prix

Cette transaction doit :

- Identifier tous les produits dont la date de péremption ou DLUO approche (dans les 7 jours)
- Proposer une réduction automatique de prix (par exemple 30% de réduction)
- Générer une liste d'alertes pour les gérants
- Mettre à jour les prix en conséquence

### Fonctionnalité 3 : Clôture d'une commande (retrait ou livraison)

Cette transaction doit :

- Mettre à jour le statut de la commande selon le mode de récupération
- Si paiement en boutique : enregistrer le paiement
- Si livraison : calculer et afficher les frais de livraison finaux
- Gérer le stock selon la stratégie choisie et le mode de paiement
- Enregistrer la date effective de récupération/livraison

Ces requêtes et transactions doivent être testées sur Oracle (avec SQL\*Plus ou SQL Developer, Adminer ne gérant pas les transactions) pour en vérifier leur bon fonctionnement, y compris pour des exécutions concurrentes.

Points d'attention pour les tests de concurrence :

- Deux clients commandant simultanément le dernier stock d'un produit
- Mise à jour de prix pendant qu'une commande est en cours

### 3.4 Implantation des fonctionnalités

Les fonctionnalités précédemment étudiées devront être implantées dans un démonstrateur programmé en Java/JDBC. Nous vous recommandons l'implantation d'une interface textuelle simple (quelques menus et saisies utilisateurs/affichages).

L'interface devrait permettre à minima :

- De consulter le catalogue de produits avec disponibilités
- De passer une commande (choix produits, quantités, mode paiement, mode récupération)
- De consulter les alertes de péremption
- De clôturer une commande

**ATTENTION!!! L'évaluation ne portera que sur les aspects Base de Données et pas sur le code Java.**

## 4 Déroulement du projet

Le projet sera constitué de 15 heures en séances encadrées auxquelles s'ajoutent les soutenances de 45min (vous participez seulement à la soutenance de votre équipe).

### 4.1 Séances encadrées

Comme toute épreuve pratique à l'Ensimag, la présence aux séances encadrées est obligatoire et sera contrôlée.

La première séance comprendra une présentation du projet. Une fois le projet lancé (après une ou deux séances), vous aurez une présentation de JDBC et de l'analyse de fonctionnalités.

Vous pourrez poser des questions à votre encadrant à tout moment lors des séances, mais il jouera deux rôles alternativement :

- Il sera soit votre client (le gérant de l'épicerie qui connaît les besoins applicatifs, mais ne connaît rien en base de données)
- Soit un expert en bases de données (qui ne connaît pas et ne veut pas connaître les besoins applicatifs)

Posez bien vos questions.

## 4.2 Outils

Vous disposez de deux outils principaux pour le bon déroulement du projet :

- **Chamilo** : vous y trouverez les documentations techniques pour accéder à Oracle et pour utiliser et vous procurer JDBC (exemple de code, documentation, etc.), des liens Internet utiles.
- **Teide** : l'application de gestion de projet. Vous devrez utiliser Teide pour constituer vos équipes, déposer vos rendus (documentation du projet à déposer à chaque séance sans la valider, code source Java et SQL, et supports pour la soutenance en fin de projet).

## 4.3 Suivi

Le projet donne lieu à une réunion de suivi intermédiaire entre l'encadrant et chacune des équipes (4ème séance encadrée). Cette réunion sera provoquée par les équipes elles-mêmes ou par l'encadrant en fonction de l'avancement du travail. Les thèmes discutés lors du suivi concernent l'analyse et la modélisation Entités/Associations.

## 4.4 Livrables

Vous aurez trois livrables à fournir pour ce projet :

1. Documentation du projet (un fichier PDF) Vous devrez maintenir la documentation du projet tout au long de son déroulement.

La documentation doit comprendre :

- L'analyse du problème sous forme de contraintes (DF, de valeur, de multiplicité, contextuelles). Expliquez vos choix sur les points difficiles.
- La conception Entités/Associations notée en UML, comme vu en cours, en expliquant vos choix. Précisez bien les contraintes non représentées dans le schéma.  
Le schéma peut être réalisé avec l'outil Dia (disponible sur toutes les plateformes), n'importe quel outil de dessin vectoriel, voire même sur papier puis scanné, tant qu'il est lisible et respecte les notations graphiques vues en cours.
- La traduction en relationnel du diagramme E/A, en précisant les formes normales des relations et en justifiant vos choix. Les contraintes non implantables en relationnel devront également être listées.
- L'analyse des fonctionnalités (transactions), leur implantation sous forme de requêtes SQL, le tout bien commenté.
- Un bilan du projet (organisation, points difficiles rencontrés, etc. en prenant du recul sur le travail effectué).
- Un petit mode d'emploi de votre démonstrateur est également le bienvenu.

La qualité et la complétude de la documentation seront prises en compte dans la notation.

2. Sources Java et SQL (un fichier ZIP ou TGZ) Vous devrez rendre en fin de projet :

- Un script SQL permettant de créer votre schéma relationnel
  - Un script SQL permettant de peupler la base de données
  - L'implantation des fonctionnalités (squelette des transactions et requêtes SQL, par exemple avec un fichier SQL commenté comme vu en cours)
  - Le code source Java du démonstrateur.
3. Supports pour la soutenance Les slides prévus pour la soutenance devront être déposés sur Teide (visez 15 minutes de présentation).

## 4.5 Soutenance finale

La soutenance durera en tout 25-30 minutes divisées de la façon suivante :

- 15 minutes pour présentez votre projet et faire une démonstration bien préparée (scénario, jeux de données en place, etc.)
- 10 minutes pour que le jury puisse vous faire un retour sur le projet et vous poser des questions

L'objectif de la soutenance est double : vous devez convaincre le client que le produit répond bien à ses besoins, et vous devrez convaincre l'expert que votre équipe a fait du bon travail. La soutenance doit donc reprendre les différents points de la documentation en insistant sur les points que vous jugerez les plus pertinents.

## 5 Conclusion

Le projet se déroule en quatre semaines, soutenances incluses. Cela va aller très vite. Une bonne organisation de l'équipe est primordiale.

Il est recommandé de diviser les tâches dès le début du projet et de toujours garder des "candides" par tâche (des personnes qui ne participent pas du tout à cette tâche), de façon à ce qu'ils puissent apporter un regard critique sur ce qui a été fait lors des réunions d'équipe.