

## BASES DE DONNÉES AVANCÉES

Adeel AHMAD

Nous voulons mettre en œuvre un service en-ligne « Expédition Collisimo » à l'aide d'un système de gestion de base de données relationnelle MySQL. Ce service permet d'envoyer un colis international depuis la boîte aux lettres des clients. Le client peut affranchir en-ligne son colis et le déposer dans sa boîte aux lettres. Un agent appelé *Shipper* prend en charge le colis déposé dans la boîte aux lettres pour son expédition. Les *Shippers* changent pendant les déplacements du colis. Le but de cette application est de permettre un suivi automatisé des envois et le client est informé par mail de la prise en charge et de la livraison. L'application permet un moyen rapide de vérifier l'état de l'envoi et les résultats en ligne donnent en temps réel les progrès détaillés de l'expédition à travers du réseau de livraison. L'interface utilisateur du système serait développer en Java.

Soit le schéma suivant, décrivant la base de données du service « Expédition Collisimo ». Dans ce schéma, les clés primaires sont soulignées et les clés étrangères sont en *italique*.

**Client** (numéro, nom, prénom, email, rôle, adresse, ville, *numPays*, signature)

Exemple : (4561010001, 'Dupont', 'Lynda', 'lynda@courriel.fr' 'expéditeur', '50, rue Ferdinand Buisson', '62100 Calais', 14, null)

**Pays** (numéro, nom)

Exemple : (14, 'France')

**ClientColis** (*numClient*, *numColis*)

Exemple : (4561010001, 'JD0146003874')

**Colis** (numéro, contenus, poids, dateDépart, heureDépart, dateArrivée, heureArrivée, état)

Exemple : ('JD0146003874', 'documents QTY 02', '00.50', '23-01-2016', '17h45', '25-01-2016', '11h30', 'Shipment Picked Up')

**Progrès** (*numColis*, étape, état, emplacement, dateEvenement, heureEvenement)

Exemple : ('JD0146003874', 2, 'Processed at', 'Calais-France', '23-01-2016', '20h52')

**Facture** (numéro, *numClient*, *numColis*, montant)

Exemple : (9567034761, 4561010001, 'JD0146003874', 45.00)

**Shipper** (numéro, nom, prénom)

Exemple : (7548765, 'Dupont', 'Jean-Louis')

**ShipperColis** (*numShipper*, *numColis*)

Exemple : (7548765, 'JD0146003874')

Les types des colonnes sont les suivants :

Nom de l'attribut	Description	Remarque
numClient	NUMBER(10)	
nom	VARCHAR(25)	Renseigné
prénom	VARCHAR(25)	Renseigné
email	VARCHAR(50)	Renseigné
rôle	VARCHAR(11)	Uniquement les valeurs 'expéditeur' ou 'destinateur' Valeur par défaut 'expéditeur'
adresse	VARCHAR(250)	Renseigné
ville	VARCHAR(25)	Renseigné
numPays	NUMBER(3)	
nomPays	VARCHAR(25)	Renseigné et pas d' homonyme
signature	BLOB	
numColis	VARCHAR(12)	
contenus	VARCHAR(25)	
poids	NUMBER(2,2)	
dateDépart	DATE	Renseigné
dateArrivée	DATE	
heureDépart	TIME	Renseigné
heureArrivée	TIME	
état	VARCHAR(25)	Uniquement les valeurs 'Shipment Picked UP' ou 'Processed at' ou 'Departed Facility at' ou 'Arrived at Sort Facility' ou 'Transferred Through' ou 'With delivery courier at' ou 'Delivered'
étape	NUMBER(2)	Renseigné
emplacement	VARCHAR(50)	Renseigné
dateEvenement	DATE	Renseigné
heureEvenement	TIME	Renseigné
numFacture	NUMBER(10)	Renseigné
montant	DECIMAL(3,2)	Renseigné

---

# 1 JDBC

- 1.1 Écrivez un programme Java JDBC qui se connecte à une base de données relationnelle « Expedition-Collisimo » et créer les tables conformément au schéma de base de données décrit ci-dessus.
- 1.2 Écrire un programme Java JDBC qui se connecte à la base de données créée en Exo1 et qui vous permet d'insérer les données décrites dans les exemples ci-dessus dans la base de données « ExpeditionCollisimo ».
- 1.3 Écrire un programme Java JDBC qui permet d'afficher le pays d'origine du Colis numéro 'JD0146003874'.
- 1.4 Écrire un programme Java JDBC qui permet de savoir le progrès (étape, état, emplacement, heure) du Colis numéro 'JD0146003874' trier par les dates d'événements. Utiliser la classe `ResultSetMetaData` pour récupérer les noms et les types des colonnes.
- 1.5 Suite à Exo4, au lieu de simplement afficher les colonnes, créez une méthode générique capable d'afficher n'importe quel `ResultSet` sous forme de table formatée (comme dans une console SQL), en utilisant uniquement `ResultSetMetaData`.
- 1.6 Écrire un programme Java JDBC qui utilise un `JdbcRowSet` ou un `CachedRowSet` pour :
  - Récupérer tous les colis d'un pays spécifique.
  - Modifier l'état d'un colis directement dans le `RowSet` (mode déconnecté).
  - Synchroniser les modifications avec la base de données.
- 1.7 Écrire un programme Java JDBC qui permet de supprimer tous les colis livrés par Jean-Louis Dupont, avant la semaine dernière.
  - Assurez-vous de bien gérer le type `java.sql.Date` ou `java.time.LocalDate`. C'est souvent là que les développeurs JDBC rencontrent des difficultés avec les fuseaux horaires et les formats SQL.