

# Programmation avancée

Master 2 ISN

---

## Projet : Gestion des réservations d'une agence de voyage

---

Victor Naninck - Théo Dugauguez - Reda Mdair

Janvier 2025

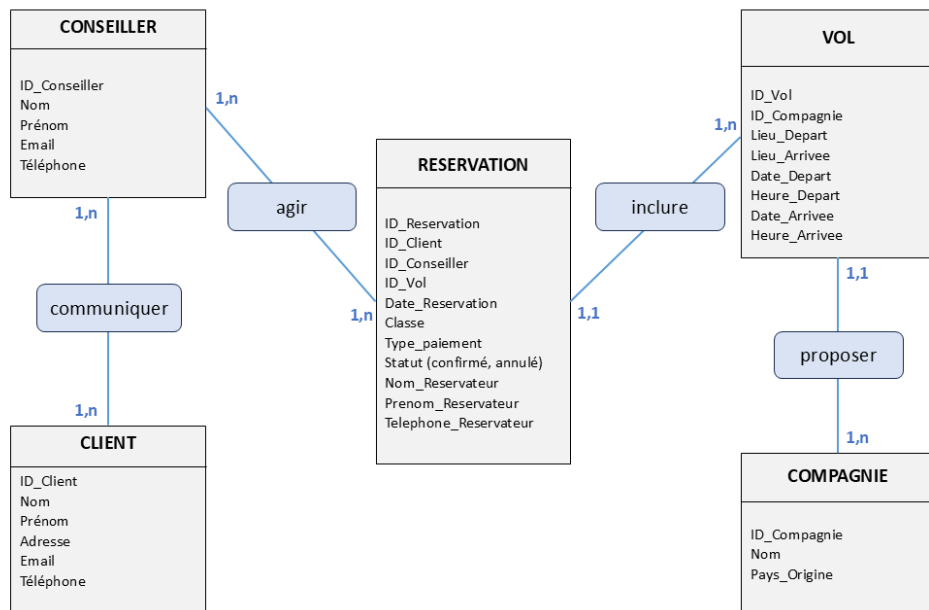
# 1. Introduction

Ce projet vise à développer une application Java pour gérer les réservations d'une agence de voyage. L'objectif est de remplacer les fichiers Excel actuellement utilisés par une solution moderne basée sur une base de données, avec une interface intuitive pour les utilisateurs.

Les informations de réservation incluent notamment le client bénéficiaire, le conseiller associé, l'itinéraire du vol, et la compagnie aérienne opérant ce dernier. C'est ainsi que nous avons défini un modèle conceptuel de données (MCD) à cinq entités :

- Client
- Conseiller
- Réservation
- Vol
- Compagnie

Les entités sont reliées entre elles par des relations clés : par exemple, une réservation est associée à un client, un conseiller, et un vol. Le modèle se trouve dans le fichier `MCD.png`, et nous l'affichons également ci-dessous.



Les réservations constituent l'unité centrale de notre projet et sont reliées aux autres entités via des clés étrangères. Actuellement, l'agence de voyage enregistre toutes les informations liées à une réservation sur une seule ligne dans un fichier Excel. Cette méthode, bien qu'intuitive, est loin d'être optimisée : risque accru d'erreurs humaines (fautes de frappe ou incohérences), manque d'automatisation (absence de relations entre les entités), etc.

Notre solution associe un identifiant unique à chaque entité, permettant de référencer efficacement les informations dans les différentes tables. Ainsi, pour une réservation donnée, il suffira d'insérer les identifiants des entités concernées pour afficher tous les attributs associés, grâce à l'intégrité des relations définies dans la base de données.

L'organisation du dossier remis suit une structure modulaire, facilitant la navigation et la compréhension des différentes parties du projet. En voici l'arborescence :

<b>docker/</b>	contient la configuration du conteneur Docker
├─ docker-compose.yml	
├─ init.sql	initialise la base de données directement depuis Docker Compose
<b>src/</b>	dossier principal des sources du projet Java
├─ module-info.java	
├─ <b>config/</b>	classe gérant la connexion à la base de données
│   ├─ DatabaseConnection.java	
│   └─ test.java	
├─ <b>models/</b>	définit les entités principales du projet
│   ├─ Client.java	
│   ├─ Conseiller.java	
│   ├─ Compagnie.java	
│   ├─ Vol.java	
│   └─ Reservation.java	
├─ <b>dao/</b>	contient les accès aux données pour interagir avec la base
│   ├─ ClientDAO.java	
│   ├─ ConseillerDAO.java	
│   ├─ CompagnieDAO.java	
│   ├─ VolDAO.java	
│   └─ ReservationDAO.java	
├─ <b>fx/</b>	classes responsables de la gestion des interfaces utilisateur
│   ├─ ClientFX.java	
│   ├─ ConseillerFX.java	
│   ├─ CompagnieFX.java	
│   ├─ VolFX.java	
│   ├─ ReservationFX.java	
│   ├─ ExcelFX.java	
│   └─ Main.java.	classe principale de l'interface graphique
├─ <b>javafx-sdk-23.0.1/</b>	bibliothèque JavaFX (windows)
├─ mysql-connector-j-9.1.0.jar	connecteur MySQL (windows)
├─ MCD.png	modèle conceptuel de données
├─ MPD.png	modèle physique de données
├─ MLD.txt	modèle linéaire de données
├─ fichier_actuel.pdf	compte rendu et guide du projet
└─ executable_javafx.jar	fichier permettant l'accès à l'interface depuis un terminal

Le fichier `docker-compose.yml` permet de configurer un conteneur MySQL pour le projet Java. Son volume est implémenté comme :

```
volumes:
  - dbdata:/var/lib/mysql
  - ./init.sql:/docker-entrypoint-initdb.d/init.sql
```

Cette configuration simplifie la mise en place d'une base MySQL initialisée avec les données nécessaires au

projet. La commande `docker-compose up -d` démarre le conteneur, avec une base de donnée pré-remplie (par l'intermédiaire du script `init.sql` situé dans le même répertoire).

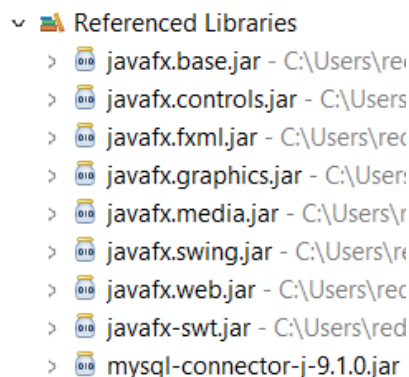
## 2. Gestion du projet via Eclipse

Le dossier `src` contient l'ensemble des fichiers source du projet, organisés en différents répertoires pour faciliter la lisibilité du code. Nous avons principalement travaillé sur ce dossier en utilisant l'IDE Eclipse, qui nous a permis de structurer efficacement le projet et de tester les fonctionnalités au fur et à mesure. Les fichiers principaux sont les suivants :

- `DatabaseConnection.java` : Responsable de l'établissement et de la gestion des connexions avec la base de données MySQL.
- `Main.java` : Le point d'entrée de l'application, qui initialise la base de données et lance l'interface principale.

### Installation de JavaFX

Concernant l'interface graphique, nous avons téléchargé la bibliothèque JavaFX, disponible dans le dossier `javafx-sdk-23.0.1`. L'installation a été suivie d'un ajout des librairies associées au SDK JavaFX ainsi que d'un connecteur MySQL (`mysql-connector-j-9.1.0.jar`), aux bibliothèques référencées sur Eclipse.



### Organisation

Les différents packages collaborent de manière fluide pour produire une application fonctionnelle et modulaire.

- Le package `models` contient les entités principales (par exemple, `Client`, `Vol`, `Reservation`), qui définissent la structure des données manipulées par l'application.
- Le package `dao` implémente les opérations CRUD (Create, Read, Update, Delete) pour chaque entité. Les classes DAO (`ClientDAO`, `VolDAO`, etc.) interagissent directement avec les instances des classes du package `models`, en utilisant leurs attributs et méthodes pour convertir les données récupérées de la base en objets Java.
- Le package `fx` regroupe les classes responsables de la gestion des interfaces utilisateur. Ces interfaces utilisent les méthodes des classes DAO pour afficher et modifier les données en temps réel.

Le flux de travail commence par une interaction utilisateur dans l'interface graphique (par exemple, ajout d'une réservation), ce qui déclenche un appel à une méthode dans le package `fx`. Cette méthode interagit avec les classes DAO, qui transmettent les requêtes à la base de données via `DatabaseConnection`. Une fois la base mise à jour, les informations sont retournées et affichées dans l'interface.

L'interface graphique simule l'espace personnel d'un conseiller de l'agence. Suite à une demande d'un client, le conseiller insère les informations du client dans la table `Client`, puis rajoute une réservation à son nom via les vols disponibles dans la table `Vol`.  
L'ouverture de l'interface se présente comme suit :

ID Client	Nom	Prénom	Adresse	Email	Téléphone		
1	DUPUIS	Elise	17 rue de la Paix, Paris	elise.dupuis@exemple.com	0611111111		
2	LEBLANC	Aurelien	22 boulevard Saint Germain, Paris	aurelien.leblanc@exemple.com	0612121212		
3	FERRARI	Isabella	45 rue des Orangers, Bordeaux	isabella.ferrari@exemple.com	0613131313		
4	MARTIN	Leo	8 avenue de la Liberté, Marseille	leo.martin@exemple.com	0614141414		
5	KIM	Yumi	12 rue des Fleurs, Lyon	yumi.kim@exemple.com	0615151515		
6	BERGER	Julian	9 rue des Roses, Toulouse	julian.berger@exemple.com	0616161616		
7	MANY	Sarah	23 avenue du Parc, Strasbourg	sarah.many@exemple.com	0617171717		
8	LEROY	Chloe	34 rue de la République, Nantes	chloe.leroy@exemple.com	0618181818		
9	MORENO	Diego	78 rue du Vieux Port, Nice	diego.moreno@exemple.com	0619191919		

Figure 1: Accueil de l'interface

Elle permet d'afficher le contenu des cinq entités du projet, ainsi qu'un fichier Excel récapitulatif de toutes les informations détaillées d'une réservation sur une seule ligne.

Trois boutons sont disponibles :

- **Ajouter** : opérationnel à tout instant.
- **Modifier** : nécessite de sélectionner une ligne au préalable.
- **Supprimer** : nécessite de sélectionner une ligne au préalable.

Remarque : pour supprimer un client, il faut d'abord supprimer toutes ses réservations enregistrées.

Les conseillers interagissent uniquement avec les informations associées aux clients et aux réservations. Certaines tables contiennent des données statiques ou définies par des composantes administratives. Les conseillers ne peuvent alors que les consulter :

- **Vol** : Informations sur les vols (lieux, horaires), définies par les compagnies aériennes.

- **Compagnie** : Détails des compagnies aériennes.
- **Conseiller** : Informations sur les conseillers, gérées par l'administration de l'agence.

Aucun bouton n'est mis à disposition pour ces tables depuis l'interface graphique. Cette distinction garantit une utilisation optimale de l'application. Les conseillers concentrent leurs efforts sur la gestion des clients et des réservations. Les données sensibles ou administratives, comme celles des vols et des compagnies, restent protégées contre les modifications accidentelles ou non autorisées.

Voici ci-dessous l'interface affichée pour une modification de réservation. Un formulaire pré-rempli apparaît, permettant de modifier les informations souhaitées.

Interface d'un conseiller pour effectuer une réservation pour un client

ID	ID Client	ID Conseiller	ID Vol	Date Réservation	Classe	Paiement	Statut	Nom
1	1	4	1	2025-01-10	economique	CB	confirme	DUPUIS
2	2	5	3	2025-01-11	business	virement	confirme	LEBLANC
3	3	1	5	2025-01-12	premiere classe	paypal	annule	FERRARI
4	4	3	7	2025-01-13	economique	especes	confirme	MARTIN
5	5	2	6	2025-01-13	business	CB	confirme	KIM

Modifier une Réservation

ID Client: 4

ID Conseiller: 3

ID Vol: 7

Date Réservation: 13/01/2025

Classe: économique

Paiement: especes

Statut: confirme

Prénom Réserveur: Leo

Téléphone Réserveur: 0614141414

Enregistrer Annuler

Figure 2: Modification d'une réservation

Un conseiller fait notamment recours à cette commande s'il veut annuler la réservation d'un client et la basculer du statut "confirmé" au statut "annulé".

Une fois l'ajout ou la modification d'une réservation faite, celle-ci se met automatiquement à jour dans le fichier Excel (sans même répéter l'exécution du projet), dont voici la structure :

ID Client	Nom Client	Prénom Client	Adresse Client	Email Client	Téléphone Client	ID Réservation	Date Réservation	Classe	Type Paiement	Statut	Nom Réserveur	Prénom Réserveur
1	DUPUIS	Elise	17 rue de la Paix, Paris	elise.dupuis@exemple.com	0611111111	1	2025-01-10	economique	CB	confirme	DUPUIS	Elise
2	LEBLANC	Aurelien	22 boulevard Saint Germain, Paris	aurelien.leblanc@exemple.com	0612121212	2	2025-01-11	business	virement	confirme	LEBLANC	Aurelien
3	FERRARI	Isabella	45 rue des Orangers, Bordeaux	isabella.ferrari@exemple.com	0613131313	3	2025-01-12	premiere classe	paypal	annule	FERRARI	Isabella
4	MARTIN	Leo	8 avenue de la Liberté, Marseille	leo.martin@exemple.com	0614141414	4	2025-01-13	economique	especes	confirme	MARTIN	Leo
5	KIM	Yumi	12 rue des Fleurs, Lyon	yumi.kim@exemple.com	0615151515	5	2025-01-13	business	CB	confirme	KIM	Yumi

Figure 3: Fichier Excel (la suite des colonnes nécessite de coulisser la barre de défilement vers la droite)

Cet onglet présente une vue globale de toutes les entités liées à une réservation (client, vol, conseiller, etc.), consolidées sur une seule ligne. Il se construit grâce à la classe `ExcelFX` qui utilise des jointures entre les différentes entités dans la base de données pour produire une vue complète.

### 3. Gestion du projet via un terminal de commande

Outre l'interface graphique, l'application permet également de gérer directement la base de données MySQL via un terminal de commande. Cette fonctionnalité offre une flexibilité supplémentaire pour les administrateurs ou les développeurs souhaitant exécuter des commandes SQL manuelles ou automatiser des tâches.

#### Accès à la base de données

La base de données est hébergée dans un conteneur Docker, ce qui simplifie son déploiement et son accès. Voici les étapes que nous avons effectué pour interagir avec elle :

1. Démarrer le conteneur Docker en exécutant la commande suivante dans le terminal :

```
docker-compose up -d
```

2. Accéder à la base MySQL via le conteneur en utilisant la commande suivante :

```
docker exec -it projet_RM_VN_TD mysql -u projet_RM_VN_TD -p
```

3. Saisir le mot de passe configuré (**motdepasse**) lorsque le terminal le demande.

#### Exécution de commandes SQL

Une fois connecté, il est possible d'exécuter des commandes SQL pour interagir avec les données :



```
Windows PowerShell
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 9.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| performance_schema |
| projet_RM_VN_TD    |
+-----+
3 rows in set (0.04 sec)

mysql> USE projet_RM_VN_TD
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_projet_RM_VN_TD |
+-----+
| Client |
| Compagnie |
| Conseiller |
| Reservation |
| Vol |
+-----+
5 rows in set (0.01 sec)
```

Par exemple, pour afficher la liste des vols via la table `Vol`, il faut insérer la commande :

```
SELECT * FROM Vol;
```

```
mysql> SELECT * FROM Vol;
```

ID_Vol	ID_Compagnie	Lieu_Depart	Lieu_Arrivee	Date_Depart	Heure_Depart	Date_Arrivee	Heure_Arrivee
1	1	Dubai	Paris	2025-01-20	14:00:00	2025-01-20	18:00:00
2	2	Doha	Londres	2025-01-21	15:30:00	2025-01-21	19:30:00
3	3	Singapour	New York	2025-01-22	23:30:00	2025-01-23	17:00:00
4	4	Londres	Los Angeles	2025-01-23	21:00:00	2025-01-24	05:30:00
5	5	Paris	Tokyo	2025-01-24	22:30:00	2025-01-26	17:00:00
6	6	Casablanca	Berne	2025-01-25	10:00:00	2025-01-25	12:55:00
7	7	Francfort	Tokyo	2025-01-26	13:00:00	2025-01-27	01:00:00
8	8	Atlanta	Toronto	2025-01-27	09:00:00	2025-01-27	10:30:00
9	9	Dublin	Bruxelles	2025-01-28	08:00:00	2025-01-28	09:15:00
10	10	Istanbul	Rome	2025-01-29	12:00:00	2025-01-29	13:30:00
11	11	Hong Kong	San Francisco	2025-01-30	16:00:00	2025-01-31	08:00:00
12	12	Tokyo	Los Angeles	2025-01-31	19:00:00	2025-01-31	11:00:00
13	13	Sydney	Auckland	2025-02-01	09:00:00	2025-02-01	10:30:00
14	14	Kuala Lumpur	Bali	2025-02-02	11:00:00	2025-02-02	12:00:00
15	15	Amsterdam	Bordeaux	2025-02-03	16:30:00	2025-02-03	18:30:00
16	16	Madrid	Lisbonne	2025-02-04	10:00:00	2025-02-04	11:00:00
17	17	Vienne	Zurich	2025-02-05	14:00:00	2025-02-05	15:00:00
18	18	New Delhi	Mumbai	2025-02-06	09:00:00	2025-02-06	10:30:00
19	19	Johannesburg	Le Cap	2025-02-07	11:00:00	2025-02-07	12:00:00

19 rows in set (0.00 sec)

Cette dernière se génère notamment à partir de clés étrangères pointant vers la table `Compagnie`, dont voici la structure grâce à la commande :

```
SELECT * FROM Compagnie;
```

```
mysql> SELECT * FROM Compagnie;
```

ID_Compagnie	Nom	Pays_Origine
1	Emirates	Emirats Arabes Unis
2	Qatar Airways	Qatar
3	Singapore Airlines	Singapour
4	British Airways	Royaume-Uni
5	Air France	France
6	Royal Air Maroc	Maroc
7	Lufthansa	Allemagne
8	Delta Airlines	États-Unis
9	Ryanair	Irlande
10	Turkish Airlines	Turquie
11	Cathay Pacific	Hong Kong
12	All Nippon Airways	Japon
13	Qantas	Australie
14	Malaysia Airlines	Malaisie
15	KLM	Pays-Bas
16	Iberia	Espagne
17	Austrian Airlines	Autriche
18	IndiGo	Inde
19	South African Airways	Afrique du Sud

19 rows in set (0.01 sec)

```
mysql>
```

## Accès à l'interface graphique

L'interface JavaFx du projet peut également être accessible depuis un terminal de commande. Nous avons pour cela exporté le projet depuis l'IDE Eclipse sous l'option `Runnable JAR File` et en cochant la gestion de la bibliothèque `Package required libraries into generated JAR`. Cela garantit que toutes les



dépendances, y compris JavaFX et les connecteurs MySQL, sont incluses dans le fichier JAR. Par la suite, le fichier généré nommé `executable_javafx` est lancé via la commande :

```
java -p "javafx-sdk-23.0.1/lib" --add-modules  
javafx.controls,javafx.base,javafx.fxml,javafx.graphics,javafx.media,javafx.web  
--add-opens=javafx.graphics/javafx.scene=ALL-UNNAMED --add-exports  
javafx.base/com.sun.javafx.event=ALL-UNNAMED -jar executable_javafx.jar
```

## Particularités du terminal de commande

L'accès via le terminal offre une grande flexibilité pour effectuer des opérations avancées qui ne sont pas prises en charge par l'interface graphique. Cependant, un usage inadéquat peut entraîner des erreurs ou des pertes de données si des commandes SQL mal formées sont exécutées.

Cet accès est accordé exclusivement à l'administration et non aux conseillers, qui eux peuvent uniquement consulter l'interface graphique. Cette méthode est essentielle pour diagnostiquer et résoudre rapidement des problèmes liés aux données, ou pour tester des modifications de la structure de la base de données.

De plus, toutes les tâches liées aux privilèges et aux droits d'accès à la base de données, telles que la création de nouveaux utilisateurs, la gestion des permissions, la gestion des tables non modifiables depuis l'interface doivent être effectuées exclusivement depuis le terminal. Cela garantit un contrôle sécurisé et centralisé des autorisations, essentiel pour la robustesse du système.