

Rapport : Système de gestion de billets pour une compagnie aérienne

January 16, 2025

Nom du groupe : GAIDT_SI_3

Membres du groupe :

- BOUDRIKA ILIAS
- EL MAJDI WALID
- DIALLO SOULEYMANE
- EL KHADER SAAD
- DJERI-ALASSANI OUBENOUPOU

Contents

0.1	Introduction	2
0.2	Objectifs	2
0.3	Modèle Conceptuel des Données (MCD)	2
0.3.1	Entités principales	2
0.3.2	Relations entre les entités	2
0.3.3	Diagramme du MCD	3
0.4	Schéma relationnel	3
0.4.1	Tables principales	3
0.5	Tarification dynamique	4
0.6	Exemple d'implémentation SQL	4
0.7	Conclusion	5

0.1 Introduction

Ce projet vise à concevoir une base de données complète pour un système de gestion des billets et des réservations pour une compagnie aérienne. Ce système doit intégrer les fonctionnalités liées aux avions, aux sièges, aux classes, aux prix dynamiques, aux vols et aux réservations des passagers. L'objectif est de répondre aux exigences opérationnelles tout en garantissant une gestion efficace des données.

0.2 Objectifs

- Gérer les informations liées aux avions, y compris les modèles, les capacités et les classes de sièges.
- Implémenter une structure pour les vols, incluant les informations sur les départs, les destinations et les dates.
- Mettre en œuvre une tarification dynamique en fonction de la période et de la proximité de la date de voyage.
- Faciliter les réservations et la gestion des billets pour les passagers.
- Assurer la flexibilité et l'évolutivité du système pour répondre aux besoins futurs.

0.3 Modèle Conceptuel des Données (MCD)

Le Modèle Conceptuel des Données (MCD) permet de représenter les entités, leurs attributs et les relations entre elles. Voici une représentation de notre MCD.

0.3.1 Entités principales

- **Aircraft** : Représente les avions avec leurs caractéristiques (*AircraftID*, *Model*, *Capacity*).
- **Seat** : Représente les sièges (*SeatID*, *SeatNum*, *AircraftID*).
- **Class** : Définit les classes disponibles (*ClassID*, *NomClass*, *FacteurPrix*).
- **Flight** : Représente les vols (*FlightID*, *Departure*, *Destination*, *TravelDate*, *AircraftID*).
- **Passenger** : Stocke les informations des passagers (*PassengerID*, *Name*, *Tel*, *Address*).
- **Ticket** : Gère les réservations et leurs statuts (*TicketID*, *Price*, *ReservationStatus*).
- **DynamicPricing** : Gère les règles de tarification (*PricingRuleID*, *RuleDescription*, *Season*, *DaysBeforeTravel*, *PriceMultiplier*).

0.3.2 Relations entre les entités

- Un avion (**Aircraft**) peut avoir plusieurs sièges (**Seat**), mais chaque siège appartient à un seul avion.
- Les sièges sont associés à une classe (**Class**).
- Chaque vol (**Flight**) est associé à un avion.

- Un passager (**Passenger**) peut réserver un ou plusieurs billets (**Ticket**).
- La tarification dynamique (**DynamicPricing**) est appliquée à chaque billet selon les règles définies.

0.3.3 Diagramme du MCD

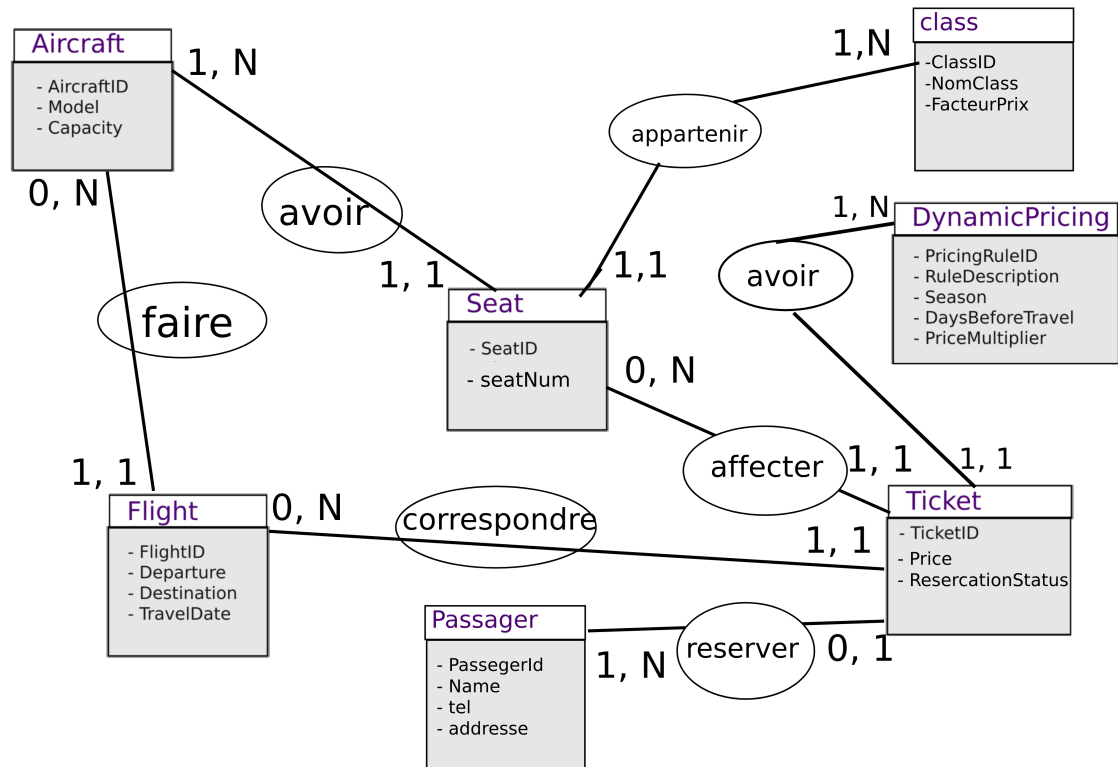


Figure 1: Modèle Conceptuel des Données (MCD) du système de gestion de billets

0.4 Schéma relationnel

Le schéma relationnel dérivé du MCD est présenté ci-dessous.

0.4.1 Tables principales

- **Aircraft** : (*AircraftID* (PK), *Model*, *Capacity*).
- **Seat** : (*SeatID* (PK), *SeatNum*, *AircraftID* (FK)).
- **Class** : (*ClassID* (PK), *NomClass*, *FacteurPrix*).
- **Flight** : (*FlightID* (PK), *Departure*, *Destination*, *TravelDate*, *AircraftID* (FK)).
- **Passenger** : (*PassengerID* (PK), *Name*, *Tel*, *Address*).
- **Ticket** : (*TicketID* (PK), *Price*, *ReservationStatus*, *PassengerID* (FK), *FlightID* (FK)).
- **DynamicPricing** : (*PricingRuleID* (PK), *RuleDescription*, *Season*, *DaysBeforeTravel*, *PriceMultiplier*).

0.5 Tarification dynamique

Le module de tarification dynamique calcule le prix du billet en fonction des règles suivantes :

- **Saison** : Les prix augmentent pendant les périodes de forte demande (exemple : vacances).
- **Proximité de la date de voyage** : Les prix augmentent à mesure que la date de départ approche.
- **Classe** : Les billets en classe Affaires coûtent plus cher en raison d'un facteur de prix plus élevé.

Formule générale :

$$\text{Prix} = \text{BasePrix} \times \text{FacteurPrix} \times \text{PriceMultiplier}$$

0.6 Exemple d'implémentation SQL

```
-- Création de la table Aircraft
CREATE TABLE Aircraft (
    AircraftID INT PRIMARY KEY,
    Model VARCHAR(50),
    Capacity INT
);

-- Création de la table Seat
CREATE TABLE Seat (
    SeatID INT PRIMARY KEY,
    SeatNum INT,
    AircraftID INT,
    FOREIGN KEY (AircraftID) REFERENCES Aircraft(AircraftID)
);

-- Création de la table Class
CREATE TABLE Class (
    ClassID INT PRIMARY KEY,
    NomClass VARCHAR(50),
    FacteurPrix FLOAT
);

-- Création de la table Flight
CREATE TABLE Flight (
    FlightID INT PRIMARY KEY,
    Departure VARCHAR(50),
    Destination VARCHAR(50),
    TravelDate DATE,
    AircraftID INT,
    FOREIGN KEY (AircraftID) REFERENCES Aircraft(AircraftID)
);

-- Création de la table Passenger
```

```

CREATE TABLE Passenger (
    PassengerID INT PRIMARY KEY,
    Name VARCHAR(50),
    Tel VARCHAR(15),
    Address VARCHAR(100)
);

-- Création de la table Ticket
CREATE TABLE Ticket (
    TicketID INT PRIMARY KEY,
    Price FLOAT,
    ReservationStatus VARCHAR(20),
    PassengerID INT,
    FlightID INT,
    FOREIGN KEY (PassengerID) REFERENCES Passenger(PassengerID),
    FOREIGN KEY (FlightID) REFERENCES Flight(FlightID)
);

-- Création de la table DynamicPricing
CREATE TABLE DynamicPricing (
    PricingRuleID INT PRIMARY KEY,
    RuleDescription VARCHAR(100),
    Season VARCHAR(20),
    DaysBeforeTravel INT,
    PriceMultiplier FLOAT
);

```

0.7 Conclusion

Le système proposé est conçu pour répondre efficacement aux besoins de gestion des billets et des réservations d'une compagnie aérienne. Il garantit une modularité, une flexibilité et une évolutivité pour les futures évolutions. La prise en charge de la tarification dynamique ajoute une valeur commerciale essentielle, maximisant les revenus en fonction de la demande et du moment de réservation.