

# Conception base de données

## Merise

### Aéroport

---

Pour les besoins de la gestion d'un aéroport on souhaite mémoriser dans une base de données les informations nécessaires à la description des faits suivants :

- Chaque avion géré est identifié par un numéro d'immatriculation.
- Il est la propriété soit d'une société, soit d'un particulier.

Dans les deux cas on doit connaître :

- Le nom, l'adresse et le numéro de téléphone du propriétaire, ainsi que la date d'achat de l'avion.

Chaque avion est d'un certain type, celui-ci étant caractérisé par son nom, le nom du constructeur, la puissance du moteur, le nombre de places.

La maintenance des avions est assurée par les mécaniciens de l'aéroport :

- Par sécurité, les interventions sont toujours effectuées par deux mécaniciens (l'un répare, l'autre vérifie).
- Pour toute intervention effectuée, on conserve l'objet de l'intervention, la date et la durée.
- Pour chaque mécanicien on connaît son nom, son adresse, son numéro de téléphone et les types d'avion sur lesquels il est habilité à intervenir.

Un certain nombre de pilotes sont enregistrés auprès de l'aéroport pour chaque pilote on connaît :

- Son nom, son adresse, son numéro de téléphone, son numéro de brevet de pilote
- Les types d'avion qu'il est habilité à piloter avec le nombre total de vols qu'il a effectué sur chacun de ces types.

Pour vous aider à établir certaines règles, voici une série de questions types auxquelles l'application doit pouvoir répondre :

- Liste des avions de la société "XXX".
- Liste des avions propriété de particuliers.
- Durée totale des interventions faites par le mécanicien Durand au mois d'août.
- Liste des avions de plus de 4 places, avec le nom du propriétaire.
- Liste des interventions (objet, date) faites sur l'avion numéro « 3242XZY78K3 ».

## Analyse : L'interview

---

- Un particulier peu il avoir plusieurs avions ?  
Oui
- Un avion peu t'il avoir plusieurs propriétaires ?  
Non
- Pour les sociétés avez-vous besoins d'information à propos de celle-ci ? si oui lesquels ?  
Non

## Règles de gestion

---

- Un avion est d'un type
- Un type concerne plusieurs avions
- Un propriétaire a un ou plusieurs avions
- Un avion est la propriété de 1 propriétaire
- Un mécanicien entretient un ou plusieurs avions
- Un avion est entretenu par un ou plusieurs mécaniciens
- Un pilote utilise un ou plusieurs types d'appareil
- Un appareil est piloté par un ou plusieurs pilotes
- Un pilote est autorisé à piloter un a plusieurs type d'avions
- Un type d'avions est autorisé a 1 ou plusieurs pilotes
- Un pilote a effectué un a plusieurs vols
- Un vol a été effectué par 2 à plusieurs pilotes

## Le dictionnaire de données

---

Entité	Mnémonique	Signification	Type (longueur)	Contraintes
Planes	Registration_Number	Numéro d'immatriculation de l'avion	Alphanumérique (11)	Identifiant, obligatoire
	Purchase_Date	Date d'achat	Date	Obligatoire
Planes_Type	Name	Nom de l'avion	Alphanumérique (10)	Facultatif
	Constructor_Name	Nom du constructeur	Alphabétique (50)	Obligatoire
	Engine_Power	Puissances des moteurs	Entier (3)	Obligatoire
	Number_Of_Place	Nombre de place	Entier (3)	Obligatoire
Owners	Owner_Name	Nom du propriétaire	Alphanumérique (50)	Identifiant, obligatoire
	Owner_Type	Type de propriétaire particulier ou société	Booléen	Obligatoire
	Owner_Address	Adresse du propriétaire	Alphanumérique (255)	Obligatoire
	Owner_Phone_Number	Numéro de téléphone	Alphanumérique (12)	Obligatoire
Mechanics	Mechanic_Lastname	Prénom du mécanicien	Alphabétique (50)	Identifiant, obligatoire
	Mechanic_Firstname	Nom du mécanicien	Alphabétique (50)	Identifiant, obligatoire
	Mechanic_City	Ville du mécanicien	Alphabétique (50)	Obligatoire
	Mechanic_Zip_Code	Code Postal du mécanicien	Entier (3)	Obligatoire
	Mechanic_Street	Nom de la rue du mécanicien	Alphabétique (100)	Obligatoire
	Mechanic_Street_Number	Numéro de la rue du mécanicien	Entier (3)	Obligatoire
	Mechanic_Phone_Number	Numéro de téléphone du mécanicien	Alphanumérique (12)	Obligatoire
Maintenance_Sheets	Maintenance_Date	Date de maintenance	Date Heure	Obligatoire, Format YYYY-MM-DD HH-II-SS
	Maintenance_Object	Objet de la maintenance	Texte	Obligatoire

## Dictionnaire de données suite :

---

<b>Pilots</b>	License_number	Numéro de brevet du pilote	Alphanumérique (13)	Identifiant, obligatoire
	Pilot_Lastname	Nom du pilote	Alphabétique (50)	Obligatoire
	Pilot_City	Ville du pilote	Alphabétique (50)	Obligatoire
	Pilot_Zip_Code	Code postal du pilote		
	Pilot_Street	Rue du pilote	Alphabétique (100)	Obligatoire
	Pilot_Street_Number	Numéro de rue du pilote	Entier (3)	Obligatoire
	Pilot_Firstname	Prénom du pilote	Alphabétique (50)	Obligatoire
	Pilot_Phone_Number	Numéro de téléphone du pilote	Alphanumérique (12)	Obligatoire
<b>Pilots_authorized_Plane</b>	Number_Of_Fly	Nombre de vol qu'a effectué un pilote	Entier (3)	Obligatoire

## Dépendances fonctionnelles

---

### Matrice des dépendances

---

Entité	Mnémonique	Registration_Number	Name	Owner_Name	Mechanic_Lastname	Mechanic_Firstname	License_number
Planes	Registration_Number	1					1
	Purchase_Date	1					1
Planes_Type	Name	1					1
	Constructor_Name		1				1
	Engine_Power		1				1
	Number_Of_Place		1				1
Owners	Owner_Name	1					
	Owner_Type_			1			
	Owner_City			1			
	Owner_Zip_Code			1			
	Owner_Street			1			
	Owner_Street_Number			1			
	Owner_Phone_Number			1			

## Matrice dépendances suite

---

Mechanics	Mechanic_Lastname	1		1			
	Mechanic_Firstname	1					
	Mechanic_Adress				1	1	
	Mechanic_Phone_Number				1	1	
	Mechanic_City				1	1	
	Mechanic_Zip_Code				1	1	
	Mechanic_Street				1	1	
	Mechanic_Street_Number				1	1	
Maintenance_Sheets	Maintenance_Date	1			1	1	
	Object	1			1	1	
Pilots	License_number						
	Pilot_Lastname						1
	Pilot_Firstname						1
	Pilot_Phone_Number						1
	Pilot_City						1
	Pilot_Zip_Code						1
	Pilot_Street						1
	Pilot_Street_Number						1

## Dépendances fonctionnelle simple

---

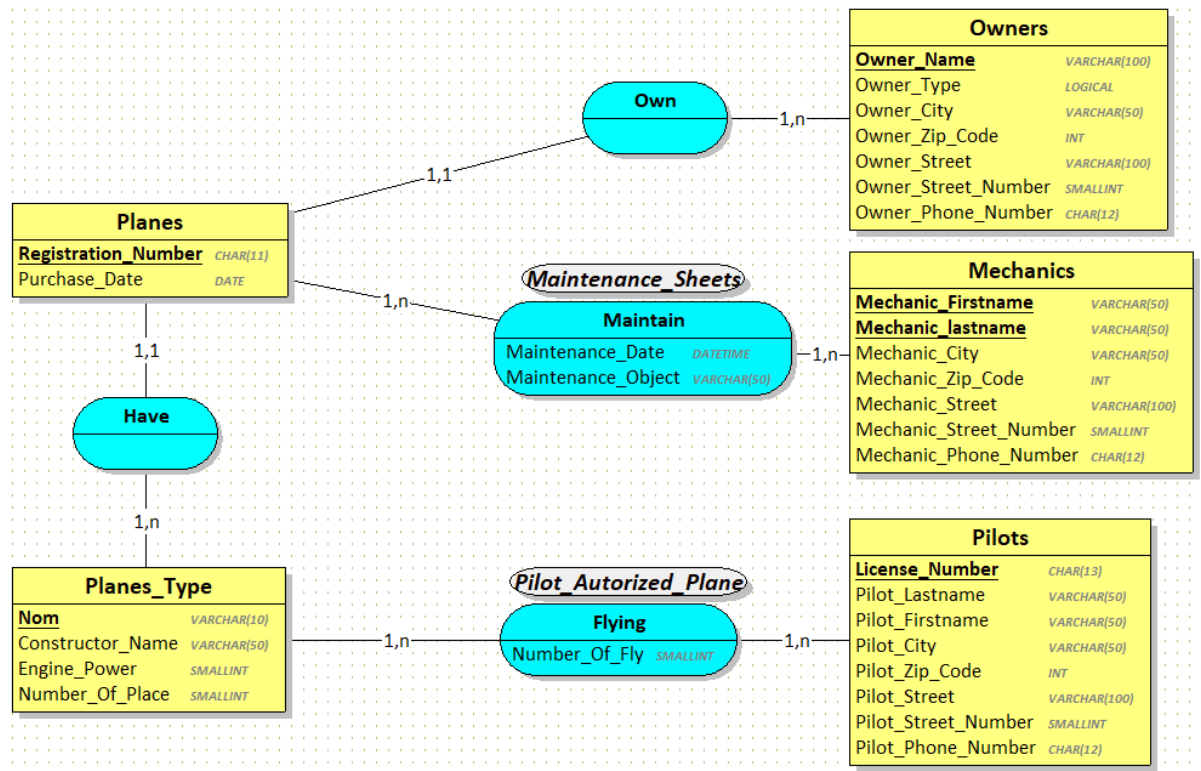
Planes(Registration_Number)	→ Purchase_Date, Planes_type(Name), Owner_Name
Planes_type(Name)	→ constructor_name, Engine_Power, Numer_Of_Place
Owner_Name Owner_Phone_Number	→ Owner_City, Owner_Zip_Code, Owner_Street, Owner_Street_Number,
Mechanics(Mechanic_Lastname, Mechanic_Firstname) Mechanic_Phone_Number, Maintenance_Date	→ Mechanic_City, Mechanic_Zip_Code, Mechanic_Street, Mechanic_Street_Number,
Maintenance_Date	→ Maintenance_Object, Planes( Registration_Number)
Pilots(License_Number)	→ Pilot_Lastname, Pilot_Firstname, Pilot_Phone_Number, Number_Of_Fly

## Dépendances fonctionnelle composées

---

Planes( Registration_Number), Owner_Name	→ Planes_type(Name)
Planes_Type(Nom), Pilot_License_Number	→ Number_Of_Fly

## Modèle conceptuel



## Modèle logique

Planes\_Type = (Nom, Constructor\_Name, Engine\_Power, Number\_Of\_Place)

Owners = (Owner\_Name, Owner\_Type, Owner\_City, Owner\_Zip\_Code, Owner\_Street, Owner\_Street\_Number, Owner\_Phone\_Number)

Planes = (Registration\_Number, Purchase\_Date, #Owner\_Name, #Nom)

Mechanics = (Mechanic\_Firstname, Mechanic\_Lastname, Mechanic\_City, Mechanic\_Zip\_Code, Mechanic\_Street, Mechanic\_Street\_Number, Mechanic\_Phone\_Number)

Maintenance\_Sheets = (#Registration\_Number, #(Mechanic\_Firstname, Mechanic\_Lastname), Maintenance\_Date, Maintenance\_Object)

Pilots = (License\_Number, Pilot\_Lastname, Pilot\_Firstname, Pilot\_City, Pilot\_Zip\_Code, Pilot\_Street, Pilot\_Street\_Number, Pilot\_Phone\_Number)

Pilot\_Autorized\_Plane = (#Nom, #License\_Number, Number\_Of\_Fly)



## Modèle physique

---

```
DROP DATABASE IF EXISTS Aeroport;
CREATE DATABASE Aeroport;
USE Aeroport;
CREATE TABLE Planes_Type
(
  Plane_Type_Name VARCHAR(10) NOT NULL,
  Constructor_Name VARCHAR(50) NOT NULL,
  Engine_Power SMALLINT UNSIGNED NOT NULL,
  Number_Of_Place SMALLINT UNSIGNED NOT NULL,
  CONSTRAINT PK_Plane_Type_Name PRIMARY KEY(Plane_Type_Name)
) ENGINE InnoDB CHARSET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE TABLE Owners
(
  Owner_Name VARCHAR(100) NOT NULL,
  Owner_Type BOOLEAN NOT NULL,
  Owner_City VARCHAR(50) NOT NULL,
  Owner_Zip_Code MEDIUMINT NOT NULL,
  Owner_Street VARCHAR(100) NOT NULL,
  Owner_Street_Number SMALLINT UNSIGNED NOT NULL,
  Owner_Phone_Number CHAR(12),
  CONSTRAINT PK_Owner_Name PRIMARY KEY(Owner_Name),
  CONSTRAINT CK_Owner_Zip_Code CHECK(Owner_Zip_Code < 96000)
) ENGINE InnoDB CHARSET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE TABLE Planes
(
  Registration_Number CHAR(11) NOT NULL,
  Purchase_Date DATE NOT NULL,
  Plane_Type_Name VARCHAR(10) NOT NULL,
  Owner_Name VARCHAR(100) NOT NULL,
  CONSTRAINT FK_Planes_Type_To_Plane FOREIGN KEY(Plane_Type_Name) REFERENCES
  Planes_Type(Plane_Type_Name),
  CONSTRAINT FK_Owners_To_Plane FOREIGN KEY(Owner_Name) REFERENCES
  Owners(Owner_Name),
  CONSTRAINT PK_Registration_Number PRIMARY KEY(Registration_Number)
) ENGINE InnoDB CHARSET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE TABLE Mechanics
(
  Mechanic_Lastname VARCHAR(50) NOT NULL,
  Mechanic_Firstname VARCHAR(50) NOT NULL,
  Mechanic_City VARCHAR(50) NOT NULL,
  Mechanic_Zip_Code MEDIUMINT NOT NULL,
  Mechanic_Street VARCHAR(100) NOT NULL,
```

```

Mechanic_Street_Number SMALLINT UNSIGNED NOT NULL,
Mechanic_Phone_Number CHAR(12),
CONSTRAINT PK_Mechanic_Lastname_Firstname PRIMARY KEY(Mechanic_Lastname,
Mechanic_Firstname),
CONSTRAINT CK_Mechanic_Zip_Code CHECK(Mechanic_Zip_Code < 96000)
) ENGINE InnoDB CHARSET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE TABLE Maintenance_Sheets
(
Registration_Number CHAR(11) NOT NULL,
Mechanic_Lastname VARCHAR(50) NOT NULL,
Mechanic_Firstname VARCHAR(50) NOT NULL,
Maintenance_Date DATETIME NOT NULL,
Maintenance_Object TEXT NOT NULL,
CONSTRAINT FK_Planes_To_Maintenance_Sheets FOREIGN KEY(Registration_Number) REFERENCES
Planes(Registration_Number),
CONSTRAINT FK_Mechanics_To_Maintenance_Sheets FOREIGN KEY(Mechanic_Lastname,
Mechanic_Firstname) REFERENCES Mechanics(Mechanic_Lastname, Mechanic_Firstname),
CONSTRAINT PK_Registration_Number_Mechanic_Lastname_Mechanic_Firstname PRIMARY
KEY(Registration_Number, Mechanic_Lastname, Mechanic_Firstname)
) ENGINE InnoDB CHARSET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE TABLE Pilots
(
License_Number CHAR(13) NOT NULL,
Pilot_Lastname VARCHAR(50) NOT NULL,
Pilot_Firstname VARCHAR(50) NOT NULL,
Pilot_City VARCHAR(50) NOT NULL,
Pilot_Zip_Code MEDIUMINT NOT NULL,
Pilot_Street VARCHAR(100) NOT NULL,
Pilote_Street_Number SMALLINT UNSIGNED NOT NULL,
Pilot_Phone_Number CHAR(12),
CONSTRAINT PK_License_Number PRIMARY KEY(License_Number),
CONSTRAINT CK_Pilot_Zip_Code CHECK(Pilot_Zip_Code < 96000)
) ENGINE InnoDB CHARSET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE TABLE Pilots_Autorized_Plane
(
Plane_Type_Name VARCHAR(10) NOT NULL,
License_Number CHAR(13) NOT NULL,
Number_Of_Fly SMALLINT UNSIGNED NOT NULL,
CONSTRAINT FK_Planes_To_Pilots_Autorized_Plane FOREIGN KEY(Plane_Type_Name) REFERENCES
Planes(Plane_Type_Name),
CONSTRAINT FK_Pilots_To_Pilots_Autorized_Plane FOREIGN KEY(License_Number) REFERENCES
Pilots(License_Number),
CONSTRAINT PK_Planes_Type_Name_License_Number PRIMARY KEY(Plane_Type_Name,
License_Number)
) ENGINE InnoDB CHARSET utf8mb4 COLLATE utf8mb4_general_ci;

```