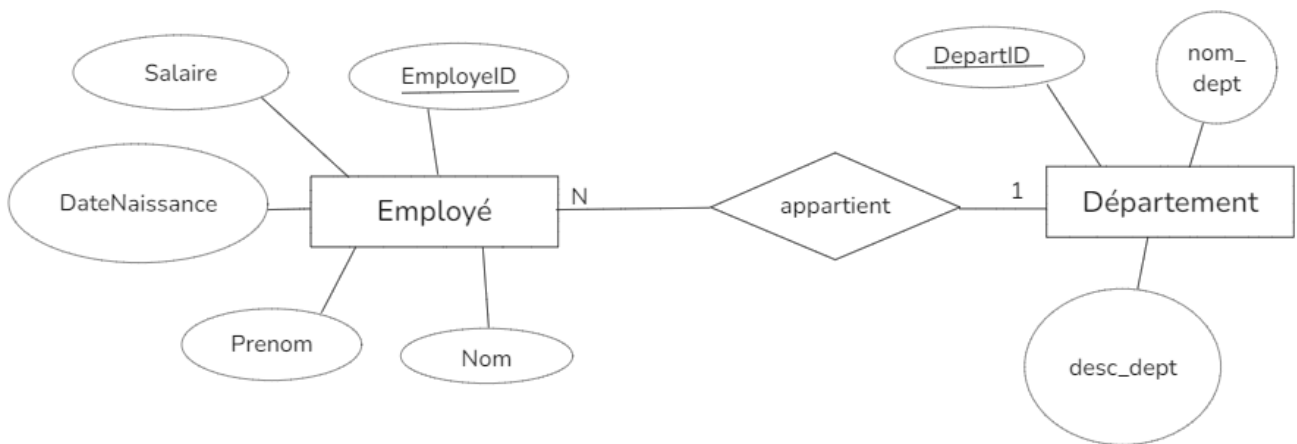


## Exercice 1

---

1. Ecrire la requête qui permet de créer la base de données "**Entreprise**", qui contient les tables décrites si dessous.
2. Ecrire la requête SQL qui permet de créer la table "**Département**" qui correspond au modèle Entité Association si dessous:



- Sachant que:
  - DepartID est un entier, qui n'est pas calculé automatiquement.
  - le nom\_dept: est le nom du département. Il peut être composé au maximum de 100 caractère, mais il ne doit pas être null. Et il doit être unique.
  - desc\_dept: est la description du département. Si aucune valeur n'est fournie, la valeur "Aucune Description" doit être affectée par défaut. La taille de champ doit être variable.

1.

```
CREATE DATABASE Entreprise;
```

2.

```
Use Entreprise;
CREATE TABLE Département (
    DepartID INT NOT NULL,
    nom_dept CHAR(100) NOT NULL UNIQUE,
    desc_dept VARCHAR(1000) DEFAULT 'Aucune Description',
    CONSTRAINT pk PRIMARY KEY (DepartID)
);
```

## Exercice 2

---

- Ecrire les requêtes SQL qui permettent de créer la table Employé décrite dans l'exercice 1, sachant que:
  - EmployeID est un entier, qui est calculé automatiquement. Valeur minimale 1, et s'incrémente automatiquement par 1
  - Nom et Prénom sont le nom et le prénom de l'employé. Ne peuvent prendre plus de 50 caractères. Ils ne doivent pas être null. Le nom de l'employé doit être unique.
  - DateNaissance: donnée obligatoire.
  - Salaire: un nombre réel qui représente le salaire de l'employé. Il ne peut être composé de plus de 10 chiffres au total, et pas plus de deux chiffres après la virgule. Sa valeur doit aussi être supérieur à 40.000

Use Entreprise;

```
CREATE TABLE Employes (
    EmployeID INT PRIMARY KEY IDENTITY(1,1),
    Nom VARCHAR(50) NOT NULL UNIQUE,
    Prénom VARCHAR(50) NOT NULL,
    DateNaissance DATE NOT NULL,
    Salaire DECIMAL(10, 2) CHECK (Salaire > 40000),
    DepartementID INT,
    FOREIGN KEY (DepartementID) REFERENCES Département(DepartId)
);
```

## Exercice 3

---

- Etant donné la table **Etudiant** créée par la requête SQL suivante:

```
CREATE TABLE Etudiant (
    EtudiantID INT PRIMARY KEY,
    Nom VARCHAR(50) ,
    Prénom VARCHAR(50),
    DateNaissance DATE,
    DateInscription DATE,
    Ville VARCHAR(100),
);
```

Ecrire les requêtes SQL qui permettent d'ajouter les contraintes suivantes:

1. Il ne peut y avoir qu'une seule ligne avec le même Nom, Prénom et DateNaissance.
2. DateNaissance doit être inférieur à 2007-01-01
3. DateInscription doit être supérieure à DateNaissance
4. Ville doit avoir une valeur par défaut: 'Mostaganem'
- 5.

```
alter table etudiant
add constraint uq_etud unique(Nom, Prénom, DateNaissance);
```

2.

```
alter table etudiant
add constraint date_naiss_val check(DateNaissance < '2007-01-01');
```

3.

```
alter table etudiant
add constraint insc_naiss check(DateInscription > DateNaissance);
```

4.

```
ALTER TABLE etudiant
add constraint vil_def DEFAULT 'Mostaganem' for Ville;
```

## Exercice 4

---

- Etant donné la table **Etudiant** de l'exercice 3, écrire les requêtes SQL qui permettent de:

1. ajouter la colonne Email , qui est de type chaîne de caractère de taille fixe (maximum 100 caractères), et ne doit pas être nulle. Sachant que pas plus de deux étudiants ne peuvent avoir le même Email.
2. supprimer la colonne DateInscription (effectuer les modifications nécessaires pour pouvoir supprimer cette colonne)
3. modifier le type de la colonne 'Ville' pour qu'elle soit de type caractère de taille 50 au maximum.
4. supprimer la contrainte sur 'Ville' ajoutée dans l'exercice 3
- 5.

```
ALTER TABLE etudiant
add Email char(100) NOT NULL UNIQUE;
```

2.

```
ALTER TABLE Etudiant
DROP CONSTRAINT insc_naiss;
```

```
ALTER TABLE Etudiant  
DROP COLUMN DateInscription;
```

3.

```
ALTER TABLE Etudiant  
ALTER COLUMN Ville varchar(50);
```

4.

```
ALTER TABLE Etudiant  
DROP CONSTRAINT vil_def;
```