

Création et utilisation de la BD :

```
CREATE DATABASE NomTable;
GO
USE NomTable;
GO
```

Mots clés	Fonction
CREATE DATABASE	Crée la BD, nécessite un nom pour la BD.
USE	Permet d'utiliser une BD.
GO	Assure qu'une instruction est complétée avant de poursuivre.

Création d'un Schéma :

Permet d'organiser/séparer les objets de la BD (comme un package).

```
CREATE SCHEMA NomSchemaUn;
GO
CREATE SCHEMA NomSchemaDeux;
GO
```

Mots clés	Fonction
CREATE SCHEMA	Crée le schéma, nécessite un nom pour le schéma

Création d'une table :

```
CREATE TABLE Personnages.Personnage(
    PersonnageID int IDENTITY (1,1) NOT NULL,
    Nom nvarchar(15) NOT NULL,
    Poids nvarchar(10) NOT NULL,
    EstMort bit NOT NULL,
    CONSTRAINT PK_Personnage_PersonnageID PRIMARY KEY (PersonnageID)
);
```

Mots clés	Fonction
CREATE TABLE	Crée une table. Si la table est créée dans un schéma, mettre le nom du schéma. Nécessite le nom de la table
NomCol	Fait référence à un attribut dans une entité.
Type	Le type de donnée de l'attribut.
Mots clés	Fonction
IDENTITY(x,y)	Auto-incrémentation, x = valeur init, y = incrément.
NOT NULL	Le champ ne peut être null.
CONSTRAINT ... PRIMARY KEY	Permet d'ajouter une contrainte nommée de clé primaire.
...	

- IDENTITY(x,y), NOT NULL, PRIMARY KEY;
- Les contraintes IDENTITY et NOT NULL sont placées après le type;
- La contrainte de clé primaire est placée en dernier.

Type de données utiles (chaînes de caractères, images, dates, numériques)

Type	Rôle
nchar(N)	Stocke chaîne de caractères avec taille fixe (1 à 8000) en mémoire . Inclut caractères Unicode.
nvarchar(N)	Stocke chaîne de caractères avec taille variable (1 à 8000) en mémoire . Inclut caractères Unicode.
varbinary(max)	Stocke images jusqu'à 2 Go.
bit	Vaut 0, 1 ou NULL. Booléens. On met 1 ou 0 au lieu de true ou false.
int	0 ± 2 147 483 647. 4 bytes.
numeric(p,s)	Nombres décimaux où p = nb de chiffres au total et s = nb de chiffres après la virgule. Ex : 123,45 -> p = 5 et s = 2
datetime	Format : 'YYYY-MM-DD HH:MI:SS' (ex : '2020-12-20 17:59:59').
date	Format : 'YYYY-MM-DD'.

Convention de nommage pour les contraintes :

IMP - Structure du nom de la contrainte: Type_Table_Colonne

- **Type:** PK (Primary Key), FK (Foreign Key), CK (Check), UC (Unique), DF (Default)
- **Table :** Nom de la table
- **Colonne :** Nom de la colonne concernée. Si c'est un groupe de colonnes, un autre nom cohérent peut être utilisé.

Contraintes de clé étrangère :

Note : Ces contraintes sont créées après la création des tables

Exemple d'ajout d'une contrainte de clé étrangère sur le champ ClientID de la table Vente qui fait référence au champ ClientID (clé primaire) de la table Client.

```
ALTER TABLE Ventes.Vente ADD CONSTRAINT FK_Vente_ClientID
FOREIGN KEY (ClientID)
REFERENCES Clients.Client(ClientID)
```

Note : Si nous voulons que les données de vente d'un client soit supprimées lors de la suppression d'un client, il faut ajouter :

```
ON DELETE CASCADE
```

IMP : Nommée bien vos contraintes, ici le type est FK.

Mots clés	Fonction
ALTER TABLE	Permet de modifier une table.
ADD CONSTRAINT	Permet d'ajouter une contrainte nommée.
FOREIGN KEY (Nom champ)	Pour indiquer le nom du champ qui est la clé étrangère.
REFERENCES	Pour indiquer à quel champ fait référence la clé étrangère.

Autres types de contraintes ajoutées après celles de clés étrangères :

Exemples de contraintes CHECK et UNIQUE (type CK et UC ici) :

```
ALTER TABLE Personnages.Personnage
ADD CONSTRAINT CK_Personnage_Poids CHECK (Poids in ('leger', 'moyen', 'lourd'))

ALTER TABLE Joueurs.Joueur ADD CONSTRAINT UC_Joueur_Pseudo UNIQUE (Pseudo)
```

Mots clés	Fonction
CHECK	Permet de vérifier qu'une donnée respecte une valeur précise.
in	La donnée du champ fait partie de ('val1', 'val2', ...).
UNIQUE	Pour garantir que chaque donnée sera unique.

Note : Après chaque contrainte (étrangères et autres) utilisez le séparateur GO.