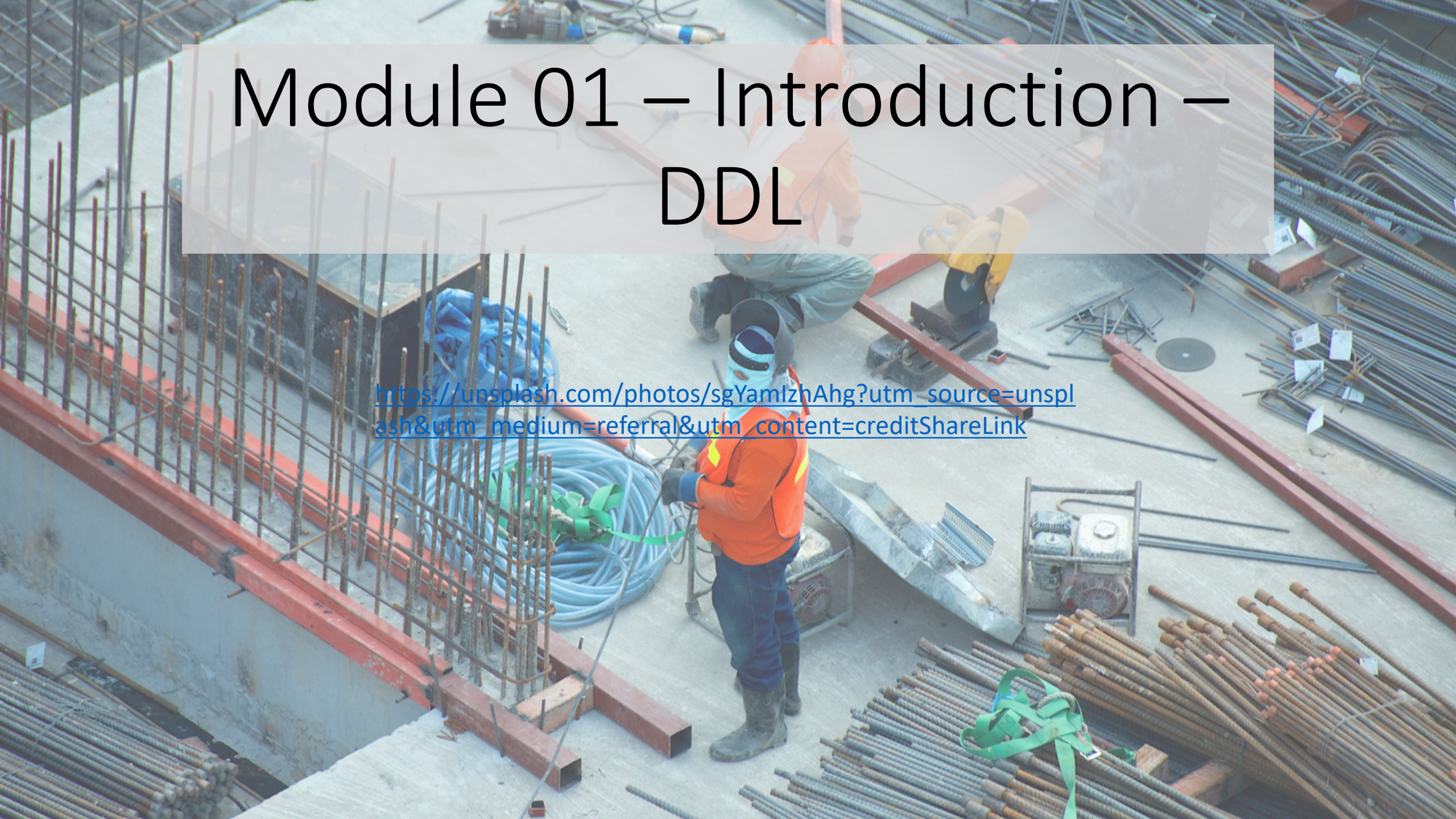
Module 01 – Introduction – DDL



https://unsplash.com/photos/sgYamIzhAhg?utm_source=unsplash&utm_medium=referral&utm_content=creditShareLink

Objectifs

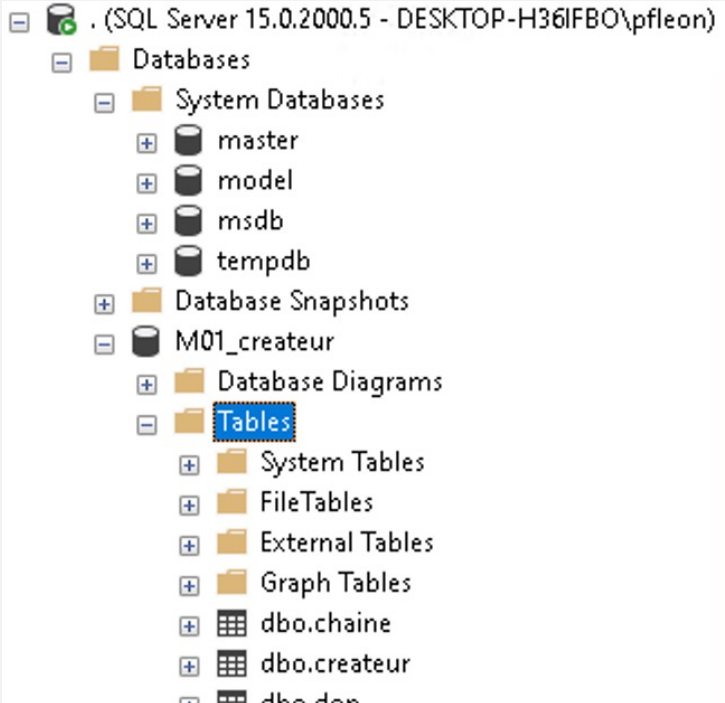
- Plan de cours
- Installation des outils
- Structure SGBD SQL Server
- Rappels – DDL
  - Création de tables
  - Types
  - Contraintes de colonnes
  - Contraintes de tables

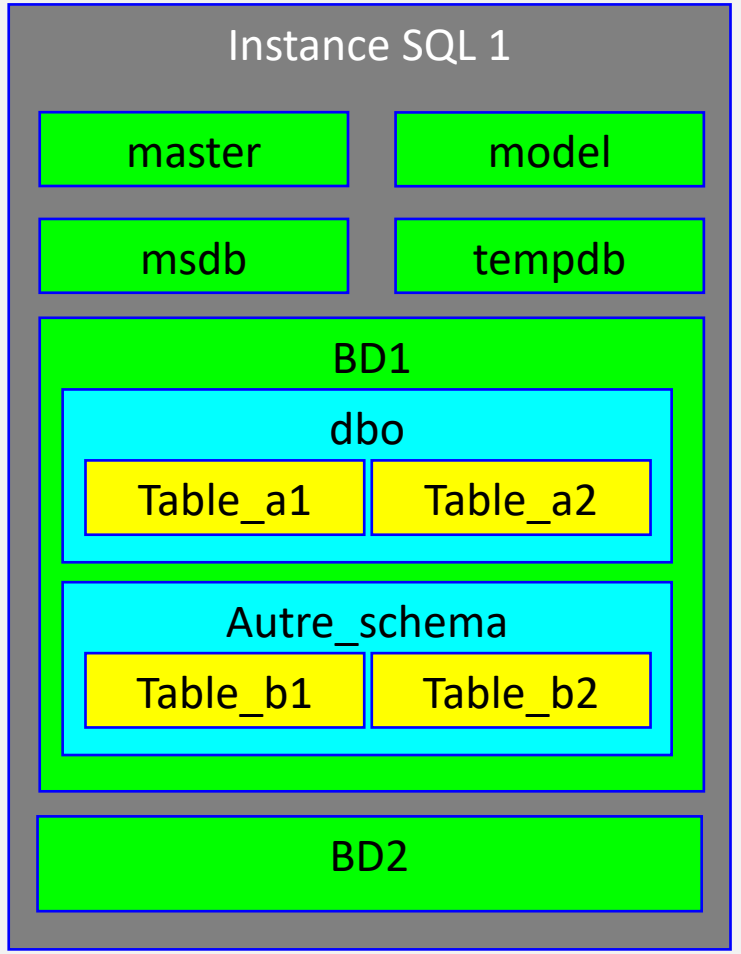
Installation des outils

- SQL Server developer : installation « basic »
  - https://www.microsoft.com/en-ca/sql-server/sql-server-downloads
- SQL Server management Studio : SSMS
  - https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16

Structure générale SQL Server

- Instance SQL 1
  - (System databases)
    - master : contient la configuration du serveur SQL et la gestion du serveur
    - model : gabarit utilisé pour créer les nouvelles bases de données
    - msdb : utilisée par SQL Server Agent
    - tempdb : contient tous les objets temporaires (ex. débutants par #)
  - BD1
    - dbo : schéma par défaut
      - Table_a1
      - Table_a2
      - …
    - autre_schema
      - Table_b1
      - Table_b2
  - BD2
    - …





Création d'une base de données

```
CREATE DATABASE database_name
[ CONTAINMENT = { NONE | PARTIAL } ]
[ ON
      [ PRIMARY ] <filespec> [ ,...n ]
      [ , <filegroup> [ ,...n ] ]
      [ LOG ON <filespec> [ ,...n ] ]
]
[ COLLATE collation_name ]
[ WITH <option> [,...n ] ]
[;]
```

https://learn.microsoft.com/en-us/sql/t-sql/statements/create-database-transact-sql?view=sql-server-ver16&tabs=sqlpool

Création d'une table

```
CREATE TABLE
    { database_name.schema_name.table_name | schema_name.table_name | table_name }
    [ AS FileTable ]
    ( {   <column_definition>
        | <computed_column_definition>
        | <column_set_definition>
        | [ <table_constraint> ] [ ,... n ]
        | [ <table_index> ] }
          [ ,... n ]
          [ PERIOD FOR SYSTEM_TIME ( system_start_time_column_name
             , system_end_time_column_name ) ]
      )
    [ ON { partition_scheme_name ( partition_column_name )
           | filegroup
           | "default" } ]
    [ TEXTIMAGE_ON { filegroup | "default" } ]
    [ FILESTREAM_ON { partition_scheme_name
           | filegroup
           | "default" } ]
    [ WITH ( <table_option> [ ,... n ] ) ]
[ ; ]
```

https://learn.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver16

Création d'une table – Colonnes

```
<column_definition> ::=
column_name <data_type>
    [ FILESTREAM ]
    [ COLLATE collation_name ]
    [ SPARSE ]
    [ MASKED WITH ( FUNCTION = 'mask_function' ) ]
    [ [ CONSTRAINT constraint_name ] DEFAULT constant_expression ]
    [ IDENTITY [ ( seed , increment ) ] ]
    [ NOT FOR REPLICATION ]
    [ GENERATED ALWAYS AS { ROW | TRANSACTION_ID | SEQUENCE_NUMBER } { START | END } [ HIDDEN ] ]
    [ [ CONSTRAINT constraint_name ] {NULL | NOT NULL} ]
    [ ROWGUIDCOL ]
    [ ENCRYPTED WITH
        ( COLUMN_ENCRYPTION_KEY = key_name ,
          ENCRYPTION_TYPE = { DETERMINISTIC | RANDOMIZED } ,
          ALGORITHM = 'AEAD_AES_256_CBC_HMAC_SHA_256'
        ) ]
    [ <column_constraint> [ ,... n ] ]
    [ <column_index> ]
```

IDENTITY : valeur unique qui s'incrémente. Souvent pour une clef primaire
(**GENERATED** : utilisé pour l'historisation des enregistrements)

https://learn.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver16

Création d'une table – Types

```
<data_type> ::=
[ type_schema_name. ] type_name
    [ ( precision [ , scale ] | max |
        [ { CONTENT | DOCUMENT } ] xml_schema_collection ) ]
```

precision : nombre de chiffres (avant/après la virgule (Max 38)
scale : nombre de chiffres après la virgule

https://learn.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver16
https://learn.microsoft.com/en-us/sql/t-sql/data-types/precision-scale-and-length-transact-sql?view=sql-server-ver16

Création d'une table – Types

Valeurs numériques exacts

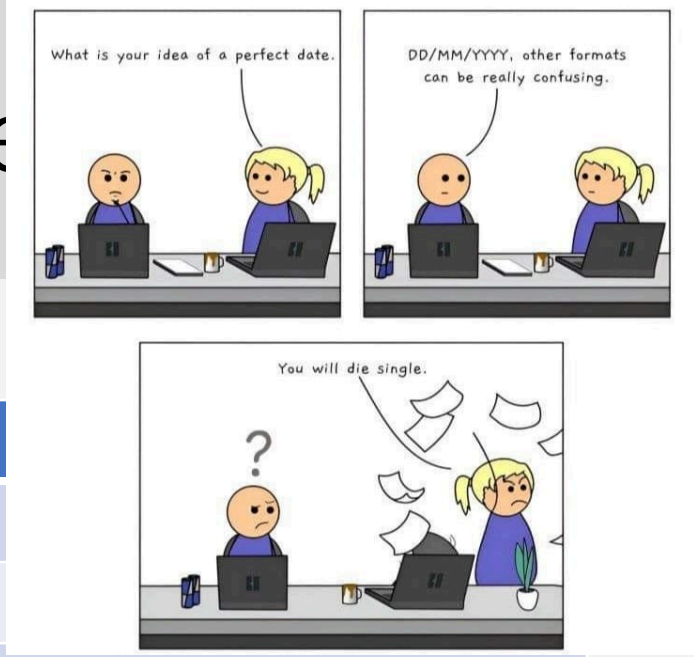
Type	Commentaire
tinyint / smallint / bigint / int	Entier non signé 1 o, entiers signés : 1 o, 2 o, 4 o, 8 o
bit	0 ou 1 ou NULL. Stockage optimisé si plus colonnes de type bit (bitwise)
decimal / numeric	*decimal* et *numeric* sont des **synonymes**. Précision de 1 à 38. 5 à 17 o.
smallmoney / money	Représente les monnaies. 4 o / 8 o. 4 décimales après la virgule

Valeurs numériques approximative

Type	Commentaire
real	4 o. équivalent à float(24)
float[(n)]	Jusqu'à 8 o. n = nombre de bits pour représenter la mantisse (de 1 à 53). L'équivalent du double est float(53).

https://learn.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver16
https://learn.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver16

Création d’une table – Type



Date et heure

Type	Commentaire
date	0001-01-01 à 9999-12-31 (3 o)
datetime	1753-01-01 à 9999-12-31 et 00:00:00 à 23:59:59.997 (8o)
datetime2[(n)]	0001-01-01 à 9999-12-31 et 00:00:00 à 23:59:59.9999999 (6 à 8 o suivant nombre chiffres n)
smalldatetime	1900-01-01 – 2079-06-06 et 00:00:00 à 23:59:59 (4 o)
time[(n)]	00:00:00.0000000 à 23:59:59.9999999 (3 à 5 o suivant nombre chiffres n)
datetimeoffset	Similaire à datetime2 mais avec le fuseau horaire par rapport à Greenwich

https://learn.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver16
https://learn.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver16

Création d'une table – Types

Caractères et chaines

Type	Commentaire
char[(n)] varchar[(n | max)]	n de 1 à 8000, max = 2^31 – 1 o
text	Fait pour stocker de grande quantité de texte 2^31-1 o (2 Go)
nchar[(n)] nvarchar[(n | max)]	Version unicode. n de 1 à 4000, max = 2^30 – 1 o
ntext	Version unicode. Fait pour stocker de grande quantité de texte 2^30-1 o (1 Go)

https://learn.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver16
https://learn.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver16

Création d'une table – Types

Autres types intéressants

Type	Commentaire
uniqueidentifier	Identifiant unique. Peut être généré par la fonction NEWID() ou NEWSEQUENTIALID(). Le second est séquentiel par rapport au dernier identifiant généré depuis le démarrage du serveur. Il permet une meilleure optimisation que les identifiants générés par NEWID qui sont eux aléatoires. Attention, NEWSEQUENTIALID est à utiliser seulement comme valeur par défaut
cursor	Type de variable seulement. Itérateur d'enregistrements (Voir cours suivants).
rowversion	Entier qui permet de stocker le numéro de version d'un enregistrement (8 o)
table	Type de variable seulement. Similaire à une table.

Conversions : https://learn.microsoft.com/en-us/sql/t-sql/data-types/data-type-conversion-database-engine?view=sql-server-ver16

Création d'une table – Contraintes

```
<column_constraint> ::=
[ CONSTRAINT constraint_name ]
{
   { PRIMARY KEY | UNIQUE }
        [ CLUSTERED | NONCLUSTERED ]
        [ ( <column_name> [ ,... n ] ) ]
        [
            WITH FILLFACTOR = fillfactor
          | WITH ( <index_option> [ ,... n ] )
        ]
        [ ON { partition_scheme_name ( partition_column_name )
            | filegroup | "default" } ]

  | [ FOREIGN KEY ]
        REFERENCES [ schema_name. ] referenced_table_name [ ( ref_column ) ]
        [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
        [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
        [ NOT FOR REPLICATION ]

  | CHECK [ NOT FOR REPLICATION ] ( logical_expression )
}
```

CLUSTERED : ordre physique des enregistrements

NONCLUSTERED : ordre logique des enregistrements

NO ACTION : le moteur lève une exception et l'action est annulée dans la table parent (ROLLBACK)

CASCADE : applique le changement dans les tables liées

SET NULL : met la colonne à NULL (Colonne doit être nullable)

SET DEFAULT : utilise la valeur par défaut

https://learn.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver16

Création d'une table – Colonnes calculées

```
<computed_column_definition> ::=
column_name AS computed_column_expression
[ PERSISTED [ NOT NULL ] ]
[
    [ CONSTRAINT constraint_name ]
    { PRIMARY KEY | UNIQUE }
        [ CLUSTERED | NONCLUSTERED ]
        [
            WITH FILLFACTOR = fillfactor
          | WITH ( <index_option> [ ,... n ] )
        ]
        [ ON { partition_scheme_name ( partition_column_name )
        | filegroup | "default" } ]

    | [ FOREIGN KEY ]
        REFERENCES referenced_table_name [ ( ref_column ) ]
        [ ON DELETE { NO ACTION | CASCADE } ]
        [ ON UPDATE { NO ACTION } ]
        [ NOT FOR REPLICATION ]

    | CHECK [ NOT FOR REPLICATION ] ( logical_expression )
]
```

https://learn.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver16

Création d'une table – Contraintes de tables

```
<table_constraint> ::=
[ CONSTRAINT constraint_name ]
{
    { PRIMARY KEY | UNIQUE }
        [ CLUSTERED | NONCLUSTERED ]
        ( column_name [ ASC | DESC ] [ ,... n ] )
        [
            WITH FILLFACTOR = fillfactor
          | WITH ( <index_option> [ ,... n ] )
        ]
        [ ON { partition_scheme_name (partition_column_name)
            | filegroup | "default" } ]
    | FOREIGN KEY
        ( column_name [ ,... n ] )
        REFERENCES referenced_table_name [ ( ref_column [ ,... n ] ) ]
        [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
        [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
        [ NOT FOR REPLICATION ]
    | CHECK [ NOT FOR REPLICATION ] ( logical_expression )
```

https://learn.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver16

Création d'une table – Index (Voir modules suivants)

```
<column_index> ::=
INDEX index_name [ CLUSTERED | NONCLUSTERED ]
    [ WITH ( <index_option> [ ,... n ] ) ]
    [ ON { partition_scheme_name ( column_name )
         | filegroup_name
         | default
         }
    ]
    [ FILESTREAM_ON { filestream_filegroup_name | partition_scheme_name | "NULL" } ]
```

https://learn.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver16

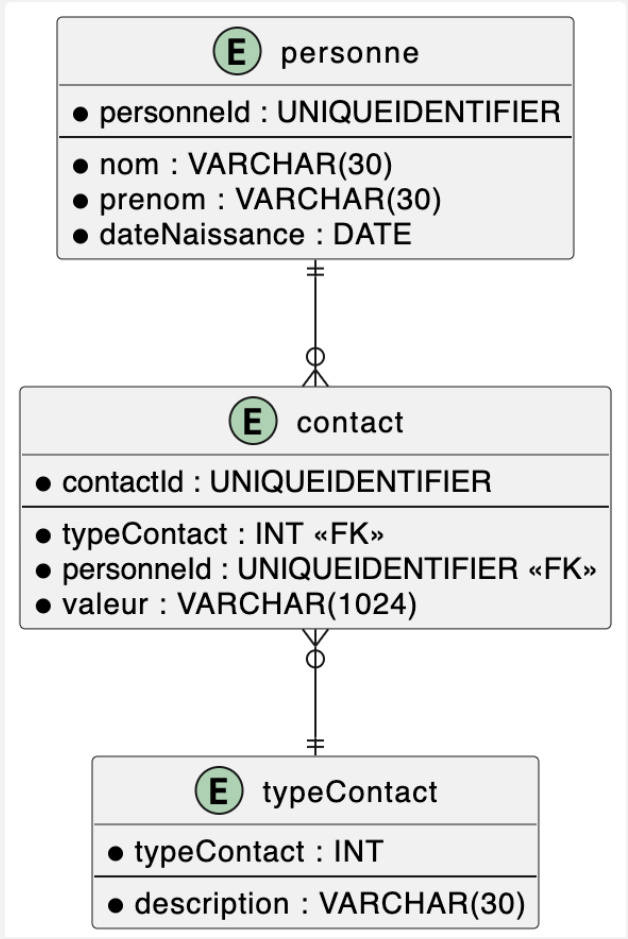
Création d'une table – Index (Voir modules suivants)

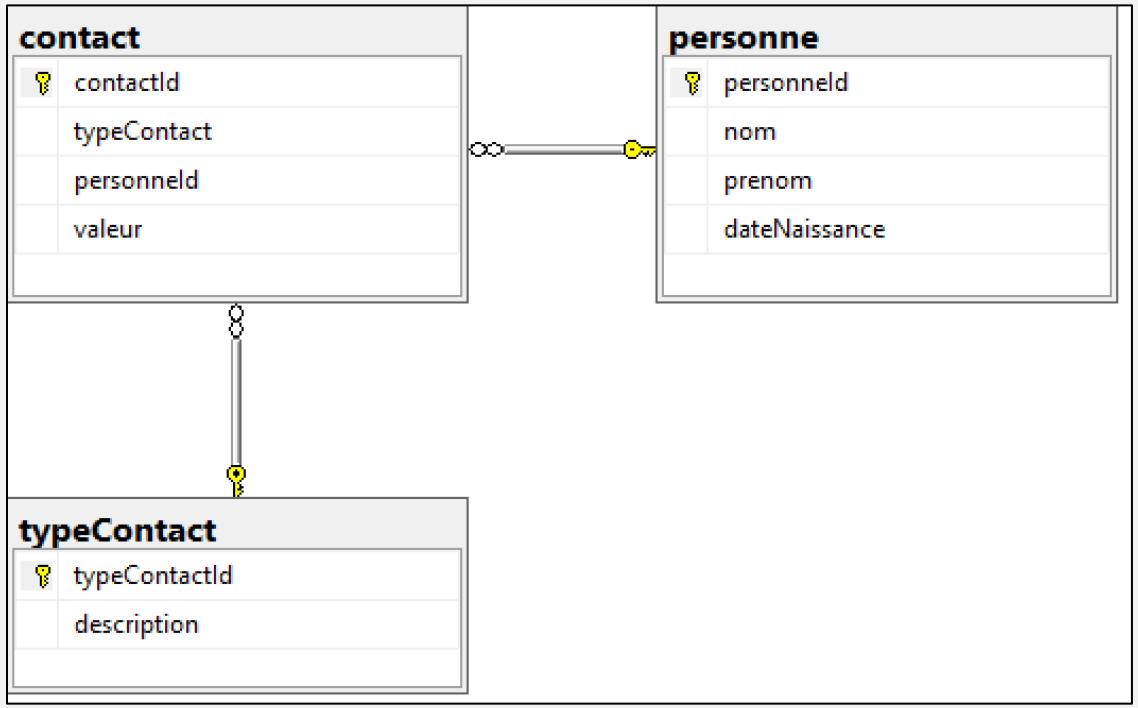
```
<table_index> ::=
{
    {
      INDEX index_name  [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ]
         ( column_name [ ASC | DESC ] [ ,... n ] )
    | INDEX index_name CLUSTERED COLUMNSTORE
    | INDEX index_name [ NONCLUSTERED ] COLUMNSTORE ( column_name [ ,... n ] )
    }
    [ WHERE <filter_predicate> ]
    [ WITH ( <index_option> [ ,... n ] ) ]
    [ ON { partition_scheme_name ( column_name )
         | filegroup_name
         | default
         }
    ]
    [ FILESTREAM_ON { filestream_filegroup_name | partition_scheme_name | "NULL" } ]

}
```

https://learn.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver16

Création d'une table – Exemple – ERD





https://learn.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver16

Création d’une table - Exemple

```
USE master;
GO

IF DB_ID ('Annuaire') IS NULL
        CREATE DATABASE Annuaire;
GO

USE Annuaire;
GO
```

Création d'une table simple - Exemple

```
CREATE TABLE personne (
    id INT IDENTITY PRIMARY KEY,
    nom VARCHAR(30) NOT NULL,
    prenom VARCHAR(30) NOT NULL,
    dateNaissance DATE NOT NULL,
    );
```

Création de tables - Exemple

```
IF NOT EXISTS (SELECT * FROM sysobjects WHERE [name]='personne' AND xtype='U')
     CREATE TABLE personne (
          personneId UNIQUEIDENTIFIER PRIMARY KEY,
          nom VARCHAR(30) NOT NULL,
          prenom VARCHAR(30) NOT NULL,
          dateNaissance DATE NOT NULL,

          INDEX IX_personne_nom NONCLUSTERED (nom),
          INDEX IX_personne_prenom NONCLUSTERED (prenom)
     );

IF NOT EXISTS (SELECT * FROM sysobjects WHERE [name]='typeContact' AND xtype='U')
     CREATE TABLE typeContact (
          typeContactId INT PRIMARY KEY,
          [description] VARCHAR(30) NOT NULL
     );

IF NOT EXISTS (SELECT * FROM sysobjects WHERE [name]='contact' AND xtype='U’)
     CREATE TABLE contact (
          contactId UNIQUEIDENTIFIER PRIMARY KEY,
          typeContact INT NOT NULL CONSTRAINT FK_contact_typeContact FOREIGN KEY REFERENCES
          typeContact(typeContactId),
          personneId UNIQUEIDENTIFIER NOT NULL,
          valeur VARCHAR(1024) NOT NULL,

          CONSTRAINT FK_contact_personneId FOREIGN KEY (personneId) REFERENCES personne(personneId)
     );
```

Convention d’écriture

- Les mots clefs, types et les noms de fonctions système sont écrit en majuscules
  - SELECT, INT, MONEY, GETDATE
- Les noms d’objets (BD, tables, colonnes) sont en camelCase
  - annuaire
- Les noms de bases de données et de tables sont au singulier
- Les contraintes sont nommées et préfixées par son type :
  - PK_<nomTable> : clef primaire
  - FK_<nomTableDuChamp><nomDuChamp> : clef étrangère
  - IX_<nomDuChamp|description> : index
  - USP_<nomProcédure> : procédure utilisateur
  - UDF_<nomFonction> : fonction utilisateur
  - SQ_<nomSequence> : séquence
  - TR_<nomTrigger> : trigger
  - CK_<nomContrainte> : nom contrainte