DICJ

Département d'informatique

Les bases de données
DDL – Data Definition Language

420-JQA-JQ

Et maintenant, on fait quoi ?

- Vous êtes maintenant des Dieux de la modélisation
- Votre modèle relationnel, comme la chenille qui se transforme en papillon, est maintenant prêt à devenir une base de données.
- Comment faire ?

À l'aide du langage SQL

- Le langage SQL (Structured Query Language) est un langage normalisé servant à exploiter des bases de données relationnelles
- Il contient des instructions de définition de données (DDL), de manipulation de données (DML) et de contrôle de données et de contrôle des transactions (DCL).
- Pour ce cours, nous nous en tiendrons au DDL et au DML.
- SQL utilise très peu de mots-clés.

Définir la structure d'une BD

- Créer une BD
- Créer une table et ses colonnes
  - Les types de données
  - Les contraintes:
    - Colonne obligatoire/facultative
    - Clés primaires
    - Numéro séquentiel généré
    - Clés étrangères
    - Unique
    - Valeurs par défaut
- Modifier une BD
- Modifier une table
- Supprimer une table
- Supprimer une BD

Créer une base de données

Syntaxe:

CREATE DATABASE BDNom;

Exemple:

CREATE DATABASE BDPubs;

Se positionner sur une base de données

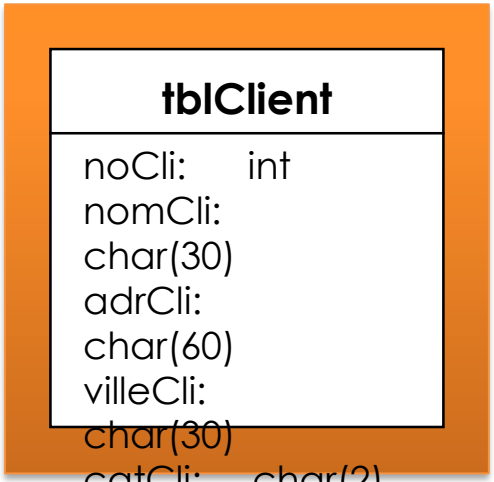
Syntaxe:

```
USE BDNom;
```

Exemple:

```
USE BDPubs;
```

Créer une table et ses colonnes




```
CREATE TABLE tblClient
(     noCli          int,
      nomCli         varchar(30),
      adrCli         varchar(60),
      villeCli       varchar(30),
      catCli         char(2),
      soldeCli       decimal(9,2),
      PRIMARY KEY (noCli)
);
```

TYPE DE DONNÉES

Les types de données – chaînes

nom	longueur
CHAR(M)	Chaîne de taille fixée à M, où 1<M<255, complétée avec des espaces si nécessaire.
CHAR(M) BINARY	Idem, mais insensible à la casse lors des tris et recherches.
VARCHAR(M)	Chaîne de taille variable, de taille maximum M, où 1<M<255.
VARCHAR(M) BINARY	Idem, mais insensible à la casse lors des tris et recherches.
TINYTEXT	Longueur maximale de 255 caractères.
TEXT	Longueur maximale de 65535 caractères.
MEDIUMTEXT	Longueur maximale de 16777215 caractères.
LONGTEXT	Longueur maximale de 4294967295 caractères.

Les typesde données - entiers

nom	borne inférieure	borne supérieure
TINYINT	-128	127
TINYINT UNSIGNED	0	255
SMALLINT	-32768	32767
SMALLINT UNSIGNED	0	65535
MEDIUMINT	-8388608	8388607
MEDIUMINT UNSIGNED	0	16777215
INT*	-2147483648	2147483647
INT* UNSIGNED	0	4294967295
BIGINT	-9223372036854775808	9223372036854775807
BIGINT UNSIGNED	0	18446744073709551615

(*) : **INTEGER** est un synonyme de **INT**

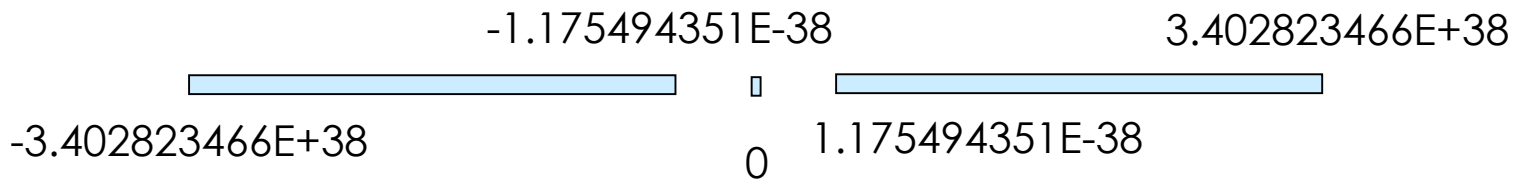
UNSIGNED permet d'avoir un type non signé

ZEROFILL : remplissage des zéros non significatifs

Les types de données – flottants

- Les flottants – dits aussi nombres réels – sont des nombres à virgule.

  Exemple du type **FLOAT** :



nom	domaine négatif : borne inférieure borne supérieure	Domaine positif : borne inférieure borne supérieure
FLOAT	-3.402823466E+38 -1.175494351E-38	1.175494351E-38 3.402823466E+38
DOUBLE*	-1.7976931348623157E+308 -2.2250738585072014E-308	2.2250738585072014E-308 1.7976931348623157E+308

(*) : **REAL** est un synonyme de **DOUBLE**.

DECIMAL(n,m) signifie des nombre de chiffre avant le point et après le point.

Exemple: DECIMAL(4,2) peut donner la valeur maximun 99,99

Les types de données – date et heure

nom	description
DATE	Date au format anglophone AAAA-MM-JJ
DATETIME	Date et heure au format anglophone AAAA-MM-JJ HH:MM:SS
TIMESTAMP	Date et l'heure sans séparateur : AAAAMMJJHHMMSS
TIMESTAMP(M)	Idem mais M vaut un entier pair entre 2 et 14. Affiche les M premiers caractères de **TIMESTAMP** sans compter les 2 premiers de l'année pour M < 12 (voir exemple en bas)
TIME	Heure au format HH:MM:SS
YEAR	Année au format AAAA

Entre TIMESTAMP(14) et TIMESTAMP(12) ce sont les 2 premiers caractères de l'année qui disparaissent; ensuite on prend les M premiers caractères de TIMESTAMP(12).

TIMESTAMP(14) AAAAMMJJHHMMSS
TIMESTAMP(12) AAMMJJHHMMSS
TIMESTAMP(6) AAMMJJ
TIMESTAMP(4) AAMM
TIMESTAMP(2) AA

Les contraintes – obligatoire/facultative

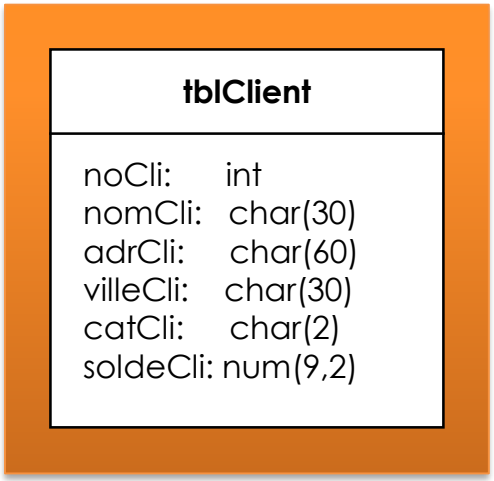
- Une colonne est **facultative** par défaut.
- Il faut donc déclarer explicitement les colonnes **obligatoires.**

```
CREATE TABLE tblClient
(   noCli           int             NOT NULL,
    nomCli          varchar(30)     NOT NULL,
    adrCli          varchar(60)     NOT NULL,
    villeCli        varchar(30),
    catCli          char(2)         NULL,
    soldeCli        decimal(9,2)    NOT NULL,
    PRIMARY KEY (noCli)
);
```

Colonnes facultatives

Colonne obligatoire

Les contraintes - Clé primaire (PK)




```
CREATE TABLE tblClient
(   noCli           int             NOT NULL,
    nomCli          varchar(30)     NOT NULL,
    adrCli          varchar(60)     NOT NULL,
    villeCli        varchar(30)     NOT NULL,
    catCli          char(2)         NULL,
    soldeCli        decimal(9,2)    NOT NULL,
    PRIMARY KEY (noCli)
);
```

Numéro séquentiel généré

- AUTO_INCREMENT permet de générer automatiquement un numéro séquentiel pour la « *clé primaire* ».

```
CREATE TABLE tblPersonne
(  noPers      smallint unsigned AUTO_INCREMENT,
   nomPers     varchar(40),
   prenPers    varchar(40),
   adrPers     tinytext,
   telPers     char(12),
   PRIMARY KEY(noPers)
);
```

PERSONNE

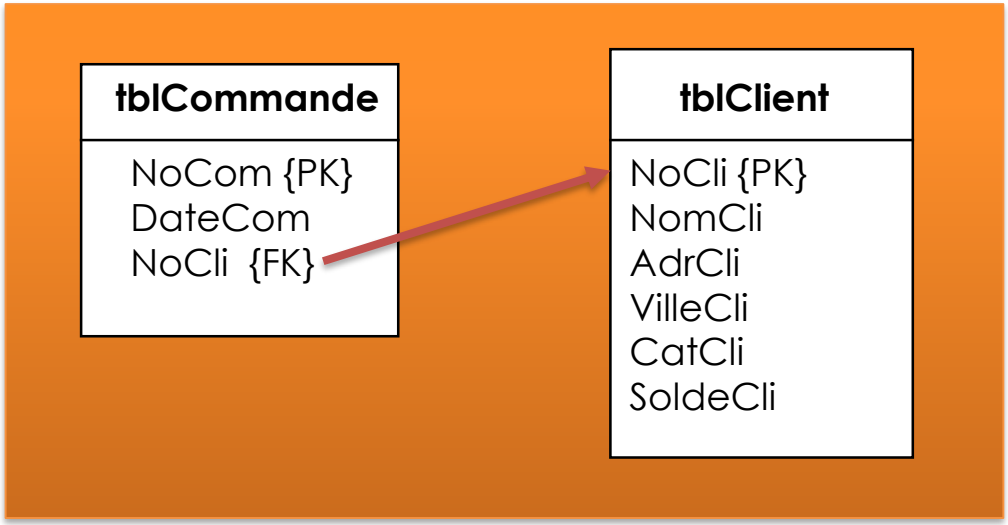
NoPers	NomPers	PrenPers	AdrPers	TelPers
1	Dupond	Marc	8 rue du Pont	418-544-5454
2	Dupont	Pierre	14 boul Dupond	418-555-4444

Les contraintes - Clé primaire COMBINÉE

- La clé primaire peut être composée de plusieurs champs

```
CREATE TABLE tblPersonne
(   nomPers     varchar(40),
    prenPers    varchar(40),
    adrPers     tinytext,
    telPers     char(12),
    PRIMARY KEY(nomPers, prenPers)
);
```

Les contraintes - Clé étrangère (FK)




```
CREATE TABLE tblCommande
(   noCom       char(12)  NOT NULL,
    noCli       char(10)  NOT NULL,
    dateCom     date      NOT NULL,
    PRIMARY KEY (noCom),
    FOREIGN KEY (noCli) REFERENCES tblClient(noCli)
);
```

Les contraintes - UNIQUE

- Pour interdire l'apparition de doublons pour un champ, on utilise la contrainte UNIQUE.

UNIQUE (champ ou *liste de champs*)

UNIQUE (**NAS**)

nomPers	prenPers	NAS
Dupond	Marc	**221-445-789**
Martin	Marc	**784-555-123**

```
CREATE TABLE tblPersonne
(  noPers      smallint unsigned AUTO_INCREMENT,
   nomPers     varchar(40),
   prenPers    varchar(40),
   NAS         char(11) UNIQUE,
   PRIMARY KEY(noPers)
);
```

Les contraintes – UNIQUE combiné

- Pour interdire les doublons d'une combinaison de champs, on liste l'ensemble des champs dans une seule commande **UNIQUE**.

  Exemple: pour interdire tout doublon du couple *nom*, *prénom*:

  UNIQUE(nomPers,prenPers)

nomPers	prenPers
Dupond	Marc
Dupont	Pierre
Martin	**Marc**
Martin	Pierre
Martin	**Marc**

Enregistrement interdit car le couple ('Martin', 'Marc') existe déjà

Contrainte Valeur par défaut

- La contrainte DEFAULT permet de déterminer la valeur qui sera assignée à la colonne si on ne donne pas de valeur spécifique
- Les valeurs par défaut sont généralement déclarées directement sur la ligne de définition du champ

```
CREATE TABLE tblClient
(  noClient     char(10)     NOT NULL,
   nomClient    varchar(30)  NOT NULL,
   adrClient    varchar(60)  NOT NULL,
   villeClient varchar(30)  NOT NULL DEFAULT 'Québec',
   catClient    char(2),
   soldeClient decimal(9,2)NOT NULL DEFAULT 0.0,
   PRIMARY KEY (noCli),
   UNIQUE (nomCli)
);
```

Un exemple

- Il est possible de mettre la contrainte sur la ligne de déclaration de la colonne, mais attention aux clés combinées.

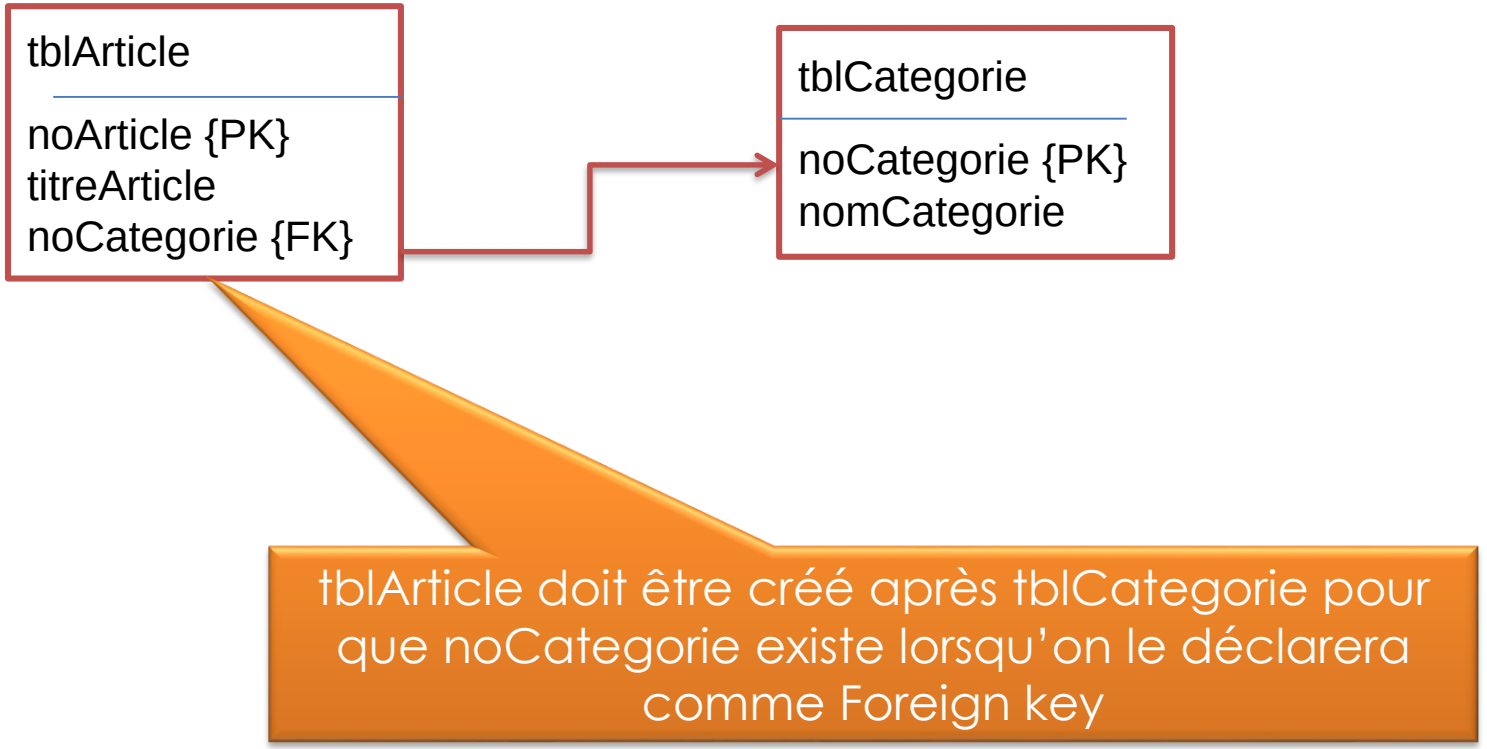
```
CREATE TABLE tblCompagnie
(  noCie         int                NOT NULL PRIMARY KEY AUTO_INCREMENT,
   nomCie        varchar(30)        NOT NULL UNIQUE,
   adrCie        varchar(60)        NOT NULL,
   villeCie      varchar(30)        NULL DEFAULT 'Québec',
   catCie        char(2),
   valeurBourse decimal(9,2)        NOT NULL
);

CREATE TABLE tblCompagnie
(  noCie      int                   NOT NULL AUTO_INCREMENT,
   nomCie     varchar(30)           NOT NULL,
   adrCie     varchar(60)           NOT NULL,
   villeCie   varchar(30)           NULL DEFAULT 'Québec',
   catCie     char(2),
   valeurCie decimal(9,2)           NOT NULL,
   PRIMARY KEY (noCie),
   UNIQUE (nomCie)
);
```

SAUF POUR LES CLÉS COMBINÉES

- Le champ déclaré UNIQUE peut accepter ou non les valeurs nulles.

Ordre de création

- L'ordre de création des tables est important.
- On ne peut référencer un champ qui n'est pas encore existant.



Supprimer une table

Syntaxe:

```
DROP TABLE NomTable;
```

Exemples:

```
DROP TABLE tblCommande;
DROP TABLE tblPersonne;
```

Attention, opération sous haute surveillance !

La table ne doit plus être référencée par une clé étrangère

Supprimer une base de données

Syntaxe:

```
DROP DATABASE NomBD;
```

Exemples:

```
DROP DATABASE BDPubs;

DROP DATABASE BDInventaire;
```

Attention ! **Tout** sera perdu!

S'assurer d'avoir un script de création

Modifier la structure d'une table

- Il est possible de modifier la structure et les contraintes d'une table déjà créée par la commande **ALTER TABLE**.
- Voici ce qu'il est possible de réaliser :
  - ajouter/supprimer une colonne
  - créer/supprimer une clé primaire
  - créer/supprimer une clé étrangère
  - changer la valeur par défaut d'un champ
  - ajouter une contrainte d'unicité (interdire les doublons)
  - modifier un champ (type de données, longueur, contraintes)
  - changer totalement la définition d'un champ

Ajouter un champ

- Il est possible d'ajouter une colonne à une table après sa création.

Syntaxe:

ALTER TABLE tblNomTable **ADD** NomColonne *typededonnees*

- Exemple : ajout d'une colonne FAX qui est une chaîne de 10 caractères

ALTER TABLE tblPersonne **ADD** FaxPers char(10)

Supprimer un champ

- Attention, supprimer un attribut implique la suppression des valeurs qui se trouvent dans la colonne qui correspond à cet attribut.

Syntaxe:

ALTER TABLE tblNomTable **DROP** NomChamp

Exemple : enlevez la colonne FAX

ALTER TABLE tblPersonne **DROP** FaxPers

Ajouter/Supprimer une clé primaire

- Il est possible d'ajouter une contrainte clé primaire à une colonne existante

```
ALTER TABLE tblNomTable ADD PRIMARY KEY (NomColonne,...)

Exemple:
ALTER TABLE tblPersonne ADD PRIMARY KEY (NoPers)
```

- Il est aussi possible de retirer la contrainte de clé primaire. Cette commande n'est pas si fréquemment utilisée et n'est pas aussi simple qu'elle peut le paraître car on doit respecter différentes règles quant au champ auto-incrémentée, aux foreign key, etc…

Ajouter une clé étrangère

- Il est possible d'ajouter une contrainte clé étrangère à une colonne existante

```
ALTER TABLE tblNomTable
ADD FOREIGN KEY (NomColonne) REFERENCES tblNomTable(NomColonne)

Exemple:
ALTER TABLE tblPersonne
ADD FOREIGN KEY (NoEmploi) REFERENCES tblEmploi(NoEmploi)
```

- Cette commande n'ajoute pas le champ NoEmploi à la table tblPersonne, elle ne fait qu'ajouter la contrainte de clé étrangère. Le champ NoEmploi doit déjà exister dans la table tblPersonne et dans la table tblEmploi.

Ajouter/Supprimer une valeur par défaut

- Ajouter une valeur par défaut

ALTER TABLE tblNomTable **ALTER** NomColonne **SET DEFAULT** *valeur*

Exemple:

ALTER TABLE tblPersonne **ALTER** provPers **SET DEFAULT** 'Québec'

- Supprimer une valeur par défaut

ALTER TABLE tblNomTable **ALTER** NomColonne **DROP DEFAULT**

Exemple:

ALTER TABLE tblPersonne **ALTER** provPers **DROP DEFAULT**

Ajouter/Supprimer une contrainte d'unicité

- Ajouter une contrainte d'unicité

```
ALTER TABLE tblNomTable ADD UNIQUE (NomColonne)

Exemple:
ALTER TABLE tblPersonne ADD UNIQUE (NomPers)
```

- Puisqu'une contrainte UNIQUE est d'abord un index, pour la supprimer il faut supprimer l'index en spécifiant son nom.

```
ALTER TABLE tblNomTable DROP INDEX NomContrainte

Exemple:
ALTER TABLE tblPersonne DROP INDEX NomPers
```

- Vous pouvez voir le nom donné à l'index en cliquant sur "+index" dans l'onglet structure. Ce nom est souvent le nom de la colonne.

Modifier un type de données

- Modifier seulement le type de données

ALTER TABLE tblNomTable **MODIFY** NomColonne *nouveautype*

Exemple:

- **ALTER TABLE** tblPersonne **MODIFY** NomPers varchar(40)

ALTER TABLE tblNomTable **MODIFY** NomColonne *nouveautype contrainte*

Exemple:
ALTER TABLE tblPersonne **MODIFY** NomPers varchar(40) NOT NULL

Modifier complètement un champ

- Modifier du même coup le nom, le type et les contraintes.

```
ALTER TABLE tblNomTable CHANGE AncienNomColonne
                    NouveauNomColonne NouveauType NouvellesContraintes

Exemple:
ALTER TABLE tblPersonne
          CHANGE Prenom PrenPers varchar(40) NOT NULL
```

- La partie NouvellesContraintes est facultative, mais même si nous souhaitons changer seulement le nom, il faut absolument fournir le type même s'il est semblable.

```
ALTER TABLE tblPersonne  CHANGE Nom NomPers varchar(25)
```

Renommer une table

- Nous pouvons modifier le nom d'une table.

```
RENAME TABLE tblNomTableTO tblNouveauNom

Exemple:
RENAME TABLE Personne TO tblPersonne
```

- Le nom est automatiquement modifié dans les références des FOREIGN KEY pointant sur la table.

Réinitialiser un champ auto incrémenté

- Pour réinitialiser la valeur d'un champ auto incrémenté, il existe une commande ALTER TABLE

```
ALTER TABLE tblPersonne  AUTO_INCREMENT = 1;
```

Ajouter une contrainte CHECK

- La contrainte CHECK permet de spécifier des valeurs ou des plages de valeurs pour un champ donné.

ALTER TABLE nom de la table
ADD CHECK (nom du champ condition)

Exemple
ALTER TABLE tblPersonne
ADD CHECK (taille < 190);