BASES DE DONNÉES

COURS 1
INTRODUCTION AUX SGBD

Laboratoire Informatique Image Interaction (L3I)
Université de La Rochelle - Pôle Sciences et Technologie - Avenue Michel Crépeau - 17042 LA ROCHELLE CEDEX 1 France
Tél : +33 (0)5 46 45 82 62 – Fax : 05.46.45.82.42 – Site internet : http://l3i.univ-larochelle.fr

BASE DE DONNÉES ?

https://fr.wikipedia.org/wiki/Base_de_données

Une base de données est une ensemble de données structurées en lien avec un sujet particulier

- Base de données de vos clients et de leurs commandes
- Base de données des produits que vous vendez et de vos fournisseurs
- Base de données de vos livres
- Base de données de tous les résultats sportifs d'un discipline
- Base de données de toutes les photos sur instagram
- …

Une base de données (relationnelle) est globalement un fichier excel avec plusieurs feuilles

BD = EXCEL

Tout comme un fichier excel, une base de données (relationnelle) contient :

- Un ensemble de feuille, appelées tables ou relations. Chaque table va contenir des informations différentes, par exemple une table avec vos clients, une avec les produits, une avec les commandes, une avec les factures…
- Chaque feuille comporte différentes colonnes appelées attributs, par exemple dans la table clients on va avoir des attributs numéro de client, nom, prénom, date de naissance, adresse…
- Chaque feuille comporte plusieurs lignes appelées enregistrements qui sont les données elles-mêmes, ainsi dans la table client chaque ligne va correspondre à un client

POURQUOI PAS UN FICHIER EXCEL ALORS

Une base de données (relationnelle) va aider à la gestion de vos « feuilles excel » en permettant par exemple :

- De limiter le contenu des cellules (nombre uniquement, date uniquement, code postal sur 5 chiffres uniquement, etc.)
  - = contrainte de domaine
- De n'autoriser dans une cellule qu'une plage de valeur fixe ou définie dans une autre table, par exemple une facture ne peut être associée qu'à un client qui existe dans la table client
- D'assurer qu'il n'y a pas de case vide (sauf si c'est autorisé)
  - = contrainte d'existence
- D'assurer l'unicité d'un attribut, par exemple pour interdire qu'il y ait deux clients avec le même numéro de client
  - = contrainte d'unicité

Tout cela peut se définir très simplement et va assurer que les données sont « propres »

POURQUOI PAS UN FICHIER EXCEL ALORS

Mais une base de données (relationnelle) va surtout permettre de chercher simplement des données, un peu comme la fonction « filtre » d'excel mais pouvant agir sur des lignes, des colonnes, plusieurs feuilles à la fois, avec des calculs de moyenne, etc.

Quelques exemples de requêtes :

- Lesquel.le.s de mes client.e.s sont nés aujourd'hui (pour leur faire une offre spécifique) ?
- Quels produits sont en rupture de stock ?
- Quels sont les fournisseurs des produits qui sont en rupture de stock ?
- Quel est mon chiffre d'affaire mensuel sur les trois dernières années ?
- …

Bien entendu on ne va pas parler à un ordinateur de cette manière (pas avant quelques années du moins)

POURQUOI PAS UN FICHIER EXCEL ALORS

Et on aimerait bien que :

- Le système soit accessible à plusieurs utilisateurs simultanément (en direct ou en ligne)
- Qu'il y ait un mécanisme de sauvegarde régulier (à chaque modification ?)
- Que le système soit protégé contre les erreurs de manipulation
- Qu'on n'ait pas à se demander comment sont stockées les données en vrai
- Que toutes les vérifications précédentes (unicité, existence, etc.) soient faites automatiquement et en permanence
- …

POURQUOI RELATIONNELLE EST ENTRE PARENTHÈSES DANS LES SLIDES PRÉCÉDENTS ?

Parce qu'il y a des systèmes de gestion de base de données qui utilisent des principes complètement différents, sans table, ligne ou colonne

Mais toujours avec des mécanismes qui « protègent » les données, permettent de les modifier et de chercher ce qu'on veut.

DÉFINITIONS

Laboratoire Informatique Image Interaction (L3I)
sité de La Rochelle - Pôle Sciences et Technologie - Avenue Michel Crépeau - 17042 LA ROCHELLE CEDEX 1 Fra
Tél : +33 (0)5 46 45 82 62 – Fax : 05.46.45.82.42 – Site internet : http://l3i.univ-larochelle.fr

BASES DE DONNÉES ET SYSTÈMES DE GESTION

Base de données : large ensemble de données

- Cohérentes
- Avec aussi peu de redondance que possible
- Indépendant d'un logiciel particulier
- Accessibles par plusieurs utilisateurs à la fois

Système de Gestion de Bases de données (SGBD) :

- Intermédiaire entre les fichiers et les utilisateurs
- L'utilisateur n'a pas à savoir comment les données sont stockées, sauvegardées, interrogées ou modifiées sans risque de conflits avec d'autres utilisateurs
- Possibilité d'interagir avec les données dans un langage simple

OBJECTIFS DES SGBD

Représentation logique :

- On peut décrire les données sans se préoccuper du stockage réel sur disque
- Le SGBD fait l'interface entre représentation logique et stockage physique
- La représentation logique peut différer entre les utilisateurs

Accès multiples :

- Lire/modifier les mêmes données en même temps
  - Réservation d'un billet d'avion, virements sur un même compte
- Assurer un traitement rapide des requêtes

Cohérence, sécurité, résistance aux pannes

LES DIFFÉRENTES APPROCHES

Approche FICHIERS

- Autant de fichiers que d'applications.
- Pas d'homogénéisation de l'information
- Exemple : ensemble de fichiers excel modifiés à la main

Approche RELATIONNELLE

- Tables de données reliées entre elles par des données communes

Autres approches :

- XML Document
  - Fichiers de données liés entre eux par des données communes
- Bases de données à objets

APPROCHE RELATIONNELLE

Tables (aussi appelées relations) :

- Toutes les données d'une BD relationnelle sont rangées dans des tables
- Une table est un ensemble de données relatives à un même sujet
  - Les tables ne sont pas des structures physiques mais LOGIQUES
  - C'est-à-dire que les données d'une même table ne sont pas forcément au même endroit sur votre disque dur

Schéma d'une base de données

- On appelle schéma la description d'une base de données :
  - Ensemble des tables
  - Ensemble des propriétés et contraintes des tables

EX : SCHÉMA DE GESTION DE COMMANDES

4 tables pour gérer :

- Les produits, les fournisseurs et les commandes
- Eviter les redondances

Un même problème réel peut souvent (toujours) être représenté de plusieurs manière

On notera ici la table lignes de commandes qui va contenir les différentes lignes d'une même commande (les différentes lignes d'un panier)

PRODUITS : référence produit, désignation, prix unitaire, numéro de fournisseur

FOURNISSEURS : numéro de fournisseur, raison sociale, ville

COMMANDES : numéro de commande, date de la commande, montant

LIGNES DE COMMANDES : numéro de commande, référence du produit, quantité

EX : SCHÉMA DE GESTION COMMERCIALE

MAGASINS (no_mag, ville , nom_gerant)

CLIENTS (no_client, nom, pays, localite, CA, type)

ARTICLES (no_art, nom, poids, couleur, qte_stock, PAchat, PVente, no_four#)

LIVRAISONS (no_livr, date_liv, no_client#, no_mag#)

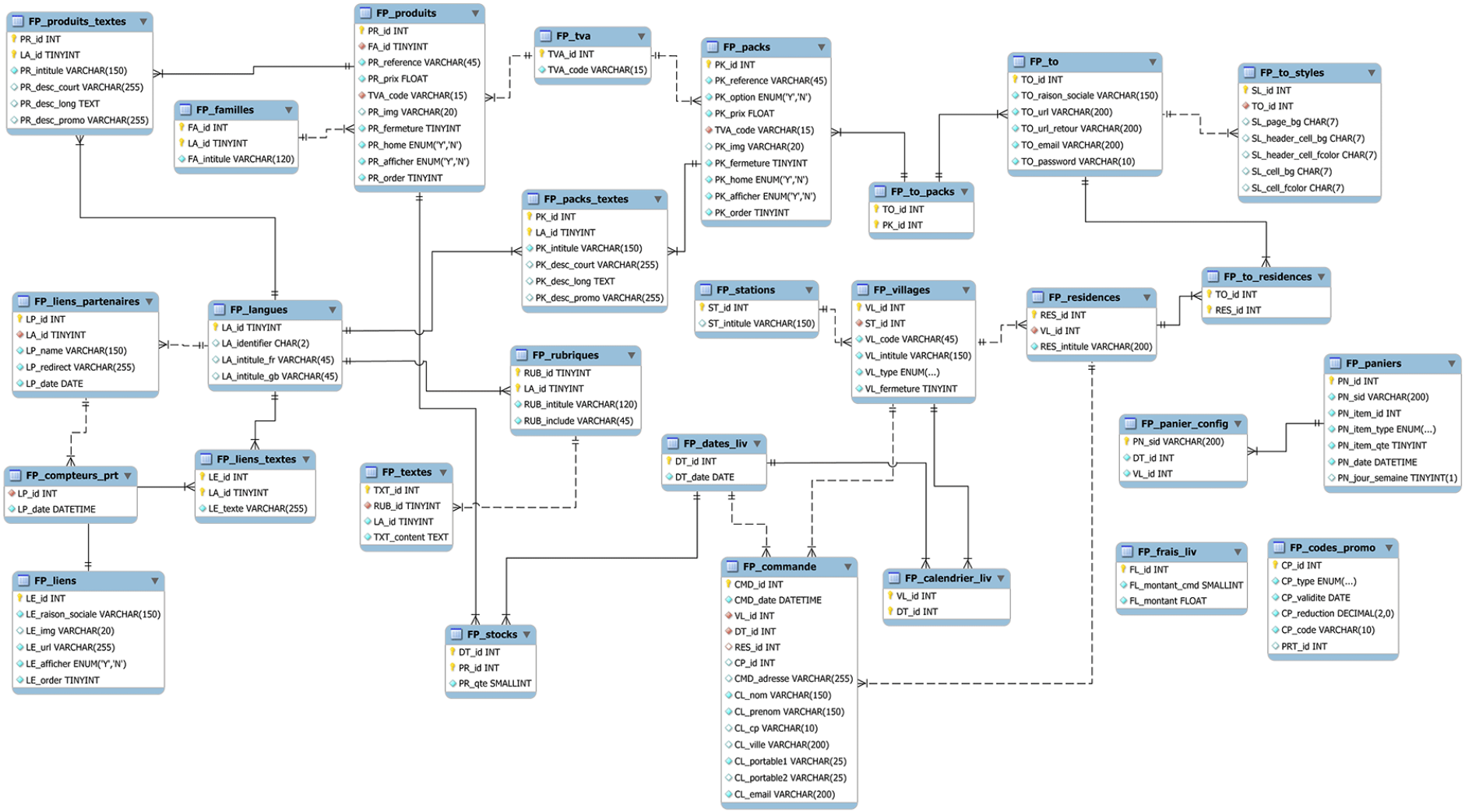
LIGNES_LIVRAISON(no_livr#,no_art#, qte, no_cde#)

FOURNISSEURS (no_four, raison_sociale)

COMMANDES (no_cde, date, no_client#, no_mag#)

LIGNES_COMMANDES (no_cde#, no_art#, qte, no_livr#, prix_total)

EX : SCHÉMA PLUS COMPLEXE



SYSTÈME DE GESTION DE BASE DE DONNÉES (SGBD)

Plusieurs SGBD existent

- MySQL
- Oracle
- SQL Server
- PostgreSQL
- Access
- …

Interfaces graphiques de gestion potentiellement très différentes

Mais le fonctionnement est similaire dans la plupart des systèmes

- On va donc plus se focaliser sur les principes que sur la présentation
- Si les principes sont acquis, passer d'un SGBD à l'autre consiste juste à apprendre à utiliser l'interface graphique…

DANS CE COURS - POSTGRESQL

PostgreSQL : SGBD

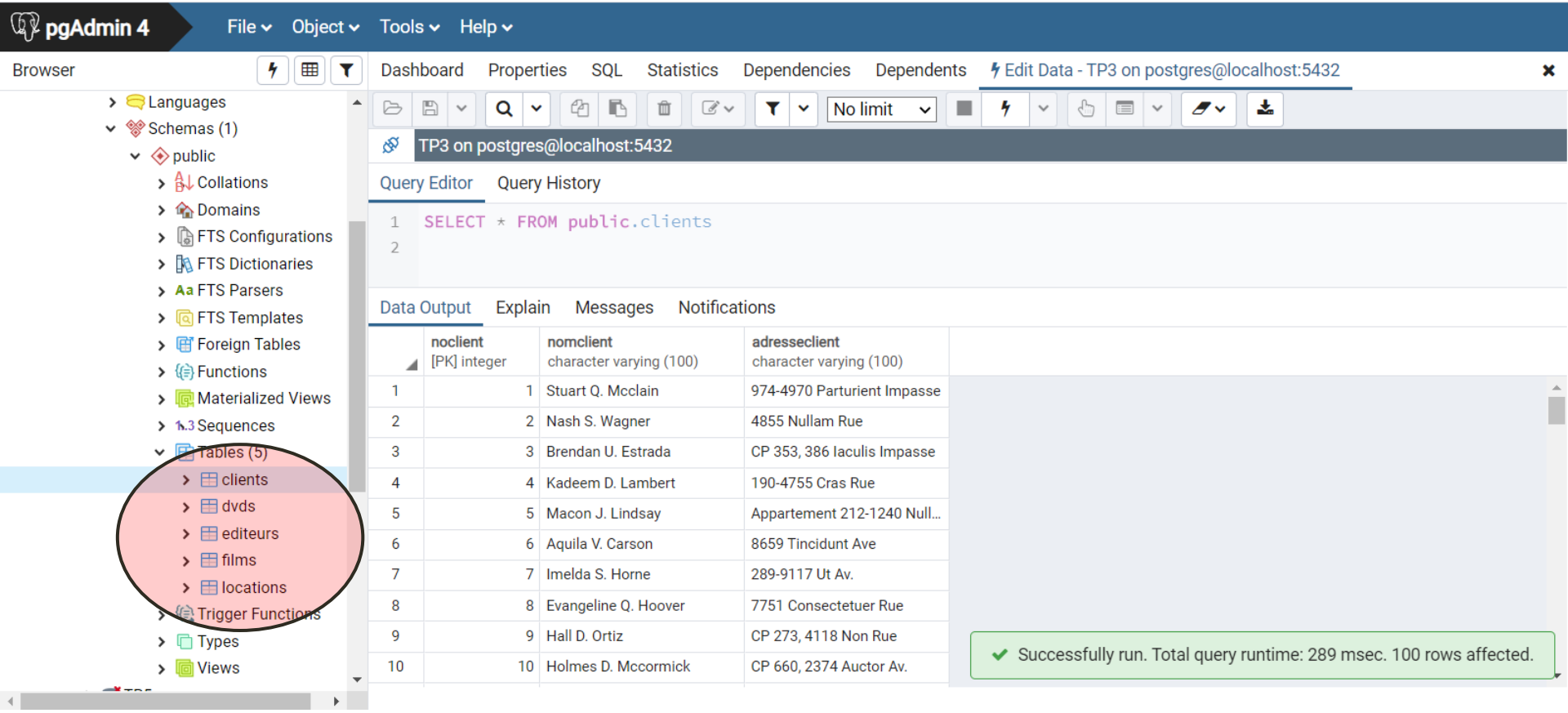
- Complètement gratuit et open source
- Puissant, robuste et fiable (plus de 20 ans de développement)
- Fonctionne sur tous les Systèmes (Linux, Windows, Mac)
- Fonctionnalités très étendues (types de données, vues, triggers, procédures stockées, inclusion de multimedia…)
- Interfaces C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC…
- Documentation très complète, en anglais et en français
- Communauté très active => forums

Pgadmin : logiciel d'interface avec PostgreSQL

- Livré avec la dernière version de PostgreSQL (facile à installer)
- Interface web (s'ouvre dans le navigateur)

PGADMIN 4

Exemple d'une base de données avec 5 tables (clients, dvds, etc.)



APPROCHE RELATIONNELLE

Chaque table a un nom unique :

- Ex : clients

Elle est constituée de colonnes (ou attributs)

- Le nom des attributs est unique dans la table
- Ex : clients (noclient, nomclient, adresseclient)

Chaque attribut est d'un type particulier (nombre, texte, date, etc.)

- Ex : noclient = nombre, nomclient = texte
- Voir plus loin pour les vrais types

Les données contenues dans la table apparaissent comme un ensemble de lignes

- Aussi appelées enregistrements, n-uplets, tuples

LES ATTRIBUTS

Chaque attribut d'une table est défini par :

- Un nom (unique dans la table)
- Un type et, le cas échéant, une longueur
- Des contraintes qui limitent l'ensemble des valeurs autorisées

Il faut faire attention à :

- Choisir des noms d'attributs qui ont du sens (plus lisible)
- Choisir les bons types (rend le système plus efficace)
- Bien limiter les valeurs autorisées pour éviter les enregistrements incorrects (mais pas trop)

TYPES DE BASE

Entiers :

- Smallint : de -32.768 à +32.767
- Integer : de -2.147.483.648 à +2.147.483.647
- Bigint : de -9.223.372.036.854.775.808 à +9.223.372.036.854.775.807

Décimaux :

- Numeric : calculs exacts mais lents. On spécifie le nombre de chiffres et de chiffres après la virgule
- Real : calculs approchés mais beaucoup plus rapides

Texte :

- varchar [(n)] : Chaîne de caractères de longueur variable (la longueur n est optionnelle)
- char [(n)] : Chaîne de caractères de longueur fixe (la longueur n est optionnelle)

Date

- timestamp : date et heure
- date : date
- time : heure seulement

EXERCICES

On veut stocker un code postal, que choisir :

- integer, varchar(5) ou char(5) ?

On veut stocker des montants de commandes, que choisir :

- numeric ou real ?

On veut stocker des noms de famille, que choisir :

- varchar(5), varchar(20), char(20), ou autre chose ?

varchar avec une contrainte d'au moins 1 caractère

EXERCICES

On veut stocker un code postal, que choisir :

- integer, varchar(5) ou char(5) ?
- Un code postal est toujours sur 5 chiffres donc varchar pas pertinent
- Le nombre 01000 n'existe pas, donc si on utilise un integer ce qui sera stocké est 1000
- Donc plutôt char(5)

On veut stocker des montants de commandes, que choisir :

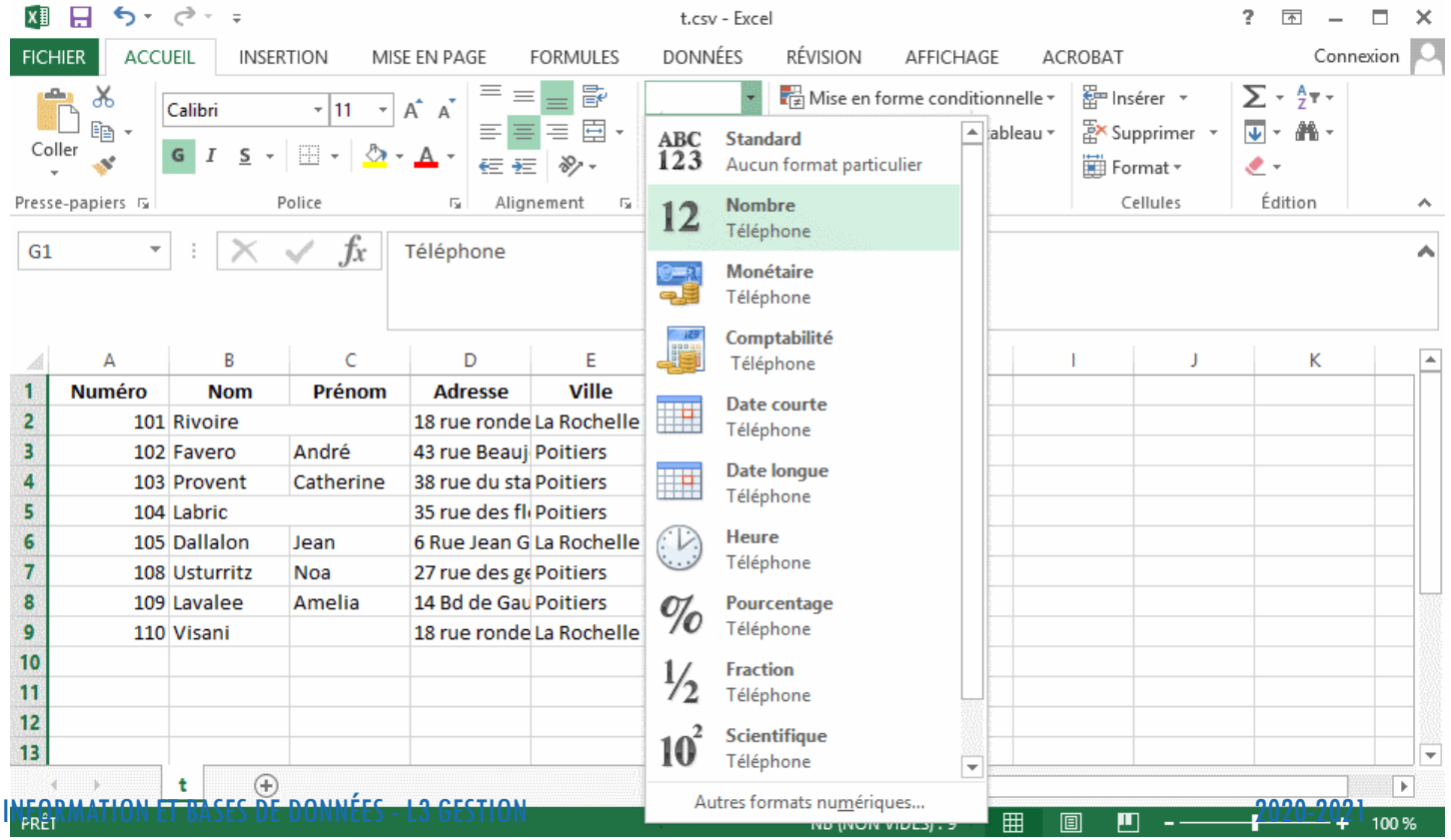
- numeric ou real ?
- real ne permet pas de faire des calculs exacts donc plutôt numeric
- Il faut de plus choisir le nombre de chiffres à stocker, par exemple numeric(9,2) ce qui permet d'avoir 9 chiffres dont 2 après la virgule, comme 1456,23 (6 chiffres donc c'est ok)

On veut stocker des noms de famille, que choisir :

- varchar(5), varchar(20), char(20), ou autre chose ?
- Est-ce que tous les noms de famille font la même longueur ? Non donc plutôt varchar
- Le nombre entre parenthèses limite la taille maximale. Est-ce que 5 lettres suffisent pour un nom de famille ? Est-ce que 20 caractères suffisent ?
- Pour éviter tout risque on peut prendre varchar(100)

EXCEL ?

Sous excel le menu format est surtout utilisé pour l'affichage mais pas pour imposer que les cellules contiennent bien une date, un nombre ou du « standard »



CONTRAINTES

On peut exprimer différentes contraintes sur les données autorisées

- Les types déjà vus

Les deux contraintes principales sont :

- Les clés primaires
- Les clés étrangères

Ces deux contraintes sont centrales pour l'utilisation des bases de données relationnelles

CLÉS PRIMAIRES

Une clé primaire est un ou plusieurs attributs qui permettent d'identifier chaque ligne de manière unique. On peut donc considérer que chaque ligne contient des attributs pour identifier la ligne et des attributs qui la décrivent

- Un client c'est un numéro de client (identifiant) et une description (nom, prénom, adresse, etc.)
- Un produit c'est un numéro de produit et une description (prix, quantité en stock, etc.)

Le SGBD va assurer qu'il n'y ait jamais deux lignes avec la même valeur de clé primaire (tous les numéros de client sont différents)

Attention :

- Un identifiant n'est pas forcément un unique attribut
- Dans ce cours toutes les tables auront une clé primaire même si dans l'absolu ce n'est pas obligatoire

```
Table PRODUIT : Référence d'un produit
Table FOURNISSEUR : Numéro du fournisseur
Table COMMANDE : Numéro de commande
Table LIGNE_COMMANDE : (Numéro de commande, Référence d'un produit)
```

CLÉ PRIMAIRE – DÉFINITION FORMELLE

Une clé primaire d'une table est un attribut ou un groupe d'attributs qui vérifie 3 propriétés

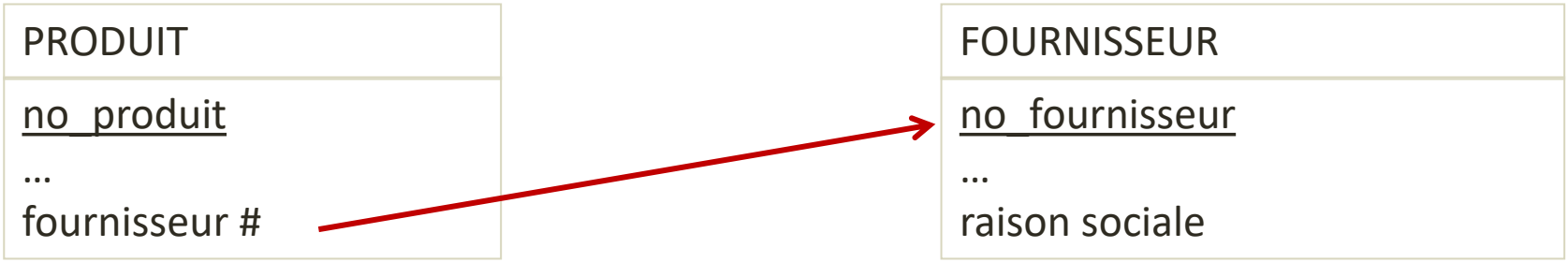
- Non nulle : valeur obligatoire (not NULL)
- Unicité : deux enregistrements ne peuvent pas avoir la même valeur de clé primaire (garanti par le système)
- Minimalité : si la clé primaire est constituée de plusieurs attributs et que l'on retire un de ces attributs (n'importe lequel), alors le groupe d'attributs restants perd la propriété d'unicité
  - Exemple : Table LIGNE_COMMANDE : (Numéro de commande, Référence d'un produit)

Un groupe d'attributs qui vérifie les propriétés mais n'est pas choisi est appelé « clé candidate »

CLÉ ÉTRANGÈRE

Une clé étrangère exprime un lien entre plusieurs tables

- Les valeurs d'une table sont limitées par celles d'un autre table. Par exemple le n° de fournisseur d'un produit est forcément un fournisseur qui existe dans la table fournisseur…



Une clé étrangère dans une table correspond à un attribut ou un groupe d'attributs d'une autre table

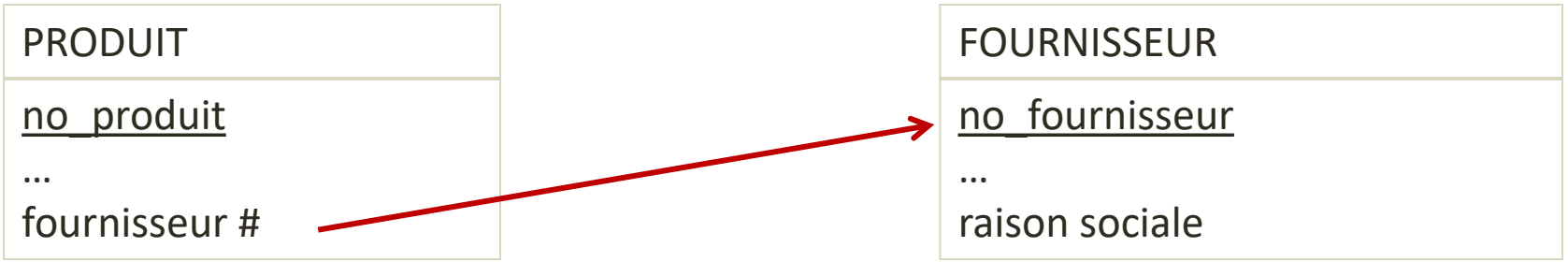
- La cible de la clé étrangère doit être unique. En pratique c'est presque toujours une clé primaire

APPROCHE RELATIONNELLE

Les clés étrangères assurent « l'intégrité référentielle »

- Elles sont notées evace un # dans la table référençante
- Les deux attributs peuvent porter des noms différents

Exemple : Un produit est fourni par un seul fournisseur et pour le connaitre il suffit d'aller chercher la ligne dans la table fournisseur dont le numéro correspond.



BASES DE DONNÉES RELATIONNELLES

Ce qui caractérise une base de données relationnelle ce sont les relations entre les tables

On peut donc dire que l'élément le plus important dans les bases de données relationnelle c'est la notion de clé étrangère qui fait le lien entre les tables

LE LANGAGE SQL

Laboratoire Informatique Image Interaction (L3I)
sité de La Rochelle - Pôle Sciences et Technologie - Avenue Michel Crépeau - 17042 LA ROCHELLE CEDEX 1 Fra
Tél : +33 (0)5 46 45 82 62 – Fax : 05.46.45.82.42 – Site internet : http://l3i.univ-larochelle.fr

LE LANGAGE SQL

Structured Query Language (SQL) est un langage servant à

- Créer, modifier, supprimer des tables, des contraintes…
  - Langage de définition des données (LDD)
- Insérer, modifier, supprimer des enregistrements…
  - Langage de manipulation des données (LMD)
- Récupérer le contenu d'une table, sélectionner des enregistrements vérifiant certains critères, effectuer des calculs…
  - Langage d'interrogation des données (LID)
- Gestion de la sécurité, droits accès…
  - Langage de contrôle des données (LCD)

La syntaxe est quasiment la même dans tous les SGBD relationnels

SYNTAXE GÉNÉRALE

Le LDD de SQL offre un ensemble de commandes pour manipuler les structures

- Créer / Supprimer une base de données :
  - CREATE DATABASE et DROP DATABASE
- Créer / Supprimer une table:
  - CREATE TABLE et DROP TABLE
- Modifier une table:
  - ALTER TABLE

```
-- syntaxe pour créer une base de données

CREATE DATABASE "Gestion";
```

CRÉATION DE TABLES

Syntaxe

- create table <nomtable> (
- <nom_attribut1> <type_att1> <contrainte_att1> <valeur_par_defaut>,
- <nom_attribut1> <type_att2> <contrainte_att2> <valeur_par_defaut>
-);

```
CREATE TABLE commandes (
  -- numéro de commande stocké sur 3 caractères, ni plus ni moins
  no_commande CHAR(3),

  -- la date de commande est juste une date
  date_commande DATE,

  -- le montant est un nombre à 9 chiffres dont 2 après la virgule
  -- par défaut ce montant est égal à 0
  montant NUMERIC(9,2) DEFAULT 0.0,

  -- la tva est un nombre à virgule (float)
  -- le système doit vérifier (check) que la TVA soit bien égale à 19.6 ou 5.5
  -- aucune autre valeur que 19.6 ou 5.5 ne pourra être insérée dans la table
  tva FLOAT CHECK (tva IN (19.6, 5.5))
);
```

EXPRESSION DE CONTRAINTES STATIQUES

Contraintes = conditions que doivent respecter les enregistrements

- Si on essaye d'insérer ou de modifier un enregistrement sans respecter les conditions, alors une erreur est levée et le tuple n'est pas inséré/modifié.

Il existe différents types de contraintes :

- Existence (la valeur de l'attribut ne peut être nulle)
- Unicité (deux enregistrements différents ne peuvent avoir la même valeur pour cet attribut)
- Clé primaire (implique unicité)
- Clé étrangère
- Contraintes de domaine : la valeur de l'attribut doit appartenir à
  - Une liste
  - Un intervalle
  - Un format de données donné

EXPRESSION DE CONTRAINTES STATIQUES

Contrainte d'existence : not null (porte sur un attribut)

- Le champ est forcément renseigné
- Si tentative d'insertion ou de modification sans valeur => erreur

```
CREATE TABLE commandes (
  no_commande CHAR(3),
  date_commande DATE,
  montant NUMERIC(9,2) NOT NULL DEFAULT 0.00 ,
);
```

EXPRESSION DE CONTRAINTES STATIQUES

Contrainte d'unicité : unique (porte sur un ou plusieurs attributs)

- Il ne peut pas y avoir deux enregistrements avec la même valeur de champ
  - Ex : login, numéro INSEE…
- Si tentative d'insertion d'une valeur déjà existante => erreur

```
CREATE TABLE membres(
  no_membre CHAR(10) NOT NULL,
  nom VARCHAR(50) NOT NULL,
  prenom VARCHAR(30) NOT NULL,
  adresse VARCHAR(60),
  PRIMARY KEY(no_membre),
  UNIQUE(no_membre)
);
```

EXPRESSION DE CONTRAINTES STATIQUES

Contrainte de clé primaire : primary key (porte sur un ou plusieurs attributs)

- primary key (attribut1, attribut2, atc.)
- Une clé primaire est forcément unique
- Définition après l'attribut ou en fin de table

```
CREATE TABLE produits(
   ref_produit CHAR(3),
   designation VARCHAR(20) NOT NULL,
   prix_unitaire NUMERIC(9,2) NOT NULL DEFAULT 0.00,
   no_four CHAR(3),
   PRIMARY KEY(ref_produit)
);

CREATE TABLE produits(
   ref_produit CHAR(3) PRIMARY KEY,
   designation VARCHAR(20) NOT NULL,
   prix_unitaire NUMERIC(9,2) NOT NULL DEFAULT 0.00,
   no_four CHAR(3)
);
```

EXPRESSION DE CONTRAINTES STATIQUES

Contrainte de clé primaire : primary key (porte sur un ou plusieurs attributs)

- primary key (attribut1, attribut2, etc.)
- Une clé primaire est forcément unique
- Une clé primaire peut être un ensemble d'attributs

Si clé primaire multiple/composite on doit la définir en fin de table

```
CREATE TABLE lignes_cde (
  no_commande CHAR(3) NOT NULL,
  reference VARCHAR(20) NOT NULL,
  quantite INTEGER NOT NULL DEFAULT 0,
  PRIMARY KEY(no_commande, reference)
);
```

EXPRESSION DE CONTRAINTES STATIQUES

Contrainte de clé étrangère : foreign key … references … (porte sur un ou plusieurs attributs de deux tables)

- L'attribut référencé est forcément unique (mais pas forcément primary)

```
CREATE TABLE commandes(
  no_commande CHAR(3),
  date_commande DATE,
  montant NUMERIC(9,2) NOT NULL DEFAULT 0.00,
  PRIMARY KEY(no_commande)
);

CREATE TABLE lignes_cde (
  no_commande CHAR(3) NOT NULL,
  reference VARCHAR(20) NOT NULL,
  quantite INTEGER NOT NULL DEFAULT 0,
  --           attribut                table et attribut cible
  FOREIGN KEY (no_commande) REFERENCES commandes (no_commande)
);
```

EXPRESSION DE CONTRAINTES STATIQUES

Contrainte de clé étrangère : foreign key … references …

- L'attribut référencé est forcément unique (mais pas forcément primary)
- La cible doit exister au moment de la création, il faut donc créer les tables dans le bon ordre
  - Une solution est d'abord de créer toutes les tables puis de rajouter les clés étrangères
  - On pourra faire sans dans ce cours

```
CREATE TABLE lignes_cde (
  no_commande CHAR(3) NOT NULL,
  reference VARCHAR(20) NOT NULL,
  quantite INTEGER NOT NULL DEFAULT 0
);

…

ALTER TABLE lignes_cde
  ADD FOREIGN KEY (no_commande) REFERENCES commandes(no_commande)
```

EXPRESSION DES AUTRES CONTRAINTES STATIQUES

Contrainte de domaine : check (porte sur un attribut)

- Expression d'une liste de valeurs possibles : IN
- Expression d'un intervalle de valeurs possibles : BETWEEN + AND
  - Fonctionne pour les nombres, les textes et les dates

```
CREATE TABLE produits (
  reference CHAR(3),
  tva FLOAT NOT NULL DEFAULT 19.6 CHECK (tva IN (19.6, 5.5)),
  quantite INTEGER DEFAULT 1 CHECK (quantite BETWEEN 1 AND 100),
  PRIMARY KEY(reference)
);
```

EXPRESSION DES AUTRES CONTRAINTES STATIQUES

Contrainte de domaine : check (porte sur un attribut)

- Expression d'un format de données : LIKE
- Les symboles _ et % sont des jokers :
  - _ permet de remplacer un caractère quelconque
  - % permet de remplacer n'importe quel nombre de caractères (y compris 0)

```
CREATE TABLE produits (
  -- reference = lettre R suivie de deux caractères quelconques
  reference CHAR(3) CHECK (reference LIKE 'R__')

  -- l'intitulé du produit doit commencer par le mot produit suivi de
  n'importe quoi
  intitule VARCHAR(255) CHECK (intitule LIKE 'produit%')
);
```

MODIFICATION DE LA STRUCTURE D'UNE TABLE

Non utilisé dans ce cours mais pour info

Syntaxe : ALTER TABLE <nomtable> ACTION <modification>

- Action = ADD : ajouter un attribut, une contrainte…
  - alter table COMMANDE add date_commande date;
  - alter table LIGNES_CDE add primary key (no_commande, reference);

- Action = ALTER : modifier un attribut, une contrainte…
  - alter table PRODUITS alter tva set default 19.6;

- Action = DROP : supprimer un attribut, une contrainte…
  - alter table COMMANDE drop date_commande;
  - alter table COMMANDES drop constraint fk_commandes_clients;

LE LANGAGE SQL

Structured Query Language (SQL) est un langage servant à

- Créer, modifier, supprimer des tables, des contraintes…
  - Langage de définition des données (LDD)
- Insérer, modifier, supprimer des enregistrements…
  - Langage de manipulation des données (LMD)
- Récupérer le contenu d'une table, sélectionner des enregistrements vérifiant certains critères, effectuer des calculs…
  - Langage d'interrogation des données (LID)
- Gestion de la sécurité, droits accès…
  - Langage de contrôle des données (LCD)

La syntaxe est quasiment la même dans tous les SGBD relationnels

SYNTAXE GÉNÉRALE

Ces commandes s'appliquent aux données et non plus à la structure

Insérer des données dans une table : INSERT

- INSERT INTO nomtable(att1, att2, att3) VALUES (valeur1, valeur2,valeur3)

INSERTION DE TUPLES

```
CREATE TABLE produits (
  ref_produit CHAR(4),
  designation VARCHAR(20) NOT NULL,
  prix_unitaire NUMERIC(9,2) NOT NULL DEFAULT 0.00,
  no_four CHAR(3),
  PRIMARY KEY (ref_produit)
);

INSERT INTO produits (ref_produit, designation , prix_unitaire, no_four)
VALUES ('R01', 'trombone',0.01,’10’)

 -- équivalent à (on ne précise pas les attributs car ils sont tous là)
INSERT INTO produits
VALUES ('R01', 'trombone',0.01,10)

-- le dernier attribut est optionnel, donc on peut avoir :
INSERT INTO produits (ref_produit, designation , prix_unitaire)
VALUES ('R01', 'trombone',0.01)
```

SYNTAXE GÉNÉRALE

Ces commandes s'appliquent aux données et non plus à la structure

Insérer des données dans une table : INSERT

- INSERT INTO nomtable(att1, att2, att3) VALUES (valeur1, valeur2,valeur3)

Modifier des données dans une table: UPDATE

- UPDATE nomtable SET att1= valeur1, att2=valeur2, att3=valeur3 WHERE conditions

Supprimer des données de la table : DELETE

- DELETE FROM nomtable WHERE conditions

MODIFICATION ET SUPPRESSION

```
CREATE TABLE produits (
  ref_produit CHAR(4),
  designation VARCHAR(20) NOT NULL,
  prix_unitaire NUMERIC(9,2) NOT NULL DEFAULT 0.00,
  no_four CHAR(3),
  PRIMARY KEY (ref_produit)
);

-- modifie la référence et la désignation du produit R01
UPDATE produits
SET ref_produit='R001', designation = 'Trombone‘
WHERE ref_produit='R01';

-- supprime tous les produits du fournisseur 1
DELETE FROM produits WHERE no_four=1;
```

A RETENIR

Structuration d'une base avec tables, attributs, enregistrements

Les différentes contraintes s'appliquant sur un ou plusieurs attributs

- Les types
- Les conditions d'unicité (UNIQUE), d'existence (NOT NULL), de domaine (CHECK)
- Les valeurs par défaut
- Les clés primaires (PRIMARY KEY)
- Les clés étrangères (FOREIGN KEY)