

Gestion des données

Bases de données relationnelles

HEI 2019 / 2020

Introduction

- Les bases de données relationnelles sont le type de base de données majoritaire actuellement.
- La communication avec la grande majorité des BDD relationnelles se fait avec le langage Structured Query Language (SQL).

Exemples de base de données relationnelle



MariaDB

- Pendant ce cours et celui de développement web, MariaDB sera utilisé.
- MariaDB est un fork de MySQL et les deux logiciels sont actuellement encore très similaires.

Logiciel serveur

- MariaDB, comme les autres bases de données, est un logiciel de type serveur.
- Une fois installé sur un système, il fonctionne en tâche de fond et expose un port pour pouvoir s'y connecter.
- Le port par défaut pour MariaDB est 3306.

Logiciel client

- Pour pouvoir manipuler la base de données, il va falloir s'y connecter en utilisant un logiciel client.
 - La plupart des logiciels client pour MySQL sont compatibles avec MariaDB.
- Le logiciel client va permettre d'interagir avec les données, la plupart du temps via une interface graphique.

Connexion

- Pour se connecter à la BDD, un compte utilisateur est nécessaire.
 - Dans notre cas, connexion par identifiant et mot de passe
- Un compte utilisateur est associé à un certains nombre de droits qui vont déterminer les actions possibles.

Databases

- Un serveur MariaDB contient plusieurs ensembles indépendants appelés **databases** ou **schemas**.
- Cela permet d'avoir au sein de la même instance du serveur de base de données plusieurs ensembles logiques pour, par exemple, servir à 2 sites web différents.

Langage SQL

- Le langage SQL permet la communication avec la BDD relationnelle. C'est un standard utilisé par la majorité des BDD.
 - Chaque BDD va ajouter quelques spécificités au standard.
- Nous allons voir 2 domaines du langage SQL :
 - Le **langage de manipulation des données** permettant d'ajouter, lire, modifier, et supprimer des objets.
 - Le **langage de définition des données** permettant de décrire la structure des objets et leurs relations.

Les tables

Tables

- Les objets vus en modélisation sont représentés dans une BDD relationnelle sous la forme de tableaux appelés **tables**.
- Une table est constituée de **colonnes** représentant les différents attributs de l'objet.
- Chaque instance d'un type d'objet est donc représenté par une **ligne** dans une table.

Colonnes

- Les tables SQL contiennent des données formatés.
- Chaque colonne possède des caractéristiques comme :
 - Type : le type de données
 - Not Null : le caractère obligatoire ou non
 - Default : la valeur par défaut

Types de données (non exhaustif)

Type	Description	Remarque
varchar(n)	Chaine de caractères de taille variable (max n caractères)	
int	Nombre entier de 32 bits	De 0 à 4 294 967 295 en non signé De -2 147 483 648 to 2 147 483 647 en signé
tinyint	Nombre entier de 8 bits	De 0 à 255 en non signé De -128 à 127 en signé
float	Nombre à virgule flottante	
decimal(n,d)	Nombre à virgule de n chiffres dont d après la virgule	
date	Date	
datetime	Date et heure	
blob	Données binaires (65mo max)	

Syntaxe SQL

- **CREATE TABLE** permet d'ajouter une nouvelle table dans la base de données

```
CREATE TABLE [nom] (  
    [colonne 1],  
    [colonne 2],  
    [colonne 3]  
);
```

Définition des colonnes

- Pour chaque colonne, il est nécessaire de spécifier son nom ainsi que son type. Les autres caractéristiques sont facultatives.

```
CREATE TABLE person (  
    name          VARCHAR(50) NOT NULL,  
    firstname     VARCHAR(50) NOT NULL,  
    birthdate     DATE NULL,  
    admin         TINYINT NULL DEFAULT 0  
);
```

Clé primaire

- L'identifiant d'un objet est appelé **Clé primaire**.
- La clé primaire peut être précisée dans la requête de création de table après la définition des colonnes avec l'instruction **PRIMARY KEY**.
- Il suffit de préciser la ou les colonnes constituant la clé primaire.

PRIMARY KEY ([colonne1] , [colonne2])

Exemple avec une clé primaire

```
CREATE TABLE person (  
    name          VARCHAR(50) NOT NULL,  
    firstname     VARCHAR(45) NOT NULL,  
    birthdate     DATE NULL,  
    admin         TINYINT NULL DEFAULT 0,  
    PRIMARY KEY (name, firstname)  
);
```

Auto-incrément

- Très fréquemment, la clé primaire des objets va être un champ technique géré par la base de données elle-même.
- MariaDB permet d'avoir une colonne de type entier avec un auto-incrément.
- Ainsi si aucune valeur n'est spécifiée pour cette colonne, la valeur sera un incrément de la valeur existante maximale.

Exemple avec une clé primaire en auto-incrément

```
CREATE TABLE person (  
    id          INT NOT NULL AUTO_INCREMENT,  
    name        VARCHAR(50) NOT NULL,  
    firstname   VARCHAR(45) NOT NULL,  
    birthdate   DATE NULL,  
    admin       TINYINT NULL DEFAULT 0,  
    PRIMARY KEY (id)  
);
```

Les relations

Relations conceptuelles vs relations pratiques

- Les relations de nos modélisations de données ne peuvent pas être représentées telles quelles dans notre base de données.
- Il va falloir appliquer des règles pour pouvoir ajouter des colonnes et clés dans nos tables permettant de les représenter.

Types de relation

- Suivant la cardinalité des relations, il est possible d'en extraire 3 types de relation. Ces 3 types se basent sur la cardinalité maximale de chaque coté de la relation.
- On a donc des relations :
 - De type 1-1
 - De type 1-N
 - De type N-N

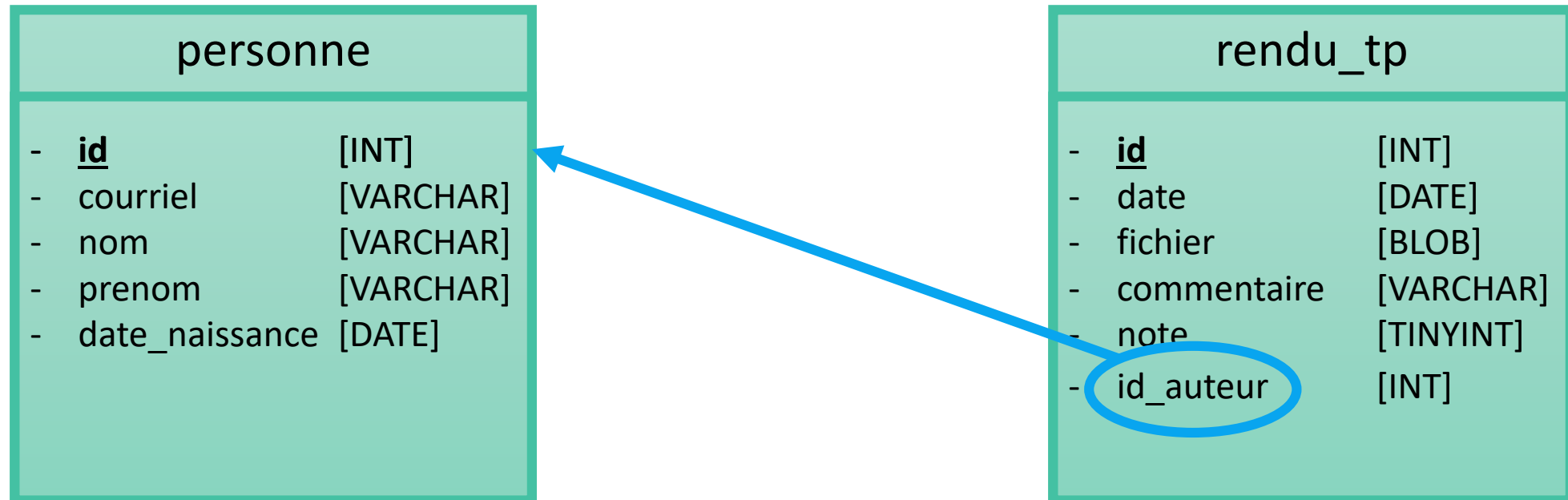
Relation 1-N

- Dans une relation 1-N, le lien entre les 2 objets va être porté par l'objet « du côté N » de la relation.



- La clé primaire de la table **1** va être ajouté dans une ou des colonnes supplémentaires de la table **2**.

Relation 1-N



Clé étrangère

- Pour faire le lien entre les 2 tables, une clé étrangère est ajoutée sur le champ référençant la clé primaire de l'autre table.

```
CONSTRAINT [nom_cle]      Colonne(s) locale(s)
  FOREIGN KEY ( [colonne1], [colonne2] )
  REFERENCES [table] ( [colonne1], [colonne2] )
```

Table distante Colonne(s) distante(s)

- La référence doit forcément correspondre à une clé primaire.

Exemple de clé étrangère

```
CREATE TABLE rendu_tp (  
    id INT NOT NULL AUTO_INCREMENT,  
    date DATE NOT NULL,  
    fichier BLOB NOT NULL,  
    id_auteur INT NOT NULL,  
    PRIMARY KEY (id),  
    CONSTRAINT fk_rendu_auteur  
        FOREIGN KEY (id_auteur) REFERENCES person (id)  
);
```

Clé étrangère et contrainte

- Une clé étrangère est une contrainte.
- La valeur de la colonne marquée comme une clé étrangère **doit** exister dans l'autre table.
 - En conséquence, il n'est pas possible de supprimer une ligne qui est pointée par une clé étrangère.

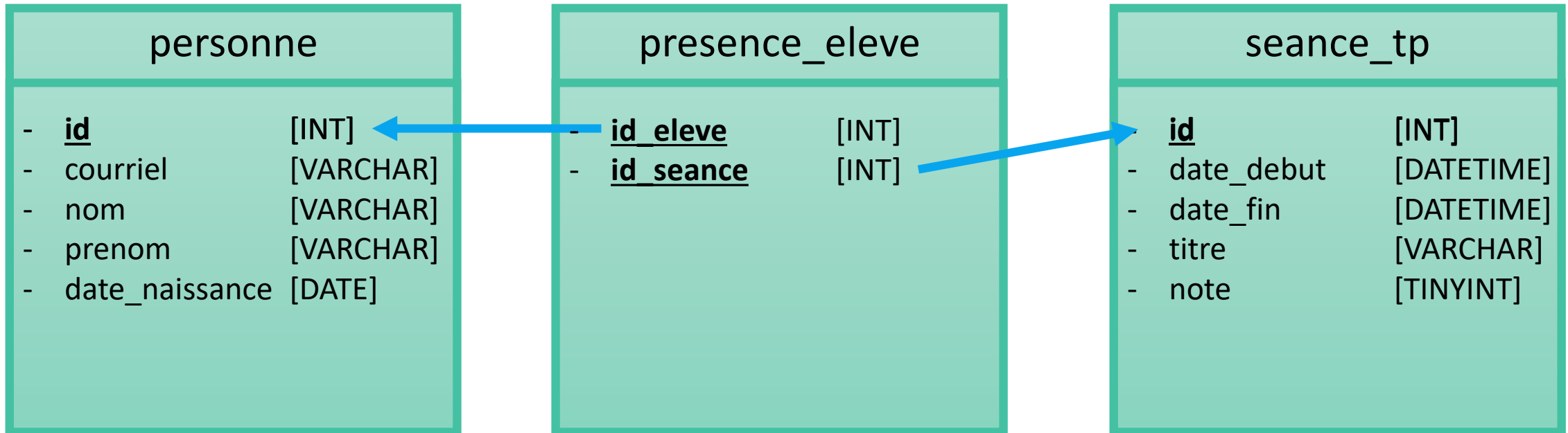
Relation 1-1

- Une relation 1-1 va être construite comme une relation 1-N : une clé étrangère va être ajoutée dans l'une des 2 tables.
- Pour choisir dans quel objet positionner la clé étrangère, il faut réfléchir à quel objet est le plus dépendant de l'autre, quel objet peut être supprimé sans impacter l'autre.

Relation N-N

- Une relation N-N est un peu plus complexe à mettre en place car elle va nécessiter la création d'une table supplémentaire appelée **table de correspondance**.
- La clé primaire de chacune des 2 tables est ajoutée à cette nouvelle table en tant que clé étrangère.
- La clé primaire de la table de correspondance est la combinaison des 2 clés primaires.

Relation N-N



Autres opérations

Supprimer une table

- L'instruction **DROP TABLE** permet de supprimer une table.

DROP TABLE [nom] ;

- Il n'est pas possible de supprimer une table si elle est la cible d'une clé étrangère.

Modifier une table

- L'instruction **ALTER TABLE** permet de modifier une table. Elle consiste en un ensemble d'action sur les colonnes ou clés d'une table.

```
ALTER TABLE rendu_tp  
  DROP COLUMN id_auteur,  
  ADD COLUMN rendu_tpcol VARCHAR(45) NULL,  
  DROP INDEX fk_rendu_auteur ;
```

Conclusion

Définition des données

- On vient de voir les bases du langage de définition des données en SQL, particulièrement pour une base MariaDB.
- Cela permet de matérialiser la modélisation en une base de données pratique avec lequel un logiciel va interagir.
- Ce ne sont pas des requêtes SQL effectuées fréquemment dans la vie d'un projet. Elles interviennent principalement dans la phase de construction du logiciel.

Interfaces graphiques

- S'il est important de connaître les différentes requêtes de définition des données, il existe dans les clients graphiques des interfaces qui permettent de créer les tables et clés plus facilement.

The screenshot displays a database management interface for creating a new table named 'rendu_tp' in the 'hei' schema. The table is configured with the 'utf8 - default collation' and 'InnoDB' engine. A table of columns is shown below, with the 'date' column selected for further configuration.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
fichier	BLOB	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
id_auteur	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

The 'date' column configuration panel shows the following details:

- Column Name: date
- Data Type: DATE
- Collation: Table Default
- Comments: (empty text area)
- Storage: ☐ Virtual ☐ Stored
- ☐ Primary Key ☒ Not Null ☐ Unique
- ☐ Binary ☐ Unsigned ☐ Zero Fill
- ☐ Auto Increment ☐ Generated

At the bottom, a navigation bar includes tabs for Columns, Indexes, Foreign Keys, Triggers, Partitioning, and Options.

Exercice : connexion à la base de données

- Se connecter à : <http://donnees.chticod.eu/>
 - Le logiciel phpMyAdmin a été installé sur le serveur et configuré pour se connecter à une base MariaDB.
- Un compte a été créé pour chacune et chacun sous la forme :
 - Identifiant : prenom.nom (tel que dans votre adresse email HEI)
 - Mot de passe : ITI2019

Exercice : bibliothèque en pratique

- Reprendre le modèle de données du logiciel de gestion de bibliothèque vu précédemment.
- Créer les tables et relations dans une base MariaDB

Pour les prochaines fois

- Il va être nécessaire d'avoir installés sur votre ordinateur personnel :
 - Une base de données MariaDB
 - Un logiciel client pour s'y connecter
- Si vous avez déjà un serveur MySQL sur votre ordinateur, cela suffit.
- Vous êtes libre de sélectionner les logiciels de votre choix mais je vous propose d'utiliser XAMPP.

XAMPP

- Site de XAMPP : <https://www.apachefriends.org>
- XAMPP permet une installation simple d'un serveur MariaDB avec un client phpMyAdmin préconfigurés.
- D'expérience sur les années précédentes, XAMPP est la solution la plus simple pour installer la base de données et un client.