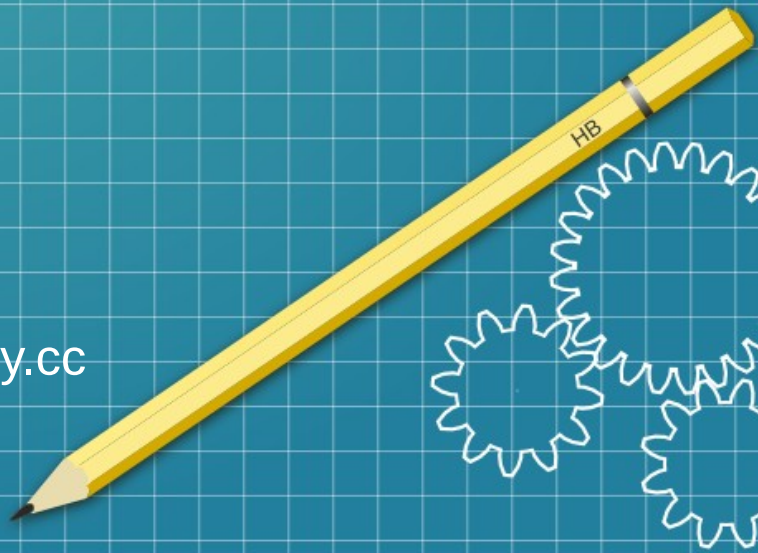
Base de données

Développement Web

stephane.pau@lery.cc



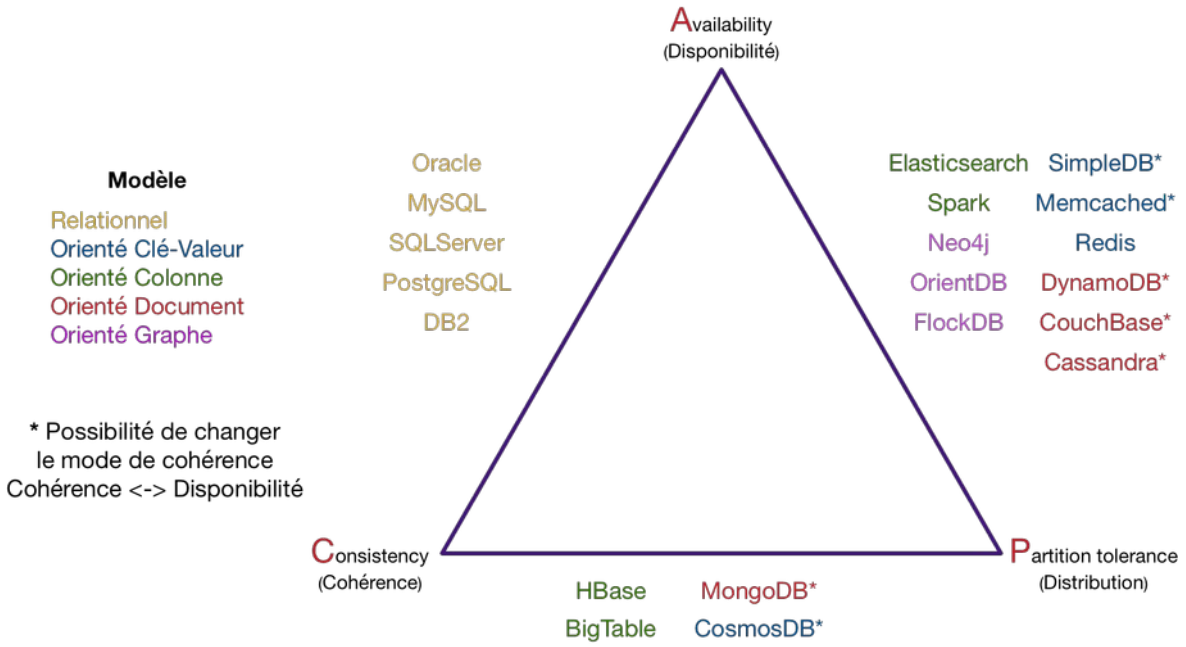
Stocker les données

- Fichier INI (clé → valeur)
- Fichier standard (.doc, xls, access,)
- Logiciels spécialisés : SGBD
- Bases de données NoSQL
- Entrepôt GED

- Besoins
  - Cohérence
  - Disponibilité
  - Distribué

Stocker les données

- Théorème CAP



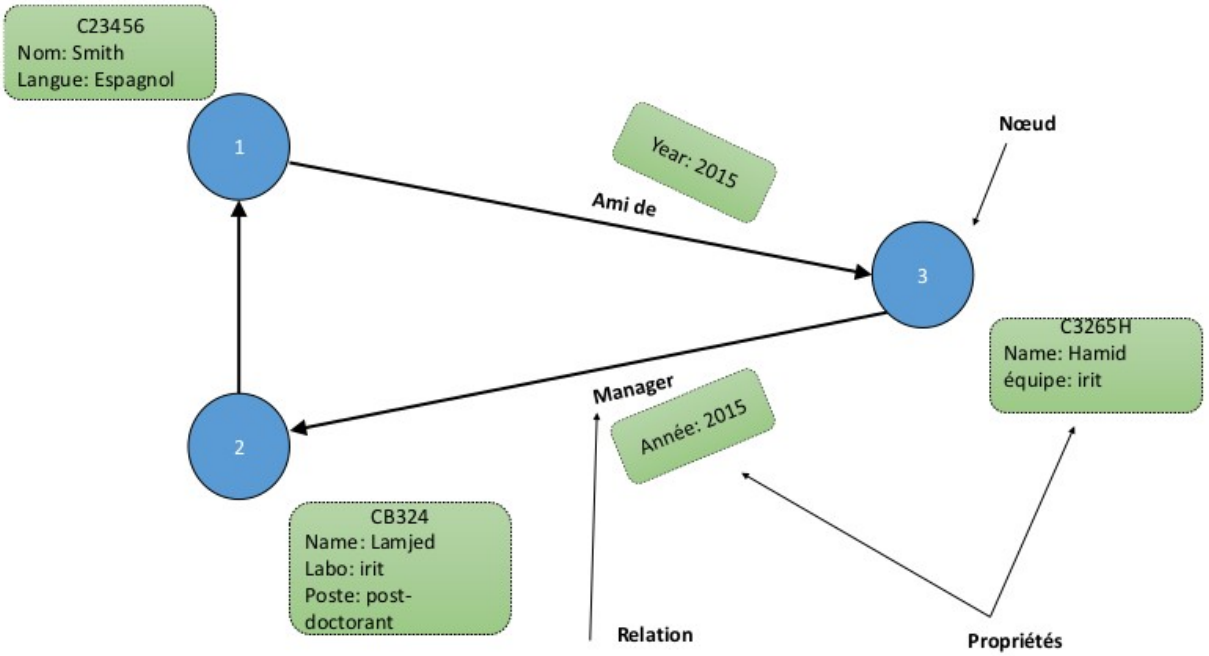
Base NoSQL (Not Only SQL)

- Modèle orienté Clé-valeur
  - Semblable à un tableau associatif à deux colonnes
  - Clé et valeur de format libre
- Modèle orienté documents
  - Paires clé-valeur
  - la clé identifie une valeur structurée (le document). Le document est généralement au format JSON

Base NoSQL (Not Only SQL)

- Modèle orienté colonnes
  - Pour chaque enregistrement (ligne), le nombre de colonne et le type peut être différent. Cela permet de gérer le versioning
- Modèle orienté graphe
  - Ce modèle repose sur 3 notions
    - noeud
    - relation
    - propriété
  - La recherche des proches voisins d'un noeud est rapide

Modèle orienté graphe



SGBDR

Nous allons étudier un sous-ensemble très important

SGDB(R) ou RDBMS

Système de **G**estion de **B**ase de **D**onnées **R**elationnel

Relational **D**ata**B**ase **M**anagement **S**ystem

SGBD

- Nécessiter de modéliser avant de développer
  - Comprendre le besoin client
  - Partager un langage commun
- Deux systèmes de modélisation
  - Merise (Méthode d'étude et de réalisation informatique pour les systèmes d'entreprise)
  - UML (Unified Modeling Langage)

Forme normale

- En respectant un forme normal pour une base de données, on évite les incohérences et la redondance des données.
- Il existe 8 formes normales dont les 3 premières sont les plus importantes.
- Le respect d’une forme normale inférieure implique que la forme normale du niveau supérieure est déjà respectée (Le respect de la forme normale 2 n’est possible que si la forme normale 1 est respectée)

1FN

- Première Forme normale
  - Tous les attributs contiennent une valeur atomique
  - 1 table représente 1 entité
  - Les attributs sont constants dans le temps (date de naissance plutôt que âge)

1FN

Table clients

id	société	Adresse	Contact_1	Contact_2	
1053	Orange	Rue de la fibre	Alice	Pierre	
292	3M	Av. du scotch	Alain	Mona	

Création d’une nouvelle entité « contact »

Table clients

id	société	Adresse	contact_id
1053	Orange	Rue de la fibre	1
292	3M	Av. du scotch	2
1053	Orange	Rue de la fibre	3
292	3M	Av. du scotch	4

Table contacts

id	nom
1	Alice
2	Alain
3	Pierre
4	Mona

2FN

- Seconde forme normale
  - Pas de redondance des données

2FN

Table clients

id	société	Adresse	contact_id
1053	Orange	Rue de la fibre	1
292	3M	Av. du scotch	2
1053	Orange	Rue de la fibre	3
292	3M	Av. du scotch	4

Supprimer la redondance : Création d'une table intermédiaire

Table clients

id	Société	Adresse
1053	Orange	Rue de la fibre
292	3M	Av. du scotch

Table clien

clients_id	contacts_id
1053	1
1053	3
292	2
292	4

Table contacts

id	nom
1	Alice
2	Alain
3	Pierre
4	Mona

3FN

- Il ne doit pas y avoir de dépendance transitive. C’est à dire que certains éléments de la table ne dépendent pas de de l’objet principal de la table
- Si cette règle n’est pas respectée, on peut arriver à terme à une redondance des entités

3FN

Table clients

id	Société	Adresse
1053	Orange	Rue de la fibre
292	3M	Av. du scotch

Table clien

clients_id	contacts_id
1053	1
1053	3
292	2
292	4

Table contacts

id	nom
1	Alice
2	Alain
3	Pierre
4	Mona

- Plusieurs sociétés peuvent être dans la même rue

3FN

- Solution respectant la 3ième forme normale

Table clients

id	Société	Adresse_id
1053	Orange	1721
292	3M	1945

Table clien

clients_id	contacts_id
1053	1
1053	3
292	2
292	4

Table contacts

id	nom
1	Alice
2	Alain
3	Pierre
4	Mona

Table Adresse

id	adresse
1721	Rue de la fibre
1945	Av. du scotch

Modélisation Merise

- Installer logiciel JMerise ou Looping https://www.looping-mcd.fr/
- Baser sur 3 modèles
  - MCD : Modèle Conceptuel de Données
  - MLD : Modèle Logique de Données
  - MPD : Modèle Physique de Données

Dictionnaire de Données

- Le dictionnaire contient les informations suivantes :
  - Nom de l’attribut
  - Description
  - Type de la donnée
    - A (Alphabétique). Uniquement des caractères alphabétiques
    - N (Numérique). Uniquement des nombres (entiers ou réels)
    - AN (Alphanumérique)
    - Date
    - Booléen (True ou False)
    - Taille Nombre de chiffres ou caractères
    - Commentaire Règle métier particulière

Dictionnaire de Données

- Exemple

Code mnémonique	Désignation	Type	Taille	Remarque
id_i	Identifiant numérique d'un inscrit	N		
nom_i	Nom d'un inscrit	A	30	
prenom_i	Prénom d'un inscrit	A	30	
rue_i	Rue où habite un inscrit	AN	50	
ville_i	Ville où habite un inscrit	A	50	
cp_i	Code postal d'un inscrit	AN	5	
tel_i	Numéro de téléphone fixe d'un inscrit	AN	15	
tel_port_i	Numéro de téléphone portable d'un inscrit	AN	15	
email_i	Adresse e-mail d'un inscrit	AN	100	

Merise : Etape 1

- Recensement
  - Rechercher toutes les données qui seront utilisées selon le travail demandé
  - Utiliser le dictionnaire de données
- Regrouper les données qui identifient les objets. Ce sont les entités.
  - Une donnée ne peut être affectée qu'à une seule entité
  - Une donnée n'est affectée à une classe d'entités que s'il n'existe qu'une seule valeur de cette donnée pour chaque entité de la classe

Entité

- Règle 1 : Une donnée ne peut être affectée qu'à une seule entité
  - Exemple : Vous avez la donnée "nom". Pour votre cas d'étude, le nom représente le nom d'une personne ou la désignation d'un produit
  - Vous avez en fait 2 données différentes
- Règle 2 : Une donnée n'est affectée à une classe d'entités que s'il n'existe qu'une seule valeur de cette donnée pour chaque entité de la classe
  - Vous souhaitez enregistrer les adresses postales de l'entreprise. Celle-ci dispose de plusieurs adresses. On créera une entité adresse

Merise

- MCD : Modèle Conceptuel de Données

  Le Modèle Conceptuel de Données a pour but de décrire les données et les objets (entités) utilisés et manipulés par le système d'information au travers de l'application.

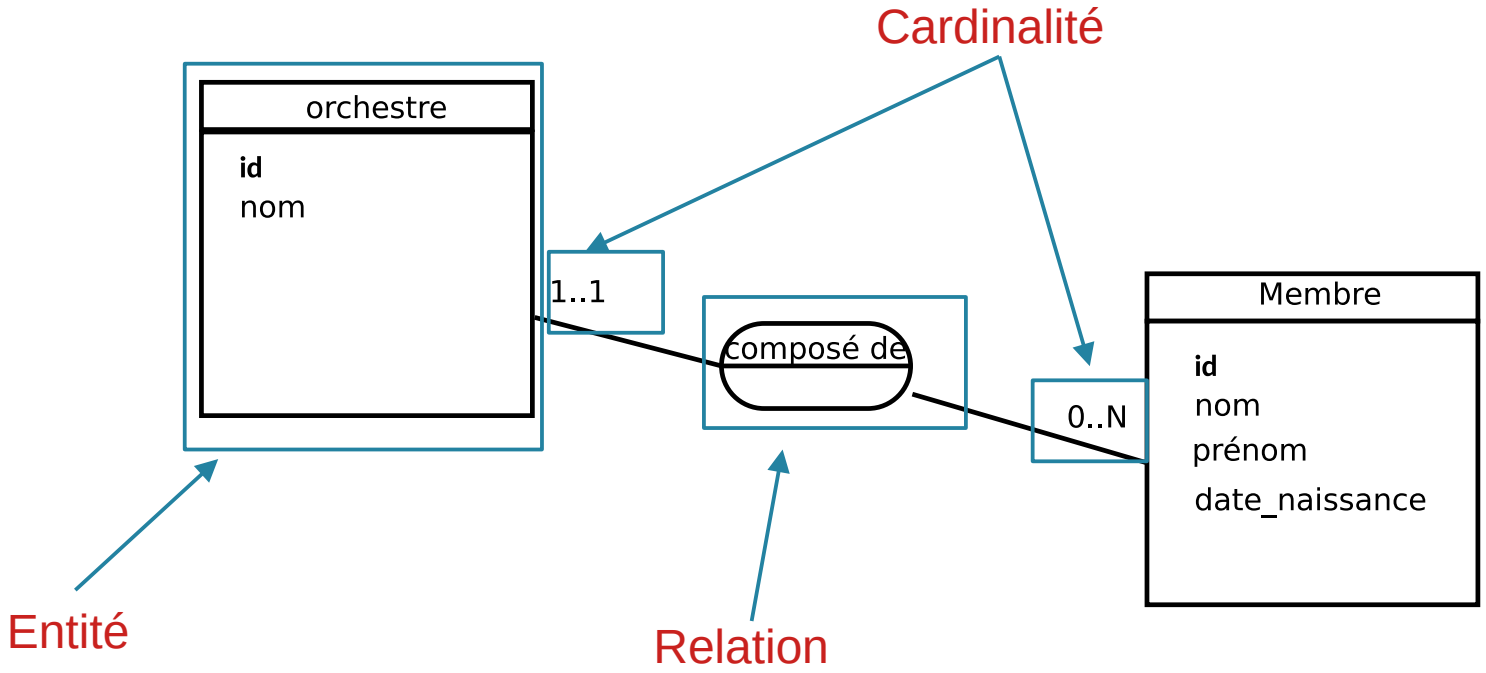
- Représentation d'une entité
  - Rectangle séparé en deux
    - Libellé
    - attributs

Merise

- Relations
  - Une relation (appelée aussi parfois association) représente les liens qui peuvent exister entre plusieurs entités
- Cardinalité
  - Les cardinalités permettent de caractériser le lien qui existe entre une entité et la relation à laquelle elle est reliée. On indique la borne minimale et la borne maximale (plusieurs est représenté par la lettre N ou *)
- Identifiants
  - L'identifiant d'une classe d'entité permettant de désigner de façon unique une instance de cette entité. Dans le modèle Merise, cet attribut est souligné.

Merise

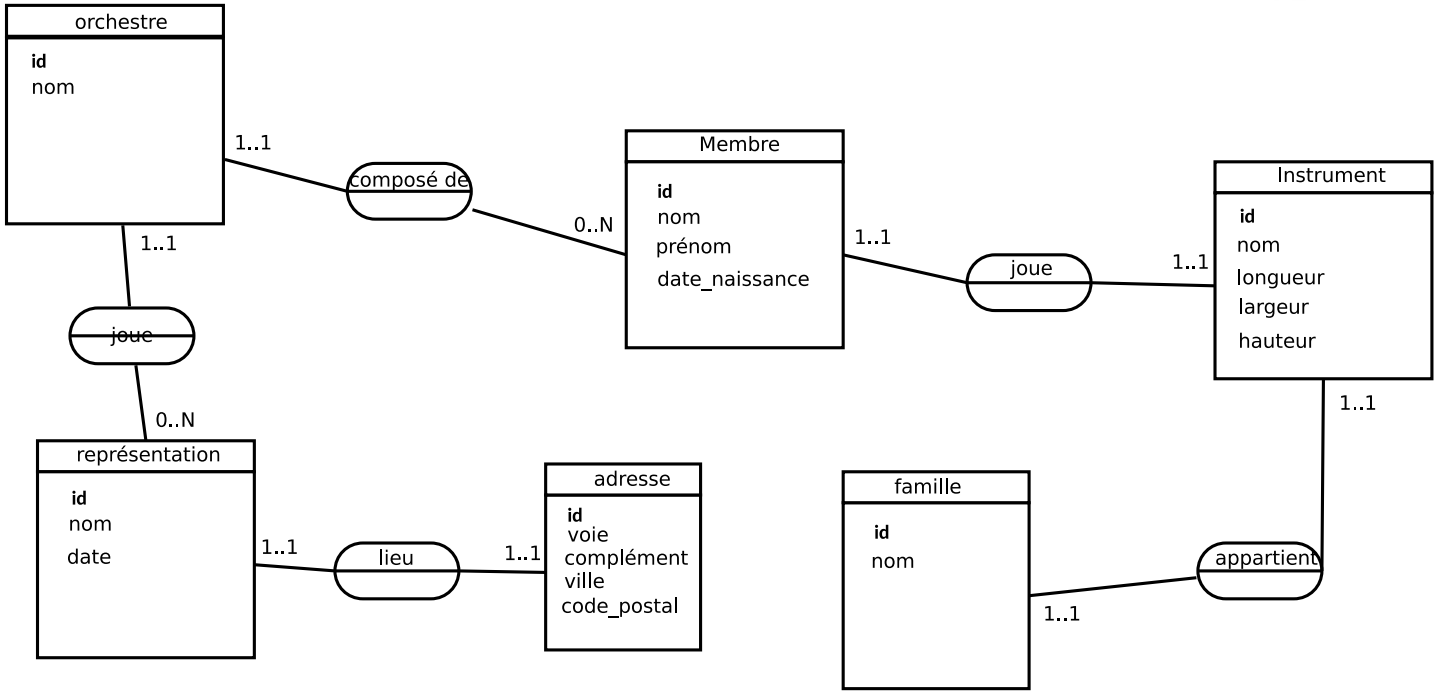
- Exemple de MCD



Exercice

- Un groupe de musique (orchestre) souhaite gérer les instruments utilisés par les membres du groupe pour organiser les tournées

Exemple : Orchestre



Du MCD au MLD

- MLD : Modèle Logique de Données

- Une entité devient une table nommée
- Un attribut devient une colonne
- Les identifiants deviennent les clés primaires
- Les clés étrangères sont indiquées
- Les tables pivot (table de relations) sont ajoutées

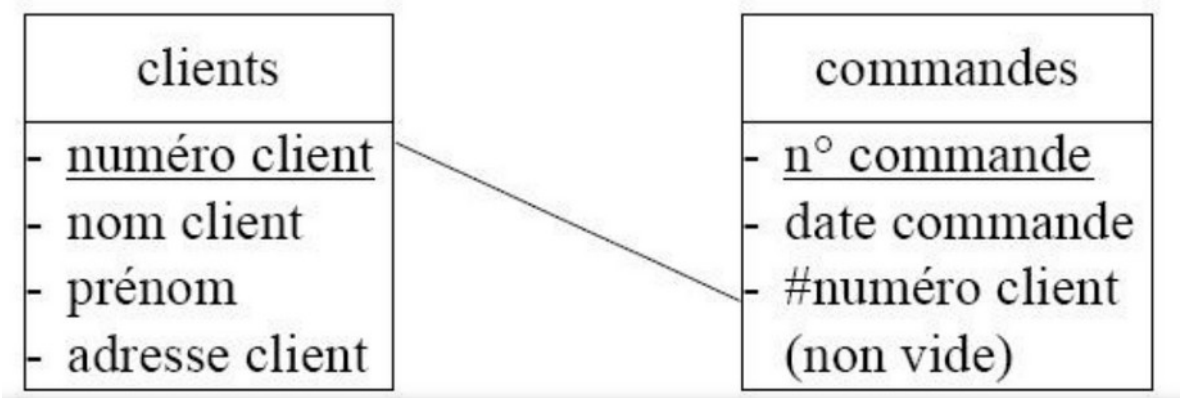
Clé primaire

- Caractéristique d’une clé primaire
  - Valeur unique
  - NULL est interdit
  - Une seule clé primaire par table (peut être composé de plusieurs colonnes
  - Une clé primaire ne change pas au cours du temps
  - Sur le MCD, la clé primaire est souligné

Clé étrangère

- La clé étrangère est précédée du signe #
- Une table peut avoir plusieurs clés étrangère

- Exemple



Du MLD au MPD

- MPD : Modèle Physique de Données

- Le type des attributs est spécifié ainsi que la taille

- Les types SQL
  - INT (et dérivés TINYINT, MEDIUMINT, BIGINT)
  - VARCHAR (et dérivés MEDIUMTEXT, TEXT)
  - BOOLEAN
  - DATE
  - TIME
  - BINARY (Large Object BLOB)
  - DATETIME
  - TIMESTAMP
  - GUID

MCD -> MLD -> MPD

- MCD
  - Graphique compréhensible par une personne qui n'est pas du métier
- MLD
  - Traduction du MCD selon les principes du modèle relationnel (présence des clés)
- MPD
  - Très proche du SGBDR utilisé. On utilise la notation du langage SQL

Installer un SGBDR open-source

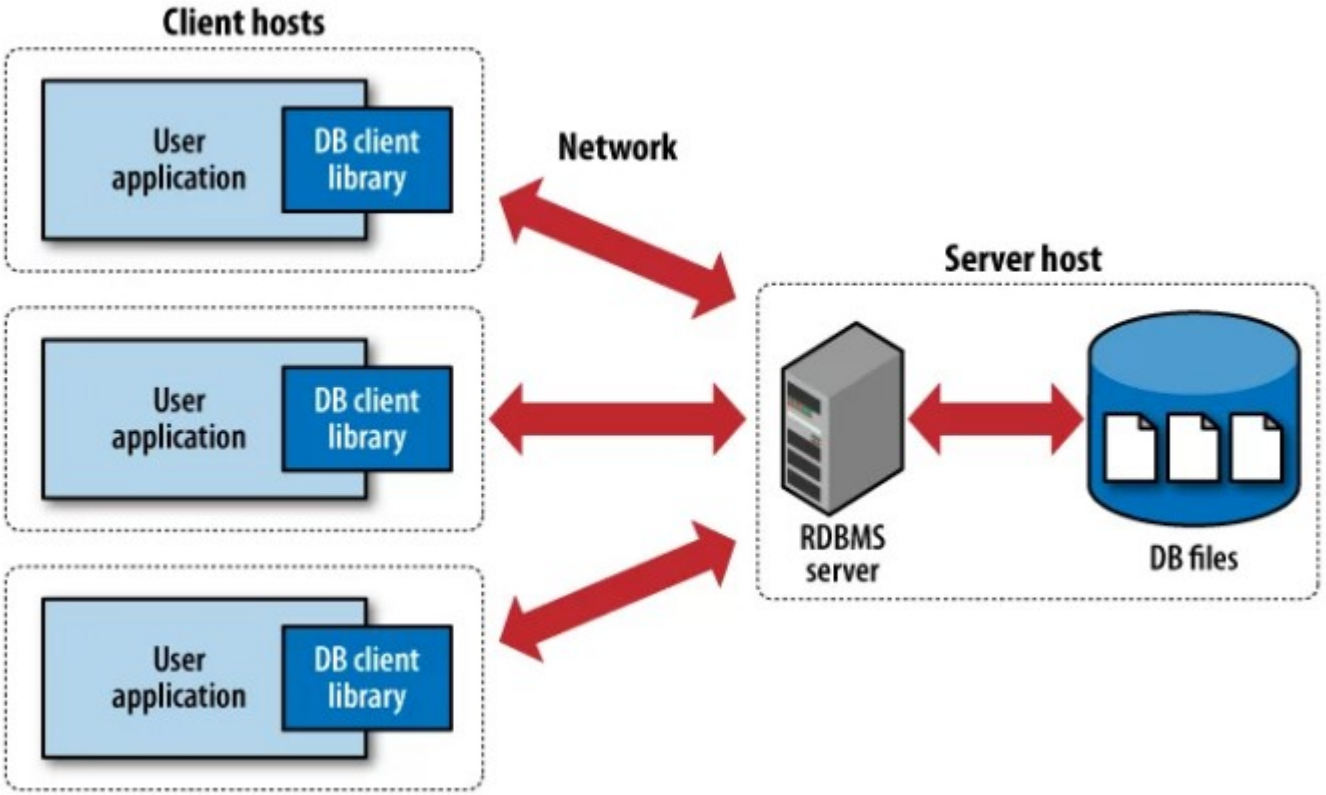
- MySQL/MariaDB
- Postgresql

Installer un outil permettant de voir les données

- dbeaver-ce
- phpmyadmin/pgadmin

Créer des tables et ajouter des données

Architecture client serveur



Nécessité d’utiliser un langage d’interrogation du serveur SGBD

SQL

- Structured Query Language
  - INSERT
  - SELECT
  - UPDATE
  - DELETE
- Opérations
  - CREATE
  - READ
  - UPDATE
  - DELETE
- Verbes HTTP
  - POST
  - GET
  - PUT
  - DELETE

Synthèse du langage SQL

SQL

- Requête sur la base représentation
  - Création tables
  - Insertion de données
  - Recherche de données
  - Modification d’un enregistrement
  - Suppression d’une ligne

SQL

- Requête sur la base représentation
  - Création tables
    - CREATE TABLE nom_de_la_table
      (
        colonne1 type_donnees,
        colonne2 type_donnees,
        colonne3 type_donnees,
        colonne4 type_donnees
)
  - Insertion de données
  - Recherche de données
  - Modification de données
  - Suppression de données

SQL

- Requête sur la base représentation
  - Création tables
    - CREATE TABLE nom_de_la_table
      (
        colonne1 type_donnees,
        colonne2 type_donnees,
        colonne3 type_donnees,
        colonne4 type_donnees
)
  - Insertion de données
    - INSERT INTO table VALUES ('valeur 1', 'valeur 2', …)
    - INSERT INTO table (nom_colonne_1, nom_colonne_2, ...) VALUES ('valeur 1', 'valeur 2', ...)
  - Recherche de données
  - Modification de données
  - Suppression de données

SQL

- Requête sur la base représentation
  - Création tables
    - CREATE TABLE nom_de_la_table
      (
        colonne1 type_donnees,
        colonne2 type_donnees,
        colonne3 type_donnees,
        colonne4 type_donnees
)
  - Insertion de données
    - INSERT INTO table VALUES ('valeur 1', 'valeur 2', …)
    - INSERT INTO table (nom_colonne_1, nom_colonne_2, ...) VALUES ('valeur 1', 'valeur 2', ...)
  - Recherche de données
    - SELECT nom_colonnes FROM nom_table
  - Modification de données
  - Suppression de données

SQL

- Requête sur la base représentation
  - Création tables
    - CREATE TABLE nom_de_la_table
      (
        colonne1 type_donnees,
        colonne2 type_donnees,
        colonne3 type_donnees,
        colonne4 type_donnees
)
  - Insertion de données
    - INSERT INTO table VALUES ('valeur 1', 'valeur 2', …)
    - INSERT INTO table (nom_colonne_1, nom_colonne_2, ...) VALUES ('valeur 1', 'valeur 2', ...)
  - Recherche de données
    - SELECT nom_colonnes FROM nom_table
  - Modification de données
    - UPDATE table SET nom_colonne_1 = 'nouvelle valeur' WHERE condition
  - Suppression de données

SQL

- Requête sur la base représentation
  - Création tables
    - CREATE TABLE nom_de_la_table

      (

        colonne1 type_donnees,

        colonne2 type_donnees,

        colonne3 type_donnees,

        colonne4 type_donnees

)
  - Insertion de données
    - INSERT INTO table VALUES ('valeur 1', 'valeur 2', …)
    - INSERT INTO table (nom_colonne_1, nom_colonne_2, ...) VALUES ('valeur 1', 'valeur 2', ...)
  - Recherche de données
    - SELECT nom_colonnes FROM nom_table
  - Modification de données
    - UPDATE table SET nom_colonne_1 = 'nouvelle valeur' WHERE condition
  - Suppression de données
    - DELETE FROM `table` WHERE condition

SQL

- Requête sur la base représentation
  - Création tables
    - CREATE TABLE nom_de_la_table

      (

        colonne1 type_donnees,

        colonne2 type_donnees,

        colonne3 type_donnees,

        colonne4 type_donnees

)
  - Insertion de données
    - INSERT INTO table VALUES ('valeur 1', 'valeur 2', …)
    - INSERT INTO table (nom_colonne_1, nom_colonne_2, ...) VALUES ('valeur 1', 'valeur 2', ...)
  - Recherche de données
    - SELECT nom_colonnes FROM nom_table
  - Modification d'un champ
    - UPDATE table SET nom_colonne_1 = 'nouvelle valeur' WHERE condition
  - Suppression d'une ligne
    - DELETE FROM `table` WHERE condition
  - Suppression d'une table ou base
    - DROP TABLE 'table' ;

Ajout des clés étrangères

- ALTER
  - ALTER TABLE nom_table

    instruction
- Ajout de clé
  - ALTER TABLE nom_table

    ADD CONSTRAINT nom_contrainte FOREIGN KEY (nom_cle_etrangere) REFERENCES nom_table_relation (nom_champ);
  - ATER TABLE nom_table

    ADD CONSTRAINT nom_contrainte FOREIGN KEY (nom_cle_etrangere) REFERENCES nom_table_relation (nom_champ) ON DELETE CASCADE;

Conventions

- Caractères ASCII
- Mots clés SQL en majuscules
- Nom des tables et des tuples en minuscules
- Nom de la table au pluriel
- Nom des tuples au singulier
- Nom composé séparé par un tiret_bas
- Nommer l’identifiant unique de la table par id
- Nommer la clé étrangère nom_table_id

Exercice

- Une ville dispose de salles de réunions dont l'ouverture des portes est commandée par un automate. Un digicode à l'extérieur permet de déverrouiller la porte.
- La ville souhaite mettre à disposition un formulaire de prise de rendez-vous permettant aux habitants et entreprises de réserver une salle.
- La réservation se fait à la journée et est payante. Un code (sur 4 chiffres) est généré à l'issue du paiement et envoyé par mail à l'acheteur.
- Le paiement peut se faire en ligne ou sur place

Faites le MCD suivant ce cahier des charges

AGREGATS

- COUNT()
- AVG()
- MAX()
- MIN()
- SUM()

- SELECT fonction(colonne) FROM table

Filtres

- WHERE
- GROUP BY
- HAVING
- Note
  - AS : Alias pour raccourcir le nom d’une table ou d’un champ

Filtres

- WHERE
  - SELECT nom_colonnes FROM nom_table WHERE condition
    - =, >, <, !=
  - AND
  - OR
- GROUP BY

- HAVING

Filtres

- WHERE
  - SELECT nom_colonnes FROM nom_table WHERE condition
- GROUP BY
  - SELECT colonne1, fonction(colonne2)
    FROM table
    GROUP BY colonne1
- HAVING

Filtres

- WHERE
  - SELECT nom_colonnes FROM nom_table WHERE condition
- GROUP BY
  - SELECT colonne1, fonction(colonne2)
    FROM table
    GROUP BY colonne1
- HAVING
  - SELECT colonne1, fonction(colonne2)
    FROM nom_table
    HAVING fonction(colonne2) operateur valeur

Filtres

- LIMIT
- ORDER BY

Filtres

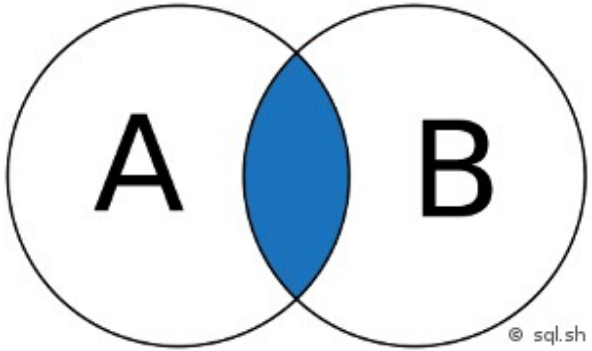
- LIMIT
  - SELECT *
    FROM table
    LIMIT 10
- ORDER BY

Filtres

- LIMIT
  - SELECT *
    FROM table
    LIMIT 10
- ORDER BY
  - SELECT colonne1, colonne2
    FROM table
    ORDER BY colonne1

- Par défaut, ORDER BY ordonne dans l’ordre chronologique (ou alphabétique selon le type de colonne1)

JOINTURES



- INNER JOIN (le plus courant. INNER peut être omis)
  - SELECT *

    FROM table1

    INNER JOIN table2 ON table1.id = table2.table1_id

- Retrouver les membres de l’orchestre « Daft Punk Revival »

IMPLICIT JOIN

ATTENTION : BAD PRACTICE

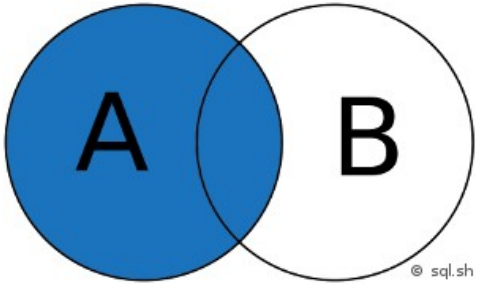
- Il est possible d'omettre le mot clé JOIN en utilisant une clause WHERE vérifiant l'égalité de la clé primaire et de la clé étrangère de la seconde table. Mal utilisée, cette requête peut sortir des résultats inattendus.

```
SELECT * FROM table1
INNER JOIN table2 ON table1.id = table2.fk_id
```

S'écrit

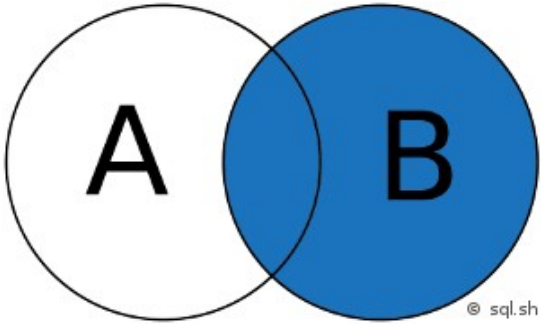
```
SELECT * FROM table1, table2
WHERE table1.id = table2.fk_id ;
```

LEFT JOIN



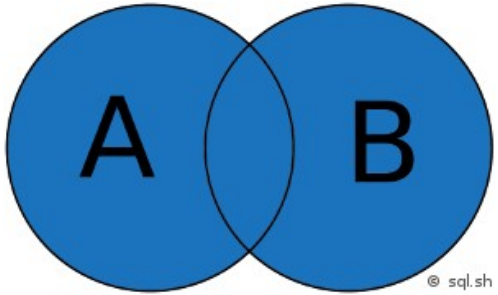
- SELECT *
  FROM table1
  LEFT JOIN table2 ON table1.id = table2.table1_id

RIGHT JOIN



- SELECT *
  FROM table1
  RIGHT JOIN table1 ON table1.id=table2.table1_id
- Equivalent à LEFT JOIN

FULL JOIN



- SELECT *
  FROM table1
  FULL JOIN table2 ON table1.id = table2.table1_id

Exercice

- Une auto-école souhaite organiser le planning des élèves et moniteurs

Réaliser le MCD et le MLD suivant ce cahier des charges

Fonctions sur les dates

- YEAR()
- MONTH()
- DAY()
- NOW()

- Afficher le mois de naissance du membre dont le nom est BANGALTER
- Afficher le nombre de membre né en novembre
- Afficher le mois de naissance et le nombre de membre né chaque mois par ordre de mois ascendant

Fonctions sur les chaînes de caractères

- Il existe un grand nombre de fonctions sur les chaînes de caractères. La majorité de ces fonctions se retrouvent dans les langages de programmation. Elles sont donc peu utilisées par les développeurs. La manipulation et le traitement des chaînes se faisant généralement après l'extraction des données (côté front).

- Faire une requête qui affiche le prénom et le nom des membres
- Faire une requête qui recherche si un membre Thomas BANGALTER existe dans la table

Index

- Un index permet d’accélérer la réponse d’une requête SELECT
  - Le moteur créé un hash du champ d’une colonne lui permettant de classer les lignes de la table
  - Les INDEX se positionnent sur les colonnes utilisées dans les clauses WHERE, GROUP BY ou ORDER BY principalement
- Un index ralentira les insertions ou les UPDATE sur la table (mise à jour du hash)

Explain

- EXPLAIN ‘requete SELECT’;
- Faire un EXPLAIN sur la requête permettant de classer les membres par ordre alphabétique
- SELECT * FROM membres ORDER BY nom ASC;

Explain - résultat

- Les colonnes à regarder
  - Table : nom de la table à laquelle la ligne fait référence
  - possible_keys : les index que le moteur pourrait utiliser pour accélérer l'exécution de la requête
  - Key : l'index qui a été utilisé pour la requête (NULL si aucun)
  - Rows : Nombre de lignes que le moteur va devoir analyser pour exécuter la requête
  - Extra : Si cette colonne retourne des résultats, c'est qu'une optimisation est possible

Explain

- ALTER TABLE table_name ADD INDEX `index_name` (`column_name`);#index_name peut être omis.
- Ajouter un index sur la colonne ‘nom’ de la table ‘membres’

- EXPLAIN ‘requete SELECT’;
- Refaire un EXPLAIN sur la requête permettant de classer les membres par ordre alphabétique

Les ORM

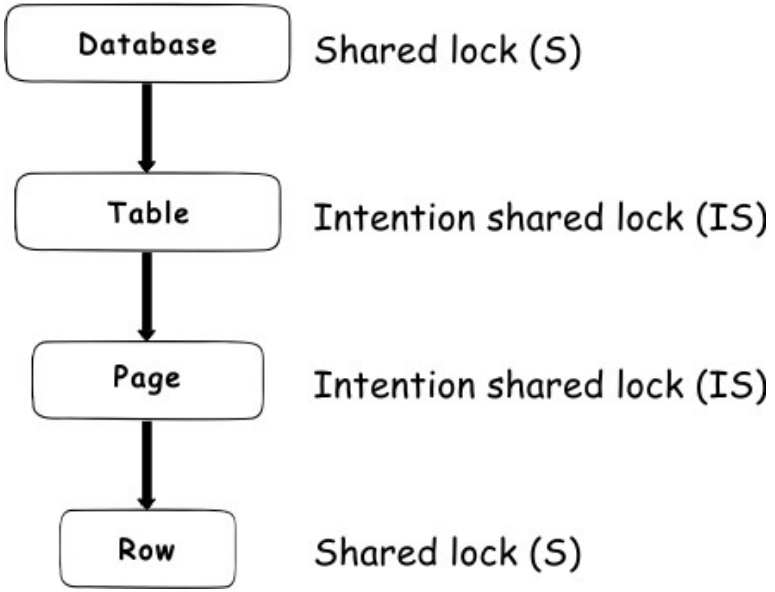
- Object Relational Mapping
  - L’ORM est une couche d’abstraction permettant de manipuler des objets et de ne plus utiliser des requêtes
- Avantages
  - Proche de l’OOP pour le développeur
  - C’est l’ORM qui gère les différences SQL entre les différentes SGBDR

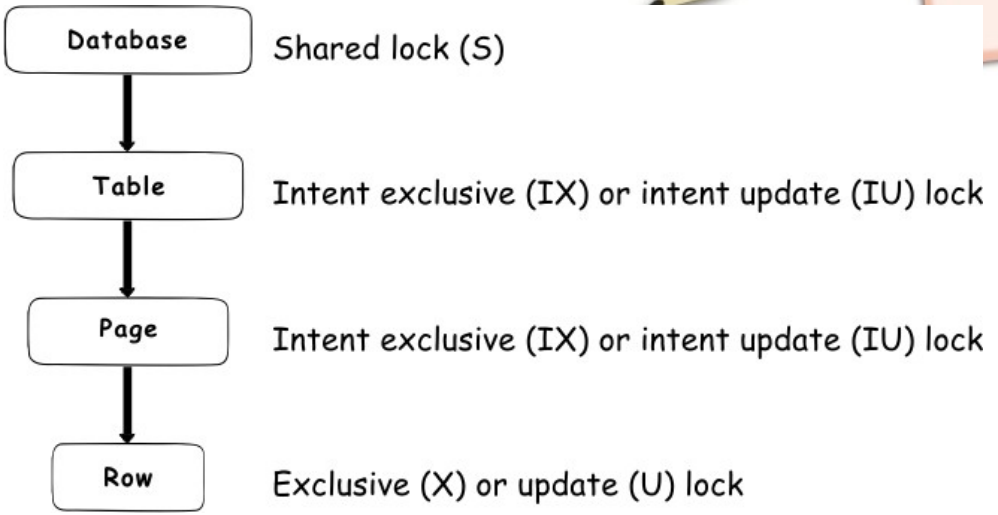
- Quelques ORMs
  - Doctrine
  - Eloquent
  - Hibernate

VUE

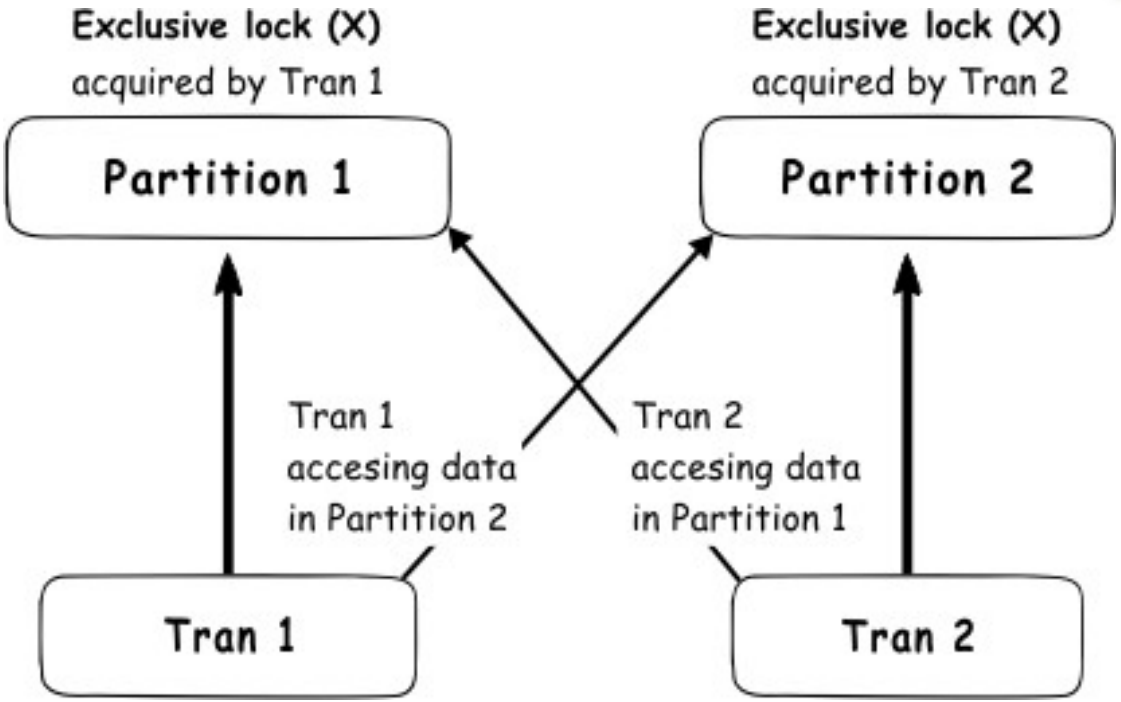
- Certaines requêtes peuvent mettre plusieurs heures à retourner les résultats recherchés
- CREATE VIEW permet de sauvegarder le résultat d’une requête et de l’utiliser ensuite
- CREATE VIEW nom_table_virtuelle AS requete
- Il est ensuite possible d’effectuer des recherches, des tris sur la table nom_table_virtuelle
- Les opérations CRUD effectuées sur la table source sont reportées sur la table vue
- Les opérations CRUD effectuées sur la table vue ne sont pas reportées sur la table source

Verrous (lock)





Dead lock


Exclusive lock (X)
acquired by Tran 1
Exclusive lock (X)
acquired by Tran 2
Partition 1
Partition 2
Tran 1
accesing data
in Partition 2
Tran 2
accesing data
in Partition 1
Tran 1
Tran 2

Exercice

- Une compagnie aérienne souhaite organiser les vols de ses avions avec ses pilotes et les passagers à bord
- 2 pilotes par avions
- Lignes desservies (aéroport de départ et d’arrivée)
- Avions de différentes tailles
- Chaque avion est identifié par un nom (Ex : « AF325 »)

Faire le MCD suivant ce cahier des charges

Exercice (suite)

- Liste des avions de la compagnie
- Nombre d’avions de la compagnie
- Trouver tous les vols réalisés par l’avion « AF325 »
- Trouver le nom des passagers du vol Paris Marseille du 30 août 2021 par ordre alphabétique
- Trouver le nom de l’avion qui a réalisé le vol « MP-4335 »

Procédures stockées

- Dans son expression la plus simple, une procédure stockée est une requête SQL nommée qui est stockée dans la base de donnée.

```
DELIMITER $$

CREATE PROCEDURE GetMembers()
BEGIN
    SELECT * FROM membre ORDER BY nom;
END$$

DELIMITER ;
```

- Les procédures stockées disposent d'un langage de contrôle simple (IF, LOOP, CASE, ...)

Procédures stockées

- Exécuter une procédure
  - SQL

```sql
CALL GetMembers();
```

  - Depuis un ORM

```php
public function getMembers()
{
    $allMembers = DB::select( query: 'CALL GetMembers();');
    return $allMembers;
}
```

- Voir les procédures stockées

```sql
SHOW PROCEDURE STATUS;
```

Procédures stockées

- Paramètres
  - IN

```
DELIMITER $$

CREATE PROCEDURE GetBandMembers(IN bandName VARCHAR(25))
BEGIN
    SELECT * FROM membres
    INNER JOIN orchestres ON membres.orchestre_id=orchestres.id
    WHERE orchestres.nom LIKE bandName;
END$$

DELIMITER ;
```

Procédures stockées

- Faire une procédure stockée permettant de retrouver le nom des membres par ordre alphabétique possédant une famille d'instrument (soit à corde, à vent ou percussion)
- Sur l'adresse mail stephane.pau.ext@eduservices.org
  - Envoyer une image (png) du MCD de la BDD
  - Les requêtes de création de la base
  - Les requêtes permettant d'insérer les données
  - La procédure stockée et le résultat
  - Faire une transaction permettant de passer un membre d'un orchestre à un autre. Les paramètres connus sont le nom du membre et des orchestres (ou l'id de l'orchestre)

Procédures stockées

- Paramètres
  - OUT (limité aux types simples)

```
DELIMITER $$

CREATE PROCEDURE CountMembers(IN bandName VARCHAR(25), OUT nombre INT)
BEGIN
SELECT COUNT(*) INTO nombre FROM membres
INNER JOIN orchestres ON membres.orchestres_id=orchestres.id
WHERE orchestres.nom LIKE'bandName';
END$$

DELIMITER ;
```

  - CALL PROCEDURE GetBandMembers(‘Taxi Girl’, @nombre) ;
  - SELECT @nombre ;

Procédures stockées

Avantages

- Limite le trafic réseau
- Sécurisation des accès (accès fins définis par le DBA)
- Utile pour les applications centrées sur les données (stats, reporting)

Inconvénients

- Compétences multiples (DBA+Développeur)
- Logique métier dans la BDD
- Difficile à maintenir
- Langage limité

Subquery

- Possibilité de faire une requête dans une autre requête.
- Exemple : Dans la table membres, retrouver les musiciens plus jeune que la moyenne

```
SELECT nom
FROM membres
WHERE date_naissance > (SELECT AVG(date_naissance )
                        FROM membres );
```

- Retrouver un membre qui n'appartient à aucun orchestre

```
SELECT membres.nom
FROM membres
WHERE membres.orchestre_id NOT IN (SELECT DISTINCT orchestres.id
                                   FROM orchestres);
```

Base de données opensource

- Volonté de libérer la donnée publique
- https://www.data.gouv.fr/fr/
- https://datanova.laposte.fr/explore/?sort=modified
- https://opendata.datainfogreffe.fr/page/offre-open-data/

BULK INSERT

- Il arrive souvent que la société gère initialement ses données avec un fichier excel
- La commande LOAD DATA INFILE permet de stocker directement les données depuis le fichier excel

```
LOAD DATA INFILE 'clients.csv'
INTO TABLE customers
fields terminated BY ','
optionally enclosed BY '"'
escaped BY '"'
lines terminated BY '\n'
ignore 1 lines;
```

TRIGGER

- Les TRIGGER sont des requêtes qui sont exécutées lorsqu'une modification C(R)UD va intervenir (BEFORE) ou est intervenue (AFTER) sur une table.
- Exemple : un bar souhaite enregistrée ses clients dans une base de données. Ses clients doivent avoir plus de 18 ans.

```
DELIMITER $$
CREATE TRIGGER  Check_age  BEFORE INSERT ON customers
FOR EACH ROW
BEGIN
IF NEW.age < 18 THEN
SET MESSAGE_TEXT = 'ERROR:
         AGE MUST BE AT LEAST 18 YEARS!';
END IF;
END$$

DELIMITER;
```

TRIGGER

- Les triggers sont remplacés par des observers dans les frameworks récents.
- Les évènements d’insertion, mise à jour ou suppression peuvent être suivi depuis le code

- Exemple avec Laravel
- php artisan make:observer UserObserver --model=Customer

TRIGGER

```php
<?php

namespace App\Observers;

use App\Models\Customer;

class UserObserver
{
    /**
     * Handle the User "created" event.
     *
     * @param  \App\Models\Customer  $customer
     * @return void
     */
    public function created(Customer $customer)
    {
        //
    }

    /**
     * Handle the User "updated" event.
     *
     * @param  \App\Models\Customer  $customer
     * @return void
     */
    public function updated(Customer $customer)
    {
        //
    }

    /**
     * Handle the User "deleted" event.
     *
     * @param  \App\Models\Customer  $customer
     * @return void
     */
    public function deleted(Customer $customer)
    {
        //
    }
}
```

TRIGGER

- Avantage : Avec les « *observers* » du framework, le traitement peut etre plus poussé que depuis le moteur de base de données. On peut par exemple envoyer un mail, lever une alerte,...

Sauvegarde des données

- Copie brute de la table à intervalle régulier
- mysqldump -u [utilisateur] -p –all-databases > dump.sql

- Réplication
- Nécessite 2 instances du serveur SQL. Un des serveurs est maître et le second est dit esclave. Les données sont écrites uniquement sur le serveur maître. Un fichier de log des opérations Create Update et Delete est générée. Le serveur esclave vient lire le fichier et répète les opérations effectuées. C’est une réplication asynchrone car il existe un léger délai pour la réplication (quelques milliscecondes à quelques secondes)

Exercice

- Un revendeur informatique (ordinateurs, smartphones,…) souhaite suivre son stock et sa valeur. Ses produits proviennent de différents fournisseurs. Il ne dispose que d’un seul entrepôt de stockage. Il vend des produits neufs et reconditionnés
- Les produits ont un prix d’achat, un prix de vente et un volume ou un poids.
- Un même produit peut être acheté auprès de différents fournisseurs.
- Ce commerçant suit ses ventes (et les factures) avec cette base.

Exercice

- Faire le MCD et le MPD (format image) en énonçant vos règles métier
- Écrire les requêtes SQL permettant permettant de générer la base
- Écrire le dictionnaire de données
- Ajouter le fournisseur Samsung (neuf), le fournisseur Dell (neuf) et le fournisseur GoodDeal (reconditionné)
- Ajouter des smartphones et ordinateurs (10 minimum)
- Écrire la requête d'une vente à un client d'un produit provenant du stock sous la forme d'une transaction
- Écrire la requête calculant le coût d'achat du stock
- Écrire la requête calculant la somme totale des achats d'un client sur l'année 2021
- Supprimer un produit du stock
- Comment améliorer la vitesse de la base de données (détail)

Exercice

- Une médiathèque souhaite gérer son fonds documentaire avec une base de données relationnelles. Elle dispose de livres, CD, DVD. Elle est constituée en association et les membres peuvent emprunter les documents mis à disposition. Les membres doivent s’acquitter d’une cotisation annuelle dépendante de leurs revenus. La médiathèque souhaite ouvrir la location à d’autres types de documents (carte postale, documents numériques, livres audio,…) à terme.
- Chaque document à une référence unique

Exercice

- Faire le MCD
- Écrire le dictionnaire de données avec les règles métiers
- Ajouter des membres (avec requete SQL INSERT)
- Ajouter des auteurs (avec LOAD DATA INFILE)
- Créer des emprunts
- Écrire la requête permettant de savoir quels documents sont empruntés

Rappel suite correction

- MCD
  - Indiquer les cardinalités
  - Pas de flèches sur les relations
  - Cohérence sur la notation (pluriel/singulier, français/anglais, Majuscule, minuscule)
- Relation m,n indique qu’il y aura une table de relation (pivot)
- Chaînage de plusieurs JOIN sur une même requete

Exercice

- Une ville située dans une zone minière souhaite suivre l'activité des concessions* minières qu'elle concède aux exploitants. Un concessionnaire peut gérer plusieurs carrières. Pour chaque carrière, la ville souhaite avoir une analyse d'impact sur l'eau, la faune et la flore. La concession dispose d'équipements pour le fonctionnement des carrières (tombereau, concasseur, trommel, séchoir,…). Pour des raisons administratives, sur une période donnée, un équipement est dédié à une carrière.

  Pour accélérer les travaux de minage, le concessionnaire effectue des tirs dans les mines à l'aide de différents explosifs (C4, TNT, dynamite,…) . Ces tirs ont une certaine puissance (mesurée en équivalent kg de TNT), un volume sonore (mesuré en dB) et une position à laquelle est déposée la charge (longitude, latitude, hauteur). La hauteur peut être négative si elle est faite en sous-sol. Afin de protéger la population, la ville souhaite avoir accès à ces informations pour protéger la population.

* Une concession est une entité juridique comme toute entreprise

Exercice

- Faire le MCD répondant aux règles métiers édictées sur la page précédente
- Faire la requête permettant de déterminer l'emplacement* des matériels à une date donnée.
- Comment simuler une requête paramétrée avec une procédure stockée ?
- Faire une procédure permettant de connaître les impacts de la carrière "Alun-7". Il est possible de sélectionner un des impacts.
- Lors de fortes charges, comment optimiser la base de données ?
- Faire une requête permettant de connaître le nombre de tirs à une date donnée pour chaque carrière

* L'emplacement consiste à savoir à quelle carrière le matériel est affecté

Merci