

GGMD - Exercices applicatifs 1

BDR : Fragmentation d'une base de données

UCBL - Département Informatique de Lyon 1 – 2023

L'objectif de ces exercices est de vous familiariser avec les principes des bases de données réparties. Il s'agit, d'une part, de distribuer une BD centralisée sur plusieurs sites, et d'autre part, d'offrir un accès unifié à ces données réparties pour l'utilisateur final. Nous commencerons également à nous intéresser aux aspects liés à l'interrogation de la base.

1 A propos des données

Nous considérons la base de données *insee_decès* dont le schéma des données est le suivant :

personne(idp, nom, prenom, datenaiss, lieunaiss, datedecès, lieudecès, age)

La relation *personne* stocke les informations relatives aux décès de français entre le 01/01/1970 et 31/12/2019. On dispose pour chaque personne du nom, de ses prénoms, de sa date et lieu de naissance, de sa date et lieu de décès et de son âge au moment du décès. Les lieux de naissance et de décès correspondent à un code commune (code INSEE).

region(reg, nom, cheflieu, zone)

La relation *region* stocke les régions de France (Métropole et DOM-TOM), avec :

- *reg* est une chaîne de caractères alphanumérique qui identifie une région,
- *nom* désigne le nom de la région,
- *cheflieu* référence le code de la commune qui est le chef-lieu de la région.
- *zone* désigne la zone géographique de rattachement de la région. Ces zones sont définies de la manière suivante :

Zone	Régions
1	Hauts-de-France, Normandie, Île-de-France, Grand-Est
2	Bretagne, Pays-de-la-Loire, Centre-Val de Loire, Bourgogne-Franche-Comté, Guadeloupe, Martinique, Guyane, La Réunion, Mayotte
3	Nouvelle-Aquitaine, Occitanie, Auvergne-Rhône-Alpes, Provence-Alpes-Côte-d'Azur, Corse

N.B. Pour des raisons de simplification, nous ne considérerons pas dans ces exercices les décès des personnes nées en France et mortes à l'étranger, ainsi que les personnes nées à l'étranger et mortes en France.

departement(dep, nom, #cheflieu, #reg)

La relation *departement* stocke les différents départements français, avec :

- *dep* est une chaîne de caractères alphanumérique qui représente le code identifiant le département,
- *nom* désigne le nom du département,

- *cheflieu* référence le code de la commune qui est le chef-lieu du département,
- *reg* référence la région de rattachement du département.

commune(com, nom, #dep)

La relation *commune* stocke les différentes communes de France telle qu'elles existent en 2022, avec :

- *com* une chaîne de caractères qui correspond au code INSEE de la commune (à ne pas confondre avec le code postal),
- *nom* désigne l'appellation de la commune (en 2022)
- *dep* référence le département de rattachement de la commune.

mairie(#codeInsee, cp, nomOrga, nomCom, email, tel, url, adresse, latitude, longitude, dateMaj)

La relation *mairie* stocke les informations issue de *data.gouv.fr* sur les mairies de France, avec :

- *codeInsee* correspond au code INSEE de la commune associé à la mairie,
- *cp* désigne le code postal de la commune
- *nomOrga* désigne l'appellation courante de la mairie
- *nomCom* désigne le nom de la commune,
- *email* désigne l'adresse mail de contact de la mairie
- *tel* désigne le numéro de téléphone de contact de la mairie
- *url* désigne l'adresse du site web de la mairie
- *adresse* désigne l'adresse postale de la mairie
- *latitude* et *longitude* désignent les coordonnées de géolocalisation de la mairie
- *dateMaj* désigne la date de dernière mise à jour des informations relative à la mairie.

2 Fragmentation de la base

Exercice 1 : Requête

L'objectif de cet exercice est de vous familiariser avec le modèle centralisé de la base de données insee_deces .

Soit la requête Q1 : "Donner les personnes (idp, nom, prenom) nées dans la région 'Auvergne-Rhône-Alpes' et décédées dans la région 'Pays de la Loire'".

1. Écrire Q1 en Algèbre Relationnelle.
2. Écrire Q1 en SQL.

Exercice 2 : Définition des fragments

L'objectif de l'exercice est d'analyser les contraintes proposées sous forme d'hypothèses pour en déduire une décomposition de la base de données.

On considère les sites de 'Paris', 'Nantes' et 'Marseille' auxquels on associera respectivement les numéros 1, 2 et 3.

On souhaite proposer une fragmentation de la base *insee_deces* sur les sites de 'Paris', 'Nantes' et 'Marseille' dans le respect des contraintes suivantes :

C1. Le site de 'Paris' stocke les régions de la zone 1. Le site de 'Nantes' stocke les régions de la zone 2 et le site de 'Marseille' stocke les régions de la zone 3.

C2. Chaque site gère les départements rattachés aux régions qu'il stocke.

C3. Chaque site gère les communes rattachées aux départements qu'il stocke.

C4. L'âge des personnes est stockées sur le site de 'Paris'.

C5. Chaque site gère les nom et prénoms, date de naissance et lieu de naissance des personnes nées dans une commune qu'il stocke.

C6. Chaque site gère les nom et prénoms, date de décès et lieu de décès des personnes décédées dans une commune qu'il stocke.

C7. Les données sur les mairies sont nécessaires sur tous les sites. Le site de Nantes sera en charge des mises à jour des informations susceptibles d'être actualisées (téléphone, email...)

1. A partir des contraintes, spécifier la manière dont vous envisagez de décomposer la base *insee_deces* (type de fragmentation...).
2. Définir algébriquement les différents fragments permettant d'effectuer une décomposition de la base *insee_deces* respectant les les contraintes **C1.** ... **C6.**.
3. Votre fragmentation est-elle *correcte*? Pourquoi?

3 Accès unifié à une base de données répartie

Exercice 3 : Construction d'une vue globale sur les données

L'objectif de cet exercice est de montrer le caractère restructurable de votre décomposition.

1. Définir algébriquement les vues nécessaires pour accéder (selon le schéma initial de la base de données) aux données des fragments.

4 Interrogation de la base de données répartie

Exercice 4 : Réécriture de requêtes

L'objectif de cet exercice est de montrer le processus de simplification algébrique des requêtes inter-sites.

1. Donner un arbre algébrique A_1 exprimant la requête Q1 en fonction des vues globales définies dans l'exercice précédent.
2. Donner un arbre algébrique A'_1 exprimant la requête Q1 en fonction de vos fragments.
3. Quelles simplifications proposez-vous sur A'_1 pour éviter des calculs inutiles? Donner l'arbre algébrique A''_1 résultant de ces simplifications.

ANNEXE

Rappels sur les différents types de fragmentation

Soient les relations suivantes : $T(\underline{A}, B, C, D, E)$ et $U(\underline{X}, Y, Z)$.

Type de fragmentation	Intérêt	Exemple de définition algébrique
Verticale	Isoler certains attributs dans des relations différentes	$F_{11} = \Pi_{A,B,C}(T)$ $F_{12} = \Pi_{A,D,E}(T)$
Horizontale	Isoler des tuples dans des relations différentes	$F_{21} = \sigma_{B>10}(T)$ $F_{22} = \sigma_{B\leq 10}(T)$
Horizontale dérivée	Isoler des tuples dans des relations différentes en fonction d'autres données fragmentées	$F_{3i} = U \bowtie_{X=C} (F_{2i}), i \in \{1, 2\}$.
Fragmentation mixte	Isoler des fragments de tuples résultant d'une combinaison de fragmentation dans des relations différentes	$F_{4i} = \Pi_{X,Y}(U) \bowtie_{X=C} (F_{2i}), i \in \{1, 2\}$ $F_{4'i} = \Pi_{X,Z}(U) \bowtie_{X=C} (F_{2i}), i \in \{1, 2\}$

Rappels sur les équivalences algébriques

Principe	Exemple algébrique
Commutativité des opérateurs binaires	$T_1 \times T_2 \equiv T_2 \times T_1$ $T_1 \bowtie T_2 \equiv T_2 \bowtie T_1$ $T_1 \cup T_2 \equiv T_2 \cup T_1$ $T_1 \cap T_2 \equiv T_2 \cap T_1$
Associativité des opérateurs binaires	$(T_1 \times T_2) \times T_3 \equiv T_1 \times (T_2 \times T_3)$ $(T_1 \bowtie T_2) \bowtie T_3 \equiv T_1 \bowtie (T_2 \bowtie T_3)$ $(T_1 \cup T_2) \cup T_3 \equiv T_1 \cup (T_2 \cup T_3)$ $(T_1 \cap T_2) \cap T_3 \equiv T_1 \cap (T_2 \cap T_3)$
Distributivité de la sélection sur les opérateurs binaires	Soit $pred_i$ un prédicat ou combinaison de prédicats portant sur les attributs de T_i , on a : $\sigma_{pred_1}(T_1 \times T_2) \equiv \sigma_{pred_1}(T_1) \times T_2$ $\sigma_{pred_1}(T_1 \bowtie T_2) \equiv \sigma_{pred_1}(T_1) \bowtie T_2$ Soit $pred_{12}$ un prédicat ou combinaison de prédicats portant sur les attributs communs à T_1 et T_2 , on a : $\sigma_{pred_{12}}(T_1 \cup T_2) \equiv \sigma_{pred_{12}}(T_1) \cup \sigma_{pred_{12}}(T_2)$
Distributivité de la projection sur les opérateurs binaires	Soit A_i l'ensemble des attributs de T_i , On pose $A = A'_1 \cup A'_2$ avec $A'_1 \subseteq A_1$ et $A'_2 \subseteq A_2$ et $I \subseteq A'_1 \cap A'_2$ un sous-ensemble d'attributs communs à T_1 et T_2 , on a : $\Pi_A(T_1 \times T_2) \equiv \Pi_{A'_1}(T_1) \times \Pi_{A'_2}(T_2)$ $\Pi_A(T_1 \bowtie_I T_2) \equiv \Pi_{A'_1}(T_1) \bowtie_I \Pi_{A'_2}(T_2)$ $\Pi_A(T_1 \cup T_2) \equiv \Pi_A(T_1) \cup \Pi_A(T_2)$
Idempotence des opérateurs unaires	$\forall A \subseteq A', \Pi_A(\Pi_{A'}(T)) \equiv \Pi_A(T)$ $\sigma_{pred_1}(\sigma_{pred_2}(T)) \equiv \sigma_{pred_1 \wedge pred_2}(T)$

Focus utile :	La sélection appliquée à une union équivaut à l'union des sélections	$\sigma_{condition}(\cup_{i=1}^n T_i) \equiv \cup_{i=1}^n \sigma_{condition}(T_i)$
	La jointure appliquée à une union équivaut à l'union des jointures	$T_1 \bowtie (\cup_{i=2}^n T_i) \equiv \cup_{i=2}^n (T_1 \bowtie T_i)$