

SCI6052 Information documentaire numérique

Cours 4

Fonction recherche de fichiers de Windows, recherche en texte intégral (suite). Antidictionnaire. Logiciels de recherche en texte intégral (LRTI); démonstration de NatQuest Pro.

**Systemes de gestion de bases de données (SGBD) textuels (début).
Champs, types de données, validation; index-mots et index-occurrences; dictionnaire de données.**

Au programme aujourd'hui

- ➔ **Retour sur Diagrammes de Venn + Arbres renversés**
 - › Retour sur les devoirs
 - › Opérateurs booléens dans la « vraie vie »
- ➔ **Recherche de fichiers en texte intégral (*suite*)**
 - › Logiciels spécialisés pour la recherche en texte intégral (LRTI)
 - Exemple de NatQuest Pro (Agir)
 - › Antidictionnaire, opérateurs de distance
- ➔ **Systemes de gestion de bases de données textuels (SGBD) (*introduction*)**
 - › SGBD, SGBD textuels
 - › Structure des données & Dictionnaire de données
 - › Validation à la saisie
 - › Index-mots & index-occurrences
 - › Comparaison avec LRTI

Cours 4 – Objectifs visés, matériel associé et évaluation

Examen intra



OG1 Comprendre les principaux volets de la gestion d'information documentaire numérique

OS f) *Expliciter les principes de base de la recherche d'information textuelle*

Compétences à développer :

- Être en mesure de définir ce qu'est un antidictionnaire et de décrire les principaux points à considérer dans sa préparation (acétates 11-13)
- Comprendre à quoi servent les opérateurs de distance (adjacence, distance) (acétate 16)

OS g) *Expliciter les avantages et les désavantages des bases de données textuelles pour la gestion d'information documentaire numérique*

Compétences à développer :

- Pouvoir définir ce qu'est un SGBD textuel (acétates 22-24)
- Comprendre la différence entre un index-mots et un index-occurrences ainsi que leurs avantages/désavantages respectifs entre autres par rapport à la recherche d'expression et la recherche d'occurrences exactes (acétates 25-29, TP4)
- Être en mesure d'expliquer le contexte d'utilisation d'un SGBD textuel par rapport à un LRTI et à la recherche dans un système d'exploitation (acétate 37)

OG2 Mettre sur pied des systèmes de gestion d'information documentaire numérique au moyen d'outils représentatifs de la réalité des milieux

OS f) *Rechercher efficacement de l'information dans des données structurées selon différents critères*

Compétences à développer :

- Être en mesure d'effectuer des requêtes (élémentaires et complexes) utilisant les différents opérateurs de recherche d'un SGBD textuel (DB/TextWorks) (TP4)

Cours 4 – Objectifs visés et activités associées

Mise en application



OG1 Comprendre les principaux volets de la gestion d'information documentaire numérique

OS f) Expliciter les principes de base de la recherche d'information textuelle

OG2 Mettre sur pied des systèmes de gestion d'information documentaire numérique au moyen d'outils représentatifs de la réalité des milieux

OS e) Rechercher efficacement de l'information dans des données structurées selon différents critères

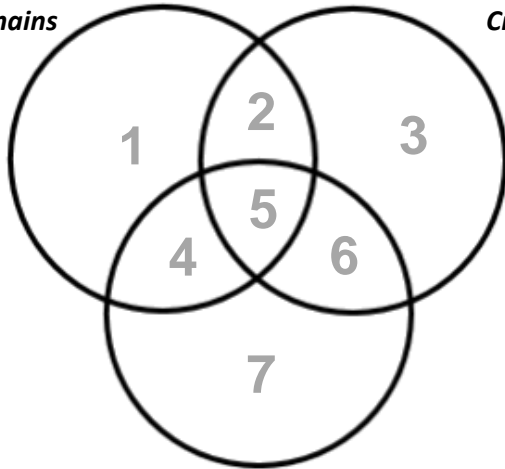
Activités :

- TP4
- Exercices sur la recherche dans DB/TextWorks (StudiUM) (*après cours* 5) [<https://studium.umontreal.ca/mod/quiz/view.php?id=2369356>]

Devoir 1 : Danse de l'amateur de booléens

Frappe des mains

Crie Hourra!

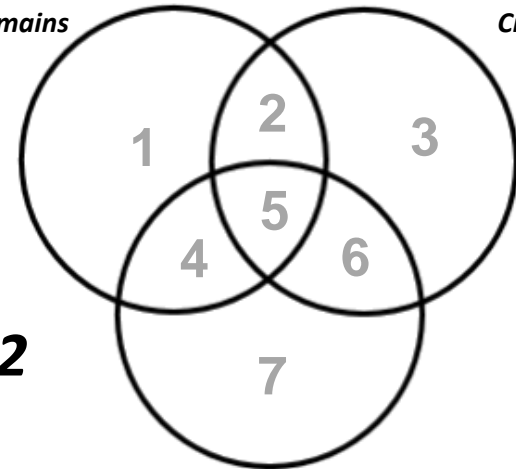


#1

Tape des pieds

Frappe des mains

Crie Hourra!

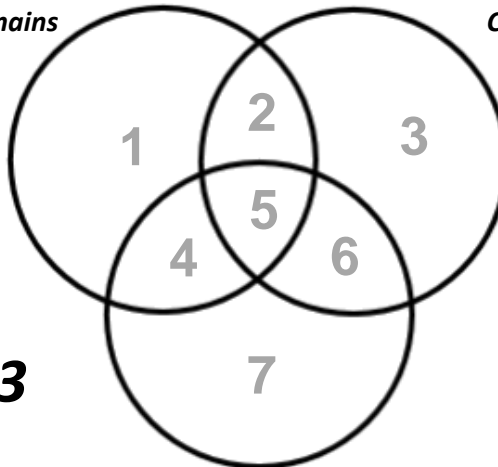


#2

Tape des pieds

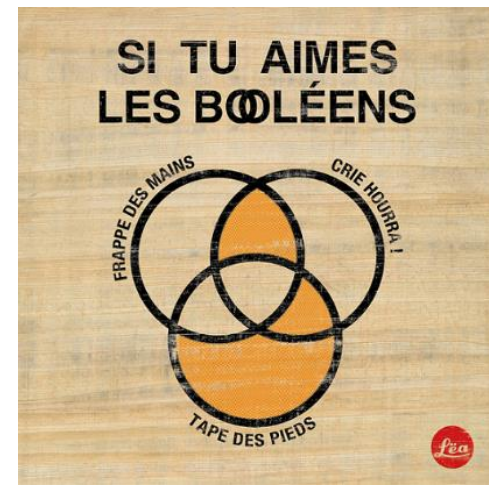
Frappe des mains

Crie Hourra!

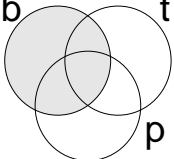


#3

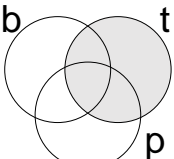
Tape des pieds



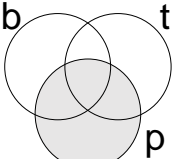
begonia SAUF ((tulipe OU pensée) SAUF ((begonia ET tulipe) ET pensée))



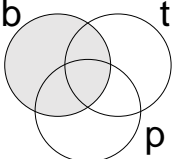
begonia



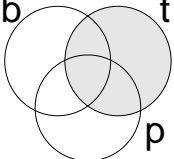
tulipe



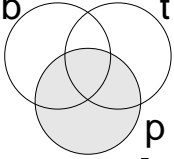
pensée



begonia



tulipe



pensée

Opérateurs booléens

Aussi dans la « vraie vie »

- ➔ Les expressions avec opérateurs booléens surviennent aussi dans la « vraie vie »...
- ➔ Elles sont parfois complexes et difficiles à comprendre
 - › Parle-t-on « booléen » ou « français »?
- ➔ Encore plus difficiles à exprimer sans ambiguïté « juste en texte »
- ➔ Les arbres renversés peuvent aider à exprimer ces expressions sans ambiguïté
- ➔ ... et à les expliquer et/ou comprendre

Recherche de fichiers en texte intégral

Outils spécialisés LRTI : Introduction

- ➔ LRTI = Logiciels de recherche de fichiers en texte intégral
- ➔ Outils spécialement conçus pour la recherche d'information textuelle
 - Permettent la recherche dans le contenu de corpus de documents textuels numériques **existants**
 - Peuvent même dans certains cas exploiter la **structure interne** des fichiers lorsqu'elle est explicite
 - Offrent un **langage d'interrogation élaboré** plus puissant que celui offert par exemple par les outils intégrés au système d'exploitation comme Windows Search
 - Donnent plus de contrôle au niveau de l'indexation par le biais, entre autres, de **l'antidictionnaire**
- ➔ Exemples de LRTI : NatQuest Pro (Agir), dtSearch (dtSearch)

Recherche de fichiers en texte intégral

Outils spécialisés : NatQuest Pro [1/7]

➔ NatQuest Pro (Agir) = LRTI

- Recherche de fichiers en texte intégral mais peut aussi exploiter la **structure interne** des fichiers pour les découper en documents et/ou en champs
- Permet de **visualiser directement** les fichiers
- Peut traiter **plusieurs dizaines de formats** de fichier différents, dont la plupart des formats les plus populaires (Word, PDF, OpenOffice, etc.)
- Recherche **indexée uniquement**, pour des raisons d'efficacité sur de grands volumes de données
- Contenu textuel = **uniquement** contenu des fichiers comme tel, aucune métadonnée exploitable
- Fonctionne par corpus (**base de données**) et non par dossier

Recherche de fichiers en texte intégral

Outils spécialisés : NatQuest Pro [2/7]

➔ Indexation du corpus

➤ **Index** visualisable lors de la recherche

➤ **Antidictionnaire**

- Modifiable dans le Bloc-notes
 - Peut être différent pour chaque base
 - Fichier *<nom-de-la-base>.NQS*
 - Fichier texte ISO-8859-1 (ANSI)
- Signes diacritiques (accents, cédilles, ponctuation, etc.) et casse **ignorés** (peut être désactivé)

Recherche de fichiers en texte intégral

Antidictionnaire : Suite [1/3]

➔ *Principes de base*

- › Liste de mots vides (*stop words*) = mots considérés comme **vides de sens** et donc **inutiles pour la recherche** (par ex., pronoms, articles, adverbes)
- › À l'indexation, chaque mot est comparé à l'antidictionnaire : s'il y est, il n'est pas versé dans le fichier inversé (index)

➔ **Définition de son contenu n'est pas un geste anodin!**

- › Permet de *réduire* la taille du fichier inversé (index)
- › Peut aussi être utilisé pour « *traiter* » les requêtes
- › Peut avoir un impact sur *l'efficacité* du repérage
 - Par exemple, si on inclut un mot significatif dans l'antidictionnaire, il sera éliminé de l'index et donc « invisible » à la recherche
- › Parfois nécessaire de le *modifier* selon le contexte, par ex.
 - Selon le domaine : son (adjectif possessif), son (bruit), son (céréale)
 - Selon la langue : the (article défini en anglais), thé (boisson)

Recherche de fichiers en texte intégral

Antidictionnaire : Suite [2/3]

- ➔ S'il est **trop vide**, risque de bruit
 - › L'index est encombré de mots inutiles qui peuvent causer du bruit dans certains outils

- ➔ S'il est **trop plein**, risque de silence, dû à
 - › Problèmes d'homographie
 - › Problèmes de polysémie
 - › Problèmes aggravés par l'élimination des signes diacritiques

Recherche de fichiers en texte intégral

Antidictionnaire : Suite [3/3]

- ➔ **Homographie** et **polysémie** affectent la recherche en général, pas seulement les choix de mots vides
- ➔ Problèmes d'**homographie**
 - Deux mots différents **s'écrivent de la même façon**, mais un seul nous intéresse
 - thé (breuvage) vs the (article anglais)
 - lis (verbe lire) vs lis (fleur) vs LIS (Library & Information Studies)
- ➔ Problèmes de **polysémie**
 - Un même mot **a plusieurs sens**, dont un seul nous intéresse
 - table (de logarithmes) vs table (d'opération)
 - travail (en relations industrielles) vs travail (en obstétrique)

Recherche de fichiers en texte intégral

Outils spécialisés : NatQuest Pro [3/7]

➔ Interrogation [1/2]

› **Troncature** (*) n'importe où dans un terme; remplace 0 à n caractères

- `informa*tion` repêche `information` et `informatisation`

› **Masque**¹ (?) n'importe où dans un terme; remplace 1 car.

- `ent??prise` repêche `entreprise` et `enterprise`

› Opérateurs **booléens**

- OU : [espace] (implicite)
- ET : « & »
- SAUF : « - » (espace, tiret, espace)

› **Pluriels simples**¹ (c'est-à-dire pluriels réguliers) automatiques (peut être désactivé en recherche)

- « chat » recherche aussi « chats » (et vice-versa)
- « simple » car, en fait, cherche simplement le mot avec ou sans « s »
 - Si on cherche « dix », repèrerait théoriquement « dixs »
 - Si on cherche « chorus », repèrerait théoriquement « choru »

¹ Différences par rapport à la recherche avec Windows Search

Recherche de fichiers en texte intégral

Outils spécialisés : NatQuest Pro [4/7]

➔ Interrogation [2/2]

> Recherche d'expression : guillemets

- *Particularité* : les mots vides dans l'expression cherchée ne comptent pas et le pluriel simple s'applique aux mots de l'expression
- "base donnée" repêche bases de données ainsi que base de données et base sur la donnée

> Opérateurs de **distance**¹

- *nw* (où *n* = nombre maximum de mots significatifs entre les termes)
- *adj* (équivalent à *0w* et à la recherche d'expression)
- *Particularité*: tiennent compte de l'ordre mais pas des mots vides
 - Information 3w science trouvera information à l'École de bibliothéconomie et des sciences

> Requêtes complexes : utilisation des **parenthèses**

> Interrogation en **langue naturelle** possible

¹ Différences par rapport à la recherche avec Windows Search

Recherche de fichiers en texte intégral

Autres opérateurs : opérateurs de distance

- ➔ Surtout utilisés pour la **recherche en texte intégral dans le plein texte des documents**
- ➔ **Adjacence** (adj)
 - › Repère des mots « immédiatement » juxtaposés. Peut ou non tenir compte de l'ordre (dépend du logiciel)
 - › `information ADJ numérique` peut repérer, selon les logiciels, `information numérique ainsi que numérique, information`
- ➔ **Distance** (*nM*)
 - › Spécifie la distance (en mots) maximale entre deux termes de recherche (ne tient généralement pas compte de l'ordre mais ce n'est pas toujours le cas)
 - › La manière de compter les mots peut varier selon les logiciels ainsi que le fait de retrouver ou non les mots dans un certain ordre
 - › son `3M farine` pourrait retrouver `farine de son` ainsi que son `frère aime la farine`

Recherche de fichiers en texte intégral

Outils spécialisés : NatQuest Pro [5/7]

➔ Structure interne des fichiers exploitable [1/2]

➤ Exemple de fichier avec une structure interne explicite

Délimiteur de champ



```
notices.txt - Bloc-notes
Fichier Edition Format Affichage ?
VN: Ovid Technologies
AU: McCaffrey, Mike.
TI: Common Look and Feel for the Internet 2.0.
SO: DttP. v. 37 no2 (Summer 2009) p. 10-11.
SH: Web 20
      Internet -- Standards
CP: United States
YR: 2009
VN: Ovid Technologies
AU: Glenn, Valerie D.
TI: Government and Web 2.0.
SO: DttP. v. 37 no2 (Summer 2009) p. 13-14, 17.
SH: Web 20
      Information retrieval -- Social aspects
CP: United States
YR: 2009
```

Délimiteur de notice



1 notice = 1 document

*Fichier de notices
bibliographiques extraites
d'une base de données*

Recherche de fichiers en texte intégral

Outils spécialisés : NatQuest Pro [6/7]

➔ Structure interne des fichiers exploitable [2/2]

> Possibilité de **subdiviser** les fichiers en « **documents** »¹

- On doit alors « déclarer » un séparateur de document à NatQuest avant l'indexation
 - Dans l'exemple précédent : `VN: Ovid Technologies`
- Les résultats retournent en ce cas des documents et non des fichiers

> Possibilité de **délimiter** les documents par **champ**¹

- Si les documents contiennent des délimiteurs de champs, on peut les « déclarer » à NatQuest avant l'indexation
 - Par exemple, dans l'exemple précédent : `TI, SO, SH`
- Permet la recherche dans un champ précis (par exemple `:ti:web` identifie les documents où « web » se retrouve dans un champ « titre »)
- Opérateurs d'intervalle (par exemple `1985@2005`) et de comparaison (`<` `>` `<=` `>=`) pour les champs à valeur numérique

¹ Différences par rapport à la recherche avec Windows Search

Recherche de fichiers en texte intégral

Outils spécialisés : NatQuest Pro [7/7]

- ➔ Présentation des résultats de recherche
 - **Liste de documents** (un document correspond à un fichier si aucune subdivision n'a été spécifiée)
 - Documents triés par **pertinence estimée**
 - Algorithme de similitude basé entre autres sur l'ordre des mots et leur proximité ainsi que sur la présence du plus grand nombre de termes
 - Termes recherchés **mis en évidence**¹
 - Permet de **naviguer**¹
 - d'un document à l'autre
 - d'une occurrence à l'autre à l'intérieur d'un même document

¹ Différences par rapport à la recherche avec Windows Search

Recherche de fichiers en texte intégral

Outils spécialisés LRTI : Conclusion [1/2]

- ➔ LRTI comme NatQuest Pro sont des outils spécialisés utiles pour la recherche en texte intégral dans des :
 - **Documents textuels qui existent déjà** (reconnaissent plusieurs formats)
 - Ne permettent pas de les modifier toutefois
 - **Documents textuels non structurés** en format numérique
 - **Documents textuels structurés** en format numérique
 - Fichiers qui possèdent une structure explicite par section et même par champ
 - La recherche dans les champs possède toutefois certaines limites (impossible de distinguer les occurrences multiples)
 - **Corpus volumineux**
 - La recherche indexée permet des recherches efficaces

Recherche de fichiers en texte intégral

Outils spécialisés LRTI : Conclusion [2/2]

- ➔ Corpus doit être **relativement stable**
 - › Particulièrement si on doit procéder à une nouvelle indexation chaque fois qu'on ajoute, supprime ou modifie un document

- ➔ Langage d'interrogation puissant
 - › Opérateurs booléens
 - › Opérateurs de distance
 - › Recherche d'expression
 - › Troncature et masque
 - › Recherche d'intervalle
 - › Opérateurs de comparaison

- ➔ Contrôle sur l'index par le biais de l'antidictionnaire

Système de gestion de bases de données (SGBD)

- ➔ « **Système composé d'un dictionnaire de données et d'un langage de requête qui permet la définition, la création et la mise à jour des informations d'une base de données de même que l'accès à celles-ci.** » (O'Leary et O'Leary, 2014, p. 282)
- ➔ Trois grands groupes de fonctions
 - › **Création** d'une base de données
 - › **Mises-à-jour** de la base : ajouts, suppressions, modifications (pour les trois : à la pièce et en lot)
 - › **Recherche** d'information à l'intérieur d'une base de données : extractions, rapports
- ➔ Différents modèles de données (c'est-à-dire de manière de structurer des données et d'interagir avec ces dernières) existent dont
 - › textuel
 - › relationnel

Systèmes de gestion de bases de données (SGBD)

SGBD textuels

- ➔ Peuvent être utilisés pour créer
 - Bases de **données bibliographiques**
 - Ex. : catalogue des documents disponibles dans une petite bibliothèque ou un centre de documentation
 - Bases de **données en texte intégral**
 - Subdivision par champs : différents champs pour données bibliographiques + un champ pour texte intégral
 - Bases de **données factuelles**
 - Etc.
- ➔ Exemple de SGBD textuel : DB/TextWorks

Systèmes de gestion de bases de données (SGBD)

Modèle de données

➔ Modèle textuel

➤ Exemple

Une ligne = un enregistrement/fiche
 Une colonne = un champ
 Une cellule = données (zéro, une ou plusieurs occurrences*)

<i>Auteur</i>	<i>Titre</i>	<i>Année</i>	<i>Descripteurs</i>
Marcoux, Yves	Les SGBD textuels	2007	bases de données informatique documentaire
O'Leary, Timothy	Éléments d'informatique	2006	
Deweze, André	Informatique documentaire	1993	informatique documentaire
Jacquesson, Alain	L'informatisation des bibliothèques	1995	informatique documentaire bibliothèques

- Appelé modèle *textuel* car...
 - données = mots ou chiffres (principalement)
 - opérateurs « textuels » très développés
 - opérations mathématiques peu développées

* *occurrence* = entrée distincte
 Par ex., il y a deux occurrences dans le champ
 Descripteurs du premier enregistrement :
 « bases de données » et « informatique
 documentaire »

SGBD textuels

Indexation [1/2]

➔ Recherche indexée (DB/TextWorks)

➔ Deux types d'index

➤ **Index-mots** (*word-index*)

- Chaque mot présent dans le champ est inscrit dans l'index (sauf* les mots de l'antidictionnaire) avec sa position
- *Note* : c'est le seul type d'index existant dans les LRTI et dans les outils de recherche intégrés aux systèmes d'exploitation

➤ **Index-occurrences** (*term-index*)

- Chaque occurrence complète du champ est inscrite dans l'index avec sa position
 - Les mots de l'antidictionnaire ne sont pas enlevés, mais les articles initiaux et la ponctuation le sont*

SGBD textuels

Indexation [2/2]

Index-mots

mot (no doc, no occ., no mot)*

** sans les mots vides*

bibliotheques ... (1, 1, 4)

(1, 2, 1)

(2, 1, 2)

gestion (1, 1, 3)

informatique (2, 1, 1)

integres (1, 1, 2)

scolaires (1, 2, 2)

systemes (1, 1, 1)

No doc	Descripteurs
1	Systèmes intégrés de gestion de bibliothèques Les bibliothèques scolaires
2	Informatique et bibliothèques

Index-occurrences

mot (no doc, no occurrence)

bibliotheques scolaires (1, 2)

informatique et bibliotheques (2, 1)

systemes integres de gestion de

bibliotheques (1, 1)

SGBD textuels

Champ à occurrences multiples & index-occurrences [1/3]

➔ Permet de faire des recherches **impossibles** autrement

Exemple 1

Besoin : Je cherche des documents sur les sciences de l'information

Environnements de recherche : (1) NatQuest Pro (recherche « structurée » dans un fichier texte) et (2) DB/TextWorks (base de données)

Dans LRTI (NatQuest Pro)

nodoc: 13
subject: information science
nodoc: 46
subject: secret information; science fiction
nodoc: 78
subject: information science; science fiction; secret information

Requête

:subject: "information science"

Résultats → 13, 46, 78 (46 = bruit)

car ne reconnaît pas les occurrences multiples dans un champ

Dans SGBD textuel (DB/TextWorks)

<i>nodoc</i>	<i>subject</i>
13	information science
46	secret information science fiction
78	information science science fiction secret information

Requête

subject ct information science

Résultats → 13, 78

car l'opérateur « ct » restreint la recherche à l'intérieur des occurrences

SGBD textuels

Champ à occurrences multiples & index-occurrences [2/3]**Exemple 2**

Besoin : Je cherche des documents *généraux* sur les bibliothèques, mais pas sur un type de bibliothèque en particulier

Environnements de recherche : (1) NatQuest Pro (recherche « structurée » dans un fichier texte) et

(2) DB/TextWorks (base de données)

Dans LRTI (NatQuest Pro)

nodoc: 2

descripteurs: bibliothèques publiques

nodoc: 26

descripteurs: bibliothèques; dépenses publiques

nodoc: 147

descripteurs: aménagement; bibliothèques

Requête

:descripteurs:bibliothèques

Résultats → **2, 26, 147** (2 = bruit)

**:descripteurs:bibliothèques -
publiques**

Résultats → **147** (manque 26 = silence)

Dans SGBD textuel (DB/TextWorks)

<i>nodoc</i>	<i>descripteurs</i>
2	bibliothèques publiques
26	bibliothèques dépenses publiques
147	aménagement bibliothèques

Requête

descripteurs = bibliothèques

Résultats → **26, 147**

car l'opérateur « = » permet de
trouver des occurrences complètes

SGBD textuels

Champ à occurrences multiples & index-occurrences [3/3]

- ➔ En d'autres mots, occurrences multiples et index-occurrences permettent de
 - **Cloisonner** les occurrences (*exemple 1*)
 - Recherche d'expression se fait **uniquement** à l'intérieur d'une occurrence (et non simplement d'un champ, l'expression pouvant alors chevaucher plus d'une occurrence)
 - Rechercher des **occurrences exactes** (*exemple 2*)
 - Recherche une occurrence entière
 - Ainsi, ne repêche pas les fiches où l'expression recherchée constitue un sous-ensemble d'une occurrence
 - Permettent ainsi **d'éliminer du bruit** et donc **d'augmenter la précision** d'une recherche ainsi **que d'éliminer du silence** pour **améliorer le rappel**

SGBD textuels

Structure des données

➔ **Décisions** à prendre au moment de définir la structure d'une base dans un SGBD textuel

› Pour chaque champ

- **Type** de champ?

- Textuel, numérique (automatique ou non), date (automatique ou non), etc.

- **Indexé** ou non?

- **Si indexé**

- Type d'index (index-mots et/ou index-occurrences)?
- Si index-mots : antidictionnaire appliqué ou non?
- Si index-occurrences : articles initiaux ignorés ou non?

SGBD textuels

Dictionnaire de données [1/3]

- ➔ Toutes les décisions concernant la base de données sont consignées dans un document à l'intention des personnes : le **dictionnaire de données** de la base
 - « description de la structure des données utilisées dans la base de données. Il définit le nom utilisé pour désigner un champ en particulier et le type de données que contient ce champ (alphabétique, numérique ou autre). Il peut en préciser la taille en nombre de caractères. C'est aussi dans le dictionnaire de données qu'on peut spécifier certaines contraintes sur les valeurs qu'un champ peut prendre afin d'en assurer la cohérence. » (O'Leary et O'Leary, 2014, p. 222)
 - *Exemple* : Dictionnaire des données de la base de données INFORM dans le cahier des protocoles, TP4 – Annexe 3

SGBD textuels

Dictionnaire de données [2/3]

➔ Contenu du dictionnaire [1/2]

➤ Information générale et technique sur la base de données

- Brève présentation de la base
- Logiciel utilisé
- Liste des articles initiaux et des mots de l'antidictionnaire
- Types de documents traités
- etc.

➤ Définition de la **structure** (en fonction du logiciel/système sélectionné)

SGBD textuels

Dictionnaire de données [3/3]

➔ Contenu du dictionnaire [2/2]

› Information détaillée sur chaque **champ**

- Nom du champ
- Type de contenu
- Fonctions de validation
- Caractérisation des entrées
- Règles d'écriture
- Nombre de caractères maximal
- Nombre d'occurrences permis
- Responsabilité du contenu
- Liste de contenus permis (gérée par le logiciel ou non)
- Notes
- Etc.

SGBD textuels

Saisie des données [1/3]

⇒ Bordereaux pour la saisie des données

- On peut définir différents bordereaux en fonction des besoins
 - *Ex.* : bordereau pour fiches d'articles de périodiques et bordereau pour fiches de monographies dans une base de données bibliographiques

⇒ Paramètres de **validation**

- Fonctions permettant d'exercer un **certain contrôle** sur la saisie des données
 - *Ex.* : S'assurer de se conformer à un certain format de date
- Pour chaque champ on peut inclure un ou plusieurs paramètres de validation

SGBD textuels

Saisie des données [2/3]

➔ Paramètres de validation [1/2]

Type de validation	Explication & Exemple dans DB/TextWorks
Par masque ou par intervalle <i>[pour chaque occurrence individuellement]</i>	<ul style="list-style-type: none"> Ex. : Masque pour un numéro de téléphone : (###) ###-#### <i>(où # représente un nombre entre 0 et 9)</i> Minimum pour l'âge : 18
Par liste de contenus acceptés <i>[pour chaque occurrence individuellement]</i>	<ul style="list-style-type: none"> Ex. : Liste d'auteurs, liste de descripteurs, etc.
Par contrôle du nombre d'occurrences <i>[pour chaque enregistrement]</i>	<ul style="list-style-type: none"> Occurrences <u>simples</u> ou <u>multiples</u> Ex. de champ à occurrence <i>simple</i> : date de naissance Ex. de champ à occurrences <i>multiples</i> : auteurs Entrée <u>obligatoire</u> : Le champ doit obligatoirement être saisi (au moins 1 occurrence) dans chaque enregistrement

SGBD textuels

Saisie des données [3/3]

➔ Paramètres de validation [2/2]

Type de validation	Explication & Exemple dans DB/TextWorks
Par contrôle de l'unicité des valeurs <i>[pour la base de données]</i>	<ul style="list-style-type: none">• <i>Entrées uniques</i> : Chaque occurrence doit avoir une valeur distincte de celle des autres occurrences dans ce champ dans la base<ul style="list-style-type: none">Ex. pour un champ à occurrence simple : <code>numero d'identification</code>Ex. pour un champ à occurrences multiples : <code>cotes</code> (si plusieurs exemplaires, avec chacun sa cote unique)

SGBD textuels

Comparaison SGBD textuels & LRTI

<i>LRTI (par ex. NatQuest)</i>	<i>SGBD textuels (par ex. DB/TextWorks)</i>
<p>Saisie des données : données déjà saisies</p>	<p>Saisie des données : se fait dans le logiciel lui-même (permet ainsi de contrôler la saisie)</p>
<p>Structuration des données :</p> <ul style="list-style-type: none"> ▪ Structuration par champ facultative ▪ Pas de champ à occurrences multiples 	<p>Structuration des données :</p> <ul style="list-style-type: none"> ▪ Structuration par champ obligatoire ▪ Possibilité d'avoir des champs à occurrences multiples
<p>Indexation :</p> <ul style="list-style-type: none"> ▪ Un seul index par base ▪ Index-mots seulement 	<p>Indexation :</p> <ul style="list-style-type: none"> ▪ Peut inclure plusieurs index par base <ul style="list-style-type: none"> ▪ <i>jusqu'à 2 index par champ</i> ▪ Index-mots et index-occurrences
	<p>Recherche : Fonctionnalités de recherche plus élaborées (par ex. exploitation des occurrences)</p>
<p>Résultats : Un item correspond à 1 document ou à 1 fichier</p>	<p>Résultats : Un item correspond à 1 fiche/enregistrement</p>