

SCI6306 Bases de données documentaires (A2023)

Christine Dufour, EBSI, UdeM

A2023 8 septembre 2023

Cours 1 : Introduction

SCI6306

Christine Dufour, EBSI, UdeM

Table des matières

| | |
|---|-----------|
| I - Cours 1 - Introduction | 3 |
| 1. + Au programme aujourd'hui..... | 3 |
| 2. Présentation du plan de cours..... | 4 |
| 3. Environnements et ressources pour le cours..... | 5 |
| 4. Principes généraux des SGBD | 5 |
| 4.1. Définitions | 5 |
| 4.2. Objectifs des SGBD..... | 6 |
| 4.3. Classes d'utilisateurs | 6 |
| 4.4. Architecture des systèmes de base de données..... | 6 |
| 4.5. Origine des SGBD | 7 |
| 5. Principaux modèles de SGBD | 7 |
| 5.1. Modèle textuel (fichier plat)..... | 7 |
| 5.2. Modèles hiérarchique et réseau | 7 |
| 5.3. Modèle relationnel (rappel SCI6005)..... | 8 |
| 5.4. Familles NoSQL | 11 |
| 5.5. Modèle hypertextuel | 12 |
| 6. Introduction aux SGBD relationnels..... | 14 |
| 6.1. Schématisation des principaux processus associés à une BD relationnelle..... | 14 |
| 6.2. Exemples d'implantation d'une BD relationnelle..... | 15 |
| 6.3. Formule retenue pour le cours | 15 |
| 7. Ressources en lien avec le cours..... | 16 |
| Bibliographie | 17 |
| Webographie | 18 |

I Cours 1 - Introduction

- + Au programme aujourd'hui
- Présentation du plan de cours
- Environnements et ressources pour le cours
- Principes généraux des SGBD
 - Définitions
 - Objectifs des SGBD
 - Classes d'utilisateurs
 - Architecture des systèmes de base de données
 - Origine des SGBD
- Principaux modèles de SGBD
 - Modèle textuel (fichier plat)
 - Modèles hiérarchique et réseau
 - Modèle relationnel (rappel SCI6005)
 - Pointures de SGBD relationnels
 - Modèles textuel et relationnel : principales ressemblances
 - Modèles textuel et relationnel : différences au niveau de la structure
 - Modèles textuel et relationnel : différences au niveau de la recherche
 - Modèles textuel et relationnel : en résumé
 - Limites du modèle relationnel
 - Familles NoSQL
 - Modèle hypertextuel
- Introduction aux SGBD relationnels
 - Schématisation des principaux processus associés à une BD relationnelle
 - Exemples d'implantation d'une BD relationnelle
 - Scénario 1 : Base de données relationnelle sur un ordinateur personnel
 - Scénario 2 : Base de données relationnelle sur le Web
 - Formule retenue pour le cours
- Ressources en lien avec le cours

1. + Au programme aujourd'hui

- Présentation du **plan de cours**
- Présentation des **environnements** et **ressources** pour le cours
- Introduction
 - Principes généraux des **systèmes de gestion de bases de données (SGBD)**
 - Historique et survol des principaux **modèles de bases de données**
 - *Rappel SCI6005* : ressemblances et différences entre le **modèle textuel** et le **modèle relationnel**
 - Introduction aux **SGBD relationnels**

2. Présentation du plan de cours

Plan de cours

Plan de cours disponible à l'URL <http://cours.ebsi.umontreal.ca/planscours/sci6306>¹

- Description du cours
- Objectifs du cours
- Méthodes pédagogiques
- Calendrier des activités
- Évaluation
- Politiques
- Ressources
- Modes de communication

Remarque : Préalables pour le cours

- **Seul préalable** : SCI6005 (ou l'équivalent) principalement pour la connaissance de la *structuration de l'information dans une base de données* et le *développement Web* (HTML et CSS)
- **Aucune attente** quant à une éventuelle expérience ou connaissance en **programmation**.
- Par contre, si HTML, par sa syntaxe précise et exacte, a semblé difficile dans le SCI6005, ce cours pourrait être un réel défi... Tant SQL que PHP, comme les exemples ci-dessous le montrent, demandent de la précision !

Exemple : Exemple d'une requête SQL simple

```
1 SELECT DISTINCT etud.no_etud, nom
2 FROM etud, suit
3 WHERE etud.no_etud = suit.no_etud;
```

Exemple : Exemple d'une requête SQL complexe

```
1 SELECT etud.no_etud, nom, COUNT(no_prof) AS [Nombre de prof vus], (SELECT count(*) FROM
  suit WHERE etud.no_etud = suit.no_etud AND suit.no_cours IN (SELECT no_cours FROM cours
  WHERE no_prof is null)) AS [Nb de cours sans prof] FROM etud, (SELECT DISTINCT
  suit.no_etud, cours.no_prof FROM cours, suit WHERE suit.no_cours=cours.no_cours) AS aux
  WHERE etud.no_etud=aux.no_etud GROUP BY etud.no_etud, nom;
```

Exemple : Exemple de code PHP

```
1 <?php
2 $balise=$_GET["balise"];
3
4 try
5 {
6     include 'App_Data/ouverture_bd.php';
7     $req = $bdd->prepare('SELECT balises.* FROM balises WHERE balise like :balise;');
8
9     $req->execute(array('balise' => $balise));
10
11     while ($donnees = $req->fetch())
12     {   ?>
13         <p><strong>Balise : <?php echo $donnees['balise'];?></strong></p>
14         <p><em>Définition</em> : <?php echo $donnees['definition'];?></p>
15         <?php
16     }
17
18     $req->closeCursor();
```

¹<http://cours.ebsi.umontreal.ca/planscours/sci6306> [page Web publique]

```

19     $bdd=null;
20 }
21
22 catch(Exception $e)
23 {
24     die('Erreur : '.$e->getMessage());
25 }
26 ?>

```

3. Environnements et ressources pour le cours

Site de cours

Les notes de cours sont accessibles sur un site web public à l'URL <https://cours.ebsi.umontreal.ca/sci6306/>. Les notes de cours seront mises en ligne au fur et à mesure de l'avancement de la session. Elles seront présentées par semaine et accessibles soit en lecture directement dans votre navigateur web ou en version PDF pour impression (Section >**Ressources** au sein de chacune des semaines).

StudiUM

StudiUM sera utilisé pour présenter certaines ressources à accès restreint qui ne peuvent être présentées sur le site du cours (énoncés des travaux pratiques par exemple) (<https://studium.umontreal.ca/course/view.php?id=250499>¹). De plus, certains outils de StudiUM seront exploités comme les espaces de dépôt pour les travaux et les forums de discussion.

Équipes sur Microsoft Teams

Des canaux privés définis dans l'équipe **Gr - EBSI-A23-SCI6306** sur *Microsoft Teams* permettront de clavarder en dehors des heures de cours. Plus précisément :

- **Canal individuel privé** : canal pour le travail individuel sur les requêtes SQL - toutes les personnes inscrites au cours auront un canal privé à leur nom pour discuter avec la professeure
- **Canal individuel d'équipe** : canal pour le travail d'équipe (projet de session) - chaque équipe pour ce travail aura un canal privé pour discuter entre elles/eux et avec la professeure - il est à noter que les canaux ne seront définis qu'une fois les équipes formées; il est ainsi normal de ne pas y avoir accès en début de session

Remarque :

Un mode d'emploi abrégé sur l'utilisation de *Microsoft Teams* est accessible à l'URL https://cours.ebsi.umontreal.ca/sci6005/mode_emploi/teams.

4. Principes généraux des SGBD

4.1. Définitions

Définition : Système de base de données

Un *système de base de données* est un système informatique servant à **maintenir des informations** et à les **rendre disponibles à la demande**. Il possède quatre composantes :

1. Les données
2. Le matériel
3. Le logiciel
4. Les utilisateurs

¹<https://studium.umontreal.ca/course/view.php?id=180464>

🔍 Définition : Système de gestion de bases de données (SGBD)

Un système de gestion de bases de données (SGBD) est la **composante logicielle** la plus importante d'un système de base de données. Il prend en charge les **requêtes** pour **accéder à la base de données**, pour y faire des **ajouts**, des **suppressions**, des **mise à jour**, etc.

4.2. Objectifs des SGBD

Un SGBD vise à offrir des mécanismes pour :

- S'assurer de la **cohérence** des données
- Assurer la **sécurité** et le **partage** des données
- Assurer l'**indépendance** des données (par rapport au matériel et au logiciel)
- Permettre d'exploiter les **liens** entre les données
- Obtenir une bonne **performance** (vitesse et gestion d'espace)

4.3. Classes d'utilisateurs

On retrouve trois principaux types d'intervenants et intervenantes avec les SGBD : les utilisatrices et utilisateurs finaux, les administratrices et administrateurs de bases de données et les programmeuses et programmeurs d'application.

Classes d'intervenants et intervenantes avec les SGBD

| | |
|--|--|
| Utilisatrices et utilisateurs finaux | Interagissent avec la base de données principalement pour l' interroger et imprimer des rapports . Tout dépendant leur niveau de connaissance du langage d'interrogation, elles et ils peuvent soit utiliser des requêtes prédéfinies ou en formuler elles-mêmes ou eux-mêmes. |
| Administratrices et administrateurs de bases de données | Assurent la gestion technique nécessaire pour implémenter les SGBD : définition de la structure conceptuelle et physique, définitions des règles de sécurité, interaction avec les utilisateurs finaux, supervision des performances, etc. |
| Programmeuses et programmeurs d'application | Programment des applications pour interagir avec la base de données (par exemple des pages ASP ou PHP pour mettre une base de données en ligne). |

Les **professionnelles et professionnels de l'information** interviennent principalement comme **utilisatrices et utilisateurs finaux** ou **administratrices et administrateurs de bases de données**. Il leur est aussi possible d'agir comme *programmeuses et programmeurs d'application* pour un système à petite échelle.

4.4. Architecture des systèmes de base de données

Il y a trois niveaux à prendre en considération dans la définition des données dans un SGBD :

- **Niveau interne** (vue physique) : niveau où les données sont enregistrées
- **Niveau externe** (vue d'un utilisateur particulier) : niveau où les utilisateurs voient les données
- **Niveau conceptuel** (vue commune à tous les utilisateurs) : niveau intermédiaire où les données sont organisées dans le logiciel

Lorsque l'on contribue au développement d'une base de données, il est important de toujours garder ces trois niveaux en tête.

4.5. Origine des SGBD

Tout SGBD est basé sur un **modèle de données**, constitué de :

- Une façon de **structurer** les données
- Des **opérations** pour agir sur les données

Différents modèles de données ont vu le jour, dont certains sont maintenant obsolètes, sous l'impulsion de l'évolution des langages de programmation, de la capacité des systèmes ainsi que des besoins sous-jacents.

5. Principaux modèles de SGBD

Au fil de l'évolution de la technologie et des besoins pour des bases de données, différents modèles de données ont vu le jour. Les modèles présentés sont quelques uns des principaux modèles, mais d'autres existent.

5.1. Modèle textuel (fichier plat)

- Les données, dans le modèle textuel, sont structurées en **une seule table**, chaque ligne représente un enregistrement et chaque colonne, un champ. Les enregistrements dans cette table de données ne sont pas reliés entre eux.
- Ce type de bases de données propose habituellement des fonctionnalités bien adaptées aux **données textuelles**, c'est-à-dire composées surtout de phrases ou de mots, comme, p. ex., des opérateurs de proximité.
- Inmagic DB/TextWorks¹ est un exemple de base de données basée sur le modèle textuel. Plusieurs bases de données commerciales accessibles en ligne sur des serveurs sont gérées par des SGBD textuels.

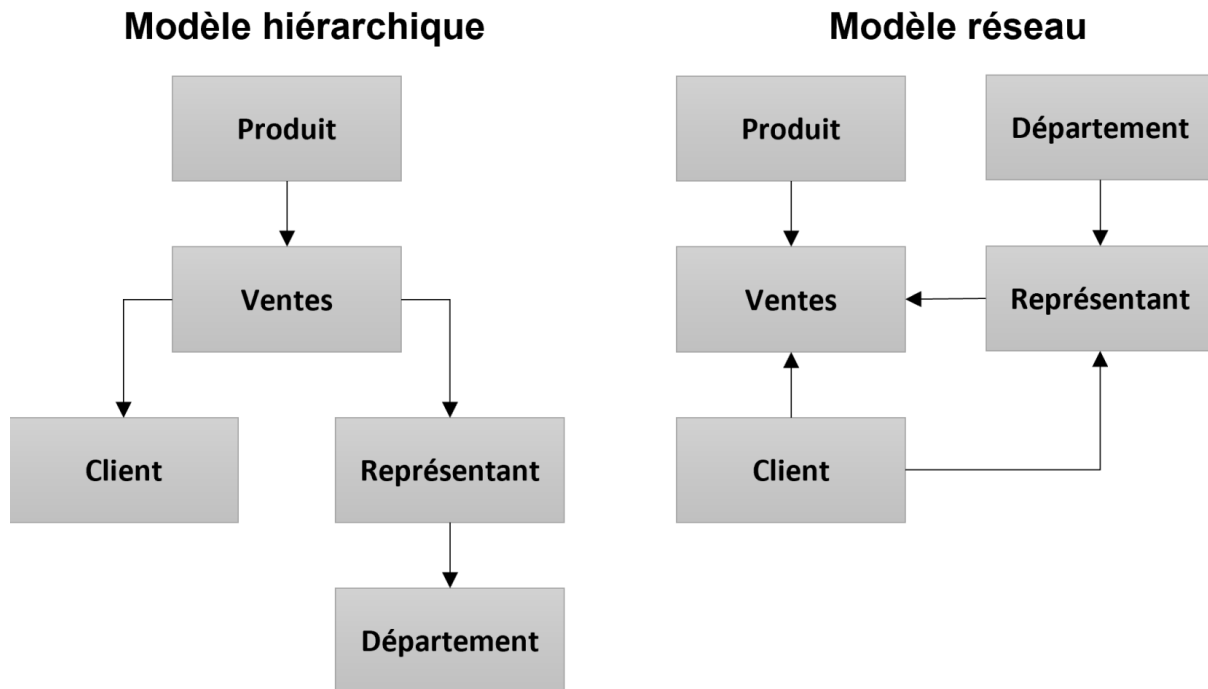
5.2. Modèles hiérarchique et réseau

Tant le modèle hiérarchique que le modèle réseau sont aujourd'hui **obsolètes**. Ils demeurent toutefois des jalons importants dans l'évolution des bases de données, le modèle hiérarchique étant apparu en premier suivi du modèle réseau. Une de leur principale faiblesse est que tous deux présentaient une **indépendance limitée** des données par rapport aux applications et aux technologies. Ils ont été dépassés par le relationnel à partir des années 1980 (arrêt du développement de nouvelles applications selon ces modèles).

Caractéristiques des modèles hiérarchique et réseau

| Modèle hiérarchique | Modèle réseau |
|--|--|
| <ul style="list-style-type: none"> • Données structurées en utilisant des pointeurs unidirectionnels (un élément mène uniquement à un autre) • Requêtes à saveur procédurale plutôt que déclarative (comment aller chercher les données plutôt que quoi aller chercher) • Modèle défini par des produits et non sur un modèle abstrait (d'où une grande dépendance aux technologies) • Produit le plus connu : <i>Information management system</i> (IMS) de IBM (1968) | <ul style="list-style-type: none"> • Données structurées en utilisation des pointeurs bidirectionnels (un élément peut pointer et être pointé par plus d'un élément) • Modèle défini dans un rapport du Data Base Task Group de CODASYL (1971) • Basé sur le langage de programmation COBOL • Produit le plus connu : <i>Integrated Database Management System</i> (IDMS) de Computer Associates |

¹ <https://lucidea.com/inmagic-dbtextworks/>



Source : Harrison, Guy. *Next generation databases: NoSQL, NewSQL, and Big data.* [New York] : aPress, 2015, p. 16. (Notre traduction)

5.3. Modèle relationnel (rappel SCI6005)

- Défini **formellement** (mathématiquement) avant toute implantation par l'informaticien britannique Edgar Frank Codd en 1970, ce qui en fait un "vrai" modèle. Le premier produit basé sur ce modèle a vu le jour à la fin des années 70.
- Modèle encore **le plus utilisé actuellement**. Beaucoup de systèmes intégrés de bibliothèques (SIGB) et autres applications documentaires sont construites sur des SGBD relationnels. Aussi très présent au niveau des systèmes d'information Web, quoiqu'il ne soit plus le seul!
- Ses **limites** pour les systèmes distribués à **grande échelle** sur le Web comme Twitter et Facebook ont conduit à l'apparition des familles de bases de données NoSQL (*Not Only SQL*)

a) Pointures de SGBD relationnels

On retrouve des SGBD relationnels de taille variable selon les contextes et les besoins auxquels ils répondent. Dans le cadre du cours, nous explorerons les SGBD relationnels de petite pointure.

Pointures de SGBD relationnels

| Petite pointure = SGBDR personnels | Grande pointure = SGBDR corporatifs |
|--|---|
| <ul style="list-style-type: none"> • Des BD relativement petites, pour une personne ou un groupe restreint • Un volume des données et/ou d'accès réduit • Des situations où la performance n'est pas un facteur critique • Ex. : Access, FileMaker Pro, MySQL | <ul style="list-style-type: none"> • Haute performance • Habituellement sur un serveur dédié • Permet l'intégration de toutes les BD d'une organisation • Rôle de l'administrateur de bases de données très important • Ex. : Oracle, SQL Server |

b) Modèles textuel et relationnel : principales ressemblances

Les SGBD textuel et relationnel se ressemblent quant au **contexte global de leur utilisation** ainsi que l'**unité de base** utilisée pour structurer les données.

Contexte global d'utilisation des SGBD textuel et relationnel

| | SGBD textuel | SGBD relationnel |
|---|---------------------------|---------------------------|
| | Par ex. DB/TextWorks | Par ex. Access et MySQL |
| Type de « contenants » d'information | BD créée avec le logiciel | BD créée avec le logiciel |
| Recherche de fichiers | non | non |
| Recherche de fiches/enregistrements | toujours | toujours |
| Structuration en champs | toujours | toujours |

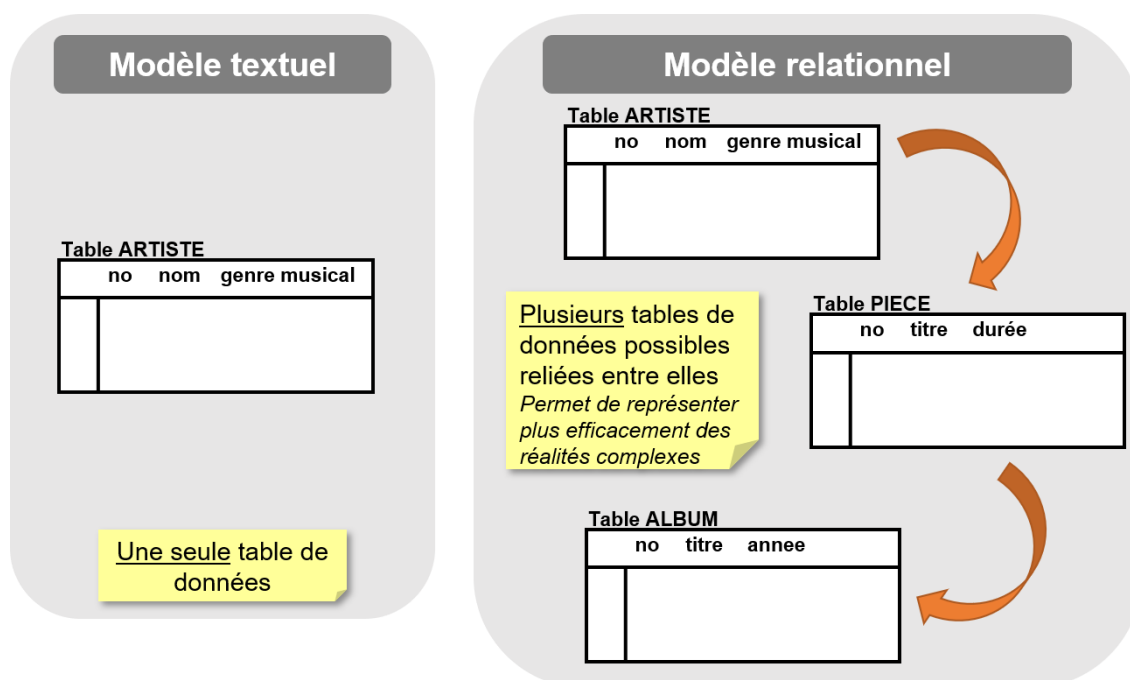
| <i>Colonnes : champs</i> | no_cours | local | no_prof | titre | jour | heure | statut | <i>Cellules : données</i> |
|--------------------------|----------|--------|---------|-----------------|------|----------|-------------|---------------------------|
| | 20001 | T-5334 | 30001 | Histoire 101 | 1 | 09:00:00 | optionnel | |
| | 20002 | T-5334 | 30002 | Éthique 302 | 1 | 13:00:00 | obligatoire | |
| | 20003 | NULL | 30001 | Philosophie 813 | 1 | 09:00:00 | optionnel | |

Unité de base : Table(s) de données

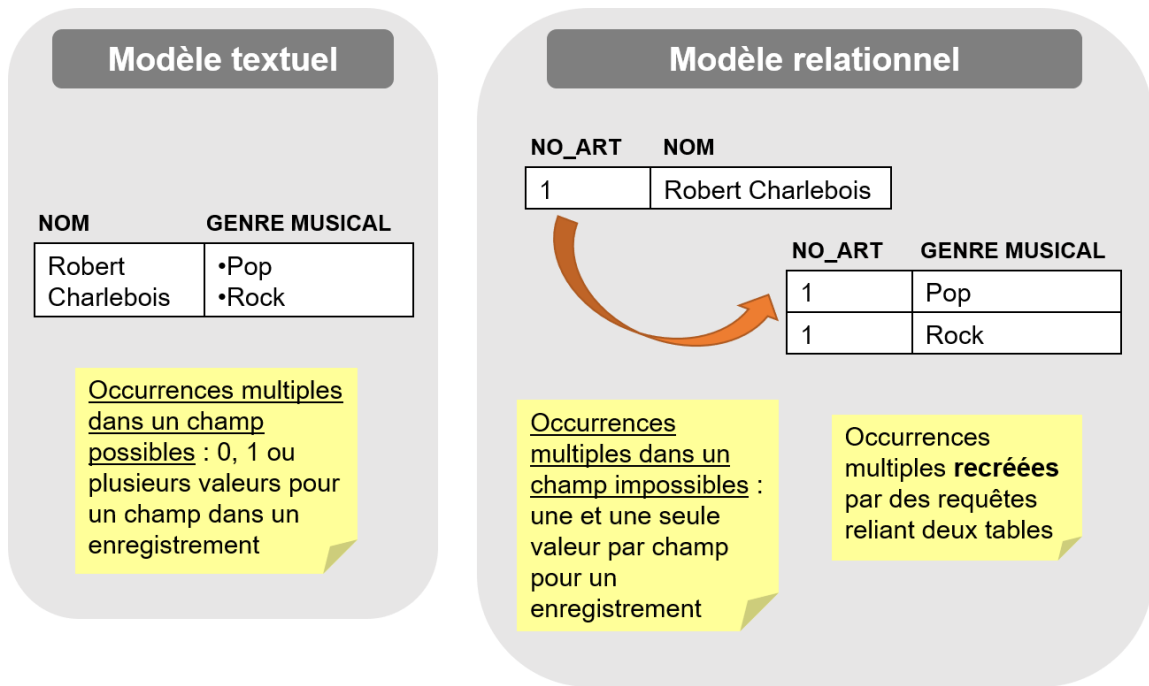
Lignes : fiches (ou enregistrements)

c) Modèles textuel et relationnel : différences au niveau de la structure

Les modèles textuel et relationnel diffèrent quant au **nombre de tables de données** qu'elles peuvent utiliser pour structurer une base de données ainsi qu'au **nombre d'occurrences** que l'on peut retrouver dans un champ.



SGBD textuel et relationnel : nombre de tables de données



SGBD textuel et relationnel : nombre d'occurrences dans un champ

d) Modèles textuel et relationnel : diff rences au niveau de la recherche

En sus des diff rences et ressemblances entre les deux mod les concernant leur mani re de structurer les donn es, ces derniers se ressemblent sur certains points et diff rent sur d'autres quant aux op rations de recherche dans une base de donn es.

Diff rences au niveau de la recherche entre les mod les textuel et relationnel

| | Textuel : Db/TextWorks | Relationnel : MySQL |
|---|--|---|
| Recherche index e | toujours | facultatif (sauf pour la recherche avanc e plein texte) |
| Visualisation de l'index | oui | non |
| Antidictionnaire | un par base, modifiable (applicabilit  par champs) | non (sauf pour la recherche avanc e en plein texte) |
| Autres crit res sur les m tadonn es | non | non |
| Signes diacritiques ignor s | facultatif | facultatif |
| Op rateurs bool ens | oui | oui |
| Troncature | oui | oui |
| Op rateurs de distance | oui (dans l'ordre ou non) | non (sauf pour la recherche avanc e plein texte) |
| Autres op rateurs (p. ex. math matiques) | limit s | oui |

e) Modèles textuel et relationnel : en résumé

Il est important de connaître les caractéristiques des modèles textuel et relationnel afin de mieux en comprendre l'utilisation et faire des choix plus éclairés vers l'un ou l'autre de ces modèles en fonction du contexte.

Résumé des principales caractéristiques des modèles textuel et relationnel

| | SGBD textuel Par ex. DB/TextWorks | SGBD relationnel Par ex. MySQL |
|---|--|---|
| Nombre de table de données | une | une ou plus |
| Occurrences multiples dans un champ | oui | non Les occurrences multiples sont recréées, à la demande, par des requêtes reliant des tables |
| Possibilité d'avoir 0 occurrence dans un champ | oui | non La valeur NULL est utilisée pour "simuler" l'absence d'occurrence |
| Types de données | plus limités | plus riches et plus forts |
| Langage d'interrogation | plus limité globalement, mais présente plus d'opérateurs pour les données textuelles | plus riche (SQL) sauf pour certains aspects du traitement des données textuelles qui nécessitent de passer par la recherche avancée plein texte |

f) Limites du modèle relationnel

- Le fait d'avoir mathématiquement et formellement défini le modèle relationnel avant toute implantation lui a permis de "bien vieillir", comme il a ainsi gagné en indépendance par rapport aux technologies.
- Ceci dit, l'évolution entre autres du matériel informatique, des langages de programmation, des exigences des interfaces-utilisateurs, des applications multimédia ainsi que de la réseautique en a fait ressortir des limitations, ce qui a amené d'autres modèles à émerger. C'est le cas entre autres des familles NoSQL nées des besoins pour des systèmes distribués à grande échelle comme Twitter ou Facebook.

5.4. Familles NoSQL

- L'appellation NoSQL (*Not only SQL*) date de 2009. Le **modèle relationnel**, dans sa manière de représenter et de manipuler les données du modèle relationnel, se révèle **peu efficace** dans le contexte d'environnements Web distribués à **grande échelle** qui possèdent de **grands volumes** de données (p. ex Twitter, Facebook, eBay).
- Les familles NoSQL délaissent les propriétés des transactions relationnelles qui permettent de garantir et de maintenir la cohérence des données (propriétés ACID) au profit de contraintes qui priorisent la disponibilité des données (contraintes BASE).
- Il ne s'agit pas d'un modèle NoSQL, mais de **plusieurs familles NoSQL** qui se sont développées en parallèle et qui répondent à des besoins différents.

Exemples de familles NoSQL

| Familles | Exemples d'utilisation |
|---|---|
| Orientées graphes pour traiter les réseaux massifs | Recommandations Twitter |
| Orientés colonnes pour faciliter les traitements privilégiant les colonnes | Calcul de l'âge moyen des utilisateurs |
| Orientées clé/valeur pour gagner en efficacité en lecture/écriture et pour le changement d'échelle | Système de sauvegarde de type Dropbox |
| Orientée "document" pour supporter des structures variables | Gestion des métadonnées des produits vendus chez eBay |

Ces nouvelles familles de bases de données présentent comme principal avantage de contourner les limites du modèle relationnel contraignantes pour le contexte de systèmes Web distribués à grande échelle (fortes performances, résistance au changement d'échelle, entre autres). Elles ne sont toutefois pas exemptes de limites, par exemple du fait de leur relative jeunesse et du développement en parallèle de plusieurs familles.

Avantages et désavantages des familles NoSQL

| Avantages | Désavantages |
|--|--|
| <ul style="list-style-type: none"> • Faible coût relatif • Fortes performances • Résistance au changement d'échelle • Capacité à faire évoluer la représentation des données | <ul style="list-style-type: none"> • "Jeunesse" • Pas de langage de requête abstrait partagé • Travail de programmation spécifique plus important • Cohérence des données moins aisée à garantir |

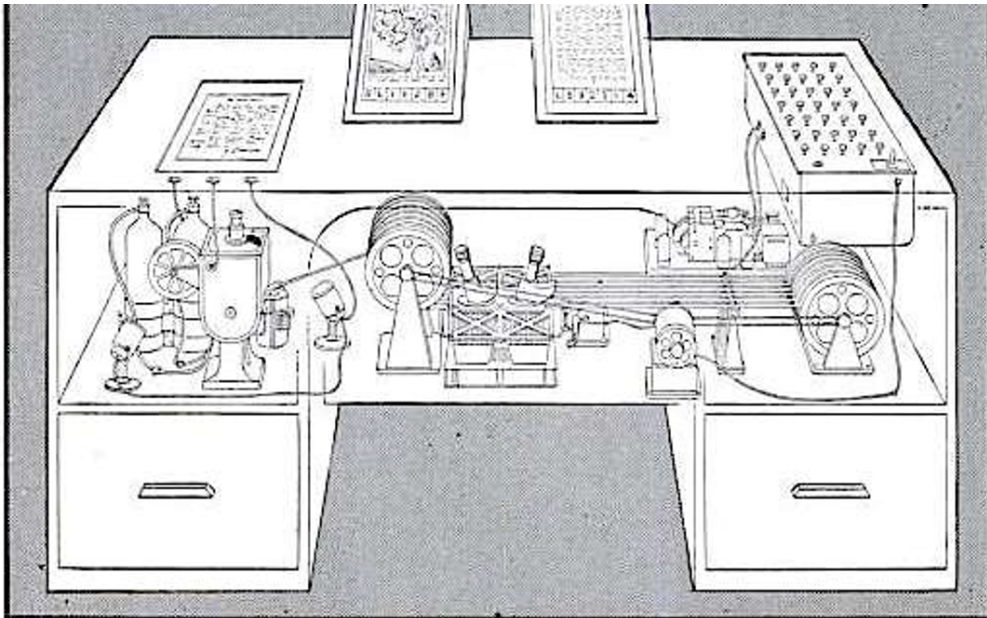
Remarque :

L'information présentée dans cette section provient de *S4.4 : Bases de données non relationnelles / Habert^{HABERT, Benoît, 2016}**, que vous pouvez consulter pour plus d'informations sur ces dernières.

5.5. Modèle hypertextuel

Le modèle hypertextuel remonte, dans sa conceptualisation, à **1945. Vannevar Bush**, dans son texte "*As We May Think*",^{BUSH 1945}* propose l'idée d'une machine, nommée **MEMEX**, pour résoudre les problèmes liés à l'explosion documentaire. Cette dernière serait en effet en mesure de ranger et rendre accessible tous les documents en permettant :

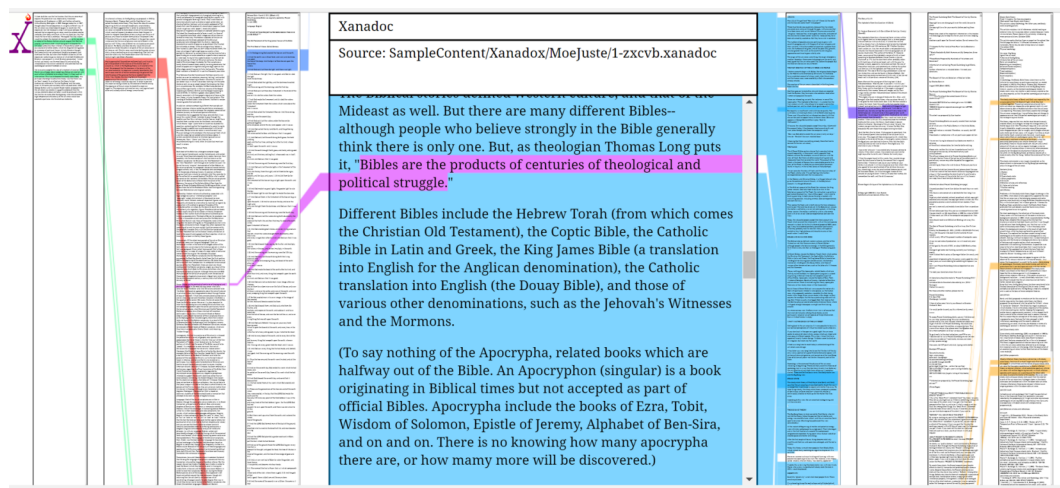
- La **visualisation** des documents sur microfilm sur un écran, documents pouvant être complétées par des images ;
- La **localisation** des documents ;
- L'inclusion de **pistes associatives** entre les documents.



MEMEX in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. Slanting translucent viewing screens magnify supermicro-film filed by code numbers. At left is a mechanism which automatically photographs longhand notes, pictures and letters, then files them in the desk for future reference.

Source : Bush, Vannevar. *As We May Think*. *The Atlantic Monthly*, 176(1), 112-124, 1945.

Le néologisme "**hypertexte**" a été proposé par le philosophe et sociologue américain **Theodore Nelson** en **1965**. Il est à l'origine du projet **Xanadu** dont l'objectif était de regrouper tous les documents de l'humanité. Ce système, accessible à tous, permettrait l'**ajout de documents** et la définition de **liens** entre ces derniers. Au départ, il s'agissait uniquement d'une conceptualisation sans implémentation. Depuis, des démonstrations ont vu le jour après bien des années dont **OpenXanadu**¹ en **2014**^{HERN 2014} *.

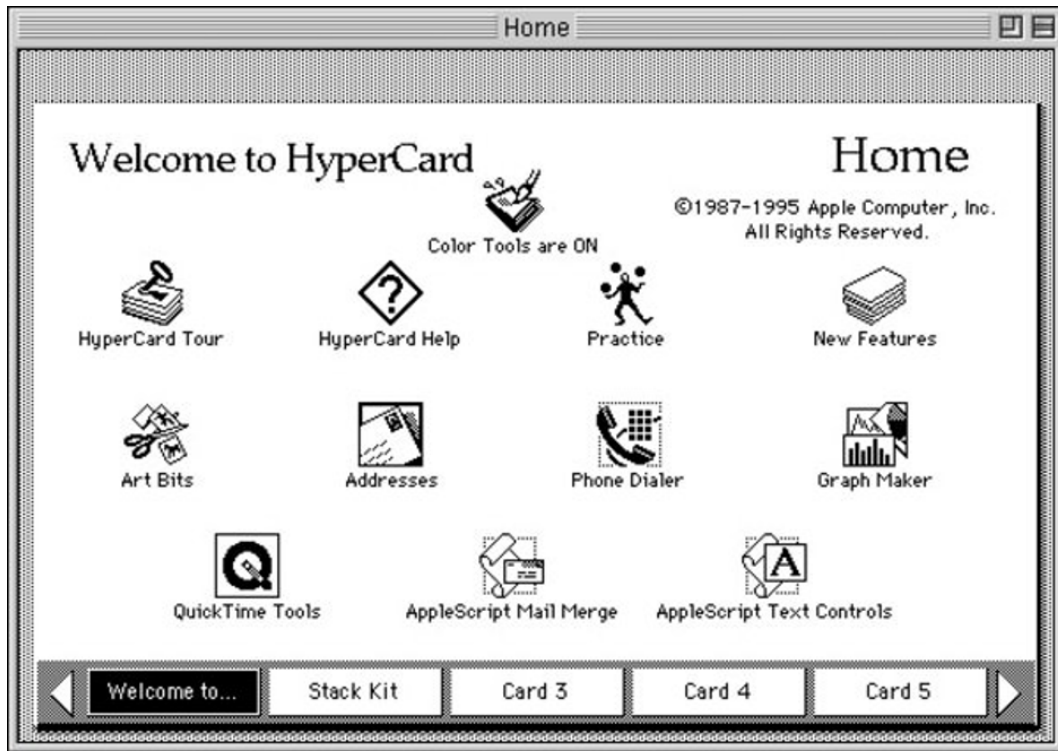


Source : *OpenXanadu (sources only)*. <https://xanadu.com/xanademos/MoeJusteOrigins.html>

Le **premier logiciel hypertexte grand public**, créé par Bill **Atkinson**, est **HyperCard** (Apple) en 1987, logiciel dont le développement s'est terminé en 1998. Il permettait de présenter des contenus multimédias sur des fiches cartonnées, cartes qui pouvaient être liées entre elles par des boutons. Pour les curieux (ou les nostalgiques!), Internet Archive a mis en ligne en 2017 pour fêter les 30 ans d'HyperCard une collection de projets HyperCard².

¹ <http://xanadu.com/xanademos/MoeJusteOrigins.html>

² <https://archive.org/details/hypercardstacks&tab=about>



Source : Scott, Jason. 11 août 2017. *HyperCard On The Archive (Celebrating 30 Years of HyperCard)*. Internet Archive Blogs. <https://blog.archive.org/2017/08/11/hypercard-on-the-archive-celebrating-30-years-of-hypercard/>

L'application **la plus connue** du modèle hypertextuel, le Web, a été créée au CERN par l'équipe de Tim Berners-Lee en 1989. Il s'agit d'une version "allégée" des visions initialement proposées pour l'hypertexte entre autres par T. Nelson et V. Bush :

- Pas de gestion des droits d'auteur ;
- Liens unidirectionnels se cassant lorsque la ressource disparaît ;
- Pas d'aperçu des ressources liées ;
- Pas de système d'annotation ni de gestion de versions.

6. Introduction aux SGBD relationnels

Les concepts de base que l'on retrouve dans le modèle relationnel sont :

- La notion de **tables** (les données sont structurées en champs) ;
- La notion de **relations** (les tables sont liées entre elles) ;
- La notion de **requêtes** (afin de chercher dans les tables, mais aussi de manipuler les données dans les tables, etc.).

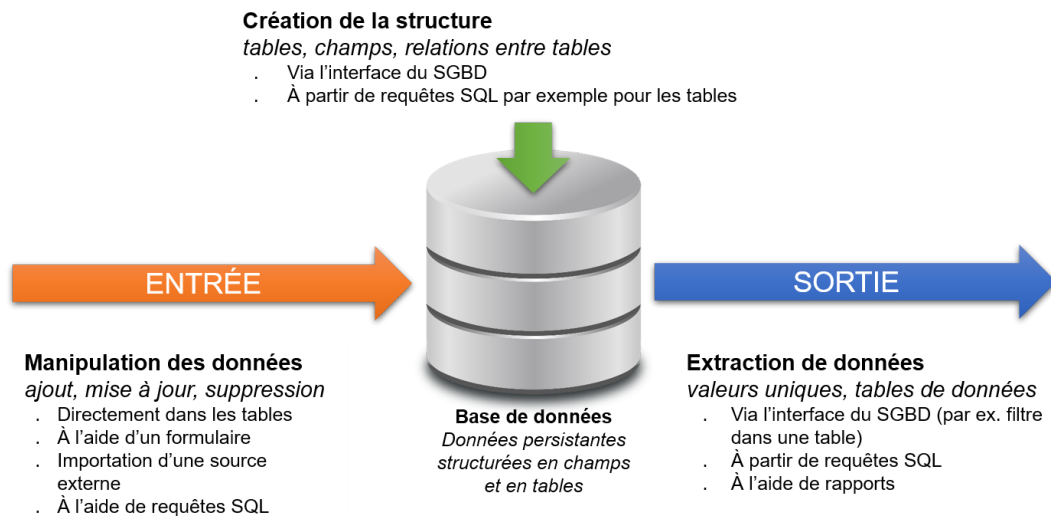
Nous verrons au fur et à mesure du déroulement de la session les caractéristiques de ces différents éléments ainsi que diverses fonctionnalités reliées.

6.1. Schématisation des principaux processus associés à une BD relationnelle

Que ce soit en tant qu'utilisateur, d'administrateur de bases de données ou de programmeur d'applications, il est important de bien visualiser les différents processus associés à une BD relationnelle. On retrouve trois "moments" autour d'une base de données :

- La **création de la base de données**, qui relève de l'administrateur de la base de données ;
- L'**entrée de données** dans la base de données, qui peut relever de l'administrateur de la base de données, mais qui peut aussi être déléguée à un "super utilisateur" par le truchement d'interfaces de saisie développées spécifiquement à cette fin par le programmeur d'applications ;

- L'**extraction de données**, qui peut se faire par un utilisateur final lorsqu'il effectue une recherche. Le programmeur d'applications ou l'administrateur de bases de la base de données aura la tâche de concevoir les différents formats de sortie à cette fin.



Processus autour d'une base de données

6.2. Exemples d'implantation d'une BD relationnelle

Différentes solutions peuvent être mises en place lors de l'implantation d'une BD relationnelle. Deux scénarios classiques seront présentés pour le **contexte suivant** :

- Base de données relationnelle servant à gérer une **collection de pièces musicales** ;
- Quatre éléments représentés : (1) des **pièces musicales**, (2) des **artistes** ayant enregistré des pièces, (3) des **albums** contenant des pièces, (4) le **lien entre les pièces et les artistes**.

a) Scénario 1 : Base de données relationnelle sur un ordinateur personnel

- *Contexte* : volume de données restreint, un seul utilisateur à la fois
- *Ex. SGBD* : Microsoft Access (Windows) / LibreOffice Base (Windows, Mac, Linux) / FileMaker Pro (Windows, Mac)
- *Entrée* : formulaires préparés dans le SGBD pour la saisie
- *Sortie* : (1) États pour imprimer le contenu de la base de données, (2) Requêtes SQL prédéfinies pour des besoins « type »

b) Scénario 2 : Base de données relationnelle sur le Web

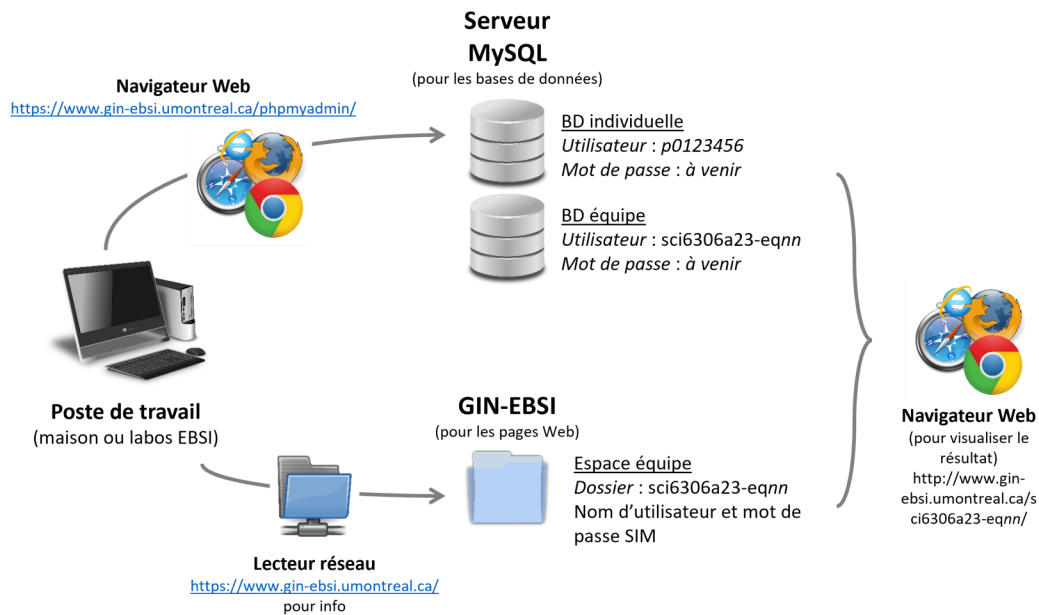
- *Contexte* : accès distribué à de multiples utilisateurs sur des plateformes variées
- *Ex. SGBD* : MySQL sur un serveur Web / Microsoft Access (Windows)
- *Entrée* : formulaires Web pour la saisie (par ex. en PHP ou ASP)
- *Sortie* : (1) Interface de recherche Web (par ex. en PHP ou ASP), (2) Pages Web dynamiques (par ex. en PHP ou ASP) présentant les résultats pour des besoins « type »

6.3. Formule retenue pour le cours

Dans le cadre du cours, c'est le scénario 2 qui est retenu, soit une **base de données relationnelle sur le Web**. Le SGBD utilisé sera **MySQL**, avec **phpMyAdmin** comme interface de gestion. Le langage de programmation Web utilisé sera **PHP** et nous travaillerons sur le **serveur de l'EBSI**.

Des espaces individuels et d'équipe seront créés

- Chaque **étudiant.e** aura une **base de données personnelle** sur le **serveur MySQL** pour le travail individuel
- Chaque **équipe** aura une **base de données partagée sur le serveur MySQL** pour le travail d'équipe
- Chaque équipe aura un **espace partagé sur le serveur GIN-EBSI** pour le travail d'équipe pour héberger les pages Web dynamiques développées en PHP



Scénario d'implantation pour le cours

7. Ressources en lien avec le cours

Matériel de cours

- Notes de cours [cf. [sci6306_cours1_notes.pdf](#)]

Lectures obligatoires

- Introduction au modèle relationnel par comparaison avec le modèle textuel / Marcoux^{MARCOUX, Yves, 2007}*
- Définition et caractéristiques des bases de données / Dufour^{DUFOUR, Christine, 2016}*
- Bases de données sur le Web / Dufour^{DUFOUR, Christine, 2016}*

Lecture complémentaire

- Bases de données non relationnelles / Habert^{HABERT, Benoît, 2016}*

Bibliographie

BUSH, Vannevar. *As We May Think*. The Atlantic Monthly, 176(1), 112-114, 1945 <https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/>

HERN, Alex. *World's most delayed software released after 54 years of development*. The Guardian, 6 juin 2014 <https://www.theguardian.com/technology/2014/jun/06/vapourware-software-54-years-xanadu-ted-nelson-chapman>

Webographie

DUFOUR, Christine. 2016. *S4.1 Définition et caractéristiques des bases de données*. In *MOOC Architecture de l'information, Séquence 4 – Web et bases de données*. <https://archinfo00.hypotheses.org/231>

DUFOUR, Christine. 2016. *S4.2 Bases de données sur le Web*. In *MOOC Architecture de l'information, Séquence 4 – Web et bases de données*. <https://archinfo00.hypotheses.org/233>

HABERT, Benoît. 2016. *S4.4 Bases de données non relationnelles*. In *MOOC Architecture de l'information, Séquence 4 – Web et bases de données*. <https://archinfo00.hypotheses.org/237>

MARCOUX, Yves. 2007. *Introduction au modèle relationnel par comparaison avec le modèle textuel*. <http://marcoux.ebsi.umontreal.ca/enseign/6306/textuel-versus-relationnel.htm>