**Berkeley DB** fait partie de la famille des bases de données **« clé-valeur »** et est sans doute le produit de cette famille le plus utilisé au monde avec plusieurs dizaines de millions de déploiements revendiqués par l'éditeur. La première version de BDB remonte à 1986 et est la propriété d'Oracle depuis février 2006.

Les BD clé-valeur à leur tour sont membres de la famille NoSQL (Not only SQL en anglais) désigne une catégorie de [systèmes de gestion de base de données](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es) (SGBD) qui n'est plus fondée sur l'architecture classique des [bases relationnelles](https://fr.wikipedia.org/wiki/SGBDR). L'unité logique n'y est plus la [table](https://fr.wikipedia.org/wiki/Table_%28base_de_donn%C3%A9es%29), et les données ne sont en général pas manipulées avec [SQL](https://fr.wikipedia.org/wiki/Structured_Query_Language).

À l'origine, servant à manipuler des [bases de données](https://fr.wikipedia.org/wiki/Bases_de_donn%C3%A9es) géantes pour des sites web de très grande audience. Le NoSQL s'est aussi étendu par le bas après 2010. Il renonce aux fonctionnalités classiques des [SGBD relationnels](https://fr.wikipedia.org/wiki/SGBDR) au profit de la simplicité. Les performances restent bonnes avec la montée en charge ([scalabilit](https://fr.wikipedia.org/wiki/Scalabilit%C3%A9" \o "Scalabilité)y) en multipliant simplement le nombre de serveurs. Les systèmes géants sont les premiers concernés : **énorme quantité de données**, **structuration relationnelle faible** (ou de moindre importance que la capacité d'accès très rapide, quitte à multiplier les serveurs). Un modèle typique en NoSQL est le système clé-valeur, avec une base de données pouvant se résumer topologiquement à un simple [tableau associatif](https://fr.wikipedia.org/wiki/Tableau_associatif) unidimensionnel avec des millions — voire des milliards — d'entrées. Parmi les applications typiques, on retrouve des analyses temps-réel, statistiques, du stockage de logs (journaux), etc.

Un [système de gestion de base de données](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es) (SGBD) relationnel permet de réaliser des [transactions](https://fr.wikipedia.org/wiki/Transaction_informatique) atomiques, cohérentes, isolées, et durables (ACID).

Les capacités ACID garantissent que si plusieurs utilisateurs font de manière simultanée des modifications des données, toutes les modifications vont être prises en compte, dans un ordre précis et maîtrisé de manière à avoir un résultat cohérent (intégrité des données) avec l'historique des modifications faites par chacun. La mise en œuvre stricte des capacités **ACID entraîne des coûts logiciels importants et un niveau de performance moindre à infrastructure matérielle équivalente.**

Les SGBD d'annuaires ont servi de modèle en permettant de lever certaines de ces contraintes en fonction de l'usage, en particulier dans les cas où la grande majorité des accès aux bases de données **consistent en lectures sans modification** (dans ce cas, seule la propriété de persistance importe).

Les solutions du marché implémentent ce protocole en ajoutant leurs techniques propres pour limiter les conséquences de l'impossibilité d'ACID lors des écritures et mises à jour de données.

* La famille de Berkeley DB :

1. Oracle BDB Core : Cette version de Berkeley DB est écrite en C ANSI et peut être utilisée comme une bibliothèque pour accéder aux données persistées. Oracle BDB fournit de nombreuses interfaces pour différents langages de programmation (C, Perl, PHP, Python…).
2. Oracle BDB XML Edition : Cette version est une surcouche de BDB. Cette version XML permet de stocker et récupérer facilement des documents XML. L'utilisation de XQuery permet d'accéder aux documents XML stockés.
3. BDB Java Edition : Cette version de Berkeley DB, 100 % Java est une base embarquée adoptant le format de stockage clé-valeur. De plus, BDB JE est très performante et flexible. Cette base supporte les transactions, permet de stocker des objets Java en utilisant seulement quelques annotations.

* La famille de Berkeley DB :

Chaque membre de la famille BDB possède différentes fonctionnalités. Dans toutes les familles de BDB, nous pouvons remarquer les fonctionnalités suivantes :

|  |  |
| --- | --- |
| **Feature Set** | **Description** |
| Data Store (DS) | 1 Writer and n reader |
| Concurrent Data Store (CDS) | n writers and n snapshot reader |
| Transactional Data Store (TD) | Full ACID support on top of CDS |
| High Availability (HA) | Replication for fault tolerence. Fail over recovery support |

Le tableau suivant décrit les différences entre BDB "core" et BDB JE :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **DS** | **CDS** | **TS** | **HA** |
| BDB/BDB XML | X | X | X | X |
| BDD JE |  | X | X |  |

* Introduction à Berkeley DB Java Edition :

1. Installation :

Tout d'abord, débutons par le commencement. Pour pouvoir utiliser BDB Java Edition, vous devez l'installer à l'adresse suivante :

<http://www.oracle.com/technetwork/products/berkeleydb/downloads/index.html>. L'installation de Berkeley DB est assez simple, il suffit de décompresser l'archive que vous venez de télécharger ! Le seul fichier dont vous aurez besoin pour faire fonctionner les exemples que nous verrons plus tard est le : "je-\*.jar".

1. Les API d'accès aux données :

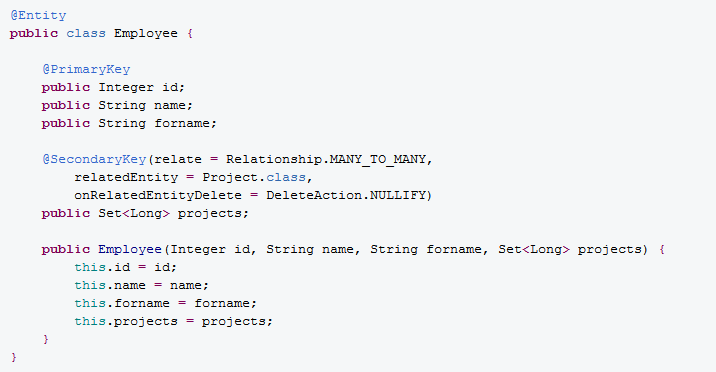
* l'API de base fournit un simple modèle clé-valeur pour stocker et récupérer les données.
* l'API DPL (Direct Persistence Layer) qui vous laisse persister n'importe quelle classe Java comportant un constructeur par défaut en base de données. Cette API vous permet de récupérer les données avec des fonctionnalités de recherche assez riches.
* la Collection API qui étend la Java Collection API avec le support de la persistance et des transactions en plus de l'accès aux données.

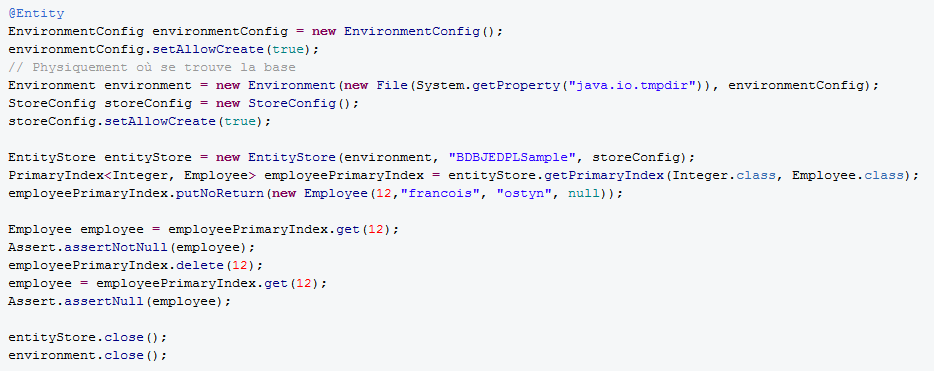
**L'API de base:**

Tout d'abord, avant de commencer à vous parler de DPL, même si nous sommes dans l'univers « NoSQL », il est toujours possible de faire une analogie avec le SQL :

|  |  |
| --- | --- |
| **Terme SQL** | **Équivalence Oracle DPL** |
| Database | Environment |
| Table | Primary Index and its related secondary indices |
| Primary Index | Primary Index |
| Secondary Index | Secondary Index |
| Tuple/Row | Entity |

L'exemple que je vais vous présenter se décompose en deux parties. Premièrement, un entity bean et une classe de management qui gère le CRUD (Create-Replace-Update-Delete) au-dessus de l'entity bean.



Le code ci-dessous nous montre comment accéder aux données présentes dans le POJO Employee :

Dans l'exemple ci-dessus nous ne gérons ni les transactions, ni des relations complexes.

**Description de l'environnement de Berkeley DB Java Edition :**

Une base de données BDB JE est composée d'un ou plusieurs fichiers de logs qui sont situés dans le répertoire d'environnement.

Par défaut, leur taille est de 10 Mb et peut être modifiée (par exemple, augmenter la taille des logs permet de limiter le nombre d'I/O).

Le support des transactions :

Pour gérer les transactions avec DPL, il faut faire quelques modifications dans son code.

**Berkeley DB** (BDB) est un [moteur de base de données](http://www.wikiwand.com/fr/Moteur_de_base_de_donn%C3%A9es" \o "). Il est utilisé dans de nombreux environnements, sous [Unix](http://www.wikiwand.com/fr/Unix" \o "Unix), [GNU/Linux](http://www.wikiwand.com/fr/Linux" \o "), [Microsoft Windows](http://www.wikiwand.com/fr/Microsoft_Windows" \o "Microsoft Windows) et dans des systèmes embarqués. Son éditeur revendique 200 millions de déploiements.

## Histoire

Depuis la version 2.0, Berkeley DB est disponible sous deux licences, une libre, certifiée par l'[OSI](http://www.wikiwand.com/fr/Open_Source_Initiative" \o "Open Source Initiative) et une licence commerciale. Les versions précédentes étaient sous [licence BSD](http://www.wikiwand.com/fr/Licence_BSD" \o "Licence BSD).

Berkeley DB est développée par Sleepycat Software qui a été racheté en février 2006 par [Oracle Corporation](http://www.wikiwand.com/fr/Oracle_Corporation" \o "Oracle Corporation)[[2]](http://www.wikiwand.com/fr/Berkeley_DB" \l "citenote2).

## Description

Elle se présente sous la forme d'une [bibliothèque](http://www.wikiwand.com/fr/Biblioth%C3%A8que_logicielle) écrite en [C](http://www.wikiwand.com/fr/C_%28langage%29) et offrant une interface publique ([API](http://www.wikiwand.com/fr/Application_programming_interface)). Des connecteurs existent pour de très nombreux langages de programmation : [C](http://www.wikiwand.com/fr/C_%28langage%29), [C++](http://www.wikiwand.com/fr/C%2B%2B), [Java](http://www.wikiwand.com/fr/Java_%28langage%29), [Perl](http://www.wikiwand.com/fr/Perl_%28langage%29), [Python](http://www.wikiwand.com/fr/Python_%28langage%29), [Tcl](http://www.wikiwand.com/fr/Tcl_%28langage%29), etc.

La base n'est composée que d'[enregistrements](http://www.wikiwand.com/fr/Enregistrement_%28informatique%29) dont le format est librement déterminé par le programme appelant. Il n'y a pas de notion de [table](http://www.wikiwand.com/fr/Table_%28BDD%29), et la base n'est pas interrogeable via un langage de manipulation de données comme [SQL](http://www.wikiwand.com/fr/Structured_Query_Language). Chaque enregistrement est composé d'une paire clé / valeur, la clé n'étant pas unique.

## Fonctionnalités

Ses principales fonctionnalités sont :

* la gestion des transactions ;
* la possibilité de verrouiller des enregistrements ;
* une gestion simplifiée des sauvegardes et de la réplication. On peut effectuer des sauvegardes « à chaud », c.-à-d. sans arrêter la base ;
* la gestion d'un système de cache mémoire interne ;
* elle supporte de grosses capacités de données (jusqu'à 4 Go par enregistrement et 256 To par base) ;
* les données peuvent être chiffrées ([algorithme](http://www.wikiwand.com/fr/Algorithmique" \o "Algorithmique) [AES](http://www.wikiwand.com/fr/Standard_de_chiffrement_avanc%C3%A9" \o "Standard de chiffrement avancé)) ;
* support des transactions [Xa](http://www.wikiwand.com/fr/XA_%28bases_de_donn%C3%A9es%29).

La base de donnée est déclinée en trois versions :

* la version de base ;
* une version optimisée pour Java, utilisée par exemple par [Sun](http://www.wikiwand.com/fr/Sun_Microsystems" \o "Sun Microsystems) dans certains de ses produits ;
* Berkeley DB XML, une surcouche ajoutant des nombreuses fonctionnalités [XML](http://www.wikiwand.com/fr/Extensible_Markup_Language" \o "Extensible Markup Language) comme le support de [XML Query](http://www.wikiwand.com/fr/XML_Query" \o ").

## Applications utilisant BDB

* [OpenLDAP](http://www.wikiwand.com/fr/OpenLDAP), un serveur [LDAP](http://www.wikiwand.com/fr/Lightweight_Directory_Access_Protocol) libre
* [Subversion](http://www.wikiwand.com/fr/Subversion_%28logiciel%29), un système de [gestion de versions](http://www.wikiwand.com/fr/Gestion_de_versions)
* [RPM](http://www.wikiwand.com/fr/RPM_Package_Manager)
* [Asterisk](http://www.wikiwand.com/fr/Asterisk_%28logiciel%29)
* Citadel/UX, une suite collaborative pour le courrier électronique.
* [SQLite](http://www.wikiwand.com/fr/SQLite) un moteur de base de données embarquées