

CavBDM2 - BDD

Généré par Doxygen 1.8.13

Contents

1 Attention

Ce projet a été conçu pour fonctionner avec `make`

Il est recommandé pour les utilisateurs n'aimant pas le terminal de manipuler le projet uniquement avec **eclipse**

Des *builds targets* sont configurés, il suffit de double cliquer dessus pour lancer la commande `make` correspondante (voir rapport)

Commandes

- **make all** : Compile tout les fichiers
- **make test** : Lancement de la série de tests automatiques
- **make doc** : Génération de la documentation (doxygen)
- **make rapport** : Génération du rapport (latex)
- **make clean** : Nettoyage du projet (supression des objets et binaires)
- **make demo-tp1**: Lancer la démo tp1
- **make demo-tp2**: Lancer la démo tp2
- **make rm-rs**: Supprime le fichier `res/RS.txt`

Arborescence

- **bin** : Binaire exécutable
 - **demo** : Exécutable de démonstration
 - **test** : Exécutable de test
- **doc** : Documentation doxygen sous differents formats
- **rapport** : Source du rapport
- **res** : Ressources necessaire au projet (fichier de bdd)
- **script** : Script utilisé pour les test
- **src** : Source du projet
 - **bdd** : Source de la bibliothèque
 - **demo** : Sources des differentes démonstrations d'utilisation
 - **test** : Sources des dufferents tests
- *sujet.pdf* : Sujet du projet
- *README.md* : C'est moi
- *rappot.pdf* : Le rapport du projet

Démonstration disponibles

Les sources de démonstration sont disponibles dans: `src/demo` Les exécutables de test sont générés dans: `bin/demo`
La commande `make` pour lancer les démonstrations est: `make demo-tp1`, `make demo-tp2`, `make demo-tp3`

- `tp1-natural-join`: Natural join R et S
- `tp2-merge-join-without-duplicate`: Merge join sans duplication
- `tp3-merge-join-with-duplicate`: Merge join avec duplication
- `tp4-hash-join`: Hash join

Tests disponibles

Les sources de test sont disponibles dans: `src/test` Les exécutables de test sont générés dans: `bin/test` Le script de test est dans `script/test.sh` La commande `make` pour lancer les tests est: `make test`

- `00-storeFileBuffer`: Ecriture d'un buffer dans un fichier
- `01-natural-join-1`: Natural join R et S
- `02-natural-join-2`: Natural join S et R
- `03-buf-quick-sort`: Fonction de tri d'un buffer
- `04-merge-join-without-duplicate-1`: Merge join sans duplication R et S
- `05-merge-join-without-duplicate-2`: Merge join sans duplication S et R
- `06-merge-join-with-duplicate-1`: Merge join avec duplication R et S
- `07-merge-join-with-duplicate-2`: Merge join avec duplication S et R
- `08-hash-put-equilibre` : Test ajout équilibré dans une table de hash
- `09-hash-put-desequilibre` : Test ajout déséquilibré dans une table de hash
- `10-hash-full` : Test remplissage complet dans une table de hash
- `11-hash-get` : Test récupération d'une entrée dans la table de hash
- `12-hash-remove` : Test suppression / rehash dans une table de hash
- `13-hash-join` : Test hash join

2 Navigation

- **bdd** : Source de la BDD
- **demo** : Exemple d'utilisation de la BDD
- **test** : Test unitaire de la BDD

3 Index des classes

3.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

buf	Structure de donnée représentant un buffer	??
hashset		??
hashtable	Structure de donnée représentant une hashtable	??

4 Index des fichiers

4.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

src/bdd/buffer.c	??
src/bdd/buffer.h	??
src/bdd/hashJoin.c	??
src/bdd/hashJoin.h	??
src/bdd/hashtable.c	??
src/bdd/hashtable.h	??
src/bdd/hexdump.c	??
src/bdd/hexdump.h	??
src/bdd/mergeJoinWithDuplicate.c	??
src/bdd/mergeJoinWithDuplicate.h	??
src/bdd/mergeJoinWithoutDuplicate.c	??
src/bdd/mergeJoinWithoutDuplicate.h	??
src/bdd/naturalJoin.c	??
src/bdd/naturalJoin.h	??
src/bdd/quicksort.c	??
src/bdd/quicksort.h	??
src/demo/tp1-natural-join.c	??
src/demo/tp2-merge-join-without-duplicate.c	??
src/demo/tp3-merge-join-with-duplicate.c	??

src/demo/tp4-hash-join.c	??
src/test/00-storeFileBuffer.c	??
src/test/01-natural-join-1.c	??
src/test/02-natural-join-2.c	??
src/test/03-buf-quick-sort.c	??
src/test/04-merge-join-without-duplicate-1.c	??
src/test/05-merge-join-without-duplicate-2.c	??
src/test/06-merge-join-with-duplicate-1.c	??
src/test/07-merge-join-with-duplicate-2.c	??
src/test/08-hash-put-equilibre.c	??
src/test/09-hash-put-desequilibre.c	??
src/test/10-hash-full.c	??
src/test/11-hash-get.c	??
src/test/12-hash-remove.c	??
src/test/13-hash-join.c	??

5 Documentation des classes

5.1 Référence de la structure buf

Structure de donnée représentant un buffer.

Attributs publics

- `char * v`
Value: Valeur.
- `size_t s`
Size: Taille.
- `size_t c`
Count: Nombre de valeur entrée.

5.1.1 Description détaillée

Structure de donnée représentant un buffer.

5.1.2 Documentation des données membres

5.1.2.1 c

```
size_t buf::c
```

Count: Nombre de valeur entrée.

5.1.2.2 s

```
size_t buf::s
```

Size: Taille.

5.1.2.3 v

```
char* buf::v
```

Value: Valeur.

La documentation de cette structure a été générée à partir du fichier suivant :

- [src/bdd/buffer.c](#)

5.2 Référence de la structure hashset

Attributs publics

- char [key](#)
- char [val](#)

5.2.1 Documentation des données membres

5.2.1.1 key

```
char hashset::key
```

5.2.1.2 val

```
char hashset::val
```

La documentation de cette structure a été générée à partir du fichier suivant :

- [src/bdd/hashtable.c](#)

5.3 Référence de la structure hashtable

Structure de donnée représentant une hashtable.

Graphe de collaboration de hashtable:

Attributs publics

- struct `hashset` * `v`
Value: Valeur (liste de hashset)
- `size_t` `m`
M: Nombre de bucket.
- `size_t` `c`
Count: Nombre de valeur entrée.

5.3.1 Description détaillée

Structure de donnée représentant une hashtable.

5.3.2 Documentation des données membres

5.3.2.1 `c`

```
size_t hashtable::c
```

Count: Nombre de valeur entrée.

5.3.2.2 `m`

```
size_t hashtable::m
```

M: Nombre de bucket.

5.3.2.3 `v`

```
struct hashset* hashtable::v
```

Value: Valeur (liste de hashset)

La documentation de cette structure a été générée à partir du fichier suivant :

- `src/bdd/hashtable.c`

6 Documentation des fichiers

6.1 Référence du fichier README.md

6.2 Référence du fichier src/README.md

6.3 Référence du fichier src/bdd/buffer.c

```
#include "buffer.h"
#include "hexdump.h"
#include "quicksort.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Graphe des dépendances par inclusion de buffer.c:

6.4 Référence du fichier src/bdd/buffer.h

```
#include <stdio.h>
```

Graphe des dépendances par inclusion de buffer.h: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Fonctions

- struct `buf` * `buf_create` (size_t size)
- void `buf_destroy` (struct `buf` *`buf`)
- char `buf_put` (struct `buf` *`buf`, char value)
- void `buf_quicksort` (struct `buf` *`buf`)
 - Quick sort buffer.*
- void `buf_dump` (const struct `buf` *`buf`)
 - Hex dump buffer.*
- struct `buf` * `storeFileBuffer` (const char *file_name, size_t buffer_size)
 - Ouvre un fichier Stoque son contenu dans un buffer Ferme le fichier.*
- char `writeBufferInFile` (const char *file_name, const struct `buf` *)
 - Ouvre un fichier Ecrit le contenu du buffer dans le fichier Ferme le fichier.*
- char `buf_val` (const struct `buf` *`buf`, int index)
 - Retourne la valeur à la position d'un buffer.*
- size_t `buf_count` (const struct `buf` *`buf`)
 - Retourne le nombre de caractere enregistré dans le buffer.*

6.4.1 Documentation des fonctions

6.4.1.1 `buf_count()`

```
size_t buf_count (
    const struct buf * buf )
```

Retourne le nombre de caractere enregistré dans le buffer.

Paramètres

in	<i>buf</i>	Buffer
----	------------	--------

Renvoie

Nombre de caractere enregistré

6.4.1.2 buf_create()

```
struct buf* buf_create (
    size_t size )
```

Creation d'un buffer

Paramètres

in	<i>size</i>	Taille du buffer
----	-------------	------------------

6.4.1.3 buf_destroy()

```
void buf_destroy (
    struct buf * buf )
```

Détruit le buffer

Paramètres

in	<i>buf</i>	buffer à détruire
----	------------	-------------------

6.4.1.4 buf_dump()

```
void buf_dump (
    const struct buf * buf )
```

Hex dump buffer.

Paramètres

in	<i>buf</i>	buffer à trier
----	------------	----------------

6.4.1.5 buf_put()

```
char buf_put (
    struct buf * buf,
    char value )
```

Ajoute un caractere dans le buffer s'il reste de la place

Paramètres

in	<i>buf</i>	buffer d'entrée
in	<i>value</i>	valeur à entrer

Renvoie

0 succès -1 erreur: buffer plein

6.4.1.6 buf_quicksort()

```
void buf_quicksort (
    struct buf * buf )
```

Quick sort buffer.

Paramètres

	<i>buf</i>	Trie le buffer en entrée
in	<i>buf</i>	buffer à trier

6.4.1.7 buf_val()

```
char buf_val (
    const struct buf * buf,
    int index )
```

Retourne la valeur à la position d'un buffer.

Paramètres

in	<i>buf</i>	Buffer
in	<i>index</i>	Index of value

Renvoie

Value

Paramètres

in	<i>buf</i>	Buffer
in	<i>index</i>	Index of value

Renvoie

Value Si -1: Erreur index > count

6.4.1.8 storeFileBuffer()

```
struct buf* storeFileBuffer (
    const char * file_name,
    size_t buffer_size )
```

Ouvre un fichier Stoque son contenu dans un buffer Ferme le fichier.

Paramètres

in	<i>File_name</i>	Fichier
in	<i>buffer_size</i>	Taille du buffer, si trop petit pour contenir le fichier entirerement alors le reste du fichier est ignoré

Renvoie

buffer si NULL alors erreur de lecture du fichier

6.4.1.9 writeBufferInFile()

```
char writeBufferInFile (
    const char * file_name,
    const struct buf * buf )
```

Ouvre un fichier Ecrit le contenu du buffer dans le fichier Ferme le fichier.

Paramètres

in	<i>File_name</i>	Fichier
in	<i>buf</i>	Buffer à écrire dans le fichier

Renvoie

-1 si erreur dans l'ouverture du fichier

6.5 Référence du fichier src/bdd/hashJoin.c

```
#include "hashJoin.h"
```

Graphe des dépendances par inclusion de hashJoin.c:

Fonctions

- void `hash_join` (const struct `hashtable` *ht, const struct `buf` *buf_in, struct `buf` *buf_out)
Hash join.

6.5.1 Documentation des fonctions

6.5.1.1 hash_join()

```
void hash_join (
    const struct hashtable * ht,
    const struct buf * buf_in,
    struct buf * buf_out )
```

Hash join.

TP n°: 4

Titre du TP : Hash Join

Date : 28/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

Paramètres

in	<code>ht</code>	hashtable de la relation a
in	<code>buf_in</code>	relation b
out	<code>buf_out</code>	resultat du hash join avec la relation a et b

6.6 Référence du fichier src/bdd/hashJoin.h

```
#include "buffer.h"
```

```
#include "hashtable.h"
```

Graphe des dépendances par inclusion de hashJoin.h: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Fonctions

- void `hash_join` (const struct `hashtable` *ht, const struct `buf` *buf_in, struct `buf` *buf_out)
Hash join.

6.6.1 Documentation des fonctions

6.6.1.1 hash_join()

```
void hash_join (
    const struct hashtable * ht,
    const struct buf * buf_in,
    struct buf * buf_out )
```

Hash join.

TP n°: 4

Titre du TP : Hash Join

Date : 28/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

Paramètres

in	<code>ht</code>	hashtable de la relation a
in	<code>buf_in</code>	relation b
out	<code>buf_out</code>	resultat du hash join avec la relation a et b

6.7 Référence du fichier src/bdd/hashtable.c

```
#include "buffer.h"
#include "hashtable.h"
#include "hexdump.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Graphe des dépendances par inclusion de hashtable.c:

Classes

- struct `hashset`
- struct `hashtable`

Structure de donnée représentant une hashtable.

Macros

- `#define _GNU_SOURCE`

Fonctions

- `int hash` (`const struct hashtable *ht, char key`)
- `struct hashtable * hashtable_create` (`size_t m`)
- `void hashtable_destroy` (`struct hashtable *ht`)
- `char hashtable_is_full` (`struct hashtable *ht`)
- `char hashtable_put` (`struct hashtable *ht, char key, char val`)
- `char hashtable_get` (`const struct hashtable *ht, char key`)
- `char storeBufferToHashtable` (`struct buf *buf, struct hashtable *ht`)
- `void hashtable_print` (`struct hashtable *ht`)
- `void hashtable_remove` (`struct hashtable *ht, char key`)
- `struct hashtable * storeFileInHashtable` (`const char *filename, size_t m`)

Variables

- `const char VAL_NOT_IN_HASHTABLE = -1`

6.7.1 Documentation des macros

6.7.1.1 _GNU_SOURCE

```
#define _GNU_SOURCE
```

TP n°: 4

Titre du TP : Hash Join

Date : 27/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

6.7.2 Documentation des fonctions

6.7.2.1 hash()

```
int hash (  
    const struct hashtable * ht,  
    char key )
```

Hash fonction (linear)

Paramètres

in	<i>ht</i>	table de hash cible
in	<i>key</i>	clef

6.7.2.2 hashtable_create()

```
struct hashtable* hashtable_create (
    size_t m )
```

Creation d'une hashtable

Paramètres

in	<i>m</i>	Nombre de bucket
----	----------	------------------

6.7.2.3 hashtable_destroy()

```
void hashtable_destroy (
    struct hashtable * ht )
```

Détruit le buffer

Paramètres

in	<i>buf</i>	buffer à détruire
----	------------	-------------------

6.7.2.4 hashtable_get()

```
char hashtable_get (
    const struct hashtable * ht,
    char key )
```

6.7.2.5 hashtable_is_full()

```
char hashtable_is_full (
    struct hashtable * ht )
```

6.7.2.6 hashtable_print()

```
void hashtable_print (
    struct hashtable * ht )
```

6.7.2.7 hashtable_put()

```
char hashtable_put (
    struct hashtable * ht,
    char key,
    char val )
```

6.7.2.8 hashtable_remove()

```
void hashtable_remove (
    struct hashtable * ht,
    char key )
```

6.7.2.9 storeBufferToHashtable()

```
char storeBufferToHashtable (
    struct buf * buf,
    struct hashtable * ht )
```

6.7.2.10 storeFileInHashtable()

```
struct hashtable* storeFileInHashtable (
    const char * filename,
    size_t m )
```

6.7.3 Documentation des variables

6.7.3.1 VAL_NOT_IN_HASHTABLE

```
const char VAL_NOT_IN_HASHTABLE = -1
```

TP n°: 4

Titre du TP : Hash Join

Date : 27/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

6.8 Référence du fichier src/bdd/hashtable.h

```
#include <stdio.h>
```

```
#include "buffer.h"
```

Graphes des dépendances par inclusion de hashtable.h: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Fonctions

- struct `hashtable` * `hashtable_create` (size_t m)
- void `hashtable_destroy` (struct `hashtable` *ht)
- char `hashtable_is_full` (struct `hashtable` *ht)
- char `hashtable_put` (struct `hashtable` *ht, char key, char val)
- char `hashtable_get` (const struct `hashtable` *ht, char key)
- void `hashtable_remove` (struct `hashtable` *ht, char key)
- void `hashtable_print` (struct `hashtable` *ht)
- char `storeBufferToHashtable` (struct `buf` *buf, struct `hashtable` *ht)
- int `hash` (const struct `hashtable` *ht, char key)
- struct `hashtable` * `storeFileInHashtable` (const char *filename, size_t m)

Variables

- const char `VAL_NOT_IN_HASHTABLE`

6.8.1 Documentation des fonctions

6.8.1.1 hash()

```
int hash (
    const struct hashtable * ht,
    char key )
```

Hash fonction (linear)

Paramètres

in	<i>ht</i>	table de hash cible
in	<i>key</i>	clef

6.8.1.2 hashtable_create()

```
struct hashtable* hashtable_create (
    size_t m )
```

Creation d'une hashtable

Paramètres

in	<i>m</i>	Nombre de bucket
----	----------	------------------

6.8.1.3 hashtable_destroy()

```
void hashtable_destroy (
    struct hashtable * ht )
```

Suppression d'une hashtable

Paramètres

in	<i>hashtable</i>	Détruit le buffer
in	<i>buf</i>	buffer à détruire

6.8.1.4 hashtable_get()

```
char hashtable_get (
    const struct hashtable * ht,
    char key )
```

6.8.1.5 hashtable_is_full()

```
char hashtable_is_full (
    struct hashtable * ht )
```

6.8.1.6 hashtable_print()

```
void hashtable_print (
    struct hashtable * ht )
```

6.8.1.7 hashtable_put()

```
char hashtable_put (
    struct hashtable * ht,
    char key,
    char val )
```

6.8.1.8 hashtable_remove()

```
void hashtable_remove (
    struct hashtable * ht,
    char key )
```

6.8.1.9 storeBufferToHashtable()

```
char storeBufferToHashtable (
    struct buf * buf,
    struct hashtable * ht )
```

6.8.1.10 storeFileInHashtable()

```
struct hashtable* storeFileInHashtable (
    const char * filename,
    size_t m )
```

6.8.2 Documentation des variables

6.8.2.1 VAL_NOT_IN_HASHTABLE

```
const char VAL_NOT_IN_HASHTABLE
```

TP n°: 4

Titre du TP : Hash Join

Date : 27/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

6.9 Référence du fichier src/bdd/hexdump.c

```
#include <stdio.h>
```

Grappe des dépendances par inclusion de hexdump.c:

Fonctions

- void [hexDump](#) (char *desc, const void *addr, int len)
- char [printable_or_dot](#) (char c)

6.9.1 Documentation des fonctions

6.9.1.1 hexDump()

```
void hexDump (
    char * desc,
    const void * addr,
    int len )
```

TP n°: 3

Titre du TP : Merge Join Duplicate

Date : 21/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

6.9.1.2 printable_or_dot()

```
char printable_or_dot (
    char c )
```

6.10 Référence du fichier src/bdd/hexdump.h

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Fonctions

- void [hexDump](#) (char *desc, const void *addr, int len)
- char [printable_or_dot](#) (char c)

6.10.1 Documentation des fonctions

6.10.1.1 hexDump()

```
void hexDump (
    char * desc,
    const void * addr,
    int len )
```

TP n°: 3

Titre du TP : Merge Join Duplicate

Date : 21/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

6.10.1.2 printable_or_dot()

```
char printable_or_dot (
    char c )
```

6.11 Référence du fichier src/bdd/mergeJoinWithDuplicate.c

```
#include "mergeJoinWithDuplicate.h"
#include "buffer.h"
```

Graphe des dépendances par inclusion de mergeJoinWithDuplicate.c:

Fonctions

- void `merge_join_with_duplicate` (const struct `buf` *buf_a, const struct `buf` *buf_b, struct `buf` *buf_out)
Merge join with duplicate.

6.11.1 Documentation des fonctions

6.11.1.1 merge_join_with_duplicate()

```
void merge_join_with_duplicate (
    const struct buf * buf_a,
    const struct buf * buf_b,
    struct buf * buf_out )
```

Merge join with duplicate.

TP n°: 3

Titre du TP : Merge Join Duplicate

Date : 21/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

Paramètres

in	<code>buf_a</code>	relation a
in	<code>buf_b</code>	relation b
out	<code>buf_out</code>	resultat du merge join avec duplication de la relation a et b

6.12 Référence du fichier src/bdd/mergeJoinWithDuplicate.h

```
#include "buffer.h"
```

Graphe des dépendances par inclusion de mergeJoinWithDuplicate.h: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Fonctions

- void `merge_join_with_duplicate` (const struct `buf` *buf_a, const struct `buf` *buf_b, struct `buf` *buf_out)
Merge join with duplicate.

6.12.1 Documentation des fonctions

6.12.1.1 merge_join_with_duplicate()

```
void merge_join_with_duplicate (
    const struct buf * buf_a,
    const struct buf * buf_b,
    struct buf * buf_out )
```

Merge join with duplicate.

TP n°: 3

Titre du TP : Merge Join Duplicate

Date : 21/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

Paramètres

in	<code>buf_a</code>	relation a
in	<code>buf_b</code>	relation b
out	<code>buf_out</code>	resultat du merge join avec duplication de la relation a et b

6.13 Référence du fichier src/bdd/mergeJoinWithoutDuplicate.c

```
#include "mergeJoinWithoutDuplicate.h"
#include "buffer.h"
```

Graphe des dépendances par inclusion de mergeJoinWithoutDuplicate.c:

Fonctions

- void `merge_join_without_duplicate` (const struct `buf` *buf_a, const struct `buf` *buf_b, struct `buf` *buf_out)
Merge join without duplicate.

6.13.1 Documentation des fonctions

6.13.1.1 `merge_join_without_duplicate()`

```
void merge_join_without_duplicate (
    const struct buf * buf_a,
    const struct buf * buf_b,
    struct buf * buf_out )
```

Merge join without duplicate.

TP n°: 3

Titre du TP : Merge Join Duplicate

Date : 21/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

Paramètres

in	<code>buf_a</code>	relation a
in	<code>buf_b</code>	relation b
out	<code>buf_out</code>	resultat du merge_join de la relation a et b

6.14 Référence du fichier `src/bdd/mergeJoinWithoutDuplicate.h`

```
#include "buffer.h"
```

Graphe des dépendances par inclusion de `mergeJoinWithoutDuplicate.h`: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Fonctions

- void `merge_join_without_duplicate` (const struct `buf` *buf_a, const struct `buf` *buf_b, struct `buf` *buf_out)
Merge join without duplicate.

6.14.1 Documentation des fonctions

6.14.1.1 merge_join_without_duplicate()

```
void merge_join_without_duplicate (
    const struct buf * buf_a,
    const struct buf * buf_b,
    struct buf * buf_out )
```

Merge join without duplicate.

TP n°: 3

Titre du TP : Merge Join Duplicate

Date : 21/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

Paramètres

in	<i>buf_a</i>	relation a
in	<i>buf_b</i>	relation b
out	<i>buf_out</i>	resultat du merge_join de la relation a et b

6.15 Référence du fichier src/bdd/naturalJoin.c

```
#include "naturalJoin.h"
```

```
#include "buffer.h"
```

Graphe des dépendances par inclusion de naturalJoin.c:

Fonctions

- void `natural_join` (const struct `buf` *buf_a, const struct `buf` *buf_b, struct `buf` *buf_out)

6.15.1 Documentation des fonctions

6.15.1.1 natural_join()

```
void natural_join (
    const struct buf * buf_a,
    const struct buf * buf_b,
    struct buf * buf_out )
```

TP n°: 3

Titre du TP : Merge Join Duplicate

Date : 21/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques : Natual join

Paramètres

in	<i>buf_a</i>	relation externe
in	<i>buf_b</i>	relation interne
out	<i>buf_out</i>	resultat du natural join de buf_a et buf_b

6.16 Référence du fichier src/bdd/naturalJoin.h

```
#include "buffer.h"
```

Graphe des dépendances par inclusion de naturalJoin.h: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Fonctions

- void `natural_join` (const struct `buf` *buf_a, const struct `buf` *buf_b, struct `buf` *buf_out)

6.16.1 Documentation des fonctions

6.16.1.1 `natural_join()`

```
void natural_join (
    const struct buf * buf_a,
    const struct buf * buf_b,
    struct buf * buf_out )
```

TP n°: 3

Titre du TP : Merge Join Duplicate

Date : 21/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques : Natual join

Paramètres

in	<i>buf_a</i>	relation externe
in	<i>buf_b</i>	relation interne
out	<i>buf_out</i>	resultat du natural join de buf_a et buf_b

6.17 Référence du fichier src/bdd/quicksort.c

```
#include <stdio.h>
#include <stdlib.h>
```

Graphe des dépendances par inclusion de quicksort.c:

Fonctions

- void `swap` (char *x, char *y)
- int `choose_pivot` (char i, char j)
- void `quicksort` (char *list, int m, int n)

6.17.1 Documentation des fonctions

6.17.1.1 `choose_pivot()`

```
int choose_pivot (
    char i,
    char j )
```

6.17.1.2 `quicksort()`

```
void quicksort (
    char * list,
    int m,
    int n )
```

6.17.1.3 `swap()`

```
void swap (
    char * x,
    char * y )
```

TP n°: 3

Titre du TP : Merge Join Duplicate

Date : 21/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

6.18 Référence du fichier src/bdd/quicksort.h

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Fonctions

- void `quicksort` (char *list, char m, char n)

6.18.1 Documentation des fonctions

6.18.1.1 quicksort()

```
void quicksort (
    char * list,
    char m,
    char n )
```

TP n°: 3

Titre du TP : Merge Join Duplicate

Date : 21/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques : Remarques : Ce fichier a été récupéré du site: <http://www.zentut.com/c-tutorial/c-quicksort-algo>
il a été réadapté dans le cadre du projet

6.19 Référence du fichier src/demo/tp1-natural-join.c

```
#include "../bdd/naturalJoin.h"
```

Graphe des dépendances par inclusion de tp1-natural-join.c:

Fonctions

- int `main` (int argc, char **argv)

6.19.1 Documentation des fonctions

6.19.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

6.20 Référence du fichier src/demo/tp2-merge-join-without-duplicate.c

```
#include "../bdd/mergeJoinWithoutDuplicate.h"
```

Graphe des dépendances par inclusion de tp2-merge-join-without-duplicate.c:

Fonctions

- int [main](#) (int argc, char **argv)

6.20.1 Documentation des fonctions

6.20.1.1 main()

```
int main (  
    int argc,  
    char ** argv )
```

6.21 Référence du fichier src/demo/tp3-merge-join-with-duplicate.c

```
#include "../bdd/mergeJoinWithDuplicate.h"
```

Graphe des dépendances par inclusion de tp3-merge-join-with-duplicate.c:

Fonctions

- int [main](#) (int argc, char **argv)

6.21.1 Documentation des fonctions

6.21.1.1 main()

```
int main (  
    int argc,  
    char ** argv )
```

6.22 Référence du fichier src/demo/tp4-hash-join.c

```
#include "../bdd/hashJoin.h"
```

Graphe des dépendances par inclusion de tp4-hash-join.c:

Fonctions

- int [main](#) (int argc, char **argv)

6.22.1 Documentation des fonctions

6.22.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

6.23 Référence du fichier src/test/00-storeFileBuffer.c

```
#include "../bdd/buffer.h"
```

Graphe des dépendances par inclusion de 00-storeFileBuffer.c:

Fonctions

- int [main](#) (int argc, char **argv)

6.23.1 Documentation des fonctions

6.23.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

6.24 Référence du fichier src/test/01-natural-join-1.c

```
#include "../bdd/naturalJoin.h"
```

Graphe des dépendances par inclusion de 01-natural-join-1.c:

Fonctions

- int [main](#) (int argc, char **argv)

6.24.1 Documentation des fonctions

6.24.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

6.25 Référence du fichier src/test/02-natural-join-2.c

```
#include "../bdd/naturalJoin.h"
```

Graphe des dépendances par inclusion de 02-natural-join-2.c:

Fonctions

- int [main](#) (int argc, char **argv)

6.25.1 Documentation des fonctions

6.25.1.1 main()

```
int main (  
    int argc,  
    char ** argv )
```

6.26 Référence du fichier src/test/03-buf-quick-sort.c

```
#include "../bdd/buffer.h"
```

Graphe des dépendances par inclusion de 03-buf-quick-sort.c:

Fonctions

- int [main](#) (int argc, char **argv)

6.26.1 Documentation des fonctions

6.26.1.1 main()

```
int main (  
    int argc,  
    char ** argv )
```

6.27 Référence du fichier src/test/04-merge-join-without-duplicate-1.c

```
#include "../bdd/mergeJoinWithoutDuplicate.h"
```

Graphe des dépendances par inclusion de 04-merge-join-without-duplicate-1.c:

Fonctions

- int [main](#) (int argc, char **argv)

6.27.1 Documentation des fonctions

6.27.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

6.28 Référence du fichier src/test/05-merge-join-without-duplicate-2.c

```
#include "../bdd/mergeJoinWithoutDuplicate.h"
```

Graphe des dépendances par inclusion de 05-merge-join-without-duplicate-2.c:

Fonctions

- int [main](#) (int argc, char **argv)

6.28.1 Documentation des fonctions

6.28.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

6.29 Référence du fichier src/test/06-merge-join-with-duplicate-1.c

```
#include "../bdd/mergeJoinWithDuplicate.h"
```

Graphe des dépendances par inclusion de 06-merge-join-with-duplicate-1.c:

Fonctions

- int [main](#) (int argc, char **argv)

6.29.1 Documentation des fonctions

6.29.1.1 main()

```
int main (
    int argc,
    char ** argv )
```


6.30 Référence du fichier src/test/07-merge-join-with-duplicate-2.c

```
#include "../bdd/mergeJoinWithDuplicate.h"
```

Graphe des dépendances par inclusion de 07-merge-join-with-duplicate-2.c:

Fonctions

- int [main](#) (int argc, char **argv)

6.30.1 Documentation des fonctions

6.30.1.1 main()

```
int main (  
    int argc,  
    char ** argv )
```

6.31 Référence du fichier src/test/08-hash-put-equilibre.c

```
#include "../bdd/hashtable.h"
```

Graphe des dépendances par inclusion de 08-hash-put-equilibre.c:

Fonctions

- int [main](#) (int argc, char **argv)

6.31.1 Documentation des fonctions

6.31.1.1 main()

```
int main (  
    int argc,  
    char ** argv )
```

TP n°: 4

Titre du TP : Hash Join

Date : 27/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

6.32 Référence du fichier src/test/09-hash-put-desequilibre.c

```
#include "../bdd/hashtable.h"
```

Graphe des dépendances par inclusion de 09-hash-put-desequilibre.c:

Fonctions

- int `main` (int argc, char **argv)

6.32.1 Documentation des fonctions

6.32.1.1 main()

```
int main (  
    int argc,  
    char ** argv )
```

TP n°: 4

Titre du TP : Hash Join

Date : 27/10/17

Nom : Lefranc Prenom : Joaquim email : `lefrancjoaquim@gmail.com`

Nom : Skoda Prenom : Jérôme email : `contact@jeromeskoda.fr`

Remarques :

6.33 Référence du fichier src/test/10-hash-full.c

```
#include "../bdd/hashtable.h"
```

Graphe des dépendances par inclusion de 10-hash-full.c:

Fonctions

- int `main` (int argc, char **argv)

6.33.1 Documentation des fonctions

6.33.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

TP n°: 4

Titre du TP : Hash Join

Date : 27/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

6.34 Référence du fichier src/test/11-hash-get.c

```
#include "../bdd/hashtable.h"
```

Graphe des dépendances par inclusion de 11-hash-get.c:

Fonctions

- int [main](#) (int argc, char **argv)

6.34.1 Documentation des fonctions

6.34.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

TP n°: 4

Titre du TP : Hash Join

Date : 27/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

6.35 Référence du fichier src/test/12-hash-remove.c

```
#include "../bdd/hashtable.h"
```

Graphe des dépendances par inclusion de 12-hash-remove.c:

Fonctions

- int `main` (int argc, char **argv)

6.35.1 Documentation des fonctions

6.35.1.1 main()

```
int main (  
    int argc,  
    char ** argv )
```

TP n°: 4

Titre du TP : Hash Join

Date : 27/10/17

Nom : Lefranc Prenom : Joaquim email : lefrancjoaquim@gmail.com

Nom : Skoda Prenom : Jérôme email : contact@jeromeskoda.fr

Remarques :

6.36 Référence du fichier src/test/13-hash-join.c

```
#include "../bdd/hashJoin.h"
```

Graphe des dépendances par inclusion de 13-hash-join.c:

Fonctions

- int `main` (int argc, char **argv)

6.36.1 Documentation des fonctions

6.36.1.1 main()

```
int main (  
    int argc,  
    char ** argv )
```

