API Documentation

API Documentation

November 21, 2016

Contents

1 Module application

1.1 Functions

parse_input(*cmd*, *table*)

parse_insert(*values*, *table*)

parse_select(*values*, *table*)

parse_update(*values*, *table*)

parse_delete(*value*, *table*)

parse_btree(*table*)

2 Module btree

2.1 Class BTree

2.1.1 Methods

init(*self*)

pretty_print(*self*, *curr_dblock*=`None`)

has_key(*self*, *key_value*)

find_key(*self*, *key_value*)

insert(*self*, *key_value*, *rowid*, *curr_dblock*=`None`)

split_during_insert(*self*, *splited_data*, *src_dblock*, *datablock_type*)

update(*self*, *key_value*, *rowid*)

delete(*self*, *key_value*)

new_root(*self*, *left_dblock*, *right_dblock*)

2.1.2 Class Variables

Name	Description
root	**Value:** `attr.ib()`
buffer	**Value:** `attr.ib()`

3 Module buffer

3.1 Variables

Name	Description
BUFFER_SIZE	**Value:** `256`

3.2 Class Buffer

3.2.1 Methods

init(*cls*, *datafile*)

Creates a new Datablock in memory from a string of bytes

flush(*self*)

new_datablock(*self*, *datablock_type*, *address*)

get_datablock_in_buffer(*self*, *address*)

is_datablock_in_buffer(*self*, *address*)

get_datablock(*self*, *address*)

search_dblock_with_free_space(*self*, *free_space*, *datablock_type*)

Load in the buffer the first data block with the space needed

linear_search_record(*self*, *datablock_type*, *value*, *field*=`None`, *unique*=`False`)

Search datablocks to contain a spacific value

get_datablock_free_space(*self*, *dblock*, *free_space*, *datablock_type*)

get_next_empty_datablock(*self*, *address*=`0`)

3.2.2 Class Variables

Name	Description
datafile	**Value:** `attr.ib()`

4 Module config_datablock

4.1 Class ConfigDatablock

4.1.1 Methods

get_data(*self*)

Convert header and records to bytes

Overrides: datablock.Datablock.get_data

records_size(*self*)

update_btree_root(*self*, *new_addr*)

from_bytes(*cls*, *address*, *data*=`None`, *count_record*=`0`)

Creates a new TableDatablock in memory from a string of bytes

Overrides: datablock.Datablock.from_bytes

unpack(*count_record*, *data*)

Inherited from datablock.Datablock(Section 5.1)

delete(), empty()

4.1.2 Class Variables

Name	Description
btree_root	**Value:** `attr.ib(default= 16300)`
Inherited from datablock.Datablock (Section 5.1) DATABLOCK_SIZE, address, count_record, deleted, type	

5 Module datablock

5.1 Class Datablock

5.1.1 Methods

get_data(*self*)

delete(*self*)

empty(*self*)

from_bytes(*cls*, *address*, *data*, *count_record*)

Creates a new Datablock in memory from a string of bytes

5.1.2 Class Variables

Name	Description
address	**Value:** `attr.ib()`
type	**Value:** `attr.ib(default= 0)`
count_record	**Value:** `attr.ib(default= 0)`
deleted	**Value:** `attr.ib(default= False)`
DATABLOCK_SIZE	**Value:** `2* 1024`

6 Module datafile

6.1 Class Datafile

6.1.1 Methods

create_new(*self*)

get_datablock(*self*, *address*)

write_datablock(*self*, *dblock*)

next_available_datablock(*self*, *address*=0)

datablocks(*self*)

node_datablocks(*self*)

new_datablock(*self*, *datablock_type*, *address*)

6.1.2 Class Variables

Name	Description
filename	**Value:** `attr.ib()`
filesize	**Value:** `32* 1024* 1024`
NUM_DATABLOCKS	**Value:** `filesize/ Datablock.DATABLOCK_SIZE`

7 Module leaf_datablock

7.1 Class LeafDatablock

datablock.Datablock
└ **leaf_datablock.LeafDatablock**

7.1.1 Methods

get_data(*self*)

Convert header and records to bytes Format: TypeCountHeaderRecords

Overrides: datablock.Datablock.get_data

from_bytes(*cls*, *address*, *data*=`None`, *count_record*=`0`)

Creates a new TableDatablock in memory from a string of bytes

Overrides: datablock.Datablock.from_bytes

unpack(*count_record*, *data*)

find_key(*self*, *key_value*)

free_space(*self*)

can_insert(*self*)

insert(*self*, *key_value*, *rowid*)

insert_and_split(*self*, *key_value*, *rowid*)

Insert and split leaf in the btree

update_rowid(*self*, *key_value*, *rowid*)

delete(*self*, *key_value*)

Overrides: datablock.Datablock.delete

update_data(*self*, *keys*=`[]`, *rowids*=`[]`)

Inherited from datablock.Datablock(Section 5.1)

empty()

7.1.2 Class Variables

Name	Description
rowids	**Value:** `attr.ib(default= [])`
keys	**Value:** `attr.ib(default= [])`
Inherited from datablock.Datablock (Section 5.1) DATABLOCK_SIZE, address, count_record, deleted, type	

8 Module node_datablock

8.1 Class NodeDatablock

datablock.Datablock
└ **node_datablock.NodeDatablock**

8.1.1 Methods

get_data(*self*)

Convert header and records to bytes Format: TypeCountHeaderRecords

Overrides: datablock.Datablock.get_data

from_bytes(*cls*, *address*, *data*=`None`, *count_record*=`0`)

Creates a new TableDatablock in memory from a string of bytes

Overrides: datablock.Datablock.from_bytes

unpack(*count_record*, *data*)

find_key(*self*, *key_value*)

update_data(*self*, *keys*=`[]`, *nexts*=`[]`)

free_space(*self*)

can_insert(*self*)

insert(*self*, *key_value*, *left_addr*, *right_addr*)

insert_and_split(*self*, *key_value*, *left_addr*, *right_addr*)

Insert and split node in the btree

Inherited from datablock.Datablock(Section 5.1)

delete(), empty()

8.1.2 Class Variables

Name	Description
nexts	**Value:** `attr.ib(default= [])`
keys	**Value:** `attr.ib(default= [])`
Inherited from datablock.Datablock (Section 5.1) DATABLOCK_SIZE, address, count_record, deleted, type	

9 Module record

9.1 Functions

len_less_than_200(*instance*, *attribute*, *value*)

9.2 Class Record

9.2.1 Methods

pack(*self*)

writeble_size(*self*, *str_obj*)

size(*self*)

9.2.2 Class Variables

Name	Description
code	**Value:** `attr.ib(validator= attr.validators.instance_of(int))`
description	**Value:** `attr.ib(validator= len_less_than_200)`
rowid	**Value:** `attr.ib(default= None)`
deleted	**Value:** `attr.ib(default= False)`

10 Module rowid

10.1 Class Rowid

Rowid points to a record in a datablock

10.1.1 Methods

get_record(*self*)

Returns a Record from the Rowid

pack(*self*)

10.1.2 Class Variables

Name	Description
dblock	**Value:** `attr.ib()`
pos	**Value:** `attr.ib(validator= attr.validators.instance_of(int))`

11 Module table

11.1 Class Table

11.1.1 Methods

init(*cls*, *datafile*)

insert(*self*, *code*, *desc*)

Inserts code and desc into table

select_code(*self*, *code*)

Finds record with code Uses btree for index

select_desc(*self*, *desc*)

Finds record by description Can't use btree

update(*self*, *code*, *desc*)

Updates record code with new description desc Finds record through select_code()

delete(*self*, *code*)

Deletes record by code Finds record through select_code()

exit(*self*)

11.1.2 Class Variables

Name	Description
BTREE_ROOT_DEFAULT	**Value:** `16300`
buffer	**Value:** `attr.ib(validator= attr.validators.instance_of(Buffer))`
btree	**Value:** `attr.ib(default= None)`

12 Module table_datablock

12.1 Class TableDatablock

12.1.1 Methods

get_data(*self*)

Convert header and records to bytes

Overrides: datablock.Datablock.get_data

save_record(*self*, *record*)

Saves a Record to the datablock

records_size(*self*)

free_contiguous_space(*self*, *space_needed*)

write_data(*self*, *record*, *position*=`None`)

update_record(*self*, *record*, *desc*)

delete_record(*self*, *record*)

search_by(*self*, *value*, *field*)

get_record_by_pos(*self*, *position*)

Get specific record by its position

from_bytes(*cls*, *address*, *data*=`None`, *count_record*=`0`)

Creates a new TableDatablock in memory from a string of bytes

Overrides: datablock.Datablock.from_bytes

unpack(*count_record*, *data*)

unpack_records(*record_str*, *header*, *address*)
Returns a list of Records included in the datablock

Inherited from datablock.Datablock(Section 5.1)

delete(), empty()

12.1.2 Class Variables

Name	Description
header	**Value:** `attr.ib(default= [])`
records	**Value:** `attr.ib(default= [])`
Inherited from datablock.Datablock (Section 5.1) DATABLOCK_SIZE, address, count_record, deleted, type	

Index