



# Demonstration

Εργασία Μαθήματος Σ. Δ. Β. Δ.

ΠΕΤΡΟΣ-ΕΥΣΤΑΘΙΟΣ ΦΑΤΟΥΡΟΣ

ΑΝΤΩΝΙΟΣ ΤΖΙΟΡΒΑΣ Π18153

ΜΑΡΙΟΣ-ΙΩΑΝΝΗΣ ΣΤΑΜΑΤΟΠΟΥΛΟΣ Π18144



Τρέχοντας το αρχείο demo.py προκύπτει το εξής (επεξηγήσεις στο παρακάτω screenshot):

```
petrasv@PetrasVM-VirtualBox:~/minidb$ python3 -i demo.py
New table "meta_length"
New table "meta_locks"
New table "meta_insert_stack"
New table "meta_indexes"
New table "superheroes"
Deleted 1 rows
Deleted 1 rows
Deleted 1 rows
Deleted 1 rows
Deleted 1 rows
limit was reached (limit = 5)
delete all records from insert stack and insert them into the main file
--- Impostor exists. Record was not inserted.
New table "celebrities"
---Custom database created---

----Database "order"----
savedir: "dbdata/order_db"

## meta_insert_stack ##
table_name (str)  indexes (list)
-----
superheroes      [1]
celebrities      []

--Table "superheroes"--
.no_of_columns: 3
Column_names: ['heroname', 'realname', 'age']
pk_idx: 0
.file_organization_type: "sequential"
columns: [['Deadpool', None, 'Hawkeye', 'Wolverine', 'Impostor'], ['Made_Wilson', None, 'Clint_Barton', 'James_Howlett', 'Clint_Barton'], [50, None, 38, 197, 38]]
data: [['Deadpool', 'Made_Wilson', 50], [None, None, None], ['Hawkeye', 'Clint_Barton', 38], ['Wolverine', 'James_Howlett', 197], ['Impostor', 'Clint_Barton', 38]]
sequential_file_columns: [['Batman', 'Captain_America', 'Hulk', 'Iron_Man', 'Spiderman'], ['Bruce_Wayne', 'Steve_Rogers', 'Bruce_Banner', 'Tony_Stark', 'Peter_Parker'], [26, 66, 43, 53, 28]]
sequential_file_data: [['Batman', 'Bruce_Wayne', 26], ['Captain_America', 'Steve_Rogers', 66], ['Hulk', 'Bruce_Banner', 43], ['Iron_Man', 'Tony_Stark', 53], ['Spiderman', 'Peter_Parker', 28]]
order: [[0, 'Batman'], [1, 'Captain_America'], [2, 'Deadpool'], [3, 'Hawkeye'], [4, 'Hulk'], [5, 'Impostor'], [6, 'Iron_Man'], [7, 'Spiderman'], [8, 'Wolverine']]

--Table "celebrities"--
.no_of_columns: 2
Column_names: ['fullname', 'profession']
pk_idx: 0
.file_organization_type: "heap"
columns: [['Tom_Cruise', 'Ariana_Grande', 'James_Charles', 'Johnny_Depp', 'Nicki_Minaj'], ['movie_actor', 'pop_singer', 'instagram_star', 'movie_actor', 'rapper']]
data: [['Tom_Cruise', 'movie_actor'], ['Ariana_Grande', 'pop_singer'], ['James_Charles', 'instagram_star'], ['Johnny_Depp', 'movie_actor'], ['Nicki_Minaj', 'rapper']]

>>> 
```

Δημιουργήθηκε η ΒΔ *orderDB* και οι πίνακες *superheroes* και *celebrities*.

Ακολουθούν μερικές εντολές *select* για τον πίνακα *superheroes* (διατεταγμένη διάταξη).

```
>>> select_O(orderDB, "superheroes", '*', condition=None, order_by=None, asc=False, top_k=None, save_as=None, return_object=False)
## superheroes ##
heroname (str) #PK#  realname (str)  age (int)
-----
Batman             Bruce_Wayne  26
Captain_America    Steve_Rogers 66
Hulk                Bruce_Banner 43
Iron_Man            Tony_Stark   53
Spiderman           Peter_Parker 28
Deadpool            Wade_Wilson 50
Hawkeye             Clint_Barton 38
Wolverine           James_Howlett 197
Impostor            Clint_Barton 38

>>> select_O(orderDB, "superheroes", '*', condition="realname==Clint_Barton", order_by=None, asc=False, top_k=None, save_as=None, return_object=False)
## superheroes ##
heroname (str) #PK#  realname (str)  age (int)
-----
Hawkeye           Clint_Barton    38
Impostor           Clint_Barton    38

>>> select_O(orderDB, "superheroes", '*', condition="heroname==Iron_Man", order_by=None, asc=False, top_k=None, save_as=None, return_object=False)
## superheroes ##
heroname (str) #PK#  realname (str)  age (int)
-----
Iron_Man          Tony_Stark       53

>>> select_O(orderDB, "superheroes", '*', condition="age>50", order_by=None, asc=False, top_k=None, save_as=None, return_object=False)
## superheroes ##
heroname (str) #PK#  realname (str)  age (int)
-----
Captain_America    Steve_Rogers    66
Iron_Man            Tony_Stark       53
Wolverine           James_Howlett    197

>>> select_O(orderDB, "superheroes", '*', condition="heroname>Iron", order_by=None, asc=False, top_k=None, save_as=None, return_object=False)
## superheroes ##
heroname (str) #PK#  realname (str)  age (int)
-----
Iron_Man            Tony_Stark       53
Spiderman            Peter_Parker     28
Wolverine            James_Howlett    197
```

- Στο select all οι εγγραφές εμφανίζονται σε διατεταγμένη σειρά
- Στις ερωτήσεις ταυτότητας και διαστήματος πάνω στο κλειδί οι εγγραφές εμφανίζονται σε διατεταγμένη σειρά
- Στις ερωτήσεις ταυτότητας και διαστήματος εκτός κλειδιού οι εγγραφές εμφανίζονται με την σειρά που βρίσκονται στο αρχείο

Ακολουθούν μερικές εντολές *select* για τον πίνακα *celebrities* (διάταξη σωρού).

```
>>> select_O(orderDB, "celebrities", '*', condition=None, order_by=None, asc=False, top_k=None, save_as=None, return_object=False)

## celebrities ##
fullname (str) #PK#      profession (str)
-----
Tom_Cruise             movie_actor
Ariana_Grande           pop_singer
James_Charles           instagram_star
Johnny_Depp             movie_actor
Nicki_Minaj             rapper

>>> select_O(orderDB, "celebrities", '*', condition=None, order_by="profession", asc=True, top_k=3, save_as=None, return_object=False)

## celebrities ##
fullname (str) #PK#      profession (str)
-----
James_Charles           instagram_star
Tom_Cruise             movie_actor
Ariana_Grande           pop_singer

>>> select_O(orderDB, "celebrities", '*', condition="profession==movie_actor", order_by=None, asc=False, top_k=None, save_as=None, return_object=False)

## celebrities ##
fullname (str) #PK#      profession (str)
-----
Tom_Cruise             movie_actor
Johnny_Depp            movie_actor

>>> █
```