

■ Επιλογή Δομών:

- **Tuple**: Αυτή η δομή περιέχει το στοιχείο και σε ποιά γραμμή βρίσκεται (row_id).
- **Shell**: Είναι μία δομή που “κρατάει” τον αριθμό των γραμμων και στηλών των στοιχείων και έναν πίνακα με δείκτες σε Tuple (καθε αρχείο αντιστοιχεί σε ένα Shell).
- **Table**: Είναι μία δομή που “κρατάει” τον αριθμό των Shells που περιέχει και έναν πίνακα με δείκτες σε Shell.
- **Query**: Είναι μία δομή που περιέχει 3 strings (relations, predicates, projections) και έναν δείκτη σε επόμενο query.
- **Batch**: Είναι μία δομή που “κρατάει” έναν δείκτη στο πρώτο query του batch (δηλαδή στον 1ο κόμβο της ουράς), έναν στο τελευταίο (έτσι αποφεύγουμε την προσπέλαση, όταν θέλουμε να προσθέσουμε έναν κόμβο) και το πόσα queries έχει αυτο το batch.
- **Parsed Query**: Είναι μία δομή που περιέχει έναν πίνακα με relations, έναν με joins, έναν με filters κι έναν με projections.
- **Filter Result**: Είναι μία δομή που “κρατάει” έναν πίνακα με row_ids, την σχέση που φιλτραραμε και τον αριθμό των αποτελεσμάτων.
- **Filter Outcome**: Είναι μία δομή που “κρατάει” έναν πίνακα δείκτες σε Filter_Result και τον αριθμό των φίλτρων που εκτελέσαμε.
- **Execution Queue Node**: Είναι μία δομή που περιέχει έναν δείκτη σε Join και έναν δείκτη σε επόμενο κομβο-join.
- **Execution Queue**: Έχει έναν δείκτη head που δείχνει στον 1ο κόμβο και έναν tail που δείχνει στον τελευταίο της.

• Initializer:

- **Argument_Manager**:
 - *Check_Arguments_Number*:
 - Ελέγχει για τον αριθμό των args.
 - *Go_Through_Argv_And_Get_Filenames*:
 - Ελέγχει ότι τα flags ότι έχουν δοθεί σωστά και αποθηκεύει τα ονόματα των αρχείων για διάβασμα.
 - *Get_Argument_Data*:
 - Αφού κληθεί η *Check_Arguments_Number* και η *Go_Through_Argv_And_Get_Filenames*, δημιουργείται μια δομή, *ArgumentData*, που περιέχει τα ονόματα των input-files.

- **Table_Allocator:**
 - *Create_Table_Allocator:*
 - Δημιουργεί μια δομή που αποθηκεύει το directory και το file που περιεχει τα ονοματα των αρχείων-σχεσεων.
 - *Allocate_Table:*
 - Καλεί την Count_Shells για να ξέρει ποσα ειναι τα αρχεία-σχέσεις και δημιουργεί μια δομή Table.
 - *Fill_Table:*
 - Ανοίγει το Init αρχείο για διάβασμα και για καθε αρχειο-σχεση καλεί την Fill_Shell, η οποία κανει allocate και γεμίζει το shell με τα δεδομένα.
 - Ανοίγει το αρχείο για διάβασμα, καλεί την Read_from_File για να διαβασει ποσες γραμμες και στήλες ...για να “γεμίσει” την σχέση απο το αρχειο και κλείνει τον fp που είχε ανοίξει.
- **Work_Reader:**
 - *Read_next_Batch:*
 - Δημιουργεί ενα Batch, διαβάζει γραμμή-γραμμή τα queries και τα εισάγει στην λιστα.
- **Work_Executor:**
 - *Start_Work:*
 - Ανοίγει το work αρχειο για διαβάζει batch-batch, καλώντας την Read_next_Batch, οταν διαβασει ενα batch αρχίζει κ κάνει pop ενα-ενα τα queries για να τα εκτελέσει.
- **Query_execution:**
 - **Query_parser:**
 - *Parse_Query:*
 - Δημιουργεί μια Parsed_Query δομή και την γεμίζει καλώντας τις Setup_Relations, Setup_Joins_and_Filters, Setup_Projections.
 - **Preparator:**
 - *Prepare_Execution_Queue:*
 - Δημιουργεί μια Execution_Queue, ελέγχει για Self_Joins (και τα βάζει πρώτα αν βρει), ελεγχκει για Joins μεταξύ ίδιων στηλών (ομοίως αν βρει τα εισαγει στην ουρά) και τελος εισάγει και οργανώνει τα υπολοιπα.
 - **Query_executor:**
 - **Filter_executor:**
 - *Execute:*
 - Προσπελαύνει μια φορα τον πίνακα με τα δεδομένα της σχέσης που θελουμε και κραταμε το row_id οποιας γραμμής πληρει το φίλτρο (στην στήλη που θέλουμε).
 - *Execute_Filters:*

- Κανει allocate εναν πινακα με δείκτες σε Filter_Result και για καθε φιλτρο σε ένα query: βρισκει την σχέση που πρέπει και καλεί την Execute. Τελος, κανει allocate μια δομη Filter_Outcome, βαζει τον δείκτη Filter_Outcome->Filter_Result να δείχνει στον πινακα με τα απολεσματα μας και αποθηκεύει στο Filter_Outcome->num_of_filters ποσα φιλτρα εκτελεσαμε.
- **Join_executor:**
 -
- **Query_executor:**
 - *Execute_Query:*
 - Κανει parse το Query και αποθηκεύει με ποιες σχέσεις θα δουλεψουμε στο συγκεκριμενο query, καλεί την Prepare_Execution_Queue για να ετοιμάσει με ποια σειρά θα εκτελεστούν τα joins, καλεί την Execute_Filters (καθως τα φιλτρα θα εκτελεστουν πρωτα και δεν μας νοιαζει η σειρά) και τελος
- **main:**
 - Αρχικά δημιουργεί τον ArgManager και αρχικοποιεί το Table καλώντας τις Create_Table_Allocator, Allocate_Table και Fill_Table.
 - Στην συνέχεια, καλεί την Start_Work.
 - Τέλος, αποδεσμεύει την μνήμη που έχει δεσμεύσει για όλες τις δομές.