Data Base System Implementation
COP-6726

Submitted by:
Naman Arora
UFID: 3979-0439
Nikunj Sarda
UFID: 9360-6581

Introduction:

This document is intended to present the outcomes and the process of the completed work for the Project 4_2. The query optimization for DB file implementation have been successfully materialized and tested. The repository for all the work is present here.

Source File Hierarchy:

The directories are arranged in a flat manner with exception of a tpch-dbgen project directory. The tpch-dbgen is the git repository for the TPCH sub program. Please make sure the *.tbl files are present BEFORE test is run.

Some Points to Note:

→ Note 1: flex version equal or higher than 2.6 is needed to run the above implementation and not get below error during compilation phase:
"lexer_func.l:67:1: error: 'yyfunclineno' undeclared (first use in this function); did you mean 'yyfuncleng'?"
while running the make to generate test.out file. If such an error arises, however, please uncomment line 16 in 'lexer_func.l'.
→ Note 2:
Pipe id for input from database file is is -2 and Pipe id for final output is -1

The compilation process:

The compilation process is accomplished by a recursive call to Makefiles in each subdirectory by the Makefile in the src/ directory.
The provided NamanArora_NikunjSarda_p42.zip has one sub-directory, src/.

```
$> unzip NamanArora_NikunjSarda_p42.zip
$> cd proj42
```

Now to build:

```
$> make
```

The above command will create a42.out
To run the associated Gtests for the current submission:

```
$> make gtest.out
$> ./gtest.out
```

To clean the repository, excluding *.bin and *.tbl files,

$> make clean

To clean the repository, including *.bin and *.tbl files

$> make distclean

Change Log:

➔ Added qp_tree.cc

➔ Added qp_tree_helper.cc

qp_tree.cc:

➔ query :: query(struct FuncOperator *finalFunction, struct TableList *tables, struct AndList *boolean, struct NameList *groupingAtts, struct NameList *attsToSelect, int distinctAtts, int distinctFunc): constructor to structure query, initializing the variable to initial value

➔ query :: ~query(): Destructor to structure query

➔ operation :: operation(): constructor to structure operation, initializing the variable to initial value

➔ operation :: operation(int flag): constructor to structure operation, initializing the variable to initial value and type (struct variable) to flag

➔ operation :: operation(struct AndList *a_list, Qptree *ref): constructor to structure operation, initializing the variable to provided values

➔ operation :: ~operation(): destructor to structure operation

➔ void operation :: print(): method to print process information

➔ bool sel_op_comp :: operator()(operation *l, operation *r): method to compare cost of left and right operation and return boolean value accordingly

➔ bool join_op_comp :: operator()(operation *l, operation *r): method to compare cost of left and right operation and return boolean value accordingly

➔ tableInfo :: tableInfo(): constructor to structure tableInfo

➔ tableInfo :: ~tableInfo(): destructor to structure tableInfo

➔ struct operation *tableInfo :: dispense_select(): method to evaluate cost for select

➔ Qptree :: Qptree(char *stat_fname, char *catalog_file):constructor to class Qptree to initialize class variable

➔ Qptree :: ~Qptree(): destructor to class Qptree

➔ void Qptree :: process(struct query *q): method to construct the expense tree

➔ void print_in_order(struct operation *tree): method to print the expense tree

qp_tree_helper.cc:

➔ void Qptree :: get_attr(char *att_name, pair<string, unordered_map<string, tableInfo> :: iterator> &p): method to get the attribute via name passed to the function and stored it in pair object

➔ void Qptree :: process(struct TableList *tables): method to process info for given tables list

➔ void Qptree :: process(struct operation *op, struct AndList *a_list, struct OrList *o_list, char **rels, int *curr_indx): method to process “or” operation

➔ void Qptree :: process(struct AndList *a_list): method to process “and” operator starting from right end

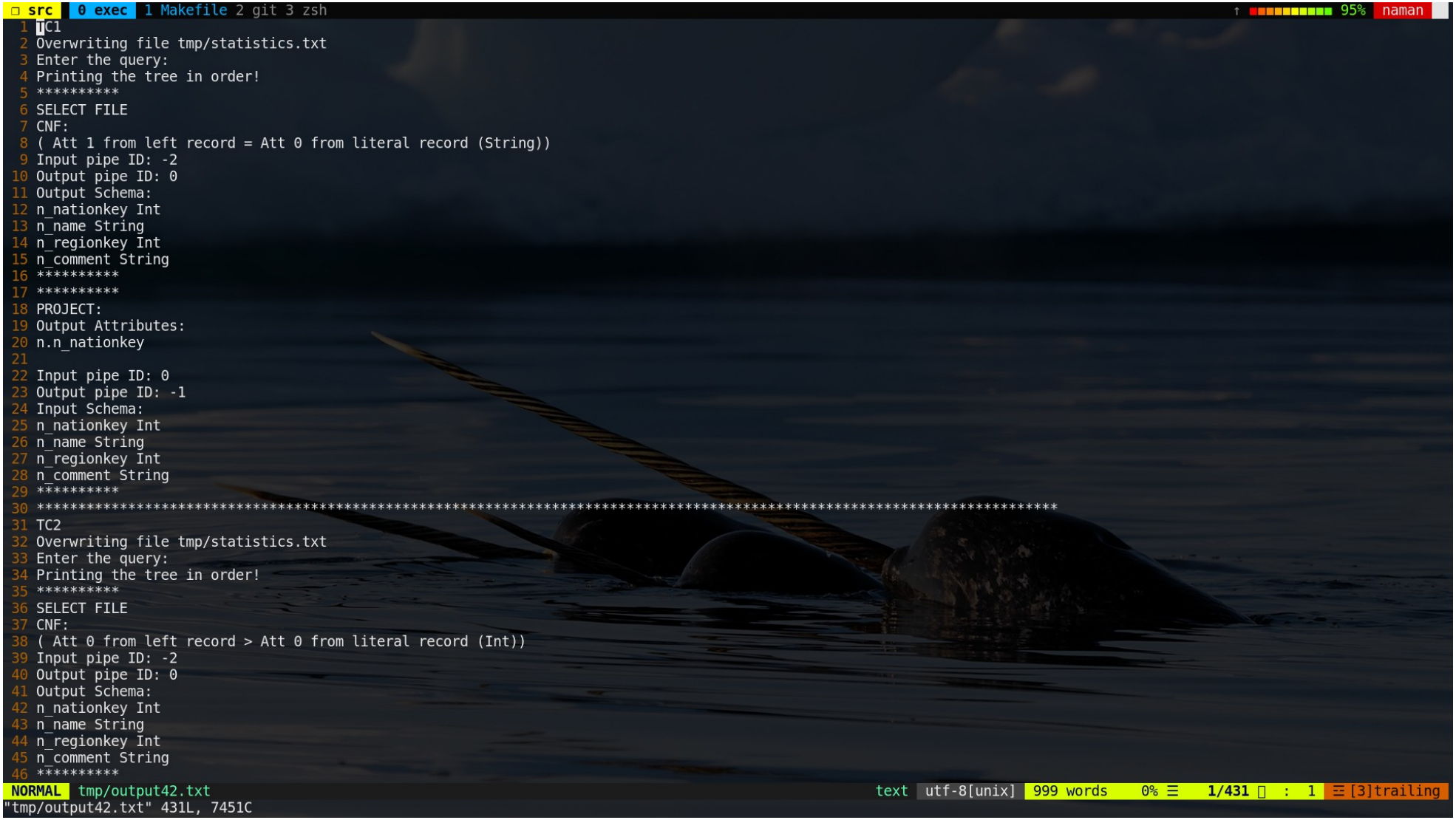
➔ int Qptree :: dispense_pipe(): method to return pipe count

➔ struct operation *Qptree :: dispense_join(struct operation *j_op, int indx, vector<operation *> &j_vec, stack<operation *> &j_stk): method to get least cost join

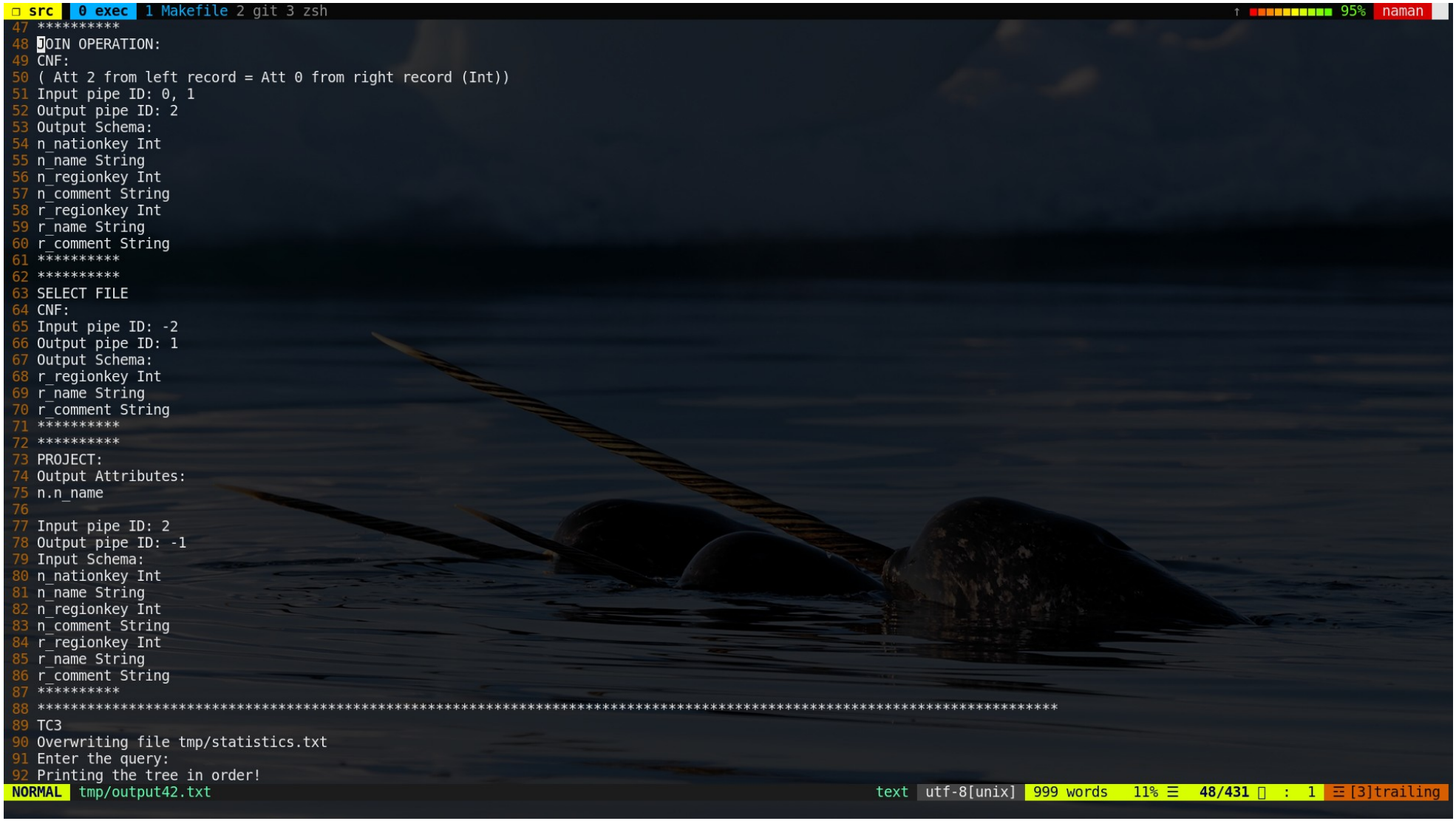
➔ void Qptree :: process_join(struct operation *j_op, vector<operation *> &j_vec, stack<operation *> &j_stk): method to process joins according to there expense

Outputs:

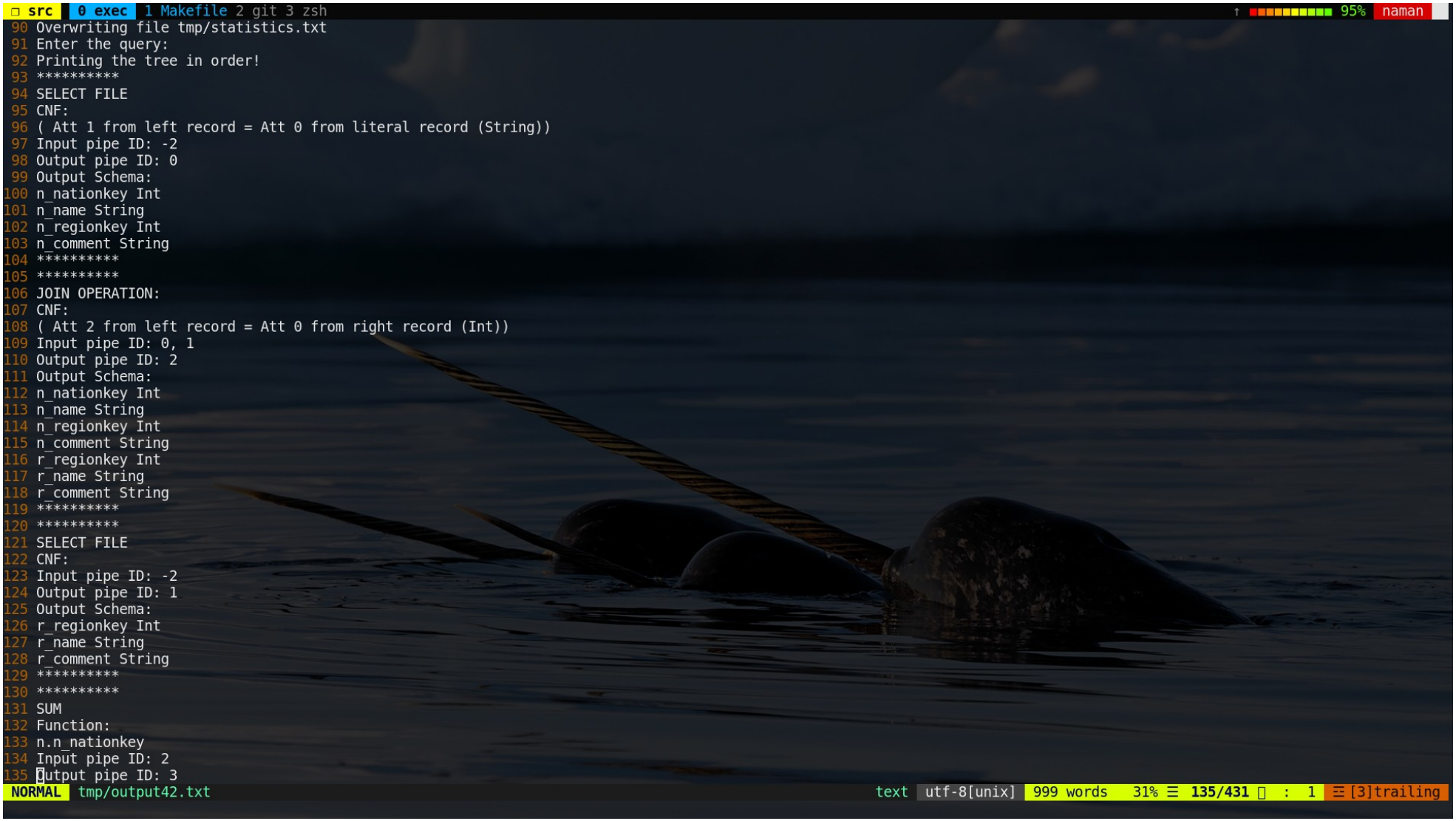
output42.txt



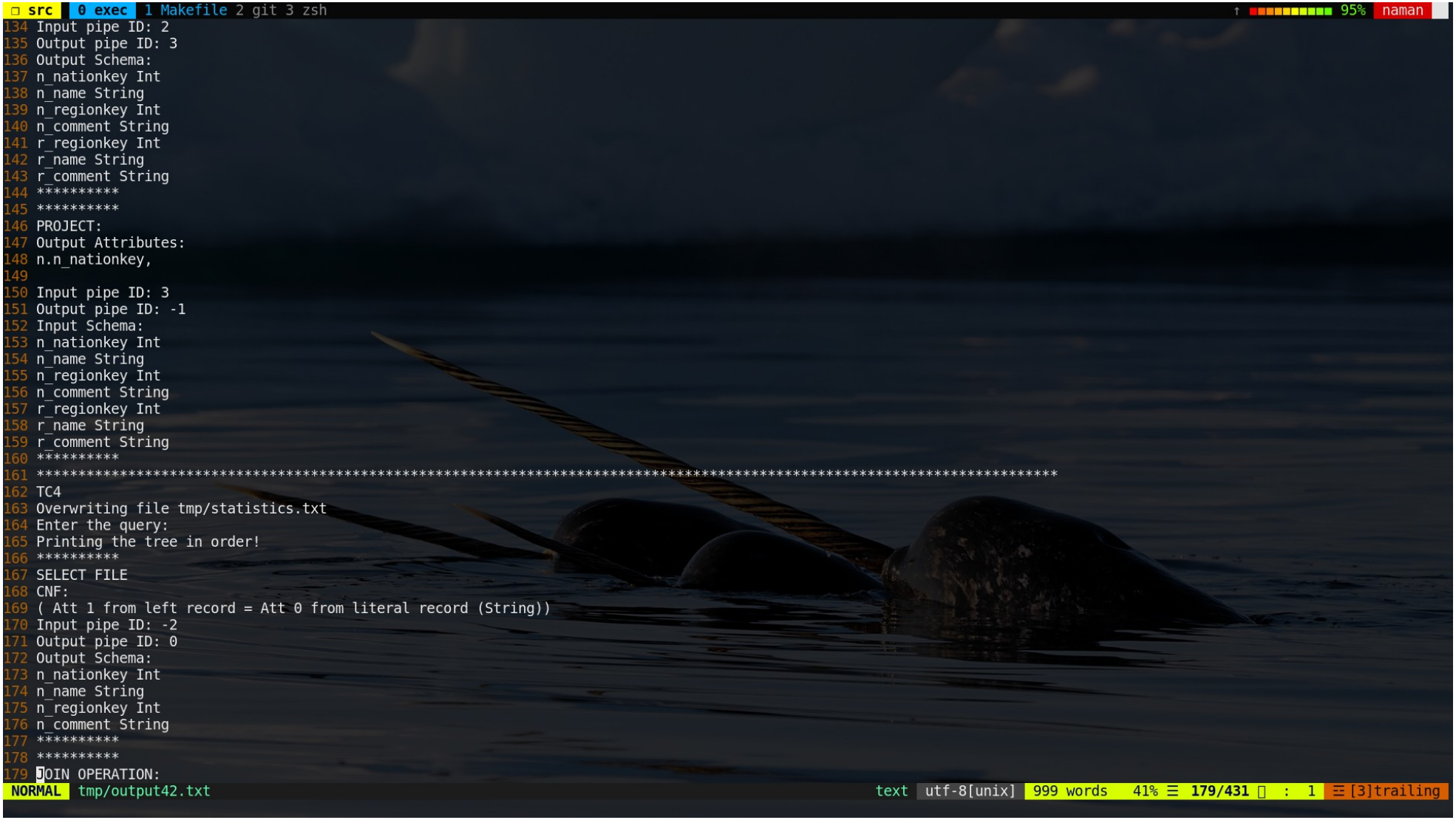
output42.txt



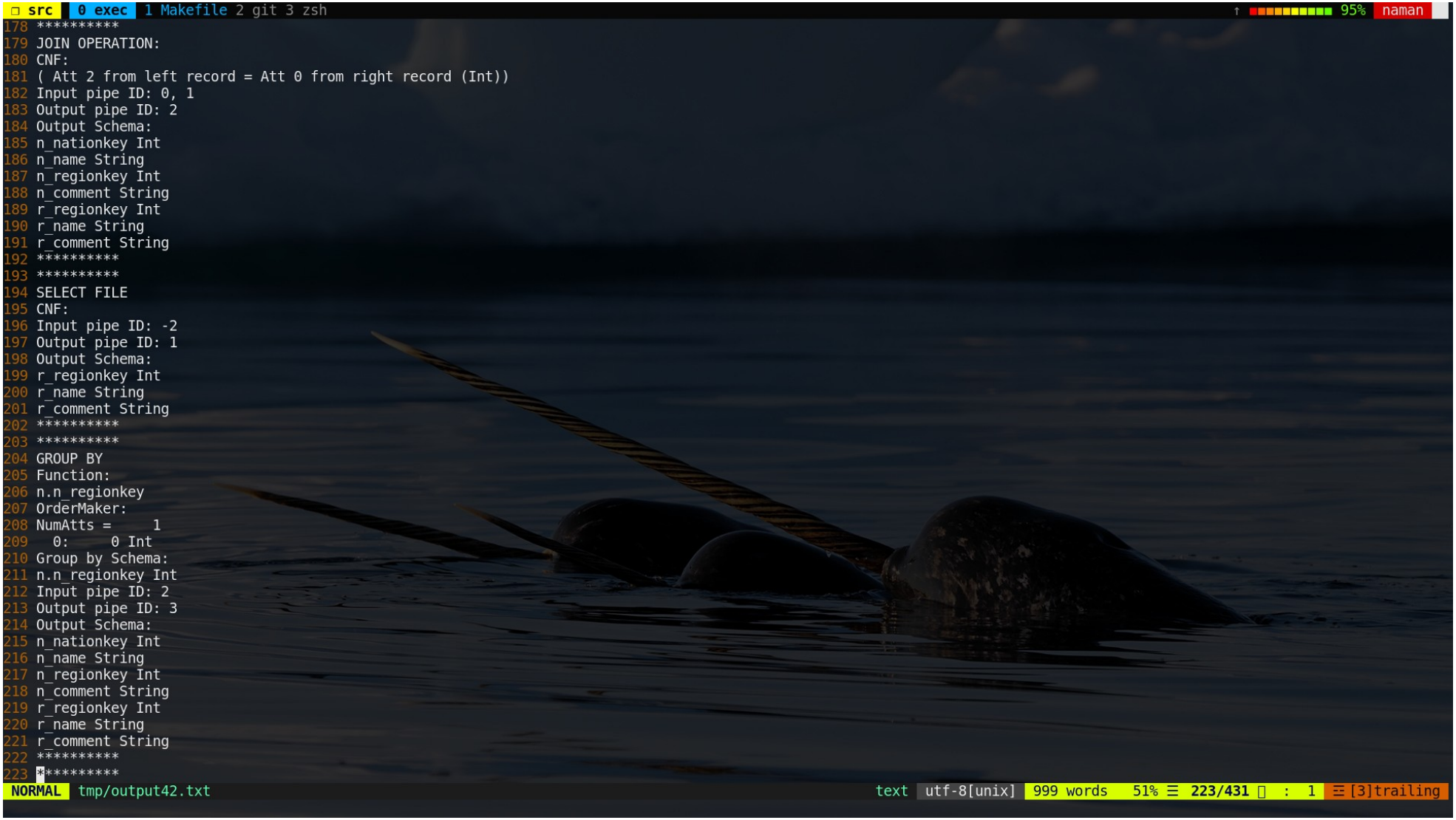
output42.txt



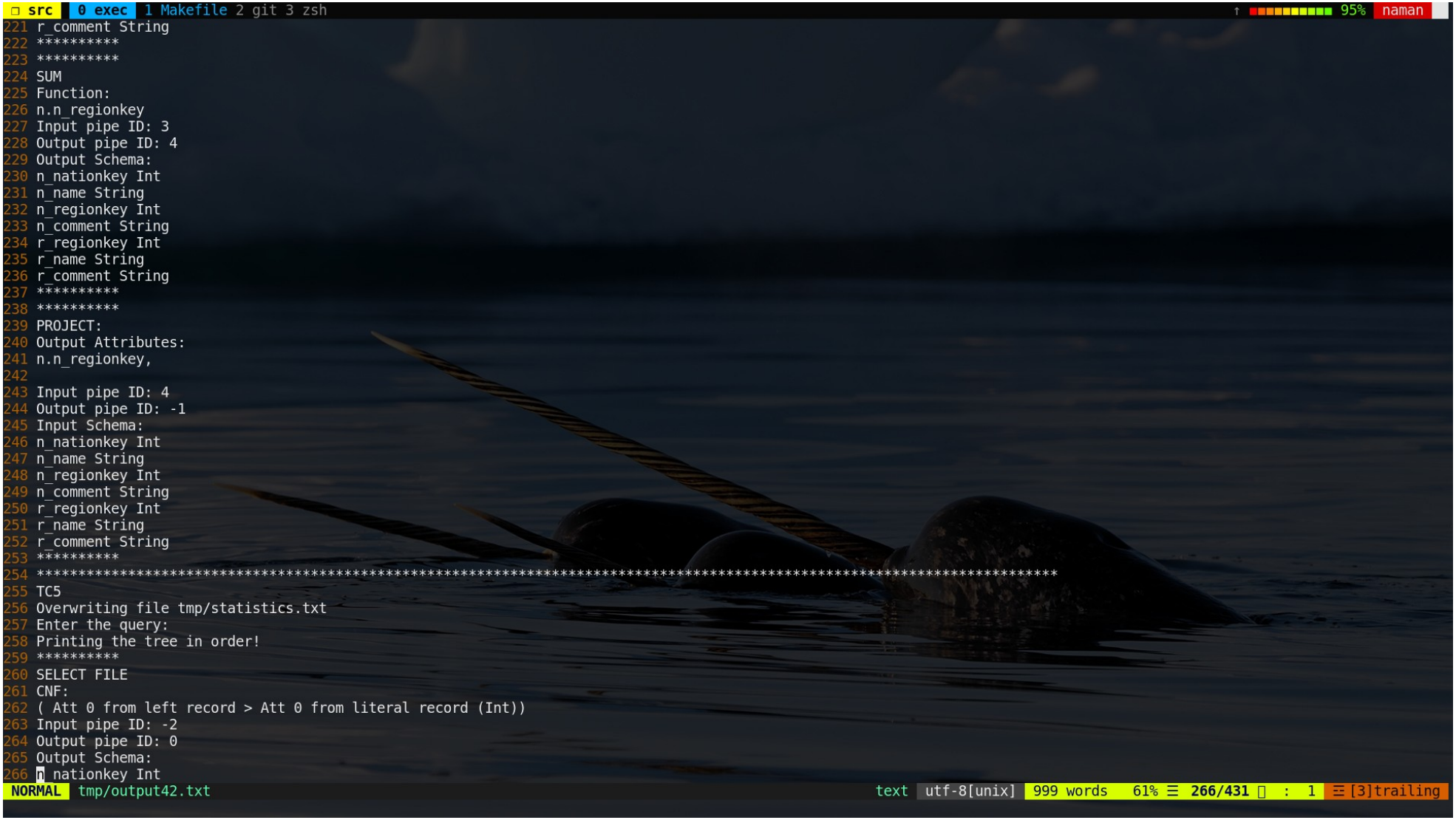
output42.txt



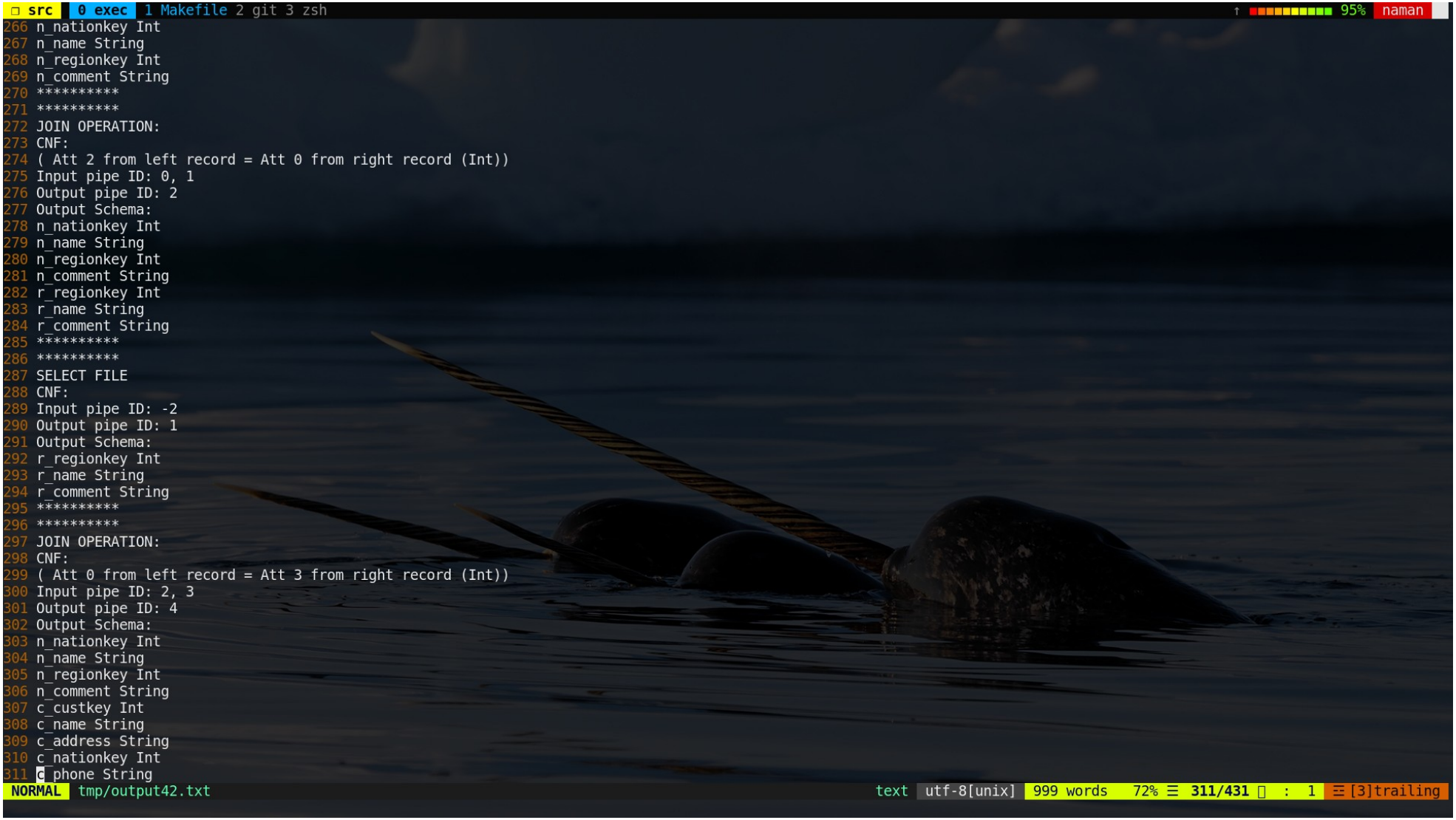
output42.txt



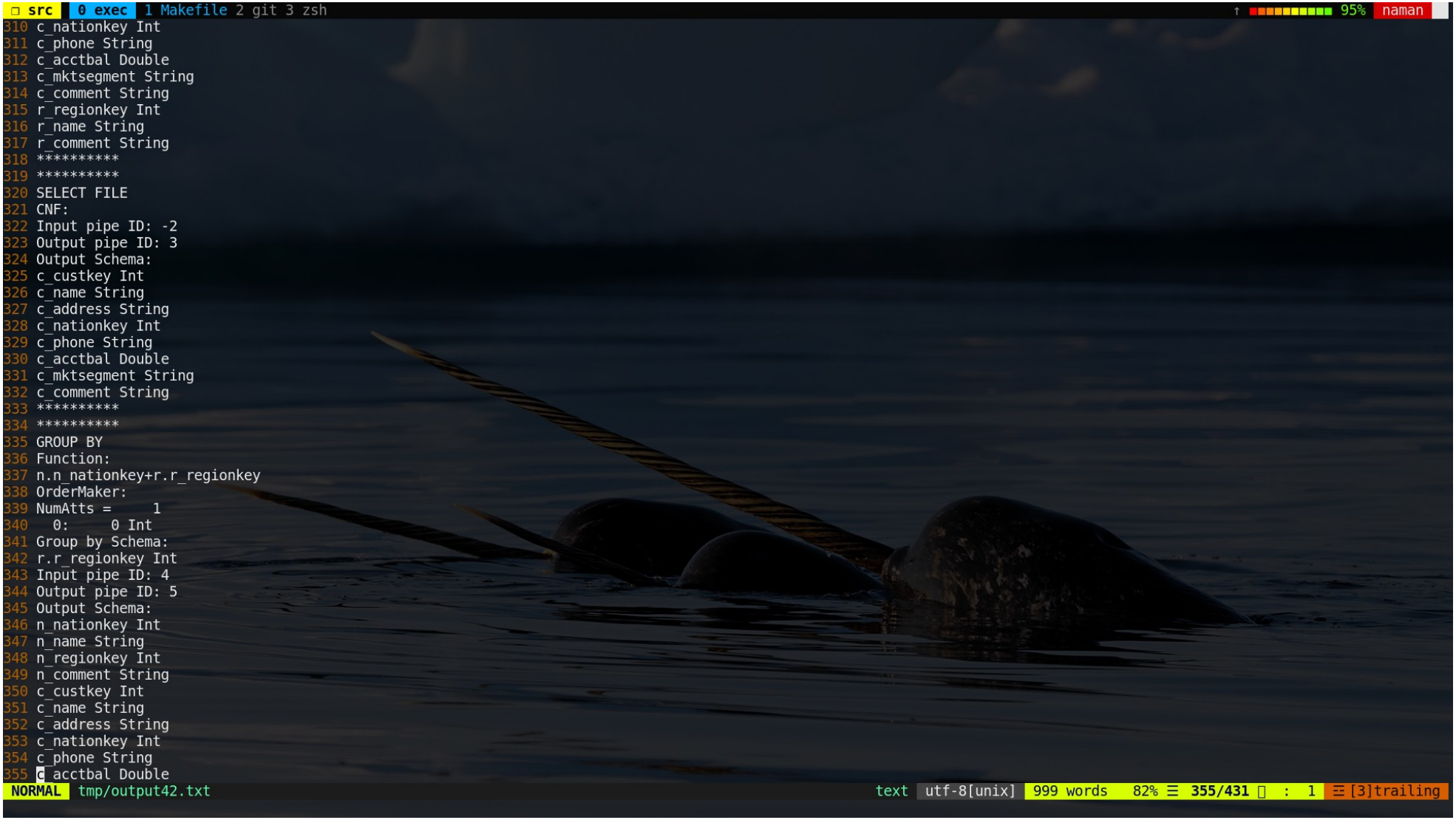
output42.txt



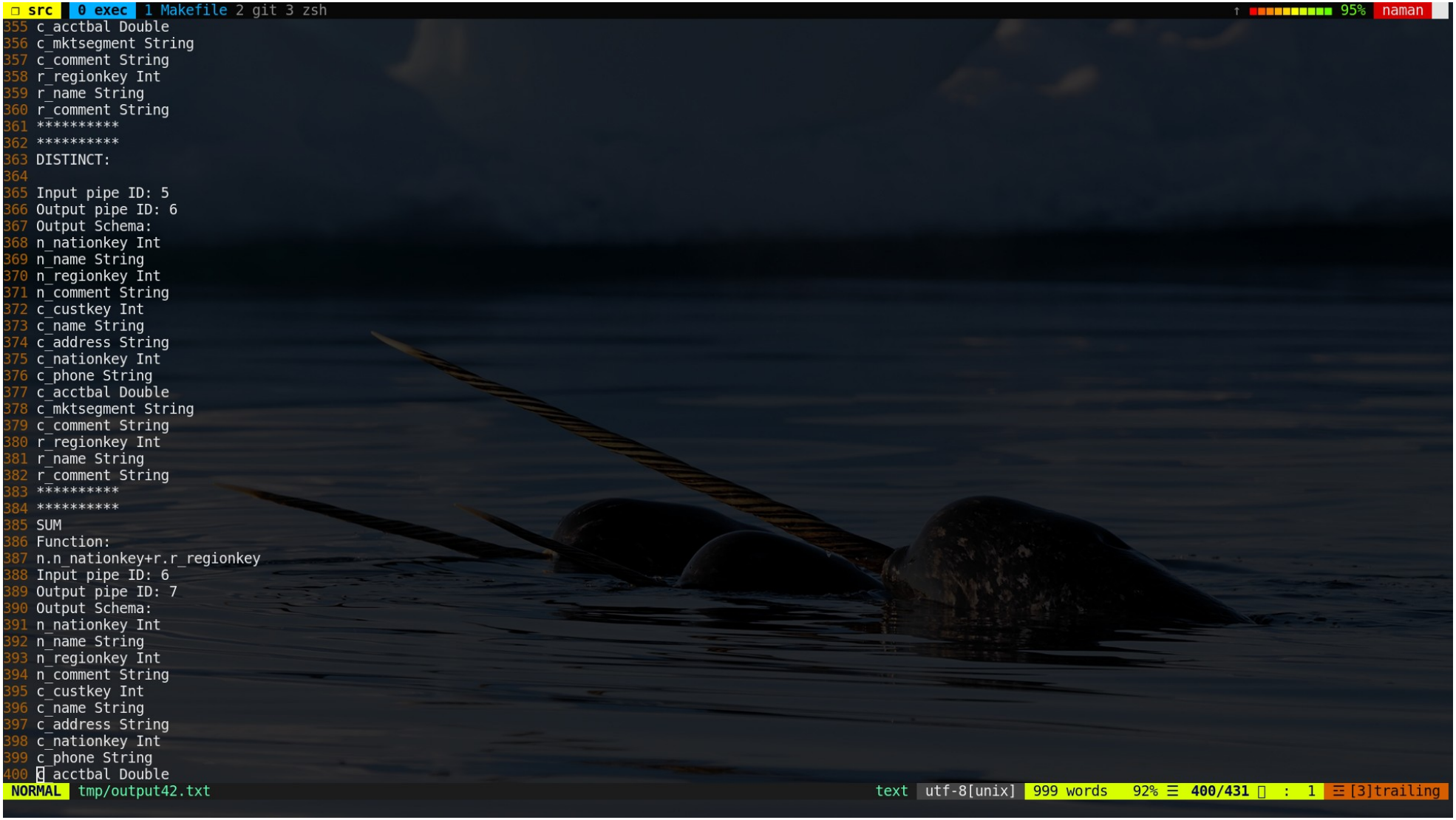
output42.txt



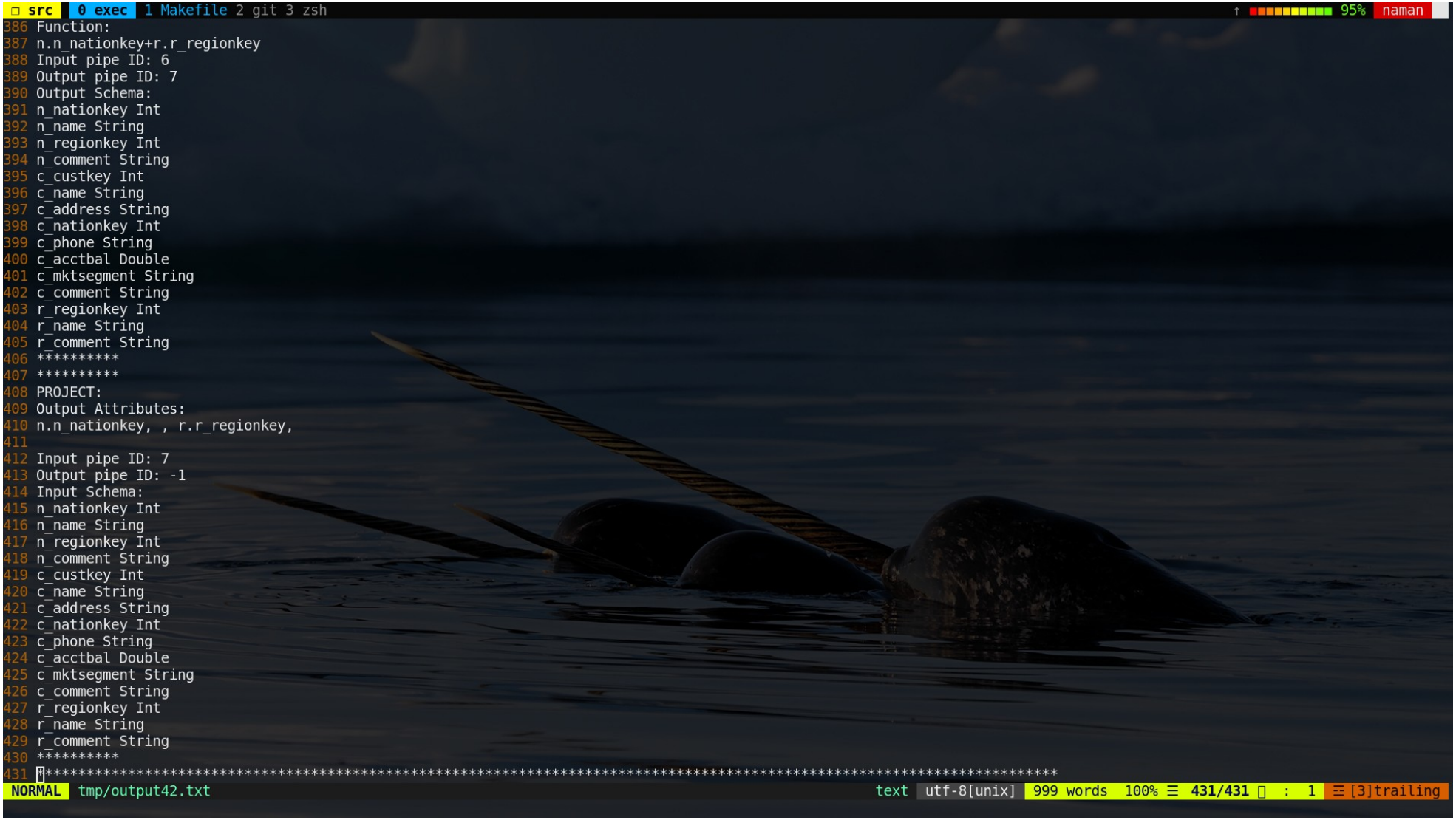
output42.txt



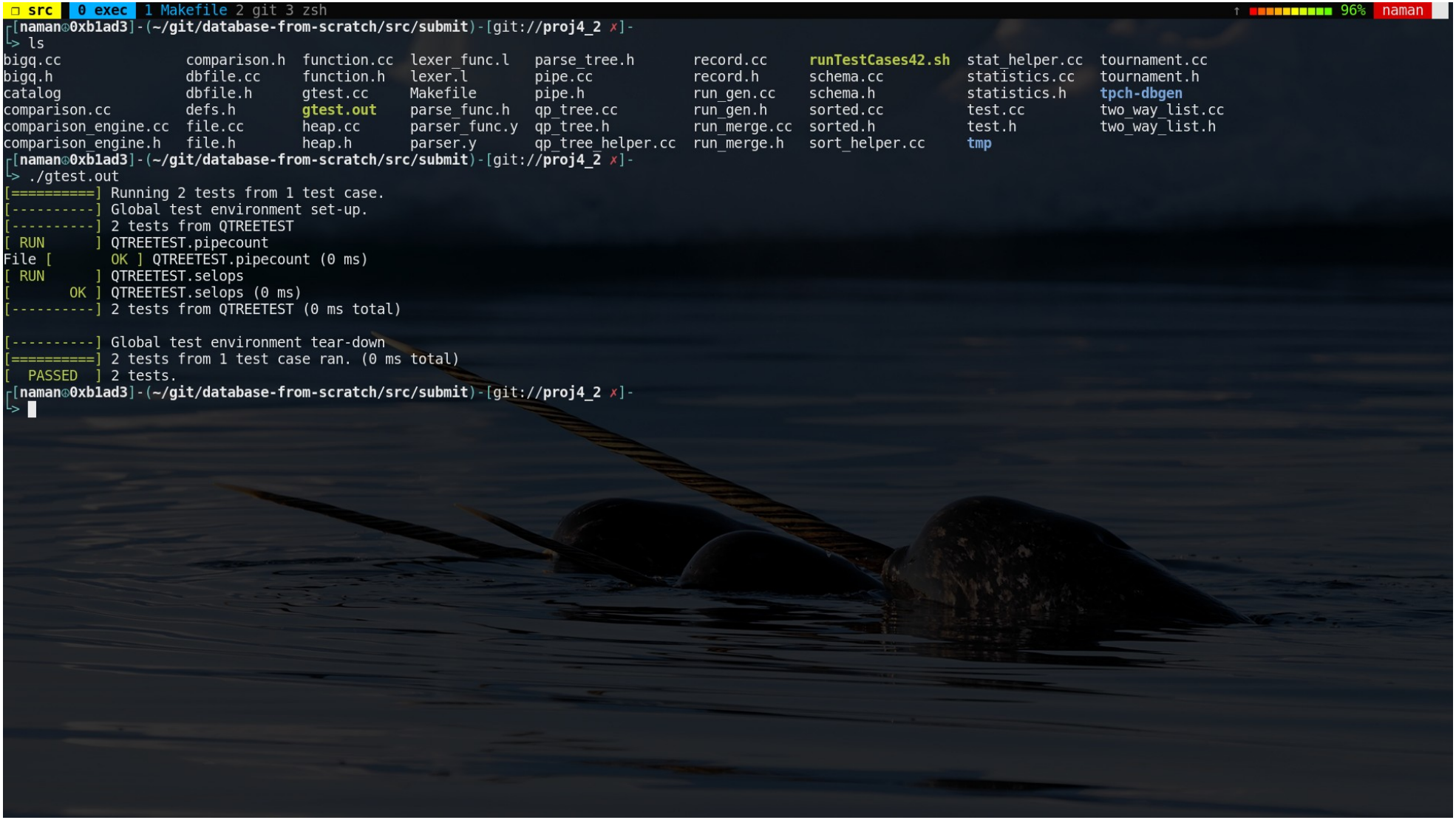
output42.txt



output42.txt



GTest


src 0 exec 1 Makefile 2 git 3 zsh
96% naman
[naman@0xb1ad3]-(~/git/database-from-scratch/src/submit)-[git://proj4_2 ✗]-
> ls
bigq.cc comparison.h function.cc lexer_func.l parse_tree.h record.cc runTestCases42.sh stat_helper.cc tournament.cc
bigq.h dbfile.cc function.h lexer.l pipe.cc record.h schema.cc statistics.cc tournament.h
catalog dbfile.h gtest.cc Makefile pipe.h run_gen.cc schema.h statistics.h tpch-dbgen
comparison.cc defs.h gtest.out parse_func.h qp_tree.cc run_gen.h sorted.cc test.cc two_way_list.cc
comparison_engine.cc file.cc heap.cc parser_func.y qp_tree.h run_merge.cc sorted.h test.h two_way_list.h
comparison_engine.h file.h heap.h parser.y qp_tree_helper.cc run_merge.h sort_helper.cc tmp
[naman@0xb1ad3]-(~/git/database-from-scratch/src/submit)-[git://proj4_2 ✗]-
> ./gtest.out
[==========] Running 2 tests from 1 test case.
[----------] Global test environment set-up.
[----------] 2 tests from QTREETEST
[RUN] QTREETEST.pipecount
File [OK] QTREETEST.pipecount (0 ms)
[RUN] QTREETEST.selops
[OK] QTREETEST.selops (0 ms)
[----------] 2 tests from QTREETEST (0 ms total)
[----------] Global test environment tear-down
[==========] 2 tests from 1 test case ran. (0 ms total)
[PASSED] 2 tests.
[naman@0xb1ad3]-(~/git/database-from-scratch/src/submit)-[git://proj4_2 ✗]-
>