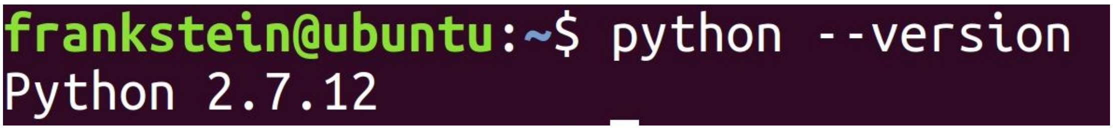
Deepdive 环境搭建及操作

一、虚拟机环境配置

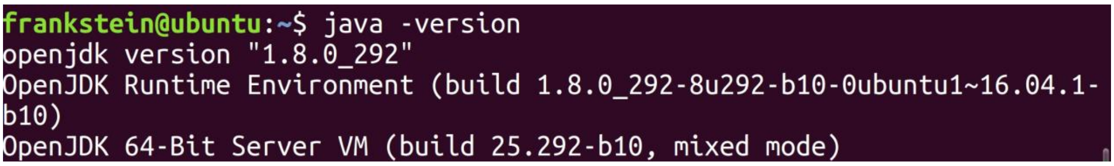
查看 python 版本



```
frankstein@ubuntu:~$ python --version
Python 2.7.12
```

安装 openjdk-8-jdk

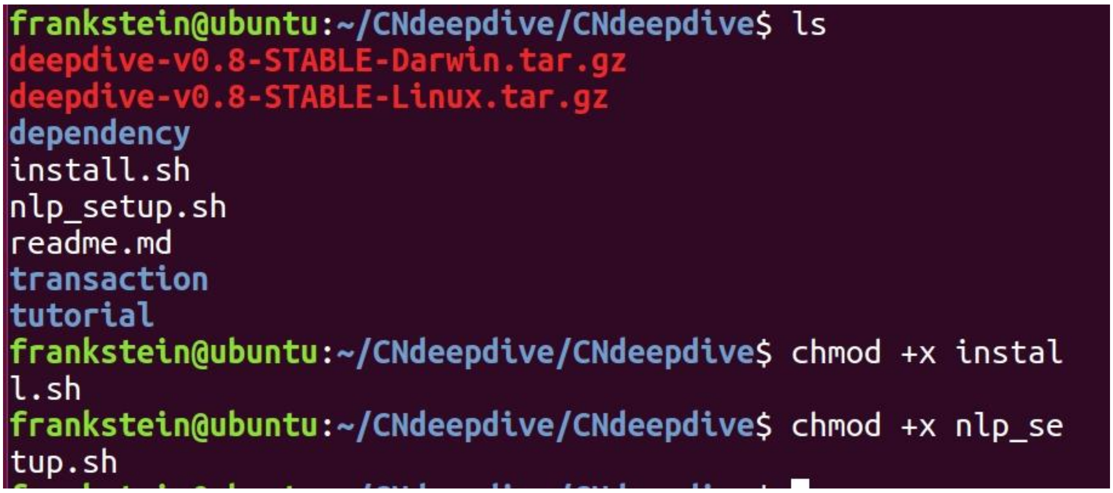
查看 java 版本，验证是否安装成功



```
frankstein@ubuntu:~$ java -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~16.04.1-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
```

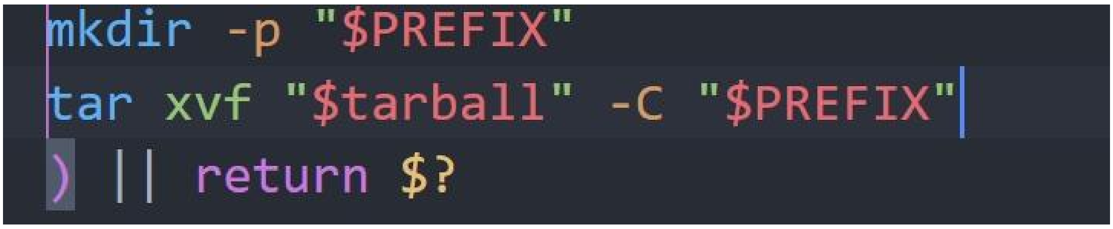
二、安装 CNDeepdive

为 install.sh 和 nlp_setup.sh 文件增加可执行权限



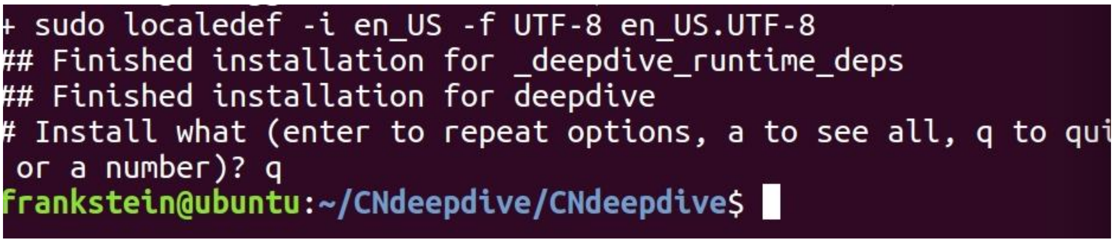
```
frankstein@ubuntu:~/CNdeepdive/CNdeepdive$ ls
deepdive-v0.8-STABLE-Darwin.tar.gz
deepdive-v0.8-STABLE-Linux.tar.gz
dependency
install.sh
nlp_setup.sh
readme.md
transaction
tutorial
frankstein@ubuntu:~/CNdeepdive/CNdeepdive$ chmod +x install.sh
frankstein@ubuntu:~/CNdeepdive/CNdeepdive$ chmod +x nlp_setup.sh
```

打开 install.sh 文件，修改第 193 行内容



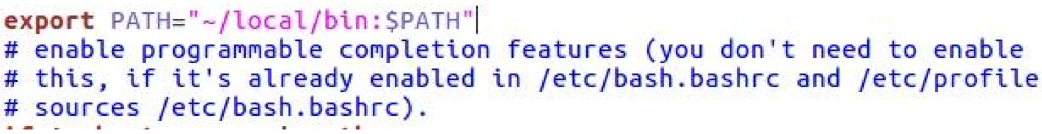
```
    mkdir -p "$PREFIX"
    tar xvf "$tarball" -C "$PREFIX"
    ) || return $?
```

运行 install.sh 文件，输入 1 安装 deepdive



```
+ sudo localedef -i en_US -f UTF-8 en_US.UTF-8
## Finished installation for _deepdive_runtime_deps
## Finished installation for deepdive
# Install what (enter to repeat options, a to see all, q to qui
 or a number)? q
frankstein@ubuntu:~/CNdeepdive/CNdeepdive$ 
```

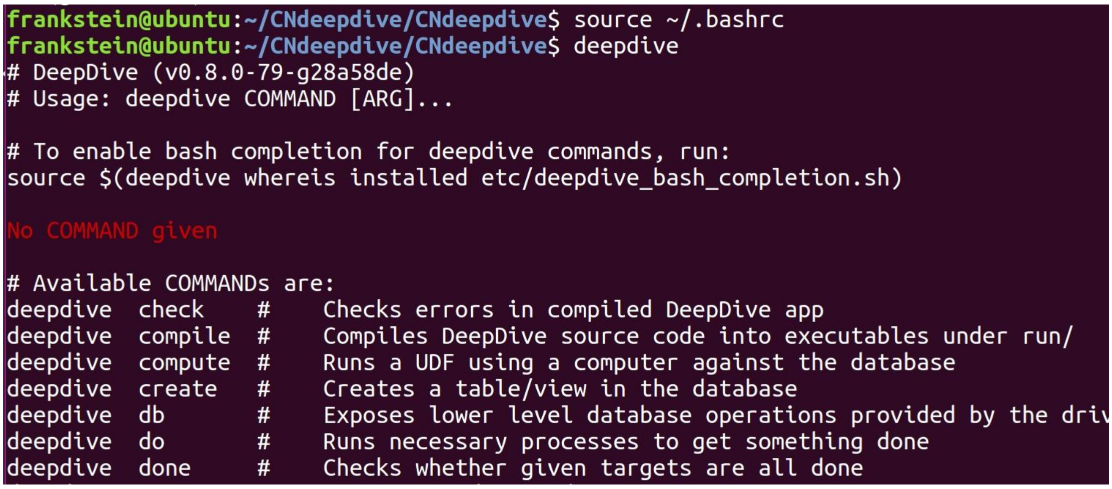
完成后需要配置 CNdeepdive 的环境变量 添加下面的语句



```
export PATH="~/local/bin:$PATH"
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
```

运行下面的语句让改动生效

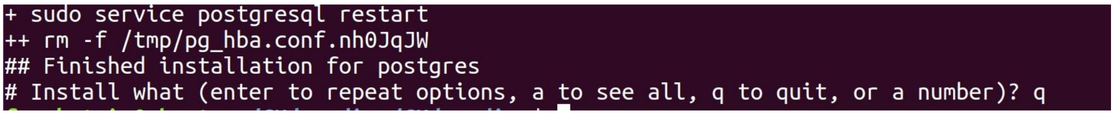
在控制台输入 deepdive，出现下面的输出证明配置成功



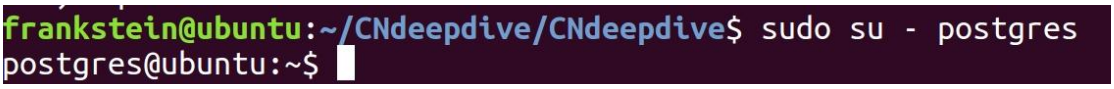
```
frankstein@ubuntu:~/CNdeepdive/CNdeepdive$ source ~/.bashrc
frankstein@ubuntu:~/CNdeepdive/CNdeepdive$ deepdive
# DeepDive (v0.8.0-79-g28a58de)
# Usage: deepdive COMMAND [ARG]...

# To enable bash completion for deepdive commands, run:
source $(deepdive whereis installed etc/deepdive_bash_completion.sh)

No COMMAND given

# Available COMMANDs are:
deepdive  check    #    Checks errors in compiled DeepDive app
deepdive  compile  #    Compiles DeepDive source code into executables under run/
deepdive  compute  #    Runs a UDF using a computer against the database
deepdive  create   #    Creates a table/view in the database
deepdive  db       #    Exposes lower level database operations provided by the driv
deepdive  do       #    Runs necessary processes to get something done
deepdive  done     #    Checks whether given targets are all done
```

三、安装 postgresql

运行 install.sh 文件，输入 6 安装 postgresql



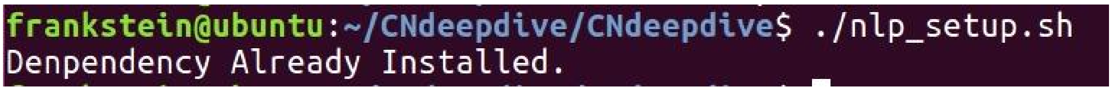
```
+ sudo service postgresql restart
++ rm -f /tmp/pg_hba.conf.nh0JqJW
## Finished installation for postgres
# Install what (enter to repeat options, a to see all, q to quit, or a number)? q
```



```
frankstein@ubuntu:~/CNdeepdive/CNdeepdive$ sudo su - postgres
postgres@ubuntu:~$ 
```

四、安装 nlp

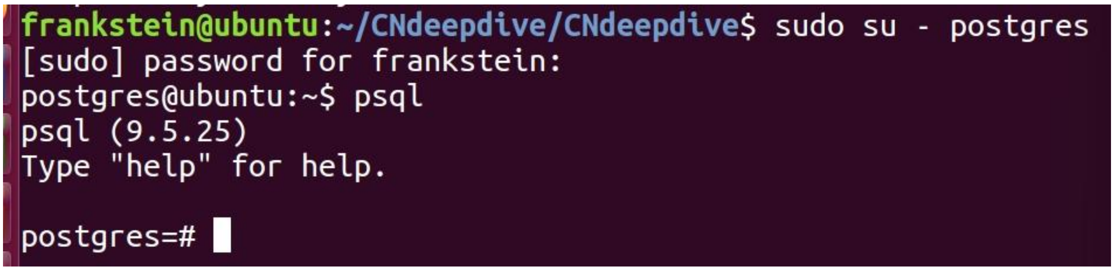
运行 nlp_setup.sh 文件



```
frankstein@ubuntu:~/CNdeepdive/CNdeepdive$ ./nlp_setup.sh
Denpendency Already Installed.
```

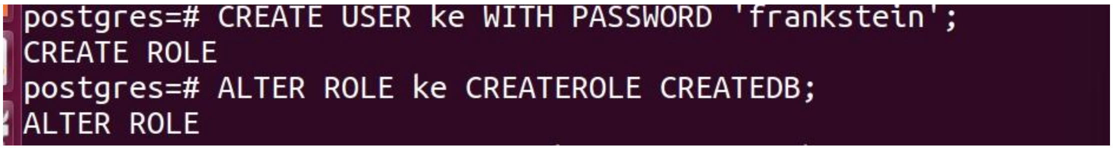
五、Smoke Example

进入 postgresql 控制台



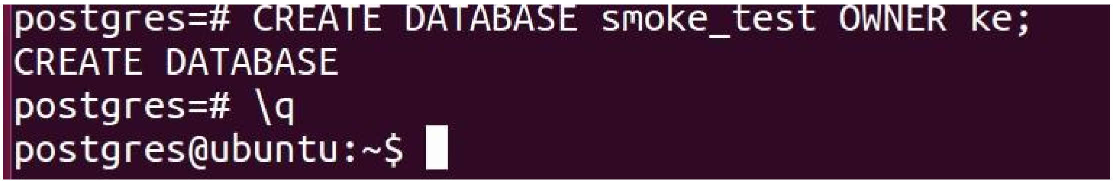
```
frankstein@ubuntu:~/CNdeepdive/CNdeepdive$ sudo su - postgres
[sudo] password for frankstein:
postgres@ubuntu:~$ psql
psql (9.5.25)
Type "help" for help.

postgres=# 
```

创建用户 设置用户权限



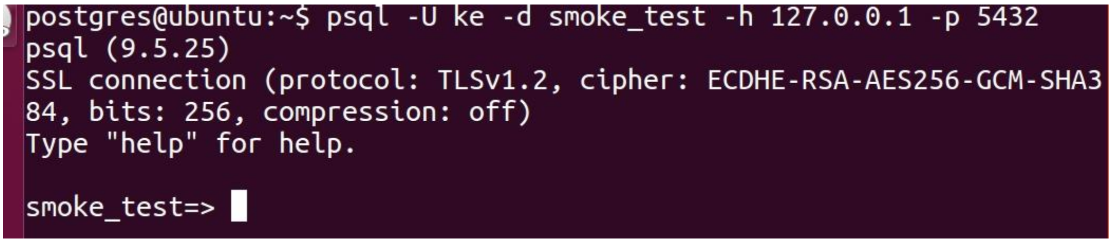
```
postgres=# CREATE USER ke WITH PASSWORD 'frankstein';
CREATE ROLE
postgres=# ALTER ROLE ke CREATEROLE CREATEDB;
ALTER ROLE
```

创建 smoke_test 数据库，并指定所有者
退出 postgresql 控制台



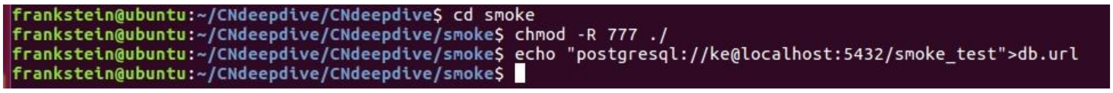
```
postgres=# CREATE DATABASE smoke_test OWNER ke;
CREATE DATABASE
postgres=# \q
postgres@ubuntu:~$ 
```

查看是否创建成功



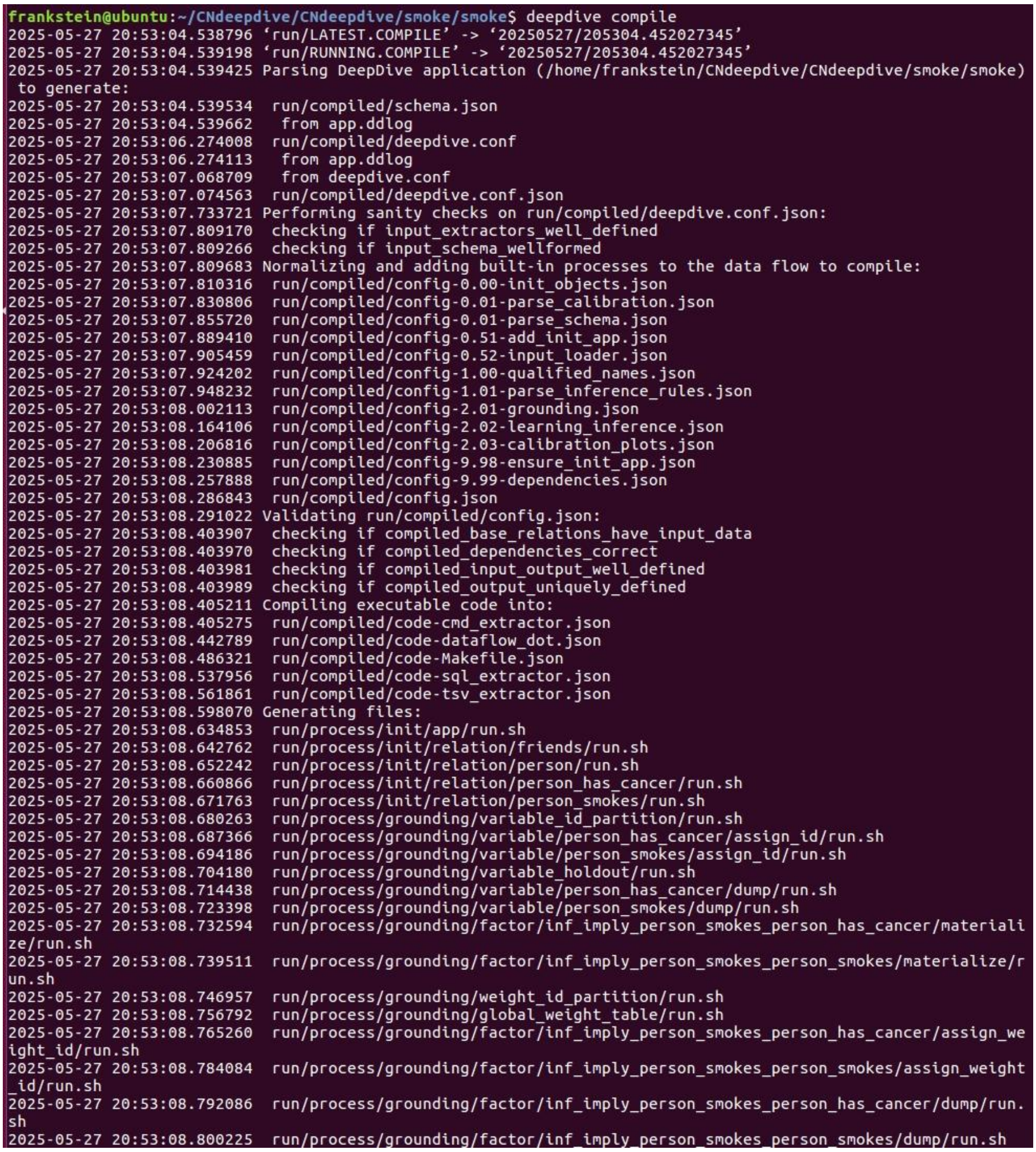
```
postgres@ubuntu:~$ psql -U ke -d smoke_test -h 127.0.0.1 -p 5432
psql (9.5.25)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA3
84, bits: 256, compression: off)
Type "help" for help.

smoke_test=> 
```

进入 smoke 文件夹后，修改用户权限
建立数据库配置文件

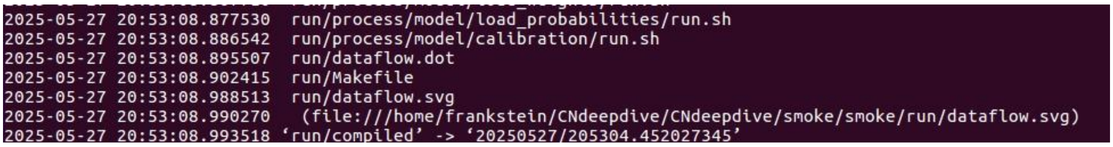


```
frankstein@ubuntu:~/CNdeepdive/CNdeepdive$ cd smoke
frankstein@ubuntu:~/CNdeepdive/CNdeepdive/smoke$ chmod -R 777 ./
frankstein@ubuntu:~/CNdeepdive/CNdeepdive/smoke$ echo "postgresql://ke@localhost:5432/smoke_test">db.url
frankstein@ubuntu:~/CNdeepdive/CNdeepdive/smoke$ 
```

编译

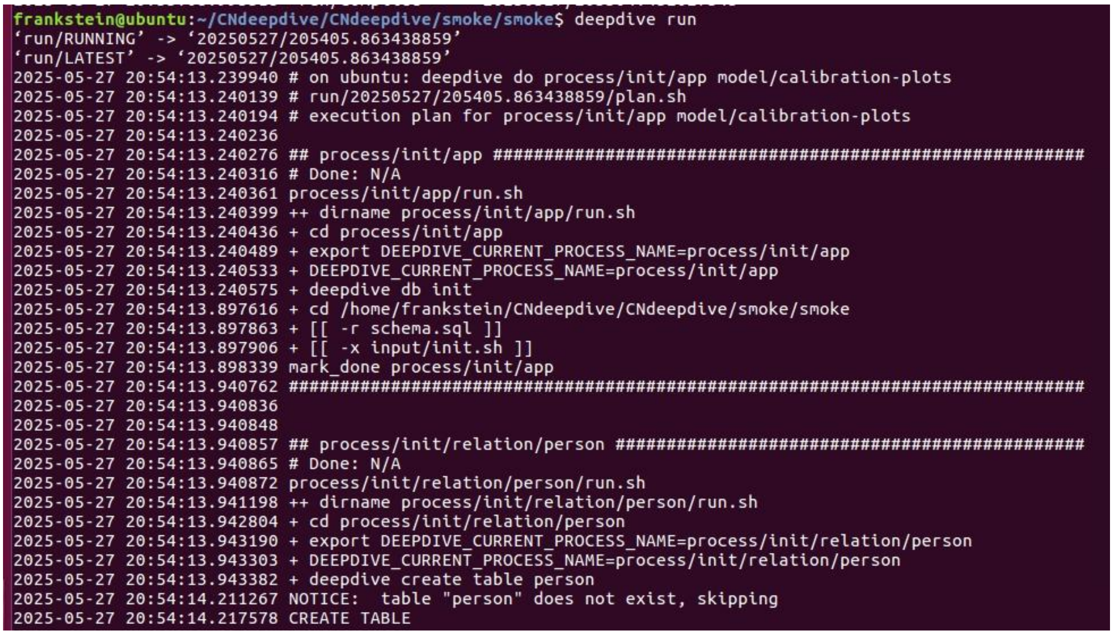


```
frankstein@ubuntu:~/CNdeepdive/CNdeepdive/smoke/smoke$ deepdive compile
2025-05-27 20:53:04.538796 ‘run/LATEST.COMPILE’ -> ‘20250527/205304.452027345’
2025-05-27 20:53:04.539198 ‘run/RUNNING.COMPILE’ -> ‘20250527/205304.452027345’
2025-05-27 20:53:04.539425 Parsing DeepDive application (/home/frankstein/CNdeepdive/CNdeepdive/smoke/smoke)
 to generate:
2025-05-27 20:53:04.539534  run/compiled/schema.json
2025-05-27 20:53:04.539662   from app.ddlog
2025-05-27 20:53:06.274008  run/compiled/deepdive.conf
2025-05-27 20:53:06.274113   from app.ddlog
2025-05-27 20:53:07.068709   from deepdive.conf
2025-05-27 20:53:07.074563  run/compiled/deepdive.conf.json
2025-05-27 20:53:07.733721 Performing sanity checks on run/compiled/deepdive.conf.json:
2025-05-27 20:53:07.809170  checking if input_extractors_well_defined
2025-05-27 20:53:07.809266  checking if input_schema_wellformed
2025-05-27 20:53:07.809683 Normalizing and adding built-in processes to the data flow to compile:
2025-05-27 20:53:07.810316  run/compiled/config-0.00-init_objects.json
2025-05-27 20:53:07.830806  run/compiled/config-0.01-parse_calibration.json
2025-05-27 20:53:07.855720  run/compiled/config-0.01-parse_schema.json
2025-05-27 20:53:07.889410  run/compiled/config-0.51-add_init_app.json
2025-05-27 20:53:07.905459  run/compiled/config-0.52-input_loader.json
2025-05-27 20:53:07.924202  run/compiled/config-1.00-qualified_names.json
2025-05-27 20:53:07.948232  run/compiled/config-1.01-parse_inference_rules.json
2025-05-27 20:53:08.002113  run/compiled/config-2.01-grounding.json
2025-05-27 20:53:08.164106  run/compiled/config-2.02-learning_inference.json
2025-05-27 20:53:08.206816  run/compiled/config-2.03-calibration_plots.json
2025-05-27 20:53:08.230885  run/compiled/config-9.98-ensure_init_app.json
2025-05-27 20:53:08.257888  run/compiled/config-9.99-dependencies.json
2025-05-27 20:53:08.286843  run/compiled/config.json
2025-05-27 20:53:08.291022 Validating run/compiled/config.json:
2025-05-27 20:53:08.403907  checking if compiled_base_relations_have_input_data
2025-05-27 20:53:08.403970  checking if compiled_dependencies_correct
2025-05-27 20:53:08.403981  checking if compiled_input_output_well_defined
2025-05-27 20:53:08.403989  checking if compiled_output_uniquely_defined
2025-05-27 20:53:08.405211 Compiling executable code into:
2025-05-27 20:53:08.405275  run/compiled/code-cmd_extractor.json
2025-05-27 20:53:08.442789  run/compiled/code-dataflow_dot.json
2025-05-27 20:53:08.486321  run/compiled/code-Makefile.json
2025-05-27 20:53:08.537956  run/compiled/code-sql_extractor.json
2025-05-27 20:53:08.561861  run/compiled/code-tsv_extractor.json
2025-05-27 20:53:08.598070 Generating files:
2025-05-27 20:53:08.634853  run/process/init/app/run.sh
2025-05-27 20:53:08.642762  run/process/init/relation/friends/run.sh
2025-05-27 20:53:08.652242  run/process/init/relation/person/run.sh
2025-05-27 20:53:08.660866  run/process/init/relation/person_has_cancer/run.sh
2025-05-27 20:53:08.671763  run/process/init/relation/person_smokes/run.sh
2025-05-27 20:53:08.680263  run/process/grounding/variable_id_partition/run.sh
2025-05-27 20:53:08.687366  run/process/grounding/variable/person_has_cancer/assign_id/run.sh
2025-05-27 20:53:08.694186  run/process/grounding/variable/person_smokes/assign_id/run.sh
2025-05-27 20:53:08.704180  run/process/grounding/variable_holdout/run.sh
2025-05-27 20:53:08.714438  run/process/grounding/variable/person_has_cancer/dump/run.sh
2025-05-27 20:53:08.723398  run/process/grounding/variable/person_smokes/dump/run.sh
2025-05-27 20:53:08.732594  run/process/grounding/factor/inf_imply_person_smokes_person_has_cancer/materiali
ze/run.sh
2025-05-27 20:53:08.739511  run/process/grounding/factor/inf_imply_person_smokes_person_smokes/materialize/r
un.sh
2025-05-27 20:53:08.746957  run/process/grounding/weight_id_partition/run.sh
2025-05-27 20:53:08.756792  run/process/grounding/global_weight_table/run.sh
2025-05-27 20:53:08.765260  run/process/grounding/factor/inf_imply_person_smokes_person_has_cancer/assign_we
ight_id/run.sh
2025-05-27 20:53:08.784084  run/process/grounding/factor/inf_imply_person_smokes_person_smokes/assign_weight
_id/run.sh
2025-05-27 20:53:08.792086  run/process/grounding/factor/inf_imply_person_smokes_person_has_cancer/dump/run.
sh
2025-05-27 20:53:08.800225  run/process/grounding/factor/inf_imply_person_smokes_person_smokes/dump/run.sh
```

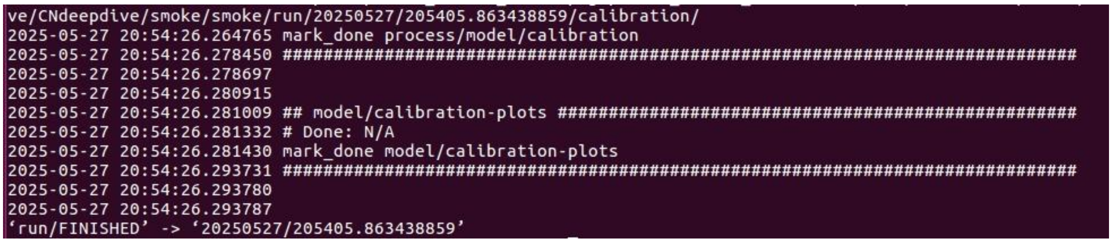


```
2025-05-27 20:53:08.877530  run/process/model/load_probabilities/run.sh
2025-05-27 20:53:08.886542  run/process/model/calibration/run.sh
2025-05-27 20:53:08.895507  run/dataflow.dot
2025-05-27 20:53:08.902415  run/Makefile
2025-05-27 20:53:08.988513  run/dataflow.svg
2025-05-27 20:53:08.990270   (file:///home/frankstein/CNdeepdive/CNdeepdive/smoke/smoke/run/dataflow.svg)
2025-05-27 20:53:08.993518 ‘run/compiled’ -> ‘20250527/205304.452027345’
```

运行

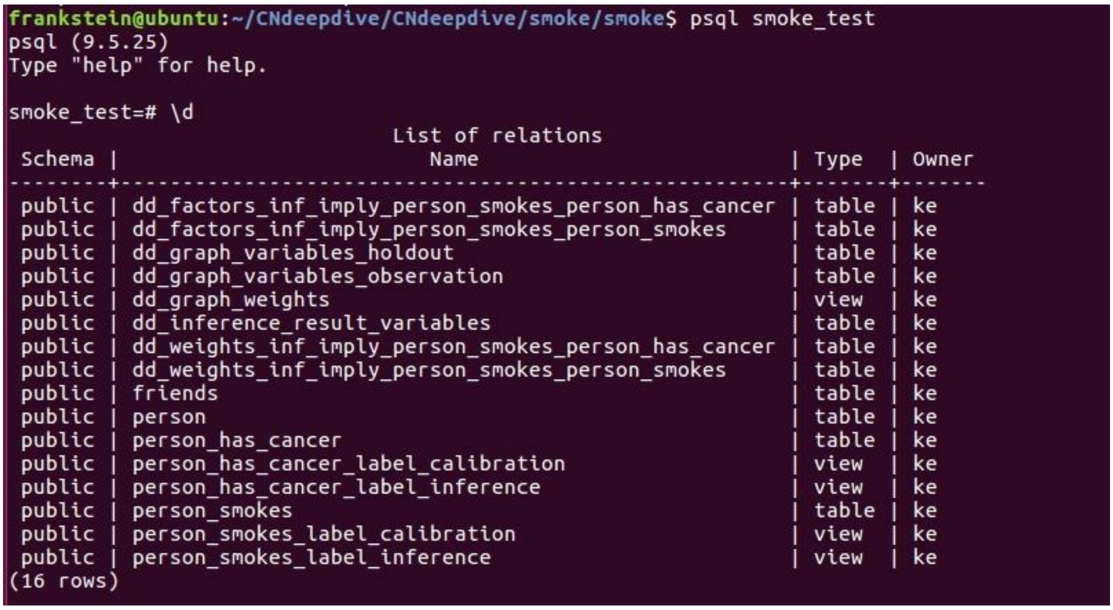


```
frankstein@ubuntu:~/CNdeepdive/CNdeepdive/smoke/smoke$ deepdive run
'run/RUNNING' -> '20250527/205405.863438859'
'run/LATEST' -> '20250527/205405.863438859'
2025-05-27 20:54:13.239940 # on ubuntu: deepdive do process/init/app model/calibration-plots
2025-05-27 20:54:13.240139 # run/20250527/205405.863438859/plan.sh
2025-05-27 20:54:13.240194 # execution plan for process/init/app model/calibration-plots
2025-05-27 20:54:13.240236
2025-05-27 20:54:13.240276 ## process/init/app ##########################################################
2025-05-27 20:54:13.240316 # Done: N/A
2025-05-27 20:54:13.240361 process/init/app/run.sh
2025-05-27 20:54:13.240399 ++ dirname process/init/app/run.sh
2025-05-27 20:54:13.240436 + cd process/init/app
2025-05-27 20:54:13.240489 + export DEEPDIVE_CURRENT_PROCESS_NAME=process/init/app
2025-05-27 20:54:13.240533 + DEEPDIVE_CURRENT_PROCESS_NAME=process/init/app
2025-05-27 20:54:13.240575 + deepdive db init
2025-05-27 20:54:13.897616 + cd /home/frankstein/CNdeepdive/CNdeepdive/smoke/smoke
2025-05-27 20:54:13.897863 + [[ -r schema.sql ]]
2025-05-27 20:54:13.897906 + [[ -x input/init.sh ]]
2025-05-27 20:54:13.898339 mark_done process/init/app
2025-05-27 20:54:13.940762 ##############################################################################
2025-05-27 20:54:13.940836
2025-05-27 20:54:13.940848
2025-05-27 20:54:13.940857 ## process/init/relation/person ##############################################
2025-05-27 20:54:13.940865 # Done: N/A
2025-05-27 20:54:13.940872 process/init/relation/person/run.sh
2025-05-27 20:54:13.941198 ++ dirname process/init/relation/person/run.sh
2025-05-27 20:54:13.942804 + cd process/init/relation/person
2025-05-27 20:54:13.943190 + export DEEPDIVE_CURRENT_PROCESS_NAME=process/init/relation/person
2025-05-27 20:54:13.943303 + DEEPDIVE_CURRENT_PROCESS_NAME=process/init/relation/person
2025-05-27 20:54:13.943382 + deepdive create table person
2025-05-27 20:54:14.211267 NOTICE:  table "person" does not exist, skipping
2025-05-27 20:54:14.217578 CREATE TABLE
```



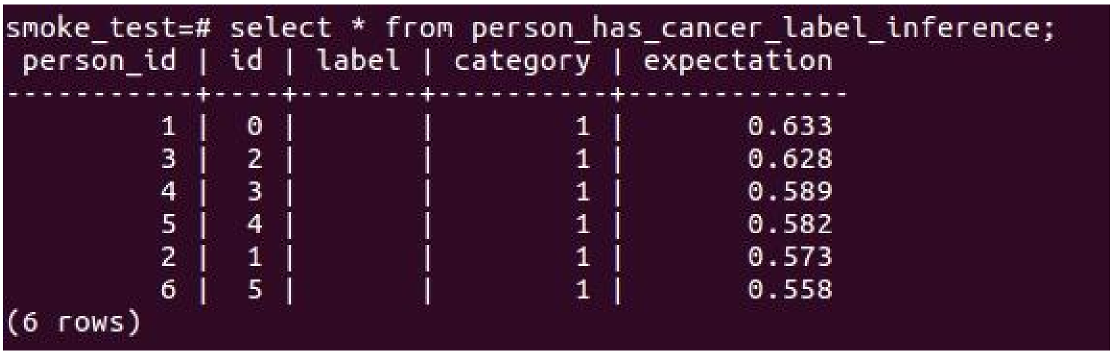
```
ve/CNdeepdive/smoke/smoke/run/20250527/205405.863438859/calibration/
2025-05-27 20:54:26.264765 mark_done process/model/calibration
2025-05-27 20:54:26.278450 ##############################################################################
2025-05-27 20:54:26.278697
2025-05-27 20:54:26.280915
2025-05-27 20:54:26.281009 ## model/calibration-plots ###################################################
2025-05-27 20:54:26.281332 # Done: N/A
2025-05-27 20:54:26.281430 mark_done model/calibration-plots
2025-05-27 20:54:26.293731 ##############################################################################
2025-05-27 20:54:26.293780
2025-05-27 20:54:26.293787
'run/FINISHED' -> '20250527/205405.863438859'
```

运行完成后查看结果



```
frankstein@ubuntu:~/CNdeepdive/CNdeepdive/smoke/smoke$ psql smoke_test
psql (9.5.25)
Type "help" for help.

smoke_test=# \d
                              List of relations
 Schema |                        Name                        | Type  | Owner
--------+----------------------------------------------------+-------+-------
 public | dd_factors_inf_imply_person_smokes_person_has_cancer | table | ke
 public | dd_factors_inf_imply_person_smokes_person_smokes     | table | ke
 public | dd_graph_variables_holdout                           | table | ke
 public | dd_graph_variables_observation                       | table | ke
 public | dd_graph_weights                                     | view  | ke
 public | dd_inference_result_variables                        | table | ke
 public | dd_weights_inf_imply_person_smokes_person_has_cancer | table | ke
 public | dd_weights_inf_imply_person_smokes_person_smokes     | table | ke
 public | friends                                              | table | ke
 public | person                                               | table | ke
 public | person_has_cancer                                    | table | ke
 public | person_has_cancer_label_calibration                  | view  | ke
 public | person_has_cancer_label_inference                    | view  | ke
 public | person_smokes                                        | table | ke
 public | person_smokes_label_calibration                      | view  | ke
 public | person_smokes_label_inference                        | view  | ke
(16 rows)
```

查看推理结果



```
smoke_test=# select * from person_has_cancer_label_inference;
 person_id | id | label | category | expectation
-----------+----+-------+----------+-------------
         1 |  0 |       |        1 |       0.633
         3 |  2 |       |        1 |       0.628
         4 |  3 |       |        1 |       0.589
         5 |  4 |       |        1 |       0.582
         2 |  1 |       |        1 |       0.573
         6 |  5 |       |        1 |       0.558
(6 rows)
```