OST
Eastern Switzerland
University of Applied Sciences

Command Line Interface Tool for Automatized Database Benchmarks

Institute: Eastern Switzerland University of Applied Science

Program: MSc Computer Science

Course: DB Seminar

Author: Roman Bögli

Supervisor: Prof. Stefan F. Keller

Stage: *interim*

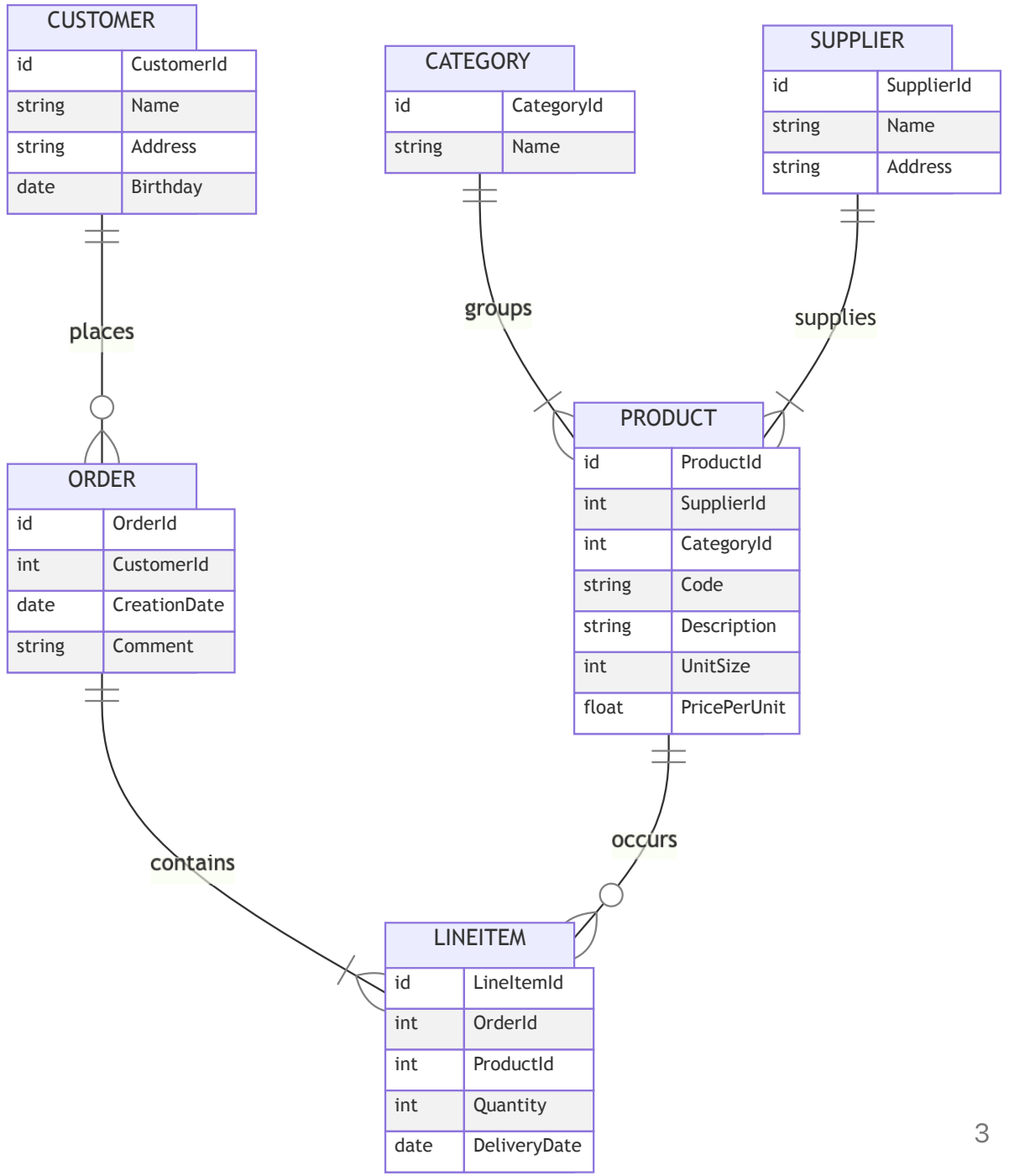
Date: April 14, 2023

Content

- Relational DBMS vs. Graph-Based DBMS
- Tool `gobench`
- Synthetic Script & Substitution
- Automation
- Result Analysis

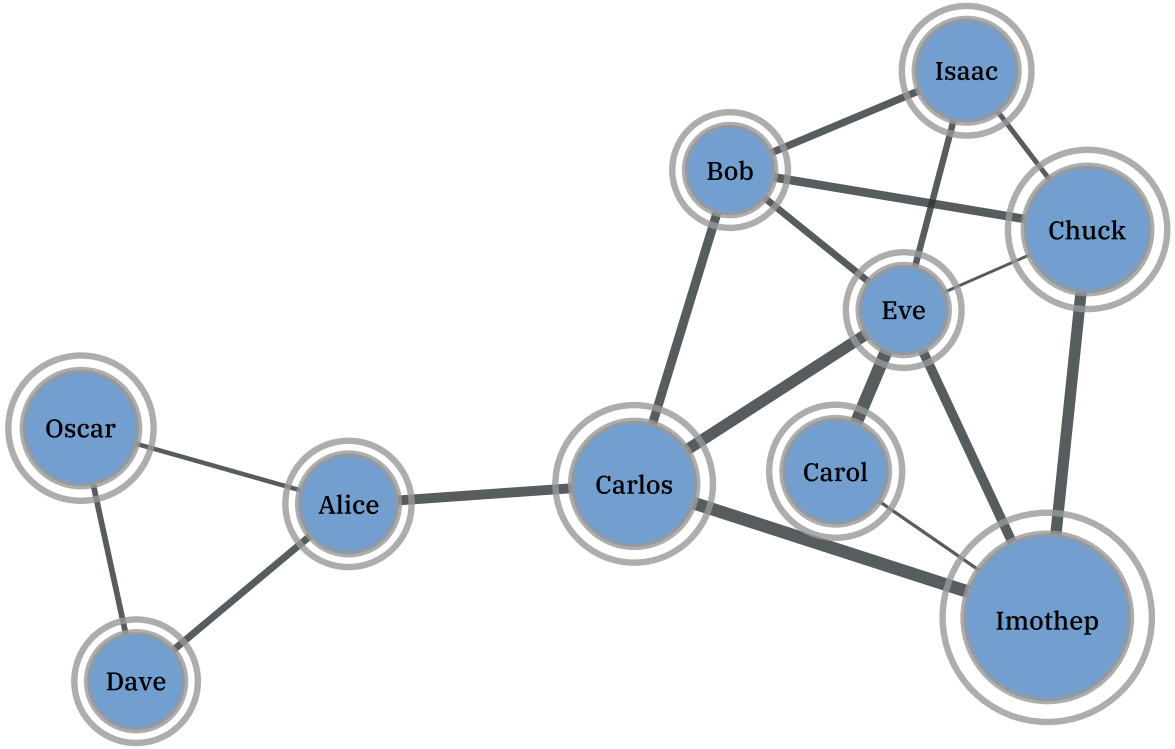
Relational DBMS

- tables are entities
- relationships using keys
- homogenous data through schema



Graph-Based DBMS

- attributed nodes and edges
- relationships are first class elements
- heterogenous data (schema-less)



Query Languages

Query adult customers

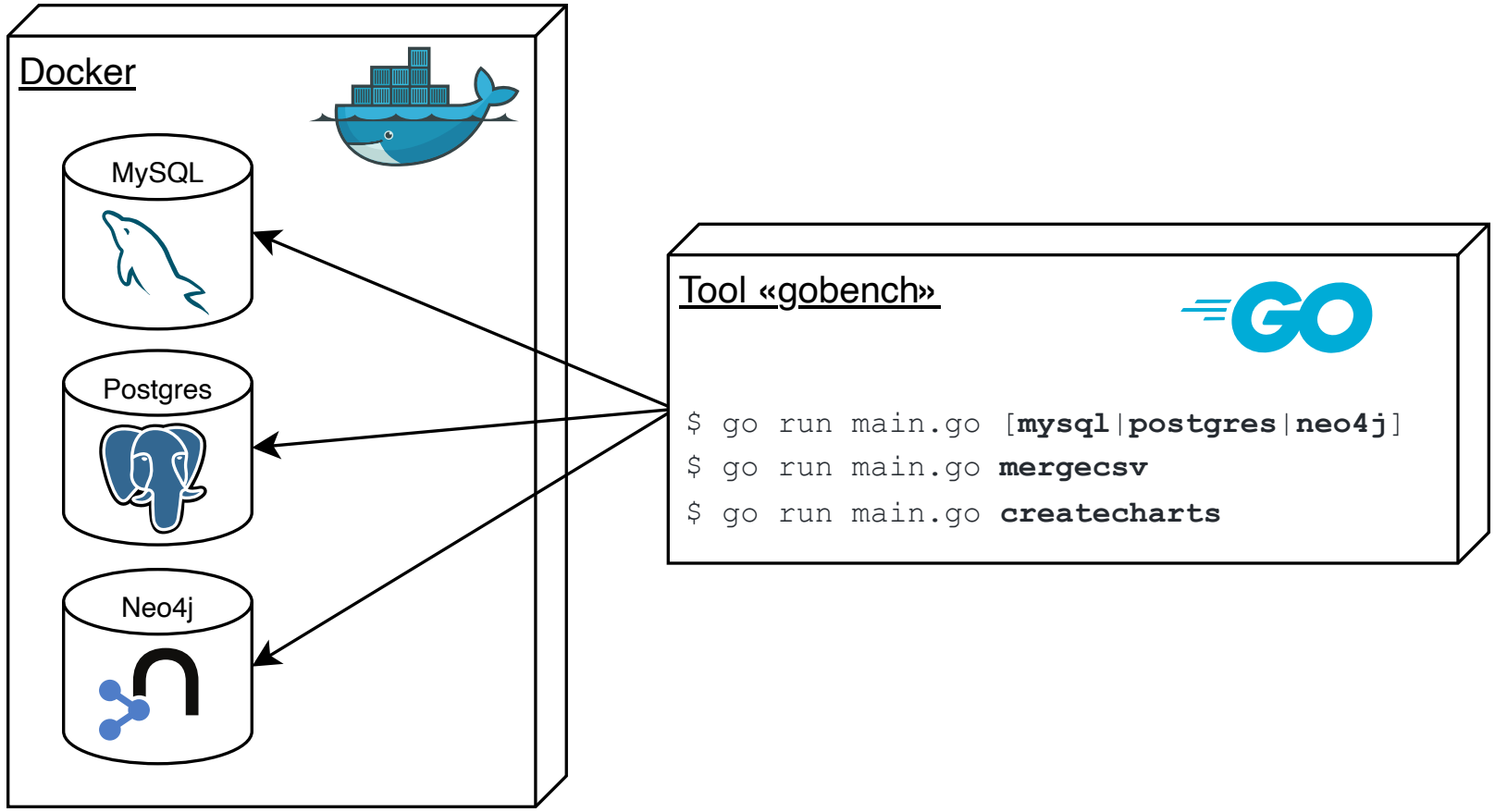
```
-- SQL
SELECT * FROM Customer c WHERE c.Age >= 18

-- Cyper
MATCH (c:Customer) WHERE c.Age > 18 RETURN c;
```

Show top clients based on revenue

```
-- SQL
SELECT c.CustomerId, c.Name, SUM(p.Total) FROM Customer c
INNER JOIN Purchase p on c.CustomerId = p.CustomerId
GROUP BY c.CustomerId, c.Name ORDER BY SUM(p.Total) DESC

-- Cyper
MATCH (c:Customer)-[:MAKES]->(p:Purchase)
RETURN c.Name, SUM(p.Total) AS TotalOrderValue ORDER BY TotalOrderValue DESC
```

System Setup

- requirements:
  - Docker
  - Go
  - gobench



Command Line Interface (CLI)

- open terminal and navigate to the location of `main.go`
  `$ cd ~/path/to/gobench/cmd`
- interact with `go run main.go` to see flags

```
▶ go run main.go -h
Available subcommands:
        mysql | postgres | neo4j | mergecsv | createcharts
        Use 'subcommand --help' for all flags of the specified command.
pflag: help requested
exit status 2
```

Possilbe CLI Commands

```
# run synthetic INSERT and SELECT statements against MySQL, each 100x
$ go run main.go mysql --host 127.0.0.1 --port 3306 --user "root" \
        --pass "password" --iter 100 --run "inserts selects"
```

```
# run statemets of custom script against Postgres, save results in file
$ go run main.go postgres --host 127.0.0.1 --port 5432 --user "postgres" \
        --pass "password" --iter 100 --script "./path/to/mysql.sql" \
        --writecsv "./path/to/results/mysql.csv"
```

```
# merge serveral result files
$ go run main.go mergecsv \
        --rootDir "~/path/with/csv-files/to-be-merged"
        --targetFile "~/anypath/allresults.csv"
```

```
# merge serveral result files
$ go run main.go createcharts \
        --dataFile "~/anypath/allresults.csv" --charttype "line"
```

Custom Script

```
-- INIT
\benchmark once \name initialize
DROP SCHEMA IF EXISTS gobench CASCADE;
CREATE SCHEMA gobench;
CREATE TABLE gobench.Customer (CustomerId INT PRIMARY KEY, Name VARCHAR(10), ... );
CREATE TABLE gobench.order (OrderId INT PRIMARY KEY, CustomerId INT NOT NULL, ... );

-- INSERTS
\benchmark loop 1.0 \name inserts
INSERT INTO gobench.Customer (CustomerId, Name, Address, Birthday)
VALUES ( {{.Iter}}, '{{call .RandString 3 10 }}', '{{call .RandString 10 50 }}', '{{call .RandDate }}');

INSERT INTO gobench.Order (OrderId, CustomerId, CreationDate, Comment)
VALUES( {{.Iter}}, (SELECT CustomerId FROM gobench.Customer ORDER BY RANDOM() LIMIT 1),
        '{{call .RandDate }}', '{{call .RandString 0 50 }}');

-- SELECTS
\benchmark loop 1.0 \name select_simple
SELECT * FROM gobench.Customer WHERE CustomerId = {{.Iter}}

-- CLEAN
\benchmark once \name clean
DROP SCHEMA IF EXISTS gobench CASCADE;
```

Statement Substitutions

Sequences of the following patterns will be substituted before the statement is executed:

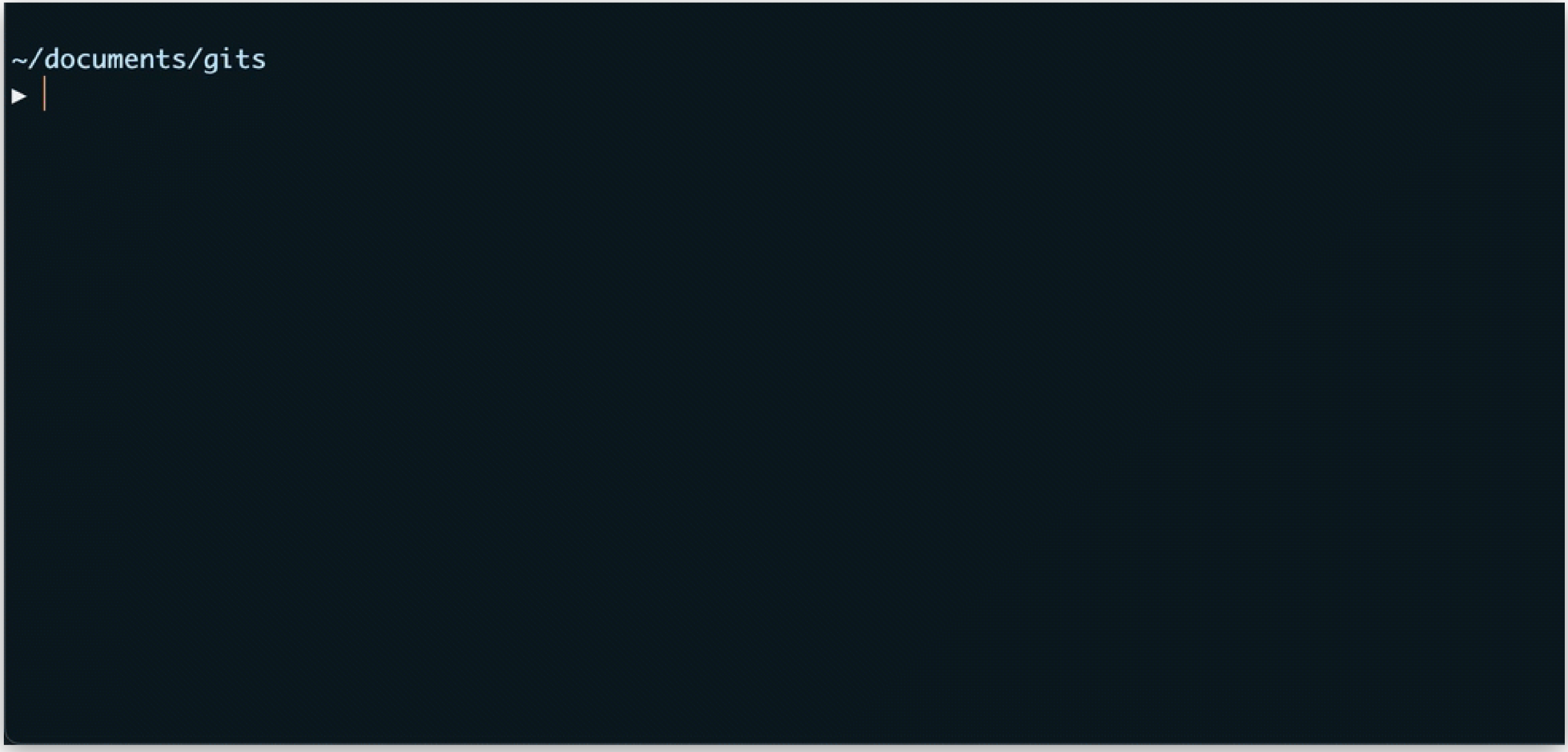
`{{.Iter}}` --> The iteration counter. Will return 1 when `\benchmark once`.
`{{call .RandIntBetween 1 100}}` --> Random integer between `1` and `100`.
`{{call .RandFloatBetween 0 1}}` --> Random float between `0` and `1`.
`{{call .RandString 3 15}}` --> Random string with length between `3` and `15`.
`{{call .RandDate}}` --> Random date.

Bash Script Doing The Work

```
$ bash bashscript.sh
```

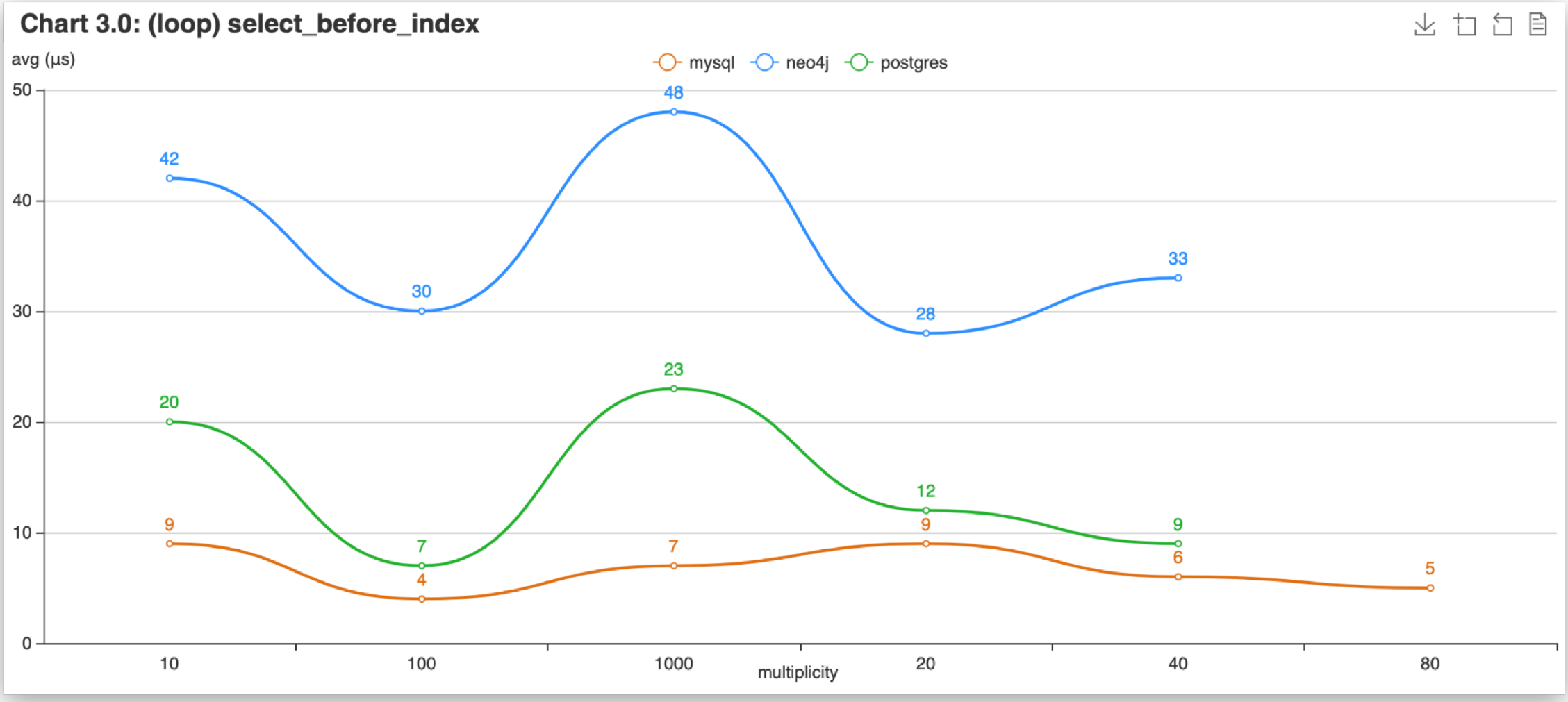
```
start_time=`date +%s`
echo -e "\nSTART BENCHMARKING...\n"
for MULT in "${MULTIPLICITIES[@]}"; do
    echo $(for i in $(seq 1 50); do printf "_"; done)
    echo -e "\nITERATIONS: ${MULT}"

    echo -e "\nTEST MYSQL"
    go run $gobench_main_path mysql \
        --host $db_host \
        --port $mysql_port \
        --user $mysql_user \
        --pass $mysql_pass \
        --iter $MULT \
        --threads $threads \
        --script "${script_base_path}/${script_set}/mysql.sql" \
        --writecsv "${result_base_path}/${script_set}/mysql_${MULT}.csv"
```

> see showcase on next slide...


~/documents/gits
▶

Result Analysis

- generating a `chart.html` file with various visualizations



thanks!