

# Resource Usage

## Database Administration Lab Guide 1

2023/2024

Consider a simplified invoice processing system with the following database schema (Figure 1):

**Client:** Id, Name, Address, Data.

**Product:** Id, Description, Data.

**Invoice:** Id, ProductId, ClientId, Data.

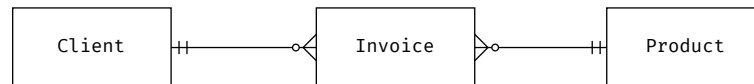


Figure 1: Benchmark's schema

The application provides the following operations:

**Sell:** Add invoice record.

**Account:** List names of products sold to some client.

**Top10:** List currently 10 most sold products

Using the provided benchmark, explore the relation between performance and configuration of memory management parameters:

- `shared_buffers`
- `work_mem`

### Steps

1. Adjust concurrency level (number of clients) and database size for the default configuration
2. Adjust configuration parameters, down and up, and repeat the benchmark.

### Questions

1. What is the impact of work memory and shared buffer allocation in the maximum achievable throughput?
2. Can the baseline performance be improved?

**Learning Outcomes** Relate resource usage with performance. Use memory configuration parameters to influence system performance.

# PostgreSQL HowTo

## With Docker

1. Create the container:

```
$ docker run --name postgres -e POSTGRES_PASSWORD=postgres \
  -p 5432:5432 -d postgres:16
```

2. Access the psql client:

```
$ docker exec -it postgres psql -U postgres
```

3. Create a new database testdb:

```
# in psql
psql> create database testdb;

# with createdb
docker exec -it postgres createdb -U postgres testdb
```

4. Connect to the new database:

```
psql> \c testdb
```

5. Get the list of relations:

```
psql> \d
```

6. To restart the server:

```
$ docker restart postgres
```

7. To stop the container:

```
$ docker stop postgres
```

## Update system parameters

```
-- check the parameter's current value (psql)
SHOW <param>;

-- change the value for the current session
-- (reverts back to the previous value when we
-- open a new session; not valid for shared_buffers)
SET <param> = <x>;

-- change the value permanently
-- (for shared_buffers, we need to restart the server;
-- for most other parameters the pg_reload_conf() is enough)
ALTER SYSTEM SET <param> = <x>;
SELECT pg_reload_conf();

-- change the value permanently, by editing the config file
SHOW config_file;
-> /path/to/postgresql.conf
-- edit the file, e.g., with vim
vim ...
-- restart the server
```