CERN School of Computing on IT Services 2024
Database Services exercises

DBOD Exercises:

The DBOD user guide:	https://dbod-user-guide.web.cern.ch/
The DBOD portal:	https://dbod.web.cern.ch/

Exercise 1. Requesting an instance

Go to the following link and write your CERN Logins to reserve an instance for yourselves:
https://cern.ch/dbod-allocation

Go to the DBoD portal and request a new PgSQL instance.
Use the following details:

DB_Name:	csc_pg*NN*
Category:	TEST
Admin group*:	csc-cc-users
Database Type:	Postgres 13.13

* - people in this e-group will have admin access to your database. **Avoid using big e-groups!**

While the instance is being created, please continue with the next exercises.
You will get an e-mail when your instance is ready.

DBOD Exercises:

The DBOD user guide:	https://dbod-user-guide.web.cern.ch/
The DBOD portal:	https://dbod.web.cern.ch/

Exercise 2. Using the DBoD portal

In preparation for the exercises, we have pre-created some MySQL instances. Each pair of students have reserved one. Visit the DBoD portal and collect some information about the MySQL instance that you have reserved for yourself.

a. Who is the current owner of the instance?
b. Put both of your names in the description of the instance. Make sure to save your changes by pressing Enter.
c. Set the expiry date to 14.11.2024
d. Review the history of the jobs concerning the database
e. Review the log of the instance
f. The my.cnf file hosts the database parameter configuration. Find the maximum number of concurrent connections to your database.
g. Try changing the innodb_buffer_pool_size to 256M, remember to submit your changes
h. What is the hostname and port number that you would use to connect to this instance? Note it down.
i. Can you change the owner of this instance?
j. Open the PostgreSQL instance you requested yesterday
k. Set the expiry date to 14.11.2024 for the PostgreSQL instance
l. Try assigning the PostgreSQL instance to the other person in your pair. The other person will get an e-mail and needs to approve.

Exercise 3. Logging into the MySQL instance

Log into the database you have investigated in Exercise 2.

If you need help, refer to the DBOD user guide to find necessary information on how to connect. You should use the mysql console client running on one of the lxplus nodes. A Graphical User Interface client might be tricky to set-up, because the DBoD instances are not available publicly from the Internet.

Username: *admin*
Password: *changeme*

Run the following query to check the value of the parameter you changed in Exercise 2:

```
select @@innodb_buffer_pool_size/1024/1024;
```

Is this the result you expected?

Keep your connection open and go to Exercise 4 for explanation.

Exercise 4. Managing your DBoD instances
Make sure you completed Exercise 3 before continuing with this exercise.

When you changed the value of your parameter, you have only changed the static file with the configuration. For your changes to be applied, you need to restart your database instance. Open the DBoD portal and look for the On/Off menu. Issue a restart of your instance.

Remember that restart operation means a downtime of your database!

Go back to the MySQL client and issue this command again. Are you satisfied with the result?

If the value still hasn't changed: go back to the DBoD portal, look at the Jobs tab. Click the "refresh jobs" button and wait until the database is restarted. It might take a few minutes because there are many people issuing jobs at the same time.

You would use the DBoD portal in a similar way to modify your PostgreSQL instance configuration. For some parameter PostgreSQL changes, the configuration can be reloaded without causing any downtime. There's another option in the menu for that.

Exercise 5. Creating backups
DBoD service offers automated backup & recovery services. A backup is performed automatically every day, but you can request additional ones at any given time (e. g. before any major operation). Let's now explore the backup & restore options of the DBoD service.

Before we start the exercise on backups let's create some fake data on your MySQL instance:

```
    create database hr;
    use hr;
    create table employees (name varchar(20),salary integer);
    insert into employees values ('Emanuel Macron',5000);
    insert into employees values ('Donald Trump',100000);
    select * from employees;
```

Let's now run a backup of the database. Go to the DBoD portal and look for the Backup and Restore tab. Find a button to create backup. You can follow the status of the operation on the Jobs tab.

We will now simulate some maintenance operation going wrong. One type of frequent errors is human mistakes when writing SQL statements. Let's pretend that someone wanted to remove "Donald Trump" from the list of employees but forgot to put the WHERE clause.
Go ahead and run:

```
    select now(); -- print the time to know when the delete happened
    delete from employees;
```

You can confirm that the employees table is now empty:

```
    select * from employees;
```

Go to the DBoD portal and run the restore, select a point in time which is before the delete command. You can follow the status of the recover operation on the Jobs tab (use the button to refresh the job list). It might take several minutes for your recover to finish. Feel free to **start the next exercise while you wait** for the operation to finish.
When the operation is done, run the following to view the table contents after the restore:

```
    use hr;
    select * from employees;
```

Exercise 6. Logging into the PostgreSQL instance

By now, you should have received an e-mail with details on how to connect to your PostgreSQL instance. The database should be also visible in the DBoD portal. Try connecting to your PostgreSQL using a preferred client. Refer to DBoD user guide for guidance.
Use the username and password from the e-mail.

Run the following query to verify the version of your database:

```
select version();
```

Go to the DBoD user guide, look for Versioning Policy of the PostgreSQL.

a. When is the end of support for the version that you're running?
b. Discuss how you would plan an upgrade operation of a production system with a minimal risk of losing data and trying to minimize the downtime.
c. How could you test if your application works with the new version of the database before performing the upgrade itself? Let's explore that in the next exercise.

Exercise 7. Clones

We will now try to spin up a clone of your PostgreSQL instance.
Beware! The clones are based on backups, hence, to create a clone you need to have at least one backup.

a. Open the DBoD portal and locate your PostgreSQL instance
b. Make sure that you have at least one backup of your instance. If everything went well, at least one backup should be automatically created for you.
c. Go to the clones tab and start the process of creation of a clone based on a recent backup. Read the message carefully.
d. While you wait, please go to the DBoD User Guide (you can find the link at the bottom of the DBoD portal) and familiarize yourself with the documentation of the cloning process.
e. In the documentation, find:
    a. How many clones of the same database you can have?
    b. How much time do you have before the clone will be automatically expired?
f. Due to a limitation of how security of the DBoD portal is implemented you need to **log out** of the dbod portal and log in again. Alternatively, you can use the privacy/incognito mode of your browser.
g. The clone might still not be visible as the process will take probably at least 30 minutes but might be even more than an hour due to many concurrent operations. Feel free to continue to the next exercise while you wait.
h. Try logging into your clone using a PostgreSQL client. The password to access the clone are the same as the source copy.

Exercise 8. Upgrade
We will now try to go through the upgrade of the PostgreSQL instance. Familiarize yourself with the documentation concerning the Upgrade process.
Read the following documents from the DBoD User Guide:

- *Instance management -> Database Upgrades -> Upgrading your instance*
- *Instance management -> Database Upgrades -> PostgreSQL upgrading*
- *Instance management -> Database Upgrades -> Upgrade checker*

For your information: The process for the MySQL upgrades is quite similar, you can check out the MySQL specific information if you want.

To summarize: the recommended way to perform an upgrade is to prepare a clone of your database and test the upgrade process on the clone. After confirming that everything worked smoothly on the clone, you would perform the same operations on your "live" database.

You should perform the next steps on your clone if it's already available. If you're still waiting for the clone to be ready, you can run the next steps on your "live" database.

a. Open your PostgreSQL instance in the DBoD portal
b. Run the upgrade checker to generate a new checker report
c. Observe the status of the checker job on the jobs tab. Wait for the job to be finished.
d. Open the "upgrade checker reports" link from the dropdown menu
e. Open the report from the upgrade checker, you can see the log of the execution
f. Go back to the DBoD portal, the upgrade option should now be available
   If the upgrade option is still greyed out, try reloading the page
g. Start the upgrade process. As usual, you can monitor the status in the jobs tab
h. Once the job is finished try logging into the database and query the version:

```
select version();
```

i. Congrats on your first database upgrade!

The process for upgrading a MySQL database would be similar. You can give it a try if you want.

Exercise 9*. MySQL character conversion
It's an optional exercise if you want to continue working on exercises with the DBoD.

With MySQL version 8, the character sets utf8 and utf8mb3 become deprecated and will be removed in some future release. If you are using the utf8 or utf8mb3 character sets in your database, we recommend converting those character sets to utf8mb4.

Your pre-created MySQL database contains some tables using the deprecated character sets.

a. Open the DBoD User Guide and familiarize yourself with the following document: *Instance management -> MySQL character set conversion*
b. Open your MySQL instance in the DBoD portal
c. Log into your MySQL instance and run the following:

```
show full columns from mock_database.reviews;
```

d. You can notice some columns using the utf8mb3_general_ci
e. Try following the MySQL character conversion procedure to convert your database to utf8mb4 character set
f. After running the conversion, run the same query to verify the results

Exercise 10*. Configuring a GUI client to access your database
It's an optional exercise if you want to continue working on exercises with the DBoD.

Try installing a GUI database client on your computer or configuring your favorite IDE to connect to a database hosted in the DBoD service. You can choose the MySQL or PostgreSQL database or both. Choose whichever client you want. It can be your favorite client, DBeaver, MySQL Workbench or any other.

Remember! Direct network access to a DBoD database from outside CERN's network is not possible. You need to use some form of tunneling to access the DBoD resources.

Feel free to use the following ssh command as a hint:

```
ssh -L 5555:dbod-<DB_NAME>.cern.ch:<PORT> <CERN_LOGIN>@lxtunnel.cern.ch
```

The ssh command will tunnel any traffic going into **your local port 5555** to the remote port specified as **dbod-<DB_NAME>.cern.ch:<PORT>**

The forwarding will work for as long as your ssh session is running.

Because of the tunneling, you need to configure your database client to use hostname: *localhost* and the port: *5555* (as specified as the first part of the -L definition).