

Database creation for Idea-case

idea-case-backend – Juhani Välimäki

8.2.2023



Haaga-Helia

0. Teacher created a Virtual machine, database and there sandboxes (=schemas) for 70 DB users

- To the Finnish **CSC**'s cloud, cPouta machine. Here are the installation notes / steps **if** someone is interested: (extra information)
 - <https://github.com/haagahelia/linux-servers-etc/>
 - https://github.com/haagahelia/linux-servers-etc/blob/main/CSC_virtual_machine_and_user_creation.md (Linux and its 2 users)
- And here are the steps used to create the MariaDB, and 70 schemas and users to that database.
 - https://github.com/haagahelia/linux-servers-etc/blob/main/mariadb_installation.md

View to some of the creation steps in the cloud...

Here the Virtual Machine resources selected.

And the Ubuntu base image.

Launch Instance

Details *

Access & Security

Networking *

Network Ports

Post-Creation

Advanced Options

Availability Zone

nova

Instance Name *

mariadb-test-server-for-few-weeks

Flavor * ?

standard.xlarge

Number of Instances *

1

Instance Boot Source * ?

Boot from image

Image Name

Ubuntu-22.04 (2.2 GB)

Specify the details for launching an instance.

The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

Name	standard.xlarge
VCPUs	6
Root Disk	80 GB
Ephemeral Disk	0 GB
Total Disk	80 GB
RAM	16,000 MB

Project Limits

Number of Instances

1 of 8 Used

Number of VCPUs

4 of 8 Used

Total RAM

8,000 of 33,000 MB Used

Number of Volumes

0 of 10 Used

Total Volume Storage

0 of 1,000 GiB Used

Cancel

Launch

View to some of the creation steps – strictish Firewall

Project

API Access

Compute

Volumes

Network

Network Topology

Networks

Routers

Security Groups

Floating IPs

Orchestration

Object Store

Identity

Project / Network / Security Groups / Manage Security Group Rule...

Manage Security Group Rules:
mariadb_security_group_23k (3c...266-f...d-42c5-a4b1-e1a...eb4d44e)

+ Add Rule

Delete Rules

Displaying 3 items

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Egress	IPv6	Any	Any	::/0	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	Delete Rule

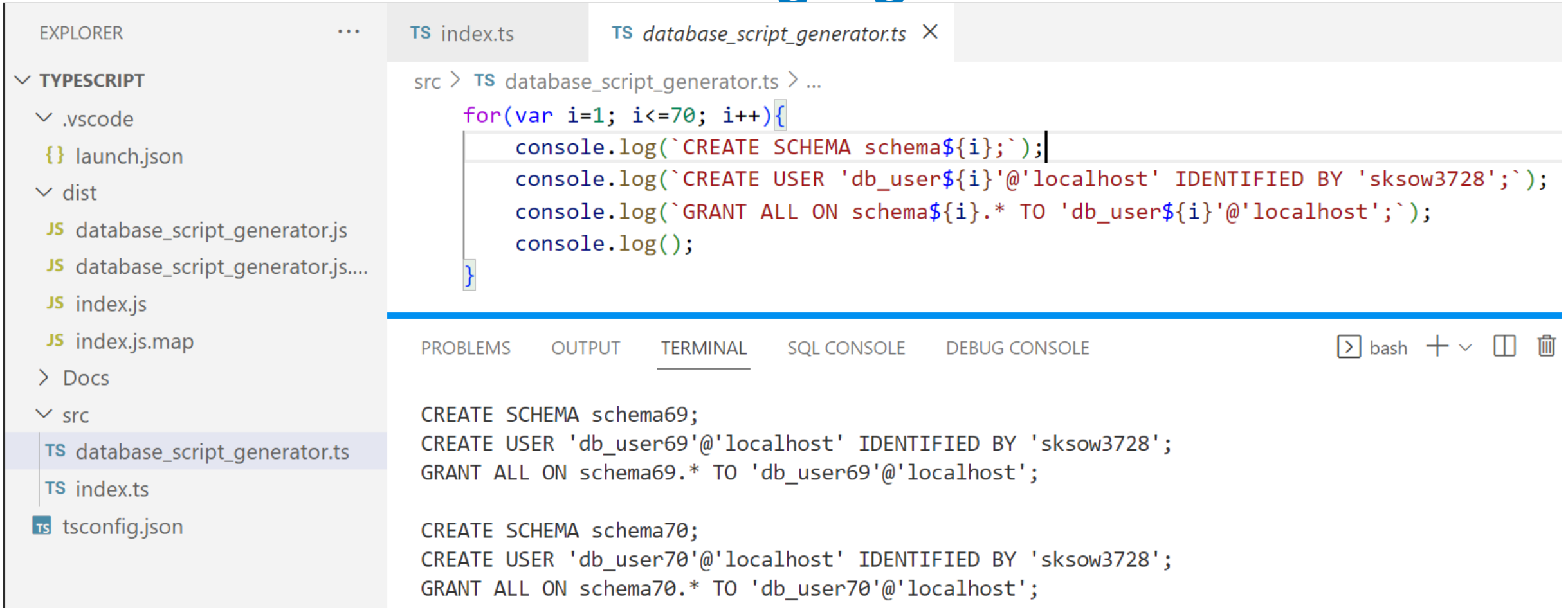
Displaying 3 items

View to some of the creation steps in the cloud...

Here running the mysql_secure_installation hardening script or wizard against installed MariaDB server

```
jyser1@mariadb-test-server-for-few-weeks: ~  
jyser1@mariadb-test-server-for-few-weeks:~$ sudo mysql_secure_installation  
  
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB  
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!  
  
In order to log into MariaDB to secure it, we'll need the current  
password for the root user. If you've just installed MariaDB, and  
haven't set the root password yet, you should just press enter here.  
  
Enter current password for root (enter for none):  
OK, successfully used password, moving on...  
  
Setting the root password or using the unix_socket ensures that nobody  
can log into the MariaDB root user without the proper authorisation.  
  
You already have your root account protected, so you can safely answer 'n'.  
  
Switch to unix_socket authentication [Y/n] n  
... skipping.  
  
You already have your root account protected, so you can safely answer 'n'.  
  
Change the root password? [Y/n] Y  
New password:  
Re-enter new password:  
Password updated successfully!  
Reloading privilege tables..  
... Success!  
  
By default, a MariaDB installation has an anonymous user, allowing anyone  
to log into MariaDB without having to have a user account created for  
them. This is intended only for testing, and to make the installation  
go a bit smoother. You should remove them before moving into a  
production environment.  
  
Remove anonymous users? [Y/n] Y  
... Success!  
  
Normally, root should only be allowed to connect from 'localhost'. This  
ensures that someone cannot guess at the root password from the network.  
  
Disallow root login remotely? [Y/n] Y  
... Success!  
  
By default, MariaDB comes with a database named 'test' that anyone can  
access. This is also intended only for testing, and should be removed  
before moving into a production environment.  
  
Remove test database and access to it? [Y/n] Y  
- Dropping test database...  
... Success!  
- Removing privileges on test database...  
... Success!  
  
Reloading the privilege tables will ensure that all changes made so far  
will take effect immediately.  
  
Reload privilege tables now? [Y/n] Y  
... Success!  
  
Cleaning up...  
  
All done! If you've completed all of the above steps, your MariaDB  
installation should now be secure.  
  
Thanks for using MariaDB!  
jyser1@mariadb-test-server-for-few-weeks:~$ sudo systemctl restart mariadb.service  
jyser1@mariadb-test-server-for-few-weeks:~$ sudo lsof -i :3306  
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME  
mariabdb 2339 mysql 20u IPv4 29952 0t0 TCP localhost:mysql (LISTEN)  
jyser1@mariadb-test-server-for-few-weeks:~$
```

View to some of creation steps. 70 DB users and schema for each. Granting rights to that schema



The image shows a Visual Studio Code editor interface. On the left is the Explorer sidebar showing a project structure with folders like .vscode, dist, and src. The src folder is expanded, showing files like database_script_generator.ts and index.ts. The main editor area has two tabs: 'TS index.ts' and 'TS database_script_generator.ts'. The 'database_script_generator.ts' tab is active, showing a TypeScript file with a loop that generates SQL commands for creating schemas and users. The code is as follows:

```
src > TS database_script_generator.ts > ...
for(var i=1; i<=70; i++){
    console.log(`CREATE SCHEMA schema${i};`);
    console.log(`CREATE USER 'db_user${i}'@'localhost' IDENTIFIED BY 'sksow3728';`);
    console.log(`GRANT ALL ON schema${i}.* TO 'db_user${i}'@'localhost';`);
    console.log();
}
```

Below the editor is a panel with tabs for PROBLEMS, OUTPUT, TERMINAL, SQL CONSOLE, and DEBUG CONSOLE. The TERMINAL tab is active, showing the output of the script. The output consists of two blocks of SQL commands, one for schema 69 and one for schema 70:

```
CREATE SCHEMA schema69;
CREATE USER 'db_user69'@'localhost' IDENTIFIED BY 'sksow3728';
GRANT ALL ON schema69.* TO 'db_user69'@'localhost';

CREATE SCHEMA schema70;
CREATE USER 'db_user70'@'localhost' IDENTIFIED BY 'sksow3728';
GRANT ALL ON schema70.* TO 'db_user70'@'localhost';
```

YOUR PART STARTS HERE

1. You need to install the needed tools...

For database connection etc. these are needed:

- (MariaDB or MySQL, if you want to install your own, instead of using my cloud DB)
- **ssh** – for tunnel creation. E.g. GitBash should have this. Maybe Powershell too.
 - Putty could also be used to create the tunnel.
- **DBeaver** – Community Edition. testing the tunnel connection, creating and filling the tables, and possibly creating ER diagrams, looking at the table data while testing, etc.

2. ... and use ssh to create the tunnel (SSH port forwarding)

- The server only has 2 Linux users. You can only use the normal user who has just normal rights
- Only port :22 open, thus you'll need to use the tunnel to connect to this remote MariaDB,
 - cannot access :3306 directly, but need to create a tunnel to it.
- **ssh -f jyser2@86.50.229.46 -L 3306:localhost:3306 -N** (Password given by teacher in Teams>Files)
- Red- and blue-marked parts might change from case to case. E.g. If some other process has already taken port 3306 in your computer, you can use **3308** as the first number.
- Note! Your project .env and such setting must match with the created tunnel. In this case tunnel starts at localhost and **3306** (or **3308**)
- In a true Linux program style the tunnel creation doesn't show anything if no problems 😊
- **lsof -i :3306** (Linux) or **netstat -aof | findstr :3306** (Windows) might help you check if tunnel process still there

2. ... and use ssh to create the tunnel

```
va u@haaga-helia: ~$ ssh -f jyser2@86.50.229.46 -L 3306:localhost:3306 -N
The authenticity of host '86.50.229.46 (86.50.229.46)' can't be established.
ED25519 key fingerprint is SHA256:uL... MTkIP2F60J/... _g4U10aR VCWvys.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '86.50.229.46' (ED25519) to the list of known hosts.
jyser2@86.50.229.46's password:

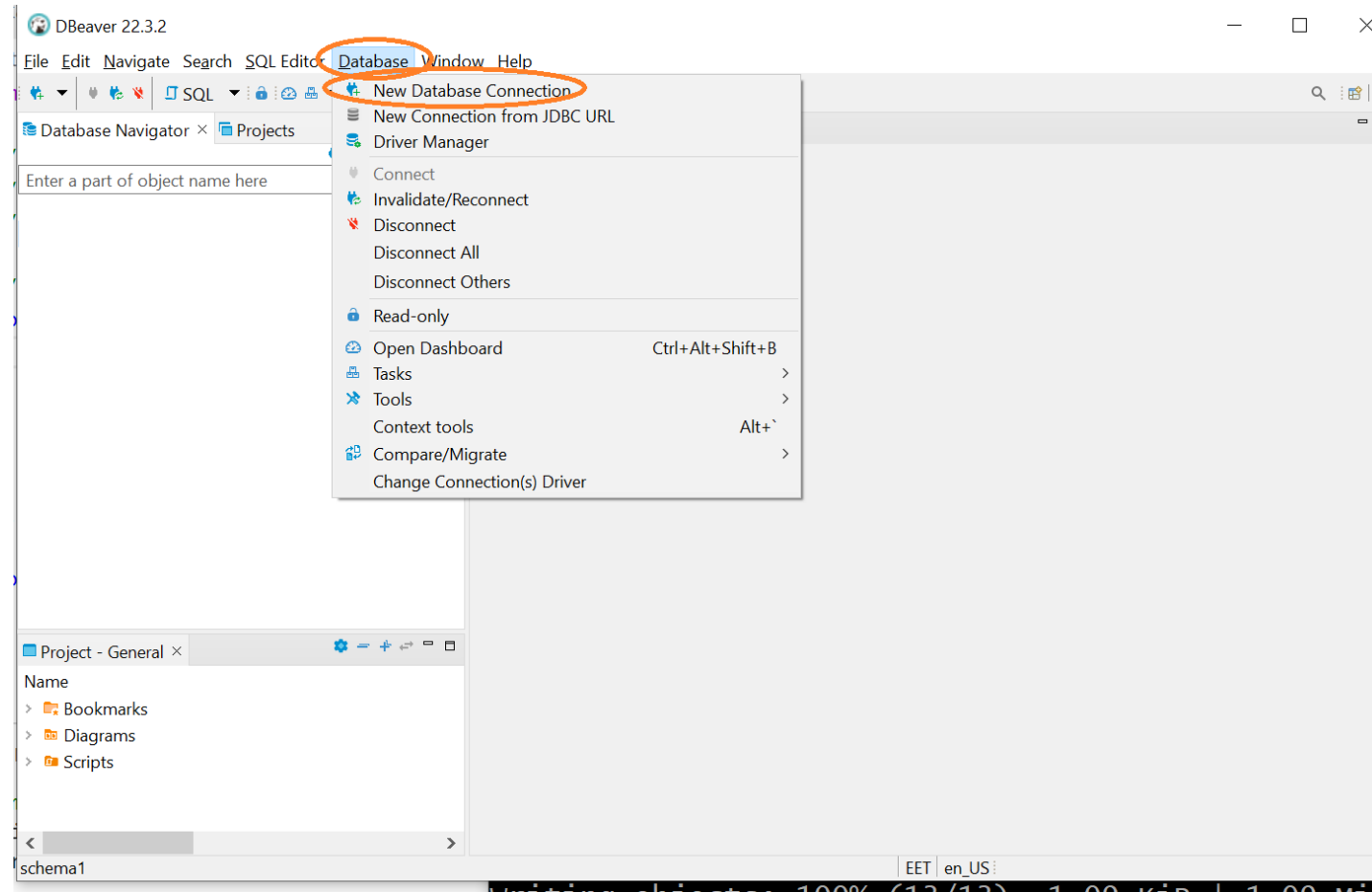
va u@haaga-helia: ~$ ssh -f jyser2@86.50.229.46 -L 3306:localhost:3306 -N
```

netstat -aof in Windows might take some time to produce results. Reason unknown.

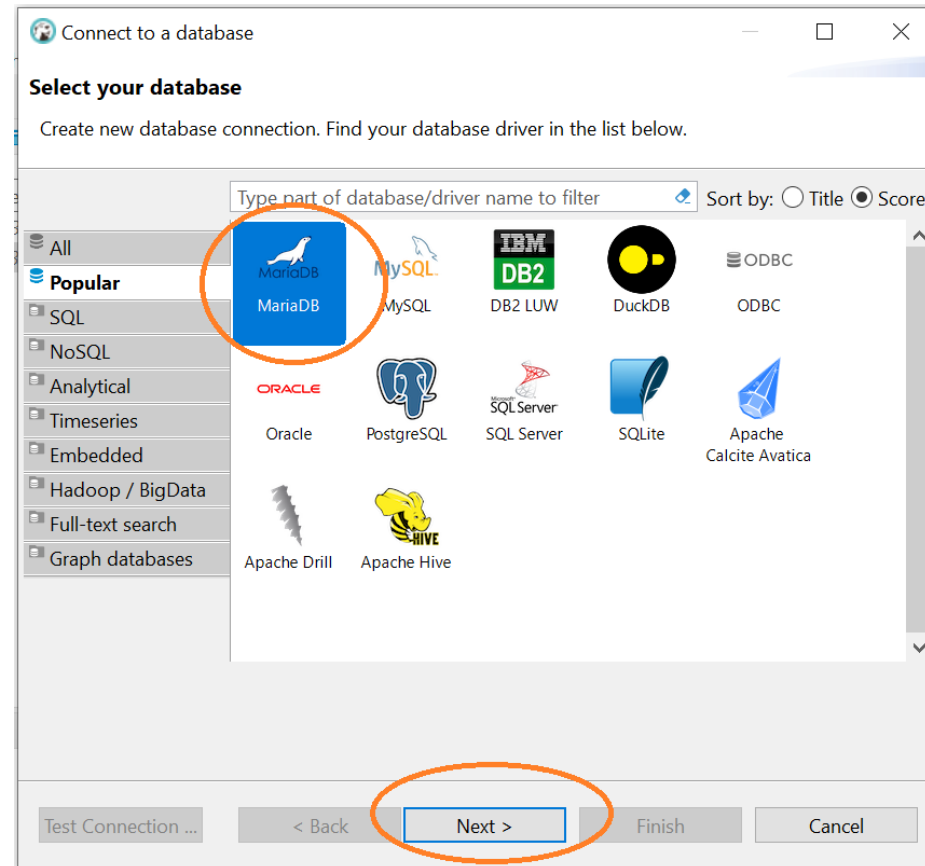
```
va u@haaga-helia: ~$ netstat -aof | findstr 3306
TCP        127.0.0.1:3306           127.0.0.1:0 .haagahelia.amk:0 LISTENING          12016
TCP        127.0.0.1:3306           kubernetes.docker.internal:61200 ESTABLISHED        12016
TCP        127.0.0.1:3306           kubernetes.docker.internal:61200 ESTABLISHED        12016
TCP        [::1]:3306              [::1]:0 .haagahelia.amk:0 LISTENING          12016

va u@haaga-helia: ~$
```

3.1.1 Use DBeaver according to the pictures to create and test the connection



3.1.2 Use DBeaver according to the pictures to create and test the connection



3.1.3 Use DBeaver according to the pictures to create and test the connection

Connect to a database

MariaDB connection settings

Connection Settings

Main Driver properties SSH Proxy SSL

Server

Server Host: localhost Port: 3306

Database: schema99

Authentication (Database Native)

Username: db_user99

Password: [masked] ☐ Save password locally

Advanced

Server Time Zone: Auto-detect

Local Client: MySQL Binaries

(Though if you installed your own MariaDB, then use your values for same fields)

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: MariaDB

Driver Settings Driver license

1. Test Connection ...

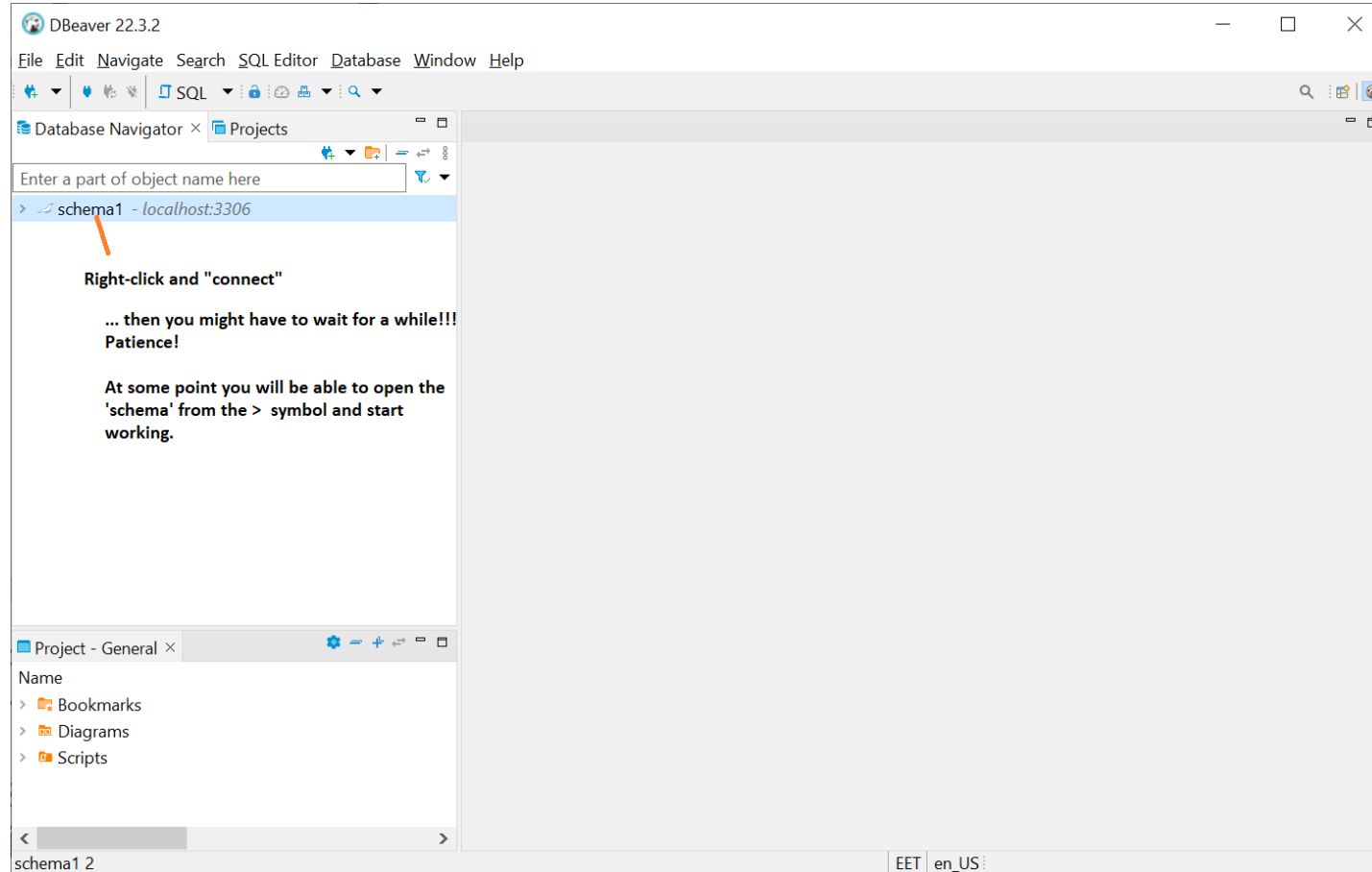
2. Finish

< Back Next > Cancel

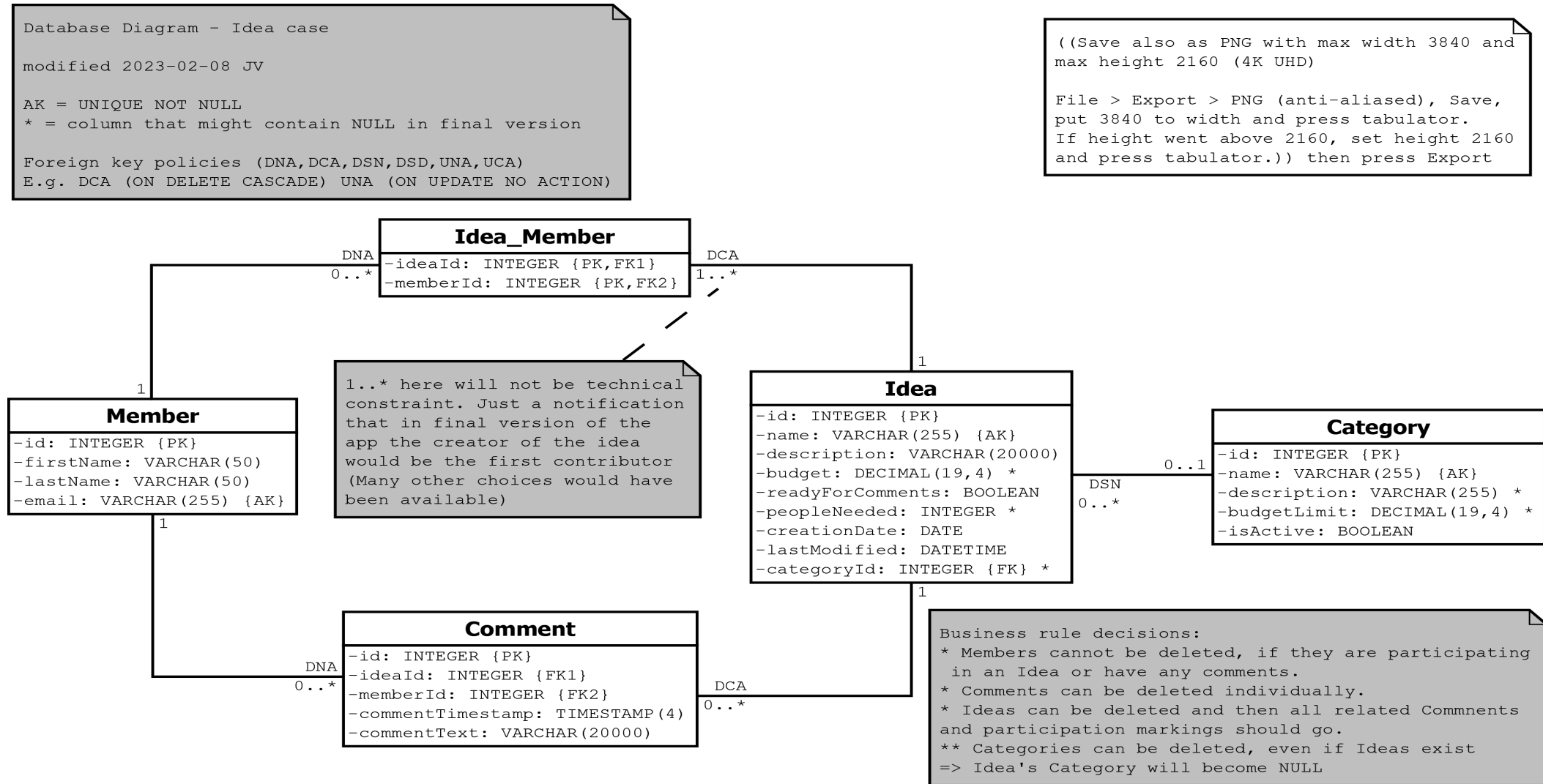
SSH tunnel's your computer's end, isn't it

Remote MariaDB related settings, right

3.1.4 Use DBeaver according to the pictures to create and test the connection

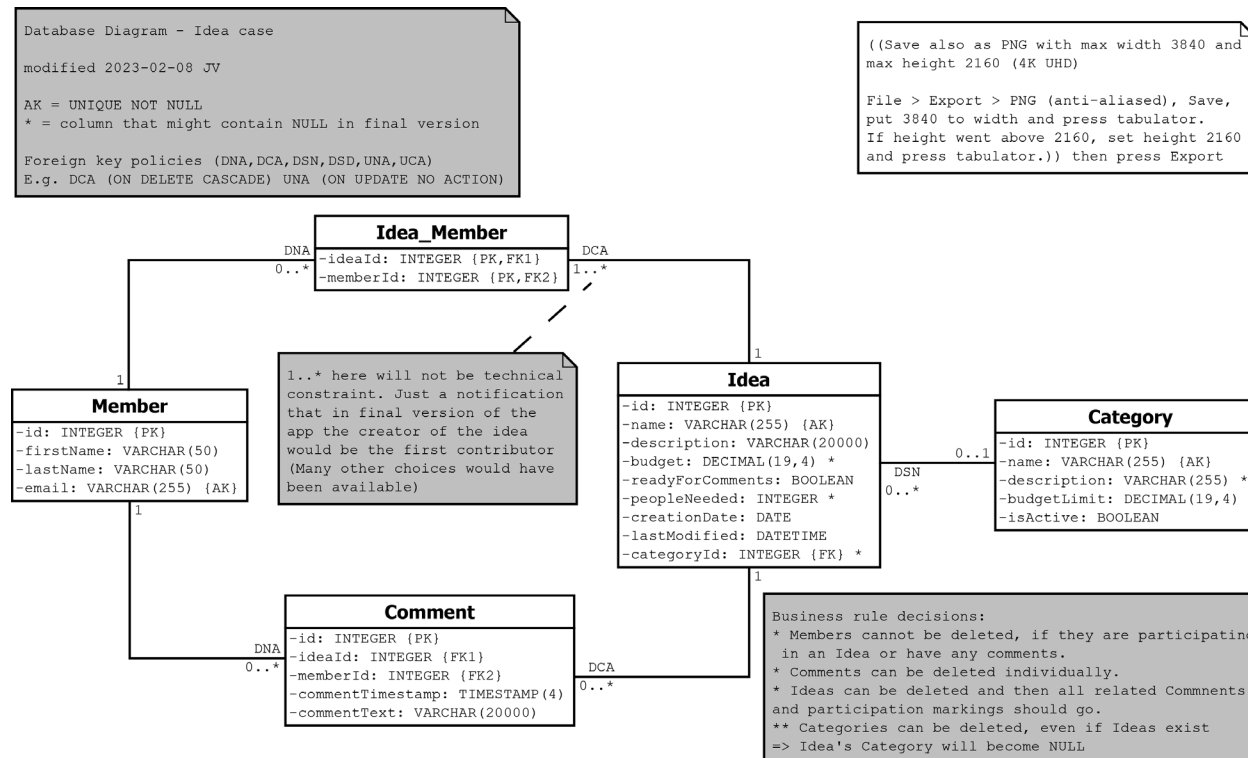


Here is how the database will look like:

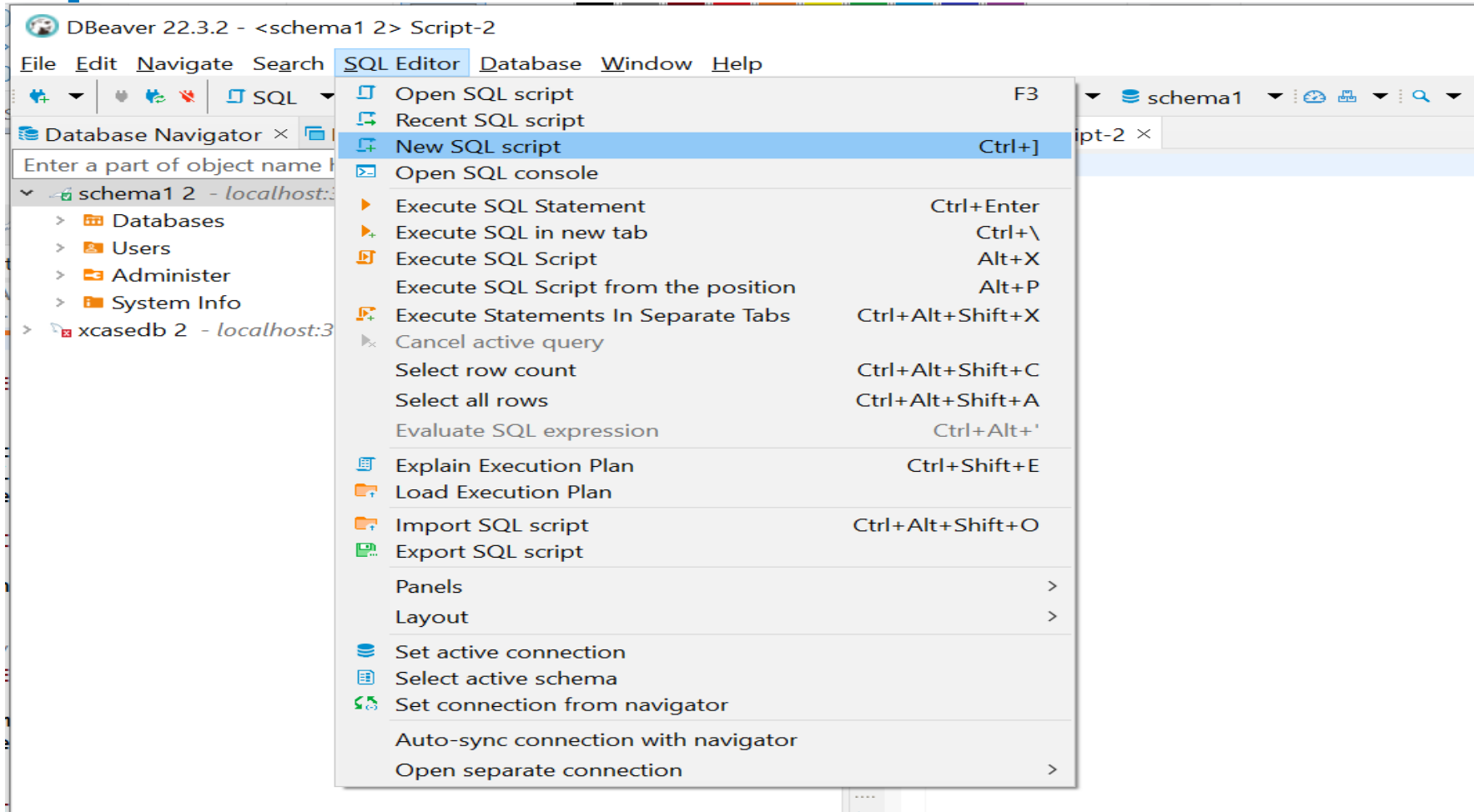


Download the SQL script for creating the database

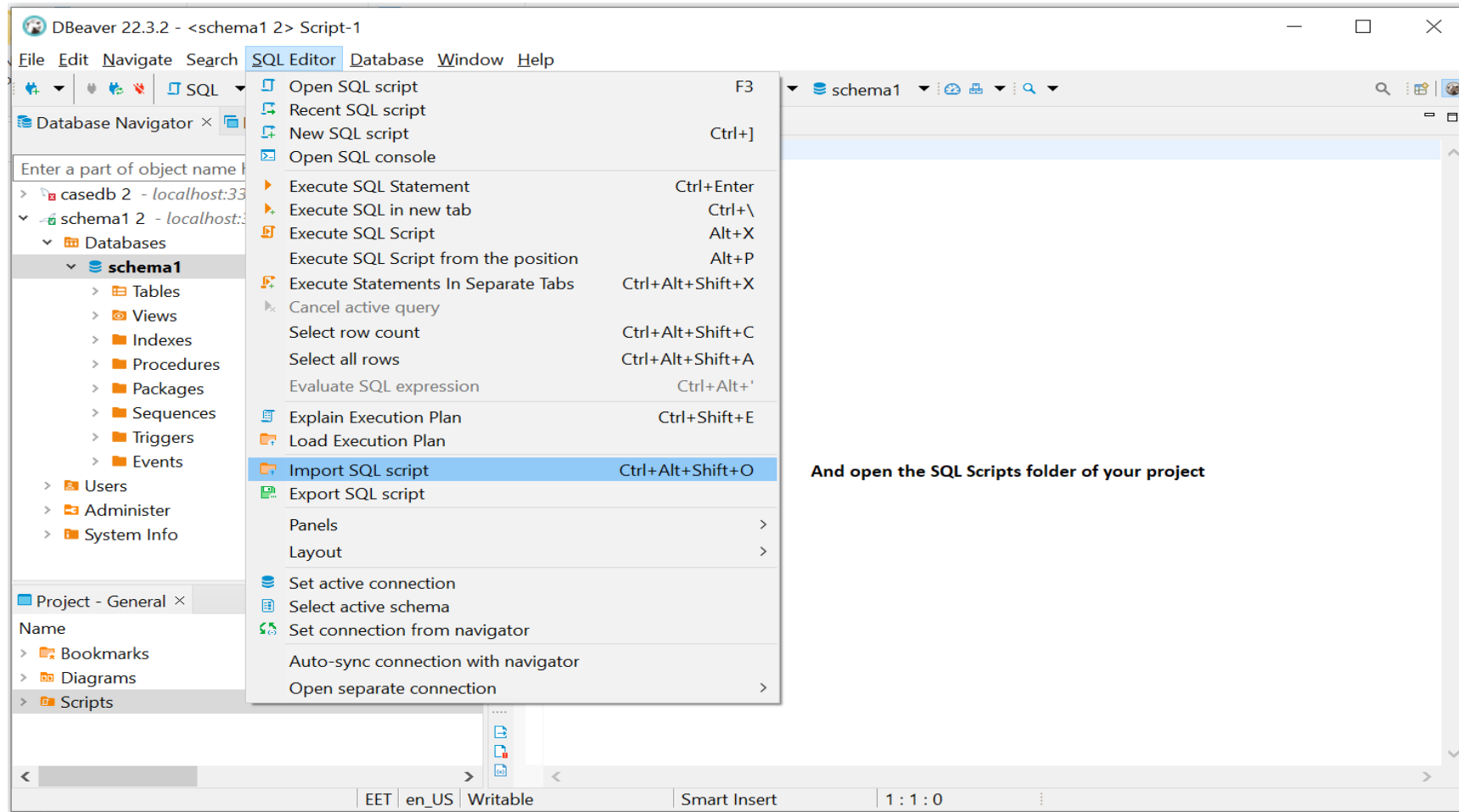
- Now download at least the https://github.com/valju/idea-case-backend/blob/master/Database/SQL_Scripts/000_drop_create_insert.sql this file to some known folder.
- Or just clone the repo
- It drops, creates and populates the needed tables.



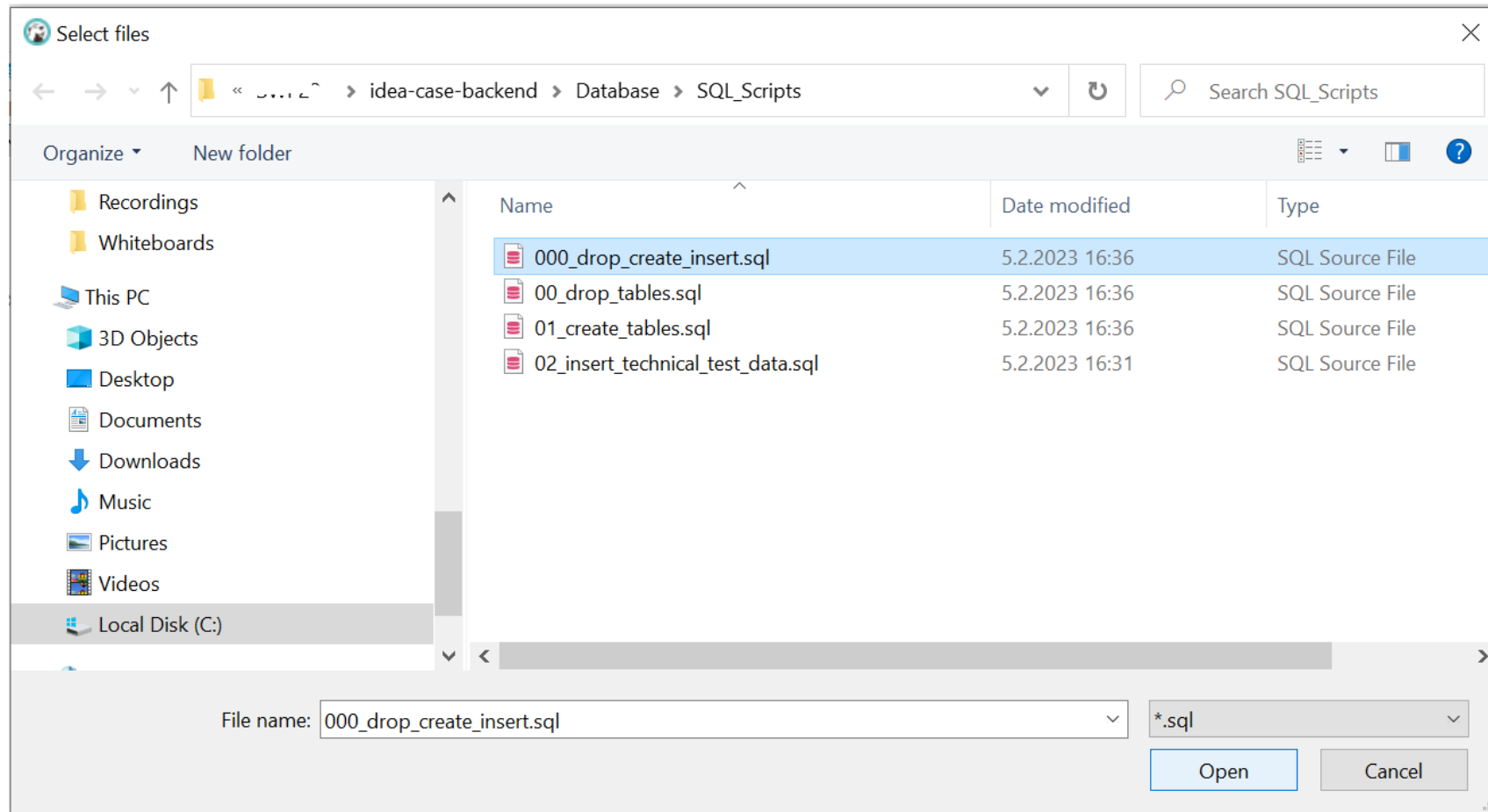
3.2.1 Use DBeaver to run SQL to create the tables and populate with test data



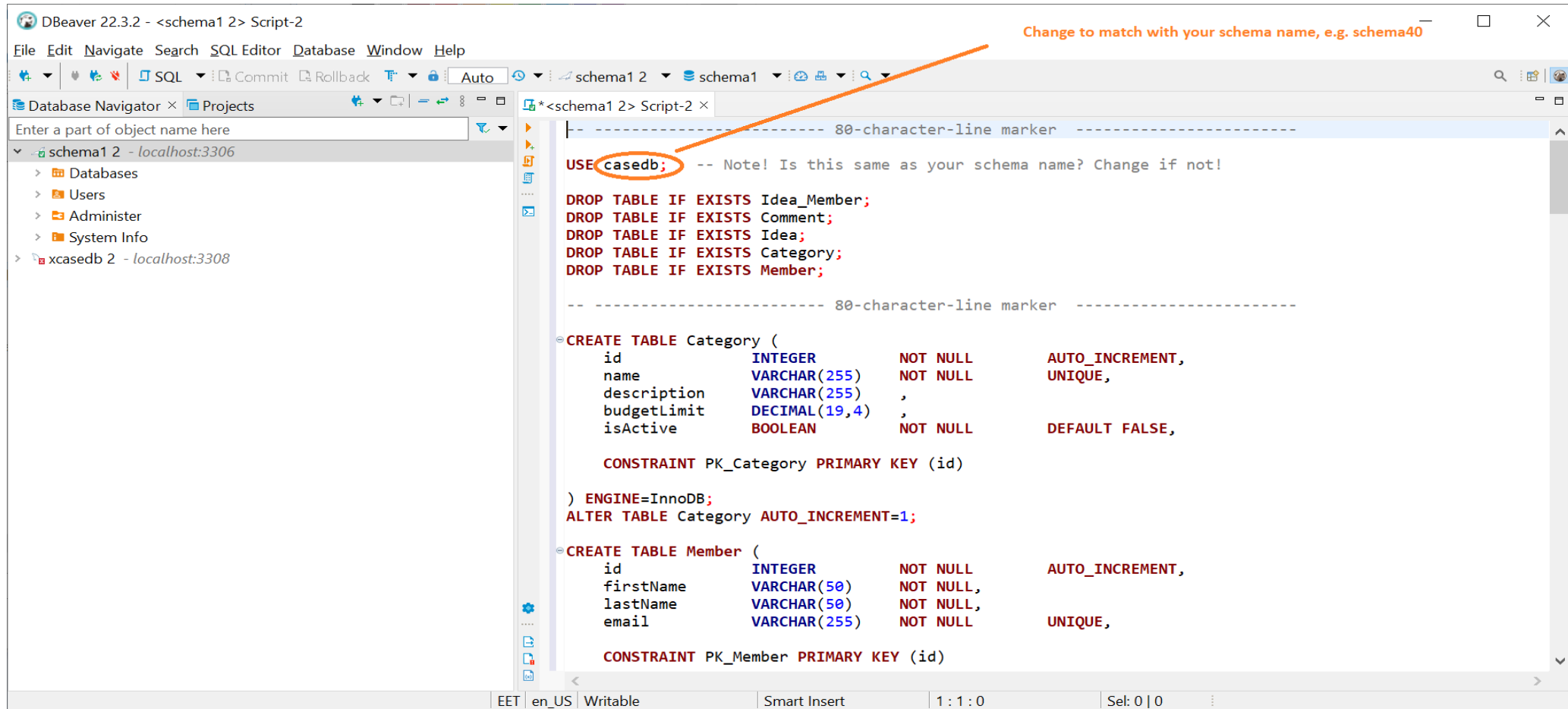
3.2.2 Use DBeaver to run SQL to create the tables and populate with test data



3.2.3 Use DBeaver to run SQL to create the tables and populate with test data



3.2.4 Use DBeaver to run SQL to create the tables and populate with test data



The screenshot shows the DBeaver 22.3.2 interface with a SQL script editor open. The script is for creating two tables, 'Category' and 'Member', in a database named 'xcasdb'. The script includes comments and SQL commands for dropping existing tables, creating new tables with specific columns and constraints, and altering the 'Category' table to set an auto-increment. The 'Database Navigator' on the left shows the 'xcasdb 2' database selected. An orange arrow points to the 'xcasdb' name in the script, with a note above it: 'Change to match with your schema name, e.g. schema40'.

```
----- 80-character-line marker -----
USE xcasdb; -- Note! Is this same as your schema name? Change if not!

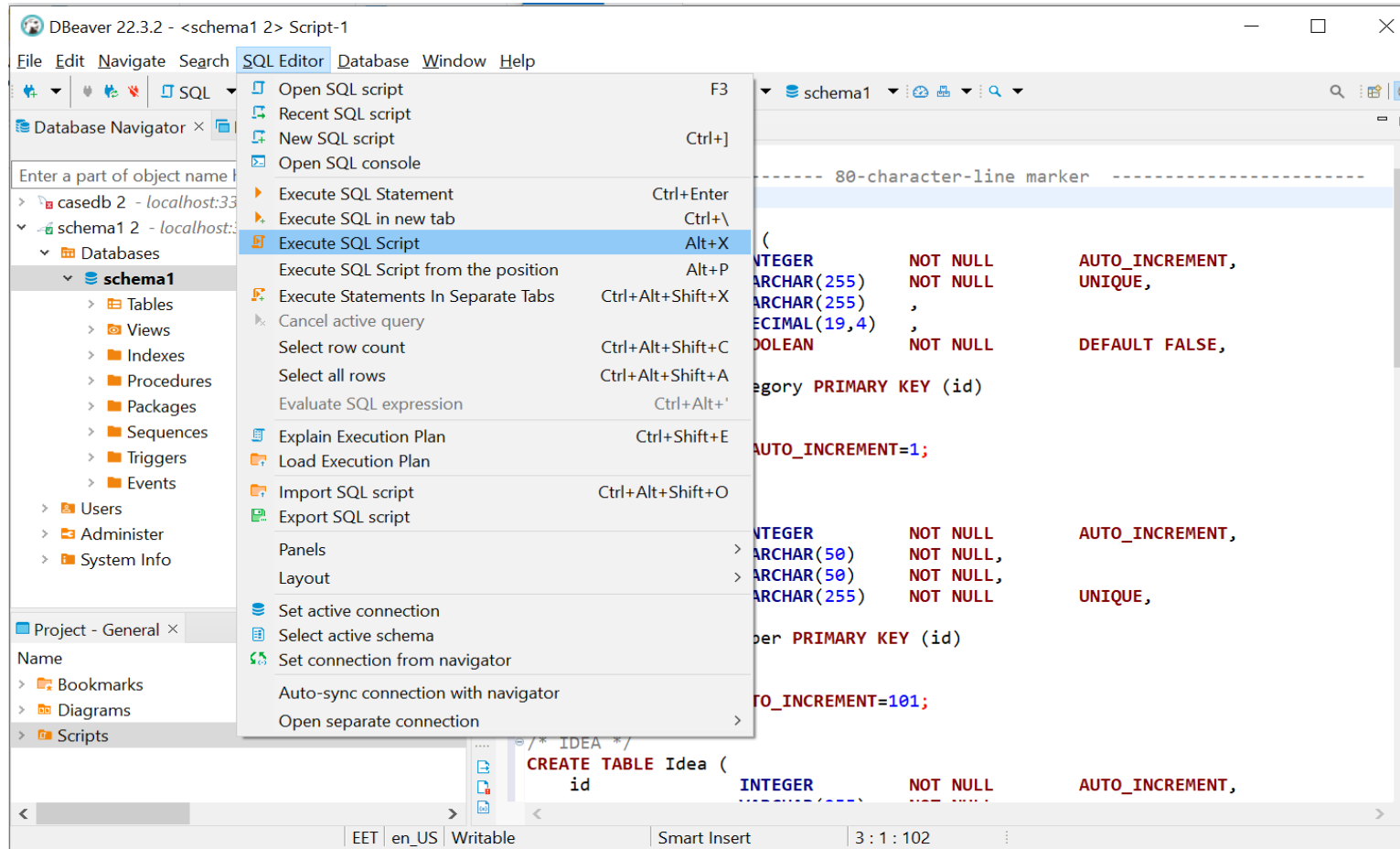
DROP TABLE IF EXISTS Idea_Member;
DROP TABLE IF EXISTS Comment;
DROP TABLE IF EXISTS Idea;
DROP TABLE IF EXISTS Category;
DROP TABLE IF EXISTS Member;

----- 80-character-line marker -----

CREATE TABLE Category (
  id          INTEGER      NOT NULL      AUTO_INCREMENT,
  name        VARCHAR(255) NOT NULL      UNIQUE,
  description  VARCHAR(255) ,
  budgetLimit DECIMAL(19,4) ,
  isActive    BOOLEAN      NOT NULL      DEFAULT FALSE,
  CONSTRAINT PK_Category PRIMARY KEY (id)
) ENGINE=InnoDB;
ALTER TABLE Category AUTO_INCREMENT=1;

CREATE TABLE Member (
  id          INTEGER      NOT NULL      AUTO_INCREMENT,
  firstName    VARCHAR(50) NOT NULL,
  lastName     VARCHAR(50) NOT NULL,
  email        VARCHAR(255) NOT NULL      UNIQUE,
  CONSTRAINT PK_Member PRIMARY KEY (id)
```

3.2.5 Use DBeaver to run SQL to create the tables and populate with test data



3.2.6 Use DBeaver to run SQL to create the tables and populate with test data – Success?

DBeaver 22.3.2 - <schema1 2> Script-3

File Edit Navigate Search SQL Editor Database Window Help

Database Navigator × Projects

Enter a part of object name here

- schema1 2 - localhost:3306
 - Databases
 - schema1
 - Users
 - Administer
 - System Info
 - xcasedb 2 - localhost:3308

SQL Editor: <schema1 2> Script-2

```
USE schema1; -- Note! Is this same as your schema name? Change if not!

DROP TABLE IF EXISTS Idea_Member;
DROP TABLE IF EXISTS Comment;
DROP TABLE IF EXISTS Idea;
DROP TABLE IF EXISTS Category;
DROP TABLE IF EXISTS Member;

----- 80-character-line marker -----

CREATE TABLE Category (
  id          INTEGER          NOT NULL          AUTO_INCREMENT,
  name        VARCHAR(255)     NOT NULL          UNIQUE,
  description  VARCHAR(255)
  ,
  budgetLimit DECIMAL(19,4)    ,
  isActive    BOOLEAN          NOT NULL          DEFAULT FALSE,

  CONSTRAINT PK_Category PRIMARY KEY (id)
) ENGINE=InnoDB;
ALTER TABLE Category AUTO_INCREMENT=1;

CREATE TABLE Member (
  id          INTEGER          NOT NULL          AUTO_INCREMENT,
  firstName   VARCHAR(50)      NOT NULL,
  lastName    VARCHAR(50)      NOT NULL,
  email       VARCHAR(255)     NOT NULL          UNIQUE,

  CONSTRAINT PK_Member PRIMARY KEY (id)
) ENGINE=InnoDB;
```

Output ×

```
Name 'PK_Category' ignored for PRIMARY key.
Name 'PK_Member' ignored for PRIMARY key.
Name 'PK_Idea' ignored for PRIMARY key.
Name 'PK_Idea_Member' ignored for PRIMARY key.
Name 'PK_Comment' ignored for PRIMARY key.
```

SCHEMATA 1 | TABLE NAMES 1 (2) | COLUMNS 1 (3) | Category 1 (4) × | Statistics 1

SELECT * FROM Category

	id	name	description	budgetLimit	isActive	Value
1	1	Outdoors	Outdoor activities, e.g. cycling through the hills.	1,000	1	
2	2	Exercises	Physical, mental and spiritual exercises carried o	500	1	
3	3	Recreation	Recreational activities like movies, swimming, toi	2,000	0	
4	4	Educational t	Jaa jaa jaa	999	1	
5	5	Outdoor Trai	Training here	100	0	
6	6	Indoor traini	No Training here	100	1	
7	7	A2 Training	Training for new technologies	2,500	1	
8	8	No Budget	Category without a budget.	[NULL]	1	
9	9	No Descripti	[NULL]	100	1	
10	10	Dummy	[NULL]	[NULL]	0	