FOXTECH

INFORME DE MIGRACIÓN DE DATOS

Introducción

Configuración sistemas gestores de bases de datos

Evidencia de la migración de la base de datos en diferentes motores (SQLite, PostgreSQL, M…

Introducción

INTEGRANTES:

- Andrea Lenes

Introducción

A continuación, podrá encontrar un informe detallado sobre el proceso de migración de datos, las herramientas utilizadas, el paso a paso y las configuraciones de los SGBD.

Objetivo general

Documentar el proceso de migración de datos.

Objetivos específicos

- Delimitar las herramientas utilizadas para la migración.
- Guiar paso a paso la migración de datos.
- Describir la configuración de los SGBD.

Justificación

En ocasiones es necesario migrar datos, sin embargo, hay algunas cosas que cambian de un SGBD a otro, por ello es importante realizar un informe de migración que permita guiar a los interesados en realizar tal proceso. En ese sentido, se presenta el informe de migración de datos

Configuración sistemas gestores de bases de datos

En la carpeta del proyecto de django crearemos un archivo llamado db.py en el cual se almacenara la configuración de cada gestor de base de datos que vamos a utilizar, cada gestor estará representado como una variable constante con los siguientes campos:

- **Engine:** Motor y su respectivo driver para realizar la conexión con la base de datos.
- **Name:** Nombre de la base de datos.
- **User:** Usuario registrado en el gestor de la base de datos.
- **Password:** Contraseña del usuario registrado en el gestor de la base de datos.
- **Host:** Url o dirección IP en donde esta alojado la base de datos.
- **Port:** Puerto de la url o dirección IP en donde esta alojado la base de datos

> ⓘ SQLite no tendrá una configuración tan extensa debido a que este motor de base de datos no maneja usuarios ni vistas.

```
import os

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

SQLITE = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3')
    }
}

POSTGRESQL = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'Foxtech',
        'USER': 'postgres',
        'PASSWORD': '1234',
        'HOST': 'localhost',
        'PORT': '5432'
    }
}

MYSQL = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'Foxtech',
        'USER': 'root',
        'PASSWORD': 'admin',
        'HOST': 'localhost',
        'PORT': '3306'
    }
}
```

Evidencia de la migración de la base de datos en diferentes motores (SQLite, PostgreSQL, MySQL)

Una vez creado el archivo db.py en donde se almacenara la configuración de la conexión a los tres motores de bases de datos, al principio del archivo settings.py se procederá a importar la librería del sistema operativo y se importara el archivo db.py al cual le colocaremos como alias "db".

```
import os
import Foxtech.db as db
```

Migración a SQLite

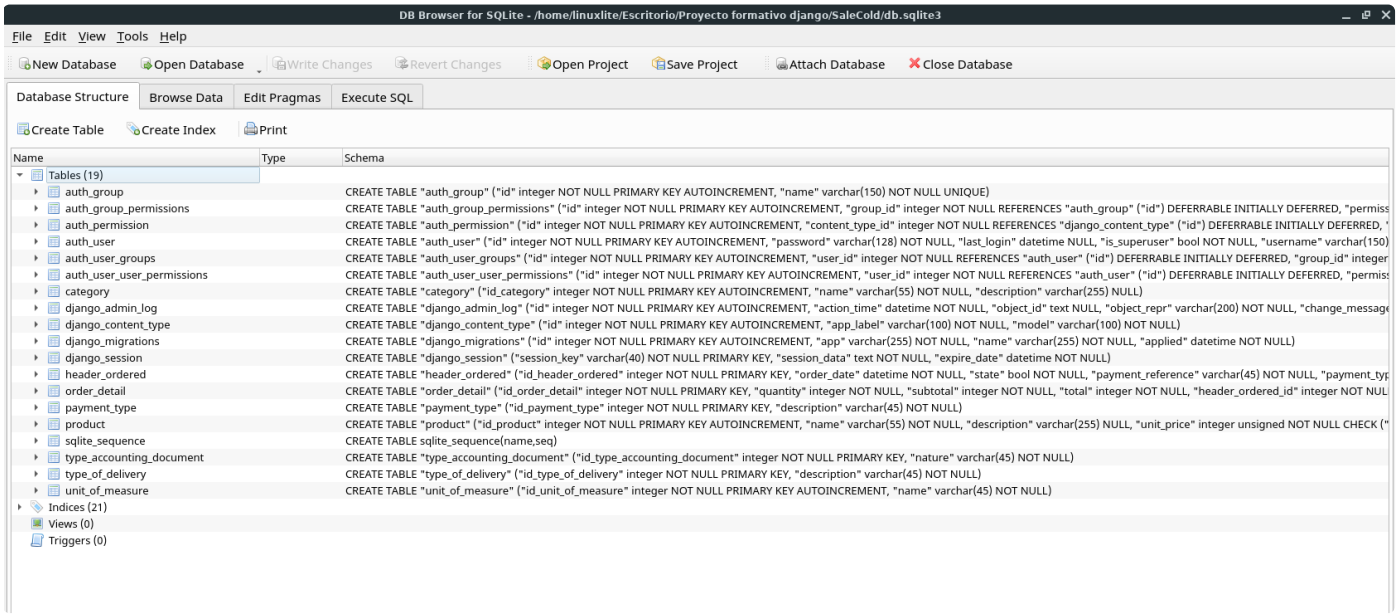
1. En el archivo settings.py en la variable constante "DATABASES", se procederá a llamar a la variable que contiene la configuración de la conexión con el motor de la base de datos en este caso SQLite.

```
DATABASES = db.SQLITE
```

2. Una vez especificado en el archivo settings.py a que motor vamos a realizar la conexión, en la terminal procederemos a ejecutar el siguiente comando para realizar la migración de la base de datos a SQLite.

```
python3 manage.py migrate
```

3. Abrimos la base de datos en el gestor de SQLite para confirmar que la migración se realizo de manera correcta.



Migración a PostgreSQL

Migración a PostgreSQL

1. En el archivo settings.py en la variable constante "DATABASES", se procederá a llamar a la variable que contiene la configuración de la conexión con el motor de la base de datos en este caso PostgreSQL.

```
DATABASES = db.POSTGRESQL
```

2. Una vez especificado en el archivo settings.py a que motor vamos a realizar la conexión, en la terminal procederemos a ejecutar el siguiente comando para realizar la migración de la base de datos a PostgreSQL.

```
python3 manage.py migrate
```

3. Abrimos la base de datos en el gestor pgAdmin4 para confirmar que la migración se realizo de manera correcta.

Migración a MySQL

1. En el archivo settings.py en la variable constante "DATABASES", se procederá a llamar a la variable que contiene la configuración de la conexión con el motor de la base de datos en este caso MySQL.

```
DATABASES = db.MYSQL
```

2. Una vez especificado en el archivo settings.py a que motor vamos a realizar la conexión, en la terminal procederemos a ejecutar el siguiente comando para realizar la migración de la base de datos a MySQL.

```
python3 manage.py migrate
```

3. Abrimos la base de datos en el gestor MySQL Workbench para confirmar que la migración se realizo de manera correcta.