

Groupe 48:
Sanaa El Marbough
Yassine Hassoune
Haytem Hilal
Mohamed Ali Talbe

Rapport du livrable 2- CSI2532

a. Le SGBD et les langages de programmation que vous avez utilisés dans votre implémentation de l'application:

Pour l'implémentation du SGBD, nous avons utilisé les technologies suivantes:

- PostgreSQL: Pour la conception de la base de donnée pour la chaîne hôtelière, po
- Python Framework Flask: Pour la partie Back-End de l'application Web
- HTML, CSS et JavaScript: pour la partie Front-End de l'application Web

b. Étapes spécifiques pour guider quelqu'un dans l'installation de vos applications.

L'application nécessite l'installation du langage de programmation python



```
Command Prompt
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ali>python --version
Python 3.10.2

C:\Users\Ali>
```

Après cela on a besoin du framework flask et la bibliothèque pscopg2, les deux appartenant à python, alors pour leurs installations on fait appel à la commande pip de python

Installation de pip:

Pour installer pip il faut entrer les commandes suivantes dans l'invite des commandes cmd:

```
>>curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
>>python get-pip.py
```

```
cmd Invite de commandes
Microsoft Windows [version 10.0.19045.4170]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\hp>curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 2574k  100 2574k    0     0 2150k      0  0:00:01  0:00:01 --:--:-- 2155k

C:\Users\hp>python get-pip.py
Collecting pip
  Using cached pip-24.0-py3-none-any.whl.metadata (3.6 kB)
Using cached pip-24.0-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 24.0
    Uninstalling pip-24.0:
      Successfully uninstalled pip-24.0
Successfully installed pip-24.0
```

Installation de flask:

Pour installer flask il faut entrer les commandes suivantes dans l'invite des commandes cmd:

```
>>python -m pip install flask
```

Installation de psycopg2:

Pour installer psycopg2 il faut entrer les commandes suivantes dans l'invite des commandes cmd:

```
>>python -m pip install psycopg2
```

```
Invite de commandes

C:\Users\hp>python -m pip install flask
Requirement already satisfied: flask in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (3.0.2)
Requirement already satisfied: Werkzeug>=3.0.0 in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (from flask) (3.0.2)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (from flask) (3.1.3)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (from flask) (2.1.2)
Requirement already satisfied: click>=8.1.3 in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (from flask) (8.1.7)
Requirement already satisfied: blinker>=1.6.2 in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (from flask) (1.7.0)
Requirement already satisfied: importlib-metadata>=3.6.0 in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (from flask) (7.1.0)
Requirement already satisfied: colorama in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (from click>=8.1.3->flask) (0.4.6)
Requirement already satisfied: zipp>=0.5 in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (from importlib-metadata>=3.6.0->flask) (3.18.1)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (from Jinja2>=3.1.2->flask) (2.1.5)

C:\Users\hp>python -m pip install psycopg2
Requirement already satisfied: psycopg2 in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (2.9.9)
```

Nous avons aussi de PostgreSQL, pour cela, on peut le télécharger et l'installer depuis ce lien:

<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

c. Une liste avec les DDL qui créent votre base de données.

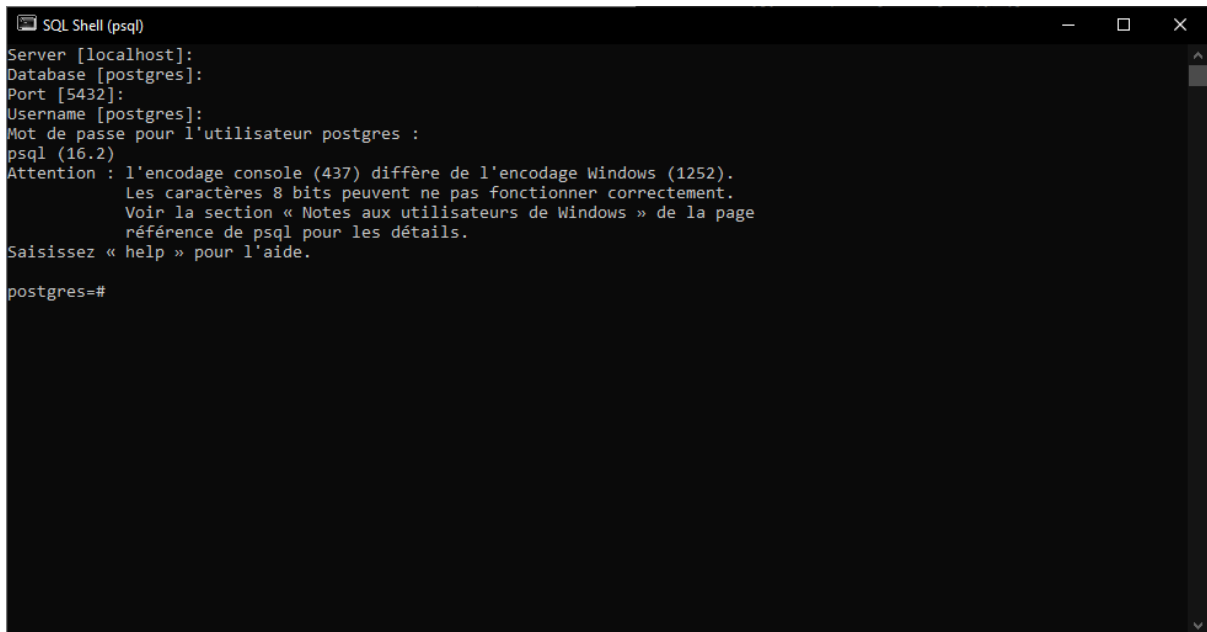
Nous avons écrit notre programme de façon à qu'il écrit et réécrit et repopule toute la base de données à chaque fois que nous exécutons l'application dans le fichier '___init__.py', du coup il n'est pas nécessaire de recréer la base données manuellement il suffit d'exécuter l'application.

Mais avant cela, il faut que les paramètres de postgres correspondent à ceux que nous avons utilisé:

```
conn = psycopg2.connect(
    host = "localhost",
    database = "flask_db",
    user = "postgres",
    port = "5432",
    password = "berserk")
```

Pour exécuter correctement notre application il est nécessaire d'avoir deux choses.

Premierement, Il faut utiliser le même mot de passe que nous, pour cela il faut aller dans le SQL shell (il suffit de taper psql dans la barre de recherche), puis entrer vos paramètres postgres normalement:



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Mot de passe pour l'utilisateur postgres :
psql (16.2)
Attention : l'encodage console (437) diffère de l'encodage Windows (1252).
            Les caractères 8 bits peuvent ne pas fonctionner correctement.
            Voir la section « Notes aux utilisateurs de Windows » de la page
            référence de psql pour les détails.
Saisissez « help » pour l'aide.

postgres=#
```

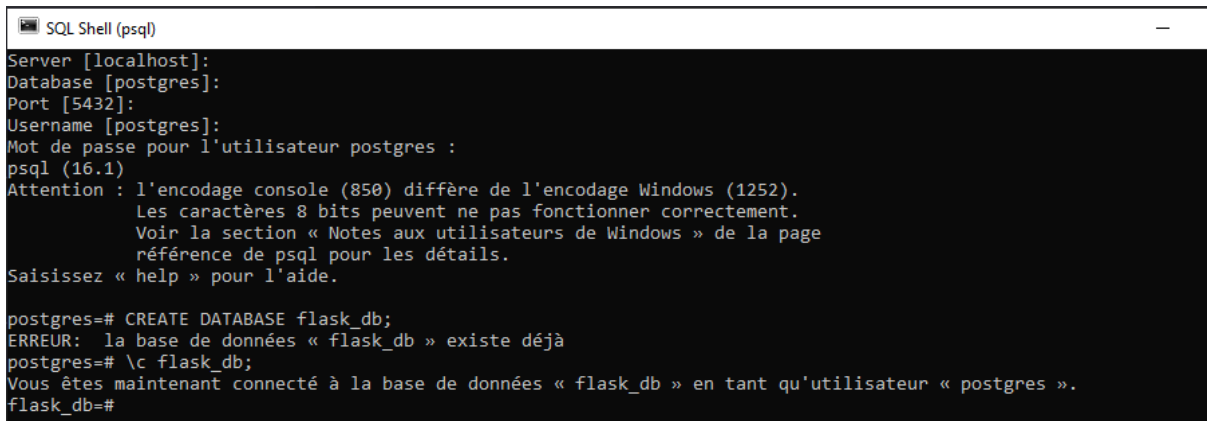
Puis utilisez cette commande dans le SQL shell:

```
>>ALTER USER user_name WITH PASSWORD 'berserk';
```

Dans un second lieu, avoir une base de données nommée flask db:

Pour cela il faut aller dans le shell a nouveau et entrer la commande suivante:

```
>>CREATE DATABASE flask_db
```



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Mot de passe pour l'utilisateur postgres :
psql (16.1)
Attention : l'encodage console (850) diffère de l'encodage Windows (1252).
            Les caractères 8 bits peuvent ne pas fonctionner correctement.
            Voir la section « Notes aux utilisateurs de Windows » de la page
            référence de psql pour les détails.
Saisissez « help » pour l'aide.

postgres=# CREATE DATABASE flask_db;
ERREUR: la base de données « flask_db » existe déjà
postgres=# \c flask_db;
Vous êtes maintenant connecté à la base de données « flask_db » en tant qu'utilisateur « postgres ».
flask_db=#
```

Il est aussi possible de simplement entrer vos paramètres postgresSQL dans notre code: ligne 19 du fichier '___init__.py' et la ligne 260 du fichier views.py.

Une fois les paramètres de postgre définis il suffit d'exécuter le fichier main.py soit en utilisant l'invite des commandes cmd avec la commande:

```
>> python main.py
```

ou en l'exécutant depuis le IDE

L'application devrait fonctionner correctement après tout cela

Une dernière remarque est que le programme est écrit de façon à ce que la base de données soit effacée et repeuplée à chaque fois ou on exécute l'application:

```
29 cur.execute('DROP TABLE IF EXISTS chaine CASCADE;')
30 cur.execute('DROP TABLE IF EXISTS hotel CASCADE;')
31 cur.execute('DROP TABLE IF EXISTS chambre;')
32 cur.execute('DROP TABLE IF EXISTS client;')
33 cur.execute('DROP TABLE IF EXISTS reservation;')
34 cur.execute('DROP TABLE IF EXISTS location;')
35 cur.execute('DROP TABLE IF EXISTS employe;')
36 cur.execute('DROP TABLE IF EXISTS paiement;')
37 cur.execute('DROP TABLE IF EXISTS archive;')
38
```

Pour enlever cela il suffit de comment out la fonction create_database() ligne 19 du fichier ' __init__.py'

On veut aussi porter a votre attention qu'il y a une petite erreur dans la page d'accueil ou les liens de la barre de navigation ne sont pas cliquables, il s'agit juste d'un bug il suffit de dezoomer un peu la page pour qu'ils fonctionnent.

Voici comme même le DDL dans tous les cas:

```
CREATE TABLE chaine (
    id serial PRIMARY KEY,
    nom varchar(150) NOT NULL,
    adresse varchar(50) NOT NULL,
    hotel_num integer NOT NULL,
    email varchar(150) NOT NULL,
    phone varchar(20) NOT NULL
);
```

```
CREATE TABLE hotel (
    id serial PRIMARY KEY,
    rang integer NOT NULL,
    adresse varchar(50) NOT NULL,
    chambre_num integer NOT NULL,
    email varchar(150) NOT NULL,
    phone varchar(20) NOT NULL,
    chaine_ID integer,
    reservation_count integer,
    CONSTRAINT fk_chaine
    FOREIGN KEY (chaine_ID)
    REFERENCES chaine (id)
    ON DELETE CASCADE
);
```

```
CREATE TABLE chambre (
```

```

id serial PRIMARY KEY,
numero integer NOT NULL,
hotel_ID integer NOT NULL,
prix integer NOT NULL,
capacite integer NOT NULL,
CONSTRAINT fk_hotel
FOREIGN KEY (hotel_ID)
REFERENCES hotel (id)
ON DELETE CASCADE
);

CREATE TABLE client (
id serial PRIMARY KEY,
email varchar(150) NOT NULL,
password varchar(150) NOT NULL,
NAS integer NOT NULL,
nom varchar(150) NOT NULL,
adresse varchar(150) NOT NULL,
card_number varchar(150) NOT NULL,
date_added date DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE reservation (
id serial PRIMARY KEY,
client_ID integer NOT NULL,
chambre_ID integer NOT NULL,
checkin DATE NOT NULL,
duration integer NOT NULL
);

CREATE TABLE employe (
id serial PRIMARY KEY,
email varchar(150) NOT NULL,
password varchar(150) NOT NULL,
NAS integer NOT NULL,
hotel_ID integer NOT NULL,
nom varchar(150) NOT NULL,
adresse varchar(150) NOT NULL,
role varchar(150) NOT NULL
);

CREATE TABLE location (
id serial PRIMARY KEY,
client_ID integer NOT NULL,
chambre_ID integer NOT NULL,
checkin DATE NOT NULL,
duration integer NOT NULL
);

```

```
CREATE TABLE paiement (
    id serial PRIMARY KEY,
    location_ID integer NOT NULL,
    card_number varchar(150) NOT NULL
);
```

```
CREATE TABLE archive (
    id serial PRIMARY KEY,
    type varchar(150) NOT NULL,
    correspond_ID integer NOT NULL,
    client_ID integer NOT NULL,
    chambre_ID integer NOT NULL,
    checkin DATE NOT NULL,
    duration integer NOT NULL
);
```

```
CREATE TABLE reservation_log (
    reservation_id integer PRIMARY KEY
);
```

d. Index utilises:

```
cur.execute('CREATE INDEX idx_chaine_hoteliere ON hotel(chaine_ID);')
cur.execute('CREATE INDEX I1_NumChambre On chambre(numero);')
cur.execute('CREATE INDEX I3_Chambre On chambre(hotel_ID);')
```

es index sont utilisés pour accélérer les requêtes en permettant un accès rapide aux données.

e. Triggers utilises:

```
cur.execute('CREATE OR REPLACE FUNCTION log_new_reservation() '
            'RETURNS TRIGGER AS $$ '
            'BEGIN '
            '    INSERT INTO reservation_log(reservation_id) '
            '    VALUES (NEW.id); '
            '    RETURN NEW; '
            'END; '
            '$$ LANGUAGE plpgsql;')

cur.execute('CREATE TRIGGER trigger_log_new_reservation '
            'AFTER INSERT ON reservation '
            'FOR EACH ROW '
            'EXECUTE FUNCTION log_new_reservation();')
```

```

cur.execute('CREATE OR REPLACE FUNCTION
update_hotel_reservation_count() '
           'RETURNS TRIGGER AS $$'
           'BEGIN '
           'UPDATE hotel SET reservation_count =
reservation_count + 1 '
           'WHERE id = (SELECT hotel_ID FROM chambre WHERE id =
NEW.chambre_ID);'
           'RETURN NEW;'
           'END;'
           '$$ LANGUAGE plpgsql;'

'CREATE TRIGGER trigger_update_hotel_reservation_count '
'AFTER INSERT ON reservation '
'FOR EACH ROW '
'EXECUTE FUNCTION update_hotel_reservation_count();')

```

f. Vues utilises:

```

cur.execute('CREATE VIEW chambres_disponibles_hotel AS '
           'SELECT c.id AS chambre_id, c.numero AS numero_chambre,
c.prix AS prix_chambre, c.capacite AS capacite_chambre '
           'FROM chambre c '
           'LEFT JOIN reservation r ON c.id = r.chambre_ID '
           'WHERE r.id IS NULL;')

cur.execute('CREATE VIEW chambres_disponibles_par_adresse AS '
           'SELECT h.adresse AS adresse_hotel, '
           'c.id AS chambre_id, '
           'c.numero AS numero_chambre, '
           'c.prix AS prix_chambre,'
           'c.capacite AS capacite_chambre '
           'FROM hotel h '
           'JOIN chambre c ON h.id = c.hotel_ID '
           'LEFT JOIN reservation r ON c.id = r.chambre_ID '
           'WHERE r.id IS NULL '
           'GROUP BY h.adresse, c.id;')

```