**Récupération d’info sur Django**

**Récupère toutes les musiques avec nom, format et la playlist affilié :**

SELECT music.path,music.id\_playlist FROM music

**Récupere toutes les musique d’une playlist**

SELECT music.path FROM music JOIN playlist

ON music.id\_playlist = playlist.id

WHERE playlist.id = 3

**Récupéré les informations de configuration de la playlist, dates de début et de fin, heure de débuts et de fin**

SELECT algorithme.begin\_month,algorithme.end\_month,algorithme.begin\_time,algorithme.end\_time FROM algorithme JOIN playlist

ON algorithme.id\_playlist = playlist.id

WHERE playlist.id = 3

**Récupère les code couleur choisie par palier de température**

SELECT led\_color.color, algorithme.min\_temp, algorithme.max\_temp FROM led\_color, algorithme

WHERE led\_color.id = algorithme.id\_led

**Envoi vers la BDD**

**Envoie d’une liste de musique avec nom, format, playlist**

**INSERT** **INTO** music(id,name,id\_playlist,path,format,artist)

**VALUES** (10,"musique 10",1,"musique 10.mp3", ‘.mp3’,’test’)

for music in liste:

cursor = con.cursor()

command = """INSERT INTO music(id,name,id\_playlist,path,format,artist)

VALUES """ + str(music)

**Envoi des informations d’une playlist : date et heure**

listeAlg = [(4,16,20,9,12),(5,12,16,9,12)]

listePl = [("test4",4),("test5",5)]

for algo in listeAlg:

cursor = con.cursor()

command = """INSERT INTO algorithme(id\_playlist,begin\_time,end\_time,begin\_month,end\_month)

VALUES """ +str(algo)

print(command)

cursor.execute(command)

for playlist in listePl:

cursor = con.cursor()

command = """INSERT INTO playlist(name,id)

VALUES """ +str(playlist)

cursor.execute(command)

**Envoie des informations de LED : couleur et palier de température**

listeLed = [("126,126,0",4),("0,126,126",5)]

listeAlgo = [(4,6,12),(5,12,18)]

for led in listeLed:

cursor = con.cursor()

command = """INSERT INTO led\_color(color,id)

VALUES """+str(led)

cursor.execute(command)

for algo in listeAlgo:

cursor = con.cursor()

command = """INSERT INTO algorithme(id\_led,min\_temp,max\_temp)

VALUES """+str(algo)

cursor.execute(command)

bite