

TP SQL : Corrigé

1. (a) `SELECT` Nom `FROM` Aeroports
`WHERE` Continent='Afrique'
`ORDER BY` Nom
- (b) `SELECT` Nom `FROM` Aeroports
`WHERE` Nom `LIKE` '%a'
- (c) `SELECT` a.Nom, b.Nom `FROM` Aeroports a
`JOIN` Aeroports b `ON` b.Longitude - a.Longitude >= 0
 AND b.Longitude - a.Longitude <= 5
 AND a.Nom != b.Nom
- (d) `SELECT` a.Nom, b.Nom `FROM` Vols
`JOIN` Aeroports a `ON` Ville1=a.Code
`JOIN` Aeroports b `ON` Ville2=b.Code
`WHERE` b.Longitude - a.Longitude >= 0
 AND b.Longitude - a.Longitude <= 20
- (e) `SELECT DISTINCT` Denomination `FROM` Vols
`JOIN` Aeroports a `ON` Ville1=a.Code
`JOIN` Aeroports b `ON` Ville2=b.Code
`JOIN` Compagnies c `ON` CodeCompagnie=c.Code
`WHERE` a.Nom = 'Hong Kong' OR b.Nom = 'Hong Kong'
- (f) `SELECT` Nom `FROM` Aeroports
`WHERE` Continent='Afrique'
 AND Longitude > (`SELECT` MAX(Longitude) `FROM` Aeroports
 `WHERE` Continent='Europe')
- (g) `SELECT` Continent `FROM` Aeroports
`WHERE` LENGTH(Nom) > 5
`GROUP BY` Continent
`HAVING` AVG(Longitude) > 0 AND AVG(Latitude) > 0

3. et 4.

```
import matplotlib.image as mpimg #module images
import matplotlib.pyplot as plt
plt.close('all')
planisphere=mpimg.imread("Peters.jpg") #ouverture d'une image
plt.imshow(planisphere) #affichage de l'image

import numpy as np
def coordonnees(longitude, latitude):
    """ Renvoie les coordonnées d'un point sur le planisphere
    à partir de ses coordonnées GPS en degrés décimaux """
    x=632/2*longitude/180+632/2
    y=-399/2*np.sin(latitude*np.pi/180)+399/2
    return (x,y)

v=resultat_requete("SELECT Nom, Longitude, Latitude \
                   FROM Aeroports")

for ville in v :
    nom = ville[0]
    longitude,latitude = float(ville[1]), float(ville[2])
    x,y= coordonnees(longitude, latitude)
    plt.plot(x,y,'o',color='black')
    plt.text(x,y,nom, color='white', size='smaller')
plt.show()

5. plt.figure(3)
plt.imshow(planisphere) # affichage de l'image

B=resultat_requete("SELECT A.Nom,Longitude,Latitude,Couleur\
                   FROM Aeroports AS A \
                   JOIN Continents AS C ON C.Nom=A.Continent")

for aeroport in B:
    x,y=coordonnees(float(aeroport[1]),float(aeroport[2]))
    plt.plot(x,y,'o',color='black')
    plt.text(x,y,aeroport[0],color=aeroport[3],size='smaller')
    # modifs ici
plt.show()
```

```
6. C=resultat_requete("SELECT A1.Longitude,A1.Latitude,\
    A2.Longitude,A2.Latitude \
    FROM Aeroports AS A1 \
    JOIN Vols AS V ON V.Ville1=A1.Code\
    JOIN Aeroports AS A2 ON V.Ville2=A2.Code")
```

```
plt.figure(4)
plt.imshow(planisphere) # affichage de l'image

for aeroport in B:
    x,y=coordonnees(float(aeroport[1]),float(aeroport[2]))
    plt.plot(x,y,'o',color='black')
    plt.text(x,y,aeroport[0],color=aeroport[3],size='smaller')
    # modifs ici

for vol in C:
    x1,y1=coordonnees(float(vol[0]),float(vol[1]))
    x2,y2=coordonnees(float(vol[2]),float(vol[3]))
    plt.plot([x1,x2],[y1,y2],linestyle="--",color="white")
plt.show()
```

```
7. def geodesique(v1,v2):
    """ Calcule la trajectoire géodésique entre deux points
    de latitude et longitude donnée v1=[lat,long]
    Retourne une liste des coordonnées des points
    de la trajectoire"""
    lambda1=v1[0]*np.pi/180
    phi1=v1[1]*np.pi/180
    lambda2=v2[0]*np.pi/180
    phi2=v2[1]*np.pi/180
    coordv1=np.array([np.cos(phi1)*np.cos(lambda1),
        np.sin(phi1)*np.cos(lambda1),
        np.sin(lambda1)])
    coordv2=np.array([np.cos(phi2)*np.cos(lambda2),
        np.sin(phi2)*np.cos(lambda2),
        np.sin(lambda2)])

    trajectoire=[]
    nbpoints=100
    for k in range(nbpoints+1):
        t1=float(k)/nbpoints
        a=1
        b=abs(2*t1*np.vdot(coordv1,coordv2))
        c=t1**2-1
        delta=(b)**2-4*a*c
        t2=(-b+delta**0.5)/(2*a)
```

```
    coordm=t1*coordv1+t2*coordv2
    lambdam=np.arcsin(coordm[2])
    if coordm[1]>=0:
        phim=np.arccos(coordm[0]/np.cos(lambdam))
    else:
        phim=-np.arccos(coordm[0]/np.cos(lambdam))
    trajectoire.append(coordonnees(
        lambdam*180/np.pi,phim*180/np.pi))

    return trajectoire
```

```
plt.figure(5)
plt.imshow(planisphere) # affichage de l'image

for aeroport in B:
    x,y=coordonnees(float(aeroport[1]),float(aeroport[2]))
    plt.plot(x,y,'o',color='black')
    plt.text(x,y,aeroport[0],color=aeroport[3],size='smaller')
    # modifs ici

for vol in C:
    trajectoire=geodesique([float(vol[0]),float(vol[1])],
        [float(vol[2]),float(vol[3])])

    Abs=[]
    Ord=[]
    for point in trajectoire:
        Abs.append(point[0])
        Ord.append(point[1])
    plt.plot(Abs,Ord)
plt.show()
```

```
8. D=resultat_requete("SELECT A1.Longitude,A1.Latitude,\
    A2.Longitude,A2.Latitude \
    FROM Aeroports AS A1 \
    JOIN Vols AS V ON V.Ville1=A1.Code \
    JOIN Aeroports AS A2 ON V.Ville2=A2.Code")
```