In [1]:

```python
import pandas as pd
import numpy as np
from sqlalchemy import create_engine
```

In [2]:

```python
engine = create_engine("mysql+pymysql://{user}:{pw}@{host}/{db}".format(host="localhost", db='hr', user="root", pw="Keetzpk@18"))
```

In [3]:

```python
import mysql.connector as server1
db1=server1.connect(host='localhost',user='root',password='Keetzpk@18',database='hr')
mycursor=db1.cursor()
```

Task1: To right a query to extract country names and regions

In [4]:

```python
Task1='Select country_name,region_name from countries inner join regions on countries.region_id=regions.region_id'
mycursor.execute(Task1)
mycursor.fetchall()
```

Out[4]:

```
[('Belgium', 'Europe'),
 ('Switzerland', 'Europe'),
 ('Germany', 'Europe'),
 ('Denmark', 'Europe'),
 ('France', 'Europe'),
 ('Italy', 'Europe'),
 ('Netherlands', 'Europe'),
 ('United Kingdom', 'Europe'),
 ('Argentina', 'Americas'),
 ('Brazil', 'Americas'),
 ('Canada', 'Americas'),
 ('Mexico', 'Americas'),
 ('United States of America', 'Americas'),
 ('Australia', 'Asia'),
 ('China', 'Asia'),
 ('HongKong', 'Asia'),
 ('India', 'Asia'),
 ('Japan', 'Asia'),
 ('Singapore', 'Asia'),
 ('Egypt', 'Middle East and Africa'),
 ('Israel', 'Middle East and Africa'),
 ('Kuwait', 'Middle East and Africa'),
 ('Nigeria', 'Middle East and Africa'),
 ('Zambia', 'Middle East and Africa'),
 ('Zimbabwe', 'Middle East and Africa')]
```

Task2: To find out countrywise count of employees and push new table into server

In [5]:

```python
data=pd.read_sql_query("""Select * from employees inner join departments inner join locations
inner join countries on employees.department_id=departments.department_id and locations.location_id=departments.location_id and countries.country_id=locations.country_id""",db1)
data
```

```
K:\Games\PROGRAMS\ANACONDA\lib\site-packages\pandas\io\sql.py:762: UserWarning: pandas only support SQLAlchemy connectable(engine/connection) ordatabase string URI or sqlite3 DBAPI2 co
nnectionother DBAPI2 objects are not tested, please consider using SQLAlchemy
  warnings.warn(
```

Out[5]:

	employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	...	location_id	location_id	street_address	postal_code	city	state_province	country_id	co
0	200	Jennifer	Whalen	JWHALEN	515.123.4444	1987-09-17	AD_ASST	4400.0	NaN	101.0	...	1700	1700	2004 Charade Rd	98199	Seattle	Washington	US	
1	201	Michael	Hartstein	MHARTSTE	515.123.5555	1996-02-17	MK_MAN	13000.0	NaN	100.0	...	1800	1800	147 Spadina Ave	M5V 2L7	Toronto	Ontario	CA	
2	202	Pat	Fay	PFAY	603.123.6666	1997-08-17	MK_REP	6000.0	NaN	201.0	...	1800	1800	147 Spadina Ave	M5V 2L7	Toronto	Ontario	CA	
3	114	Den	Raphaely	DRAPHEAL	515.127.4561	1994-12-07	PU_MAN	11000.0	NaN	100.0	...	1700	1700	2004 Charade Rd	98199	Seattle	Washington	US	
4	115	Alexander	Khoo	AKHOO	515.127.4562	1995-05-18	PU_CLERK	3100.0	NaN	114.0	...	1700	1700	2004 Charade Rd	98199	Seattle	Washington	US	

In [6]:

```python
data=data.loc[:,["employee_id","country_name"]].groupby("country_name").count()
data.insert(loc=0,column="country_name",value=data.index)
data.reset_index(drop=True)
```

Out[6]:

	country_name	employee_id
0	Canada	2
1	Germany	1
2	United Kingdom	35
3	United States of America	68

In [7]:

```python
data.to_sql( name='Employee_country',con=engine,if_exists='append',index=False)
```

The above line could not be run in my sytem due to some technical difficulties

Task3: To visually represent country wise employee count

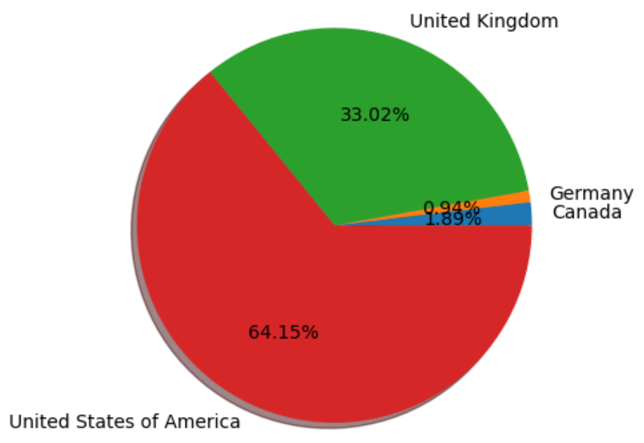
In [8]:

```python
import matplotlib.pyplot as plt
```

In [9]:

```python
plt.pie(data["employee_id"],labels=data.country_name,autopct="%.2f%%",shadow=True);
print("From below chart we are able to see that most employees come from United states of america")
```

```
From below chart we are able to see that most employees come from United states of america
```


United Kingdom
Germany
Canada
United States of America