

```
In [ ]: import pyodbc

In [ ]: def return_connection_sql():
    server = 'DESKTOP-0FTNP4D\\SQLEXPRESS'
    database = 'My_database'
    string_conexao = 'Driver={SQL Server Native Client 11.0};Server='+server+';Database='+database+';Trusted_Connection=yes;'
    conexao = pyodbc.connect(string_conexao)
    return conexao.cursor()

In [ ]: response = return_connection_sql()
try:
    print(f'Conectado ao banco de dados com sucesso:{response}')
except Exception as e:
    print(e)

Conectado ao banco de dados com sucesso:<pyodbc.Cursor object at 0x0000233224D08B0>

In [ ]: for driver in pyodbc.drivers():
    print(driver)

SQL Server
Microsoft Access Driver (*.mdb, *.accdb)
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)
Microsoft Access Text Driver (*.txt, *.csv)
SQL Server Native Client 11.0
SQL Server Native Client RDA 11.0
ODBC Driver 17 for SQL Server
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx)
MySQL ODBC 8.0 ANSI Driver
MySQL ODBC 8.0 Unicode Driver

In [ ]: def execute_query(query):
    cursor = return_connection_sql()
    return cursor.execute(query)

query_info = "SELECT * FROM HumanResources.Department"
table = execute_query(query_info)

In [ ]: for row in table:
    print(row)

(2, 'Engineering', 'Research and Development', datetime.datetime(2008, 4, 30, 0, 0))
(2, 'Tool Design', 'Research and Development', datetime.datetime(2008, 4, 30, 0, 0))
(3, 'Sales', 'Sales and Marketing', datetime.datetime(2008, 4, 30, 0, 0))
(4, 'Marketing', 'Sales and Marketing', datetime.datetime(2008, 4, 30, 0, 0))
(5, 'Purchasing', 'Inventory Management', datetime.datetime(2008, 4, 30, 0, 0))
(6, 'Research and Development', 'Research and Development', datetime.datetime(2008, 4, 30, 0, 0))
(7, 'Production', 'Manufacturing', datetime.datetime(2008, 4, 30, 0, 0))
(8, 'Production Control', 'Manufacturing', datetime.datetime(2008, 4, 30, 0, 0))
(9, 'Human Resources', 'Executive General and Administration', datetime.datetime(2008, 4, 30, 0, 0))
(10, 'Finance', 'Executive General and Administration', datetime.datetime(2008, 4, 30, 0, 0))
(11, 'Information Services', 'Executive General and Administration', datetime.datetime(2008, 4, 30, 0, 0))
(12, 'Document Control', 'Quality Assurance', datetime.datetime(2008, 4, 30, 0, 0))
(13, 'Quality Assurance', 'Quality Assurance', datetime.datetime(2008, 4, 30, 0, 0))
(14, 'Facilities and Maintenance', 'Executive General and Administration', datetime.datetime(2008, 4, 30, 0, 0))
(15, 'Shipping and Receiving', 'Inventory Management', datetime.datetime(2008, 4, 30, 0, 0))
(16, 'Executive', 'Executive General and Administration', datetime.datetime(2008, 4, 30, 0, 0))

In [ ]: query_info2 = "SELECT DISTINCT Name FROM Person.CountryRegion"
table = execute_query(query_info2)

for row in table:
    print(row)

('Afghanistan', )
('Albania', )
('Algeria', )
('American Samoa', )
('Andorra', )
('Angola', )
('Anguilla', )
('Antarctica', )
('Antigua and Barbuda', )
('Argentina', )
('Armenia', )
('Aruba', )
('Australia', )
('Austria', )
('Azerbaijan', )
('Bahamas, The', )
('Bahrain', )
('Bangladesh', )
('Barbados', )
('Belarus', )
('Belgium', )
('Belize', )
('Benin', )
('Bermuda', )
('Bhutan', )
('Bolivia', )
('Bosnia and Herzegovina', )
('Botswana', )
('Bouvet Island', )
('Brazil', )
('British Indian Ocean Territory', )
('Brunei', )
('Bulgaria', )
('Burkina Faso', )
('Burundi', )
('Cambodia', )
('Cameroon', )
('Canada', )
('Cape Verde', )
('Cayman Islands', )
('Central African Republic', )
('Chad', )
('Chile', )
('China', )
('Christmas Island', )
('Cocos (Keeling) Islands', )
('Colombia', )
('Comoros', )
('Congo', )
('Congo (DRC)', )
('Cook Islands', )
('Costa Rica', )
('Côte d'Ivoire', )
('Croatia', )
('Cuba', )
('Cyprus', )
('Czech Republic', )
('Denmark', )
('Djibouti', )
('Dominica', )
('Dominican Republic', )
('Ecuador', )
('Egypt', )
('El Salvador', )
('Equatorial Guinea', )
('Eritrea', )
('Estonia', )
('Ethiopia', )
('Falkland Islands (Islas Malvinas)', )
('Faroe Islands', )
('Fiji Islands', )
('Finland', )
('France', )
('French Guiana', )
('French Polynesia', )
('French Southern and Antarctic Lands', )
('Gabon', )
('Gambia, The', )
('Georgia', )
('Germany', )
('Ghana', )
('Gibraltar', )
('Greece', )
('Greenland', )
('Grenada', )
('Guadeloupe', )
('Guam', )
('Guatemala', )
('Guinea', )
('Guinea-Bissau', )
('Guyana', )
('Haiti', )
('Heard Island and McDonald Islands', )
('Honduras', )
('Hong Kong SAR', )
('Hungary', )
('Iceland', )
('India', )
('Indonesia', )
('Iran', )
('Iraq', )
('Ireland', )
('Israel', )
('Italy', )
('Jamaica', )
('Japan', )
('Jordan', )
('Kazakhstan', )
('Kenya', )
('Kiribati', )
('Korea', )
('Kuwait', )
('Kyrgyzstan', )
('Laos', )
('Latvia', )
('Lebanon', )
('Lesotho', )
('Liberia', )
('Libya', )
('Liechtenstein', )
('Lithuania', )
('Luxembourg', )
('Macao SAR', )
('Macedonia, Former Yugoslav Republic of', )
('Madagascar', )
('Malawi', )
('Malaysia', )
('Maldives', )
('Mali', )
('Malta', )
('Marshall Islands', )
('Martinique', )
('Mauritania', )
('Mauritius', )
('Mayotte', )
('Mexico', )
('Micronesia', )
('Moldova', )
('Monaco', )
('Mongolia', )
('Montserrat', )
('Morocco', )
('Mozambique', )
('Myanmar', )
('Namibia', )
('Nauru', )
('Nepal', )
('Netherlands', )
('Netherlands Antilles', )
('New Caledonia', )
('New Zealand', )
('Nicaragua', )
('Niger', )
('Nigeria', )
('Niue', )
('Norfolk Island', )
('North Korea', )
('Northern Mariana Islands', )
('Norway', )
('Oman', )
('Pakistan', )
('Palau', )
('Palestinian Authority', )
('Panama', )
('Papua New Guinea', )
('Paraguay', )
('Peru', )
('Philippines', )
('Pitcairn Islands', )
('Poland', )
('Portugal', )
('Puerto Rico', )
('Qatar', )
('Réunion', )
('Romania', )
('Russia', )
('Rwanda', )
('Saint Helena', )
('Saint Kitts and Nevis', )
('Saint Lucia', )
('Saint Pierre and Miquelon', )
('Saint Vincent and the Grenadine', )
('Samoa', )
('San Marino', )
('São Tomé and Príncipe', )
('Saudi Arabia', )
('Senegal', )
('Serbia and Montenegro', )
('Seychelles', )
('Sierra Leone', )
('Singapore', )
('Slovakia', )
('Slovenia', )
('Solomon Islands', )
('Somalia', )
('South Africa', )
('South Georgia and the South Sandwich Islands', )
('Spain', )
('Sri Lanka', )
('Sudan', )
('Suriname', )
('Svalbard and Jan Mayen', )
('Swaziland', )
('Sweden', )
('Switzerland', )
('Syria', )
('Taiwan', )
('Tajikistan', )
('Tanzania', )
('Thailand', )
('Timor-Leste', )
('Togo', )
('Tokelau', )
('Tonga', )
('Trinidad and Tobago', )
('Tunisia', )
('Turkey', )
('Turkmenistan', )
('Turks and Caicos Islands', )
('Tuvalu', )
('U.S. Minor Outlying Islands', )
('Uganda', )
('Ukraine', )
('United Arab Emirates', )
('United Kingdom', )
('United States', )
('Uruguay', )
('Uzbekistan', )
('Vanuatu', )
('Vatican City', )
('Venezuela', )
('Vietnam', )
('Virgin Islands, British', )
('Virgin Islands, U.S.', )
('Wallis and Futuna', )
('Yemen', )
('Zambia', )
('Zimbabwe', )
```