

```
import pyodbc
```

```
cnxn = pyodbc.connect('DRIVER={ODBC Driver 17 for SQL Server};'
```

```
    'SERVER=< DESKTOP-C7ML>;'
```

```
    'DATABASE=D_Danismanlik;'
```

```
    'UID=<username >';'
```

```
    'PWD=<password>')
```

```
cursor = cnxn.cursor()
```

```
def main_menu():
```

```
    print("D-Danışmanlık Raporlama Sistemi")
```

```
    print("1. Firma Kaynaklarını Raporla")
```

```
    print("2. Belirli Türdeki Kaynakları Raporla")
```

```
    print("3. Belirli Tarih-Aralığında Kullanımdaki Kaynakları Raporla")
```

```
    print("4. Belirli Tarih-Aralığında Kullanılmayan Kaynakları Raporla")
```

```
    print("5. Firma Müşterilerini Raporla")
```

```
    print("6. Belirli Tarih-Aralığında Kaynak Kullanacak Müşterileri Raporla")
```

```
    print("0. Çıkış")
```

```
def report_all_resources():
```

```
    # Tüm kaynakları raporla
```

```
    cursor.execute("SELECT * FROM Kaynak")
```

```
    resources = cursor.fetchall()
```

```
    if resources:
```

```
        print("\nTüm Kaynaklar:")
```

```
        print("-" * 30)
```

```
        for resource in resources:
```

```
            print(f"Kaynak ID: {resource.KaynakID}")
```

```
            print(f"Tip: {resource.Tip}")
```

```
            print(f"Özellikler: {resource.Ozellikler}")
```

```
            print("-" * 30)
```

```
    else:
```

```
        print("\nKaynak bulunamadı.")
```

```
def report_specific_resource_type():
```

```
    # Belirli türdeki kaynakları raporla
```

```
    resource_type = input("Raporlanacak kaynak tipini girin: ")
```

```
    cursor.execute("SELECT * FROM Kaynak WHERE Tip = ?", resource_type)
```

```
    resources = cursor.fetchall()
```

```
    if resources:
```

```
        print(f"\n{resource_type} Tipindeki Kaynaklar:")
```

```
        print("-" * 30)
```

```
        for resource in resources:
```

```
            print(f"Kaynak ID: {resource.KaynakID}")
```

```
            print(f"Özellikler: {resource.Ozellikler}")
```

```

        print("-" * 30)
    else:
        print(f"\n{resource_type} tipinde kaynak bulunamadı.")

import datetime

def report_resources_in_use():
    # Belirli tarih-aralığında kullanımdaki kaynakları raporla
    start_date_str = input("Başlangıç tarihini YYYY-MM-DD formatında girin: ")
    end_date_str = input("Bitiş tarihini YYYY-MM-DD formatında girin: ")

    start_date = datetime.datetime.strptime(start_date_str, '%Y-%m-%d')
    end_date = datetime.datetime.strptime(end_date_str, '%Y-%m-%d')

    cursor.execute("""SELECT Kaynak.KaynakID, Kaynak.Tip, Kaynak.Ozellikler
FROM KaynakKullanım
INNER JOIN Kaynak ON KaynakKullanım.KaynakID =
Kaynak.KaynakID
WHERE KaynakKullanım.BaslangicTarihi >= ? AND
KaynakKullanım.BitisTarihi <= ?""",
start_date, end_date)

    resources = cursor.fetchall()

    if resources:
        print(f"\n{start_date_str} - {end_date_str} Tarihleri Arasında
Kullanımdaki Kaynaklar:")
        print("-" * 30)
        for resource in resources:
            print(f"Kaynak ID: {resource.KaynakID}")
            print(f"Tip: {resource.Tip}")
            print(f"Özellikler: {resource.Ozellikler}")
            print("-" * 30)
    else:
        print(f"\n{start_date_str} - {end_date_str} tarihleri arasında
kullanımda olan kaynak bulunamadı.")

import datetime

def report_resources_not_in_use():
    # Belirli tarih-aralığında kullanılmayan kaynakları raporla
    start_date_str = input("Başlangıç tarihini YYYY-MM-DD formatında girin: ")
    end_date_str = input("Bitiş tarihini YYYY-MM-DD formatında girin: ")

    start_date = datetime.datetime.strptime(start_date_str, '%Y-%m-%d')
    end_date = datetime.datetime.strptime(end_date_str, '%Y-%m-%d')

    cursor.execute("""SELECT Kaynak.KaynakID, Kaynak.Tip, Kaynak.Ozellikler
FROM Kaynak
WHERE Kaynak.KaynakID NOT IN
(SELECT KaynakID FROM KaynakKullanım
WHERE KaynakKullanım.BaslangicTarihi <= ? AND
KaynakKullanım.BitisTarihi >= ?)""",
end_date, start_date)

    resources = cursor.fetchall()

```

```

        if resources:
            print(f"\n{start_date_str} - {end_date_str} Tarihleri Arasında
Kullanılmayan Kaynaklar:")
            print("-" * 30)
            for resource in resources:
                print(f"Kaynak ID: {resource.KaynakID}")
                print(f"Tip: {resource.Tip}")
                print(f"Özellikler: {resource.Ozellikler}")
                print("-" * 30)
        else:
            print(f"\n{start_date_str} - {end_date_str} tarihleri arasında
kullanılmayan kaynak bulunamadı.")

def report_all_customers():
    # Tüm müşterileri raporla
    cursor.execute("SELECT * FROM Musteri")

    customers = cursor.fetchall()

    if customers:
        print("\nTüm Müşteriler:")
        print("-" * 30)
        for customer in customers:
            print(f"Müşteri ID: {customer.MusteriID}")
            print(f"İsim: {customer.Isim}")
            print(f"Soyisim: {customer.Soyisim}")
            print(f"Telefon: {customer.Telefon}")
            print(f"E-mail: {customer.Email}")
            print("-" * 30)
    else:
        print("\nMüşteri bulunamadı.")

import datetime

def report_customers_with_resources():
    # Belirli tarih-aralığında kaynak kullanacak müşterileri raporla
    start_date_str = input("Başlangıç tarihini YYYY-MM-DD formatında girin: ")
    end_date_str = input("Bitiş tarihini YYYY-MM-DD formatında girin: ")

    start_date = datetime.datetime.strptime(start_date_str, '%Y-%m-%d')
    end_date = datetime.datetime.strptime(end_date_str, '%Y-%m-%d')

    cursor.execute("""SELECT Musteri.MusteriID, Musteri.Isim, Musteri.Soyisim,
Musteri.Telefon, Musteri.Email, Kaynak.KaynakID, Kaynak.Tip, Kaynak.Ozellikler
FROM KaynakKullanim
INNER JOIN Musteri ON KaynakKullanim.MusteriID =
Musteri.MusteriID
INNER JOIN Kaynak ON KaynakKullanim.KaynakID =
Kaynak.KaynakID
WHERE KaynakKullanim.BaslangicTarihi >= ? AND
KaynakKullanim.BitisTarihi <= ?""",
start_date, end_date)

    customers = cursor.fetchall()

    if customers:

```

```

        print(f"\n{start_date_str} - {end_date_str} Tarihleri Arasında Kaynak
Kullanacak Müşteriler:")
        print("-" * 30)
        for customer in customers:
            print(f"Müşteri ID: {customer.MusteriID}")
            print(f"İsim: {customer.Isim}")
            print(f"Soyisim: {customer.Soyisim}")
            print(f"Telefon: {customer.Telefon}")
            print(f"E-mail: {customer.Email}")
            print(f"Kullanacağı Kaynak ID: {customer.KaynakID}")
            print(f"Kaynak Tipi: {customer.Tip}")
            print(f"Kaynak Özellikleri: {customer.Ozellikler}")
            print("-" * 30)
        else:
            print(f"\n{start_date_str} - {end_date_str} tarihleri arasında kaynak
kullanacak müşteri bulunamadı.")

def main():
    while True:
        main_menu()
        choice = int(input("Seçiminizi yapınız: "))

        if choice == 0:
            print("Çıkıyor...")
            break
        elif choice == 1:
            report_all_resources()
        elif choice == 2:
            report_specific_resource_type()
        elif choice == 3:
            report_resources_in_use()
        elif choice == 4:
            report_resources_not_in_use()
        elif choice == 5:
            report_all_customers()
        elif choice == 6:
            report_customers_with_resources()
        else:
            print("Geçersiz seçim, lütfen tekrar deneyin.")

if __name__ == "__main__":
    main()

```