

**TUBITAK-UZAY Compulsory Summer
Internship Project**

FINAL DELIVERY REPORT

TUBITAK-UZAY-PyDBControl



TÜBİTAK
UZAY

2025 SUMMER

EDİZ ARKIN KOBAK – Computer Engineer

TABLE OF CONTENTS

| | |
|--|-----------|
| 1. Introduction | 2 |
| 2. Project Scope | 2 |
| 3. Requirements and Solutions | 2 |
| 4. Library Modules | 3 |
| 5. Test and Sample Usage Files Process | 4 |
| 5.1. Test Files (tests/ folder) | 4 |
| 5.2. Example Usage Files (examples/ folder) | 4 |
| 6. Diagrams | 6 |
| 6.1. High Level Architecture Diagram | 6 |
| 6.2. Backend Class Diagram | 7 |
| 7. User Manual | 7 |
| 7.1. Packaging the Library and Using It in Other Projects | 7 |
| 7.1.1. Prepare Project Structure | 7 |
| 7.1.2. Edit the setup.py File | 8 |
| 7.1.3. Adding pyproject.toml File (Optional) | 8 |
| 7.1.4. Create Package | 9 |
| 7.1.5. Usage in Other Projects | 9 |
| 8. Conclusion and Evaluation | 10 |

1. Introduction

TÜBİTAK-UZAY-PyDBControl, PostgreSQL tabanlı büyük veri tablolarıyla etkileşimde bulunmak üzere geliştirilen Python tabanlı bir kontrol kütüphanesidir. Amaç; farklı sistemlerden gelen, yapısal olarak büyük ve yoğun veriler içeren sekiz tabloya kolayca erişim sağlamak, veri güncellemek, karşılaştırmak, tablo tutarlılığını kontrol etmek ve tüm bu işlemleri loglamaktır. Kütüphane, PostgreSQL altyapısıyla doğrudan etkileşir ve geliştiriciye, modül olarak import edilebilen sade bir API sunar.

2. Project Scope

Bu projenin kapsamı:

- PostgreSQL veritabanına Python üzerinden güvenli ve hızlı bağlantı sağlamak
- 8 farklı tabloyla çalışmak (okuma, yazma, güncelleme, silme)
- Tablo tutarlılığını kontrol etmek.
- Aynı yapıya sahip 3 tablo arasında satır bazlı karşılaştırma yapabilmek
- Tüm işlemleri loglamak
- Minimal bağımlılıkla, Linux sistemler üzerinde çalışabilen hafif bir kütüphane olarak yapılandırmak

3. Requirements and Solutions

Aşağıda projenin temel gereksinimleri listelenmiş ve her birinin **nasıl karşılandığı** açıklanmıştır:

| Kod | Gereksinim Tanımı | Karşılanma Açıklaması |
|------|--|---|
| FR-1 | PostgreSQL veritabanına bağlantı kurulabilmelidir. | db_connector.py modülündeki DBConnector sınıfı ile kullanıcı adı, şifre, host ve port parametreleri aracılığıyla PostgreSQL veritabanına bağlantı kurulmaktadır. Hatalı girişlerde exception yakalanıp loglanmaktadır. |

| Kod | Gereksinim Tanımı | Karşılanma Açıklaması |
|------|---|--|
| FR-2 | Tablolardan filtreli veya tüm veri okunabilmelidir. | table_manager.py içinde <i>get_data()</i> fonksiyonu ile tablodaki veriler sorgu parametreleriyle birlikte alınabilmektedir. Filtreleme için SQL WHERE parametreleri desteklenir. |
| FR-3 | Tabloya veri eklenebilmelidir. | <i>insert_row()</i> fonksiyonu ile kullanıcıdan alınan dictionary veri doğrudan PostgreSQL tablosuna eklenmektedir. |
| FR-4 | Kayıtlar güncellenebilmelidir. | <i>update_row()</i> fonksiyonu, ilgili satırı birincil anahtar üzerinden bulup yeni değerlerle günceller. SQL injection'a karşı güvenli yazılmıştır. |
| FR-5 | Kayıtlar koşula göre silinebilmelidir. | <i>delete_row()</i> fonksiyonu ile belirli bir alan değeri eşleşen satırlar silinebilir. Hem performans hem güvenlik için parametrik sorgular kullanılmıştır. |
| FR-6 | Aynı yapıya sahip 3 tablo karşılaştırılabilir. | comparator.py modülünde yer alan <i>compare_tables()</i> fonksiyonu, aynı kolon yapısına sahip iki tabloyu satır bazında karşılaştırır ve farkları kullanıcıya sunar. |
| FR-7 | Tüm işlemler log dosyasına kaydedilmelidir. | logger.py ile işlemler zaman damgası, seviye ve mesaj içeriği ile logs/pydbcontrol.log dosyasına yazılmaktadır. Hem başarı hem hata durumları loglanmaktadır. |
| FR-8 | Veriler opsiyonel olarak pandas DataFrame şeklinde alınabilmelidir. | Bu requirement kütüphanenin ilk 0.1.0 sürümünde karşılanmamış, sonraki sürümlerde ihtiyaç çerçevesinde karşılanacaktır. |

4. Library Modules

Kütüphane beş temel modülden oluşmaktadır:

- **db_connector.py:** PostgreSQL bağlantısı kurar ve temel veritabanı işlemlerini yönetir.
 - **table_manager.py:** Tablo oluşturma, güncelleme, okuma, silme işlemlerini yürütür.
 - **comparator.py:** Aynı yapıya sahip tablolar arasında satır bazlı karşılaştırma yapar.
 - **logger.py:** Tüm işlemleri belirli bir dosyaya kaydeder.
 - **utils.py:** Ortak kullanılabilecek Schema validasyonu gibi yardımcı fonksiyonları barındırır.
-

5. Test and Sample Usage Files Process

PyDBControl kütüphanesinin doğruluğu, stabilitesi ve güvenilirliği aşağıdaki iki temel yol ile doğrulanmıştır:

5.1 Test Files (tests/ folder)

Proje kök dizinindeki **tests/** klasörü altında birim testler (unit test) bulunmaktadır. Bu testler, modüllerin ayrı ayrı işlevlerini doğrulamak için yazılmıştır.

| Test Konusu | Açıklama |
|--|--|
| Veritabanı Bağlantısı | DBConnector sınıfının doğru yapılandırma ile bağlantı kurup kesebildiği test edilmiştir. |
| Veri Ekleme / Okuma / Güncelleme / Silme | TableManager fonksiyonlarının örnek tablo üzerinde veri manipülasyonları yapabildiği ve sonuçları doğru döndürdüğü test edilmiştir. |
| Tablo Karşılaştırma | Comparator modülünün iki tablo arasında satır bazlı farkları başarıyla tespit edip döndürdüğü doğrulanmıştır. |
| İstisna Durumları ve Hatalı Girişler | Hatalı bağlantı bilgileri, eksik parametreler gibi durumlarda sistemin düzgün hata verdiği ve logladığı gözlemlenmiştir. |
| Loglama | Yapılan tüm işlemlerin logs/pydbcontrol.log dosyasına doğru formatta yazılıp yazılmadığı kontrol edilmiştir. |

Testler, **pytest** altyapısıyla çalıştırılabilir:

```
pytest tests/test_exampleTestFile.py
```

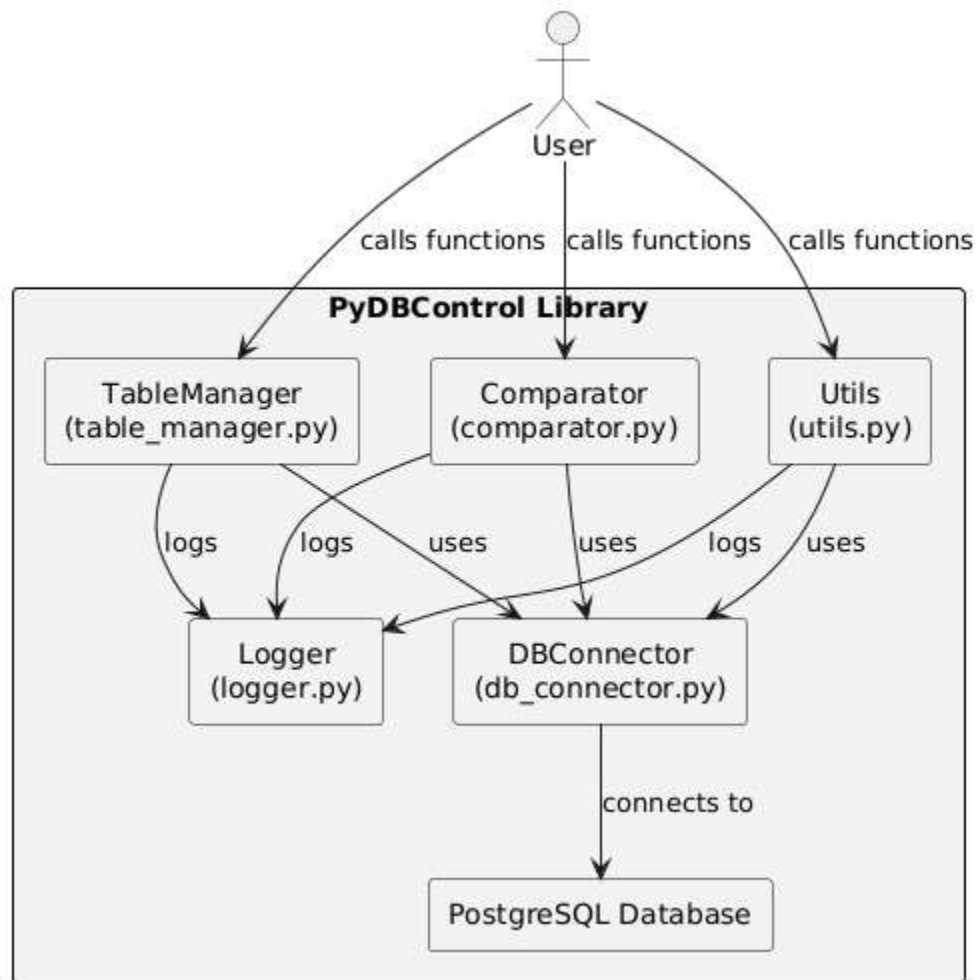
5.2 Example Usage Files (examples/ folder)

Kütüphanenin nasıl kullanılacağını gösteren tam çalışan örnek Python script'leri **examples/** klasöründe yer almaktadır. Bu dosyalar, hem manuel testler hem de son kullanıcılar için rehber niteliğindedir.

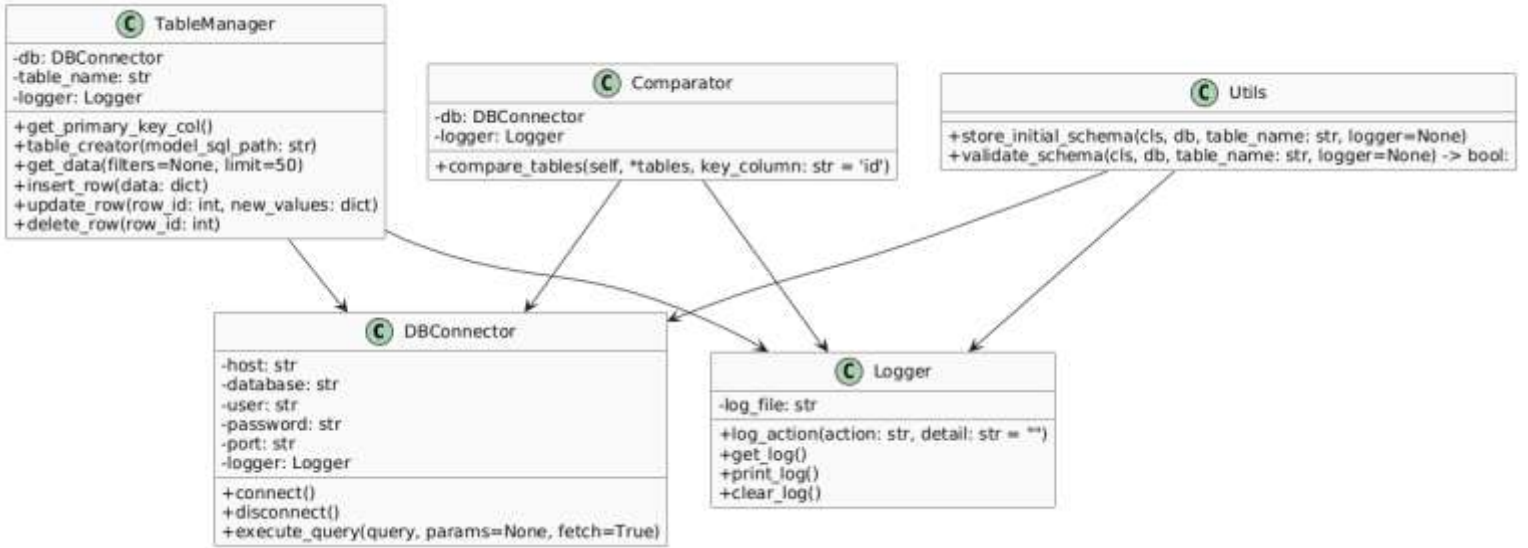
| Dosya Adı | Açıklama |
|-------------------------------------|---|
| create_full_database.py | Veritabanına bağlanıp örnek bütün tabloları oluşturur ve her tabloyu değer aralığında rasgele 2 satır ile doldurur. |
| create_full_table.py | Veritabanına bağlanıp tablo oluşturur. |
| insert_sample_data.py | Veritabanına bağlanıp seçilen tabloyu değer aralığında rastgele 10 satır ile doldurur. |
| insert_row_example.py | Veritabanına bağlanıp seçilen tabloya dictionary olarak verilen veriyi ekler. |
| fetch_data_example.py | Veritabanına bağlanıp seçilen tablodan girilen filtrelemeye göre eşleşen verileri getirir. |
| delete_row_example.py | Veritabanına bağlanıp seçilen tablodan girilen id numarasına göre satır siler. |
| update_row_example.py | Veritabanına bağlanıp seçilen tablodan, seçilen id numarasına göre dictionary olarak alınan yeni verileri girerek satırı günceller. |
| schema_validation_example.py | Veritabanına bağlanıp seçilen tablonun sütun doğruluğunu test eder. |
| compare_tables_example.py | Veritabanına bağlanıp seçilen tabloları karşılaştırır, farklılıklarını kullanıcıya gösterir. |
| logger_db_example.py | Veritabanına bağlanıp veritabanı üzerinde yapılan bütün işlemleri logs/pydbcontrol.log dosyasına kayıt eder. |

6. Diagrams

6.1 High Level Architecture Diagram



6.2 Backend Class Diagram



7. User Manual

7.1 Packaging the Library and Using It in Other Projects

PyDBControl sadece bir proje değil, aynı zamanda kolayca taşınabilir ve yeniden kullanılabilir bir **Python kütüphanesi** olarak geliştirilmiştir. Aşağıda bu kütüphaneyi nasıl paketleyip başka projelerde kullanabileceğin detaylı olarak anlatılmıştır.

7.1.1 Prepare Project Structure

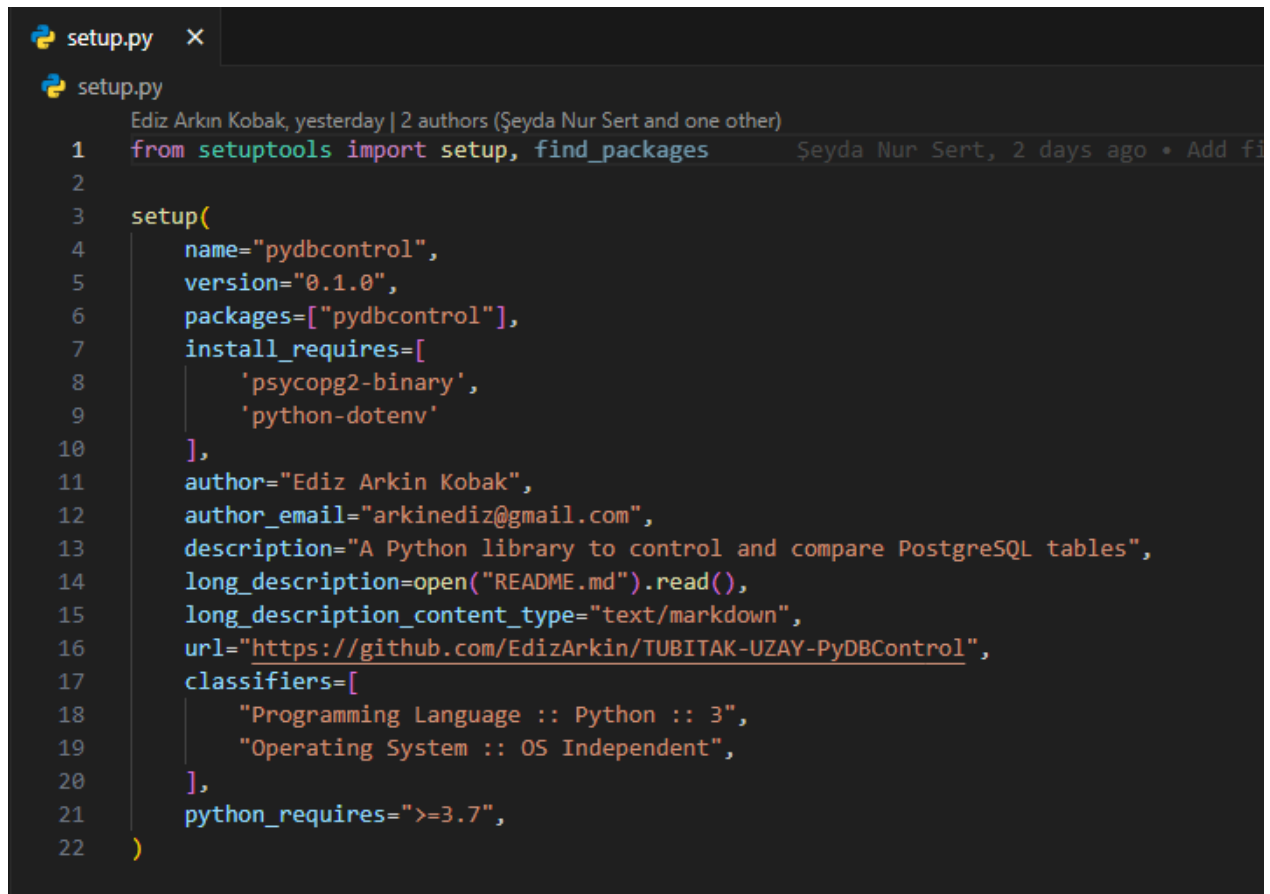
Aşağıdaki gibi bir klasör yapısı olmalıdır:

```
TUBITAK-UZAY-PyDBControl/
├── pydbcontrol/
│   ├── __init__.py
│   ├── db_connector.py
│   ├── table_manager.py
│   ├── comparator.py
│   ├── logger.py
│   └── utils.py
├── tests/
├── examples/
├── requirements.txt
├── setup.py
└── README.md
```

__init__.py dosyasının **pydbcontrol/** içinde olması şarttır. Bu sayede Python burayı bir paket olarak tanır.

7.1.2 Edit the setup.py File

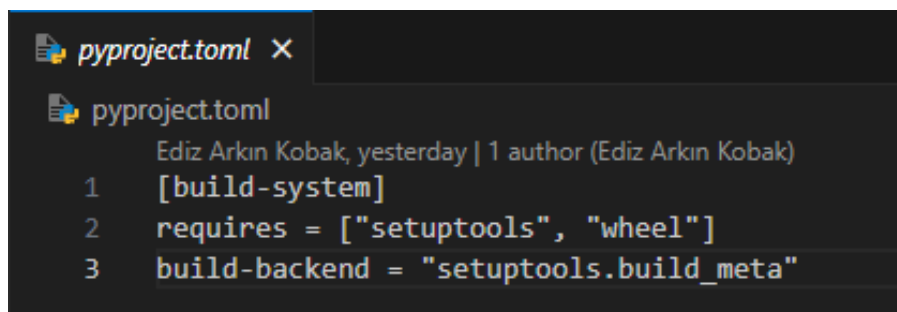
Projenin kök dizininde bir **setup.py** dosyası olmalıdır. Örnek yapı:



```
1 from setuptools import setup, find_packages
2
3 setup(
4     name="pydbcontrol",
5     version="0.1.0",
6     packages=["pydbcontrol"],
7     install_requires=[
8         'psycpg2-binary',
9         'python-dotenv'
10    ],
11     author="Ediz Arkin Kobak",
12     author_email="arkinediz@gmail.com",
13     description="A Python library to control and compare PostgreSQL tables",
14     long_description=open("README.md").read(),
15     long_description_content_type="text/markdown",
16     url="https://github.com/EdizArkin/TUBITAK-UZAY-PyDBControl",
17     classifiers=[
18         "Programming Language :: Python :: 3",
19         "Operating System :: OS Independent",
20     ],
21     python_requires=">=3.7",
22 )
```

7.1.3 Adding pyproject.toml File (Optional)

Modern Python packaging için aşağıdaki dosyayı da eklemen önerilir:



```
1 [build-system]
2 requires = ["setuptools", "wheel"]
3 build-backend = "setuptools.build_meta"
```

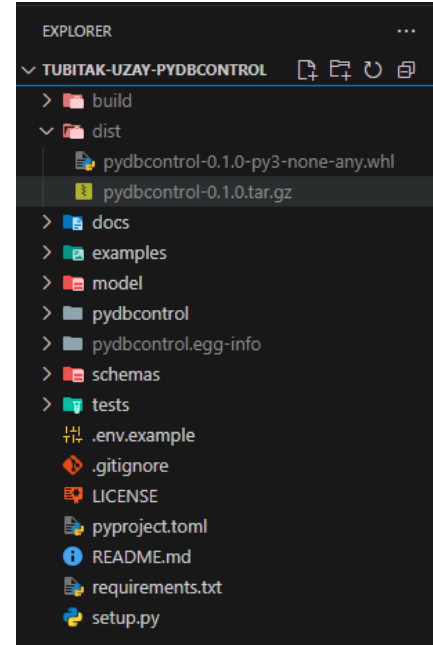
7.1.4 Create Package

Terminalde proje klasörüne gidip şunu çalıştır:

```
python setup.py sdist bdist_wheel
```

Bu işlemden sonra **dist/** klasörü içinde şunlar oluşur:

- **pydbcontrol-0.1.0.tar.gz**
- **pydbcontrol-0.1.0-py3-none-any.whl**



7.1.5 Usage in Other Projects

a. Use with Unreleased Package (local) - (Recommended)

Kütüphane **wheel** dosyasıyla kurulabilir:

```
pip install /path/to/pydbcontrol-0.1.0-py3-none-any.whl
```

b. Installation via Git Repo

requirements.txt dosyasına şunu ekle:

```
git+https://github.com/EdizArkin/TUBITAK-UZAY-PyDBControl.git
```

Veya terminalden:

```
pip install git+https://github.com/EdizArkin/TUBITAK-UZAY-PyDBControl.git
```

c. Installation in Developer Mode

Kütüphaneyi bir proje içinde canlı değişikliklerle kullanmak için:

```
pip install -e .
```

Bu sayede proje klasöründe yaptığın değişiklikler, kullanan uygulamaya anında yansır.

8. Conclusion and Evaluation

TÜBİTAK-UZAY-PyDBControl projesi, veritabanı etkileşiminde güvenilir, hızlı ve modüler bir Python arayüzü sağlamaktadır. Özellikle büyük tablolar üzerinde yapılan okuma-yazma işlemlerinde optimize edilmiş bir yapı sunması, loglama yetenekleri ve tablo karşılaştırma modülü sayesinde benzer veri kaynakları arasında tutarlılık sağlaması açısından başarıyla tamamlanmıştır.

Kod, genişlemeye açık şekilde yazılmıştır ve gelecekte farklı veritabanı motorlarını destekleyecek mimariye dönüştürülebilir.

Prepared by: Ediz Arkin Kobak

Date: 24.07.2025

Repo: <https://github.com/EdizArkin/TUBITAK-UZAY-PyDBControl>