📝 Database Pattern Implementation

🎯 Analisi del Problema

Il problema richiede la creazione di una libreria per interagire con diversi database mantenendo un'interfaccia uniforme. Le caratteristiche chiave sono:

- Selezione dinamica del database
- Interfaccia unificata
- Estensibilità per nuovi database

🎨 Pattern Scelti: Strategy + Factory Method

Ho scelto la combinazione di Strategy e Factory Method perché:

1. Strategy: Permette di definire una famiglia di algoritmi intercambiabili
2. Factory Method: Gestisce la creazione degli oggetti database in modo flessibile

Vantaggi ✅

- Maggiore flessibilità nella selezione del database
- Incapsulamento della logica di creazione
- Facilità di aggiungere nuove strategie
- Separazione tra creazione e utilizzo

Svantaggi ❌

- Maggiore complessità iniziale
- Più classi da gestire
- Overhead di performance minimo

💻 Implementazione

```
# Strategy Pattern - Interfaccia base
class DatabaseStrategy(ABC):
    @abstractmethod
    def find(self, params: Dict) -> Any:
        pass

# Esempio di strategia concreta
```

```python
class MongoDBStrategy(DatabaseStrategy):
    def find(self, params: Dict) -> Any:
        print(f"MongoDB: Ricerca con parametri {params}")
        return {"result": "MongoDB data"}

# Factory Method base
class DatabaseFactory(ABC):
    @abstractmethod
    def create_database(self, connection_string: str) -> DatabaseStrategy:
        pass
```

🎬 Output dell'implementazione

```
Connesso a MongoDB: mongodb://localhost:27017
MongoDB: Ricerca con parametri {'user': 'mario'}
```

🔍 Spiegazione dei Pattern

Strategy Pattern

- Definisce una famiglia di algoritmi (strategie di database)
- Rende gli algoritmi intercambiabili
- Permette la selezione dell'algoritmo a runtime

Factory Method Pattern

- Incapsula la creazione degli oggetti database
- Permette di estendere facilmente con nuovi tipi di database
- Mantiene il codice pulito e organizzato

Il flusso delle operazioni è:

1. La factory crea l'implementazione appropriata del database
2. Il context utilizza la strategia selezionata
3. Le operazioni vengono eseguite attraverso l'interfaccia comune