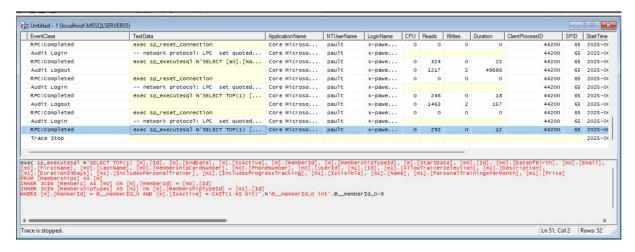
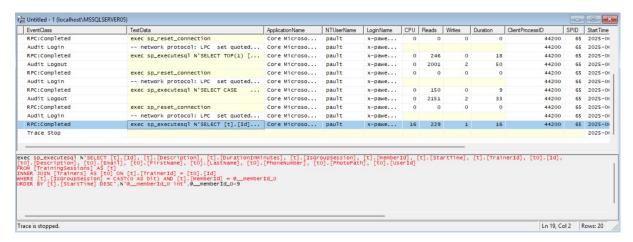
Opis działania zapytania: Podczas wywołania endpointu GET /api/memberships przez użytkownika z rolą Member, aplikacja wykonuje zapytanie SQL pobierające pierwszy aktywny rekord z tabeli Memberships, spełniający warunki Memberld = @id AND IsActive = 1 (po prostu dla aktualnie zalogowanego użytkownika). W SQL Profilerze widzimy to jako zapytanie SELECT TOP(1) z INNER JOIN na Members i MembershipTypes.



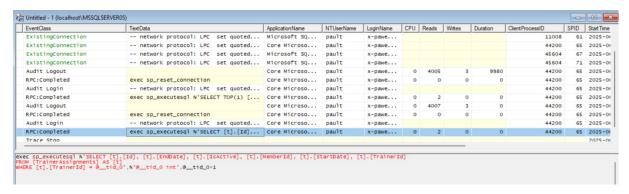
Komentarz: Dzięki uprzednio dodanemu indeksowi (IX_Memberships_MemberId_IsActive), zapytanie wykonało się z użyciem Index Seek, co pozwala uniknąć pełnego przeszukiwania tabeli (Clustered Scan). Profiler potwierdza jego efektywne wykonanie i szybki czas odpowiedzi 12ms.

Opis działania zapytania: Podczas wywołania endpointu GET /api/training-sessions/self przez użytkownika z rolą Member, aplikacja wykonuje zapytanie SQL spełniające warunki IsGroupSession = 0 AND MemberId = @id (po prostu prywatne sesje zalogowanego membera). W SQL Profilerze widzimy to jako zapytanie które jest zaznaczone na niebiesko.



Komentarz: Zapytanie wykonało się z użyciem Clustered Index Scan, przeszukując wszystkie recordy. Zajęło to 16ms, czyli więcej niż w poprzednim przypadku – dla mniejszej ilości stron danych, które zostały odczytane z pamięci (każda strona 8KB *292 to jest około 2.3MB danych).

Opis działania zapytania: Podczas wywołania endpointu GET /api/trainer-assignments/me przez użytkownika z rolą Trainer, aplikacja wykonuje zapytanie SQL spełniające warunki TrainerId = @id (po prostu zapisani memberzy dla zalogowanego trainera). W SQL Profilerze widzimy to jako zapytanie które jest zaznaczone na niebiesko.



Komentarz: Zapytanie wykonało się z użyciem Clustered Index Scan, przeszukując wszystkie recordy. Zajęło to Oms, jednakże z powodu małej ilości danych.