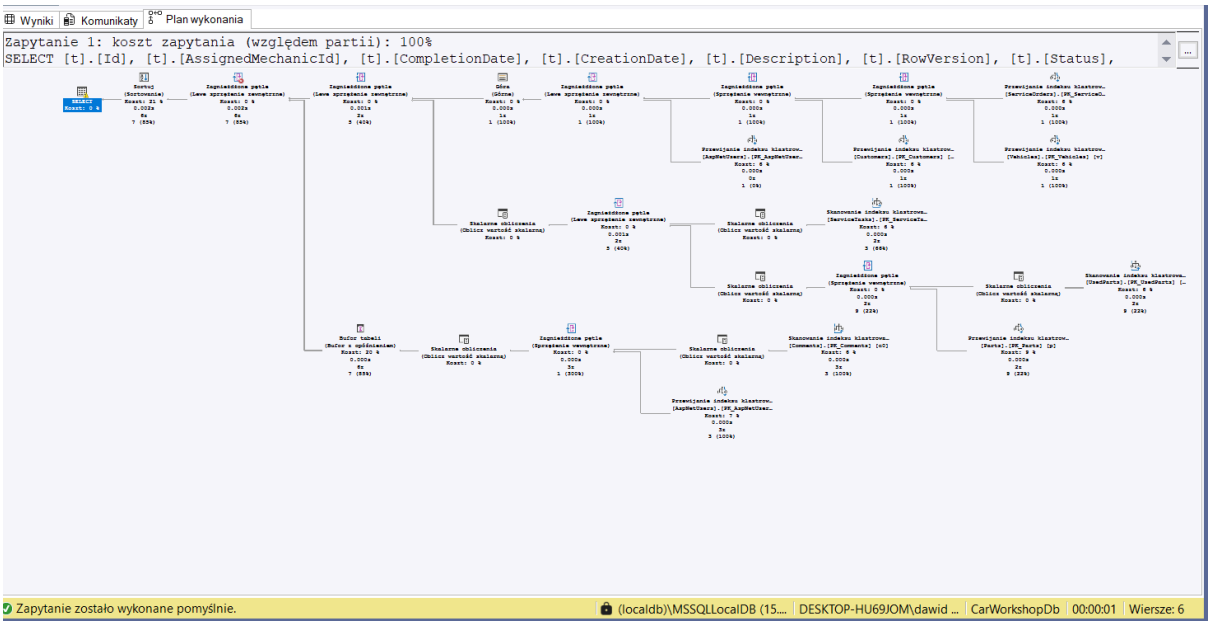
1. Indeksy – optymalizacja zapytań

📌 Zadanie:

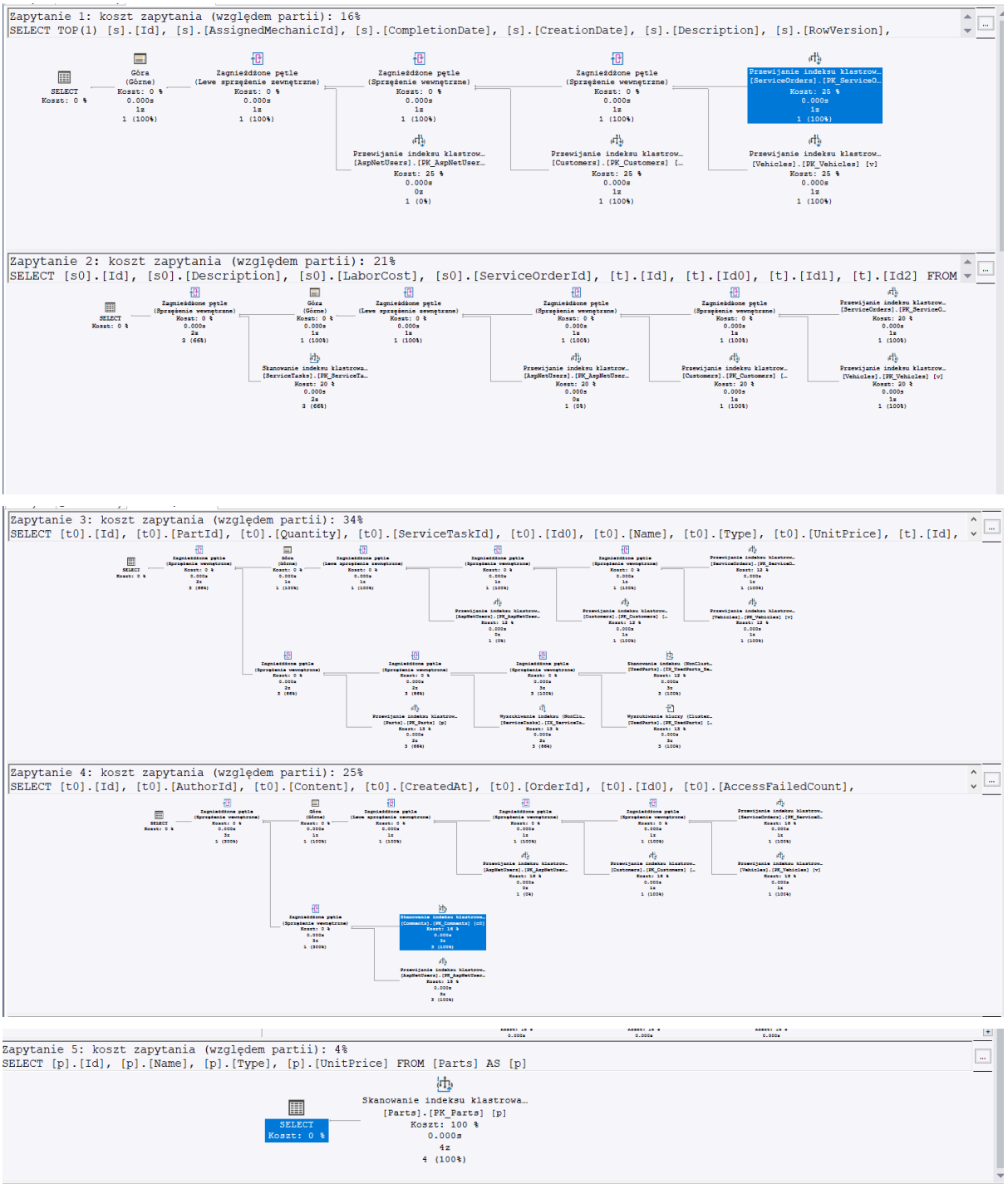
- Zidentyfikuj co najmniej dwa zapytania SELECT, które są często wykonywane i mają WHERE lub JOIN po kolumnie nie kluczowej.
- Dodaj indeksy nieklastrowane (non-clustered) do wybranych kolumn.
- Zrób analizę wydajności:
    - Zrzut planu zapytania (Query Plan) przed i po dodaniu indeksu.
    - Krótkie porównanie (np. liczba odczytów, operacje przeszukiwania vs seek).
    - Umieść to w raporcie PDF z opisem + screenshotami.

Definicja problemu :
Zapytanie GetOrderByIdAsync było problematyczne i nieoptymalne. Dominowały operacje typu **Clustered Index Scan** oraz **Nested Loops Join** z wysokim kosztem.
Rozwiązanie :
Dodanie indeksów na kluczach obcych i często wyszukiwanych kolumnach, użycie AsSplitQuery(). Zamiast kosztownych operacji **Clustered Index Scan**, w planie dominują teraz operacje **Index Seek** (Przeszukiwanie indeksu) oraz **Key Lookup.**

1. Przed optymalizacją :



2. Po optymalizacji :



1. Przed optymalizacją :

```
SELECT [t].[Id], [t].[AssignedMechanicId], [t].[CompletionDate],
[t].[CreationDate], [t].[Description], [t].[RowVersion], [t].[Status],
[t].[VehicleId], [t].[Id0], [t].[Brand], [t].[CustomerId],
[t].[ImageUrl], [t].[Model], [t].[RegistrationNumber], [t].[VIN],
[t].[Year], [t].[Id1], [t].[FullName], [t].[PhoneNumber], [t].[Id2],
[t].[AccessFailedCount], [t].[ConcurrencyStamp], [t].[Email],
[t].[EmailConfirmed], [t].[FirstName], [t].[LastName],
[t].[LockoutEnabled], [t].[LockoutEnd], [t].[NormalizedEmail],
[t].[NormalizedUserName], [t].[PasswordHash], [t].[PhoneNumber0],
```

```
[t].[PhoneNumberConfirmed], [t].[SecurityStamp], [t].[TwoFactorEnabled],
[t].[UserName], [t0].[Id], [t0].[Description], [t0].[LaborCost],
[t0].[ServiceOrderId], [t0].[Id0], [t0].[PartId], [t0].[Quantity],
[t0].[ServiceTaskId], [t0].[Id00], [t0].[Name], [t0].[Type],
[t0].[UnitPrice], [t2].[Id], [t2].[AuthorId], [t2].[Content],
[t2].[CreatedAt], [t2].[OrderId], [t2].[Id0], [t2].[AccessFailedCount],
[t2].[ConcurrencyStamp], [t2].[Email], [t2].[EmailConfirmed],
[t2].[FirstName], [t2].[LastName], [t2].[LockoutEnabled],
[t2].[LockoutEnd], [t2].[NormalizedEmail], [t2].[NormalizedUserName],
[t2].[PasswordHash], [t2].[PhoneNumber], [t2].[PhoneNumberConfirmed],
[t2].[SecurityStamp], [t2].[TwoFactorEnabled], [t2].[UserName]
FROM (
    SELECT TOP(1) [s].[Id], [s].[AssignedMechanicId],
[s].[CompletionDate], [s].[CreationDate], [s].[Description],
[s].[RowVersion], [s].[Status], [s].[VehicleId], [v].[Id] AS [Id0],
[v].[Brand], [v].[CustomerId], [v].[ImageUrl], [v].[Model],
[v].[RegistrationNumber], [v].[VIN], [v].[Year], [c].[Id] AS [Id1],
[c].[FullName], [c].[PhoneNumber], [a].[Id] AS [Id2],
[a].[AccessFailedCount], [a].[ConcurrencyStamp], [a].[Email],
[a].[EmailConfirmed], [a].[FirstName], [a].[LastName],
[a].[LockoutEnabled], [a].[LockoutEnd], [a].[NormalizedEmail],
[a].[NormalizedUserName], [a].[PasswordHash], [a].[PhoneNumber] AS
[PhoneNumber0], [a].[PhoneNumberConfirmed], [a].[SecurityStamp],
[a].[TwoFactorEnabled], [a].[UserName]
    FROM [ServiceOrders] AS [s]
    INNER JOIN [Vehicles] AS [v] ON [s].[VehicleId] = [v].[Id]
    INNER JOIN [Customers] AS [c] ON [v].[CustomerId] = [c].[Id]
    LEFT JOIN [AspNetUsers] AS [a] ON [s].[AssignedMechanicId] =
[a].[Id]
    WHERE [s].[Id] = @__id_0
) AS [t]
LEFT JOIN (
    SELECT [s0].[Id], [s0].[Description], [s0].[LaborCost],
[s0].[ServiceOrderId], [t1].[Id] AS [Id0], [t1].[PartId],
[t1].[Quantity], [t1].[ServiceTaskId], [t1].[Id0] AS [Id00],
[t1].[Name], [t1].[Type], [t1].[UnitPrice]
    FROM [ServiceTasks] AS [s0]
    LEFT JOIN (
        SELECT [u].[Id], [u].[PartId], [u].[Quantity],
[u].[ServiceTaskId], [p0].[Id] AS [Id0], [p0].[Name], [p0].[Type],
[p0].[UnitPrice] -- Zmieniono alias p na p0 w tym miejscu
        FROM [UsedParts] AS [u]
        INNER JOIN [Parts] AS [p0] ON [u].[PartId] = [p0].[Id] --
Zmieniono alias p na p0 w tym miejscu
    ) AS [t1] ON [s0].[Id] = [t1].[ServiceTaskId]
) AS [t0] ON [t].[Id] = [t0].[ServiceOrderId]
```

```
LEFT JOIN (
    SELECT [c0].[Id], [c0].[AuthorId], [c0].[Content], [c0].[CreatedAt],
[c0].[OrderId], [a0].[Id] AS [Id0], [a0].[AccessFailedCount],
[a0].[ConcurrencyStamp], [a0].[Email], [a0].[EmailConfirmed],
[a0].[FirstName], [a0].[LastName], [a0].[LockoutEnabled],
[a0].[LockoutEnd], [a0].[NormalizedEmail], [a0].[NormalizedUserName],
[a0].[PasswordHash], [a0].[PhoneNumber], [a0].[PhoneNumberConfirmed],
[a0].[SecurityStamp], [a0].[TwoFactorEnabled], [a0].[UserName]
    FROM [Comments] AS [c0]
    INNER JOIN [AspNetUsers] AS [a0] ON [c0].[AuthorId] = [a0].[Id]
) AS [t2] ON [t].[Id] = [t2].[OrderId]
ORDER BY [t].[Id], [t].[Id0], [t].[Id1], [t].[Id2], [t0].[Id],
[t0].[Id0], [t0].[Id00], [t2].[Id];
```

2. Po optymalizacji :

```
SELECT TOP(1) [s].[Id], [s].[AssignedMechanicId], [s].[CompletionDate],
[s].[CreationDate], [s].[Description], [s].[RowVersion], [s].[Status],
[s].[VehicleId], [v].[Id], [v].[Brand], [v].[CustomerId],
[v].[ImageUrl], [v].[Model], [v].[RegistrationNumber], [v].[VIN],
[v].[Year], [c].[Id], [c].[FullName], [c].[PhoneNumber], [a].[Id],
[a].[AccessFailedCount], [a].[ConcurrencyStamp], [a].[Email],
[a].[EmailConfirmed], [a].[FirstName], [a].[LastName],
[a].[LockoutEnabled], [a].[LockoutEnd], [a].[NormalizedEmail],
[a].[NormalizedUserName], [a].[PasswordHash], [a].[PhoneNumber],
[a].[PhoneNumberConfirmed], [a].[SecurityStamp], [a].[TwoFactorEnabled],
[a].[UserName]
FROM [ServiceOrders] AS [s]
INNER JOIN [Vehicles] AS [v] ON [s].[VehicleId] = [v].[Id]
INNER JOIN [Customers] AS [c] ON [v].[CustomerId] = [c].[Id]
LEFT JOIN [AspNetUsers] AS [a] ON [s].[AssignedMechanicId] = [a].[Id]
WHERE [s].[Id] = @__id_0
ORDER BY [s].[Id], [v].[Id], [c].[Id], [a].[Id];

SELECT [s0].[Id], [s0].[Description], [s0].[LaborCost],
[s0].[ServiceOrderId]
FROM [ServiceTasks] AS [s0]
WHERE [s0].[ServiceOrderId] = @__id_0
ORDER BY [s0].[Id];

SELECT [t0].[Id], [t0].[PartId], [t0].[Quantity], [t0].[ServiceTaskId],
[p].[Id] AS [Id0], [p].[Name], [p].[Type], [p].[UnitPrice], [t].[Id],
[t].[Id0], [t].[Id1], [t].[Id2], [s0].[Id]
FROM (
    SELECT TOP(1) [s].[Id], [v].[Id] AS [Id0], [c].[Id] AS [Id1],
[a].[Id] AS [Id2]
```

```
    FROM [ServiceOrders] AS [s]
    INNER JOIN [Vehicles] AS [v] ON [s].[VehicleId] = [v].[Id]
    INNER JOIN [Customers] AS [c] ON [v].[CustomerId] = [c].[Id]
    LEFT JOIN [AspNetUsers] AS [a] ON [s].[AssignedMechanicId] =
[a].[Id]
    WHERE [s].[Id] = @__id_0
) AS [t]
INNER JOIN [ServiceTasks] AS [s0] ON [t].[Id] = [s0].[ServiceOrderId]
INNER JOIN [UsedParts] AS [t0] ON [s0].[Id] = [t0].[ServiceTaskId]
INNER JOIN [Parts] AS [p] ON [t0].[PartId] = [p].[Id] -- Tutaj p.Id
powinno być OK
ORDER BY [t].[Id], [t].[Id0], [t].[Id1], [t].[Id2], [s0].[Id],
[t0].[Id]; -- Dodano t0.Id do ORDER BY

SELECT [t].[Id], [t].[AuthorId], [t].[Content], [t].[CreatedAt],
[t].[OrderId], [t0].[Id], [t0].[AccessFailedCount],
[t0].[ConcurrencyStamp], [t0].[Email], [t0].[EmailConfirmed],
[t0].[FirstName], [t0].[LastName], [t0].[LockoutEnabled],
[t0].[LockoutEnd], [t0].[NormalizedEmail], [t0].[NormalizedUserName],
[t0].[PasswordHash], [t0].[PhoneNumber], [t0].[PhoneNumberConfirmed],
[t0].[SecurityStamp], [t0].[TwoFactorEnabled], [t0].[UserName]
FROM [Comments] AS [t]
INNER JOIN [AspNetUsers] AS [t0] ON [t].[AuthorId] = [t0].[Id]
WHERE [t].[OrderId] = @__id_0
ORDER BY [t].[Id];

SELECT [p].[Id], [p].[Name], [p].[Type], [p].[UnitPrice]
FROM [Parts] AS [p];
```

Wnioski :
Prosty zabieg dodania kilku indeksów w kluczowych miejscach bazy danych sprawił, że nasze najważniejsze zapytanie o szczegóły zlecenia stało się jakieś **84% szybsze** (patrząc na względny koszt).