

Czy można w kontrolce Dane tabelaryczne uzależnić wybór wartości z listy od drugiej listy?

W kontrolce **Dane tabelaryczne** (DT) nie ma obecnie możliwości uzależnienia zawartości jednej listy od drugiej (brak obsługi kolekcji list zależnych wewnątrz tabeli). Mechanizm „obserwowania pól” nie działa pomiędzy kolumnami w obrębie tego samego wiersza.

Rozwiązanie:

Można zrealizować ten scenariusz, przenosząc proces wyboru wartości poza tabelę. Należy dodać dwie niezależne kontrolki typu **Lista** oraz przycisk typu **Własna akcja**, który zasili tabelę danymi.

Kroki konfiguracji:

- 1. Dodanie pól wyboru:** Poza kontrolką Dane tabelaryczne należy umieścić dwie niezależne listy (np. *Dział* i *Pracownik*), konfigurując między nimi standardową zależność (obserwowanie pól).
- 2. Konfiguracja Własnej akcji:** Należy dodać przycisk (np. „Dodaj”), który będzie pełnił funkcję wyzwalacza zapisu. Kontrolka ta nie zawiera kodu.
- 3. Inicjowanie wartości w tabeli:** W ustawieniach kontrolki Dane tabelaryczne (zakładka *Inicjowanie wartości*) należy zdefiniować zapytanie SQL, które po kliknięciu przycisku

pobierze wartości z pól pomocniczych i wstawi je jako nowy wiersz do tabeli.

Przykład kodu SQL dla kontrolki Dane tabelaryczne

```
IF isnull ( @^DocumentId@ ,0) <>0
```

```
Begin
```

```
If @^SenderControlName@ ='Dodaj'
```

```
Begin
```

```
declare @dcdLogin varchar (30);
```

```
IF isnull ( @Pracownik_Id@ ,0) <>0
```

```
Begin
```

```
select @dcdLogin= DCD_Login from do.DF_ConfOSDictionary where  
dcd_id= @Pracownik_Id@
```

```
IF exists (select * from @Uprawnienia@ where  
[Login]=@dcdLogin)
```

```
Return
```

```
insert into @Uprawnienia@ values (@dcdLogin, @Pracownik_Text@  
,0)
```

```
select * from @Uprawnienia@
```

```
END
```

```
END
```

```
END
```

Uwaga! Zmiana nazwy kontrolki może spowodować nieprawidłowe działanie wyrażenia.

```

1 IF ISNULL (@*DocumentId@ ,0) <>0
2 BEGIN
3
4 IF @*SenderControlName@ = "Dodaj"
5
6 BEGIN
7 DECLARE @dcdLogin VARCHAR (30);
8
9 IF ISNULL (@*Pracownik_Id@ ,0) <>0
10 BEGIN
11
12 SELECT @dcdLogin= DCD_Login FROM do.DF_Conf0SDictionary WHERE dcd_id= @*Pracownik_Id@
13
14 IF EXISTS (SELECT * FROM @Uprawnienia@ WHERE [Login]=@dcdLogin)
15 RETURN
16
17 INSERT INTO @Uprawnienia@ VALUES (@dcdLogin, @*Pracownik_Text@ ,0)
18
19 SELECT * FROM @Uprawnienia@

```

Kontrolki

- ^DocumentCompanyId
- ^UserId
- ^DocumentId
- ^WorkflowId
- ^StagId
- ^InitSenderControlName
- ^SenderControlName

Składnia:
SELECT
varchar(900) - Login
varchar(900) - Pracownik
POS - pozycja (opcjonalny numer wiersza)

Przykład użycia:
BEGIN TRY
SELECT 'txt1', 'txt2'
UNION
SELECT 'txt1', 'txt2'

Dane tabelaryczne

OGÓLNE LISTY INICJOWANIE WARTOŚCI OGRANICZENIE EDYCJI

Inicjowanie wartości: Brak

Kontrolki powiązane: Dodaj (Dodaj), Pracownik (Pracownik), Dodaj

Wzór na wartość: SQL OD

Konfiguracja kontrolki Dane tabelaryczne

Przykład kodu SQL dla kontrolki Dział:

```
select dcd_id, DCD_Name1 from do.DF_Conf0SDictionary where dcd_type=1 and DCD_Archival=0
```

Przykład kodu SQL dla kontrolki Pracownik, która obserwuje kontrolkę Dział:

```
select t2.dcd_id, t2.dcd_Name1 from do.DF_Conf0STree as t1
```

```
inner join do.DF_Conf0SDictionary as t2 on t1.DCO_DCDId=t2.DCD_ID
```

```
inner join do.DF_ConfOSTree as t3 on t1.DCO_DCOId=t3.DCO_ID
```

```
left join do.DF_ConfOSDictionary as t4 on  
t4.DCD_ID=t3.DCO_DCDId
```

```
where t1.DCO_Archival=0 and t2.dcd_type=2 and  
t4.DCD_Name1=@Dzial_Text@
```

Efekt końcowy:

The screenshot displays a software interface for document workflow management. At the top, the document ID 'DYN/1/4/2026' is shown along with navigation icons. The main area is divided into two panels. The left panel, titled 'Dokumenty', contains the 'Karta obiegu' (Flow Card) for a document created on 15-04-2026. It includes dropdown menus for 'Dział' (Department) set to 'Firma' and 'Pracownik' (Employee) set to 'Anna Lis'. Below these is a 'Dodaj' (Add) button and a table of permissions. The right panel, titled 'Przebieg' (Process), shows a flow diagram with a single step 'Rejestracja' (Registration) performed by 'Administrator (DS)' on 15.04.2026 at 13:24. Below the diagram are sections for 'Uprawnienia do obecnego etapu' (Permissions for the current step) and 'Następny etap' (Next step), which is 'Weryfikacja' (Verification).

LP	Login	Pracownik
1	ALIS	Anna Lis
2	AdamN	Adam Nowak

Karta obiegu z uzupełnionymi kontrolkami

[Kliknij tutaj, aby pobrać przykład definicji typu obiegu](#)

Rozpoczynasz pracę z Comarch BPM (dawniej DMS) i chcesz dowiedzieć się, jak korzystać z programu? A może masz już podstawową wiedzę o Comarch BPM (dawniej DMS) i chcesz dowiedzieć się więcej?

[Sprawdź Szkolenia Comarch BPM!](#)

[Powrót do początku artykułu](#)