

Variablenbezogene Auswertungen

Prozessvariablen-Template

Variablen in 3-er-Blöcken einbinden ist performanter. Besser mehrere 3-er-Blöcke benutzen als ein riesiges Join. Noch besser wenn die *values* (Indices) aus der view_instance benutzt werden können.

```
SELECT Variablen1.1, Variablen1.3, Variablen1.3
FROM
(
  SELECT var1.stringvalue AS "1", var2.stringvalue AS "2", var3.stringvalue AS
"3", inst.ID
FROM view_instance inst
LEFT JOIN view_variable var1 ON inst.ID = var1.instanceID AND var1."name" =
''
LEFT JOIN view_variable var2 ON inst.ID = var2.instanceID AND var2."name" =
''
LEFT JOIN view_variable var3 ON inst.ID = var3.instanceID AND var3."name" =
''
) AS Variablen1
INNER JOIN
(
  SELECT var1.stringvalue AS "1", var2.stringvalue AS "2", var3.stringvalue AS
"3", inst.ID
FROM view_instance inst
LEFT JOIN view_variable var1 ON inst.ID = var1.instanceID AND var1."name" =
''
LEFT JOIN view_variable var2 ON inst.ID = var2.instanceID AND var2."name" =
''
LEFT JOIN view_variable var3 ON inst.ID = var3.instanceID AND var3."name" =
''
) AS Variablen2
ON Variablen1.ID = Variablen2.ID
```

Prozessinstanzen inklusive beliebig vielen Prozessvariablen auflisten

Mit folgendem Statement kann eine Auflistung von Prozessinstanzen inklusive Darstellung von Indexvariablen sowie beliebigen Prozessvariablen erstellt werden.

Hinweis: Zur Vorsortierung der Ergebnisliste muss die komplette SQL-Anweisung als INNER-SELECT geschachtelt und anschließend sortiert werden.

```
SELECT DISTINCT
  pi."name",
  /*
    Auslesen der Indexvariablen 1 bis 10 bei Bedarf:
  */
  CASE WHEN pi.value1 IS NOT NULL THEN pi.value1 ELSE '-' END AS "Index 1",
```

```
CASE WHEN pi.value2 IS NOT NULL THEN pi.value2 ELSE '-' END AS "Index 2",
CASE WHEN pi.value3 IS NOT NULL THEN pi.value3 ELSE '-' END AS "Index 3",
/*
  Auslesen der gewünschten Prozessvariablen:
  - Für jede Variable muss eine processVariables.- Zeile eingefügt werden
*/
processVariables.VARIABLE_1_OHNE_LEERZEICHEN AS "Header Variable 1",
processVariables.VARIABLE_2_OHNE_LEERZEICHEN AS "Header Variable 2",
processVariables.VARIABLE_3_OHNE_LEERZEICHEN AS "Header Variable 3",
identP.name AS "StarterID",
pi.definitionName AS "Prozessdefinition",
pi.id AS "ProzessID"
FROM view_instance pi
LEFT JOIN view_identity identP ON pi.creationUserId = identP.id
LEFT JOIN (
  SELECT piInner.id,
  /*
    Erstellen einer MAX(...) Zeile für jede gewünschte Variable
  */
  MAX( CASE WHEN var.name = 'PROZESSVARIABLE 1' THEN
var.stringValue ELSE '-' END ) AS VARIABLE_1_OHNE_LEERZEICHEN,
  MAX( CASE WHEN var.name = 'PROZESSVARIABLE 2' THEN
var.stringValue ELSE '-' END ) AS VARIABLE_2_OHNE_LEERZEICHEN,
  MAX( CASE WHEN var.name = 'PROZESSVARIABLE 3' THEN
var.stringValue ELSE '-' END ) AS VARIABLE_3_OHNE_LEERZEICHEN
FROM view_instance piInner
  /*
    Im folgenden IN (...) Statement müssen alle Variablennamen
    aufgelistet werden, die selektiert werden sollen:
  */
  LEFT JOIN view_variable var ON var.instanceId = piInner.id AND
var.name IN ('PROZESSVARIABLE 1', 'PROZESSVARIABLE 2', 'PROZESSVARIABLE 3')
  GROUP BY piInner.id
) processVariables ON pi.id = processVariables.id
```

From:
<https://wiki.tim-solutions.de/> - **TIM Wiki** / **NEW TIM 6 Documentation**

Permanent link:
https://wiki.tim-solutions.de/doku.php?id=software:dashboard:analyses:variable_analyses_oracle

Last update: **2021/07/01 09:52**

