

# Instanzbezogene Auswertungen

## Anzahl an gestarteten Instanzen je Tag

```
SELECT
    concat(to_char(inst.creationtime,'yy'),
    concat(to_char(inst.creationtime,'mm'), to_char(inst.creationtime,'dd'))) AS
Tag, SUM(1) AS Anzahl
FROM view_instance inst
GROUP BY concat(to_char(inst.creationtime,'yy'),
concat(to_char(inst.creationtime,'mm'), to_char(inst.creationtime,'dd')))
```

## Durchschnittliche Anzahl an gestarteten Instanzen je Tag pro Monat (bei 22 Arbeitstagen im Monat)

```
SELECT
    concat(to_char(inst.creationtime,'yy'), to_char(inst.creationtime,'mm'))
AS Monat, ROUND(SUM(1)/22,1) AS "Durchschnitt pro Tag"
FROM view_instance inst
GROUP BY concat(to_char(inst.creationtime,'yy'),
to_char(inst.creationtime,'mm'))
```

## Anzahl an gestarteten Instanzen je Monat

```
SELECT
    concat(to_char(inst.creationtime,'yy'), to_char(inst.creationtime,'mm'))
AS Monat, SUM(1) AS Anzahl
FROM view_instance inst
GROUP BY concat(to_char(inst.creationtime,'yy'),
to_char(inst.creationtime,'mm'))
```

## Anzahl an gestarteten und beendeten Instanzen je Definition

```
SELECT
inst1.definitionname AS "Prozess", COUNT(DISTINCT(inst1.id)) AS "Anzahl
gestarteter Instanzen", COUNT(DISTINCT(inst2.id)) AS "Anzahl beendeter
Instanzen"
FROM view_instance inst1
LEFT JOIN view_instance inst2
ON inst1.id=inst2.id
AND inst2.END IS NOT NULL
GROUP BY inst1.definitionname
```

## Anzahl an gestarteten und beendeten Instanzen sowie die Anzahl an offenen Aufgaben einer Prozessdefinition

```
SELECT
inst1."DEFINITIONNAME" AS "Prozess",
COUNT(inst1.id) AS "Anzahl gestarteter Instanzen",
COUNT(inst2.id) AS "Anzahl beendeter Instanzen",
(SELECT COUNT(DISTINCT(task.id))
FROM view_task task
INNER JOIN view_instance inst ON task.instanceid = inst.id
INNER JOIN view_activity act ON act.instanceid = inst.id
WHERE task.isopen = '1'
AND act.end IS NULL
AND inst."END" IS NULL
AND inst.archiv = 0
AND inst.definitionname = inst1."DEFINITIONNAME") AS "Anzahl offene
Aufgaben"
FROM view_instance inst1 LEFT JOIN view_instance inst2 ON inst1.id=inst2.id
AND inst2.END IS NOT NULL
WHERE inst1."DEFINITIONNAME" LIKE 'GL-01%'
GROUP BY inst1.definitionname
```

## Anzahl an gestarteten, beendeten, abgebrochenen und laufenden Instanzen je Prozessdefinition

```
SELECT
definitionname,
SUM(1) AS "alle",
SUM(
CASE
WHEN archiv = 1 AND END IS NULL
THEN
1
ELSE
0
END) AS "abgebrochen",
SUM(
CASE
WHEN END IS NOT NULL
THEN
1
ELSE
0
END) AS "beendet",
SUM(
CASE
WHEN END IS NULL AND archiv = 0
```

```

THEN
1
ELSE
0
END) AS "laufend"
FROM view_instance GROUP BY definitionname

```

## Finanzielle Aufwände je Instanz

```

SELECT inst."name" AS Instanzname, SUM(CAST(eff.VALUE AS INT)) AS
Finanzaufwände
FROM view_effort eff
INNER JOIN view_instance inst
ON eff.instanceID=inst.id
WHERE eff.effortType='FINANCE'
GROUP BY inst.id, inst."name"

```

## Anzahl der Schleifen (>0) je Instanz

```

SELECT
  inst.DEFinitionName AS Prozess, inst."name" AS Instanz,
  SUM(
    CASE
      WHEN act.loopcount IS NOT NULL
      THEN 1
      ELSE 0
    END
  ) AS Schleifendurchläufe
FROM view_instance inst, view_activity act
WHERE act.loopcount IS NOT NULL
AND act.instanceID = inst.id
GROUP BY inst."name", inst.DEFinitionName

```

## Aktuelle Aktivität(en) und Bearbeiter aller offenen Instanzen

```

SELECT
inst."name" AS Prozessinstanz,
inst.definitionName AS Prozessdefinition,
  LISTAGG(act."name", ', ' ) WITHIN GROUP (ORDER BY act."name") AS
"Aktivität",
  LISTAGG(task."name", ', ' ) WITHIN GROUP (ORDER BY task."name") AS
"Aufgaben",
  LISTAGG(
    CASE
      WHEN ident.id IS NULL
      THEN
        ident2."name"

```

```
        ELSE
            CONCAT(CONCAT(CONCAT(CONCAT(CONCAT(ident.firstname, ' '),
ident.lastname), ' ( '), ident2."name"), ')')
        END ,
        ', ') WITHIN GROUP (ORDER BY ident2."name") AS "Username"
FROM view_instance inst
INNER JOIN view_activity act
ON act.instanceid = inst.id
AND act."start" IS NOT NULL
AND act."end" IS NULL AND act."type" = 'K'
INNER JOIN view_task task
ON task.activity = act.id
LEFT JOIN view_identity ident
ON task.actor = ident.id
LEFT JOIN view_identity ident2
ON task.pooledActor = ident2.id
WHERE inst.archiv = 0
AND inst.END IS NULL
GROUP BY inst.id, inst."name", inst.definitionName
```

## Alle offenen Instanzen mit einer bestimmten Zeichenfolge im Instanznamen

```
SELECT DISTINCT vi."name" AS "Instanzname", vi."DEFINITIONNAME" AS
"Prozessname"
FROM view_activity a, view_instance vi
WHERE a."end" IS NULL
AND a."start" IS NOT NULL
AND vi.id = a.instanceId
AND vi.archiv = 0
AND vi."END" IS NULL
AND vi."name" LIKE '%tester%'
```

## Durchschnittliche Durchlaufzeit (in hh:mm:ss) von allen Instanzen aller Prozessdefinitionen je Monat

```
SELECT Monat,
CASE
WHEN FLOOR(sekunden)<60
THEN
    CONCAT(CASE WHEN FLOOR(sekunden)<10 THEN '00:0' ELSE '00:'
END, FLOOR(sekunden))
    WHEN FLOOR(sekunden)>3600
    THEN
        CONCAT(CONCAT(CONCAT(CONCAT(h, CASE WHEN FLOOR (FLOOR(sekunden) - h*3600 -
sek)/60<10 THEN ':0' ELSE ':' END), FLOOR(FLOOR(sekunden) - h*3600 -
sek)/60), CASE WHEN sek<10 THEN ':0' ELSE ':' END), sek)
    ELSE
```

```

CONCAT(CONCAT(CONCAT(CASE WHEN FLOOR(sekunden/60)<10 THEN '00:0' ELSE
'00:' END ,FLOOR(sekunden/60)), CASE WHEN FLOOR(sek)<10 THEN ':0' ELSE ':'
END),FLOOR(sek))
END AS "Durchschn. DLZ"
FROM
(SELECT
FLOOR(AVG((TO_NUMBER (TO_CHAR (END, 'J')) - TO_NUMBER (TO_CHAR
(creationtime, 'J')) * 86400 + ( TO_NUMBER (TO_CHAR (END, 'SSSSS')) -
TO_NUMBER (TO_CHAR (creationtime, 'SSSSS')))) AS sekunden,
FLOOR(MOD(AVG((TO_NUMBER (TO_CHAR (END, 'J')) - TO_NUMBER (TO_CHAR
(creationtime, 'J')) * 86400 + ( TO_NUMBER (TO_CHAR (END, 'SSSSS')) -
TO_NUMBER (TO_CHAR (creationtime, 'SSSSS'))),60)) AS sek,
FLOOR(AVG((TO_NUMBER (TO_CHAR (END, 'J')) - TO_NUMBER (TO_CHAR
(creationtime, 'J')) * 86400 + ( TO_NUMBER (TO_CHAR (END, 'SSSSS')) -
TO_NUMBER (TO_CHAR (creationtime, 'SSSSS')))/3600) AS h,
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(creationtime,0,(INSTR(creationtime,
'.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3),' '),EXTRACT(YEAR
FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-
MM-YY HH24:MI:SS')) AS Monat
FROM view_instance
WHERE END IS NOT NULL
GROUP BY
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(creationtime,0,(INSTR(creationtime,
'.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3),' '),EXTRACT(YEAR
FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-
MM-YY HH24:MI:SS'))
)

```

## Durchschnittliche Durchlaufzeit (in hh:mm:ss) der Instanzen einer bestimmten Prozessdefinition je Monat

```

SELECT Monat,
CASE
WHEN FLOOR(sekunden)<60
THEN
CONCAT(CASE WHEN FLOOR(sekunden)<10 THEN '00:0' ELSE '00:'
END,FLOOR(sekunden))
WHEN FLOOR(sekunden)>3600
THEN
CONCAT(CONCAT(CONCAT(CONCAT(h,CASE WHEN FLOOR (FLOOR(sekunden)- h*3600-
sek)/60<10 THEN ':0' ELSE ':' END),FLOOR(FLOOR(sekunden)- h*3600-
sek)/60),CASE WHEN sek<10 THEN ':0' ELSE ':' END),sek)
ELSE
CONCAT(CONCAT(CONCAT(CASE WHEN FLOOR(sekunden/60)<10 THEN '00:0' ELSE '00:'
END ,FLOOR(sekunden/60)), CASE WHEN FLOOR(sek)<10 THEN ':0' ELSE ':'
END),FLOOR(sek))
END AS "Durchschn. DLZ"
FROM
(SELECT

```

```
FLOOR(AVG((TO_NUMBER (TO_CHAR (END, 'J')) - TO_NUMBER (TO_CHAR
(creationtime, 'J'))) * 86400 + ( TO_NUMBER (TO_CHAR (END, 'SSSSS')) -
TO_NUMBER (TO_CHAR (creationtime, 'SSSSS')))) AS sekunden,
FLOOR(MOD(AVG((TO_NUMBER (TO_CHAR (END, 'J')) - TO_NUMBER (TO_CHAR
(creationtime, 'J'))) * 86400 + ( TO_NUMBER (TO_CHAR (END, 'SSSSS')) -
TO_NUMBER (TO_CHAR (creationtime, 'SSSSS'))),60)) AS sek,
FLOOR(AVG((TO_NUMBER (TO_CHAR (END, 'J')) - TO_NUMBER (TO_CHAR
(creationtime, 'J'))) * 86400 + ( TO_NUMBER (TO_CHAR (END, 'SSSSS')) -
TO_NUMBER (TO_CHAR (creationtime, 'SSSSS')))/3600) AS h,
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3), ' '),EXTRACT(YEAR
FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-
MM-YY HH24:MI:SS')) AS Monat,
EXTRACT(YEAR FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.',
-1)-1)), 'DD-MM-YY HH24:MI:SS')) AS j,
EXTRACT(MONTH FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.',
-1)-1)), 'DD-MM-YY HH24:MI:SS')) AS m
FROM view_instance
WHERE END IS NOT NULL
AND definitionName = 'NAME DER PROZESSDEFINITION'
GROUP BY
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3), ' '),EXTRACT(YEAR
FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-
MM-YY HH24:MI:SS'))),
EXTRACT(YEAR FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.',
-1)-1)), 'DD-MM-YY HH24:MI:SS')),
EXTRACT(MONTH FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.',
-1)-1)), 'DD-MM-YY HH24:MI:SS'))
)
ORDER BY j, m
```

## Durchschnittliche Durchlaufzeit (in h) von allen Instanzen aller Prozessdefinitionen je Monat

```
SELECT Monat, h
AS "Durchschn. DLZ in H" FROM
(SELECT CAST((AVG((TO_NUMBER (TO_CHAR (END, 'J')) - TO_NUMBER (TO_CHAR
(creationtime, 'J'))) * 86400 + ( TO_NUMBER (TO_CHAR (END, 'SSSSS')) -
TO_NUMBER (TO_CHAR (creationtime, 'SSSSS')))/3600) AS DECIMAL(10,1)) AS h,
CONCAT(CONCAT( EXTRACT(YEAR FROM
to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS')) , ' '),EXTRACT(MONTH FROM
to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS')) AS Monat
FROM view_instance
WHERE END IS NOT NULL
```

```

GROUP BY
CONCAT(CONCAT( EXTRACT(YEAR FROM
to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS')) , ' '),EXTRACT(MONTH FROM
to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS'))))
)

```

## Durchschnittliche Durchlaufzeit (in h) von allen Instanzen einer Prozessdefinitionen je Monat

```

SELECT Monat, h
AS "Durchschn. DLZ in H" FROM
(SELECT CAST((AVG((TO_NUMBER (TO_CHAR (END, 'J')) - TO_NUMBER (TO_CHAR
(creationtime, 'J')) * 86400 + ( TO_NUMBER (TO_CHAR (END, 'SSSSS')) -
TO_NUMBER (TO_CHAR (creationtime, 'SSSSS')))/3600) AS DECIMAL(10,1)) AS h,
CONCAT(CONCAT( EXTRACT(YEAR FROM
to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS')) , ' '),EXTRACT(MONTH FROM
to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS')))) AS Monat
FROM view_instance
WHERE END IS NOT NULL
AND definitionname = 'S-AP-G'
GROUP BY
CONCAT(CONCAT( EXTRACT(YEAR FROM
to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS')) , ' '),EXTRACT(MONTH FROM
to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS'))))
)

```

## Instanzen je Aktivität einer Prozessdefinition (analog zur Auswertung in den PM-Clients)

```

SELECT
CASE WHEN INSTR(va.name, ' (L:)' > 0 THEN SUBSTR(va.name, 0,
INSTR(va.name, ' (L:)'-1) ELSE va.name END AS "Aktivitäten",
COUNT(va.name) AS Anzahl
FROM view_activity va JOIN view_instance vi ON vi.id = va.instanceId
WHERE va.end IS NULL
AND va."start" IS NOT NULL
AND (va.type = 'K' OR va.type = 'C')
AND vi.definitionName = 'PROZESSDEFINITIONSNAME'
AND vi.archiv = 0
GROUP BY CASE WHEN INSTR(va.name, ' (L:)' > 0 THEN SUBSTR(va.name, 0,
INSTR(va.name, ' (L:)'-1) ELSE va.name END

```

## Anzahl der gestarteten und beendeten Instanzen je Monat und Definition

```
SELECT "q1Monat" AS "Monat", "q1Prozess" AS "Prozess", "q1Gestartete" AS
"Anzahl gestartete Instanzen", "q1Beendete" AS "Anzahl beendete Instanzen"
FROM
(
    SELECT
        COUNT(DISTINCT(inst1.id)) AS "q1Gestartete",
        CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(inst1.creationtime,0,(INSTR(inst1.creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month') ,1,3), '
'),EXTRACT(YEAR FROM
to_date(SUBSTR(inst1.creationtime,0,(INSTR(inst1.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS')) AS "q1Monat",
        inst1.definitionname AS "q1Prozess",
        COUNT(DISTINCT(inst2.id)) AS "q1Beendete",
        EXTRACT(YEAR FROM
to_date(SUBSTR(inst1.creationtime,0,(INSTR(inst1.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS')) AS "q1Jahr",
        EXTRACT(MONTH FROM
to_date(SUBSTR(inst1.creationtime,0,(INSTR(inst1.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS')) AS "q1Monat2"
    FROM view_instance inst1
    LEFT OUTER JOIN view_instance inst2
    ON
        to_char(to_date(SUBSTR(inst1.creationtime,0,(INSTR(inst1.creationtime,
        '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month') =
        to_char(to_date(SUBSTR(inst2."END",0,(INSTR(inst2."END", '.', -1)-1)), 'DD-
        MM-YY HH24:MI:SS'), 'Month')
        AND
        EXTRACT(YEAR FROM
to_date(SUBSTR(inst1.creationtime,0,(INSTR(inst1.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS')) = EXTRACT(YEAR FROM
to_date(SUBSTR(inst2."END",0,(INSTR(inst2."END", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS'))
        AND inst1.definitionname=inst2.definitionname
    GROUP BY
        CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(inst1.creationtime,0,(INSTR(inst1.creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month') ,1,3), '
'),EXTRACT(YEAR FROM
to_date(SUBSTR(inst1.creationtime,0,(INSTR(inst1.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS'))),
        inst1.definitionname,
        EXTRACT(YEAR FROM
to_date(SUBSTR(inst1.creationtime,0,(INSTR(inst1.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS')),
        EXTRACT(MONTH FROM
to_date(SUBSTR(inst1.creationtime,0,(INSTR(inst1.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS'))
    UNION
```



```

SELECT
    COUNT(DISTINCT(inst1.id)) AS "q1Gestartete",
    CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(inst2.END,0,(INSTR(inst2.END,
    '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3), ' '),EXTRACT(YEAR FROM
    to_date(SUBSTR(inst2.END,0,(INSTR(inst2.END, '.', -1)-1)), 'DD-MM-YY
    HH24:MI:SS')) AS "q1Monat",
    inst2.definitionname AS "q1Prozess",
    COUNT(DISTINCT(inst2.id)) AS "q1Beendete",
    EXTRACT(YEAR FROM
    to_date(SUBSTR(inst2.creationtime,0,(INSTR(inst2.creationtime, '.', -1)-1)),
    'DD-MM-YY HH24:MI:SS')) AS "q1Jahr",
    EXTRACT(MONTH FROM to_date(SUBSTR(inst2.end,0,(INSTR(inst2.end, '.',
    -1)-1)), 'DD-MM-YY HH24:MI:SS')) AS "q1Monat2"
    FROM view_instance inst1
    RIGHT OUTER JOIN view_instance inst2
    ON
        to_char(to_date(SUBSTR(inst1.creationtime,0,(INSTR(inst1.creationtime,
        '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month') =
        to_char(to_date(SUBSTR(inst2."END",0,(INSTR(inst2."END", '.', -1)-1)), 'DD-
        MM-YY HH24:MI:SS'), 'Month')
        AND
        EXTRACT(YEAR FROM
        to_date(SUBSTR(inst1.creationtime,0,(INSTR(inst1.creationtime, '.', -1)-1)),
        'DD-MM-YY HH24:MI:SS')) = EXTRACT(YEAR FROM
        to_date(SUBSTR(inst2."END",0,(INSTR(inst2."END", '.', -1)-1)), 'DD-MM-YY
        HH24:MI:SS'))
        AND inst1.definitionname=inst2.definitionname
    WHERE inst2.END IS NOT NULL
    GROUP BY
    CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(inst2."END",0,(INSTR(inst2."END"
    , '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3), ' '),EXTRACT(YEAR
    FROM to_date(SUBSTR(inst2."END",0,(INSTR(inst2."END", '.', -1)-1)), 'DD-MM-
    YY HH24:MI:SS'))),
        inst2.definitionname,
        EXTRACT(YEAR FROM
        to_date(SUBSTR(inst2.creationtime,0,(INSTR(inst2.creationtime, '.', -1)-1)),
        'DD-MM-YY HH24:MI:SS')),
        EXTRACT(MONTH FROM to_date(SUBSTR(inst2.end,0,(INSTR(inst2.end, '.',
        -1)-1)), 'DD-MM-YY HH24:MI:SS'))
    )
    ORDER BY "q1Jahr","q1Monat2","q1Prozess"

```

## Anzahl der gestarteten Instanzen je Monat (Spalten) je Jahr (Zeilen) für eine Definition

```

SELECT
    Jahr AS "Jahr",
    SUM(
        CASE

```

```
WHEN Monat LIKE '%January%'
THEN 1
ELSE 0
END) AS "Januar",
SUM(
CASE
WHEN Monat LIKE '%February%'
THEN 1
ELSE 0
END) AS "Februar",
SUM(
CASE
WHEN Monat LIKE '%March%'
THEN 1
ELSE 0
END) AS "März",
SUM(
CASE
WHEN Monat LIKE '%April%'
THEN 1
ELSE 0
END) AS "April",
SUM(
CASE
WHEN Monat LIKE '%May%'
THEN 1
ELSE 0
END) AS "Mai",
SUM(
CASE
WHEN Monat LIKE '%June%'
THEN 1
ELSE 0
END) AS "Juni",
SUM(
CASE
WHEN Monat LIKE '%July%'
THEN 1
ELSE 0
END) AS "Juli",
SUM(
CASE
WHEN Monat LIKE '%August%'
THEN 1
ELSE 0
END) AS "August",
SUM(
CASE
WHEN Monat LIKE '%September%'
THEN 1
```

```

        ELSE 0
      END) AS "September",
    SUM(
      CASE
        WHEN Monat LIKE '%October%'
        THEN 1
        ELSE 0
      END) AS "Oktober",
    SUM(
      CASE
        WHEN Monat LIKE '%November%'
        THEN 1
        ELSE 0
      END) AS "November",
    SUM(
      CASE
        WHEN Monat LIKE '%December%'
        THEN 1
        ELSE 0
      END) AS "Dezember"

FROM
  (SELECT to_char(to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.',
-1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month') AS Monat,
    EXTRACT(YEAR FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime,
'.', -1)-1)), 'DD-MM-YY HH24:MI:SS')) AS Jahr,
    definitionname
  FROM view_instance
  WHERE definitionname = 'S-AP-G'
)
GROUP BY Jahr

```

## Anzahl der gestarteten Instanzen je Monat (Zeilen) je Jahr (Spalten) für eine Definition

```

SELECT Monat,
SUM(
CASE
WHEN Jahr = '14'
THEN 1
ELSE 0
END) AS "2014",
SUM(
CASE
WHEN Jahr = '15'
THEN 1
ELSE 0
END) AS "2015"
FROM

```

```
(SELECT
to_char(to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)),
'DD-MM-YYYY HH24:MI:SS'), 'Month') AS Monat,
EXTRACT(YEAR FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)),
'DD-MM-YYYY HH24:MI:SS')) AS Jahr,
EXTRACT(MONTH FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS')) AS m,
definitionname
FROM view_instance vi
WHERE definitionname = 'PROZESSDEFINITION'
)
GROUP BY Monat,m
ORDER BY m
```

## Kumulierte Anzahl der gestarteten Instanzen je Monat (Spalten) je Jahr (Zeilen) für eine Definition

```
SELECT
Jahr,
SUM(
CASE
WHEN Monat LIKE '%January%'
THEN 1
ELSE 0
END) AS "Januar",
SUM(
CASE
WHEN Monat LIKE '%January%' OR Monat LIKE '%February%'
THEN 1
ELSE 0
END) AS "Februar",
SUM(
CASE
WHEN Monat LIKE '%January%' OR Monat LIKE '%February%' OR Monat LIKE '%March%'
THEN 1
ELSE 0
END) AS "März",
SUM(
CASE
WHEN Monat LIKE '%January%' OR Monat LIKE '%February%' OR Monat LIKE '%March%' OR Monat LIKE '%April%'
THEN 1
ELSE 0
END) AS "April",
SUM(
CASE
WHEN Monat LIKE '%January%' OR Monat LIKE '%February%' OR Monat LIKE
```

```

'%March%' OR Monat LIKE '%April%' OR Monat LIKE '%May%'
    THEN 1
    ELSE 0
    END) AS "Mai",
SUM(
    CASE
        WHEN Monat LIKE '%January%' OR Monat LIKE '%February%' OR Monat LIKE
'%March%' OR Monat LIKE '%April%' OR Monat LIKE '%May%' OR Monat LIKE
'%June%'
        THEN 1
        ELSE 0
        END) AS "Juni",
SUM(
    CASE
        WHEN Monat LIKE '%January%' OR Monat LIKE '%February%' OR Monat LIKE
'%March%' OR Monat LIKE '%April%' OR Monat LIKE '%May%' OR Monat LIKE
'%June%' OR Monat LIKE '%July%'
        THEN 1
        ELSE 0
        END) AS "Juli",
SUM(
    CASE
        WHEN Monat LIKE '%January%' OR Monat LIKE '%February%' OR Monat LIKE
'%March%' OR Monat LIKE '%April%' OR Monat LIKE '%May%' OR Monat LIKE
'%June%' OR Monat LIKE '%July%' OR Monat LIKE '%August%'
        THEN 1
        ELSE 0
        END) AS "August",
SUM(
    CASE
        WHEN Monat LIKE '%January%' OR Monat LIKE '%February%' OR Monat LIKE
'%March%' OR Monat LIKE '%April%' OR Monat LIKE '%May%' OR Monat LIKE
'%June%' OR Monat LIKE '%July%' OR Monat LIKE '%August%' OR Monat LIKE
'%September%'
        THEN 1
        ELSE 0
        END) AS "September",
SUM(
    CASE
        WHEN Monat LIKE '%January%' OR Monat LIKE '%February%' OR Monat LIKE
'%March%' OR Monat LIKE '%April%' OR Monat LIKE '%May%' OR Monat LIKE
'%June%' OR Monat LIKE '%July%' OR Monat LIKE '%August%' OR Monat LIKE
'%September%' OR Monat LIKE '%October%'
        THEN 1
        ELSE 0
        END) AS "Oktober",
SUM(
    CASE
        WHEN Monat LIKE '%January%' OR Monat LIKE '%February%' OR Monat LIKE
'%March%' OR Monat LIKE '%April%' OR Monat LIKE '%May%' OR Monat LIKE
'%June%' OR Monat LIKE '%July%' OR Monat LIKE '%August%' OR Monat LIKE

```

```
'%September%' OR Monat LIKE '%October%' OR Monat LIKE '%November%'
THEN 1
ELSE 0
END) AS "November",
SUM(
CASE
WHEN Monat LIKE '%January%' OR Monat LIKE '%February%' OR Monat LIKE
'%March%' OR Monat LIKE '%April%' OR Monat LIKE '%May%' OR Monat LIKE
'%June%' OR Monat LIKE '%July%' OR Monat LIKE '%August%' OR Monat LIKE
'%September%' OR Monat LIKE '%October%' OR Monat LIKE '%November%' OR Monat
LIKE '%December%'
THEN 1
ELSE 0
END) AS "Dezember"
FROM
(SELECT
to_char(to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS'), 'Month') AS Monat,
EXTRACT(YEAR FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime,
'.', -1)-1)), 'DD-MM-YY HH24:MI:SS')) AS Jahr
FROM view_instance
WHERE definitionname = 'PROZESSDEFINITIONSNAME'
)
GROUP BY Jahr
```

## Kumulierte Anzahl der gestarteten Instanzen je Monat (Zeilen) je Jahr (Spalten) für eine Definition

```
SELECT Monat,
(SELECT SUM("a2015")
FROM
(SELECT aMonat, aMonatzahl,
SUM(CASE
WHEN aJahr = '15'
THEN 1
ELSE 0
END) AS "a2015"
FROM (
SELECT
to_char(to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS'), 'Month') AS aMonat,
EXTRACT(YEAR FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime,
'.', -1)-1)), 'DD-MM-YY HH24:MI:SS')) AS aJahr,
EXTRACT(MONTH FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime,
'.', -1)-1)), 'DD-MM-YY HH24:MI:SS')) AS aMonatzahl
FROM view_instance
WHERE definitionname = 'PROZESSDEFINITIONSNAME'
)
)
```

```
GROUP BY aMonat, aMonatzahl
ORDER BY aMonatzahl
)
WHERE aMonatzahl <= Monatzahl) AS "Summe 2015",

(SELECT SUM("b2016")
FROM (SELECT bMonat, bMonatzahl,
SUM(CASE
WHEN bJahr = '16'
THEN 1
ELSE 0
END) AS "b2016"
FROM (
SELECT
to_char(to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS'), 'Month') AS bMonat,
EXTRACT(YEAR FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime,
'.', -1)-1)), 'DD-MM-YY HH24:MI:SS')) AS bJahr,
EXTRACT(MONTH FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime,
'.', -1)-1)), 'DD-MM-YY HH24:MI:SS')) AS bMonatzahl
FROM view_instance
WHERE definitionname = 'PROZESSDEFINITIONSNAME'
)
GROUP BY bMonat, bMonatzahl
ORDER BY bMonatzahl
)
WHERE bMonatzahl <= Monatzahl) AS "Summe 2016",

(SELECT SUM("c2017")
FROM (SELECT cMonat, cMonatzahl,
SUM(CASE
WHEN cJahr = '17'
THEN 1
ELSE 0
END) AS "c2017"
FROM (
SELECT
to_char(to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS'), 'Month') AS cMonat,
EXTRACT(YEAR FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime,
'.', -1)-1)), 'DD-MM-YY HH24:MI:SS')) AS cJahr,
EXTRACT(MONTH FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime,
'.', -1)-1)), 'DD-MM-YY HH24:MI:SS')) AS cMonatzahl
FROM view_instance
WHERE definitionname = 'PROZESSDEFINITIONSNAME'
)
GROUP BY cMonat, cMonatzahl
ORDER BY cMonatzahl
)
WHERE cMonatzahl <= Monatzahl) AS "Summe 2017"
```

```
FROM
(SELECT Monat, Monatszahl,
SUM(CASE
WHEN Jahr = '15'
THEN 1
ELSE 0
END) AS "2015",
SUM(CASE
WHEN Jahr = '16'
THEN 1
ELSE 0
END) AS "2016",
SUM(CASE
WHEN Jahr = '17'
THEN 1
ELSE 0
END) AS "2017"
FROM
(SELECT
to_char(to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS'), 'Month') AS Monat,
EXTRACT(YEAR FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime,
'.', -1)-1)), 'DD-MM-YY HH24:MI:SS')) AS Jahr,
EXTRACT(MONTH FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime,
'.', -1)-1)), 'DD-MM-YY HH24:MI:SS')) AS Monatszahl
FROM view_instance
WHERE definitionname = 'PROZESSDEFINITIONSNAME'
)
GROUP BY Monat, Monatszahl
ORDER BY Monatszahl
)[]
```

## Minimale, maximale und durchschnittliche Laufzeit (in hh:mm:ss) der Instanzen je Prozessdefinition

```
SELECT
name AS "Prozess",

CASE WHEN
INSTR(CASE
WHEN FLOOR(avgSekunden)<60
THEN
CONCAT(CASE WHEN FLOOR(avgSekunden)<10 THEN '00:00:0' ELSE '00:00:'
END, FLOOR(avgSekunden))
WHEN FLOOR(avgSekunden)>3600
THEN
CONCAT(CONCAT(CONCAT(CONCAT(avgH, CASE WHEN FLOOR((FLOOR(avgSekunden) -
avgH*3600 - avgSek)/60)<10 THEN ':0' ELSE ':' END), FLOOR((FLOOR(avgSekunden) -
```



```

avgH*3600-avgSek)/60)), CASE WHEN avgSek<10 THEN ':0' ELSE ':' END), avgSek)
ELSE
CONCAT(CONCAT(CONCAT(CASE WHEN FLOOR(avgSekunden/60)<10 THEN '00:0' ELSE
'00:' END,FLOOR(avgSekunden/60)), CASE WHEN FLOOR(avgSek)<10 THEN ':0' ELSE
':' END), FLOOR(avgSek))
END, '-',1,1)
>0
THEN '00:00:00'
ELSE
CASE
WHEN FLOOR(avgSekunden)<60
THEN
CONCAT(CASE WHEN FLOOR(avgSekunden)<10 THEN '00:00:0' ELSE '00:00:'
END,FLOOR(avgSekunden))
WHEN FLOOR(avgSekunden)>3600
THEN
CONCAT(CONCAT(CONCAT(CONCAT(avgH, CASE WHEN FLOOR((FLOOR(avgSekunden)-
avgH*3600-avgSek)/60)<10 THEN ':0' ELSE ':' END), FLOOR((FLOOR(avgSekunden)-
avgH*3600-avgSek)/60)), CASE WHEN avgSek<10 THEN ':0' ELSE ':' END), avgSek)
ELSE
CONCAT(CONCAT(CONCAT(CASE WHEN FLOOR(avgSekunden/60)<10 THEN '00:0' ELSE
'00:' END,FLOOR(avgSekunden/60)), CASE WHEN FLOOR(avgSek)<10 THEN ':0' ELSE
':' END), FLOOR(avgSek))
END
END AS "Durschn. DLZ",

CASE WHEN
INSTR(
CASE
WHEN FLOOR(minSekunden)<60
THEN
CONCAT(CASE WHEN FLOOR(minSekunden)<10 THEN '00:00:0' ELSE '00:00:' END,
FLOOR(minSekunden))
WHEN FLOOR(minSekunden)>3600
THEN
CONCAT(CONCAT(CONCAT(CONCAT(minH,CASE WHEN FLOOR((FLOOR(minSekunden)-
minH*3600-minSek)/60)<10 THEN ':0' ELSE ':' END),FLOOR((FLOOR(minSekunden)-
minH*3600-minSek)/60)),CASE WHEN minSek<10 THEN ':0' ELSE ':' END),minSek)
ELSE
CONCAT(CONCAT(CONCAT(CASE WHEN FLOOR(minSekunden/60)<10 THEN '00:0' ELSE
'00:' END,FLOOR(minSekunden/60)),CASE WHEN FLOOR(minSek)<10 THEN ':0' ELSE
':' END),FLOOR(minSek))
END
, '-',1,1)
>0
THEN '00:00:00'
ELSE
CASE
WHEN FLOOR(minSekunden)<60
THEN
CONCAT(CASE WHEN FLOOR(minSekunden)<10 THEN '00:00:0' ELSE '00:00:' END,

```

```
FLOOR(minSekunden))
WHEN FLOOR(minSekunden)>3600
THEN
CONCAT(CONCAT(CONCAT(CONCAT(minH,CASE WHEN FLOOR((FLOOR(minSekunden)-
minH*3600-minSek)/60)<10 THEN ':0' ELSE ':' END),FLOOR((FLOOR(minSekunden)-
minH*3600-minSek)/60)),CASE WHEN minSek<10 THEN ':0' ELSE ':' END) ,minSek)
ELSE
CONCAT(CONCAT(CONCAT(CASE WHEN FLOOR(minSekunden/60)<10 THEN '00:0' ELSE
'00:' END,FLOOR(minSekunden/60)),CASE WHEN FLOOR(minSek)<10 THEN ':0' ELSE
':' END),FLOOR(minSek))
END
END
AS "Min. DLZ",
CASE WHEN
INSTR(
CASE
WHEN FLOOR(maxSekunden)<60
THEN
CONCAT(CASE WHEN FLOOR(maxSekunden)<10 THEN '00:00:0' ELSE '00:00:' END,
FLOOR(maxSekunden))
WHEN FLOOR(maxSekunden)>3600
THEN
CONCAT(CONCAT(CONCAT(CONCAT(maxH,CASE WHEN FLOOR((FLOOR(maxSekunden)-
maxH*3600-maxSek)/60)<10 THEN ':0' ELSE ':' END),FLOOR((FLOOR(maxSekunden)-
maxH*3600-maxSek)/60)),CASE WHEN maxSek<10 THEN ':0' ELSE ':' END) ,maxSek)
ELSE
CONCAT(CONCAT(CONCAT(CASE WHEN FLOOR(maxSekunden/60)<10 THEN '00:0' ELSE
'00:' END,FLOOR(maxSekunden/60)),CASE WHEN FLOOR(maxSek)<10 THEN ':0' ELSE
':' END),FLOOR(maxSek))
END, '- ',1,1)
>0
THEN '00:00:00'
ELSE
CASE
WHEN FLOOR(maxSekunden)<60
THEN
CONCAT(CASE WHEN FLOOR(maxSekunden)<10 THEN '00:00:0' ELSE '00:00:' END,
FLOOR(maxSekunden))
WHEN FLOOR(maxSekunden)>3600
THEN
CONCAT(CONCAT(CONCAT(CONCAT(maxH,CASE WHEN FLOOR((FLOOR(maxSekunden)-
maxH*3600-maxSek)/60)<10 THEN ':0' ELSE ':' END),FLOOR((FLOOR(maxSekunden)-
maxH*3600-maxSek)/60)),CASE WHEN maxSek<10 THEN ':0' ELSE ':' END) ,maxSek)
ELSE
CONCAT(CONCAT(CONCAT(CASE WHEN FLOOR(maxSekunden/60)<10 THEN '00:0' ELSE
'00:' END,FLOOR(maxSekunden/60)),CASE WHEN FLOOR(maxSek)<10 THEN ':0' ELSE
':' END),FLOOR(maxSek))
END
END
AS "Max. DLZ"
```

```

FROM
(
SELECT name, minSekunden, minSek, minH, maxSekunden, maxSek, maxH,
avgSekunden, avgSek, avgH
FROM
(
SELECT FLOOR(MIN((
((to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
))) AS minSekunden,
definitionname AS name
FROM view_instance inst
WHERE inst.END IS NOT NULL
GROUP BY definitionname
),
(
SELECT FLOOR(MOD(MIN((
((to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
)),60)) AS minSek,
definitionname AS name2
FROM view_instance inst
WHERE inst.END IS NOT NULL
GROUP BY definitionname
),
(
SELECT FLOOR(MIN((
((to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
)/3600)) AS minH,
definitionname AS name3
FROM view_instance inst
WHERE inst.END IS NOT NULL
GROUP BY definitionname
),
(SELECT FLOOR(MAX((
((to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)),

```

```
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
))) AS maxSekunden,
definitionname AS name4
FROM view_instance inst
WHERE inst.END IS NOT NULL
GROUP BY definitionname
),
(
SELECT FLOOR(MOD(MAX((
((to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1))), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1))),
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
)),60)) AS maxSek,
definitionname AS name5
FROM view_instance inst
WHERE inst.END IS NOT NULL
GROUP BY definitionname
),
(
SELECT FLOOR(MAX((
((to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1))), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1))),
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
))/3600)) AS maxH,
definitionname AS name6
FROM view_instance inst
WHERE inst.END IS NOT NULL
GROUP BY definitionname
),
(
SELECT FLOOR(avg((
((to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1))), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1))),
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
))) AS avgSekunden,
definitionname AS name7
FROM view_instance inst
WHERE inst.END IS NOT NULL
GROUP BY definitionname
),
(
SELECT FLOOR(MOD(avg((
((to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1))), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
```

```

-
((to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
)),60)) AS avgSek,
definitionname AS name8
FROM view_instance inst
WHERE inst.END IS NOT NULL
GROUP BY definitionname
),
(
SELECT FLOOR(avg((
((to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
))/3600) AS avgH,
definitionname AS name9
FROM view_instance inst
WHERE inst.END IS NOT NULL
GROUP BY definitionname
)
WHERE name = name2
AND name = name3
AND name = name4
AND name = name5
AND name = name6
AND name = name7
AND name = name8
AND name = name9
)[]

```

## Minimale, maximale und durchschnittliche Laufzeit (in h) der Instanzen je Prozessdefinition

```

SELECT definitionname AS Prozess,
CAST((avg((((to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-
MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24) -
((to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24) ))/3600) AS
DECIMAL(10,1)) AS "Durchschnittliche Durchlaufszeit",
CAST((MIN((((to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-
MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24) -
((to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24))/3600)) AS
DECIMAL(10,1)) "Minimale Durchlaufszeit",
CAST((MAX((((to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-
MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24) -
((to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)),

```

```
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24))/3600)) AS  
DECIMAL(10,1)) AS "Maximale Durchlaufszeit"  
FROM view_instance inst  
WHERE inst.END IS NOT NULL AND (((to_date(SUBSTR(inst.END,0,(INSTR(inst.END,  
'.', -1)-1)), 'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)  
-((to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.',  
-1)-1)), 'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)) > 0  
GROUP BY definitionname
```

## Anzahl gestarteter Instanzen je Monat für eine Definition

```
SELECT  
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.  
creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3),'  
' ),EXTRACT(YEAR FROM  
to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)),  
'DD-MM-YY HH24:MI:SS')))) AS Monat,  
SUM(  
CASE  
WHEN inst.definitionname = 'PROZESSDEFINITIONSNAME'  
THEN 1  
ELSE 0  
END) AS "Anzahl"  
FROM view_instance inst WHERE inst.definitionname =  
'PROZESSDEFINITIONSNAME'  
GROUP BY  
EXTRACT(YEAR FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.',  
-1)-1)), 'DD-MM-YY HH24:MI:SS')),  
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.  
creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3),'  
' ),EXTRACT(YEAR FROM  
to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)),  
'DD-MM-YY HH24:MI:SS'))))
```

## Anzahl gestarteter Instanzen je Monat für 2 Definitionen

```
SELECT  
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.  
creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3),'  
' ),EXTRACT(YEAR FROM  
to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)),  
'DD-MM-YY HH24:MI:SS')))) AS Monat,  
SUM(  
CASE  
WHEN inst.definitionname = 'NAME DER PROZESSDEFINITION 1'  
THEN 1  
ELSE 0
```

```

END) AS "Count Prozessdefinition 1",
SUM(
CASE
WHEN inst.definitionname = 'NAME DER PROZESSDEFINITION 2'
THEN 1
ELSE 0
END) AS "Count Prozessdefinition 2"
FROM view_instance inst
WHERE inst.definitionname = 'NAME DER PROZESSDEFINITION 1'
OR inst.definitionname = 'NAME DER PROZESSDEFINITION 2'
GROUP BY
EXTRACT(YEAR FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS')),
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS')), 'Month'),1,3), ' '),EXTRACT(YEAR FROM
to_date(SUBSTR(inst.creationtime,0,(INSTR(inst.creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'))),
EXTRACT(YEAR FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS')),
EXTRACT(MONTH FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'))
ORDER BY
EXTRACT(YEAR FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS')),
EXTRACT(MONTH FROM to_date(SUBSTR(creationtime,0,(INSTR(creationtime, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'))

```

## Anzahl beendeter Instanzen je Monat für eine Definition

```

SELECT
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS')), 'Month'),1,3), ' '),EXTRACT(YEAR FROM
to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS')) AS Monat,
SUM(
CASE
WHEN inst.definitionname = 'PROZESSDEFINITIONSNAME'
THEN 1
ELSE 0
END) AS "PROZESSDEFINITIONSNAME Count"
FROM view_instance inst WHERE inst.END IS NOT NULL AND
inst.definitionname = 'PROZESSDEFINITIONSNAME'
GROUP BY
EXTRACT(YEAR FROM to_date(SUBSTR(END,0,(INSTR(END, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS')),
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS')), 'Month'),1,3), ' '),EXTRACT(YEAR FROM
to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-MM-YY

```

HH24:MI:SS' )))

## Anzahl beendeter Instanzen je Monat für 2 Definitionen

```
SELECT
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3), ' '),EXTRACT(YEAR FROM
to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS' ))) AS Monat,
SUM(
CASE
WHEN inst.definitionname ='PROZESSDEFINITIONSNAME 1'
THEN 1
ELSE 0
END) AS "PROZESSDEFINITIONSNAME 1 Count",
SUM(
CASE
WHEN inst.definitionname ='PROZESSDEFINITIONSNAME 2'
THEN 1
ELSE 0
END) AS "PROZESSDEFINITIONSNAME 2 Count"
FROM view_instance inst WHERE inst.END IS NOT NULL
AND (inst.definitionname = 'S-AP-G' OR inst.definitionname = 'S-
AP-010-010_1')
GROUP BY
EXTRACT(YEAR FROM to_date(SUBSTR(END,0,(INSTR(END, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS' )),
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3), ' '),EXTRACT(YEAR FROM
to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS' ))) ,
EXTRACT(YEAR FROM to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS' )),
EXTRACT(MONTH FROM to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS' ))
ORDER BY
EXTRACT(YEAR FROM to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS' )),
EXTRACT(MONTH FROM to_date(SUBSTR(inst.END,0,(INSTR(inst.END, '.', -1)-1)),
'DD-MM-YY HH24:MI:SS' ))
```

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:instance\\_analyses\\_oracle](https://wiki.tim-solutions.de/doku.php?id=software:dashboard:analyses:instance_analyses_oracle)

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

