

Aktivitätenbezogene Auswertungen

Anzahl an erstellten, beendeten und offenen Aktivitäten je Prozessdefinition

```
SELECT inst.definitionname AS Prozess,
COUNT(DISTINCT(act1.id)) AS "Anzahl erstellte Aktivitäten",
COUNT(DISTINCT(act2.id)) AS "Anzahl beendete Aktivitäten",
COUNT(DISTINCT(act3.id)) AS "Anzahl offene Aktivitäten"
FROM view_activity act1 INNER JOIN view_instance inst ON
act1.instanceid=inst.id AND act1."start" IS NOT NULL AND act1."type" IN
('K','S')
LEFT JOIN view_activity act2 ON act1.id=act2.id AND act2."end" IS NOT NULL
LEFT JOIN view_activity act3 ON act1.id=act3.id AND act3."end" IS NULL AND
act3."start" IS NOT NULL
GROUP BY inst.definitionname
```

Anzahl an erstellten, beendeten und offenen Aktivitäten einer bestimmten Prozessdefinition

```
SELECT inst.definitionname AS Prozess,
COUNT(DISTINCT(act1.id)) AS "Anzahl erstellte Aktivitäten",
COUNT(DISTINCT(act2.id)) AS "Anzahl beendete Aktivitäten",
COUNT(DISTINCT(act3.id)) AS "Anzahl offene Aktivitäten"
FROM view_activity act1 INNER JOIN view_instance inst ON
act1.instanceid=inst.id AND act1."start" IS NOT NULL AND act1."type" IN
('K','S') AND inst.definitionName = 'PROZESSDEFINITIONSNAME'
LEFT JOIN view_activity act2 ON act1.id=act2.id AND act2."end" IS NOT NULL
LEFT JOIN view_activity act3 ON act1.id=act3.id AND act3."end" IS NULL AND
act3."start" IS NOT NULL
GROUP BY inst.definitionname
```

Durchschnittliche Zeit (in hh:mm:ss) von Erstellung bis Erledigung von allen Aktivitäten (egal von welcher Prozessdefinition und auch außerhalb der Arbeitszeit)

```
SELECT
CASE
WHEN FLOOR(sekunden)<60
THEN
CONCAT(CASE WHEN FLOOR(sekunden)<10 THEN '00:00:0' ELSE '00:00:'
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. LZ v. Aufg."
FROM (
SELECT
FLOOR(AVG((TO_NUMBER (TO_CHAR ("end", 'J')) - TO_NUMBER (TO_CHAR ("start",
'J')) * 86400 + ( TO_NUMBER (TO_CHAR ("end", 'SSSSS')) - TO_NUMBER (TO_CHAR
("start", 'SSSSS'))))) AS sekunden,
FLOOR(MOD(AVG((TO_NUMBER (TO_CHAR ("end", 'J')) - TO_NUMBER (TO_CHAR
("start", 'J')) * 86400 + ( TO_NUMBER (TO_CHAR ("end", 'SSSSS')) -
TO_NUMBER (TO_CHAR ("start", 'SSSSS')))),60)) AS sek,
FLOOR(AVG((TO_NUMBER (TO_CHAR ("end", 'J')) - TO_NUMBER (TO_CHAR ("start",
'J')) * 86400 + ( TO_NUMBER (TO_CHAR ("end", 'SSSSS')) - TO_NUMBER (TO_CHAR
("start", 'SSSSS'))))/3600) AS h
FROM view_activity
WHERE "end" IS NOT NULL
AND "start" IS NOT NULL
AND "type" IN ('K','S')
) []
```

Durchschnittliche Zeit (in hh:mm:ss) von Erstellung bis Erledigung von allen Aktivitäten je Monat (egal von welcher Prozessdefinition und auch außerhalb der Arbeitszeit) je Monat

```
SELECT
Monat,
CASE
WHEN FLOOR(sekunden)<60
THEN
CONCAT(CASE WHEN FLOOR(sekunden)<10 THEN '00:00:0' ELSE '00:00:'
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. LZ v. Aktivitäten."
FROM
(
SELECT
```

```

CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR("start",0,(INSTR("start", '.', -1)-1)),
-1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3), ' '),EXTRACT(YEAR FROM
to_date(SUBSTR("start",0,(INSTR("start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS')) AS Monat,
FLOOR(AVG((TO_NUMBER (TO_CHAR ("end", 'J')) - TO_NUMBER (TO_CHAR ("start",
'J')) * 86400 + ( TO_NUMBER (TO_CHAR ("end", 'SSSSS')) - TO_NUMBER (TO_CHAR
("start", 'SSSSS'))))) AS sekunden,
FLOOR(MOD(AVG((TO_NUMBER (TO_CHAR ("end", 'J')) - TO_NUMBER (TO_CHAR
("start", 'J')) * 86400 + ( TO_NUMBER (TO_CHAR ("end", 'SSSSS')) -
TO_NUMBER (TO_CHAR ("start", 'SSSSS'))),60)) AS sek,
FLOOR(AVG((TO_NUMBER (TO_CHAR ("end", 'J')) - TO_NUMBER (TO_CHAR ("start",
'J')) * 86400 + ( TO_NUMBER (TO_CHAR ("end", 'SSSSS')) - TO_NUMBER (TO_CHAR
("start", 'SSSSS')))/3600) AS h,
EXTRACT(YEAR FROM to_date(SUBSTR("start",0,(INSTR("start", '.', -1)-1)),
'DD-MM-YY HH24:MI:SS')) AS j,
EXTRACT(MONTH FROM to_date(SUBSTR("start",0,(INSTR("start", '.', -1)-1)),
'DD-MM-YY HH24:MI:SS')) AS m
FROM view_activity
WHERE "end" IS NOT NULL
AND "start" IS NOT NULL
AND "type" IN ('K','S')
GROUP BY
EXTRACT(YEAR FROM to_date(SUBSTR("start",0,(INSTR("start", '.', -1)-1)),
'DD-MM-YY HH24:MI:SS')),
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR("start",0,(INSTR("start", '.', -1)-1)),
-1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3), ' '),EXTRACT(YEAR FROM
to_date(SUBSTR("start",0,(INSTR("start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS')),
EXTRACT(YEAR FROM to_date(SUBSTR("start",0,(INSTR("start", '.', -1)-1)),
'DD-MM-YY HH24:MI:SS')),
EXTRACT(MONTH FROM to_date(SUBSTR("start",0,(INSTR("start", '.', -1)-1)),
'DD-MM-YY HH24:MI:SS'))
)
ORDER BY j, m

```

Durchschnittliche Zeit (in hh:mm:ss) von Erstellung bis Erledigung von allen Aktivitäten je Monat ohne Ausreißer (von allen Prozessdefinitionen und auch außerhalb der Arbeitszeit)

```

SELECT
Monat,
CASE
WHEN FLOOR(sekunden)<60
THEN
CONCAT(CASE WHEN FLOOR(sekunden)<10 THEN '00:00:0' ELSE '00:00:'
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. LZ v. Aktivitäten."
FROM
(
SELECT
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR("start",0,(INSTR("start", '.',
-1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3), ' '),EXTRACT(YEAR FROM
to_date(SUBSTR("start",0,(INSTR("start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS')) AS Monat,
FLOOR(AVG((TO_NUMBER (TO_CHAR ("end", 'J')) - TO_NUMBER (TO_CHAR ("start",
'J')) * 86400 + ( TO_NUMBER (TO_CHAR ("end", 'SSSSS')) - TO_NUMBER (TO_CHAR
("start", 'SSSSS'))))) AS sekunden,
FLOOR(MOD(AVG((TO_NUMBER (TO_CHAR ("end", 'J')) - TO_NUMBER (TO_CHAR
("start", 'J')) * 86400 + ( TO_NUMBER (TO_CHAR ("end", 'SSSSS')) -
TO_NUMBER (TO_CHAR ("start", 'SSSSS'))),60)) AS sek,
FLOOR(AVG((TO_NUMBER (TO_CHAR ("end", 'J')) - TO_NUMBER (TO_CHAR ("start",
'J')) * 86400 + ( TO_NUMBER (TO_CHAR ("end", 'SSSSS')) - TO_NUMBER (TO_CHAR
("start", 'SSSSS')))/3600) AS h
FROM view_activity
WHERE "start" IS NOT NULL
AND "end" IS NOT NULL
AND "type" IN ('K','S')
AND
(((to_date(SUBSTR("end",0,(INSTR("end", '.', -1)-1)), 'DD-MM-YY HH24:MI:SS')
- DATE '1970-01-01' ) * 60 * 60 * 24) -
((to_date(SUBSTR("start",0,(INSTR("start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)) < 10000
GROUP BY
EXTRACT(YEAR FROM to_date(SUBSTR("start",0,(INSTR("start", '.', -1)-1)),
'DD-MM-YY HH24:MI:SS')),
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR("start",0,(INSTR("start", '.',
-1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3), ' '),EXTRACT(YEAR FROM
to_date(SUBSTR("start",0,(INSTR("start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS'))))
)
```

Durchschnittliche, minimale und maximale Durchlaufszeit aller Aktivitäten einer bestimmten Prozessdefinition (in hh:mm:ss)

```
SELECT
name1 AS "Aktivität",
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
AS "Durschn. DLZ",

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 AS "Min. DLZ",

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 AS "Max. DLZ"

FROM
(

```

```
SELECT name1, minSekunden, minSek, minH, maxSekunden, maxSek, maxH,
avgSekunden, avgSek, avgH
FROM
(
SELECT FLOOR(MIN((
((to_date(SUBSTR(ta."end",0,(INSTR(ta."end", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(ta."start",0,(INSTR(ta."start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
))) AS minSekunden,
ta."name" AS name1
FROM view_activity ta, view_instance vi
WHERE vi.definitionName = 'PROZESSDEFINITIONSNAME'
AND ta."type" IN ('K','S')
AND vi.id = ta.instanceId
AND ta."end" IS NOT NULL
GROUP BY ta."name"
),
(
SELECT FLOOR(MOD(MIN((
((to_date(SUBSTR(ta."end",0,(INSTR(ta."end", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(ta."start",0,(INSTR(ta."start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
)),60)) AS minSek,
ta."name" AS name2
FROM view_activity ta, view_instance vi
WHERE vi.definitionName = 'PROZESSDEFINITIONSNAME'
AND ta."type" IN ('K','S')
AND vi.id = ta.instanceId
AND ta."end" IS NOT NULL
GROUP BY ta."name"
),
(
SELECT FLOOR(MIN((
((to_date(SUBSTR(ta."end",0,(INSTR(ta."end", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(ta."start",0,(INSTR(ta."start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
))/3600)) AS minH,
ta."name" AS name3
FROM view_activity ta, view_instance vi
WHERE vi.definitionName = 'PROZESSDEFINITIONSNAME'
AND ta."type" IN ('K','S')
AND vi.id = ta.instanceId
AND ta."end" IS NOT NULL
GROUP BY ta."name"
```

```

),
(
SELECT FLOOR(MAX((
((to_date(SUBSTR(ta."end",0,(INSTR(ta."end", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(ta."start",0,(INSTR(ta."start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
))) AS maxSekunden,
ta."name" AS name4
FROM view_activity ta, view_instance vi
WHERE vi.definitionName = 'PROZESSDEFINITIONSNAME'
AND ta."type" IN ('K','S')
AND vi.id = ta.instanceId
AND ta."end" IS NOT NULL
GROUP BY ta."name"
),
(
SELECT FLOOR(MOD(MAX((
((to_date(SUBSTR(ta."end",0,(INSTR(ta."end", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(ta."start",0,(INSTR(ta."start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
)),60)) AS maxSek,
ta."name" AS name5
FROM view_activity ta, view_instance vi
WHERE vi.definitionName = 'PROZESSDEFINITIONSNAME'
AND ta."type" IN ('K','S')
AND vi.id = ta.instanceId
AND ta."end" IS NOT NULL
GROUP BY ta."name"
),
(
SELECT FLOOR(MAX((
((to_date(SUBSTR(ta."end",0,(INSTR(ta."end", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(ta."start",0,(INSTR(ta."start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
)/3600)) AS maxH,
ta."name" AS name6
FROM view_activity ta, view_instance vi
WHERE vi.definitionName = 'PROZESSDEFINITIONSNAME'
AND ta."type" IN ('K','S')
AND vi.id = ta.instanceId
AND ta."end" IS NOT NULL
GROUP BY ta."name"
),
(
SELECT FLOOR(avg((

```

```
((to_date(SUBSTR(ta."end",0,(INSTR(ta."end", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(ta."start",0,(INSTR(ta."start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
))) AS avgSekunden,
ta."name" AS name7
FROM view_activity ta, view_instance vi
WHERE vi.definitionName = 'PROZESSDEFINITIONSNAME'
AND ta."type" IN ('K','S')
AND vi.id = ta.instanceId
AND ta."end" IS NOT NULL
GROUP BY ta."name"
),
(
SELECT FLOOR(MOD(avg((
((to_date(SUBSTR(ta."end",0,(INSTR(ta."end", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(ta."start",0,(INSTR(ta."start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
)),60)) AS avgSek,
ta."name" AS name8
FROM view_activity ta, view_instance vi
WHERE vi.definitionName = 'PROZESSDEFINITIONSNAME'
AND ta."type" IN ('K','S')
AND vi.id = ta.instanceId
AND ta."end" IS NOT NULL
GROUP BY ta."name"
),
(
SELECT FLOOR(avg((
((to_date(SUBSTR(ta."end",0,(INSTR(ta."end", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
-
((to_date(SUBSTR(ta."start",0,(INSTR(ta."start", '.', -1)-1)), 'DD-MM-YY
HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)
))/3600) AS avgH,
ta."name" AS name9
FROM view_activity ta, view_instance vi
WHERE vi.definitionName = 'PROZESSDEFINITIONSNAME'
AND ta."type" IN ('K','S')
AND vi.id = ta.instanceId
AND ta."end" IS NOT NULL
GROUP BY ta."name"
)
WHERE name1 = name2
AND name1 = name3
AND name1 = name4
AND name1 = name5
```



```

AND name1 = name6
AND name1 = name7
AND name1 = name8
AND name1 = name9
)

```

Durchschnittliche, minimale und maximale Durchlaufszeit aller Aktivitäten einer bestimmten Prozessdefinition (in h)

```

SELECT
activity.name AS Aufgabe,
CAST((avg((((to_date(SUBSTR(activity.end,0,(INSTR(activity.end, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24) -
((to_date(SUBSTR(activity."start",0,(INSTR(activity."start", '.', -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(activity.end,0,(INSTR(activity.end, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24) -
((to_date(SUBSTR(activity."start",0,(INSTR(activity."start", '.', -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(activity.end,0,(INSTR(activity.end, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24) -
((to_date(SUBSTR(activity."start",0,(INSTR(activity."start", '.', -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_activity activity
INNER JOIN
view_instance inst ON activity.instanceId=inst.id
WHERE activity.end IS NOT NULL
AND activity."start" IS NOT NULL
AND activity.type IN ('K','S')
AND (((to_date(SUBSTR(activity.end,0,(INSTR(activity.end, '.', -1)-1)), 'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24) -
((to_date(SUBSTR(activity."start",0,(INSTR(activity."start", '.', -1)-1)), 'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)) > 0
AND inst.definitionname LIKE 'GL-01%'
GROUP BY
activity.name
ORDER BY activity.name

```

Durchschnittliche, minimale und maximale Durchlaufszeit aller Aktivitäten einer bestimmten Prozessdefinition (in h) je Monat

```

SELECT
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(activity."start",0,(INSTR(activity."start", '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3),',
'),EXTRACT(YEAR FROM

```

```
to_date(SUBSTR(activity."start",0,(INSTR(activity."start", '.', -1)-1)),
'DD-MM-YY HH24:MI:SS')) AS Monat,
activity."name" AS Aufgabe,
CAST((avg((((to_date(SUBSTR(activity."end",0,(INSTR(activity."end", '.',
-1)-1)), 'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24) -
((to_date(SUBSTR(activity."start",0,(INSTR(activity."start", '.', -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(activity."end",0,(INSTR(activity."end", '.',
-1)-1)), 'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24) -
((to_date(SUBSTR(activity."start",0,(INSTR(activity."start", '.', -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(activity."end",0,(INSTR(activity."end", '.',
-1)-1)), 'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24) -
((to_date(SUBSTR(activity."start",0,(INSTR(activity."start", '.', -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_activity activity
INNER JOIN
view_instance inst ON activity.instanceId=inst.id
WHERE activity."end" IS NOT NULL
AND activity."start" IS NOT NULL
AND activity."type" IN ('K','S')
AND (((to_date(SUBSTR(activity."end",0,(INSTR(activity."end", '.', -1)-1)),
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24) -
((to_date(SUBSTR(activity."start",0,(INSTR(activity."start", '.', -1)-1)),
'DD-MM-YY HH24:MI:SS') - DATE '1970-01-01' ) * 60 * 60 * 24)) > 0
AND inst.definitionname = 'PROZESSDEFINITIONSNAME'
GROUP BY
EXTRACT(YEAR FROM to_date(SUBSTR(activity."start",0,(INSTR(activity."start",
 '.', -1)-1)), 'DD-MM-YY HH24:MI:SS')),
CONCAT(CONCAT(SUBSTR(to_char(to_date(SUBSTR(activity."start",0,(INSTR(activi
ty."start", '.', -1)-1)), 'DD-MM-YY HH24:MI:SS'), 'Month'),1,3), '
'),EXTRACT(YEAR FROM
to_date(SUBSTR(activity."start",0,(INSTR(activity."start", '.', -1)-1)),
'DD-MM-YY HH24:MI:SS'))),
activity."name"
```

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:activity_analyses_oracle

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

