

Aufgabenbezogenen Auswertungen

Erledigte Aufgaben pro Monat

```
SELECT concat(SUBSTRING(CAST(YEAR(task.start) AS CHAR),3,2), " ",  
MONTH(task.start)) AS monat, COUNT(*) AS "Summe erledigte Aufgaben"  
FROM view_task task WHERE task.end IS NOT NULL AND task.start IS NOT NULL  
GROUP BY monat
```

Erledigte Aufgaben pro Jahr

```
SELECT YEAR(task.start) AS jahr, COUNT(*) AS "Summe erledigte Aufgaben"  
FROM view_task task WHERE task.end IS NOT NULL AND task.start IS NOT NULL  
GROUP BY jahr
```

Alle offenen Aufgaben eines bestimmten Users

```
SELECT i.lastname AS Nachname, i.firstname AS Vorname, i.name AS Username,  
t.name AS Task, pi.name AS Instanzname, pi.definitionName AS Prozessname  
FROM view_activity a, view_task t, view_identity i, view_instance pi  
WHERE a.id = t.activity AND t.actor = i.id AND a.end IS NULL AND t.end IS  
NULL AND pi.id = a.instanceId AND pi.archiv = 0 AND pi.end IS NULL AND  
i.lastname = "Barth"
```

Anzahl an erstellten, erledigten und offenen Aufgaben je Prozessdefinition

```
SELECT inst.definitionname AS Prozess,  
COUNT(t1.id) AS "Anzahl erstellte Aufgaben",  
COUNT(t2.id) AS "Anzahl erledigte Aufgaben",  
COUNT(t3.id) AS "Anzahl offene Aufgaben"  
FROM view_task t1 INNER JOIN view_activity act ON t1.activity=act.id  
INNER JOIN view_instance inst ON t1.instanceid = inst.id  
LEFT JOIN view_task t2 ON t1.id=t2.id AND (t2.end IS NOT NULL OR act.END IS  
NOT NULL)  
LEFT JOIN view_task t3 ON t1.id=t3.id AND t3.end IS NULL AND inst.end IS  
NULL AND inst.archiv IS FALSE  
GROUP BY inst.definitionname
```

Anzahl an erstellten, erledigten und offenen Aufgaben einer bestimmten Prozessdefinition

```
SELECT inst.definitionname AS Prozess,  
COUNT(t1.id) AS "Anzahl erstellte Aufgaben",  
COUNT(t2.id) AS "Anzahl erledigte Aufgaben",
```

```
COUNT(t3.id) AS "Anzahl offene Aufgaben"
FROM view_task t1 INNER JOIN view_activity act ON t1.activity=act.id INNER
JOIN view_instance inst ON t1.instanceid = inst.id
LEFT JOIN view_task t2 ON t1.id=t2.id AND (t2.end IS NOT NULL OR act.END IS
NOT NULL)
LEFT JOIN view_task t3 ON t1.id=t3.id AND t3.end IS NULL AND inst.end IS
NULL AND inst.archiv IS FALSE
WHERE inst.definitionname ="Name der Prozessdefinition"
```

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

```
SELECT IF((FLOOR(sekunden))<60,
concat(IF((FLOOR(sekunden))<10,"00:00:0","00:00:"),(FLOOR(sekunden))),
IF((FLOOR(sekunden))>3600,
CAST(concat(h,IF(FLOOR((FLOOR(sekunden)- h*3600-
sek)/60)<10,":0",":"),FLOOR((FLOOR(sekunden)- h*3600-
sek)/60),IF(sek<10,":0",":"),sek) AS CHAR),
concat(IF(FLOOR(sekunden/60)<10,"00:0","00:"),FLOOR(sekunden/60),IF(FLOOR(sek)
<10,":0",":"),FLOOR(sek))
)
) AS "Durchschnittliche Liegezeit von Aufgaben" FROM (SELECT
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))) AS sekunden,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))%60) AS sek,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START))/3600) AS h
FROM view_task WHERE END IS NOT NULL) AS dusub
```

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

```
SELECT Monat, IF((FLOOR(sekunden))<60,
concat(IF((FLOOR(sekunden))<10,"00:00:0","00:00:"),(FLOOR(sekunden))),
IF((FLOOR(sekunden))>3600,
CAST(concat(h,IF(FLOOR((FLOOR(sekunden)- h*3600-
sek)/60)<10,":0",":"),FLOOR((FLOOR(sekunden)- h*3600-
sek)/60),IF(sek<10,":0",":"),sek) AS CHAR),
concat(IF(FLOOR(sekunden/60)<10,"00:0","00:"),FLOOR(sekunden/60),IF(FLOOR(sek)
<10,":0",":"),FLOOR(sek))
)
) AS "Durchschnittliche Liegezeit von Aufgaben" FROM
(SELECT concat(SUBSTRING(monthname(START),1,3),"
",SUBSTRING(CAST(YEAR(START) AS CHAR),3,2)) AS Monat,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))) AS sekunden,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))%60) AS sek,
```

```
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))/3600) AS h FROM
view_task
WHERE isOpen = "0" GROUP BY YEAR(START), MONTH(START)) AS dusub
```

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

```
SELECT Monat, IF((FLOOR(sekunden))<60,
  concat(IF((FLOOR(sekunden))<10,"00:00:0","00:00:"),(FLOOR(sekunden))),
  IF((FLOOR(sekunden))>3600,
    CAST(concat(h,IF(FLOOR((FLOOR(sekunden)-h*3600-
sek)/60)<10,":0",":"),FLOOR((FLOOR(sekunden)-h*3600-
sek)/60),IF(sek<10,":0",":"),sek) AS CHAR),
concat(IF(FLOOR(sekunden/60)<10,"00:0","00:"),FLOOR(sekunden/60),IF(FLOOR(sek
k)<10,":0",":"),FLOOR(sek))
  )
) AS "Durchschnittliche Liegezeit von Aufgaben" FROM
(SELECT concat(SUBSTRING(monthname(START),1,3),"
",SUBSTRING(CAST(YEAR(START) AS CHAR),3,2)) AS Monat,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))) AS sekunden,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START))%60)) AS sek,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))/3600) AS h FROM
view_task
WHERE isOpen = "0" AND CAST(FLOOR((unix_timestamp(END) -
unix_timestamp(START))) AS DECIMAL(10,10)) < 10000 GROUP BY YEAR(START),
MONTH(START)) AS dusub
```

Alle offenen Aufgaben mit aktiver Instanz (nicht beendet und nicht archiviert) und aktiver Aktivität (nicht beendet und nicht archiviert) mit Bearbeiter

```
SELECT tsk.name AS Aufgabe, IF (concat(ident.lastname, ", ",ident.firstname)
IS NOT NULL,
concat(ident.lastname, ", ",ident.firstname), grouped.name) AS 'Bearbeiter',
inst.name AS 'Instanz', def.name AS 'Prozess', tsk.creationTime AS
"Erstellzeit"
FROM view_task tsk
LEFT JOIN view_identity ident ON tsk.actor = ident.id
LEFT JOIN view_identity grouped ON tsk.pooledActor = grouped.id
INNER JOIN view_instance inst ON tsk.instanceId = inst.id
INNER JOIN view_definition def ON inst.definitionId = def.id
INNER JOIN view_activity act ON tsk.activity = act.id
WHERE inst.archiv = 'false' AND tsk.isOpen = 1
```

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

```
SELECT name AS Aufgabe,
IF((FLOOR(avgSekunden))<60,
concat(IF((FLOOR(avgSekunden))<10,"00:00:0","00:00:"),(FLOOR(avgSekunden))),
IF((FLOOR(avgSekunden))>3600,
CAST(concat(avgH,IF(FLOOR((FLOOR(avgSekunden)- avgH*3600-
avgSek)/60)<10,":0",":"),FLOOR((FLOOR(avgSekunden)- avgH*3600-
avgSek)/60),IF(avgSek<10,":0",":"),avgSek) AS CHAR),
concat(IF(FLOOR(avgSekunden/60)<10,"00:0","00:"),FLOOR(avgSekunden/60),IF(FLOOR(avgSek)<10,":0",":"),FLOOR(avgSek))
)
)AS "Durchschnittliche Durchlaufzeit",
IF((FLOOR(minSekunden))<60,
concat(IF((FLOOR(minSekunden))<10,"00:00:0","00:00:"),(FLOOR(minSekunden))),
IF((FLOOR(minSekunden))>3600,
CAST(concat(minH,IF(FLOOR((FLOOR(minSekunden)- minH*3600-
minSek)/60)<10,":0",":"),FLOOR((FLOOR(minSekunden)- minH*3600-
minSek)/60),IF(minSek<10,":0",":"),minSek) AS CHAR),
concat(IF(FLOOR(minSekunden/60)<10,"00:0","00:"),FLOOR(minSekunden/60),IF(FLOOR(minSek)<10,":0",":"),FLOOR(minSek))
)
) AS "Minimale Durchlaufzeit",
IF((FLOOR(maxSekunden))<60,
concat(IF((FLOOR(maxSekunden))<10,"00:00:0","00:00:"),(FLOOR(maxSekunden))),
IF((FLOOR(maxSekunden))>3600,
CAST(concat(maxH,IF(FLOOR((FLOOR(maxSekunden)- maxH*3600-
maxSek)/60)<10,":0",":"),FLOOR((FLOOR(maxSekunden)- maxH*3600-
maxSek)/60),IF(maxSek<10,":0",":"),maxSek) AS CHAR),
concat(IF(FLOOR(maxSekunden/60)<10,"00:0","00:"),FLOOR(maxSekunden/60),IF(FLOOR(maxSek)<10,":0",":"),FLOOR(maxSek))
)
) AS "Maximale Durchlaufzeit"
FROM
(SELECT inst.definitionname AS defName, task.name AS name,
FLOOR(MIN((unix_timestamp(task.END) - unix_timestamp(task.START)))) AS
minSekunden,
FLOOR(MIN((unix_timestamp(task.END) - unix_timestamp(task.START))%60) AS
minSek,
FLOOR(MIN((unix_timestamp(task.END) - unix_timestamp(task.START))/3600)) AS
minH,
FLOOR(MAX((unix_timestamp(task.END) - unix_timestamp(task.START)))) AS
maxSekunden,
FLOOR(MAX((unix_timestamp(task.END) - unix_timestamp(task.START))%60) AS
maxSek,
FLOOR(MAX((unix_timestamp(task.END) - unix_timestamp(task.START))/3600)) AS
maxH,
```

```

FLOOR(avg((unix_timestamp(task.END) - unix_timestamp(task.START)))) AS
avgSekunden,
FLOOR(avg((unix_timestamp(task.END) - unix_timestamp(task.START))%60) AS
avgSek,
FLOOR(avg((unix_timestamp(task.END) - unix_timestamp(task.START))/3600)) AS
avgH,
(unix_timestamp(task.END)-unix_timestamp(task.START)) AS datediff
FROM view_task task INNER JOIN view_instance inst ON task.instanceId=inst.id
WHERE task.END IS NOT NULL AND task.START IS NOT NULL AND
inst.definitionname = "Name der Prozessdefinition"
GROUP BY task.name
) AS datediffsub
WHERE datediff > 0
GROUP BY name

```

Durchschnittliche, minimale und maximale Durchlaufzeit aller Aufgaben einer bestimmten Prozessdefinition (in h)

```

SELECT task.name AS Aufgabe,
CAST((MIN((unix_timestamp(task.END) - unix_timestamp(task.START))/3600)) AS
DECIMAL(10,1)) AS "Minimale Durchlaufzeit",
CAST((avg((unix_timestamp(task.END) - unix_timestamp(task.START))/3600)) AS
DECIMAL(10,1)) AS "Durchschnittliche Durchlaufzeit",
CAST((MAX((unix_timestamp(task.END) - unix_timestamp(task.START))/3600)) AS
DECIMAL(10,1)) AS "Maximale Durchlaufzeit"
FROM view_task task INNER JOIN view_instance inst ON task.instanceId=inst.id
WHERE task.END IS NOT NULL AND task.START IS NOT NULL AND
(unix_timestamp(task.END)-unix_timestamp(task.START)) >0
AND inst.definitionname = "Name der Prozessdefinition"
GROUP BY task.name

```

Durchschnittliche, minimale und maximale Durchlaufzeit aller Aufgaben einer bestimmten Prozessdefinition (in h) je Monat

```

SELECT concat(SUBSTRING(monthname(task.START),1,3), "
",SUBSTRING(CAST(YEAR(task.START) AS CHAR),3,2)) AS Monat, task.name AS
Aufgabe,
    CAST(avg((unix_timestamp(task.END) - unix_timestamp(task.START))/3600) AS
DECIMAL(10,1)) AS "Durchschnittliche Durchlaufzeit",
CAST(MIN((unix_timestamp(task.END) - unix_timestamp(task.START))/3600) AS
DECIMAL(10,1)) AS "Minimale Durchlaufzeit",
CAST(MAX((unix_timestamp(task.END) - unix_timestamp(task.START))/3600) AS
DECIMAL(10,1)) AS "Maximale Durchlaufzeit"
FROM view_task task INNER JOIN view_instance inst ON task.instanceId=inst.id
WHERE task.END IS NOT NULL AND task.START IS NOT NULL AND
(unix_timestamp(task.END) - unix_timestamp(task.START)) >0
AND inst.definitionname = "Name der Prozessdefinition"

```

Last
update:
2021/07/01 09:52 software:dashboard:analyses:task_analyses https://wiki.tim-solutions.de/doku.php?id=software:dashboard:analyses:task_analyses

GROUP BY YEAR(task.START) DESC, MONTH(task.START) DESC, task.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:task_analyses

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

