T!M Datenbank Views

Um mit dem Dashboard arbeiten zu können, müssen zuerst die entsprechenden Views angelegt werden, die die T!M Daten aufbereiten und einen einfacheren und schlankeren Zugriff gewähren. Die einzelnen Abschnitte können entweder einfach in den entsprechenden MySQL Client kopiert und ausgeführt werden (Strg + Enter). Alternativ, kann das Script herunter geladen werden und auf dem MySQL Server ausgeführt werden.

tim_views.sql

```
CREATE DEFINER=`root`@`localhost` FUNCTION `currentClient`() RETURNS
INT(11)
    NO SQL
    DETERMINISTIC
RETURN @currentClient;



CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY
DEFINER VIEW `view_activity` AS SELECT
`ni`.`ID_` AS `id`,
`ni`.`NAME_` AS `name`,
`ni`.`client_ID_` AS `clientId`,
`ni`.`CREATION_TIME_` AS `creationTime`,
`ni`.`creationUser_ID_` AS `creationUserId`,
`ni`.`processInstance_ID_` AS `instanceId`,
`ni`.`realStart` AS `start`,
`ni`.`realEnd` AS `end`,
`ni`.`loopCount` AS `loopCount`,
`ni`.`duration` AS `duration`,
`ni`.`escalationTime` AS `escalationTime`,
`ni`.`estimatedEnd` AS `estimatedEnd`,
`ni`.`desiredStart` AS `milestone`,
`ni`.`desiredStartTime` AS `milestoneDate`,
`ni`.`inTime` AS `inTime`,
`ni`.`puffer` AS `timeBuffer`,
`ni`.`calFAZ` AS `calculatedEST`,
`ni`.`calFEZ` AS `calculatedEFT`,
`ni`.`calSAZ` AS `calculatedLST`,
`ni`.`calSEZ` AS `calculatedLFT`,
`ni`.`numFAZ` AS `valueEST`,
`ni`.`numFEZ` AS `valueEFT`,
`ni`.`numSAZ` AS `valueLST`,
`ni`.`numSEZ` AS `valueLFT` ,
`node`.`CLASS_` AS `type`
FROM ((`LOOM_NODEINSTANCE` `ni` JOIN `LOOM_NODE` `node` ON
(`ni`.`node_ID_` = `node`.`ID_`))) WHERE (`ni`.`client_ID_` =
`currentClient`());
```

```
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY
DEFINER VIEW `view_definition` AS SELECT
`LOOM_PROCESSDEFINITION`.`ID_` AS `id`,
`LOOM_PROCESSDEFINITION`.`NAME_` AS `name`,
`LOOM_PROCESSDEFINITION`.`CLIENT_` AS `clientId`,
`LOOM_PROCESSDEFINITION`.`CREATION_TIME_` AS `creationTime`,
`LOOM_PROCESSDEFINITION`.`CREATION_USER_` AS `creationUserId`,
`LOOM_PROCESSDEFINITION`.`DESCRIPTION_` AS `description`,
`LOOM_PROCESSDEFINITION`.`VERSION_` AS `version`,
`LOOM_PROCESSDEFINITION`.`ESCALATIONSTATUS_` AS `escalationStatus`,
`LOOM_PROCESSDEFINITION`.`OWNER_EXPRESSION_` AS `owner`,
`LOOM_PROCESSDEFINITION`.`STARTER_EXPRESSION_` AS `starter`,
`LOOM_PROCESSDEFINITION`.`DEPLOYER_EXPRESSION_` AS `deployer` ,
`LOOM_PROCESSDEFINITION`.`ARCHIV_` AS `archived`
FROM `LOOM_PROCESSDEFINITION` WHERE (`LOOM_PROCESSDEFINITION`.`CLIENT_`
= `currentClient`());



CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY
DEFINER VIEW `view_effort` AS SELECT
`eff`.`ID_` AS `id`,
`eff`.`NAME_` AS `name`,
`eff`.`client_ID_` AS `clientId`,
`eff`.`CREATION_TIME_` AS `creationTime`,
`eff`.`creationUser_ID_` AS `creationUserId`,
`eff`.`CLASS` AS `effortType`,
`eff`.`effort` AS `value`,
`eff`.`description` AS `description`,
`eff`.`costCenter_ID_` AS `costCenterId`,
`ti`.`PROCINST_` AS `instanceId`,
`eff`.`parentFolder_ID_` AS `parentFolderId`
FROM (`LOOM_EFFORT` `eff` JOIN `LOOM_TASKINSTANCE` `ti`
ON((`eff`.`parentFolder_ID_` = `ti`.`PARENT_FOLDER_`)))
WHERE (`eff`.`client_ID_` = `currentClient`());



CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY
DEFINER VIEW `view_identity` AS SELECT
`LOOM_IDENTITY`.`ID_` AS `id`,
`LOOM_IDENTITY`.`NAME_` AS `name`,
concat(`LOOM_IDENTITY`.`namelast`,' ',`LOOM_IDENTITY`.`namefirst`,'
(',`LOOM_IDENTITY`.`NAME_`,')') AS `displayName`,
`LOOM_IDENTITY`.`client_ID_` AS `clientId`,
```

```
`LOOM_IDENTITY`.`CREATION_TIME_` AS `creationTime`,
`LOOM_IDENTITY`.`creationUser_ID_` AS `creationUserId`,
`LOOM_IDENTITY`.`CLASS` AS `identityType`,
`LOOM_IDENTITY`.`email` AS `email`,
`LOOM_IDENTITY`.`namefirst` AS `firstname`,
`LOOM_IDENTITY`.`namelast` AS `lastname` ,
`LOOM_IDENTITY`.`ARCHIV_` AS `archived` ,
`LOOM_IDENTITY`.`blocked` AS `blocked` ,
`LOOM_IDENTITY`.`parent_ID_` AS `parentID`
FROM `LOOM_IDENTITY` WHERE (`LOOM_IDENTITY`.`client_ID_` =
`currentClient`());




CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY
DEFINER VIEW `view_instance` AS SELECT
`pi`.`ID_` AS `id`,
`pi`.`NAME_` AS `name`,
`pd`.`NAME_` AS `definitionName`,
`pi`.`CLIENT_` AS `clientId`,
`pi`.`CREATION_TIME_` AS `creationTime`,
`pi`.`key_` AS `key`,
`pi`.`CREATION_USER_` AS `creationUserId`,
`pi`.`PROCESSDEFINITION_` AS `definitionId`,
`pi`.`DESCRIPTION_` AS `instanceDescription`,
`pi`.`CREATION_GROUP_` AS `creationGroup`,
`pi`.`END_` AS `end`,
`pi`.`ARCHIV_` AS `archiv`,
`pi`.`ARCHIVATIONUSER_ID_` AS `archivationUserId`,
`pi`.`ROOTTOKEN_` AS `rootToken`,
`pi`.`SUPERPROCESSTOKEN_` AS `parentProcessToken`,
`pi`.`NEXT_ESCALATIONTIME_` AS `nextEscalationTime`,
`pi`.`processVariableIndex_ID_` AS `processVariableIndexId`,
`pi`.`ARCHIVATIONUSER_ID_` AS `archivationUserId`,
`pi`.`inTime` AS `inTime`,
`pvi`.`field1` AS `index1`,
`pvi`.`value1` AS `value1`,
`pvi`.`field2` AS `index2`,
`pvi`.`value2` AS `value2`,
`pvi`.`field3` AS `index3`,
`pvi`.`value3` AS `value3`,
`pvi`.`field4` AS `index4`,
`pvi`.`value4` AS `value4`,
`pvi`.`field5` AS `index5`,
`pvi`.`value5` AS `value5`,
`pvi`.`field6` AS `index6`,
`pvi`.`value6` AS `value6`,
`pvi`.`field7` AS `index7`,
`pvi`.`value7` AS `value7`,
`pvi`.`field8` AS `index8`,
```

```
`pvi`.`value8` AS `value8`,
`pvi`.`field9` AS `index9`,
`pvi`.`value9` AS `value9`,
`pvi`.`field10` AS `index10`,
`pvi`.`value10` AS `value10`
FROM (((`LOOM_PROCESSINSTANCE` `pi` JOIN `LOOM_PROCESSVARIABLEINDEX`
`pvi` ON (`pi`.`processVariableIndex_ID_` = `pvi`.`ID_`))) JOIN
`LOOM_PROCESSDEFINITION` `pd` ON((`pi`.`PROCESSDEFINITION_` =
`pd`.`ID_`)))
WHERE (`pi`.`CLIENT_` = `currentClient`());




CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY
DEFINER VIEW `view_swimlane` AS SELECT
`LOOM_SWIMLANEINSTANCE`.`ID_` AS `id`,
`LOOM_SWIMLANEINSTANCE`.`NAME_` AS `name`,
`LOOM_SWIMLANEINSTANCE`.`CLIENT_` AS `clientId`,
`LOOM_SWIMLANEINSTANCE`.`PROCINST_` AS `instanceId`,
`LOOM_SWIMLANEINSTANCE`.`ACTOR_` AS `actor`,
`LOOM_SWIMLANEINSTANCE`.`POOLEDACTOR_` AS `pooledActor`
FROM `LOOM_SWIMLANEINSTANCE` WHERE (`LOOM_SWIMLANEINSTANCE`.`CLIENT_` =
`currentClient`());




CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY
DEFINER VIEW `view_task` AS SELECT
`ti`.`ID_` AS `id`,
`ti`.`NAME_` AS `name`,
`ti`.`CLIENT_` AS `clientId`,
`ti`.`CREATION_TIME_` AS `creationTime`,
`ti`.`CREATION_USER_` AS `creationUserId`,
`ti`.`PROCINST_` AS `instanceId`,
`ti`.`DESCRIPTION_` AS `description`,
`ti`.`START_` AS `start`,
`ti`.`END_` AS `end`,
`ti`.`ISOPEN_` AS `isOpen`,
`ti`.`SWIMLANINSTANCE_` AS `swimlaneId`,
`ti`.`ACTOR_` AS `actor`,
`ti`.`POOLEDACTOR_` AS `pooledActor`,
`ti`.`NODEINSTANCE_` AS `activity`,
`ta`.`ISADHOC_` AS `isAdhoc`,
`ti`.`PARENT_FOLDER_` AS `parentFolderId`,
`ti`.`archiv_` AS `archiv`
FROM (`LOOM_TASKINSTANCE` `ti` JOIN `LOOM_TASK` `ta` ON((`ti`.`TASK_` =
`ta`.`ID_`))) WHERE (`ti`.`CLIENT_` = `currentClient`());
```

```
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY
DEFINER VIEW `view_token` AS SELECT
`tok`.`ID_` AS `id`,
`tok`.`NAME_` AS `name`,
`pi`.`CLIENT_` AS `clientId`,
`tok`.`NODEINSTANCE_` AS `nodeInstance`,
`tok`.`PROCESSINSTANCE_` AS `instanceId`,
`tok`.`PARENT_` AS `parent`
FROM (`LOOM_TOKEN` `tok` JOIN `LOOM_PROCESSINSTANCE` `pi`
ON((`tok`.`PROCESSINSTANCE_` = `pi`.`ID_`))) WHERE (`pi`.`CLIENT_` =
`currentClient`());



CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY
DEFINER VIEW `view_variable` AS SELECT
`vi`.`ID_` AS `id`,
`vi`.`NAME_` AS `name`,
`pi`.`CLIENT_` AS `clientId`,
`vi`.`PROCESSINSTANCE_` AS `instanceId`,
`vi`.`Stringvalue_` AS `stringvalue`,
`vi`.`label_` AS `label`
FROM (`LOOM_VARIABLEINSTANCE` `vi` JOIN `LOOM_PROCESSINSTANCE` `pi`
ON((`vi`.`PROCESSINSTANCE_` = `pi`.`ID_`))) WHERE (`pi`.`CLIENT_` =
`currentClient`());

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY
DEFINER VIEW `view_role` AS SELECT
`LOOM_IDENTITY_ID_` AS `identityId`,
`ROLE_ID_` AS `roleId`
FROM `MN_IDENTITY_ROLE`;
```

Sobald die Views angelegt sind, können diese mit Hilfe der beschriebenen T!M Tabellenstruktur für Auswertungen mit dem T!M Dashboard verwendet werden.