## 19.7 Database structure

Here we outline Teamwork's database relational structure. We assume familiarity with relational concepts. For readability and in order to give the essential information, we don't cover complete schemas. E.g. relationship to the area and operator table are usually removed.

19.7.1 Basic task data and dates

Basic task data is in the twk_task table. Notice the "parent" column that is a foreign key that points to the same table, and determines the task position in the tree. The recursive values of the parents are also stored (denormalized) in the "ancestorids" field for performance reasons.
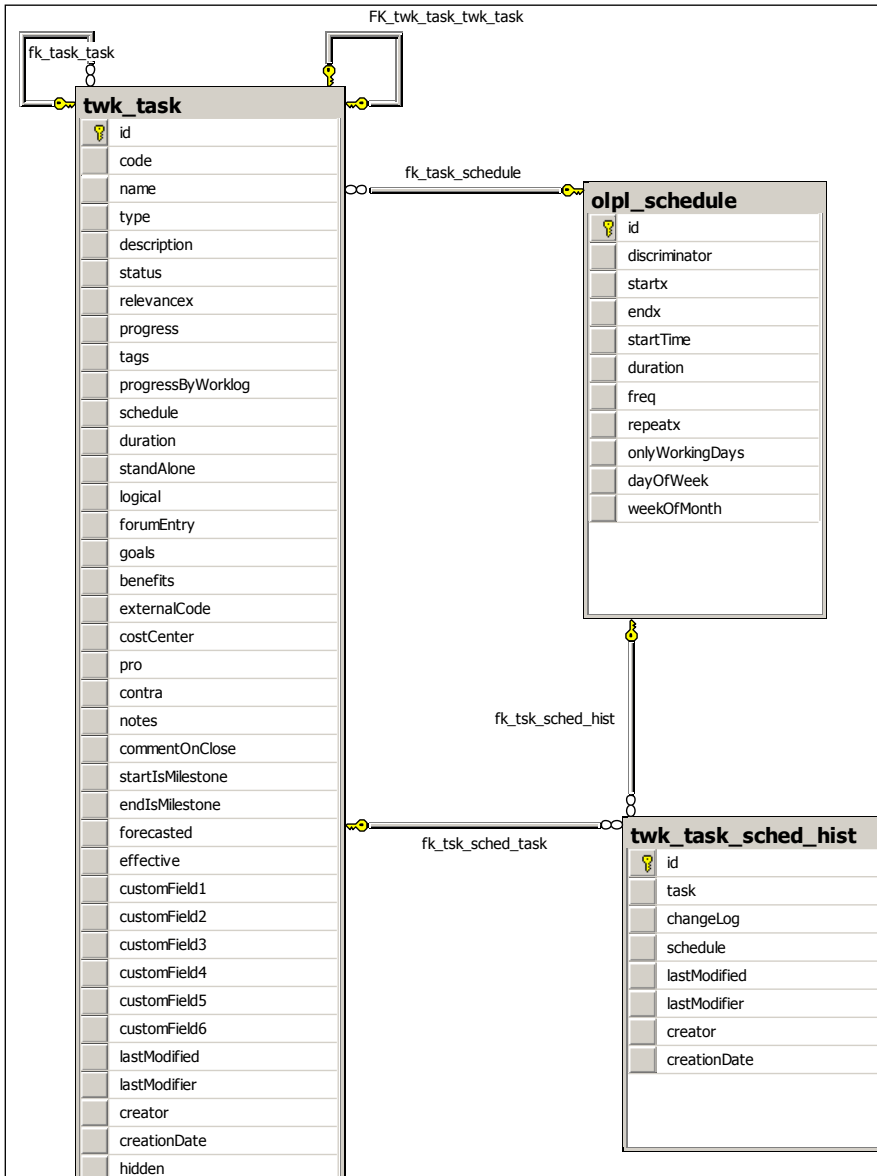
Project / task dates: the start, end and duration of the task are found by a join to the table olpl_schedule, to the filds startx, endx and duration respectively.

19.7.2 Task and assignments
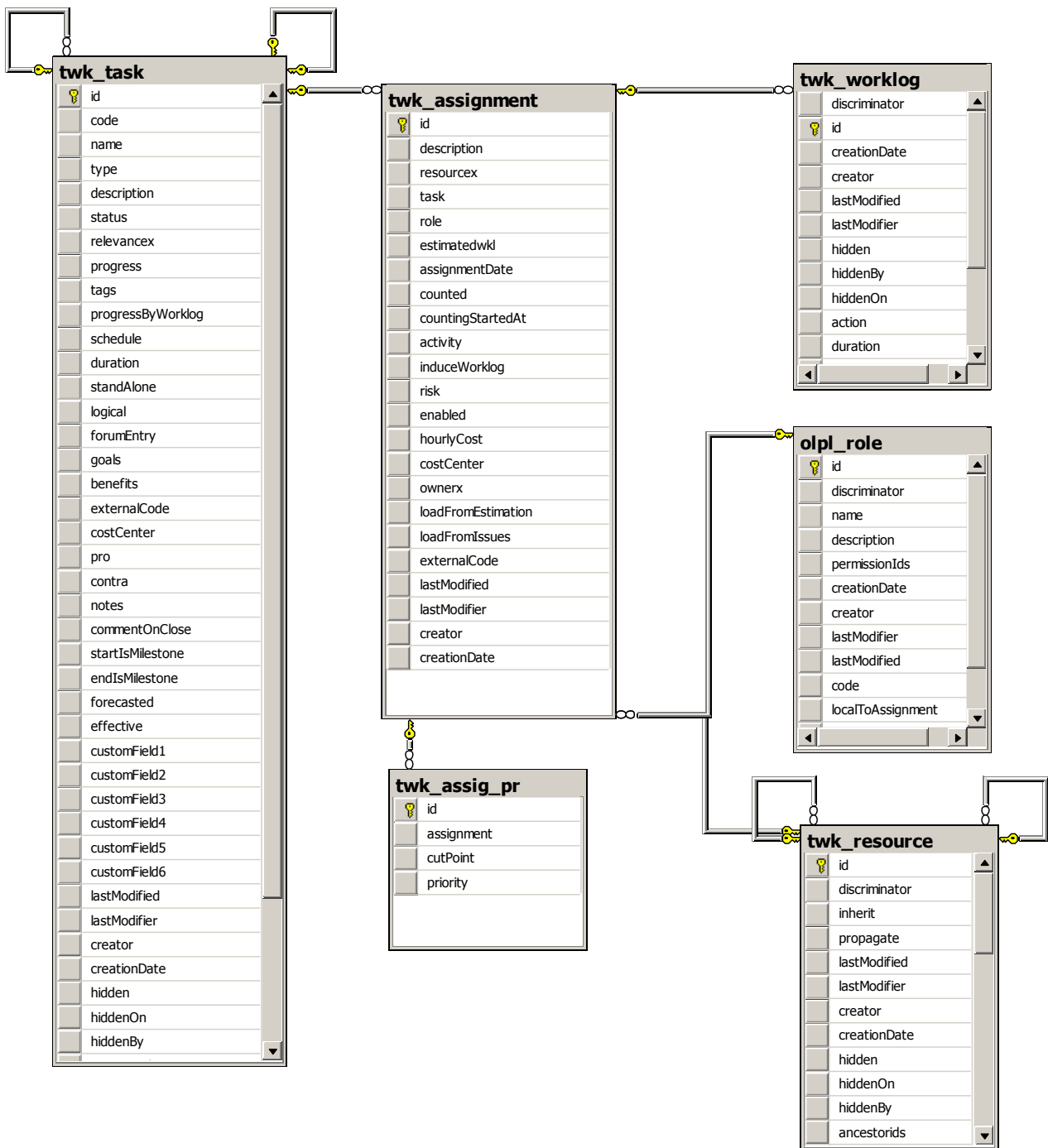
The most important relation connecting tasks with people is the assignment. The twk_assignment table is a relational table that connects tasks to resources (people) and roles. Also all worklog inserted in Teamwork is "on" an assignment. So to query the worklog on a task, you will have to join to assignments and from there join to worklogs.

twk_issue

- 🔑 id
- status
- gravity
- ownerx
- type
- impact
- descriptionx
- votes
- shouldCloseBy
- assignedTo
- task
- screenShot
- lastModified
- lastModifier
- creator
- creationDate
- lastStatusChangeDate
- assignedBy
- orderFactor
- screenShot2
- url1
- url2
- notes
- dateSignalled
- estimatedDuration
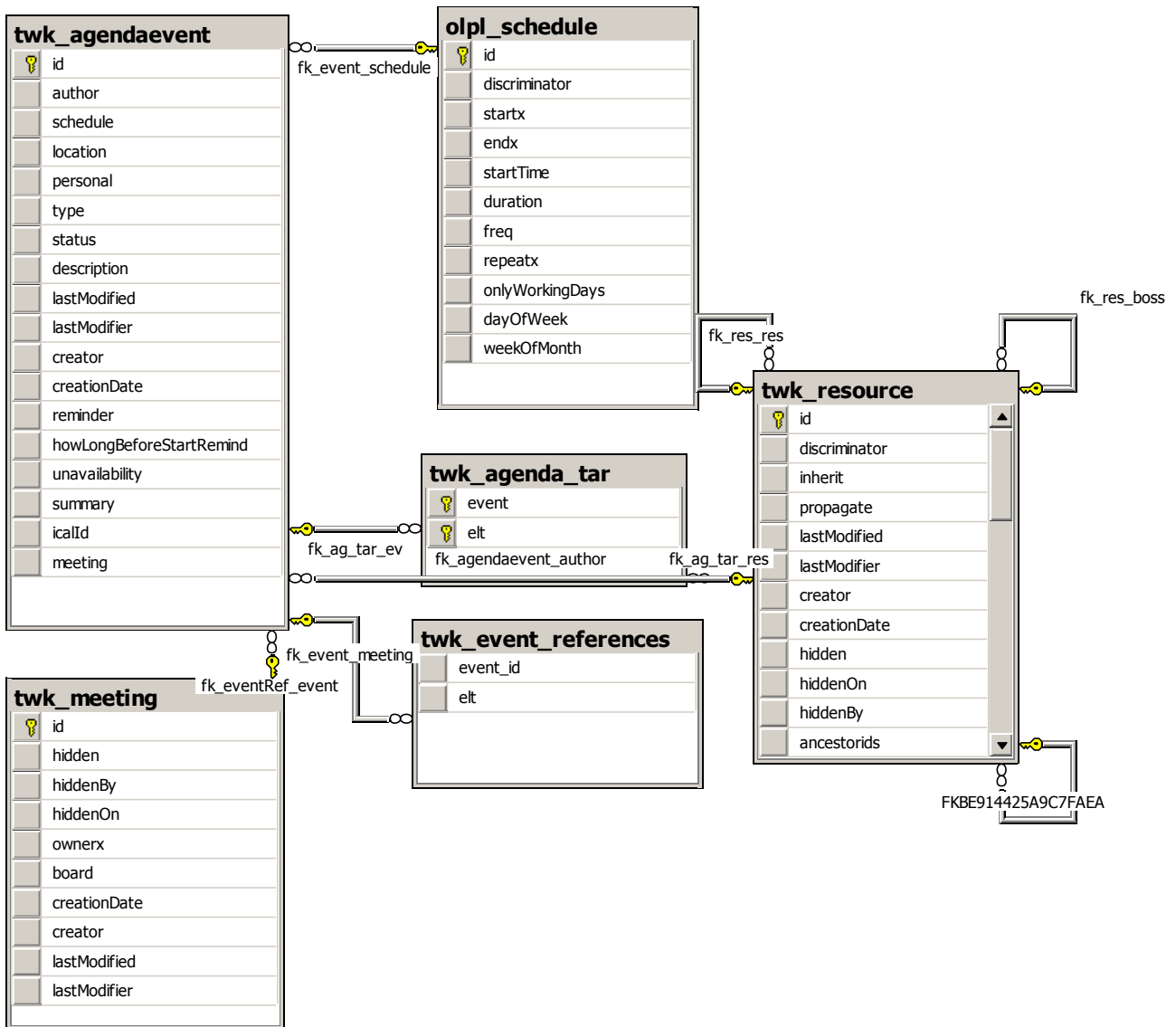- orderfactorbyres
- hidden
- hiddenBy
- hiddenOn
- tags
- areax
- customField1
- customField2
- customField3
- customField4
- customField5
- customField6
- statusx
- codex

twk_resource

- 🔑 id
- discriminator
- inherit
- propagate
- lastModified
- lastModifier
- creator
- creationDate
- hidden
- hiddenOn
- hiddenBy
- ancestorids

twk_issue_history

- 🔑 id
- creationDate
- creator
- lastModified
- lastModifier
- assignee
- issue
- statusx
- task
- olsstatus
- oldstatus
- descriptionx
- commentx

twk_task

- 🔑 id
- code
- name
- type
- description
- status
- relevancex
- progress
- estwklgreq
- schedule
- standAlone
- logical

twk_issue_status

- 🔑 id
- askForComment
- askForWorklog
- behavesAsClosed
- behavesAsOpen
- color
- description
- orderBy

twk_worklog

- 🔑 id
- duration
- inserted
- assig
- issue

Issues can be without resource – just on a task, or without task – and then are a todo on a resource. But normally have a resource and a task, which indirectly most often determines an assignment, and that is where the worklog goes.
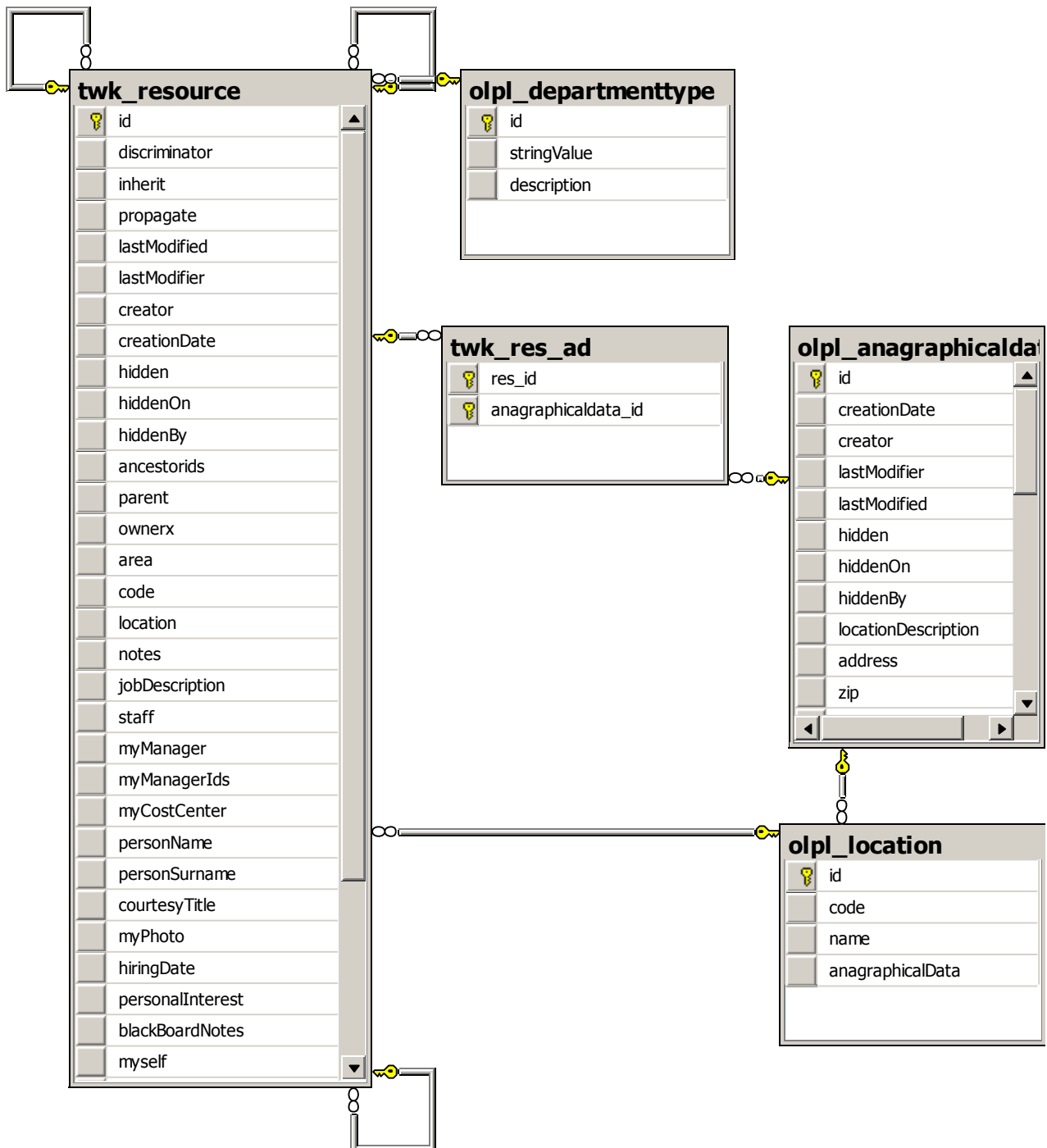
Notice that there is a denormalized reference to worklogs, and the table issue_history where the history of changes to the issue is kept.

twk_agendaevent

🔑	id
	author
	schedule
	location
	personal
	type
	status
	description
	lastModified
	lastModifier
	creator
	creationDate
	reminder
	howLongBeforeStartRemind
	unavailability
	summary
	icalId
	meeting

fk_event_schedule

olpl_schedule

🔑	id
	discriminator
	startx
	endx
	startTime
	duration
	freq
	repeatx
	onlyWorkingDays
	dayOfWeek
	weekOfMonth

fk_res_res

fk_res_boss

twk_resource

🔑	id
	discriminator
	inherit
	propagate
	lastModified
	lastModifier
	creator
	creationDate
	hidden
	hiddenOn
	hiddenBy
	ancestorids

FKBE914425A9C7FAEA

twk_agenda_tar

🔑	event
🔑	elt

fk_ag_tar_ev

fk_agendaevent_author

fk_ag_tar_res

fk_event_meeting

twk_event_references

	event_id
	elt

fk_eventRef_event

twk_meeting

🔑	id
	hidden
	hiddenBy
	hiddenOn
	ownerx
	board
	creationDate
	creator
	lastModified
	lastModifier

Every item in the agenda has a schedule and a set of participants. The schedule is directly joined, the participants are collected by joining through the twk_agenda_tar. If eventually the agenda event is also a meeting, there is a foreign key to the twk_meeting table.
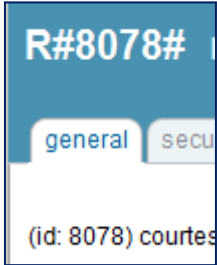
twk_resource

🔑	id
	discriminator
	inherit
	propagate
	lastModified
	lastModifier
	creator
	creationDate
	hidden
	hiddenOn
	hiddenBy
	ancestorids
	parent
	ownerx
	area
	code
	location
	notes
	jobDescription
	staff
	myManager
	myManagerIds
	myCostCenter
	personName
	personSurname
	courtesyTitle
	myPhoto
	hiringDate
	personalInterest
	blackBoardNotes
	myself

olpl_departmenttype

🔑	id
	stringValue
	description

twk_res_ad

🔑	res_id
🔑	anagraphicaldata_id

olpl_anagraphicaldata

🔑	id
	creationDate
	creator
	lastModifier
	lastModified
	hidden
	hiddenOn
	hiddenBy
	locationDescription
	address
	zip

olpl_location

🔑	id
	code
	name
	anagraphicalData

Resources are obviously involved in most Teamwork tables; here we just show the direct dependencies on this table. Notice that the resource basic data (such as "e-mail") is kept in a join though twk_res_ad to olpl_anagraphicaldata.

19.7.6 What is the logged user?

Teamwork logged users are "double" entities: they are "resources", so that assignments can be done on them, and users of the system ,so they can log in.

So a users that logs in is a record in the olpl_operator table, that is in 1-to-1 relationship with a record in the twk_resources table.

The resource record id is shown in the web interface in the resource editor general tab;
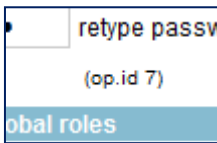
The operator record id in the security/login tab.

In almost all cases what you need to filter records is the resource record id: on issues, worklogs, assignments.

19.7.7 Filter by area?

Filtering by area is trivial, as most objects have an area or areax column for which you can filter.