ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА

ФАКУЛТЕТ ПО ИЗЧИСЛИТЕЛНА ТЕХНИКА И АВТОМАТИЗАЦИЯ

Катедра „Софтуерно Инженерство “

A blue and white logo

Description automatically generated with medium confidence

**Курсова Работа по**

**АБТМУ**

Microsoft технологии за проектиране и администриране на разпределени бази от данни

**Тема:**

Система за следене на задачи

**Разработил:**

Ивайло Пламенов Руменов

Фак. № 23651227

# Entity диаграма

# Схема на РБД

# SQL команди

CREATE DATABASE TASK\_SYNC;

CREATE TABLE Status(

id BIGINT NOT NULL,

name VARCHAR(255) NOT NULL

);

ALTER TABLE

Status ADD CONSTRAINT status\_id\_primary PRIMARY KEY(id);

CREATE TABLE Position(

id BIGINT NOT NULL,

name VARCHAR(255) NOT NULL

);

ALTER TABLE

Position ADD CONSTRAINT position\_id\_primary PRIMARY KEY(id);

CREATE TABLE Task(

id BIGINT NOT NULL,

name VARCHAR(255) NOT NULL,

work\_time BIGINT NOT NULL,

project BIGINT NOT NULL,

due DATE NOT NULL

);

ALTER TABLE

Task ADD CONSTRAINT task\_id\_primary PRIMARY KEY(id);

CREATE TABLE Project(

id BIGINT NOT NULL,

name VARCHAR(255) NOT NULL,

manager BIGINT NOT NULL

);

ALTER TABLE

Project ADD CONSTRAINT project\_id\_primary PRIMARY KEY(id);

CREATE TABLE Sub\_task(

id\_master\_task BIGINT NULL,

id\_sub\_task BIGINT NOT NULL

);

CREATE TABLE Task\_history(

id BIGINT NOT NULL,

task BIGINT NOT NULL,

status BIGINT NOT NULL,

worker BIGINT NOT NULL,

manager BIGINT NOT NULL,

time\_stamp DATETIME DEFAULT CURRENT\_TIMESTAMP,

history\_status BIGINT NOT NULL

);

ALTER TABLE

Task\_history ADD CONSTRAINT task\_history\_id\_primary PRIMARY KEY(id);

CREATE TABLE Department(

id BIGINT NOT NULL,

name VARCHAR(255) NOT NULL

);

ALTER TABLE

Department ADD CONSTRAINT department\_id\_primary PRIMARY KEY(id);

CREATE TABLE Worker(

id BIGINT NOT NULL,

name VARCHAR(255) NOT NULL,

department BIGINT NOT NULL,

position BIGINT NOT NULL

);

CREATE TABLE Worker\_projects(

id\_worker BIGINT NULL,

id\_project BIGINT NOT NULL

);

ALTER TABLE

Worker ADD CONSTRAINT worker\_id\_primary PRIMARY KEY(id);

ALTER TABLE

Task\_history ADD CONSTRAINT task\_history\_worker\_foreign FOREIGN KEY(worker) REFERENCES Worker(id);

ALTER TABLE

Task\_history ADD CONSTRAINT task\_history\_manager\_foreign FOREIGN KEY(manager) REFERENCES Worker(id);

ALTER TABLE

Task\_history ADD CONSTRAINT task\_history\_status\_foreign FOREIGN KEY(status) REFERENCES Status(id);

ALTER TABLE

Project ADD CONSTRAINT project\_manager\_foreign FOREIGN KEY(manager) REFERENCES Worker(id);

ALTER TABLE

Sub\_task ADD CONSTRAINT sub\_task\_id\_sub\_task\_foreign FOREIGN KEY(id\_sub\_task) REFERENCES Task(id);

ALTER TABLE

Task ADD CONSTRAINT task\_project\_foreign FOREIGN KEY(project) REFERENCES Project(id);

ALTER TABLE

Worker ADD CONSTRAINT worker\_department\_foreign FOREIGN KEY(department) REFERENCES Department(id);

ALTER TABLE

Worker ADD CONSTRAINT worker\_position\_foreign FOREIGN KEY(position) REFERENCES Position(id);

ALTER TABLE

Sub\_task ADD CONSTRAINT sub\_task\_id\_master\_task\_foreign FOREIGN KEY(id\_master\_task) REFERENCES Task(id);

ALTER TABLE

Task\_history ADD CONSTRAINT task\_history\_task\_foreign FOREIGN KEY(task) REFERENCES Task(id);

ALTER TABLE

Worker\_projects ADD CONSTRAINT worker\_id\_projects FOREIGN KEY(id\_worker) REFERENCES Worker(id);

ALTER TABLE

Worker\_projects ADD CONSTRAINT project\_id\_workers FOREIGN KEY(id\_project) REFERENCES Project(id);

select \* from TASK\_SYNC.dbo.Task

INSERT INTO Department(id, name) VALUES (1,'Development');

INSERT INTO Department(id, name) VALUES (2,'Marketing');

INSERT INTO Department(id, name) VALUES (3,'Human Resources');

INSERT INTO Department(id, name) VALUES (4,'Research and Development');

Insert Into Position(id, name) VALUES (1,'Project Manager');

Insert Into Position(id, name) VALUES (2,'Manager');

Insert Into Position(id, name) VALUES (3,'Programmer');

Insert Into Position(id, name) VALUES (4,'QA');

Insert Into Position(id, name) VALUES (5,'Tester');

INSERT INTO Worker(id, name, department, position) VALUES (1,'Ivan',1,1);

INSERT INTO Worker(id, name, department, position) VALUES (2,'Iliq',1,2);

INSERT INTO Worker(id, name, department, position) VALUES (3,'Koko',1,2);

INSERT INTO Worker(id, name, department, position) VALUES (4,'Plamen',1,2);

INSERT INTO Worker(id, name, department, position) VALUES (5,'Qna',1,3);

INSERT INTO Worker(id, name, department, position) VALUES (6,'Jordan',1,3);

INSERT INTO Worker(id, name, department, position) VALUES (7,'Jhon',1,3);

INSERT INTO Worker(id, name, department, position) VALUES (8,'Jhoana',1,4);

INSERT INTO Worker(id, name, department, position) VALUES (9,'Emma',1,4);

INSERT INTO Worker(id, name, department, position) VALUES (10,'Luna',1,5);

INSERT INTO Worker(id, name, department, position) VALUES (11,'Olivia',1,5);

Insert Into Project(id, name, manager) VALUES (1,'Bulgerian Trains',1);

Insert Into Worker\_projects(id\_worker, id\_project) values (1,1);

Insert Into Worker\_projects(id\_worker, id\_project) values (2,1);

Insert Into Worker\_projects(id\_worker, id\_project) values (3,1);

Insert Into Worker\_projects(id\_worker, id\_project) values (4,1);

Insert Into Worker\_projects(id\_worker, id\_project) values (5,1);

Insert Into Worker\_projects(id\_worker, id\_project) values (6,1);

Insert Into Worker\_projects(id\_worker, id\_project) values (7,1);

Insert Into Worker\_projects(id\_worker, id\_project) values (8,1);

Insert Into Worker\_projects(id\_worker, id\_project) values (9,1);

Insert Into Worker\_projects(id\_worker, id\_project) values (10,1);

Insert Into Worker\_projects(id\_worker, id\_project) values (11,1);

INSERT INTO Task(id, name, work\_time, project, due) VALUES (1,'Creating database',0,1,'2024-03-25');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (2,'Creating creation scripts',0,1,'2024-03-12');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (3,'Creating insertion script',0,1,'2024-03-13');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (4,'Creating drop scripts',0,1,'2024-03-14');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (5,'Creating Triggers',0,1,'2024-03-15');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (6,'Creating Procedures',0,1,'2024-03-16');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (7,'Test database',0,1,'2024-03-24');

INSERT INTO Sub\_task(id\_master\_task, id\_sub\_task) VALUES (1,2);

INSERT INTO Sub\_task(id\_master\_task, id\_sub\_task) VALUES (1,3);

INSERT INTO Sub\_task(id\_master\_task, id\_sub\_task) VALUES (1,4);

INSERT INTO Sub\_task(id\_master\_task, id\_sub\_task) VALUES (1,5);

INSERT INTO Sub\_task(id\_master\_task, id\_sub\_task) VALUES (1,6);

INSERT INTO Sub\_task(id\_master\_task, id\_sub\_task) VALUES (1,7);

Insert Into Status(id, name) VALUES (1,'Undefined');

Insert Into Status(id, name) VALUES (2,'To do');

Insert Into Status(id, name) VALUES (3,'Doing');

Insert Into Status(id, name) VALUES (4,'Done');

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (1,1,1,6,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (2,2,1,6,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (3,3,1,7,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (4,4,1,7,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (5,5,1,6,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (6,6,1,7,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (7,7,1,11,2,1);

INSERT INTO Task(id, name, work\_time, project, due) VALUES (8,'Creating log in Endpoint',0,1,'2024-03-25');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (9,'Creating register in Endpoint',0,1,'2024-03-12');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (10,'Creating web UI',0,1,'2024-03-13');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (11,'Test log in Endpoint',0,1,'2024-03-14');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (12,'Test register Endpoint',0,1,'2024-03-15');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (13,'Test web UI',0,1,'2024-03-16');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (14,'Deploy to server',0,1,'2024-04-01');

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (8,8,1,6,3,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (9,9,1,5,3,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (10,10,1,5,3,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (11,11,1,10,3,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (12,12,1,10,4,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (13,13,1,11,4,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (14,14,1,2,4,1);

/\*

Second project

\*/

INSERT INTO Worker(id, name, department, position) VALUES (12,'Iliqna',1,2);

INSERT INTO Worker(id, name, department, position) VALUES (13,'Koka',1,2);

INSERT INTO Worker(id, name, department, position) VALUES (14,'Plamena',1,2);

INSERT INTO Worker(id, name, department, position) VALUES (15,'Qno',1,3);

INSERT INTO Worker(id, name, department, position) VALUES (16,'Jordana',1,3);

INSERT INTO Worker(id, name, department, position) VALUES (17,'Jhona',1,3);

INSERT INTO Worker(id, name, department, position) VALUES (18,'Jhoan',1,4);

INSERT INTO Worker(id, name, department, position) VALUES (19,'Emmo',1,4);

INSERT INTO Worker(id, name, department, position) VALUES (20,'Luno',1,5);

INSERT INTO Worker(id, name, department, position) VALUES (21,'Oliv',1,5);

Insert Into Project(id, name, manager) VALUES (2,'Fitnnes application',12);

Insert Into Worker\_projects(id\_worker, id\_project) values (12,2);

Insert Into Worker\_projects(id\_worker, id\_project) values (13,2);

Insert Into Worker\_projects(id\_worker, id\_project) values (14,2);

Insert Into Worker\_projects(id\_worker, id\_project) values (15,2);

Insert Into Worker\_projects(id\_worker, id\_project) values (16,2);

Insert Into Worker\_projects(id\_worker, id\_project) values (17,2);

Insert Into Worker\_projects(id\_worker, id\_project) values (18,2);

Insert Into Worker\_projects(id\_worker, id\_project) values (19,2);

Insert Into Worker\_projects(id\_worker, id\_project) values (20,2);

Insert Into Worker\_projects(id\_worker, id\_project) values (21,2);

INSERT INTO Task(id, name, work\_time, project, due) VALUES (15,'Creating runners routine',0,2,'2024-03-25');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (16,'Set up running track',0,2,'2024-03-12');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (17,'Set up looker room',0,2,'2024-03-13');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (18,'Clean out gum',0,2,'2024-03-14');

INSERT INTO Sub\_task(id\_master\_task, id\_sub\_task) VALUES (15,16);

INSERT INTO Sub\_task(id\_master\_task, id\_sub\_task) VALUES (15,17);

INSERT INTO Sub\_task(id\_master\_task, id\_sub\_task) VALUES (15,18);

INSERT INTO Task(id, name, work\_time, project, due) VALUES (19,'Creating lifter routine',0,2,'2024-03-15');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (20,'Set up dumbbells',0,2,'2024-03-16');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (21,'Set up barbell',0,2,'2024-03-24');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (22,'Set up roll machine',0,2,'2024-03-16');

INSERT INTO Task(id, name, work\_time, project, due) VALUES (23,'Set squat rack',0,2,'2024-03-24');

INSERT INTO Sub\_task(id\_master\_task, id\_sub\_task) VALUES (19,20);

INSERT INTO Sub\_task(id\_master\_task, id\_sub\_task) VALUES (19,21);

INSERT INTO Sub\_task(id\_master\_task, id\_sub\_task) VALUES (19,22);

INSERT INTO Sub\_task(id\_master\_task, id\_sub\_task) VALUES (19,23);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (15,15,1,14,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (16,16,1,15,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (17,17,1,16,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (18,18,1,14,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (19,19,1,15,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (20,20,1,18,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (21,21,1,13,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (22,22,1,19,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (23,23,1,20,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (24,23,2,20,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (25,23,3,20,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (26,22,2,20,2,1);

Insert Into Task\_history(id, task, status, worker, manager, history\_status) VALUES (27,23,3,15,2,1);

drop table Task\_history;

drop table Sub\_task;

drop table Status;

drop table Task;

drop table Worker\_projects;

drop table Project;

drop table Worker;

drop table Department;

drop table Position;

# Програмен код на създавене обекти- индекси, изгледи, съхранени процедури функции и тригери

CREATE

UNIQUE

CLUSTERED INDEX IX\_MyIndexedView ON WrokerOnProjectWhitTasks (worker\_name,due);

-- Check if the view is schema-bound

SELECT OBJECTPROPERTY(OBJECT\_ID('dbo.WrokerOnProjectWhitTasks'), 'IsSchemaBound') AS IsSchemaBound;

create view WrokerOnProjectWhitTasks

as

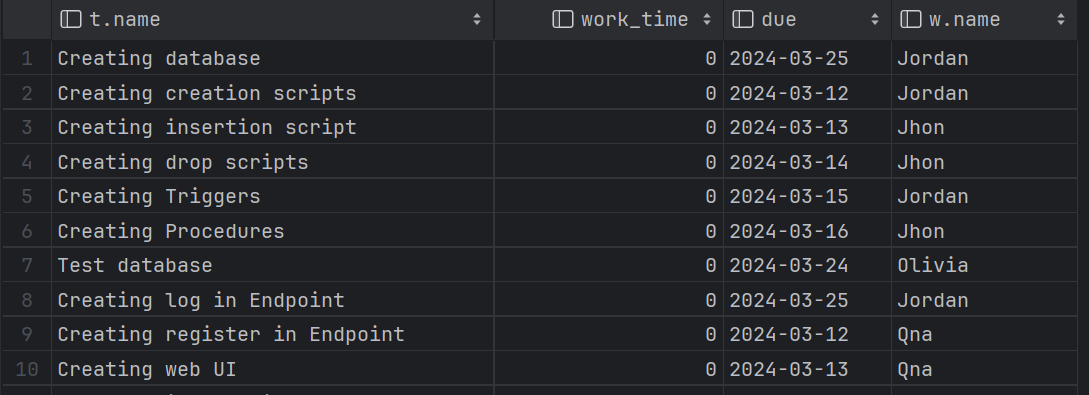
select t.name as tickate\_name, t.work\_time, t.due, w.name as worker\_name

from Task t

join Task\_history th on t.id = th.task

join Worker w on th.worker = w.id

join Worker\_projects wp on wp.id\_worker = w.id



create procedure allWorkerInProject(@projectId as INT) as

begin

Select w.name as workerName, d.name as departmentName, p.name as possitionName, pj.name as projectName

from worker w

inner join Department d on d.id = w.department

inner join Position p on p.id = w.position

inner join Worker\_projects wp on wp.id\_worker = w.id

inner join Project pj on pj.id = wp.id\_project

where pj.id = @projectId

end;

EXEC allWorkerInProject 1;

create procedure getAllUnfinishedSubTasks(@masterTaskId as INT) as

begin

Select t.name as taskName, t.due as taskDueDate, w.name as workerName, s.name as taskStatus

from Sub\_task st

inner join Task t on st.id\_sub\_task = t.id

inner join Task\_history th on th.task = t.id

left join Worker w on w.id = th.worker

inner join Status s on s.id = th.status

where st.id\_master\_task = @masterTaskId

and th.status != 4

end;

Exec getAllUnfinishedSubTasks 1;

create procedure hasUninishedSubTasks(@masterTaskId as INT) as

begin

Select count(t.id) as unfinishedTasks

from Sub\_task st

inner join Task t on st.id\_sub\_task = t.id

inner join Task\_history th on th.task = t.id

left join Worker w on w.id = th.worker

inner join Status s on s.id = th.status

where st.id\_master\_task = @masterTaskId

and th.status != 4

end;

exec hasUninishedSubTasks 2

create procedure getTaskStastDate(@taskId as INT)

as

begin

Select top 1 CONVERT(date, th.time\_stamp)

as taskCreationDate

from Task\_history th

where th.task = @taskId

order by th.time\_stamp asc;

end;

exec getTaskStastDate 1

create procedure getAllTasksInaProject(@projectId as INT) as

begin

Select t.name as taskName, t.due as taskDueDate, w.name as workerName, s.name as taskStatus

from Project pj

inner join Task t on t.project = pj.id

inner join Task\_history th on th.task = t.id

left join Worker w on w.id = th.worker

inner join Status s on s.id = th.status

where pj.id = @projectId

and th.status != 4

end;

exec getAllTasksInaProject 1;

/\*

Rollback Procedure

\*/

create procedure rollbackIfSubtaskIsNotFnished(@masterTaskId as INT)

as

begin

DECLARE

@subtasks DECIMAL

Select @subtasks = count(t.id)

from Sub\_task st

inner join Task t on st.id\_sub\_task = t.id

inner join Task\_history th on th.task = t.id

left join Worker w on w.id = th.worker

inner join Status s on s.id = th.status

where st.id\_master\_task = @masterTaskId

and th.status != 4

IF @subtasks > 0

BEGIN

RAISERROR

('All sub tasks must be finished before closing ticket.', 16, 1)

ROLLBACK TRANSACTION

END

end;

/\*

View Procedure

\*/

create procedure selectAllWorkerWhitTasks(@projectId as INT)

as

begin

select \*

from Task t

join Task\_history th on t.id = th.task

join Worker w on th.worker = w.id

join Worker\_projects wp on wp.id\_worker = w.id

where wp.id\_project = @projectId

end;

CREATE FUNCTION WorkersOnProject(@workerName as VARCHAR(100))

RETURNS table

AS

return

(

select \*

from WrokerOnProjectWhitTasks wp

where wp.worker\_name = @workerName

)

drop function WorkersOnProject

select \*

from WorkersOnProject('Jhon')

select \*

from WrokerOnProjectWhitTasks wp

where wp.worker\_name = 'Jhon'

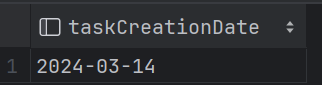
# Справки

Select top 1 CONVERT(date, th.time\_stamp)

as taskCreationDate

from Task\_history th

where th.task = 1

order by th.time\_stamp asc;

/\*

Get all Workers in a project

\*/

Select w.name as workerName, d.name as departmentName, p.name as possitionName, pj.name as projectName

from worker w

inner join Department d on d.id = w.department

inner join Position p on p.id = w.position

inner join Worker\_projects wp on wp.id\_worker = w.id

A screenshot of a computer

Description automatically generated inner join Project pj on pj.id = wp.id\_project

where pj.id = 1

/\*

Get All unfinished tasks just number

\*/

Select count(t.id) as unfinishedTasks

from Sub\_task st

inner join Task t on st.id\_sub\_task = t.id

inner join Task\_history th on th.task = t.id

left join Worker w on w.id = th.worker

inner join Status s on s.id = th.status

where st.id\_master\_task = 1

A black background with white text

Description automatically generated and th.status != 4

/\*

Get All unfinished tasks

\*/

Select t.name as taskName, t.due as taskDueDate, w.name as workerName, s.name as taskStatus

from Sub\_task st

inner join Task t on st.id\_sub\_task = t.id

inner join Task\_history th on th.task = t.id

left join Worker w on w.id = th.worker

inner join Status s on s.id = th.status

where st.id\_master\_task = 1

A screenshot of a computer

Description automatically generated and th.status != 4;