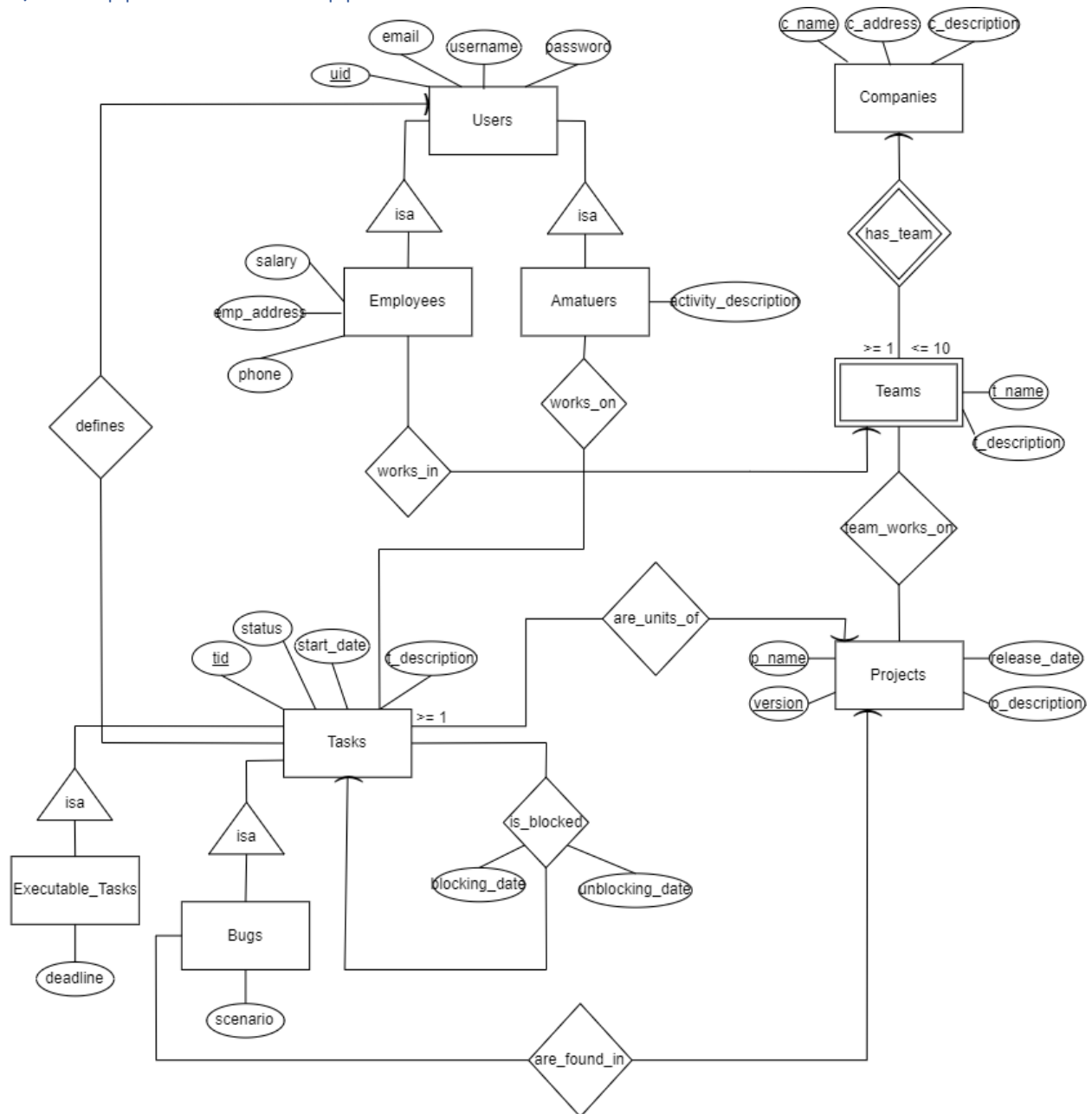


Релационен модел на базата данни

Тема No 21: Управление на проекти

Е/R модел на базата данни



Таблицы

I. ISA йерархия Потребители

Релационен подход:

Users (uid: int (> 0), email: string (320), username: string (10), password: string (64))

Employees (uid: int (> 0), salary: double (> 0.0), emp_address: string (100), phone: string (>= 10 && <= 13))

Amateurs (uid: int (> 0), activity_description: string (350))

ООП подход:

Users (uid: int (> 0), email: string (320), username: string (10), password: string (64))

PK: uid (unique, not null)

~~**Employees** (uid: int (> 0), salary: double (> 0.0), emp_address: string (100), phone: string (>= 10 && <= 13))~~ – *таблицата отпада*

~~**Amateurs** (uid: int (> 0), activity_description: string (100))~~ – *таблицата отпада*

Users_Employees (uid: int (> 0), email: string (320), username: string (10), password: string (64), salary: double (> 0.0), emp_address: string (100), phone: string (>= 10 && <= 13), team_id: int (> 0))

PK: uid (unique, not null)

FK: Users (team_id) -> Teams (tid)

Users_Amateurs (uid: int (> 0), email: string (320), username: string (10), password: string (64), activity_description: string (350))

PK: uid (unique, not null)

NULL подход:

Users (uid: int (> 0), email: string (320), username: string (10), password: string (64), team_id: int (> 0), salary: double (> 0.0), emp_address: string (100), phone: string (>= 10 && <= 13), activity_description: string (350))

Избирам ООП подхода за йерархията на потребителите със съкращаване на две от таблиците (Employees и Amateurs), защото те са нефункционални и не носят достатъчно информация за въпросните роли на потребителя. Не избирам релационния подход или NULL подхода, защото при бъдещи разширения на системата те биха били лош вариант поради излишество (повторения) на данни. С оглед на възможните транзакции към базата този подход би бил най-удачен, защото в повечето случаи ще се обръщаме към само една от таблиците в модела. За потребителите ще генерираме 3 таблици.

II. ISA йерархия Задачи

Релационен подход:

Tasks (tid: int (> 0), status: string (10 – unfinished), start_date: date (>= 01.01.2023), tdescription: string (350), uid: int (> 0), project_id: int (> 0), btid: int (> 0))

PK: tid (unique, not null)

FK: Tasks (uid) -> Users (uid)

Tasks (project_id) -> Projects (pid)

Tasks (btid) -> Tasks (tid)

Executable_Tasks (tid: int (> 0), **deadline**: date (>= 01.01.2023))

PK: tid (unique, not null)

Bugs_Tasks (tid: int (> 0), scenario: string (600), project_id: int (> 0), **blocking_date**: date (>= 01.01.2023), **unblocking_date**: date (>= blocking_date))

PK: tid (unique, not null)

FK: Bugs_Tasks (project_id) -> Projects (pid)

ООП подход:

Tasks (tid: int (> 0), status: string (10 – unfinished), start_date: date (>= 01.01.2023), tdescription: string (350))

Executables (tid: int (> 0), deadline: date (>= 01.01.2023))

Bugs (tid: int (> 0), scenario: string (600), project_id: int (> 0))

Executable_Tasks (tid: int (> 0), status: string (10 – unfinished), start_date: date (>= 01.01.2023), tdescription: string (350), deadline: date (>= start_date))

Bugs_Tasks (tid: int (> 0), status: string (10 – unfinished), start_date: date (>= 01.01.2023), tdescription: string (350), scenario: string (600), project_id: int (> 0), blocking_date: date (>= 01.01.2023), unblocking_date: date (>= blocking_date))

NULL подход:

Tasks (tid: int (> 0), status: string (10 – unfinished), start_date: date (>= 01.01.2023), tdescription: string (350), deadline: date (>= start_date), scenario: string (600), project_id: int (> 0), blocking_date: date (>= 01.01.2023), unblocking_date: date (>= blocking_date))

Избирам релационния подход, защото за удобство бих могла да си позволя да разделя атрибутите, необходими за описанието на различни типове задачи, в три таблици. С оглед на възможните транзакции към базата, обаче, този подход не би бил най-удачен, защото в част от случаи ще се обръщаме към повече от една таблица в модела, за да извлечем цялостно информацията, от която се интересуваме. Но от гледна точка на сравнително малкото количество данни в таблиците, това не би било от особено значение за бързодействието на модела. За задачите ще генерираме 3 таблици.

III. Таблицы на останалите множества от същности и взаимоотношенията им

Companies (cid: int (> 0), cname: string (50), caddress: string (100), cdescription: string (500))

PK: cid (unique, not null)

Забележка: Тъй като името на компанията ще е низ с променлива дължина, не би било удачно да е ключ, заради проблеми с индексацията на такива полета. Това налага въвеждането на ново поле за ключ, а именно cid: int (> 0).

Teams (team_id: int (> 0), cid: int (> 0), tname: string (50), tdescription: string (350))

PK: team_id (unique, not null)

FK: Teams (cid) -> Companies (cid)

Забележка: Тъй като имената на екипа и компанията ще са низове с променлива дължина, не би било удачно да сформират ключ, заради проблеми с индексацията

на такива полета. Това налага въвеждането на ново поле за ключ, а именно team_id: int (>= 0).

Projects (project_id: int (> 0), pname: string (50), version: string (20), **release_date**: date (>= 01.01.2023), pdescription: string (350))

PK: project_id (unique, not null)

Забележка: Тъй като името на проекта ще е низ с променлива дължина, не би било удачно да е ключ, заради проблеми с индексацията на такива полета. Това налага въвеждането на ново поле за ключ, а именно pid: int (>= 0).

Defines (uid: int (> 0), tid: int (> 0)) – **таблицата отпада**

AreFoundIn (tid: int (> 0), project_id: int (> 0)) – **таблицата отпада**

AreUnitsOf (tid: int (> 0), project_id: int (> 0)) – **таблицата отпада**

IsBlocked (blocking_task_id: int (> 0), blocked_task_id: int (> 0), blocking_date: date (>= 01.01.2023), unblocking_date: date (>= blocking_date)) – **таблицата отпада**

WorksIn (uid: int (> 0), team_id: int (> 0)) – **таблицата отпада**

WorksOn (uid: int (> 0), tid: int (> 0))

PK: uid (unique, not null), tid (unique, not null)

FK: WorksOn (uid) -> Users (uid)

WorksOn (tid) -> Tasks (tid)

TeamsWorkOn (team_id: int (> 0), project_id: int (> 0))

PK: team_id (not null), project_id (unique, not null)

FK: TeamsWorkOn (team_id) -> Teams (team_id)

TeamsWorkOn (project_id) -> Projects (project_id)

Забележка: дължината на полетата от тип string са проверявани от различни онлайн източници по спецификациите на SQL с цел максимална прецизност и доближаване до изискванията на задачата.

Извод

След всички преобразувания до момента стигаме до следните релации.

Таблицы

Users (uid: int (>= 0), **email**: string (320), **username**: string (10), **password**: string (64))

Users_Employees (uid: int (> 0), **email**: string (320), **username**: string (10), **password**: string (64), **salary**: double (> 0.0), emp_address: string (100), phone: string (>= 10 && <= 13), team_id: int (> 0))

Users_Amateurs (uid: int (> 0), **email**: string (320), **username**: string (10), **password**: string (64), activity_description: string (350))

Tasks (tid: int (> 0), **status**: string (10 – unfinished), **start_date**: date (>= 01.01.2023), tdescription: string (350), uid: int (> 0), project_id: int (> 0), btid: int (> 0))

Executables (tid: int (> 0), **deadline**: date (>= 01.01.2023))

Bugs (tid: int (> 0), scenario: string (600), project_id: int (> 0), **blocking_date**: date (>= 01.01.2023), **unblocking_date**: date (>= blocking_date))

Companies (cid: int (> 0), cname: string (50), caddress: string (100), cdescription: string (500))

Teams (team_id: int (> 0), cid: int (> 0), tname: string (50), tdescription: string (350))

Projects (project_id: int (> 0), pname: string (50), version: string (20), **release_date**: date (>= 01.01.2023), pdescription: string (350))

WorksOn (uid: int (> 0), tid: int (> 0))

TeamsWorkOn (team_id: int (> 0), project_id: int (> 0))

Първични ключове (PK)

PK: Users -> uid (unique, not null)

PK: Users_Employees -> uid (unique, not null)

PK: Users_Amateurs -> uid (unique, not null)

PK: Tasks -> tid (unique, not null)

PK: Executable_Tasks -> tid (unique, not null)

PK: Bugs -> tid (unique, not null)

PK: Companies -> cid (unique, not null)

PK: Teams -> tid (unique, not null)

PK: Projects -> pid (unique, not null)

PK: WorksOn -> uid (unique, not null), tid (unique, not null)

PK: TeamsWorkOn -> team_id (not null), project_id (unique, not null)

Външни ключове (FK)

FK: Users (tname, cname) -> Teams (tname, cname)

FK: Tasks (uid) -> Tasks (uid)

FK: Tasks (pname, pversion) -> Projects (pname, version)

FK: Tasks (btid) -> Tasks (tid)

FK: Bugs (found_in_pname, found_in_pversion) -> Projects (pname, version)

FK: Teams (cname) -> Companies (cname)

FK: WorksOn (uid) -> Users (uid)

FK: WorksOn (tid) -> Tasks (tid)

FK: TeamsWorkOn (team_id) -> Teams (team_id)

FK: TeamsWorkOn (project_id) -> Projects (project_id)

Ограничения

Users (email - _@_.)

Users (username - _%)

Users (password - _%)

Users_Employees (email - _@_.)

Users_Employees (username - _%)

Users_Employees (password - _%)

Users_Employees (salary - > 0.0)

Users_Amateurs (email - _@_.)

Users_Amateurs (username - _%)

Users_Amateurs (password - _%)

Tasks (status – finished / unfinished)

Tasks (start_date >= 01.01.2023)

Executable_Tasks (deadline >= start_date)

Bugs (blocking_date >= 01.01.2023)

Bugs (unblocking_date >= blocking_date)

Projects (release_date - >= 01.01.2023)