# *Національний технічний університет України*

***«Київський політехнічний інститут»***

#### ***Факультет інформатики та обчислювальної техніки***

### ***Кафедра обчислювальної техніки***

## ПРОЕКТНА РОБОТА

*з дисципліни "****Основи Баз Данних****"*

***Виконав:***

*Щербина Микита*

***Факультет:*** *ІОТ*

***Група:***  *ІО-22,*

***Перевірив:*** *Болдак Андрій Олександрович*

***Київ - 2013р.***

1 . Запросы заинтересованных лиц

1.1 Введение

В данном документе описываются запросы заинтересованных лиц которыми выступает заказчик – компания «Space Mission Projects», для которого разрабатывается система програмно-целевого подхода для оценки целей поставленых аналитиками компании.

1.1.1 Цель

Целью документа является определить главные требования к функциональности , производительности , надежности , удобства , доступности , а также определить бизнес правила и технологические ограничения, накладываемые на предмет разработки.

1.1.2 Контекст

Перечень требований , указанных в данном документе, являются основой технического задания для разработки системы програмно-целевого подхода для оценки целей .

2.1 Краткий обзор продукта

Система програмно-целевого подхода для оценки поставленых задач - это база данных аналитиков, кандидатов и экспертов. Система содержит 3 различные уровня доступа . Кандидаты(первый уровень) могут подать заявку на отбор в эксперты, следить за проведением отборных испытаний.При прохождении отбора занимают должность эксперат. Аналитики (второй уровень ) могут добавлять задачи для их анализа, следить за процессом анализа, просматривать отчеты о работе экспертов. Эксперты (третий уровень ) анализируют задачи поставленные аналитиками, проводят отборочные конкурсы для кандидатов, могут вносить изменения , получать информацию и вести документацию о целях.

3.1 Деловые правила

3.1.1 Назначение системы

Система предназначена для сбора информации о поставленых целях с последующим ее анализом и подбором ресурсов для ее решения. Для передачи отчетов о проделанной работе.

3.1.2 Политика взаимоотношений с клиентом

Клиентами системы могут быть эксперты и аналитики работающих в компании «Space Mission Projects», или кандидаты в эксперты.

Политика взаимоотношений с клиентом системы заключается предоставлении ему разного рода информации , в основном с помощью веб - сервисов , с возможностью или невозможностью внесения в нее .

3.1.3 Характеристика делового процесса

Руководство системой не осуществляется. Система работает автономно после ее запуска.

Эксперты несут ответственность за базу ресурсов , хранение, удаление или модификацию информации о анализе задач.

Аналитики несут ответственность за создание , удаление или модификацию задач.

Кандидаты несут ответственность за подачу заявки на роль эксперта.

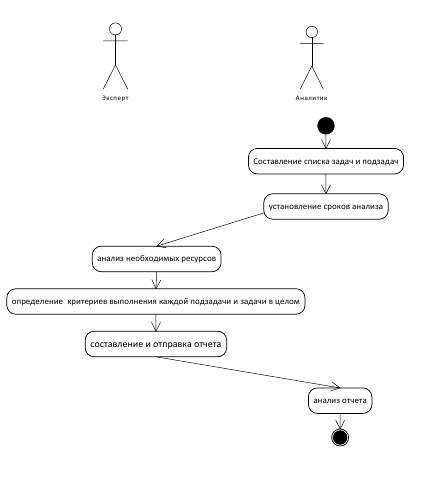
3.1.4 Сценарий регистрации нового пользователя

Пользователь регистрируется самостоятельно , заполняя регистрационную форму на сайте , вводит все необходимые данные ( логин, пароль, подтверждение пароля, имя, фамилия, отчество, документ определяющий статус, e - mail, контактный телефон , др. ).

3.1.5 Сценарии различных услуг системы

3.1.5.1 Эксперт - Аналитик

Аналитик составляет список задач и подзадач, определяет сроки анализа. Эксперт анализирует необходимые ресурсы, устанавливает критерии выполнения задач, формирует отчет. Аналитик анализирует отчет.



 3.1.5.2 Эксперт - Кандидат

Кандидат подает заявку на прохождение отбора в эксперты. Эксперт высылает отчет о зачислении кандидата на конкурсный отбор. Кандидат принимает участие в отборе. Эксперт принимают решение о зачислении кандидата, высылает результат отбора кандидату. При положительном результате отбора , кандидат вступает на должность эксперта.

4.1 Функциональность

Основные требования функциональности , предъявлены заинтересованными лицами к предмету разработки , относятся к таким категориям :

- Кандидат

- Аналитик

- Эксперт

4.1.1 Услуги для кандидатов

Для кандидатов система предоставляет следующие услуги:

* возможность подачи заявки
* возможность прохождения теста, для оценки знаний конкретной области

4.2.1 Услуги для аналитика

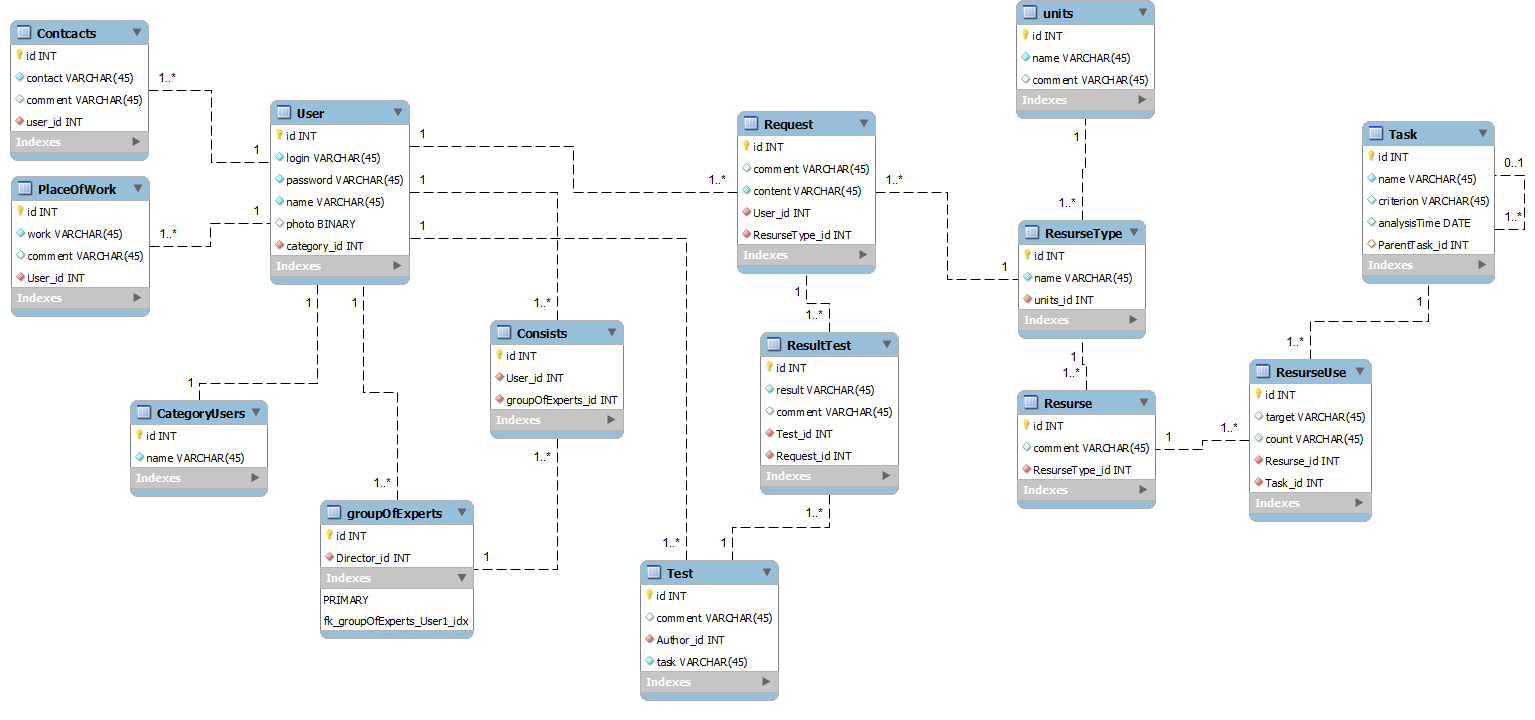
Для аналитиков система предоставляет следующие возможности:

* управление иерархией целей и задач
* получение отчета работы экспертов
* организация работы экспертов и экспертных групп

4.3.1 Услуги для эксперта

Для эксперта система предоставляет следующие возможности:

* предоставление исчерпывающей информации о задаче
* возможность редактировать базу ресурсов
* возможн ость составления тестов для оценки кандидатов
* возможность проверки тестов и выставления оценок кандидатам



**Создание таблиц**

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';

CREATE SCHEMA IF NOT EXISTS `db` DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci ;

USE `db` ;

-- -----------------------------------------------------

-- Table `db`.`CategoryUsers`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`CategoryUsers` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `db`.`User`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`User` (

`id` INT NOT NULL AUTO\_INCREMENT,

`login` VARCHAR(45) NOT NULL,

`password` VARCHAR(45) NOT NULL,

`name` VARCHAR(45) NOT NULL,

`photo` BINARY NULL,

`category\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_user\_category1\_idx` (`category\_id` ASC),

CONSTRAINT `fk\_user\_category1`

FOREIGN KEY (`category\_id`)

REFERENCES `db`.`CategoryUsers` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `db`.`Contcacts`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`Contcacts` (

`id` INT NOT NULL AUTO\_INCREMENT,

`contact` VARCHAR(45) NOT NULL,

`comment` VARCHAR(45) NULL,

`user\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_contcacts\_user\_idx` (`user\_id` ASC),

CONSTRAINT `fk\_contcacts\_user`

FOREIGN KEY (`user\_id`)

REFERENCES `db`.`User` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `db`.`PlaceOfWork`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`PlaceOfWork` (

`id` INT NOT NULL AUTO\_INCREMENT,

`work` VARCHAR(45) NOT NULL,

`comment` VARCHAR(45) NULL,

`User\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_placeOfWork\_user1\_idx` (`User\_id` ASC),

CONSTRAINT `fk\_placeOfWork\_user1`

FOREIGN KEY (`User\_id`)

REFERENCES `db`.`User` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `db`.`groupOfExperts`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`groupOfExperts` (

`id` INT NOT NULL AUTO\_INCREMENT,

`Director\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_groupOfExperts\_User1\_idx` (`Director\_id` ASC),

CONSTRAINT `fk\_groupOfExperts\_User1`

FOREIGN KEY (`Director\_id`)

REFERENCES `db`.`User` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `db`.`Consists`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`Consists` (

`id` INT NOT NULL AUTO\_INCREMENT,

`User\_id` INT NOT NULL,

`groupOfExperts\_id` INT NOT NULL,

INDEX `fk\_Consists\_User1\_idx` (`User\_id` ASC),

PRIMARY KEY (`id`),

INDEX `fk\_Consists\_groupOfExperts1\_idx` (`groupOfExperts\_id` ASC),

CONSTRAINT `fk\_Consists\_User1`

FOREIGN KEY (`User\_id`)

REFERENCES `db`.`User` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Consists\_groupOfExperts1`

FOREIGN KEY (`groupOfExperts\_id`)

REFERENCES `db`.`groupOfExperts` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `db`.`Test`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`Test` (

`id` INT NOT NULL AUTO\_INCREMENT,

`comment` VARCHAR(45) NULL,

`Author\_id` INT NOT NULL,

`task` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_Test\_User1\_idx` (`Author\_id` ASC),

CONSTRAINT `fk\_Test\_User1`

FOREIGN KEY (`Author\_id`)

REFERENCES `db`.`User` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `db`.`units`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`units` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NOT NULL,

`comment` VARCHAR(45) NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `db`.`ResurseType`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`ResurseType` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NOT NULL,

`units\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_ResurseType\_units1\_idx` (`units\_id` ASC),

CONSTRAINT `fk\_ResurseType\_units1`

FOREIGN KEY (`units\_id`)

REFERENCES `db`.`units` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `db`.`Request`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`Request` (

`id` INT NOT NULL AUTO\_INCREMENT,

`comment` VARCHAR(45) NULL,

`content` VARCHAR(45) NOT NULL,

`User\_id` INT NOT NULL,

`ResurseType\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_Request\_User1\_idx` (`User\_id` ASC),

INDEX `fk\_Request\_ResurseType1\_idx` (`ResurseType\_id` ASC),

CONSTRAINT `fk\_Request\_User1`

FOREIGN KEY (`User\_id`)

REFERENCES `db`.`User` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Request\_ResurseType1`

FOREIGN KEY (`ResurseType\_id`)

REFERENCES `db`.`ResurseType` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `db`.`ResultTest`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`ResultTest` (

`id` INT NOT NULL AUTO\_INCREMENT,

`result` VARCHAR(45) NOT NULL,

`comment` VARCHAR(45) NULL,

`Test\_id` INT NOT NULL,

`Request\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_ResultTest\_Test1\_idx` (`Test\_id` ASC),

INDEX `fk\_ResultTest\_Request1\_idx` (`Request\_id` ASC),

CONSTRAINT `fk\_ResultTest\_Test1`

FOREIGN KEY (`Test\_id`)

REFERENCES `db`.`Test` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_ResultTest\_Request1`

FOREIGN KEY (`Request\_id`)

REFERENCES `db`.`Request` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `db`.`Resurse`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`Resurse` (

`id` INT NOT NULL AUTO\_INCREMENT,

`comment` VARCHAR(45) NULL,

`ResurseType\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_Resurse\_ResurseType1\_idx` (`ResurseType\_id` ASC),

CONSTRAINT `fk\_Resurse\_ResurseType1`

FOREIGN KEY (`ResurseType\_id`)

REFERENCES `db`.`ResurseType` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `db`.`Task`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`Task` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NOT NULL,

`criterion` VARCHAR(45) NULL,

`analysisTime` DATE NULL,

`ParentTask\_id` INT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_Task\_Task1\_idx` (`ParentTask\_id` ASC),

CONSTRAINT `fk\_Task\_Task1`

FOREIGN KEY (`ParentTask\_id`)

REFERENCES `db`.`Task` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `db`.`ResurseUse`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `db`.`ResurseUse` (

`id` INT NOT NULL AUTO\_INCREMENT,

`target` VARCHAR(45) NULL,

`count` VARCHAR(45) NULL,

`Resurse\_id` INT NOT NULL,

`Task\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_ResurseUse\_Resurse1\_idx` (`Resurse\_id` ASC),

INDEX `fk\_ResurseUse\_Task1\_idx` (`Task\_id` ASC),

CONSTRAINT `fk\_ResurseUse\_Resurse1`

FOREIGN KEY (`Resurse\_id`)

REFERENCES `db`.`Resurse` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_ResurseUse\_Task1`

FOREIGN KEY (`Task\_id`)

REFERENCES `db`.`Task` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

USE `db` ;

-- -----------------------------------------------------

-- procedure routine1

-- -----------------------------------------------------

DELIMITER $$

USE `db`$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `routine1`()

BEGIN

END$$

DELIMITER ;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

**Заполнение таблиц**

/\* добавление типа пользователей \*/

/\*

insert into categoryusers (name) Value ("аналитик");

insert into categoryusers (name) Value ("эксперт");

insert into categoryusers (name) Value ("кандидат");

select \*

from categoryusers;

\*/

/\* добавление пользователей \*/

/\*

insert into user (login,password,name,category\_id,photo) value("Ivanov","password","Иванов Иван Иванович",1,null);

insert into user (login,password,name,category\_id,photo) value("Petrov","petrov","Петров Григорий Александрович",2,null);

insert into user (login,password,name,category\_id,photo) value("Sidorov","123456","Сидоров Максим Константинович",3,null);

insert into user (login,password,name,category\_id,photo) value("Ivanenko","IvArPet","Иваненко Артем Петрович",2,null);

insert into user (login,password,name,category\_id,photo) value("Irina","Irina","Григориева Ирина Петровна",3,null);

insert into user (login,password,name,category\_id,photo) value("Zuzua","Zuzua111","Зюзя Владимир Александрович",1,null);

\*/

/\* добавление контактов \*/

/\*

insert into contcacts (contact,user\_id) value ("+38068888888",1);

insert into contcacts (contact,user\_id) value ("+38068888887",1);

insert into contcacts (contact,user\_id) value ("+38050344232",2);

insert into contcacts (contact,user\_id) value ("+38068234209",3);

insert into contcacts (contact,user\_id) value ("+38069543492",6);

insert into contcacts (contact,user\_id) value ("+38092359022",5);

insert into contcacts (contact,user\_id) value ("+38023451412",4);

\*/

/\* добавление мест работы \*/

/\*

insert into placeofwork (work,User\_id) Value ("менеджер",1);

insert into placeofwork (work,User\_id) Value ("инженер",2);

insert into placeofwork (work,User\_id) Value ("стажер",3);

insert into placeofwork (work,User\_id) Value ("программист",4);

insert into placeofwork (work,User\_id) Value ("без работы",5);

insert into placeofwork (work,User\_id) Value ("менеджер",6);

\*/

/\* добавление группы экспертов \*/

/\*insert into groupofexperts (Director\_id) Value(1);

select \*

From groupofexperts;

\*/

/\* добавление составов групп экспертов \*/

/\*

insert into consists (user\_id,groupOfExperts\_id)value(2,7);

insert into consists (user\_id,groupOfExperts\_id) value(4,7);

select \*

From consists;

\*/

/\* добавление тестов \*/

/\*

insert into test (Author\_id,task) value (2,"Тест1");

insert into test (Author\_id,task) value (4,"Тест2");

insert into test (Author\_id,task) value (2,"Тест3");

insert into test (Author\_id,task) value (2,"Тест4");

select \*

From test;

\*/

/\* добавление единиц измерения\*/

/\*

insert into units (name) value ("метр");

insert into units (name) value ("час");

insert into units (name) value ("гривна");

insert into units (name) value ("человек");

insert into units (name) value ("штука");

select \*

From units;

\*/

/\* добавление типа ресурса\*/

/\*

insert into resursetype (name,units\_id) value("эксперт",4);

insert into resursetype (name,units\_id) value("сварщик",4);

insert into resursetype (name,units\_id) value("плотник",4);

insert into resursetype (name,units\_id) value("время",2);

insert into resursetype (name,units\_id) value("деньги",3);

insert into resursetype (name,units\_id) value("гвозди",5);

select \*

From resursetype;

\*/

/\* добавление заявки\*/

/\*

insert into request (user\_id,ResurseType\_id,content) value(3,1,"Хочу стать экспертом");

insert into request (user\_id,ResurseType\_id,content) value(6,3,"Хочу стать плотником");

select \*

From request;

\*/

/\* добавление результата тестирования\*/

/\*

insert into resulttest(test\_id,Request\_id,result) value(2,1,"Тест не пройден");

insert into resulttest(test\_id,Request\_id,result) value(3,2,"Тест пройден на отлично");

select \*

From resulttest;

\*/

/\* добавление ресурса\*/

/\*

insert into resurse (ResurseType\_id,comment) value(6,"100 мм - 5 штук");

insert into resurse (ResurseType\_id,comment) value(3,"1 плотник");

select \*

From resurse;

\*/

/\* добавление задачи\*/

/\*

insert into task (name,criterion,analysisTime,ParentTask\_id) value("поремонтировать стол","стол не шатается","01:00:00",null);

select \*

From task;

\*/

/\* добавление необходимых ресурсов для задачи\*/

/\*

insert into resurseuse (Resurse\_id,Task\_id,count,target) value (1,1,"5","починить стол");

insert into resurseuse (Resurse\_id,Task\_id,count,target) value (2,1,"1","починить стол");

select \*

From resurseuse;

\*/

**Результаты**

categoryusers

|  |  |
| --- | --- |
| id | name |
| 1 | аналитик |
| 2 | эксперт |
| 3 | кандидат |

consists

|  |  |  |
| --- | --- | --- |
| id | User\_id | groupOfExperts\_id |
| 7 | 2 | 7 |
| 8 | 4 | 7 |

contcacts

|  |  |  |  |
| --- | --- | --- | --- |
| id | contact | comment | user\_id |
| 1 | +38068888888 | | 1 |
| 2 | +38068888887 | | 1 |
| 3 | +38050344232 | | 2 |
| 4 | +38068234209 | | 3 |
| 5 | +38069543492 | | 6 |
| 6 | +38092359022 | | 5 |
| 7 | +38023451412 | | 4 |

groupofexperts

|  |  |
| --- | --- |
| id | Director\_id |
| 7 | 1 |

placeofwork

|  |  |  |  |
| --- | --- | --- | --- |
| id | work | comment | User\_id |
| 1 | менеджер | | 1 |
| 2 | инженер |  | 2 |
| 3 | стажер |  | 3 |
| 4 | программист | | 4 |
| 5 | без работы | | 5 |
| 6 | менеджер | | 6 |

request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | comment | content | User\_id | ResurseType\_id |
| 1 |  | Хочу стать экспертом | 3 | 1 |
| 2 |  | Хочу стать плотником | 6 | 3 |

resulttest

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | result | comment | Test\_id | Request\_id |
| 1 | Тест не пройден | | 2 | 1 |
| 2 | Тест пройден на отлично | | 3 | 2 |

resurse

|  |  |  |
| --- | --- | --- |
| id | comment | ResurseType\_id |
| 1 | 100 мм - 5 штук | 6 |
| 2 | 1 плотник | 3 |

resursetype

|  |  |  |
| --- | --- | --- |
| id | name | units\_id |
| 1 | эксперт | 4 |
| 2 | сварщик | 4 |
| 3 | плотник | 4 |
| 4 | время | 2 |
| 5 | деньги | 3 |
| 6 | гвозди | 5 |

