

## Лабораторная работа №1

«Создание распределённой базы данных»

по курсу «Распределённые информационные системы и базы данных»

*Факультет:*

ПМИ

*Группа:*

ПММ-81

*Студенты:*

Михайлов А. А.,  
Санина А. А.

*Преподаватель:*

Курлаев С. А.

# 1. Описание использованной базы данных с точки зрения предметной области

## 1.1. Цели и задачи

Цель: повышение эффективности функционирования ИТ-подразделения производственного предприятия (НИИ Физиологии СО РАМН).

Основные задачи: автоматизация процесса управления инцидентами, создание прозрачной системы учета и контроля обращений пользователей в службу технической поддержки, анализ и выявление “узких мест”, оптимизации деятельности службы технической поддержки по предоставлению услуг подразделениям.

## 1.2. Назначение и функциональность системы

В составе Института 14 лабораторий и клиника психосоматического профиля.

На обслуживание медицинской техники заключен договор с подрядной организацией.

Офисные АРМ (автоматизированные рабочие места) врачей и научных сотрудников обслуживаются собственными силами (инженерами НИИ) в первую очередь, то есть их заявки на оказание технической поддержки имеют приоритет перед заявками прочих сотрудников.

Каждый сотрудник помимо телефонного звонка и электронной почты может оставить свою заявку на сайте технической поддержки института, зарегистрировав ее с указанием:

- проблемы (краткое описание возникшей неполадки);
- приоритета (выставить предполагаемый приоритет выполнения в зависимости от характера неполадки).

Все заявки после регистрации просматриваются инженером, и в случае обнаружения явного несоответствия действительного и указанного приоритетов, заявка помещается в конец очереди. Приоритеты соответствуют приведенной ниже таблице (Таблица 1):

Таблица 1. Приоритеты входящих заявок для научных сотрудников

Приоритет	Описание	Время реагирования (ч.)
1. Самый высокий	Работа системы нарушена в целом. Ничто не может быть выполнено.	0 — 1
2. Высокий	Сбои при работе системы в режиме эксплуатации. Система функционирует частично, при этом часть работ могут выполняться.	1 — 6
3. Нормальный	Работа в системе встречается с некритическими проблемами или дефектами, вопросами, возникающими при эксплуатации ПО, настройке ПО и т.д.	1 — 12
4. Низкий	Минимальные воздействия на функционирование системы, не критические проблемы. Запросы на получение консультаций.	1 — 24
5. Самый низкий	Пользователя беспокоит какая-либо несущественная проблема, позволяющая системе функционировать в полном объеме.	1 — 48

Поступившие обращения автоматически регистрируются, и в ответ высылается письмо, содержащее в поле “Тема” регистрационный номер вида [Incident 123456]. Это служит подтверждением регистрации. Заявителю необходимо сохранять этот номер в дальнейшей переписке. Он служит идентификатором, по которому можно получить историю обработки и текущее состояние инцидента.

Кроме того, рекомендуется оформлять каждый инцидент (вопрос, сбой) отдельным письмом-обращением, это улучшит его обработку.

Время реакции на обращение зависит от приоритета инцидента: от 4 (низший) до 1 (высший). Приоритет устанавливается службой технической поддержки на основании информации, предоставленной обращении. Пользователь может повысить приоритет, предоставив дополнительную информацию в “комментарии”.

Техническая поддержка предоставляется посредством личного посещения заявителя работником службы.

Основным каналом взаимодействия в процессе обработки обращения является электронная почта. Другие средства электронных коммуникаций и удаленного доступа применяются в ходе обработки инцидента в случае необходимости и по согласованию сторон.

Телефон используется только в справочных целях и как дополнительная возможность ускорения обработки инцидента по инициативе службы технической поддержки.

После получения ответа от службы технической поддержки, заявка закрывается техником, а на электронную почту заявителя высылается уведомительное письмо.

*Информационная система (ИС) должна реализовывать следующие основные функции:*

- обеспечение надежности хранения информации;
- ИС должна быть реализована по клиент-серверной архитектуре с возможностью резервного копирования данных;
- в качестве платформы серверной части предполагается использовать MySQL Server;
- обеспечение единой “точки входа” при обращении в службу технической поддержки;
- создание учетной записи заявки должно осуществляться самим пользователем в едином стандартизованном интерфейсе;
- обеспечение возможности структурирования поступающих заявок по различным классификаторам (категориям, степени сложности, важности и т.д.), а также изменение состояния заявки в процессе ее обработки;
- обеспечение возможности просмотра поступивших заявок пользователей с учетом их статуса.

### **1.3. Характеристика объекта исследования**

Институт физиологии СО РАМН. Учреждение имеет в своём составе 14 лабораторий и клинику психосоматического профиля. В Институте работает более 250 человек, среди которых доктора, кандидаты наук и просто отлично сведущие в высокой медицине люди, хорошо разбирающиеся в человеческих внутренностях, но не компьютерных.

Для продуктивной работы и научных открытий во имя спасения жизней пациентов им просто необходим парк постоянно и безостановочно функционирующего оборудования. Причём любой сбой в железной системе должен восстанавливаться в кратчайшие сроки и с наименьшей потерей информации (в идеале таких потерь вообще следует избегать).

Именно для повсеременного функционирования парка и следует создать штат “помощников” парка — квалифицированную и моментально-реагирующую на любые просьбы службу технической поддержки. С этой службой должно быть удобно связаться любым возможным в сложившейся ситуации способом и максимально-быстро получить ответ на возникший вопрос.

Взаимодействие пользователя, техника и администратора сводится к следующей схеме. Сотрудник СО РАМН, обнаружив любую неполадку на личном АРМ, авторизуется в системе и отправляет заявку в службу технической поддержки, регистрируя её на сайте. В случае успешной регистрации, на почту приходит уведомительное письмо.

Свободный сотрудник технической поддержки берёт заявку из очереди на исполнение. Анализируя проблему, в случае необходимости связывается за получением дополнительной информации с сотрудником СО РАМН, или, при возникновении спорных вопросов, за помощью к администратору. После разрешения инцидента сотрудник ТП закрывает заявку. Все действия, связанные с изменением приоритета или статуса заявки, сопровождаются уведомительным письмом пользователю.

Администратор ТП консультирует техников, отслеживает корректную работу персонала, наделяет пользователей правами и в случае необходимости, вносит данные в БД.

Общий механизм работы системы можно представить в виде диаграммы прецедентов ( Рисунок 1):

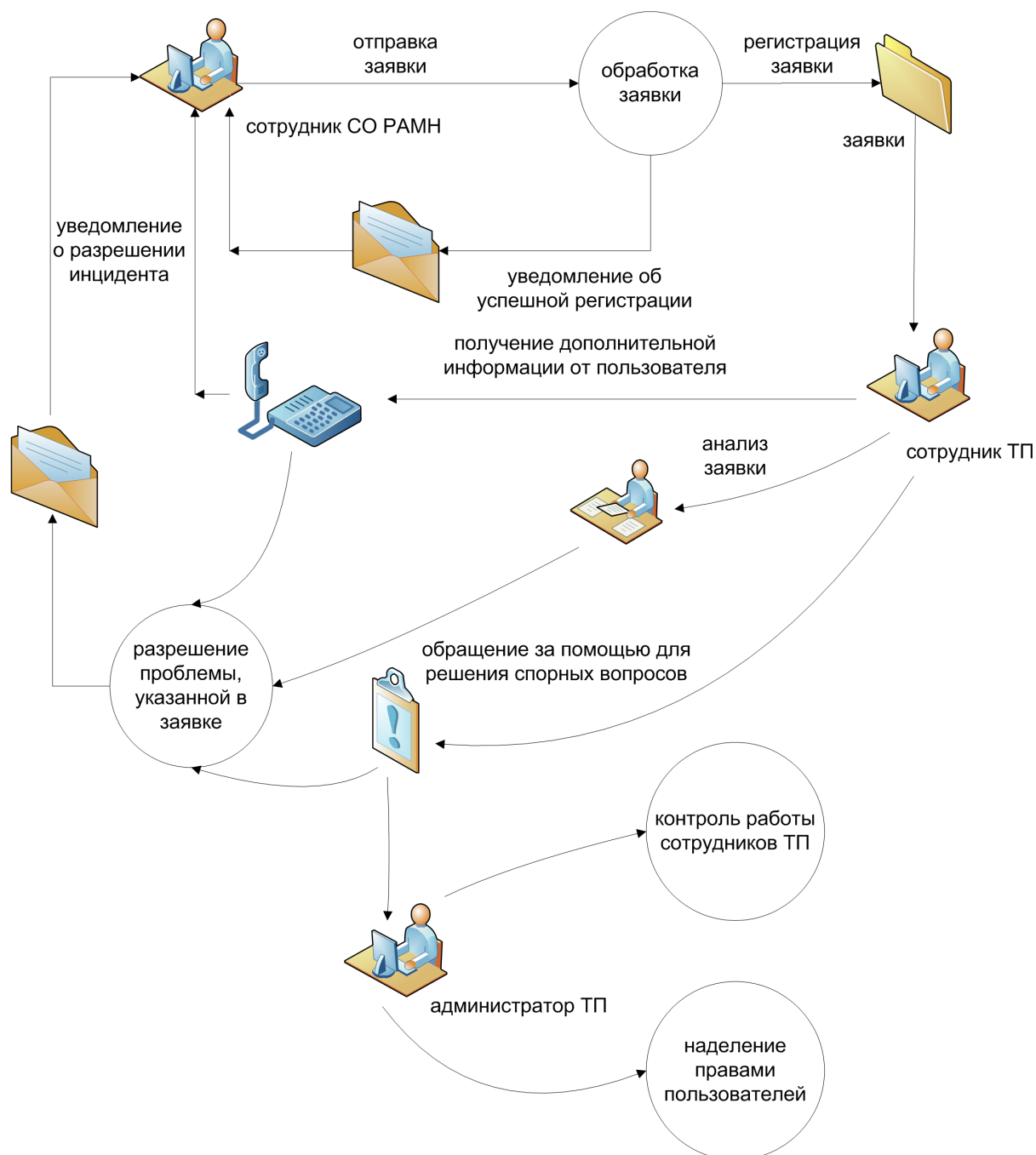


Рис. 1. Диаграмма прецедентов

#### 1.4. Организационная структура объекта автоматизации

Основная цель процесса — максимально быстрое восстановление нормального функционирования ИТ-услуг и минимизация неблагоприятного воздействия на функционирование бизнеса и, соответственно, обеспечение наилучшего из возможных уровня качества и доступности услуг. Под “нормальным функционированием” будем подразумевать функционирование в рамках Service Level Agreement (SLA).

В терминологии ITIL *инцидент (incident)* определяется как любое событие, не являющееся частью стандартных операций по предоставлению услуг, которое повлекло или могло повлечь прерывание или снижение качества этой услуги. Частный случай инцидента — запрос на обслуживание (Service Request) — инцидент, не являющийся сбоем ИТ-инфраструктуры.

Преимущества использования процесса управления инцидентами:

- посредством установки ограничений по времени снижается уязвимость бизнеса для инцидентов;
- повышается доступность необходимой бизнесу информации;
- появляется возможность определения полезных исправлений и улучшений системы;
- улучшенный мониторинг производительности отдела поддержки;

- лучшее использование персонала;
- избежание потерянных или некорректных инцидентов;
- повышение удовлетворенности клиентов.

Этапы процесса управления инцидентами:

- 1) прием и регистрация инцидента;
- 2) первоначальная классификация и поддержка;
- 3) исследование и диагностика;
- 4) решение и восстановление;
- 5) закрытие инцидента.

## 1.5. Структура базы данных на языке SQL с указанием имен и типов полей всех таблиц

```

1 BEGIN;
2 CREATE TABLE 'admins' (
3     'employee_id' integer NOT NULL PRIMARY KEY
4 )
5 ;
6 CREATE TABLE 'department_activity_sphere' (
7     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
8     'name' varchar(255) NOT NULL UNIQUE
9 )
10 ;
11 CREATE TABLE 'department' (
12     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
13     'name' varchar(255) NOT NULL UNIQUE,
14     'phone' varchar(32),
15     'email' varchar(75) NOT NULL,
16     'addr' varchar(512) NOT NULL,
17     'exists_now' bool NOT NULL,
18     'activity_sphere_id' integer NOT NULL
19 )
20 ;
21 ALTER TABLE 'department' ADD CONSTRAINT 'activity_sphere_id_refs_id_cac38110' FOREIGN KEY ('activity_sphere_id')
22 CREATE TABLE 'detail_category' (
23     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
24     'name' varchar(255) NOT NULL UNIQUE
25 )
26 ;
27 CREATE TABLE 'detail_model' (
28     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
29     'name' varchar(255) NOT NULL UNIQUE,
30     'category_id' integer NOT NULL
31 )
32 ;
33 ALTER TABLE 'detail_model' ADD CONSTRAINT 'category_id_refs_id_7151611f' FOREIGN KEY ('category_id') REFERENCES
34 CREATE TABLE 'employee_role' (
35     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
36     'name' varchar(255) NOT NULL UNIQUE
37 )
38 ;
39 CREATE TABLE 'employee' (
40     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
41     'snils' varchar(16) NOT NULL UNIQUE,
42     'name' varchar(255) NOT NULL,
43     'phone' varchar(32) NOT NULL,
44     'addr' varchar(512) NOT NULL,
45     'login' varchar(64) UNIQUE,
46     'password' varchar(128),
47     'role_id' integer NOT NULL
48 )
49 ;
50 ALTER TABLE 'employee' ADD CONSTRAINT 'role_id_refs_id_c1a88009' FOREIGN KEY ('role_id') REFERENCES 'employee_role'
51 ALTER TABLE 'admins' ADD CONSTRAINT 'employee_id_refs_id_4e8e850e' FOREIGN KEY ('employee_id') REFERENCES 'employee'
52 CREATE TABLE 'employee_operation_type' (
53     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
54     'name' varchar(255) NOT NULL UNIQUE
55 )
56 ;
57 CREATE TABLE 'employee_operation' (
58     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
59     'date' date NOT NULL,
60     'type_id' integer NOT NULL,
61     'employee_id' integer NOT NULL,
62     'department_id' integer NOT NULL
63 )
64 ;
65 ALTER TABLE 'employee_operation' ADD CONSTRAINT 'employee_id_refs_id_2bed7358' FOREIGN KEY ('employee_id') REFERENCES 'employee'
66 ALTER TABLE 'employee_operation' ADD CONSTRAINT 'type_id_refs_id_e2086d2c' FOREIGN KEY ('type_id') REFERENCES 'employee_operation_type'
67 ALTER TABLE 'employee_operation' ADD CONSTRAINT 'department_id_refs_id_a61f5496' FOREIGN KEY ('department_id') REFERENCES 'department'

```

```

68 CREATE TABLE 'equipment_category' (
69     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
70     'name' varchar(255) NOT NULL UNIQUE
71 )
72 ;
73 CREATE TABLE 'equipment_model' (
74     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
75     'name' varchar(255) NOT NULL UNIQUE,
76     'category_id' integer NOT NULL
77 )
78 ;
79 ALTER TABLE 'equipment_model' ADD CONSTRAINT 'category_id_refs_id_68b4f741' FOREIGN KEY ('category_id') REFERENCES
80 CREATE TABLE 'equipment' (
81     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
82     'name' varchar(255) NOT NULL,
83     'serial_number' varchar(255) NOT NULL UNIQUE,
84     'addr' varchar(512),
85     'equipment_model_id' integer NOT NULL
86 )
87 ;
88 ALTER TABLE 'equipment' ADD CONSTRAINT 'equipment_model_id_refs_id_76bbc77d' FOREIGN KEY ('equipment_model_id') R
89 CREATE TABLE 'equipment_operation_type' (
90     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
91     'name' varchar(255) NOT NULL UNIQUE
92 )
93 ;
94 CREATE TABLE 'equipment_operation' (
95     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
96     'detail_price' double precision,
97     'datetime' datetime NOT NULL,
98     'equipment_id' integer NOT NULL,
99     'eq_oper_type_id' integer NOT NULL
100 )
101 ;
102 ALTER TABLE 'equipment_operation' ADD CONSTRAINT 'equipment_id_refs_id_8da693a5' FOREIGN KEY ('equipment_id') REF
103 ALTER TABLE 'equipment_operation' ADD CONSTRAINT 'eq_oper_type_id_refs_id_50af72b9' FOREIGN KEY ('eq_oper_type_id
104 CREATE TABLE 'equipment_owner' (
105     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
106     'start_datetime' datetime NOT NULL,
107     'finish_datetime' datetime,
108     'equipment_id' integer NOT NULL,
109     'employee_id' integer NOT NULL
110 )
111 ;
112 ALTER TABLE 'equipment_owner' ADD CONSTRAINT 'equipment_id_refs_id_ab3ab6a4' FOREIGN KEY ('equipment_id') REFEREN
113 ALTER TABLE 'equipment_owner' ADD CONSTRAINT 'employee_id_refs_id_e8c12184' FOREIGN KEY ('employee_id') REFERENCES
114 CREATE TABLE 'task_priority' (
115     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
116     'name' varchar(255) NOT NULL UNIQUE
117 )
118 ;
119 CREATE TABLE 'task_equipment' (
120     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
121     'task_id' integer NOT NULL,
122     'equipment_id' integer NOT NULL,
123     UNIQUE ('task_id', 'equipment_id')
124 )
125 ;
126 ALTER TABLE 'task_equipment' ADD CONSTRAINT 'equipment_id_refs_id_28b1e439' FOREIGN KEY ('equipment_id') REFERENC
127 CREATE TABLE 'task' (
128     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
129     'name' varchar(255) NOT NULL,
130     'datetime' datetime NOT NULL,
131     'priority_id' integer NOT NULL,
132     'client_id' integer NOT NULL,
133     'owner_id' integer
134 )
135 ;
136 ALTER TABLE 'task' ADD CONSTRAINT 'priority_id_refs_id_6f001401' FOREIGN KEY ('priority_id') REFERENCES 'task_pri
137 ALTER TABLE 'task' ADD CONSTRAINT 'client_id_refs_id_138e7f27' FOREIGN KEY ('client_id') REFERENCES 'employee' (
138 ALTER TABLE 'task' ADD CONSTRAINT 'owner_id_refs_id_138e7f27' FOREIGN KEY ('owner_id') REFERENCES 'employee' (
139 ALTER TABLE 'task_equipment' ADD CONSTRAINT 'task_id_refs_id_eb7165db' FOREIGN KEY ('task_id') REFERENCES 'task'
140 CREATE TABLE 'repair' (
141     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
142     'comment' varchar(512) NOT NULL,
143     'datetime' datetime NOT NULL,
144     'detail_model_id' integer NOT NULL,
145     'equipment_operation_id' integer NOT NULL,
146     'task_id' integer NOT NULL
147 )
148 ;
149 ALTER TABLE 'repair' ADD CONSTRAINT 'equipment_operation_id_refs_id_f486654f' FOREIGN KEY ('equipment_operation_id
150 ALTER TABLE 'repair' ADD CONSTRAINT 'detail_model_id_refs_id_c5ba20e8' FOREIGN KEY ('detail_model_id') REFERENCES
151 ALTER TABLE 'repair' ADD CONSTRAINT 'task_id_refs_id_95a2d6c7' FOREIGN KEY ('task_id') REFERENCES 'task' ('id');
152 CREATE TABLE 'task_state' (
153     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
154     'name' varchar(255) NOT NULL UNIQUE
155 )
156 ;
157 CREATE TABLE 'task_operation' (
158     'id' integer AUTO INCREMENT NOT NULL PRIMARY KEY,
159     'work_price' double precision,
160     'datetime' datetime NOT NULL,
161     'task_id' integer NOT NULL,

```

```

162     'technic_id' integer NOT NULL,
163     'state_id' integer NOT NULL
164 )
165 ;
166 ALTER TABLE 'task_operation' ADD CONSTRAINT 'technic_id_refs_id_bf4be40c' FOREIGN KEY ('technic_id') REFERENCES '
167 ALTER TABLE 'task_operation' ADD CONSTRAINT 'state_id_refs_id_1212148b' FOREIGN KEY ('state_id') REFERENCES 'task
168 ALTER TABLE 'task_operation' ADD CONSTRAINT 'task_id_refs_id_940884b5' FOREIGN KEY ('task_id') REFERENCES 'task'
169 CREATE TABLE 'technics' (
170     'employee_id' integer NOT NULL PRIMARY KEY
171 )
172 ;
173 ALTER TABLE 'technics' ADD CONSTRAINT 'employee_id_refs_id_d7240997' FOREIGN KEY ('employee_id') REFERENCES 'empl
174 CREATE INDEX 'department_1dea4842' ON 'department' ('activity_sphere_id');
175 CREATE INDEX 'detail_model_b583a629' ON 'detail_model' ('category_id');
176 CREATE INDEX 'employee_84566833' ON 'employee' ('role_id');
177 CREATE INDEX 'employee_operation_94757cae' ON 'employee_operation' ('type_id');
178 CREATE INDEX 'employee_operation_dcc97e32' ON 'employee_operation' ('employee_id');
179 CREATE INDEX 'employee_operation_bf691be4' ON 'employee_operation' ('department_id');
180 CREATE INDEX 'equipment_model_b583a629' ON 'equipment_model' ('category_id');
181 CREATE INDEX 'equipment_83ff7b4a' ON 'equipment' ('equipment_model_id');
182 CREATE INDEX 'equipment_operation_997b9956' ON 'equipment_operation' ('equipment_id');
183 CREATE INDEX 'equipment_operation_7ba3dec2' ON 'equipment_operation' ('eq_oper_type_id');
184 CREATE INDEX 'equipment_owner_997b9956' ON 'equipment_owner' ('equipment_id');
185 CREATE INDEX 'equipment_owner_dcc97e32' ON 'equipment_owner' ('employee_id');
186 CREATE INDEX 'task_8fb0ef36' ON 'task' ('priority_id');
187 CREATE INDEX 'task_2bfe9d72' ON 'task' ('client_id');
188 CREATE INDEX 'task_5e7b1936' ON 'task' ('owner_id');
189 CREATE INDEX 'repair_3c11de5c' ON 'repair' ('detail_model_id');
190 CREATE INDEX 'repair_694a0d3b' ON 'repair' ('equipment_operation_id');
191 CREATE INDEX 'repair_57746cc8' ON 'repair' ('task_id');
192 CREATE INDEX 'task_operation_57746cc8' ON 'task_operation' ('task_id');
193 CREATE INDEX 'task_operation_043f54d9' ON 'task_operation' ('technic_id');
194 CREATE INDEX 'task_operation_d5582625' ON 'task_operation' ('state_id');
195
196 COMMIT;

```

## 1.6. ER-диаграмма

Всего объектов в базе: 284 981.

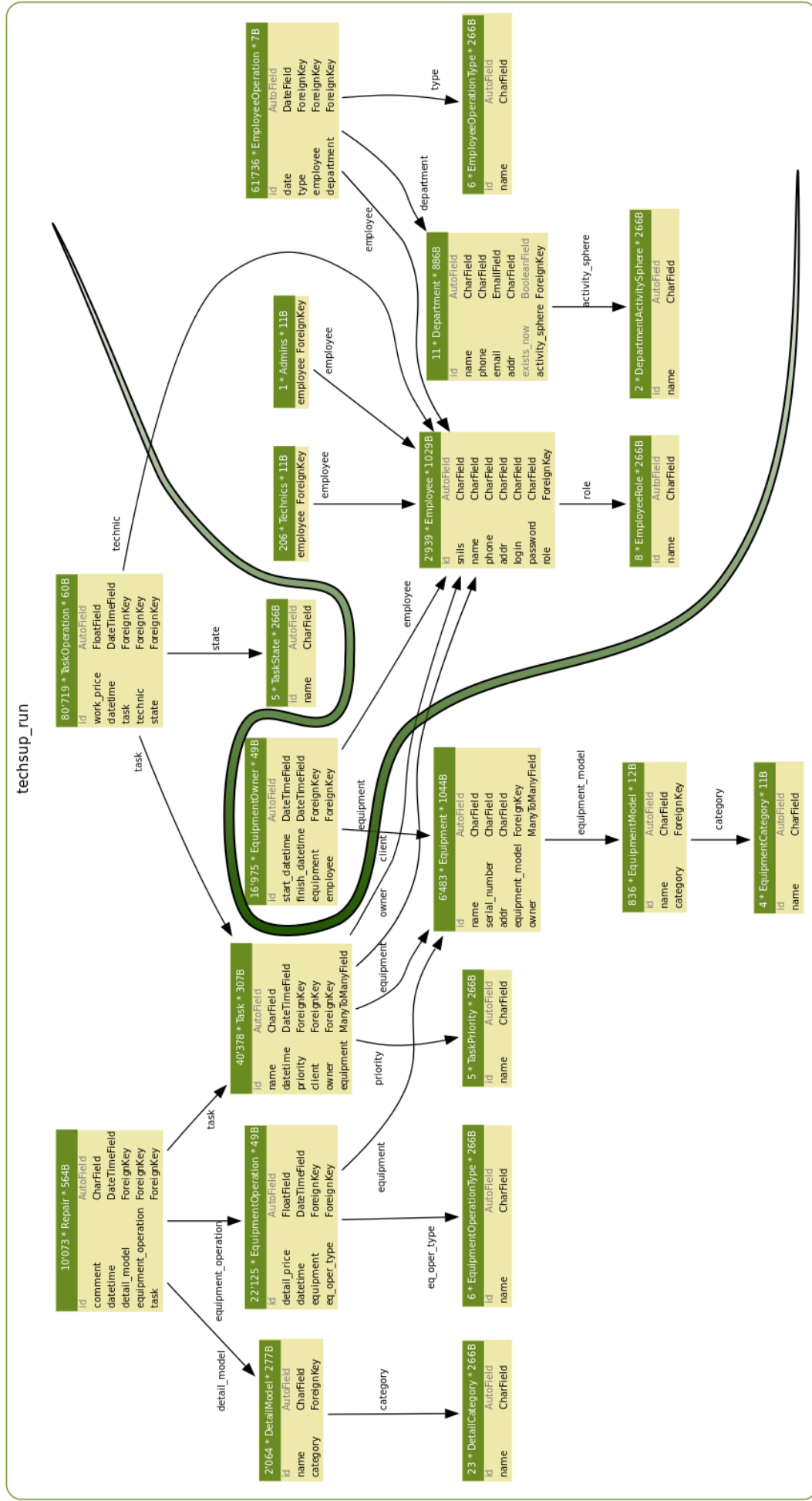


Рис. 2. ER-диаграмма



## 2. SQL-запросы для обоих вариантов разбиения баз данных

```
1 DELIMITER $$
2
3 DROP PROCEDURE IF EXISTS change_equipment_owner$$
4 CREATE PROCEDURE change_equipment_owner (
5     IN serial_number VARCHAR(128),
6     IN in_snils        VARCHAR(16)
7 )
8 BEGIN
9
10 START TRANSACTION;
11
12 -- сменить пользователя оборудования
13
14 INSERT INTO equipment_owner( equipment_id, employee_id )
15 VALUES(
16     (
17         SELECT equipment.id
18         FROM equipment
19         WHERE LOWER( equipment.serial_number ) LIKE serial_number
20     ),
21     (
22         SELECT employee.id
23         FROM employee
24         WHERE LOWER(employee.snils) LIKE in_snils
25     )
26 );
27
28 COMMIT;
29
30 END$$
31
32 --
33
34 DROP PROCEDURE IF EXISTS change_task_owner$$
35 CREATE PROCEDURE change_task_owner(
36     IN task_id INT(11),
37     IN in_snils VARCHAR(128)
38 )
39 BEGIN
40
41 START TRANSACTION;
42
43 -- изменить куратора заявки
44
45 UPDATE task
46 SET owner_id = (
47     SELECT employee.id
48     FROM employee
49     WHERE LOWER(employee.snils) LIKE in_snils
50 )
51 WHERE task.id = task_id;
52
53 COMMIT;
54
55 END$$
56
57 --
58
59 DROP PROCEDURE IF EXISTS delete_department$$
60 CREATE PROCEDURE delete_department(
61     IN name VARCHAR( 128 )
62 )
63 BEGIN
64
65 START TRANSACTION;
66
67 -- удалить организацию
68
69 UPDATE department
70 SET exists_now = FALSE
71 WHERE LOWER( department.name ) LIKE name;
72
73 COMMIT;
74
75 END$$
76
77 --
78
79 DROP PROCEDURE IF EXISTS get_equipment_operation$$
80 CREATE PROCEDURE get_equipment_operation(
81     IN serial_number VARCHAR( 128 )
82 )
83 BEGIN
84
85 -- получить информацию о всех операциях
86 -- с конкретным оборудованием
87
88 SELECT temp_eq_oper.id, temp_eq_oper.detail_price,
89        temp_eq_oper.datetime, temp_eq_oper.operation,
90        tmp.repair_id,
91        detail_model.name AS detail_model,
```

```

92         tmp.comment,
93         tmp.datetime AS repair_datetime,
94         employee.name, employee.phone
95 FROM (
96     SELECT task_operation.task_id, technic_id, MAX( task_operation.datetime )
97     FROM task_operation
98     GROUP BY task_operation.task_id
99 ) AS tmp1
100 INNER JOIN (
101     SELECT repair.id AS repair_id,
102            repair.equipment_operation_id,
103            repair.detail_model_id,
104            repair.comment,
105            repair.datetime,
106            repair.task_id
107     FROM repair
108     WHERE repair.equipment_operation_id IN (
109         SELECT id
110         FROM equipment_operation
111         WHERE equipment_id = (
112             SELECT equipment.id
113             FROM equipment
114             WHERE LOWER( equipment.serial_number ) LIKE serial_number
115         ) AND eq_oper_type_id = (
116             SELECT equipment_operation_type.id
117             FROM equipment_operation_type
118             WHERE equipment_operation_type.name = 'ремонт' )
119     )
120 ) AS tmp
121 ON tmp1.task_id = tmp.task_id
122 INNER JOIN employee
123 ON technic_id = employee.id
124 INNER JOIN detail_model
125 ON detail_model_id = detail_model.id
126 RIGHT JOIN (
127     -- получить все операции с оборудованием по коду
128     SELECT equipment_operation.id,
129            equipment_operation.eq_oper_type_id, equipment_operation.detail_price,
130            equipment_operation.datetime, equipment_operation_type.name AS operation
131     FROM equipment_operation, equipment_operation_type
132     WHERE equipment_operation_type.id = equipment_operation.eq_oper_type_id AND
133            equipment_operation.equipment_id = (
134                SELECT equipment.id
135                FROM equipment
136                WHERE LOWER( equipment.serial_number ) LIKE serial_number
137            )
138     ORDER BY equipment_operation.datetime
139 ) AS temp_eq_oper
140 ON temp_eq_oper.id = equipment_operation_id;
141
142 END$$
143
144 -- -----
145
146 DROP PROCEDURE IF EXISTS get_equipment_owner$$
147 CREATE PROCEDURE get_equipment_owner(
148     IN serial_number VARCHAR( 128 )
149 )
150 BEGIN
151
152     -- получить всех владельцев конкретного оборудования
153     -- с последним изменением статуса (принят/уволен)
154
155     SELECT employee.id AS employee_id, employee.name,
156            employee_role.name AS role,
157            employee.phone, employee.email,
158            tmp2.state, tmp2.date
159     FROM employee
160     LEFT JOIN employee_role
161     ON employee.role_id = employee_role.id
162     RIGHT JOIN (
163         SELECT *
164         FROM (
165             -- id сотрудников с последними изменёнными статусами
166             SELECT employee_operation.employee_id,
167                    employee_operation_type.name AS state,
168                    employee_operation.date
169             FROM employee_operation
170             INNER JOIN (
171                 SELECT employee_operation.employee_id, MAX( employee_operation.date ) AS date
172                 FROM employee_operation
173                 GROUP BY employee_operation.employee_id
174             ) AS tmp1
175             ON employee_operation.employee_id = tmp1.employee_id AND
176                employee_operation.date = tmp1.date
177             INNER JOIN employee_operation_type
178             ON type_id = employee_operation_type.id
179         ) AS tmp
180     WHERE tmp.employee_id IN (
181         SELECT employee_id
182         FROM equipment_owner
183         WHERE equipment_owner.equipment_id = (
184             SELECT equipment.id
185             FROM equipment

```

```

186         WHERE LOWER( equipment.serial_number ) LIKE serial_number
187     )
188 ) AS tmp2
189 ON employee.id = tmp2.employee_id
190 ORDER BY name;
191
192 END$$
193
194
195 -----
196
197 DROP PROCEDURE IF EXISTS get_equipment_sum_detail_price$$
198 CREATE PROCEDURE get_equipment_sum_detail_price(
199     IN serial_number VARCHAR( 128 )
200 )
201 BEGIN
202
203     -- получить расходы по закупке деталей
204     -- для конкретного оборудования
205
206     SELECT tmp.sum_detail_price
207 FROM (
208     SELECT equipment_id, SUM( detail_price ) AS sum_detail_price
209     FROM equipment_operation
210     GROUP BY equipment_id
211 ) AS tmp
212 WHERE tmp.equipment_id = (
213     SELECT id
214     FROM equipment
215     WHERE LOWER( equipment.serial_number ) LIKE serial_number
216 );
217
218 END$$
219
220 -----
221
222 DROP PROCEDURE IF EXISTS get_equipment_sum_work_price$$
223 CREATE PROCEDURE get_equipment_sum_work_price(
224     IN serial_number VARCHAR( 128 )
225 )
226 BEGIN
227
228     -- получить расходы на работу мастеров
229     -- для конкретного оборудования
230
231     SELECT SUM( work_price ) AS sum_work_price
232 FROM (
233     SELECT task_operation.work_price
234     FROM task_operation
235     INNER JOIN (
236         SELECT task_equipement.task_id
237         FROM task_equipement
238         WHERE equipment_id = (
239             SELECT id
240             FROM equipment
241             WHERE LOWER( equipment.serial_number ) LIKE serial_number
242         )
243     ) AS tmp
244     ON tmp.task_id = task_operation.task_id
245 ) AS tmp1;
246
247 END$$
248
249 -----
250
251 DROP PROCEDURE IF EXISTS get_task_queue_user$$
252 CREATE PROCEDURE get_task_queue_user( )
253 BEGIN
254
255     -- посмотреть очередь заявок с последним
256     -- изменённым статусом для пользователя
257
258 DROP TABLE IF EXISTS tmp_task;
259 CREATE TEMPORARY TABLE tmp_task AS
260     SELECT task.id, task_priority.name AS priority, task.name AS task
261     FROM task
262     INNER JOIN task_priority
263     ON task.priority_id = task_priority.id;
264
265 DROP TABLE IF EXISTS tmp;
266 CREATE TEMPORARY TABLE tmp AS
267     SELECT task_operation.task_id,
268         employee.name,
269         task_state.name AS state, task_operation.datetime
270 FROM task_operation
271 INNER JOIN (
272     SELECT task_operation.task_id, MAX( task_operation.datetime ) AS date
273     FROM task_operation
274     GROUP BY task_operation.task_id
275 ) AS temp
276 ON task_operation.task_id = temp.task_id AND
277 task_operation.datetime = temp.date
278 LEFT JOIN employee
279 ON technic_id = employee.id

```

```

280     INNER JOIN task_state
281     ON state_id = task_state.id;
282
283     SELECT tmp.task_id,
284            task_priority.name AS priority,
285            task.name AS task,
286            tmp.name AS technic,
287            tmp.state, tmp.datetime
288     FROM tmp
289     INNER JOIN task
290     ON tmp.task_id = task.id
291     INNER JOIN task_priority
292     ON priority_id = task_priority.id
293     ORDER BY tmp.datetime;
294
295 END$$
296
297 --
298
299 DROP PROCEDURE IF EXISTS put_new_task$$
300 CREATE PROCEDURE put_new_task(
301     IN task_name      VARCHAR( 128 ),
302     IN priority_id    INT( 11 ),
303     IN login          VARCHAR ( 64 ),
304     IN password       VARCHAR ( 128 ),
305     IN in_datetime    DATETIME,
306     IN serial_number  VARCHAR( 128 )
307 )
308 BEGIN
309
310     START TRANSACTION;
311
312     -- добавить новую заявку
313
314     INSERT INTO task ( name, datetime, priority_id, client_id )
315     VALUES( task_name,
316            in_datetime,
317            priority_id,
318            (
319                SELECT id
320                FROM employee
321                WHERE employee.login LIKE login AND
322                      employee.password LIKE password
323            )
324     );
325
326     INSERT INTO task_equipment ( task_id, equipment_id )
327     VALUES(
328         (
329             SELECT id
330             FROM task
331             WHERE task.datetime = (
332                 SELECT MAX( datetime )
333                 FROM task
334             )
335         ),
336         (
337             SELECT id
338             FROM equipment
339             WHERE LOWER( equipment.serial_number ) LIKE serial_number
340         )
341     );
342
343     COMMIT;
344
345 END$$
346
347 --
348
349 DROP PROCEDURE IF EXISTS add_task_owner$$
350 CREATE PROCEDURE add_task_owner(
351     IN in_task_id     INT(11),
352     IN in_login        VARCHAR ( 64 ),
353     IN in_password     VARCHAR ( 128 ),
354     IN in_datetime     DATETIME
355 )
356 BEGIN
357
358     START TRANSACTION;
359
360     -- добавить куратора заявки
361
362     UPDATE task
363     SET task.owner_id = (
364         SELECT id
365         FROM employee
366         WHERE employee.login LIKE in_login AND
367               employee.password LIKE in_password
368     )
369     WHERE task.id = in_task_id;
370
371     INSERT INTO task_operation( datetime, task_id, technic_id, state_id )
372     VALUES(
373         in_datetime,

```

```

374     in_task_id ,
375     (
376         SELECT id
377         FROM employee
378         WHERE employee.login    LIKE in_login AND
379               employee.password LIKE in_password
380     ),
381     (
382         SELECT id
383         FROM task_state
384         WHERE LOWER( task_state.name ) LIKE 'выполняется'
385     )
386 );
387
388 COMMIT;
389
390 END$$
391
392 -----
393
394 DROP PROCEDURE IF EXISTS close_task$$
395 CREATE PROCEDURE close_task(
396     IN in_task_id    INT(11),
397     IN in_login      VARCHAR ( 64 ),
398     IN in_password   VARCHAR ( 128 ),
399     IN in_datetime   DATETIME
400 )
401 BEGIN
402
403     START TRANSACTION;
404
405     -- закрыть заявку
406
407     INSERT INTO task_operation( datetime, task_id, technic_id, state_id )
408     VALUES(
409         in_datetime,
410         in_task_id,
411         (
412             SELECT id
413             FROM employee
414             WHERE employee.login    LIKE in_login AND
415                   employee.password LIKE in_password
416         ),
417         (
418             SELECT id
419             FROM task_state
420             WHERE LOWER( task_state.name ) LIKE 'закрыта'
421         )
422     );
423
424     COMMIT;
425
426 END$$
427
428 -----
429
430 DROP PROCEDURE IF EXISTS put_equipment_repair$$
431 CREATE PROCEDURE put_equipment_repair(
432     IN task_name      VARCHAR( 128 ),
433     IN priority_id    INT( 11 ),
434     IN serial_number  VARCHAR( 128 ),
435     IN login          VARCHAR ( 64 ),
436     IN password       VARCHAR ( 128 ),
437     IN in_datetime   DATETIME
438 )
439 BEGIN
440
441     START TRANSACTION;
442
443     -- поместить оборудование на ремонт
444
445     INSERT INTO task ( name, datetime, priority_id, client_id )
446     VALUES( task_name,
447         in_datetime,
448         priority_id,
449         (
450             SELECT id
451             FROM employee
452             WHERE employee.login    LIKE login AND
453                   employee.password LIKE password
454         )
455     );
456
457     INSERT INTO task_operation( datetime, task_id, technic_id, state_id )
458     VALUES(
459         in_datetime,
460         (
461             SELECT id
462             FROM task
463             WHERE datetime = (
464                 SELECT MAX( datetime )
465                 FROM task
466             )
467         ),

```

```

468         (
469             SELECT id
470             FROM employee
471             WHERE employee.login LIKE login AND
472                   employee.password LIKE password
473         ),
474         (
475             SELECT id
476             FROM task_state
477             WHERE task_state.name = 'новая'
478         )
479     );
480
481 INSERT INTO task_equipment ( task_id, equipment_id )
482 VALUES(
483     (
484         SELECT id
485         FROM task
486         WHERE datetime = (
487             SELECT MAX( datetime )
488             FROM task
489         )
490     ),
491     (
492         SELECT id
493         FROM equipment
494         WHERE LOWER( equipment.serial_number ) LIKE serial_number
495     )
496 );
497
498 INSERT INTO equipment_operation( datetime, equipment_id, eq_oper_type_id )
499 VALUES(
500     in_datetime,
501     (
502         SELECT id
503         FROM equipment
504         WHERE LOWER( equipment.serial_number ) LIKE serial_number
505     ),
506     (
507         SELECT id
508         FROM equipment_operation_type
509         WHERE equipment_operation_type.name = 'помещение_на_ремонт'
510     )
511 );
512
513 COMMIT;
514
515 END$$
516
517 --
518
519 DROP PROCEDURE IF EXISTS gen_equipment_repair$$
520 CREATE PROCEDURE gen_equipment_repair(
521     IN in_detail_price DOUBLE,
522     IN in_datetime DATETIME,
523     IN in_serial_number VARCHAR( 128 ),
524     IN in_comment VARCHAR( 512 ),
525     IN in_detail_model_id INT( 11 ),
526     IN in_task_id INT( 11 )
527 )
528 BEGIN
529
530 START TRANSACTION;
531
532 -- ремонт оборудования
533
534 INSERT INTO equipment_operation( detail_price, datetime, equipment_id, eq_oper_type_id )
535 VALUES(
536     in_detail_price,
537     in_datetime,
538     (
539         SELECT id
540         FROM equipment
541         WHERE LOWER( equipment.serial_number ) LIKE in_serial_number
542     ),
543     (
544         SELECT id
545         FROM equipment_operation_type
546         WHERE equipment_operation_type.name = 'ремонт'
547     )
548 );
549
550 INSERT INTO repair( comment, datetime, detail_model_id, equipment_operation_id, task_id )
551 VALUES(
552     in_comment,
553     in_datetime,
554     in_detail_model_id,
555     (
556         SELECT id
557         FROM equipment_operation
558         WHERE equipment_operation.datetime = in_datetime AND equipment_operation.equipment_id = (
559             SELECT id
560             FROM equipment
561             WHERE LOWER( equipment.serial_number ) LIKE in_serial_number

```

```

562         ) AND equipment_operation.eq_oper_type_id = (
563         SELECT id
564         FROM equipment_operation_type
565         WHERE LOWER( equipment_operation_type.name ) = 'ремонт'
566         )
567     ),
568     in_task_id
569 );
570
571 COMMIT;
572
573 END$$
574
575 -----
576
577 DROP PROCEDURE IF EXISTS get_work_each_empl$$
578 CREATE PROCEDURE get_work_each_empl( )
579 BEGIN
580
581     -- получить информацию об общем количестве выполненных заявок
582     -- для каждого работника
583
584     SELECT id, name, COUNT(*)
585     FROM (
586         SELECT employee.id, employee.name,
587             tmp.task_id, tmp.priority_id,
588             task_priority.name AS priority
589         FROM employee
590         RIGHT JOIN (
591             SELECT owner_id, task.id AS task_id, task.priority_id
592             FROM task
593             WHERE id IN (
594                 SELECT task_id
595                 FROM task_operation
596                 WHERE task_operation.state_id = (
597                     SELECT id
598                     FROM task_state
599                     WHERE task_state.name = 'закрыта'
600                 )
601             )
602         ) AS tmp
603         ON employee.id = tmp.owner_id
604         RIGHT JOIN task_priority
605         ON priority_id = task_priority.id
606     ) AS tmp1
607     WHERE priority_id > 0
608     GROUP BY id, name;
609
610 END$$
611
612 -----
613
614 DROP PROCEDURE IF EXISTS get_work_each_prior_empl$$
615 CREATE PROCEDURE get_work_each_prior_empl( )
616 BEGIN
617
618     -- получить информацию о количестве выполненных заявок
619     -- для каждого работника по приоритетам заявок
620
621     SELECT id, name, priority, COUNT(*)
622     FROM (
623         SELECT employee.id, employee.name,
624             tmp.task_id, tmp.priority_id,
625             task_priority.name AS priority
626         FROM employee
627         RIGHT JOIN (
628             SELECT owner_id, task.id AS task_id, task.priority_id
629             FROM task
630             WHERE id IN (
631                 SELECT task_id
632                 FROM task_operation
633                 WHERE task_operation.state_id = (
634                     SELECT id
635                     FROM task_state
636                     WHERE task_state.name = 'закрыта'
637                 )
638             )
639         ) AS tmp
640         ON employee.id = tmp.owner_id
641         RIGHT JOIN task_priority
642         ON priority_id = task_priority.id
643     ) AS tmp1
644     WHERE priority_id > 0
645     GROUP BY id, priority, name;
646
647 END$$
648
649 -----
650
651 DROP PROCEDURE IF EXISTS get_work_rezult_all$$
652 CREATE PROCEDURE get_work_rezult_all( )
653 BEGIN
654
655

```

```

656  -- получить информацию об общем количестве
657  -- выполненных заявок
658
659  SELECT COUNT(*) AS summ
660  FROM (
661      SELECT employee.id, employee.name,
662             tmp.task_id, tmp.priority_id,
663             task_priority.name AS priority
664      FROM employee
665      RIGHT JOIN (
666          SELECT owner_id, task.id AS task_id, task.priority_id
667          FROM task
668          WHERE id IN (
669              SELECT task_id
670              FROM task_operation
671              WHERE task_operation.state_id = (
672                  SELECT id
673                  FROM task_state
674                  WHERE task_state.name = 'закрыта'
675              )
676          )
677      ) AS tmp
678      ON employee.id = tmp.owner_id
679      RIGHT JOIN task_priority
680      ON priority_id = task_priority.id
681  ) AS tmp1
682  WHERE priority_id > 0;
683
684  END$$
685
686  -- -----
687
688  DROP PROCEDURE IF EXISTS get_work_rezult_each_priority$$
689  CREATE PROCEDURE get_work_rezult_each_priority( )
690  BEGIN
691
692      -- получить информацию об общем количестве
693      -- выполненных заявок по приоритетам
694
695      SELECT priority, COUNT(*) AS summ
696      FROM (
697          SELECT employee.id, employee.name,
698                 tmp.task_id, tmp.priority_id,
699                 task_priority.name AS priority
700          FROM employee
701          RIGHT JOIN (
702              SELECT owner_id, task.id AS task_id, task.priority_id
703              FROM task
704              WHERE id IN (
705                  SELECT task_id
706                  FROM task_operation
707                  WHERE task_operation.state_id = (
708                      SELECT id
709                      FROM task_state
710                      WHERE task_state.name = 'закрыта'
711                  )
712              )
713          ) AS tmp
714          ON employee.id = tmp.owner_id
715          RIGHT JOIN task_priority
716          ON priority_id = task_priority.id
717      ) AS tmp1
718      WHERE priority_id > 0
719      GROUP BY priority_id, priority;
720
721  END$$
722
723  -- -----
724
725  DROP PROCEDURE IF EXISTS add_admin$$
726  CREATE PROCEDURE add_admin(
727      IN in_snils VARCHAR( 16 )
728  )
729  BEGIN
730
731      -- добавить администратора
732
733      INSERT INTO admins
734      VALUES (
735          (
736              SELECT id
737              FROM employee
738              WHERE LOWER( employee.snils ) LIKE in_snils
739          )
740      );
741
742  END$$
743
744  -- -----
745
746  DROP PROCEDURE IF EXISTS add_technic$$
747  CREATE PROCEDURE add_technic(
748      IN in_snils VARCHAR( 16 )
749  )

```



```

750 BEGIN
751
752  — добавить техника
753
754  INSERT INTO technics
755    VALUES (
756      (
757        SELECT id
758        FROM employee
759        WHERE LOWER( employee.snils ) LIKE in_snils
760      )
761    );
762
763  END$$
764
765  — —————
766
767  DROP PROCEDURE IF EXISTS add_new_employee$$
768  CREATE PROCEDURE add_new_employee(
769      IN in_snils      VARCHAR( 16 ),
770      IN in_name       VARCHAR( 128 ),
771      IN in_phone      VARCHAR( 32 ),
772      IN in_addr       VARCHAR( 256 ),
773      IN in_login      VARCHAR( 64 ),
774      IN in_password   VARCHAR( 128 ),
775      IN in_role_name  VARCHAR( 128 ),
776      IN in_department_name VARCHAR( 128 ),
777      IN in_date DATE
778  )
779  BEGIN
780
781  START TRANSACTION;
782
783  — принять нового сотрудника на работу
784
785  INSERT INTO employee ( snils , name , phone , addr , login , password , role_id )
786    VALUES(
787      in_snils , in_name , in_phone , in_addr ,
788      in_login , in_password ,
789      (
790        SELECT id
791        FROM employee_role
792        WHERE LOWER( employee_role.name ) LIKE in_role_name
793      )
794    );
795
796  INSERT INTO employee_operation ( date , type_id , employee_id , department_id )
797    VALUES (
798      in_date ,
799      (
800        SELECT id
801        FROM employee_operation_type
802        WHERE LOWER( employee_operation_type.name ) = 'принят'
803      ) ,
804      (
805        SELECT id
806        FROM employee
807        WHERE LOWER( employee.snils ) LIKE in_snils
808      ) ,
809      (
810        SELECT id
811        FROM department
812        WHERE LOWER( department.name ) LIKE in_department_name
813      )
814    );
815  COMMIT;
816
817  END$$
818
819  — —————
820
821  DROP PROCEDURE IF EXISTS add_existing_employee$$
822  CREATE PROCEDURE add_existing_employee(
823      IN in_snils      VARCHAR( 16 ),
824      IN in_login      VARCHAR( 64 ),
825      IN in_password   VARCHAR( 128 ),
826      IN in_role_name  VARCHAR( 128 ),
827      IN in_department_name VARCHAR( 128 ),
828      IN in_date DATE
829  )
830  BEGIN
831
832  START TRANSACTION;
833
834  — принять существующего сотрудника на работу
835
836  UPDATE employee
837    SET employee.login      = in_login ,
838        employee.password = in_password ,
839        employee.role_id   = (
840          SELECT id
841          FROM employee_role
842          WHERE LOWER( employee_role.name ) LIKE in_role_name
843        )

```

```

844 WHERE employee.id = (
845     SELECT id
846     FROM employee
847     WHERE LOWER( employee.snils ) LIKE in_snils
848 );
849
850 INSERT INTO employee_operation ( date, type_id, employee_id, department_id )
851 VALUES (
852     in_date,
853     (
854         SELECT id
855         FROM employee_operation_type
856         WHERE LOWER( employee_operation_type.name ) = 'принят'
857     ),
858     (
859         SELECT id
860         FROM employee
861         WHERE LOWER( employee.snils ) LIKE in_snils
862     ),
863     (
864         SELECT id
865         FROM department
866         WHERE LOWER( department.name ) LIKE in_department_name
867     )
868 );
869
870 COMMIT;
871
872 END$$
873
874 -- -----
875
876 DROP PROCEDURE IF EXISTS add_employee$$
877 CREATE PROCEDURE add_employee(
878     IN in_snils      VARCHAR( 16 ),
879     IN in_name       VARCHAR( 128 ),
880     IN in_phone      VARCHAR( 32 ),
881     IN in_addr       VARCHAR( 256 ),
882     IN in_login      VARCHAR( 64 ),
883     IN in_password   VARCHAR( 128 ),
884     IN in_role_name  VARCHAR( 128 ),
885     IN in_department_name VARCHAR( 128 ),
886     IN in_date       DATE
887 )
888 BEGIN
889
890     START TRANSACTION;
891
892     -- принять сотрудника на работу
893
894     IF EXISTS
895     (
896         SELECT *
897         FROM employee
898         WHERE LOWER( employee.snils ) LIKE in_snils
899     )
900     THEN
901         IF EXISTS
902         (
903             SELECT *
904             FROM (
905                 SELECT employee_id, type_id, MAX( employee_operation.date ),
906                     employee_operation_type.name AS employee_operation_name
907                 FROM employee_operation, employee_operation_type
908                 WHERE employee_id = (
909                     SELECT id
910                     FROM employee
911                     WHERE LOWER( employee.snils ) LIKE in_snils
912                 ) AND type_id = employee_operation_type.id
913             ) AS tmp
914             WHERE tmp.employee_operation_name = 'уволен'
915         )
916         THEN
917             CALL add_existing_employee( in_snils,
918                 in_login, in_password,
919                 in_role_name, in_department_name,
920                 in_date
921             );
922         END IF;
923     ELSE
924         CALL add_new_employee( in_snils, in_name, in_phone, in_addr,
925             in_login, in_password,
926             in_role_name, in_department_name,
927             in_date
928         );
929     END IF;
930
931     CASE in_role_name
932     WHEN 'администратор' THEN CALL add_admin( in_snils );
933     WHEN 'техник' THEN CALL add_techinc( in_snils );
934     ELSE
935         BEGIN
936             END;
937     END CASE;

```

```

938 COMMIT;
939
940
941 END$$
942
943 -----
944
945 DROP PROCEDURE IF EXISTS delete_admin$$
946 CREATE PROCEDURE delete_admin(
947     IN in_snils VARCHAR( 16 )
948 )
949 BEGIN
950     -- удалить администратора
951
952     DELETE FROM admins
953     WHERE admins.employee_id =
954         (
955             SELECT id
956             FROM employee
957             WHERE LOWER( employee.snils ) LIKE in_snils
958         );
959
960 END$$
961
962 -----
963
964 DROP PROCEDURE IF EXISTS delete_technic$$
965 CREATE PROCEDURE delete_technic(
966     IN in_snils VARCHAR( 16 )
967 )
968 BEGIN
969     -- удалить техника
970
971     DELETE FROM technics
972     WHERE technics.employee_id =
973         (
974             SELECT id
975             FROM employee
976             WHERE LOWER( employee.snils ) LIKE in_snils
977         );
978
979 END$$
980
981 -----
982
983 DROP FUNCTION IF EXISTS get_employee_role$$
984 CREATE FUNCTION get_employee_role(
985     IN in_snils VARCHAR( 16 )
986 )
987 RETURNS CHAR( 128 )
988 BEGIN
989     -- получить должность сотрудника
990
991     RETURN(
992         SELECT name
993         FROM employee_role
994         INNER JOIN (
995             SELECT role_id
996             FROM employee
997             WHERE LOWER( employee.snils ) = in_snils
998         ) AS tmp
999         ON employee_role.id = tmp.role_id
1000     );
1001
1002 END$$
1003
1004 -----
1005
1006 DROP PROCEDURE IF EXISTS delete_employee$$
1007 CREATE PROCEDURE delete_employee(
1008     IN in_snils VARCHAR( 16 ),
1009     IN in_date DATE
1010 )
1011 BEGIN
1012     START TRANSACTION;
1013
1014     -- уволить сотрудника
1015
1016     CREATE TEMPORARY TABLE temp AS(
1017         SELECT *
1018         FROM (
1019             SELECT employee_id, type_id, MAX( employee_operation.date ),
1020                 employee_operation_type.name AS employee_operation_name,
1021                 employee_operation.department_id
1022             FROM employee_operation, employee_operation_type
1023             WHERE employee_id = (
1024                 SELECT id
1025                 FROM employee
1026                 WHERE LOWER( employee.snils ) LIKE in_snils

```

```

1032     ) AND type_id = employee_operation_type.id
1033     ) AS tmp
1034     WHERE tmp.employee_operation_name = 'принят'
1035 );
1036
1037 IF EXISTS
1038 (
1039     SELECT *
1040     FROM temp
1041 )
1042 THEN
1043     INSERT INTO employee_operation ( date, type_id, employee_id, department_id )
1044     VALUES (
1045         in_date,
1046         (
1047             SELECT id
1048             FROM employee_operation_type
1049             WHERE LOWER( employee_operation_type.name ) = 'уволен'
1050         ),
1051         (
1052             SELECT id
1053             FROM employee
1054             WHERE LOWER( employee.snils ) LIKE in_snils
1055         ),
1056         (
1057             SELECT department_id
1058             FROM temp
1059         )
1060     );
1061
1062     UPDATE employee
1063     SET employee.login = NULL,
1064         employee.password = NULL
1065     WHERE LOWER( employee.snils ) LIKE in_snils;
1066 END IF;
1067
1068 IF (
1069     ( SELECT get_employee_role( in_snils ) ) = 'администратор'
1070 )
1071 THEN CALL delete_admin( in_snils );
1072 ELSEIF (
1073     ( SELECT get_employee_role( in_snils ) ) = 'техник'
1074 )
1075 THEN CALL delete_technic( in_snils );
1076 END IF;
1077
1078 DROP TABLE IF EXISTS temp;
1079
1080 COMMIT;
1081
1082 END$$
1083
1084 -- -----
1085
1086 DROP PROCEDURE IF EXISTS add_equipment_owner$$
1087 CREATE PROCEDURE add_equipment_owner(
1088     IN in_snils          VARCHAR( 16 ),
1089     IN in_serial_number  VARCHAR( 128 ),
1090     IN in_start_datetime DATETIME
1091 )
1092 BEGIN
1093
1094     START TRANSACTION;
1095
1096     -- добавить пользователя оборудованием
1097
1098     IF NOT EXISTS (
1099         SELECT *
1100         FROM equipment_owner
1101         WHERE equipment_owner.equipment_id = (
1102             SELECT id
1103             FROM equipment
1104             WHERE LOWER( equipment.serial_number ) LIKE in_serial_number
1105         ) AND
1106         equipment_owner.employee_id = (
1107             SELECT id
1108             FROM employee
1109             WHERE LOWER( employee.snils ) LIKE in_snils
1110         ) AND
1111         equipment_owner.finish_datetime = NULL
1112     )
1113 THEN
1114     INSERT INTO equipment_owner ( equipment_id, employee_id, start_datetime )
1115     VALUES (
1116         (
1117             SELECT id
1118             FROM equipment
1119             WHERE LOWER( equipment.serial_number ) LIKE in_serial_number
1120         ),
1121         (
1122             SELECT id
1123             FROM employee
1124             WHERE LOWER( employee.snils ) LIKE in_snils
1125         ),

```

```

1126         in_start_datetime
1127     );
1128 END IF;
1129
1130 COMMIT;
1131
1132 END$$
1133
1134 --
1135
1136 DROP PROCEDURE IF EXISTS add_equipment$$
1137 CREATE PROCEDURE add_equipment(
1138     IN in_name          VARCHAR( 128 ),
1139     IN in_serial_number VARCHAR( 128 ),
1140     IN in_addr          VARCHAR( 256 ),
1141     IN in_equipment_model VARCHAR( 255 ),
1142     IN in_datetime DATETIME
1143 )
1144 BEGIN
1145
1146     START TRANSACTION;
1147
1148     -- добавление оборудования
1149
1150     INSERT INTO equipment ( name, serial_number, addr, equipment_model_id )
1151     VALUES (
1152         in_name,
1153         in_serial_number,
1154         in_addr,
1155         (
1156             SELECT id
1157             FROM equipment_model
1158             WHERE LOWER( equipment_model.name ) LIKE in_equipment_model
1159             UNION SELECT 154 LIMIT 1
1160         )
1161     );
1162
1163     INSERT INTO equipment_operation ( datetime, equipment_id, eq_oper_type_id )
1164     VALUES (
1165         in_datetime,
1166         (
1167             SELECT id
1168             FROM equipment
1169             WHERE LOWER( equipment.serial_number ) LIKE in_serial_number
1170         ),
1171         (
1172             SELECT id
1173             FROM equipment_operation_type
1174             WHERE LOWER( equipment_operation_type.name ) = 'поступление'
1175         )
1176     );
1177
1178     COMMIT;
1179
1180 END$$
1181
1182 --
1183
1184 DROP PROCEDURE IF EXISTS add_equipment_and_owner$$
1185 CREATE PROCEDURE add_equipment_and_owner(
1186     IN in_snils          VARCHAR( 16 ),
1187     IN in_name          VARCHAR( 128 ),
1188     IN in_serial_number VARCHAR( 128 ),
1189     IN in_addr          VARCHAR( 256 ),
1190     IN in_equipment_model VARCHAR( 255 ),
1191     IN in_datetime DATETIME
1192 )
1193 BEGIN
1194
1195     START TRANSACTION;
1196
1197     -- добавление оборудования и владельца оборудования
1198
1199     CALL add_equipment( in_name, in_serial_number, in_addr, in_equipment_model, in_datetime );
1200     CALL add_equipment_owner( in_snils, in_serial_number, in_datetime );
1201
1202     COMMIT;
1203
1204 END$$
1205
1206 --
1207
1208 DROP PROCEDURE IF EXISTS delete_equipment$$
1209 CREATE PROCEDURE delete_equipment(
1210     IN in_serial_number VARCHAR( 128 ),
1211     IN in_datetime DATETIME
1212 )
1213 BEGIN
1214
1215     START TRANSACTION;
1216
1217     -- списание оборудования
1218
1219     IF NOT EXISTS (

```

```

1220 SELECT *
1221 FROM (
1222     SELECT equipment_operation_type.name
1223     FROM equipment_operation_type
1224     INNER JOIN (
1225         SELECT equipment_id, eq_oper_type_id, datetime
1226         FROM (
1227             SELECT equipment_id, eq_oper_type_id, datetime
1228             FROM equipment_operation
1229             WHERE equipment_id = (
1230                 SELECT id
1231                 FROM equipment
1232                 WHERE LOWER( equipment.serial_number ) LIKE in_serial_number
1233             )
1234         ) AS tmp
1235         WHERE datetime = (
1236             SELECT MAX( datetime )
1237             FROM equipment_operation
1238             WHERE equipment_operation.equipment_id = (
1239                 SELECT id
1240                 FROM equipment
1241                 WHERE LOWER( equipment.serial_number ) LIKE in_serial_number
1242             )
1243         )
1244     ) AS temp
1245     ON equipment_operation_type.id = temp.eq_oper_type_id
1246 ) AS tmp_operation
1247 WHERE tmp_operation.name = 'списание'
1248 )
1249 THEN
1250     INSERT INTO equipment_operation( datetime, equipment_id, eq_oper_type_id )
1251     VALUES(
1252         in_datetime,
1253         (
1254             SELECT id
1255             FROM equipment
1256             WHERE LOWER( equipment.serial_number ) LIKE in_serial_number
1257         ),
1258         (
1259             SELECT id
1260             FROM equipment_operation_type
1261             WHERE LOWER( equipment_operation_type.name ) = 'списание'
1262         )
1263     );
1264
1265     UPDATE equipment_owner
1266     SET equipment_owner.finish_datetime = in_datetime
1267     WHERE equipment_owner.equipment_id = (
1268         SELECT id
1269         FROM equipment
1270         WHERE LOWER( equipment.serial_number ) LIKE in_serial_number
1271     );
1272 END IF;
1273
1274 COMMIT;
1275
1276 END$$
1277
1278 -----
1279
1280 DROP PROCEDURE IF EXISTS delete_equipment_owner$$
1281 CREATE PROCEDURE delete_equipment_owner(
1282     IN in_snils          VARCHAR( 16 ),
1283     IN in_serial_number  VARCHAR( 128 ),
1284     IN in_datetime       DATETIME
1285 )
1286 BEGIN
1287
1288     START TRANSACTION;
1289
1290     -- отписать владельца от оборудования
1291
1292     IF EXISTS (
1293         SELECT *
1294         FROM equipment_owner
1295         WHERE equipment_owner.equipment_id = (
1296             SELECT id
1297             FROM equipment
1298             WHERE LOWER( equipment.serial_number ) LIKE in_serial_number
1299         ) AND
1300         equipment_owner.employee_id = (
1301             SELECT id
1302             FROM employee
1303             WHERE LOWER( employee.snils ) LIKE in_snils
1304         ) AND
1305         equipment_owner.finish_datetime = NULL
1306     )
1307     THEN
1308         UPDATE equipment_owner
1309         SET equipment_owner.finish_datetime = in_datetime
1310         WHERE equipment_owner.equipment_id = (
1311             SELECT id
1312             FROM equipment
1313             WHERE LOWER( equipment.serial_number ) LIKE in_serial_number

```

```

1314 );
1315 END IF;
1316
1317 COMMIT;
1318
1319 END$$
1320
1321 -- -----
1322
1323 DROP FUNCTION IF EXISTS get_department_id_for_employee$$
1324 CREATE FUNCTION get_department_id_for_employee(
1325                                     in_snils   VARCHAR( 16 )
1326                                     )
1327 RETURNS INT
1328 BEGIN
1329     -- получить id-подразделения для сотрудника
1330
1331
1332 RETURN(
1333     SELECT department_id
1334     FROM (
1335         SELECT department_id, MAX( employee_operation.date )
1336         FROM employee_operation
1337         WHERE employee_operation.employee_id = (
1338             SELECT id
1339             FROM employee
1340             WHERE LOWER( employee.snils ) = in_snils
1341         )
1342     ) AS temp
1343 );
1344
1345 END$$
1346
1347 -- -----
1348
1349 DROP PROCEDURE IF EXISTS add_vacation$$
1350 CREATE PROCEDURE add_vacation(
1351                                     IN in_snils   VARCHAR( 16 ),
1352                                     IN date_start DATE,
1353                                     IN date_end   DATE
1354                                     )
1355 BEGIN
1356
1357 START TRANSACTION;
1358
1359     -- установить даты отпусков
1360
1361 INSERT INTO employee_operation ( date, type_id, employee_id, department_id )
1362 VALUES(
1363     date_start ,
1364     (
1365         SELECT id
1366         FROM employee_operation_type
1367         WHERE LOWER( employee_operation_type.name ) = 'отправлен_в_отпуск'
1368     ),
1369     (
1370         SELECT id
1371         FROM employee
1372         WHERE LOWER( employee.snils ) = in_snils
1373     ),
1374     ( SELECT get_department_id_for_employee( in_snils ) )
1375 );
1376
1377 INSERT INTO employee_operation ( date, type_id, employee_id, department_id )
1378 VALUES(
1379     date_end ,
1380     (
1381         SELECT id
1382         FROM employee_operation_type
1383         WHERE LOWER( employee_operation_type.name ) = 'отозван_из_отпуска'
1384     ),
1385     (
1386         SELECT id
1387         FROM employee
1388         WHERE LOWER( employee.snils ) = in_snils
1389     ),
1390     ( SELECT get_department_id_for_employee( in_snils ) )
1391 );
1392
1393 COMMIT;
1394
1395 END$$
1396
1397 -- -----
1398
1399 DELIMITER ;

```

- 2.1. среднее время выполнения запросов (в мс) для обоих разбиений базы данных для каждого запроса и суммарное время выполнения запросов на обоих разбиениях;
  - 2.2. вывод, объясняющий, почему время выполнения запросов в одном из двух разбиений оказалось меньшим (большим, равным), чем в другом.
3. Задача линейного программирования



Таблица 2. Значения  $d_{il}$

$T_l/Z_i$	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.	21.	22.
1.																						
2.																						
3.																						
4.																						
5.																						
6.																						
7.																						
8.																						
9.																						
10.																						

Где  $d_{il}$  — размер одной строки данных (в байтах) таблицы  $T_l$ , пересылаемой при выполнении  $i$ -го запроса.

Таблица 3. Значения  $\lambda_{ij}$

$K_j/Z_i$	1.	2.
1.		
2.		
3.		
4.		
5.		
6.		
7.		
8.		
9.		
10.		

Где  $\lambda_{ij}$  — интенсивность появления  $i$ -го запроса, инициированного из узла  $K_j$ .  $\lambda_{ij}$  определяется произвольно на основе экспертной оценки пользователя и показывает, как часто запрос  $i$  будет вызываться из узла  $K_j$ .

## A. Tables

- 1) admins
- 2) department
- 3) department\_activity\_sphere
- 4) detail\_category
- 5) detail\_model
- 6) employee
- 7) employee\_operation
- 8) employee\_operation\_type
- 9) employee\_role
- 10) equipment
- 11) equipment\_category
- 12) equipment\_model
- 13) equipment\_operation
- 14) equipment\_operation\_type
- 15) equipment\_owner
- 16) repair
- 17) task
- 18) task\_equipment
- 19) task\_operation
- 20) task\_priority
- 21) task\_state
- 22) technics