Федеральное государственное автономное образовательное учреждение

высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа по Базам Данных №3

Вариант 31073

Выполнил  
Пчелкин Илья Игоревич

P3106

Проверил

Вербовой А. А.

Санкт-Петербург 2025

Оглавление

[Задание 3](#_Toc193830126)

[Даталогическая модель 3](#_Toc193830127)

[Функциональные зависимости 3](#_Toc193830128)

[Нормальные формы 4](#_Toc193830129)

[Денормализованная модель 4](#_Toc193830130)

[Реализация модели с триггерами на SQL 5](#_Toc193830131)

[Вывод 8](#_Toc193830132)

# Задание

Для отношений, полученных при построении предметной области из лабораторной работы №1, выполните следующие действия:

* Опишите функциональные зависимости для отношений полученной схемы (минимальное множество);
* Приведите отношения в 3NF (как минимум). Постройте схему на основеNF (как минимум).
* Опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основеNF;
* Преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF. Если ваша схема находится уже в BCNF, докажите это;
* Какие денормализации будут полезны для вашей схемы? Приведите подробное описание.

Придумайте триггер и связанную с ним функцию, относящиеся к вашей предметной области, согласуйте их с преподавателем и реализуйте на языке PL/pgSQL.

**Описание предметной области из из лабораторной работы №1:**

Самое забавное, что если бы Чандра отключил пожарную сигнализацию или пошел курить в шлюз, никто бы не возражал. Но Чандра не любит выставлять напоказ свои маленькие человеческие слабости; теперь он вообще не отлучается от ЭАЛа...

# Даталогическая модель

Изображение выглядит как диаграмма, План, Технический чертеж, схематичный

Автоматически созданное описание

# Функциональные зависимости

Person: Characters\_id -> sex, age

Robot: Characters\_id -> purpose

Characters: id -> name

Event: id -> Action\_id, Location\_id, Characters\_id, begin\_time, end\_time, character\_type

Action: id -> name

Location: id -> name

Signalization: Location\_id -> is\_active

Corridor: Location\_id\_1, Location\_id\_2 -> {∅}

# Нормальные формы

1НФ:

Cоответствует, потому что:

* На пересечении каждой строки и столбца — одно значение
* Каждая таблица имеет первичный ключ

2НФ:  
Cоответствует, потому что:

* Соответствует 1НФ
* Атрибуты, не входящие в первичный ключ, в полной функциональной зависимости от первичного ключа отношения

3НФ:  
Cоответствует, потому что:

* Соответствует 2НФ
* Отсутствуют транзитивные зависимости

НФБК:

Cоответствует, потому что:

* Соответствует 3НФ
* Каждый атрибут, определяющий другие атрибуты в каждой из таблиц является суперключом, т. е. в каждой таблице детерминантом является суперключ, т. к. все избыточные функциональные зависимости вынесены в отдельные таблицы.

# Денормализованная модель

Для денормализации можно убрать таблицы Location, Action и Signalization и поместить Location\_name, Action\_name и Signalization\_is\_active в таблицу Event:

Изображение выглядит как диаграмма, чек, линия, текст

Автоматически созданное описание

# Реализация модели с триггерами на SQL

drop table if exists Person cascade;  
drop table if exists Robot cascade;  
drop table if exists Action cascade;  
drop table if exists Location cascade;  
drop table if exists Event cascade;  
drop table if exists Signalization cascade;  
drop table if exists Corridor cascade;  
drop table if exists Characters cascade;  
  
drop type if exists person\_sex cascade;  
drop type if exists character\_type cascade;  
create type person\_sex as enum ('Male', 'Female');  
create type character\_type as enum ('Person', 'Robot-helper', 'Robot-security');  
  
  
create table Action  
(  
 id int primary key,  
 name text  
);  
create table Location  
(  
 id int primary key,  
 name text  
);  
create table Characters  
(  
 id int primary key,  
 name varchar(20)  
);  
  
create table Event  
(  
 id int primary key,  
 Action\_id int not null references Action (id),  
 Location\_id int not null references Location (id),  
 Characters\_id int not null references Characters (id),  
 begin\_time timestamp,  
 end\_time timestamp,  
 character\_type character\_type  
);  
create table Person  
(  
 id int references Characters (id),  
 sex person\_sex,  
 age int check (age >= 0 and age <= 130)  
);  
create table Robot  
(  
 id int references Characters (id),  
 purpose text  
);  
create table Signalization  
(  
 Location\_id int references Location (id),  
 is\_active boolean  
);  
create table Corridor  
(  
 Location\_id\_1 int not null references Location (id),  
 Location\_id\_2 int not null references Location (id),  
 check (Location\_id\_1 != Location\_id\_2)  
 );  
  
  
  
  
insert into Location(id, name)  
values (1, 'Променад');  
insert into Location(id, name)  
values (2, 'Столовая');  
insert into Location(id, name)  
values (3, 'Шлюз');  
  
insert into Action(id, name)  
values (1, 'Пылесосить');  
insert into Action(id, name)  
values (2, 'Кушать');  
insert into Action(id, name)  
values (3, 'Курить');  
insert into Action(id, name)  
values (4, 'Охранять');  
  
insert into Characters(id, name)  
values (1, 'Чандра');  
insert into Characters(id, name)  
values (2, 'Cандра');  
insert into Characters(id, name)  
values (3, 'ЭАЛ');  
insert into Characters(id, name)  
values (4, 'ПЭЛ');  
  
  
drop trigger if exists check\_action on Event;  
  
  
create or replace function check\_character\_action()  
returns trigger as $$  
begin  
 if (new.Action\_id = 1 and new.character\_type = 'Robot-security') then raise exception 'Робот-охранник не может выполнять действие "Пылесосить"!';  
 elsif (new.Action\_id = 2 and new.character\_type = 'Robot-security') then raise exception 'Робот-охранник не может выполнять действие "Кушать"!';  
 elsif (new.Action\_id = 2 and new.character\_type = 'Robot-helper') then raise exception 'Робот-помощник не может выполнять действие "Кушать"!';  
 elsif (new.Action\_id = 3 and new.character\_type = 'Robot-security') then raise exception 'Робот-охранник не может выполнять действие "Курить"!';  
 elsif (new.Action\_id = 3 and new.character\_type = 'Robot-helper') then raise exception 'Робот-помощник не может выполнять действие "Курить"!';  
 elsif (new.Action\_id = 4 and new.character\_type = 'Robot-helper') then raise exception 'Робот-помощник не может выполнять действие "Охранять"!';  
end if;  
return new;  
end;  
$$ language plpgsql;  
  
create trigger check\_action  
before insert or update on Event  
for each row execute function check\_character\_action();  
  
insert into Event(id, Action\_id, Location\_id, Characters\_id, begin\_time, end\_time, character\_type)  
values (2, 3, 3, 1, '2125-02-23 22:15:23', '2125-02-23 22:20:12', 'Person');  
insert into Event(id, Action\_id, Location\_id, Characters\_id, begin\_time, end\_time, character\_type)  
values (1, 2, 2, 2, '2125-02-23 15:00:11', '2125-02-23 15:30:33', 'Person');  
insert into Event(id, Action\_id, Location\_id, Characters\_id, begin\_time, end\_time, character\_type)  
values (3, 1, 1, 3, '2125-02-23 18:00:00', '2125-02-23 18:10:00', 'Robot-helper');  
insert into Event(id, Action\_id, Location\_id, Characters\_id, begin\_time, end\_time, character\_type)  
values (4, 4, 1, 4, '2120-12-01 18:00:00', NULL, 'Robot-security');  
  
  
  
  
  
insert into Person(id, sex, age)  
values (1, 'Male', 30);  
insert into Person(id, sex, age)  
values (2, 'Female', 25);  
  
insert into Robot(id, purpose)  
values (3, 'Робот-помощник');  
insert into Robot(id, purpose)  
values (4, 'Робот-охранник');  
  
insert into Signalization(Location\_id, is\_active)  
values (1, FALSE);  
insert into Signalization(Location\_id, is\_active)  
values (2, FALSE);  
insert into Signalization(Location\_id, is\_active)  
values (3, TRUE);  
  
insert into Corridor(Location\_id\_1, Location\_id\_2)  
values (1, 2);  
insert into Corridor(Location\_id\_1, Location\_id\_2)  
values (1, 3);

# Вывод

При выполнении данной лабораторной работы я узнал про нормальные формы бд, триггеры, функциональные зависимости и денормализацию.