НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

# ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

**БАЗЫ ДАННЫХ**

# **Информационная Система Военного округа**

Сироткин Михаил Александрович

Группа 21203

Новосибирск 2024

## Содержание

[Задание 3](#_Toc166470492)

[Схема базы данных 4](#_Toc166470493)

[Создание таблиц 6](#_Toc166470494)

[Ограничения по поддержанию целостности 20](#_Toc166470495)

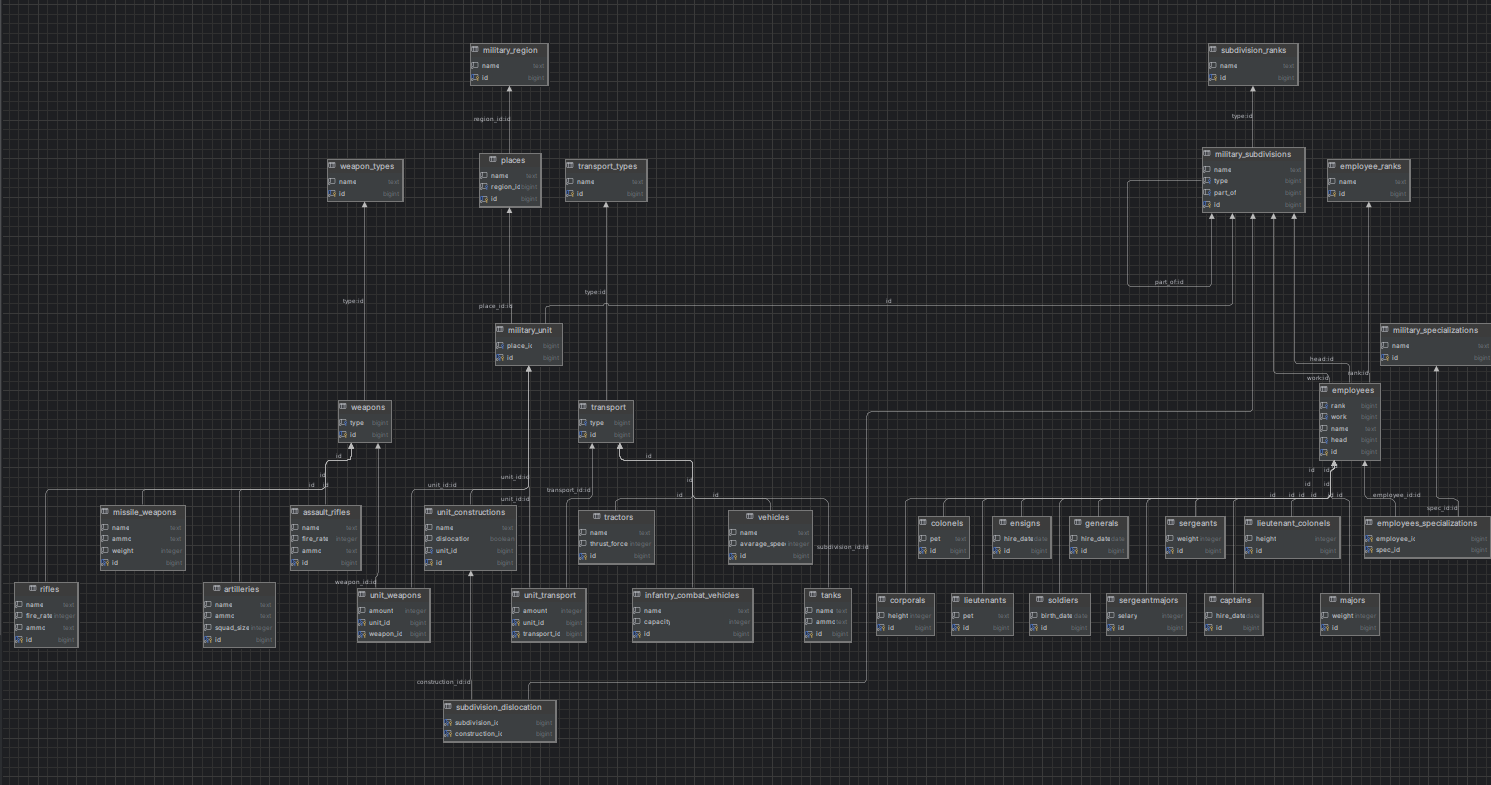
## Задание

Разработать структуру базы данных для информационной системы аптеки и реализовать приложение в архитектуре клиент-сервер, выполняющее операции внесения данных в базу данных, редактирование данных и запросы.

Военные части округа расквартированы по различным местам дислокации, причем в одном месте  могут располагаться несколько частей. Каждая воинская часть состоит из рот, роты из взводов,  взводы из отделений, в свою очередь воинские части объединяются в дивизии, корпуса или  бригады, а те в армии. Военный округ представлен офицерским составом (генералы, полковники,  подполковники, майоры, капитаны, лейтенанты) и рядовым и сержантским составом (старшины,  сержанты, прапорщики, ефрейторы, рядовые). Каждая из перечисленных категорий  военнослужащих может иметь характеристики, присущие только этой категории: для генералов  это может быть дата окончания академии, дата присвоения генеральского звания и т.д. Каждое из  подразделений имеет командира, причем военнослужащие офицерского состава могут  командовать любым из вышеперечисленных подразделений, а военнослужащие рядового и  сержантского состава только взводом и отделением. Все военнослужащие имеют одну или  несколько воинских специальностей.

Каждой воинской части придана боевая и транспортная техника: БМП, тягачи, автотранспорт и пр.  и вооружение: карабины, автоматическое оружие, артиллерия, ракетное вооружение и т.д. Каждая  из перечисленных категорий боевой техники и вооружения также имеет специфические, присущие  только ей атрибуты и по каждой категории может быть несколько видов техники и вооружения.  Инфраструктура военной части представлена набором сооружений (сооружение ©1, сооружение  ©2 . . .), некоторые из которых предназначены для дислокации подразделений части.

## Схема базы данных



* **places –** места дислокации военных частей (идентификатор места, название места, идентификатор округа из **military\_region**).
* **military\_subdivisions** – военные подразделения (идентификатор подразделения, название подразделения, идентификатор типа подразделения из таблицы **subdivision\_ranks**, идентификатор подразделения в котором содержится это подразделение).
* **subdivision\_ranks –** типы военных подразделений (идентификатор типа, название подразделения).
* **employees –** военнослужащие (идентификатор военнослужащего, идентификатор звания военнослужащего из **employee\_ranks**, идентификатор подразделения, где работает военнослужащий, из таблицы **military\_subdivisions**, ФИО военнослужащего, идентификатор подразделения, начальником которого является данный работник из таблицы **military\_subdivisions**).
* **employee\_ranks –** звания военнослужащих (идентификатор звания, название звания).
* **generals –** специальная информация о генералах (идентификатор генерала из таблицы **employees,** дата приёма на работу генерала).
* **colonels –** специальная информация о полковниках (идентификатор полковника из таблицы **employees,** животное, которым владеет полковник).
* **lieutenant\_colonels –** специальная информация о подполковниках (идентификатор подполковника из таблицы **employees,** рост подполковника).
* **majors –** специальная информация о майорах (идентификатор майора из таблицы **employees,** вес майора).
* **captains –** специальная информация о капитанах (идентификатор капитана из таблицы **employees,** дата приёма на работу капитана).
* **lieutenants –** специальная информация о лейтенантах (идентификатор лейтенанта из таблицы **employees,** животное, которым владеет лейтенант).
* **sergeantmajors –** специальная информация о старшинах (идентификатор старшины из таблицы **employees,** зарплата старшины).
* **sergeants –** специальная информация о сержантах (идентификатор сержанта из таблицы **employees,** вес сержанта).
* **ensigns –** специальная информация о прапорщиках (идентификатор ефрейтора из таблицы **employees,** дата приёма на работу прапорщика).
* **corporals –** специальная информация о ефрейторах (идентификатор ефрейтора из таблицы **employees,** рост ефрейтора).
* **soldiers –** специальная информация о солдатах (идентификатор солдата из таблицы **employees,** дата рождения солдата).
* **military\_specializations –** военные специализации (идентификатор специализации, название специализации).
* **employees\_specializations –** связь специализации и военнослужащего (идентификатор военнослужащего из таблицы **employees,** идентификатор специальности из таблицы **military\_specializations**).
* **military\_unit –** связь между военным подразделением (военной части) и местом её расположения (идентификатор военной части из таблицы **military\_subdivisions**, идентификатор из таблицы **places**).
* **military\_region –** военные круга (идентификатор военного округа, название военного округа).
* **transport\_types –** типы транспортных средств (идентификатор типа транспортного средства, название типа транспортного средства).
* **infantry\_combat\_vehicles –** специальная информация о БМП (идентификатор из таблицы **transport**, название БМП, вместимость человек).
* **tanks –** специальная информация о танках (идентификатор из таблицы **transport**, название танка, тип боеприпасов, которыми стреляет танк).
* **tractors –** специальная информация о тягочах (идентификатор из таблицы **transport**, название тягоча, сила тяги)
* **vehicles –** специальная информация о автомобилях (идентификатор из таблицы **transport**, название автомобиля, средняя скорость).
* **transport –** связь между транспортом и его идентификатором (идентификатор транспорта, тип транспорта из таблицы **transport\_types**).
* **unit\_transport –** информация о том, какой транспорт и сколько его единиц состоит на счету воинской части (идентификатор воинской части из таблички **military\_subdivisions**, идентификатор транспортного средства из таблицы **transport,** количество транспортных средств).
* **weapon\_types –** типы оружия (идентификатор типа оружия, название типа оружия).
* **rifles –** специальная информация о оружии типа винтовка (идентификатор винтовки из таблицы **weapons**, название винтовки, количество выстрелов в минуту, тип боеприпасов).
* **assault\_rifles -** специальная информация о оружии типа автомат (идентификатор автомата из таблицы **weapons**, название автомата, количество выстрелов в минуту, тип боеприпасов).
* **artilleries -** специальная информация о оружии типа артиллерия (идентификатор артиллерии из таблицы **weapons**, название артиллерии, тип боеприпасов, размер группы для управления артиллерией).
* **missile\_weapons -** специальная информация о оружии типа ракетное вооружение (идентификатор ракетного вооружения из таблицы **weapons**, название ракетного вооружения, тип боеприпасов, вес ракетного вооружения).
* **weapons –** связь между оружием и его идентификатором (идентификатор оружия, идентификатор типа оружия из таблицы **weapon\_types**).
* **unit\_weapons –** информация о том, какое оружие и сколько его единиц состоит на счету воинской части (идентификатор воинской части из таблицы **military\_subdivisions**, идентификатор оружия из таблицы **weapons,** количество оружия).
* **unit\_constructions –** сооружения состоящие на счету у воинской части (идентификатор сооружения, название сооружения, можно ли в сооружении дислоцировать подразделение, идентификатор воинской части из таблицы **military\_subdivisions**).
* **subdivision\_dislocation –** связь между воинскими подразделениями и сооружениями военной части (идентификатор воинского подразделения из таблицы **military\_subdivisions**, идентификатор сооружения из таблицы **unit\_constructions**).

## Создание таблиц

CREATE TABLE IF NOT EXISTS "places" (

"id" bigserial,

"name" text NOT NULL,

"region\_id" bigint NOT NULL,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "military\_subdivisions" (

"id" bigserial,

"name" text NOT NULL,

"type" bigint NOT NULL,

"part\_of" bigint ,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "subdivision\_ranks" (

"id" bigserial,

"name" text NOT NULL,

PRIMARY KEY ("id")

);

INSERT INTO "subdivision\_ranks" ("name")

VALUES('army'),

('brigade'),

('corp'),

('division'),

('unit'),

('company'),

('platoon'),

('squad');

CREATE TABLE IF NOT EXISTS "employees" (

"id" bigserial,

"rank" bigint NOT NULL,

"work" bigint NOT NULL,

"name" text NOT NULL,

"head" bigint,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "employee\_ranks" (

"id" bigserial,

"name" text NOT NULL,

PRIMARY KEY ("id")

);

INSERT INTO "employee\_ranks"("name")

VALUES('general'),

('colonel'),

('lieutenant colonel'),

('major'),

('captain'),

('lieutenant'),

('sergeantmajor'),

('sergeant'),

('ensign'),

('corporal'),

('soldier');

CREATE TABLE IF NOT EXISTS "generals" (

"id" bigint NOT NULL,

"hire\_date" date NOT NULL CHECK("hire\_date" <= CURRENT\_DATE),

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "colonels" (

"id" bigint NOT NULL,

"pet" text NOT NULL,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "lieutenant\_colonels" (

"id" bigint NOT NULL,

"height" integer NOT NULL CHECK("height" > 0),

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "majors" (

"id" bigint NOT NULL,

"weight" integer NOT NULL CHECK("weight" > 0),

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "captains" (

"id" bigint NOT NULL,

"hire\_date" date NOT NULL CHECK("hire\_date" <= CURRENT\_DATE),

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "lieutenants" (

"id" bigint NOT NULL,

"pet" text NOT NULL,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "sergeantmajors" (

"id" bigint NOT NULL,

"selary" integer NOT NULL CHECK("selary" > 0),

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "sergeants" (

"id" bigint NOT NULL,

"weight" integer NOT NULL CHECK("weight" > 0),

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "ensigns"(

"id" bigint NOT NULL,

"hire\_date" date NOT NULL CHECK("hire\_date" <= CURRENT\_DATE),

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "corporals" (

"id" bigint NOT NULL,

"height" integer NOT NULL CHECK(height > 0),

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "soldiers" (

"id" bigint NOT NULL,

"birth\_date" date NOT NULL,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "military\_specializations" (

"id" bigserial,

"name" text NOT NULL,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "employees\_specializations" (

"employee\_id" bigint NOT NULL,

"spec\_id" bigint NOT NULL,

PRIMARY KEY("employee\_id", "spec\_id")

);

CREATE TABLE IF NOT EXISTS "military\_unit" (

"id" bigint NOT NULL UNIQUE,

"place\_id" bigint NOT NULL,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "military\_region" (

"name" text NOT NULL,

"id" bigserial,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "transport\_types"(

"id" bigserial,

"name" text NOT NULL,

PRIMARY KEY ("id")

);

INSERT INTO "transport\_types"("name")

VALUES('infantry combat vehicle'),

('tank'),

('tractor'),

('vehicle');

CREATE TABLE IF NOT EXISTS "infantry\_combat\_vehicles"(

"id" bigint NOT NULL,

"name" text NOT NULL,

"capacity" integer NOT NULL CHECK("capacity" > 0),

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "tanks"(

"id" bigint NOT NULL,

"name" text NOT NULL,

"ammo" text NOT NULL,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "tractors"(

"id" bigint NOT NULL,

"name" text NOT NULL,

"thrust\_force" integer NOT NULL CHECK("thrust\_force" > 0),

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "vehicles"(

"id" bigint NOT NULL,

"name" text NOT NULL,

"avarage\_speed" integer NOT NULL CHECK("avarage\_speed" > 0),

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "transport"(

"id" bigserial,

"type" bigint NOT NULL,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "unit\_transport"(

"unit\_id" bigint NOT NULL,

"transport\_id" bigint NOT NULL,

"amount" integer NOT NULL CHECK("amount" > 0),

PRIMARY KEY("unit\_id", "transport\_id")

);

CREATE TABLE IF NOT EXISTS "weapon\_types"(

"id" bigserial,

"name" text NOT NULL,

PRIMARY KEY ("id")

);

INSERT INTO "weapon\_types"("name")

VALUES('rifle'),

('assault rifle'),

('artillery'),

('missile weapons');

CREATE TABLE IF NOT EXISTS "rifles"(

"id" bigint NOT NULL,

"name" text NOT NULL,

"fire\_rate" integer NOT NULL CHECK("fire\_rate" > 0),

"ammo" text NOT NULL,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "assault\_rifles"(

"id" bigint NOT NULL,

"name" text NOT NULL,

"fire\_rate" integer NOT NULL CHECK("fire\_rate" > 0),

"ammo" text NOT NULL,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "artilleries"(

"id" bigint NOT NULL,

"name" text NOT NULL,

"ammo" text NOT NULL,

"squad\_size" integer NOT NULL CHECK ("squad\_size" > 0),

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "missile\_weapons"(

"id" bigint NOT NULL,

"name" text NOT NULL,

"ammo" text NOT NULL,

"weight" integer NOT NULL,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "weapons"(

"id" bigserial,

"type" bigint NOT NULL,

PRIMARY KEY ("id")

);

CREATE TABLE IF NOT EXISTS "unit\_weapons"(

"unit\_id" bigint NOT NULL,

"weapon\_id" bigint NOT NULL,

"amount" integer NOT NULL CHECK("amount" > 0),

PRIMARY KEY("unit\_id", "weapon\_id")

);

CREATE TABLE IF NOT EXISTS "unit\_constructions"(

"id" bigserial,

"name" text NOT NULL,

"dislocation" boolean NOT NULL,

"unit\_id" bigint NOT NULL,

PRIMARY KEY("id")

);

CREATE TABLE IF NOT EXISTS "subdivision\_dislocation"(

"subdivision\_id" bigint NOT NULL,

"construction\_id" bigint NOT NULL,

PRIMARY KEY ("subdivision\_id", "construction\_id")

);

ALTER TABLE "unit\_constructions" ADD CONSTRAINT "unit\_constructions\_fk1" FOREIGN KEY ("unit\_id") REFERENCES "military\_unit"("id");

ALTER TABLE "subdivision\_dislocation" ADD CONSTRAINT "subdivision\_dislocation\_fk1" FOREIGN KEY ("construction\_id") REFERENCES "unit\_constructions"("id");

ALTER TABLE "subdivision\_dislocation" ADD CONSTRAINT "subdivision\_dislocation\_fk2" FOREIGN KEY ("subdivision\_id") REFERENCES "military\_subdivisions"("id");

ALTER TABLE "unit\_weapons" ADD CONSTRAINT "unit\_weapon\_fk1" FOREIGN KEY ("weapon\_id") REFERENCES "weapons"("id");

ALTER TABLE "unit\_weapons" ADD CONSTRAINT "unit\_weapon\_fk2" FOREIGN KEY ("unit\_id") REFERENCES "military\_unit"("id");

ALTER TABLE "weapons" ADD CONSTRAINT "weapon\_fk1" FOREIGN KEY ("type") REFERENCES "weapon\_types"("id");

ALTER TABLE "unit\_transport" ADD CONSTRAINT "unit\_transport\_fk1" FOREIGN KEY ("transport\_id") REFERENCES "transport"("id");

ALTER TABLE "unit\_transport" ADD CONSTRAINT "unit\_transport\_fk2" FOREIGN KEY ("unit\_id") REFERENCES "military\_unit"("id");

ALTER TABLE "transport" ADD CONSTRAINT "transport\_fk1" FOREIGN KEY ("type") REFERENCES "transport\_types"("id");

ALTER TABLE "military\_subdivisions" ADD CONSTRAINT "military\_subdivisions\_fk2" FOREIGN KEY ("type") REFERENCES "subdivision\_ranks"("id");

ALTER TABLE "military\_subdivisions" ADD CONSTRAINT "military\_subdivisions\_fk3" FOREIGN KEY ("part\_of") REFERENCES "military\_subdivisions"("id");

ALTER TABLE "employees" ADD CONSTRAINT "employees\_fk1" FOREIGN KEY ("rank") REFERENCES "employee\_ranks"("id");

ALTER TABLE "employees" ADD CONSTRAINT "employees\_fk2" FOREIGN KEY ("work") REFERENCES "military\_subdivisions"("id");

ALTER TABLE "employees" ADD CONSTRAINT "employees\_fk3" FOREIGN KEY ("head") REFERENCES "military\_subdivisions"("id");

ALTER TABLE "generals" ADD CONSTRAINT "generals\_fk0" FOREIGN KEY ("id") REFERENCES "employees"("id");

ALTER TABLE "colonels" ADD CONSTRAINT "colonels\_fk0" FOREIGN KEY ("id") REFERENCES "employees"("id");

ALTER TABLE "lieutenant\_colonels" ADD CONSTRAINT "lieutenant colonels\_fk0" FOREIGN KEY ("id") REFERENCES "employees"("id");

ALTER TABLE "majors" ADD CONSTRAINT "majors\_fk0" FOREIGN KEY ("id") REFERENCES "employees"("id");

ALTER TABLE "captains" ADD CONSTRAINT "captains\_fk0" FOREIGN KEY ("id") REFERENCES "employees"("id");

ALTER TABLE "lieutenants" ADD CONSTRAINT "lieutenants\_fk0" FOREIGN KEY ("id") REFERENCES "employees"("id");

ALTER TABLE "sergeantmajors" ADD CONSTRAINT "sergeantmajors\_fk0" FOREIGN KEY ("id") REFERENCES "employees"("id");

ALTER TABLE "sergeants" ADD CONSTRAINT "sergeants\_fk0" FOREIGN KEY ("id") REFERENCES "employees"("id");

ALTER TABLE "corporals" ADD CONSTRAINT "corporals\_fk0" FOREIGN KEY ("id") REFERENCES "employees"("id");

ALTER TABLE "soldiers" ADD CONSTRAINT "soldiers\_fk0" FOREIGN KEY ("id") REFERENCES "employees"("id");

ALTER TABLE "ensigns" ADD CONSTRAINT "ensigns\_fk0" FOREIGN KEY ("id") REFERENCES "employees"("id");

ALTER TABLE "infantry\_combat\_vehicles" ADD CONSTRAINT "infantry combat vehicles\_fk0" FOREIGN KEY ("id") REFERENCES "transport"("id");

ALTER TABLE "tanks" ADD CONSTRAINT "tanks\_fk0" FOREIGN KEY ("id") REFERENCES "transport"("id");

ALTER TABLE "tractors" ADD CONSTRAINT "tractors\_fk0" FOREIGN KEY ("id") REFERENCES "transport"("id");

ALTER TABLE "vehicles" ADD CONSTRAINT "vehicles\_fk0" FOREIGN KEY ("id") REFERENCES "transport"("id");

ALTER TABLE "rifles" ADD CONSTRAINT "rifles\_fk0" FOREIGN KEY ("id") REFERENCES "weapons"("id");

ALTER TABLE "assault\_rifles" ADD CONSTRAINT "assault rifles\_fk0" FOREIGN KEY ("id") REFERENCES "weapons"("id");

ALTER TABLE "artilleries" ADD CONSTRAINT "artilleries\_fk0" FOREIGN KEY ("id") REFERENCES "weapons"("id");

ALTER TABLE "missile\_weapons" ADD CONSTRAINT "missile\_weapons\_fk0" FOREIGN KEY ("id") REFERENCES "weapons"("id");

ALTER TABLE "employees\_specializations" ADD CONSTRAINT "employees\_specializations\_fk0" FOREIGN KEY ("employee\_id") REFERENCES "employees"("id");

ALTER TABLE "employees\_specializations" ADD CONSTRAINT "employees\_specializations\_fk1" FOREIGN KEY ("spec\_id") REFERENCES "military\_specializations"("id");

ALTER TABLE "military\_unit" ADD CONSTRAINT "military\_unit\_fk0" FOREIGN KEY ("id") REFERENCES "military\_subdivisions"("id");

ALTER TABLE "military\_unit" ADD CONSTRAINT "military\_unit\_fk1" FOREIGN KEY ("place\_id") REFERENCES "places"("id");

ALTER TABLE "places" ADD CONSTRAINT "places\_fk0" FOREIGN KEY ("region\_id") REFERENCES "military\_region"("id");

## Ограничения по поддержанию целостности

При добавлении нового воинского подразделения нужно учитывать, что у воинских подразделений есть чёткая иерархия, описанная в задании.

При добавлении нового военнослужащего нужно учитывать, что он не может управлять всеми воинскими подразделениями в зависимости от званий, это описано в задании.

Так же нужно поддерживать правильную связь идентификаторов в таблице **employees** и в таблицах для каждого воинского звания.

Аналогичная ситуация и с таблицей **transport** и **weapons**.

При добавлении нового воинского звания, типа транспорта и типа вооружения нужно создать новую таблицу для соответствующей разновидности.

В таблицах для отдельных рангов военнослужащих нужно учитывать, что вес, рост и зарплата не могут быть меньше нуля. А для полей с датой удобно было бы запретить устанавливать дату позже текущей.

Для таблицы **subdivision\_dislocation** нужно учитывать возможность разместиться в соответствующем сооружении.