**SQL-запрос для создания базы данных**

-- Database: hakaton

-- DROP DATABASE IF EXISTS hakaton;

CREATE DATABASE hakaton

WITH

OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'Russian\_Russia.1251'

LC\_CTYPE = 'Russian\_Russia.1251'

LOCALE\_PROVIDER = 'libc'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1

IS\_TEMPLATE = False;

**SQL-запрос для создания таблицы grafik (график работы отделения)**

-- Table: public.grafik

-- DROP TABLE IF EXISTS public.grafik;

CREATE TABLE IF NOT EXISTS public.grafik

(

id\_otdelenia integer NOT NULL,

den\_nedeli integer NOT NULL,

pereryv\_start time with time zone,

pereryv\_stop time with time zone,

start time with time zone NOT NULL,

stop time with time zone NOT NULL,

CONSTRAINT "idOtedelenia" FOREIGN KEY (id\_otdelenia)

REFERENCES public.otdelenia (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.grafik

OWNER to postgres;

**SQL-запрос для создания таблицы okna (окна в отделении, которые обрабатывают посетителей)**

-- Table: public.okna

-- DROP TABLE IF EXISTS public.okna;

CREATE TABLE IF NOT EXISTS public.okna

(

id\_otelenia integer NOT NULL,

id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

naim\_okna text COLLATE pg\_catalog."default" NOT NULL,

time\_obrabotki time with time zone NOT NULL,

talony\_telegram boolean NOT NULL DEFAULT false,

CONSTRAINT okna\_pkey PRIMARY KEY (id),

CONSTRAINT "idOtdelenia" FOREIGN KEY (id\_otelenia)

REFERENCES public.otdelenia (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.okna

OWNER to postgres;

**SQL-запрос для создания таблицы otdelenia (отделения почты Донбасса)**

-- Table: public.otdelenia

-- DROP TABLE IF EXISTS public.otdelenia;

CREATE TABLE IF NOT EXISTS public.otdelenia

(

id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

adres text COLLATE pg\_catalog."default" NOT NULL,

naimenovanie text COLLATE pg\_catalog."default" NOT NULL,

ecs boolean NOT NULL,

index text COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT otdelenia\_pkey PRIMARY KEY (id)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.otdelenia

OWNER to postgres;

**SQL-запрос для создания таблицы role (должность занимаемая сотрудником, 1 – администратор, 2 – руководитель отдела, 3- оператор)**

-- Table: public.role

-- DROP TABLE IF EXISTS public.role;

CREATE TABLE IF NOT EXISTS public.role

(

id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

naim text COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT role\_pkey PRIMARY KEY (id)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.role

OWNER to postgres;

**SQL-запрос для создания таблицы sotr (список сотрудников)**

-- Table: public.sotr

-- DROP TABLE IF EXISTS public.sotr;

CREATE TABLE IF NOT EXISTS public.sotr

(

id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

login text COLLATE pg\_catalog."default" NOT NULL,

password text COLLATE pg\_catalog."default" NOT NULL,

id\_roli integer NOT NULL,

fio text COLLATE pg\_catalog."default" NOT NULL,

id\_okna integer,

CONSTRAINT sotr\_pkey PRIMARY KEY (id),

CONSTRAINT "idOkna" FOREIGN KEY (id\_okna)

REFERENCES public.okna (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT "idRoli" FOREIGN KEY (id\_roli)

REFERENCES public.role (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.sotr

OWNER to postgres;

**SQL-запрос для создания таблицы talony (список талонов)**

-- Table: public.talony

-- DROP TABLE IF EXISTS public.talony;

CREATE TABLE IF NOT EXISTS public.talony

(

id\_okna integer NOT NULL,

date date NOT NULL,

"time" time with time zone NOT NULL,

nom\_talona integer NOT NULL,

obrabotan boolean NOT NULL DEFAULT false,

id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

id\_uslugi integer NOT NULL,

CONSTRAINT talony\_pkey PRIMARY KEY (id),

CONSTRAINT "idOkna" FOREIGN KEY (id\_okna)

REFERENCES public.okna (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT "idUslugi" FOREIGN KEY (id\_uslugi)

REFERENCES public.uslugi (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.talony

OWNER to postgres;

**SQL-запрос для создания таблицы talony\_telegram (список талонов, оформленных через телеграмм, с помощью нее можно отследить сколько талонов пользователь взял за день и при необходимости не выдавать больше n-го количества)**

-- Table: public.talony\_telegram

-- DROP TABLE IF EXISTS public.talony\_telegram;

CREATE TABLE IF NOT EXISTS public.talony\_telegram

(

id\_talona integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

id\_acc integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

CONSTRAINT "idAcc" FOREIGN KEY (id\_acc)

REFERENCES public.telegram\_acc (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT "idTalona" FOREIGN KEY (id\_talona)

REFERENCES public.talony (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.talony\_telegram

OWNER to postgres;

**SQL-запрос для создания таблицы telegram\_acc (зарегистрированные телеграмм аккаунты)**

-- Table: public.telegram\_acc

-- DROP TABLE IF EXISTS public.telegram\_acc;

CREATE TABLE IF NOT EXISTS public.telegram\_acc

(

id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

fio text COLLATE pg\_catalog."default" NOT NULL,

usermane text COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT telegram\_acc\_pkey PRIMARY KEY (id)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.telegram\_acc

OWNER to postgres;

**SQL-запрос для создания таблицы terminal (описывает данные теминала, с помощью его id и пароля можно на него зайти и оформлять талоны)**

-- Table: public.terminal

-- DROP TABLE IF EXISTS public.terminal;

CREATE TABLE IF NOT EXISTS public.terminal

(

id\_otdelenia integer NOT NULL,

id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

password text COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "idOtdelenia" FOREIGN KEY (id\_otdelenia)

REFERENCES public.otdelenia (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.terminal

OWNER to postgres;

**SQL-запрос для создания таблицы uslugi (список услуг, которые могут предоставляться)**

-- Table: public.uslugi

-- DROP TABLE IF EXISTS public.uslugi;

CREATE TABLE IF NOT EXISTS public.uslugi

(

id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

naim text COLLATE pg\_catalog."default" NOT NULL,

shifr text COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT uslugi\_pkey PRIMARY KEY (id)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.uslugi

OWNER to postgres;

**SQL-запрос для создания таблицы uslugi\_okna (описывает какие услуги может предоставлять окно и в каком интервале времени)**

-- Table: public.uslugi\_okna

-- DROP TABLE IF EXISTS public.uslugi\_okna;

CREATE TABLE IF NOT EXISTS public.uslugi\_okna

(

id\_okna integer NOT NULL,

id\_uslugi integer,

time\_start time with time zone DEFAULT '00:00:00+03'::time with time zone,

time\_stop time with time zone DEFAULT '23:59:59+03'::time with time zone,

CONSTRAINT "idOkna" FOREIGN KEY (id\_okna)

REFERENCES public.okna (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT "idUslugi" FOREIGN KEY (id\_uslugi)

REFERENCES public.uslugi (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.uslugi\_okna

OWNER to postgres;

Схема базы данных

