

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

старший преподаватель

должность, уч. степень, звание

М. В. Величко

инициалы, фамилия

подпись, дата

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Лабораторная работа №9
Объектно-реляционные базы
данных. Проектирование и
создание

по курсу: «Проектирование баз данных»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

Z1431

подпись, дата
фамилия

М.Д..Быстров

инициалы,

Санкт-Петербург 2024

ЗАДАНИЕ

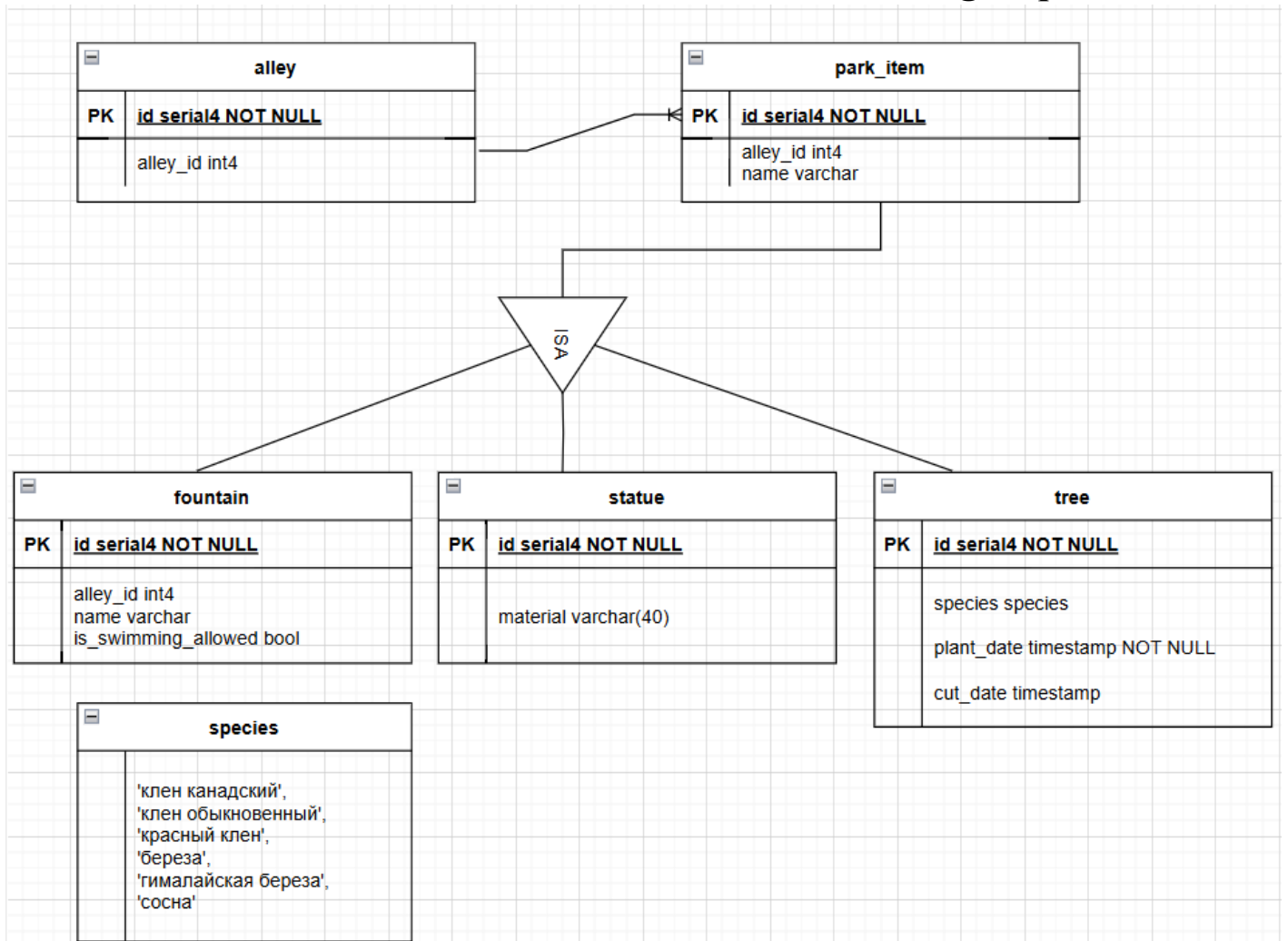
Спроектировать физическую модель базы данных, находящуюся в третьей нормальной форме и включающей наследование и хотя бы один пользовательский тип в соответствии с заданным вариантом. Написать соответствующий скрипт создания базы данных

Вариант 4:

парк: статуи, фонтаны, деревья ,породы, дата высадки, дата обрезки, расположение, аллеи

- а. аллеи, на которых встречаются разные виды кленов (клен в названии)
- б. аллеи, на которых есть и статуи и фонтаны
- в. дерево, которое было посажено позже всех
- г. порода, деревьев которой больше всего
- д. аллея, на которой нет фонтанов

Физическая модель БД для СУБД Postgresql



Скрипт создания базы данных

```
CREATE SCHEMA public;

ALTER SCHEMA public OWNER TO pg_database_owner;

COMMENT ON SCHEMA public IS 'standard public schema';

CREATE TYPE public.species AS ENUM (
    'клен канадский',
    'клен обыкновенный',
    'красный клен',
    'береза',
    'гималайская береза',
    'сосна'
);

ALTER TYPE public.species OWNER TO postgres;

SET default_tablespace = '';

SET default_table_access_method = heap;

CREATE TABLE public.alley (
    id integer NOT NULL,
    num integer
);

ALTER TABLE public.alley OWNER TO postgres;

CREATE SEQUENCE public.alley_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER SEQUENCE public.alley_id_seq OWNER TO postgres;

ALTER SEQUENCE public.alley_id_seq OWNED BY public.alley.id;

CREATE TABLE public.park_item (
```

```

    id integer NOT NULL,
    alley_id integer
);

ALTER TABLE public.park_item OWNER TO postgres;

COMMENT ON TABLE public.park_item IS 'Главная таблица элементов парка';

COMMENT ON COLUMN public.park_item.alley_id IS 'id аллеи';

CREATE TABLE public.fountain (
    is_swimming_allowed boolean
)
INHERITS (public.park_item);

ALTER TABLE public.fountain OWNER TO postgres;

CREATE SEQUENCE public.park_item_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER SEQUENCE public.park_item_id_seq OWNER TO postgres;

ALTER SEQUENCE public.park_item_id_seq OWNED BY public.park_item.id;

CREATE TABLE public.statue (
    material character varying(40)
)
INHERITS (public.park_item);

ALTER TABLE public.statue OWNER TO postgres;

CREATE TABLE public.tree (
    species public.species,
    plant_date timestamp without time zone NOT NULL,
    cut_date timestamp without time zone
)
INHERITS (public.park_item);

ALTER TABLE public.tree OWNER TO postgres;

```

```
ALTER TABLE ONLY public.alley ALTER COLUMN id SET DEFAULT
nextval('public.alley_id_seq'::regclass);

ALTER TABLE ONLY public.fountain ALTER COLUMN id SET DEFAULT
nextval('public.park_item_id_seq'::regclass);

ALTER TABLE ONLY public.park_item ALTER COLUMN id SET DEFAULT
nextval('public.park_item_id_seq'::regclass);

ALTER TABLE ONLY public.statue ALTER COLUMN id SET DEFAULT
nextval('public.park_item_id_seq'::regclass);

ALTER TABLE ONLY public.tree ALTER COLUMN id SET DEFAULT
nextval('public.park_item_id_seq'::regclass);

SELECT pg_catalog.setval('public.alley_id_seq', 1, false);

SELECT pg_catalog.setval('public.park_item_id_seq', 1, false);

ALTER TABLE ONLY public.alley
    ADD CONSTRAINT alley_pk PRIMARY KEY (id);

ALTER TABLE ONLY public.park_item
    ADD CONSTRAINT park_item_pk PRIMARY KEY (id);

ALTER TABLE ONLY public.fountain
    ADD CONSTRAINT fountain_unique UNIQUE (id);

ALTER TABLE ONLY public.statue
    ADD CONSTRAINT statue_unique UNIQUE (id);

ALTER TABLE ONLY public.tree
    ADD CONSTRAINT tree_unique UNIQUE (id);

ALTER TABLE ONLY public.fountain
    ADD CONSTRAINT fountain_alley_fk FOREIGN KEY (alley_id) REFERENCES
public.alley(id) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY public.park_item
    ADD CONSTRAINT park_item_alley_fk FOREIGN KEY (alley_id) REFERENCES
public.alley(id) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY public.statue
```

```
ADD CONSTRAINT statue_alley_fk FOREIGN KEY (alley_id) REFERENCES public.alley(id)
ON UPDATE RESTRICT ON DELETE RESTRICT;
```

```
ALTER TABLE ONLY public.tree
ADD CONSTRAINT tree_alley_fk FOREIGN KEY (alley_id) REFERENCES public.alley(id)
ON UPDATE RESTRICT ON DELETE RESTRICT;
```

```
CREATE OR REPLACE FUNCTION public.park_item_check_id()
RETURNS trigger
LANGUAGE plpgsql
```

```
AS $function$
```

```
DECLARE
```

```
is_found boolean;
```

```
BEGIN
```

```
select exists(select * from park_item i
where i.id = NEW.id)
into is_found;
```

```
if (is_found = true) then
RAISE EXCEPTION 'duplicate ids';
end if;
```

```
return new;
```

```
END;
```

```
$function$
```

```
;
```

```
CREATE TRIGGER check_id before insert or update ON public.park_item FOR EACH ROW
EXECUTE FUNCTION public.park_item_check_id();
```

```
CREATE TRIGGER check_id before insert or update ON public.fountain FOR EACH ROW
EXECUTE FUNCTION public.park_item_check_id();
```

```
CREATE TRIGGER check_id before insert or update ON public.tree FOR EACH ROW EXECUTE
FUNCTION public.park_item_check_id();
```

```
CREATE TRIGGER check_id before insert or update ON public.statue FOR EACH ROW EXECUTE
FUNCTION public.park_item_check_id();
```

```
REVOKE USAGE ON SCHEMA public FROM PUBLIC;
```

Вывод

В ходе выполнения 9 лабораторной работы была спроектирована и создана реляционная база данных в СУБД Postgresql. В базе данных присутствует созданный пользовательский тип (перечисление) и использовано наследование при создании таблиц.

В отчете представлен скрипт по созданию базы данных. Версия СУБД Postgresql – 17.