ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет инфокоммуникационных технологий

Дисциплина:
«Базы данных»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

«Создание БД в СУБД PostgreSQL. Резервное копирование и восстановление БД»

Выполнила:
студентка группы K32421
Панкова Кристина
Сергеевна

(подпись)

Проверила:
Говорова Марина Михайловна

(отметка о выполнении)

(подпись)

Санкт-Петербург
2023 г.

Цель работы: овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.

Практическое задание:

1. Создать базу данных с использованием pgAdmin 4 (согласно индивидуальному заданию).
2. Создать схему в составе базы данных.
3. Создать таблицы базы данных.
4. Установить ограничения на данные: *Primary Key, Unique, Check, Foreign Key*.
5. Заполнить таблицы БД рабочими данными.
6. Создать резервную копию БД.
7. *Указание:*
8. *Создать две резервные копии:*
9. *с расширением CUSTOM для восстановления БД;*
10. *с расширением PLAIN для листинга (в отчете);*
11. *при создании резервных копий БД настроить параметры Dump options для Type of objects и Queries* .
12. Восстановить БД.

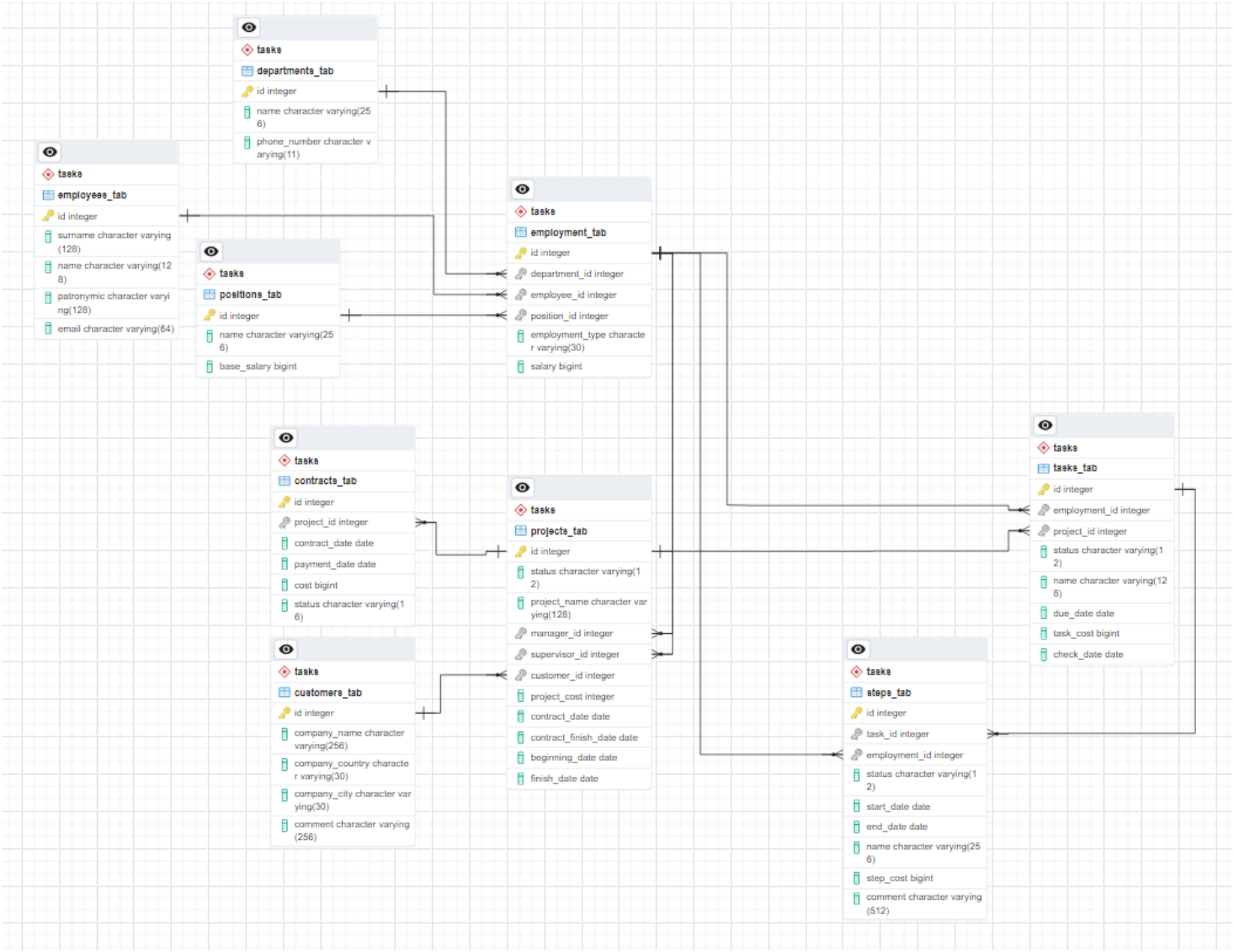
Индивидуальное задание: Вариант 4

Ход выполнения работы:

Название БД:

БД «Учет выполнения заданий».

Схема логической модели базы данных, сгенерированная в Generate ERD:



pg_dump:

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 14.7
-- Dumped by pg_dump version 14.7

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;
```

```
-- Name: tasks; Type: DATABASE; Schema: -; Owner: postgres

CREATE DATABASE tasks WITH TEMPLATE = template0 ENCODING = 'UTF8' LOCALE =
'Russian_Russia.1251';


ALTER DATABASE tasks OWNER TO postgres;

\connect tasks

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;


COMMENT ON DATABASE tasks IS 'Database for task-tracking service';

-- tasks - схема

CREATE SCHEMA tasks;

ALTER SCHEMA tasks OWNER TO postgres;

COMMENT ON SCHEMA tasks IS 'Schema for task-tracking service';

SET default_tablespace = '';

SET default_table_access_method = heap;

-- contracts_tab - таблица договоров и привязанных к ним счетов

CREATE TABLE tasks.contracts_tab (
    id integer NOT NULL,
    project_id integer NOT NULL,
    contract_date date NOT NULL,
    payment_date date,
    cost bigint NOT NULL,
    status character varying(16) NOT NULL,
    CONSTRAINT contracts_tab_cost_check CHECK ((cost > 0)),
    CONSTRAINT contracts_tab_status_check CHECK (((status)::text = ANY
((ARRAY['Not_started'::character varying, 'Prepayment_done'::character varying,
'Fully_payed'::character varying])::text[])))
);
```

```
ALTER TABLE tasks.contracts_tab OWNER TO postgres;

CREATE SEQUENCE tasks.contracts_tab_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE tasks.contracts_tab_id_seq OWNER TO postgres;

ALTER SEQUENCE tasks.contracts_tab_id_seq OWNED BY tasks.contracts_tab.id;

-- customers_tab - таблица с компаниями-заказчиками

CREATE TABLE tasks.customers_tab (
    id integer NOT NULL,
    company_name character varying(256) NOT NULL,
    company_country character varying(30) NOT NULL,
    company_city character varying(30) NOT NULL,
    comment character varying(256)
);


ALTER TABLE tasks.customers_tab OWNER TO postgres;


CREATE SEQUENCE tasks.customers_tab_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE tasks.customers_tab_id_seq OWNER TO postgres;

ALTER SEQUENCE tasks.customers_tab_id_seq OWNED BY tasks.customers_tab.id;

-- departments_tab - таблица отделов

CREATE TABLE tasks.departments_tab (
    id integer NOT NULL,
    name character varying(256) NOT NULL,
    phone_number character varying(11) NOT NULL
);
```

```
ALTER TABLE tasks.departments_tab OWNER TO postgres;


CREATE SEQUENCE tasks.departments_tab_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE tasks.departments_tab_id_seq OWNER TO postgres;

ALTER SEQUENCE tasks.departments_tab_id_seq OWNED BY tasks.departments_tab.id;

-- employees_tab - таблица работников

CREATE TABLE tasks.employees_tab (
    id integer NOT NULL,
    surname character varying(128) NOT NULL,
    name character varying(128) NOT NULL,
    patronymic character varying(128),
    email character varying(64) NOT NULL
);


ALTER TABLE tasks.employees_tab OWNER TO postgres;


CREATE SEQUENCE tasks.employees_tab_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE tasks.employees_tab_id_seq OWNER TO postgres;


ALTER SEQUENCE tasks.employees_tab_id_seq OWNED BY tasks.employees_tab.id;

-- employment_tab - таблица занятости

CREATE TABLE tasks.employment_tab (
    id integer NOT NULL,
    department_id integer NOT NULL,
    employee_id integer NOT NULL,
```

```
    position_id integer NOT NULL,
    employment_type character varying(30) NOT NULL,
    salary bigint NOT NULL
);


ALTER TABLE tasks.employment_tab OWNER TO postgres;

CREATE SEQUENCE tasks.employment_tab_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE tasks.employment_tab_id_seq OWNER TO postgres;


ALTER SEQUENCE tasks.employment_tab_id_seq OWNED BY tasks.employment_tab.id;

-- positions_tab - таблица с должностями

CREATE TABLE tasks.positions_tab (
    id integer NOT NULL,
    name character varying(256) NOT NULL,
    base_salary bigint NOT NULL
);


ALTER TABLE tasks.positions_tab OWNER TO postgres;

CREATE SEQUENCE tasks.positions_tab_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE tasks.positions_tab_id_seq OWNER TO postgres;

ALTER SEQUENCE tasks.positions_tab_id_seq OWNED BY tasks.positions_tab.id;

-- projects_tab таблица с проектами

CREATE TABLE tasks.projects_tab (
    id integer NOT NULL,
    status character varying(12) NOT NULL,
```

```
    project_name character varying(128) NOT NULL,
    manager_id integer NOT NULL,
    supervisor_id integer NOT NULL,
    customer_id integer NOT NULL,
    project_cost integer NOT NULL,
    contract_date date NOT NULL,
    contract_finish_date date,
    beginning_date date,
    finish_date date,
    CONSTRAINT projects_tab_beginning_date_check CHECK ((beginning_date >
'2000-01-01'::date)),
    CONSTRAINT projects_tab_contract_date_check CHECK ((contract_date > '2000-01-01'::date)),
    CONSTRAINT projects_tab_contract_finish_date_check CHECK ((contract_finish_date >
'2000-01-01'::date)),
    CONSTRAINT projects_tab_finish_date_check CHECK ((finish_date > '2000-01-01'::date)),
    CONSTRAINT projects_tab_project_cost_check CHECK ((project_cost > 0)),
    CONSTRAINT projects_tab_status_check CHECK (((status)::text = ANY
((ARRAY['Not_started'::character varying, 'In_progress'::character varying,
'Frozen'::character varying, 'Done'::character varying])::text[])))
);


ALTER TABLE tasks.projects_tab OWNER TO postgres;

CREATE SEQUENCE tasks.projects_tab_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE tasks.projects_tab_id_seq OWNER TO postgres;

ALTER SEQUENCE tasks.projects_tab_id_seq OWNED BY tasks.projects_tab.id;

-- steps_tab таблица с шагами по заданиям

CREATE TABLE tasks.steps_tab (
    id integer NOT NULL,
    task_id integer NOT NULL,
    employment_id integer NOT NULL,
    status character varying(12) NOT NULL,
    start_date date,
    end_date date,
    name character varying(256) NOT NULL,
    step_cost bigint NOT NULL,
    comment character varying(512),
    CONSTRAINT steps_tab_end_date_check CHECK ((end_date > '2000-01-01'::date)),
    CONSTRAINT steps_tab_start_date_check CHECK ((start_date > '2000-01-01'::date)),
```

```
    CONSTRAINT steps_tab_status_check CHECK (((status)::text = ANY
((ARRAY['Not_started'::character varying, 'In_progress'::character varying,
'Frozen'::character varying, 'Done'::character varying])::text[]))),
    CONSTRAINT steps_tab_step_cost_check CHECK ((step_cost > 0))
);


ALTER TABLE tasks.steps_tab OWNER TO postgres;


CREATE SEQUENCE tasks.steps_tab_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE tasks.steps_tab_id_seq OWNER TO postgres;

ALTER SEQUENCE tasks.steps_tab_id_seq OWNED BY tasks.steps_tab.id;

-- tasks_tab - таблица с заданиями

CREATE TABLE tasks.tasks_tab (
    id integer NOT NULL,
    employment_id integer NOT NULL,
    project_id integer NOT NULL,
    status character varying(12) NOT NULL,
    name character varying(128) NOT NULL,
    due_date date,
    task_cost bigint NOT NULL,
    check_date date,
    CONSTRAINT tasks_tab_check_date_check CHECK ((check_date > '2000-01-01'::date)),
    CONSTRAINT tasks_tab_due_date_check CHECK ((due_date > '2000-01-01'::date)),
    CONSTRAINT tasks_tab_status_check CHECK (((status)::text = ANY
((ARRAY['Not_started'::character varying, 'In_progress'::character varying,
'Frozen'::character varying, 'Done'::character varying])::text[]))),
    CONSTRAINT tasks_tab_task_cost_check CHECK ((task_cost > 0))
);


ALTER TABLE tasks.tasks_tab OWNER TO postgres;

CREATE SEQUENCE tasks.tasks_tab_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
```

```
    CACHE 1;


ALTER TABLE tasks.tasks_tab_id_seq OWNER TO postgres;

ALTER SEQUENCE tasks.tasks_tab_id_seq OWNED BY tasks.tasks_tab.id;


ALTER TABLE ONLY tasks.contracts_tab ALTER COLUMN id SET DEFAULT
nextval('tasks.contracts_tab_id_seq'::regclass);
ALTER TABLE ONLY tasks.customers_tab ALTER COLUMN id SET DEFAULT
nextval('tasks.customers_tab_id_seq'::regclass);
ALTER TABLE ONLY tasks.departments_tab ALTER COLUMN id SET DEFAULT
nextval('tasks.departments_tab_id_seq'::regclass);
ALTER TABLE ONLY tasks.employees_tab ALTER COLUMN id SET DEFAULT
nextval('tasks.employees_tab_id_seq'::regclass);
ALTER TABLE ONLY tasks.employment_tab ALTER COLUMN id SET DEFAULT
nextval('tasks.employment_tab_id_seq'::regclass);
ALTER TABLE ONLY tasks.positions_tab ALTER COLUMN id SET DEFAULT
nextval('tasks.positions_tab_id_seq'::regclass);
ALTER TABLE ONLY tasks.projects_tab ALTER COLUMN id SET DEFAULT
nextval('tasks.projects_tab_id_seq'::regclass);
ALTER TABLE ONLY tasks.steps_tab ALTER COLUMN id SET DEFAULT
nextval('tasks.steps_tab_id_seq'::regclass);
ALTER TABLE ONLY tasks.tasks_tab ALTER COLUMN id SET DEFAULT
nextval('tasks.tasks_tab_id_seq'::regclass);

-- вставка данных

INSERT INTO tasks.contracts_tab VALUES (5, 1, '2023-03-01', '2023-03-02', 1000000,
'Fully_payed');
INSERT INTO tasks.contracts_tab VALUES (6, 2, '2023-03-01', '2023-03-02', 1000000,
'Fully_payed');
INSERT INTO tasks.contracts_tab VALUES (7, 3, '2023-02-01', '2023-03-02', 100000,
'Prepayment_done');
INSERT INTO tasks.contracts_tab VALUES (8, 3, '2023-02-01', NULL, 9000000, 'Not_started');



INSERT INTO tasks.customers_tab VALUES (1, 'Apple', 'USA', 'New York', NULL);
INSERT INTO tasks.customers_tab VALUES (2, 'Yandex', 'Russia', 'Moscow', NULL);
INSERT INTO tasks.customers_tab VALUES (3, 'X5 Group', 'Russia', 'Moscow', NULL);
INSERT INTO tasks.customers_tab VALUES (4, 'Studio Ghibli', 'Japan', 'Tokyo', NULL);


INSERT INTO tasks.departments_tab VALUES (1, 'Tech', '89112223454');
INSERT INTO tasks.departments_tab VALUES (2, 'Management', '89765796543');
INSERT INTO tasks.departments_tab VALUES (3, 'Animation', '87657962342');

INSERT INTO tasks.employees_tab VALUES (1, 'Litina', 'Kseniya', 'Sergeevna',
'litinaksu@yandex.ru');
```

```
INSERT INTO tasks.employees_tab VALUES (2, 'Lmaovets', 'Pavel', NULL,
'pashadotnet@gmail.com');
INSERT INTO tasks.employees_tab VALUES (3, 'Solovyova', 'Anna', 'Pavlova',
'hamlover@yandex.ru');


INSERT INTO tasks.employment_tab VALUES (1, 1, 1, 1, 'full-time', 100000);
INSERT INTO tasks.employment_tab VALUES (2, 1, 2, 1, 'part-time', 50000);
INSERT INTO tasks.employment_tab VALUES (3, 2, 2, 1, 'part-time', 60000);
INSERT INTO tasks.employment_tab VALUES (4, 3, 3, 3, 'full-time', 950000);


INSERT INTO tasks.positions_tab VALUES (1, 'Developer', 100000);
INSERT INTO tasks.positions_tab VALUES (2, 'Manager', 100000);
INSERT INTO tasks.positions_tab VALUES (3, 'Animator', 850000);

INSERT INTO tasks.projects_tab VALUES (1, 'Not_started', 'Management reorganisation', 2, 2,
3, 1000000, '2023-03-01', '2023-09-01', '2023-04-01', NULL);
INSERT INTO tasks.projects_tab VALUES (2, 'In_progress', 'Software development', 1, 2, 2,
5000000, '2023-02-01', '2024-01-01', '2023-03-01', NULL);
INSERT INTO tasks.projects_tab VALUES (3, 'In_progress', 'Animation outsourcing', 3, 1, 4,
1000000, '2023-02-01', '2023-06-01', '2023-02-14', NULL);

INSERT INTO tasks.steps_tab VALUES (1, 1, 1, 'In_progress', '2023-03-26', '2023-03-29',
'Logging configurations and tests', 10000, NULL);
INSERT INTO tasks.steps_tab VALUES (2, 2, 3, 'Done', '2023-03-01', '2023-04-01', 'Main scene
finishing touches', 5000, NULL);

INSERT INTO tasks.tasks_tab VALUES (1, 1, 2, 'In_progress', 'Java to Kotlin migration',
'2023-05-01', 100000, '2023-04-01');
INSERT INTO tasks.tasks_tab VALUES (2, 3, 3, 'In_progress', 'Animation skits', '2023-04-01',
500000, '2023-03-29');

-- pg_catalog
/*
SELECT pg_catalog.setval('pgc_uncle_bins.version_info_id_seq', 1, false);
SELECT pg_catalog.setval('tasks.contracts_tab_id_seq', 8, true);
SELECT pg_catalog.setval('tasks.contracts_tab_project_id_seq', 1, false);
SELECT pg_catalog.setval('tasks.customers_tab_id_seq', 4, true);
SELECT pg_catalog.setval('tasks.departments_tab_id_seq', 3, true);
SELECT pg_catalog.setval('tasks.employees_tab_id_seq', 3, true);
SELECT pg_catalog.setval('tasks.employment_tab_department_id_seq', 1, false);
SELECT pg_catalog.setval('tasks.employment_tab_employee_id_seq', 1, false);
SELECT pg_catalog.setval('tasks.employment_tab_id_seq', 4, true);
SELECT pg_catalog.setval('tasks.employment_tab_position_id_seq', 1, false);
SELECT pg_catalog.setval('tasks.positions_tab_id_seq', 3, true);
SELECT pg_catalog.setval('tasks.projects_tab_customer_id_seq', 1, false);
SELECT pg_catalog.setval('tasks.projects_tab_id_seq', 3, true);
SELECT pg_catalog.setval('tasks.projects_tab_manager_id_seq', 1, false);
SELECT pg_catalog.setval('tasks.projects_tab_project_cost_seq', 1, false);
SELECT pg_catalog.setval('tasks.projects_tab_supervisor_id_seq', 1, false);
```

```
SELECT pg_catalog.setval('tasks.steps_tab_employment_id_seq', 1, false);
SELECT pg_catalog.setval('tasks.steps_tab_id_seq', 2, true);
SELECT pg_catalog.setval('tasks.steps_tab_task_id_seq', 1, false);
SELECT pg_catalog.setval('tasks.tasks_tab_employment_id_seq', 1, false);
SELECT pg_catalog.setval('tasks.tasks_tab_id_seq', 2, true);
SELECT pg_catalog.setval('tasks.tasks_tab_project_id_seq', 1, false); */

-- ограничения целостности

ALTER TABLE ONLY tasks.contracts_tab
    ADD CONSTRAINT contracts_tab_pkey PRIMARY KEY (id);

ALTER TABLE ONLY tasks.customers_tab
    ADD CONSTRAINT customers_tab_pkey PRIMARY KEY (id);

ALTER TABLE ONLY tasks.departments_tab
    ADD CONSTRAINT departments_tab_pkey PRIMARY KEY (id);

ALTER TABLE ONLY tasks.employees_tab
    ADD CONSTRAINT employees_tab_pkey PRIMARY KEY (id);

ALTER TABLE ONLY tasks.employment_tab
    ADD CONSTRAINT employment_tab_department_id_employee_id_position_id_key UNIQUE
(department_id, employee_id, position_id);

ALTER TABLE ONLY tasks.employment_tab
    ADD CONSTRAINT employment_tab_pkey PRIMARY KEY (id);

ALTER TABLE ONLY tasks.positions_tab
    ADD CONSTRAINT positions_tab_pkey PRIMARY KEY (id);

ALTER TABLE ONLY tasks.projects_tab
    ADD CONSTRAINT projects_tab_pkey PRIMARY KEY (id);

ALTER TABLE ONLY tasks.steps_tab
    ADD CONSTRAINT steps_tab_pkey PRIMARY KEY (id);

ALTER TABLE ONLY tasks.tasks_tab
    ADD CONSTRAINT tasks_tab_pkey PRIMARY KEY (id);

ALTER TABLE ONLY tasks.contracts_tab
    ADD CONSTRAINT contracts_tab_project_id_fkey FOREIGN KEY (project_id) REFERENCES
tasks.projects_tab(id);

ALTER TABLE ONLY tasks.employment_tab
    ADD CONSTRAINT employment_tab_department_id_fkey FOREIGN KEY (department_id) REFERENCES
tasks.departments_tab(id);

ALTER TABLE ONLY tasks.employment_tab
    ADD CONSTRAINT employment_tab_employee_id_fkey FOREIGN KEY (employee_id) REFERENCES
tasks.employees_tab(id);
```

```
ALTER TABLE ONLY tasks.employment_tab
    ADD CONSTRAINT employment_tab_position_id_fkey FOREIGN KEY (position_id) REFERENCES
tasks.positions_tab(id);

ALTER TABLE ONLY tasks.projects_tab
    ADD CONSTRAINT projects_tab_customer_id_fkey FOREIGN KEY (customer_id) REFERENCES
tasks.customers_tab(id);

ALTER TABLE ONLY tasks.projects_tab
    ADD CONSTRAINT projects_tab_manager_id_fkey FOREIGN KEY (manager_id) REFERENCES
tasks.employment_tab(id);

ALTER TABLE ONLY tasks.projects_tab
    ADD CONSTRAINT projects_tab_supervisor_id_fkey FOREIGN KEY (supervisor_id) REFERENCES
tasks.employment_tab(id);

ALTER TABLE ONLY tasks.steps_tab
    ADD CONSTRAINT steps_tab_employment_id_fkey FOREIGN KEY (employment_id) REFERENCES
tasks.employment_tab(id);

ALTER TABLE ONLY tasks.steps_tab
    ADD CONSTRAINT steps_tab_task_id_fkey FOREIGN KEY (task_id) REFERENCES
tasks.tasks_tab(id);

ALTER TABLE ONLY tasks.tasks_tab
    ADD CONSTRAINT tasks_tab_employment_id_fkey FOREIGN KEY (employment_id) REFERENCES
tasks.employment_tab(id);

ALTER TABLE ONLY tasks.tasks_tab
    ADD CONSTRAINT tasks_tab_project_id_fkey FOREIGN KEY (project_id) REFERENCES
tasks.projects_tab(id);


-- Completed on 2023-03-27 01:53:30

--
-- PostgreSQL database dump complete
--
```

Вывод:

В процессе выполнения данной лабораторной работы я овладела практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.