

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИТМО»**

**Отчет**

по лабораторной работе «Создание БД PostgreSQL в pgAdmin. Резервное  
копирование и восстановление БД»  
по дисциплине «**Базы данных**»

Автор: Зайцев Кирилл Дмитриевич

Факультет: Инфокоммуникационных технологий (ИКТ)

Группа: K32402

Преподаватель: Говорова М. М.

Дата: 24.04.2023

**ИТМО**

Санкт-Петербург 2022

**Цель работы:** овладение практическими навыками установки СУБД PostgreSQL и создания базы данных в pgadmin 4, создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД

**ЛР 1.1 Установка PostgreSQL и pgAdmin. Создание БД**

**ЛР 1.2 Создание БД PostgreSQL в pgAdmin. Резервное копирование и восстановление 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.

Указание:

Создать две резервные копии:

- с расширением *CUSTOM* для восстановления БД;
- с расширением *PLAIN* для листинга (в отчете);
- при создании резервных копий БД настроить параметры *Dump options* для *Type of objects* и *Queries* .

7. Восстановить БД.

**Код создания таблиц, схемы с установленными ограничениями:**

```
drop table bilet;  
drop table passazhir;  
drop table kassir;  
drop table servis;  
drop table mesto;  
drop table ekipazh;  
drop table samolet;  
drop table modelka;  
drop table tranzitnieReisi;  
drop table sotrudnik;  
drop table kassa;  
drop table reis;  
drop table raspisanieReisov;  
drop table aeroportNaznacheniya;  
drop table aeroportVileta;  
drop table aviaCompany;  
create table aeroportVileta
```

```

(IATACodeOUT varchar(18) PRIMARY KEY,
gorod varchar(50),
strana varchar(50),
CHECK((gorod != "") AND (strana != ""))
);
create table aeroportNaznacheniya
(IATACodeIN varchar(18) PRIMARY KEY,
gorod varchar(50),
strana varchar(50),
CHECK((gorod != "") AND (strana != ""))
);
create table aviaCompany
(codeAviacompany varchar(18) PRIMARY KEY,
nazvanie varchar(18) unique,
strana varchar(18)
CHECK((nazvanie != "") AND (strana != ""))
);
create table raspisanieReisov
(codeRaspisaniya int PRIMARY KEY,
IATACodeOUT varchar(18),
IATACodeIN varchar(18),
codeAviacompany varchar(18),
vremyaOtpravleniya time,
vremyaPribitiya time,
pereodichnost interval,
tipReisa varchar(18),
CHECK((IATACodeOUT != "") AND (IATACodeIN != "") AND (codeAviacompany != "") AND
(tipReisa != "")),
FOREIGN KEY(codeAviacompany) references aviaCompany(codeAviacompany),
FOREIGN KEY(IATACodeOUT) references aeroportVileta(IATACodeOUT),
FOREIGN KEY(IATACodeIN) references aeroportNaznacheniya(IATACodeIN));
create table tranzitnieReisi
(codePereleta int PRIMARY KEY,
codeRaspisaniya int,
IATACodeOUT varchar(18),
IATACodeIN varchar(18),
codeAviacompany varchar(18),
vremyaVAeroporty time,
dataVremyaPribitiya date,
dataVremyaVileta date,
CHECK((IATACodeOUT != "") AND (IATACodeIN != "") AND (codeAviacompany != "") AND
(codeRaspisaniya > 0)),
FOREIGN KEY(codeAviacompany) references aviaCompany(codeAviacompany),
FOREIGN KEY(IATACodeOUT) references aeroportVileta(IATACodeOUT),
FOREIGN KEY(IATACodeIN) references aeroportNaznacheniya(IATACodeIN));
create table reis
(codeReisa varchar(18) PRIMARY KEY,
codeRaspisaniya int,
IATACodeOUT varchar(18),
IATACodeIN varchar(18),
codeAviacompany varchar(18),
aviacompanyIspolnitel varchar(18),

```

```

dataVremyaVileta date,
dataVremyaPribitiya date,
statusReisa varchar(20),
CHECK(((codeRaspisaniya > 0) AND (IATACodeOUT != "") AND (IATACodeIN != "") AND
(codeAviacompany != "") AND (aviacompanyIspolnitel != "") AND (statusReisa != "")),
FOREIGN KEY(codeAviacompany) references aviaCompany(codeAviacompany),
FOREIGN KEY(IATACodeOUT) references aeroportVileta(IATACodeOUT),
FOREIGN KEY(IATACodeIN) references aeroportNaznacheniya(IATACodeIN),
FOREIGN KEY(codeRaspisaniya) references raspisanieReisov(codeRaspisaniya));
create table modelka
(codeModeli int PRIMARY KEY, rashodTopliva int, gruzoPodemnost int, skorost int, tip
varchar(4), proizvoditel varchar(18), kolichestvoMest int,
CHECK((rashodTopliva > 0) AND (gruzoPodemnost > 0) AND (skorost > 0) AND (tip != "") AND
(proizvoditel != "") AND (kolichestvoMest > 0))
);
create table samolet
(bortovoiNomer varchar(18) PRIMARY KEY,
codeAviacompany varchar(18),
codeReisa varchar(18),
codeModeli int,
dataVipuska date,
naletVchasah int,
dataPoslednegoRemonta date,
CHECK(((codeAviacompany != "") AND (codeReisa != "") AND (codeModeli > 0) AND
(naletVchasah > 0)),
FOREIGN KEY(codeReisa) references reis(codeReisa),
FOREIGN KEY(codeModeli) references modelka(codeModeli),
FOREIGN KEY(codeAviacompany) references aviaCompany(codeAviacompany));
create table sotrudnik
(codeSotrudnika int PRIMARY KEY,
passport varchar(24),
FIO varchar(50),
naletVChasah int,
CHECK((passport != "") AND (FIO != "") AND (naletVChasah > 0))
);
create table ekipazh
(codeSotrudnika int,
sostoyanie bit,
rezultatMedosmotra varchar(20),
CHECK((codeSotrudnika > 0) AND (rezultatMedosmotra != "")),
FOREIGN KEY(codeSotrudnika) references sotrudnik(codeSotrudnika));
create table mesto
(codeMesta int PRIMARY KEY,
codeReisa varchar(18),
codeAviacompany varchar(18),
nomerMesta varchar(5),
tipMesta varchar(18),
cenaMesta int,
statusZanyatosti bit,
CHECK(((codeReisa != "") AND (codeAviacompany != "") AND (nomerMesta != "") AND (tipMesta
!= "") AND (cenaMesta > 0)),
FOREIGN KEY(codeAviacompany) references aviaCompany(codeAviacompany),

```

```

FOREIGN KEY(codeReisa) references reis(codeReisa));
create table passazhir
(codePassazhira int PRIMARY KEY,
passport varchar(24) unique,
FIO varchar(50),
telephone varchar(12),
pochta varchar(50),
CHECK((passport != "") AND (FIO != "") AND (telephone != "") AND (pochta != ""))
);
create table servis
(codeServisa int PRIMARY KEY,
domen varchar(50),
CHECK(domen != ""))
);
create table kassa
(codeKassi int PRIMARY KEY,
adresKassi varchar(50),
CHECK(adresKassi != ""))
);
create table kassir
(codeKassira int PRIMARY KEY,
codeKassi int,
passport varchar(24),
CHECK((codeKassi > 0) AND (passport != "")),
FOREIGN KEY(codeKassi) references kassa(codeKassi));
create table bilet
(codeBileta int PRIMARY KEY,
codeReisa varchar(18),
codeAviacompany varchar(18),
codeRaspisaniya int,
IATACodeOUT varchar(18),
IATACodeIN varchar(18),
codePassazhira int,
codeServisa int,
codeKassira int,
codeKassi int,
codeMesta int,
cena int,
statusOplati bit,
statusSostoyaniya varchar(10),
tipOplati varchar(12),
CHECK((codeReisa != "") AND (codeAviacompany != "") AND (codeRaspisaniya > 0) AND
(IATACodeOUT != "") AND (IATACodeIN != "") AND (codePassazhira > 0)
AND (codeServisa > 0) AND (codeKassira > 0) AND (codeKassi > 0) AND (codeMesta
> 0) AND (cena > 0) AND (statusSostoyaniya != "") AND (tipOplati != "")),
FOREIGN KEY(codeAviacompany) references aviaCompany(codeAviacompany),
FOREIGN KEY(codeReisa) references reis(codeReisa),
FOREIGN KEY(IATACodeOUT) references aeroportVileta(IATACodeOUT),
FOREIGN KEY(IATACodeIN) references aeroportNaznacheniya(IATACodeIN),
FOREIGN KEY(codeRaspisaniya) references
raspisanieReisov(codeRaspisaniya),
FOREIGN KEY(codePassazhira) references passazhir(codePassazhira),

```

```
FOREIGN KEY(codeServisa) references servis(codeServisa),  
FOREIGN KEY(codeKassira) references kassir(codeKassira),  
FOREIGN KEY(codeKassi) references kassa(codeKassi),  
FOREIGN KEY(codeMesta) references mesto(codeMesta));
```

Код заполнения БД:

```
insert into modelka values(1, 27, 14000, 900, 'gruz', 'Boing', 4);  
insert into modelka values(2, 20, 14000, 900, 'pass', 'Boing', 240);
```

```
insert into sotrudnik values(1, 'sotr1', 'Ivanov Ivan Ivanovich', 4);  
insert into sotrudnik values(2, 'sotr2', 'Ivanov Ivan Andreevich', 4);
```

```
insert into ekipazh values(1, '0', 'Alkogol');  
insert into ekipazh values(2, '1', 'Zdorov');
```

```
insert into aviaCompany values('lufthanza1', 'lufthanza', 'Germania');  
insert into aviaCompany values('aeroflot1', 'aeroflot', 'Russia');
```

```
insert into aeroportVileta values('ULLI', 'Sankt-Peterburg', 'Russia');  
insert into aeroportNaznacheniya values('FRFT', 'Frankfurt', 'Germania');
```

```
insert into raspisanieReisov values(1, 'ULLI', 'FRFT', 'lufthanza1', '07:20', '10:00', '8:00', 'baza');
```

```
insert into tranzitniereisi values(1, 1, 'ULLI', 'FRFT', 'lufthanza1', '1:00', '04.04.2023 10:00',  
'04.04.2023 7:20');
```

```
insert into passazhir values(1, 'pass1', 'Kirill Kirilov Kirillovich', '+79214591803',  
'kuku@itmo.ru');
```

```
insert into reis values('LK234', 1, 'ULLI', 'FRFT', 'lufthanza1', 'lufthanza', '04.04.2023 7:20',  
'04.04.2023 10:00', 'Ispolnen');
```

```
insert into samolet values('BB148', 'lufthanza1', 'LK234', 2, '01.01.2023', 300, '01.01.2023');
```

```
insert into mesto values(1, 'LK234', 'lufthanza1', 'A1', 'Econom', 100, '1');
```

```
insert into servis values(1, 'flyaway.com');
```

```
insert into kassa values(2, 'Nevski pr. 1');
```

```
insert into kassir values(3, 2, 'kas1');
```

```
insert into билет values(1, 'LK234', 'lufthanza1', 1, 'ULLI', 'FRFT', 1, 1, 3, 2, 1, 150, '1', 'vipolnen',  
'onlineCard')
```

Созданы две резервные копии расширения PLAIN и CUSTOM, с настроенными параметрами и создана тестовая БД, в который выполнялось восстановление БД на основе созданных бэкапов.

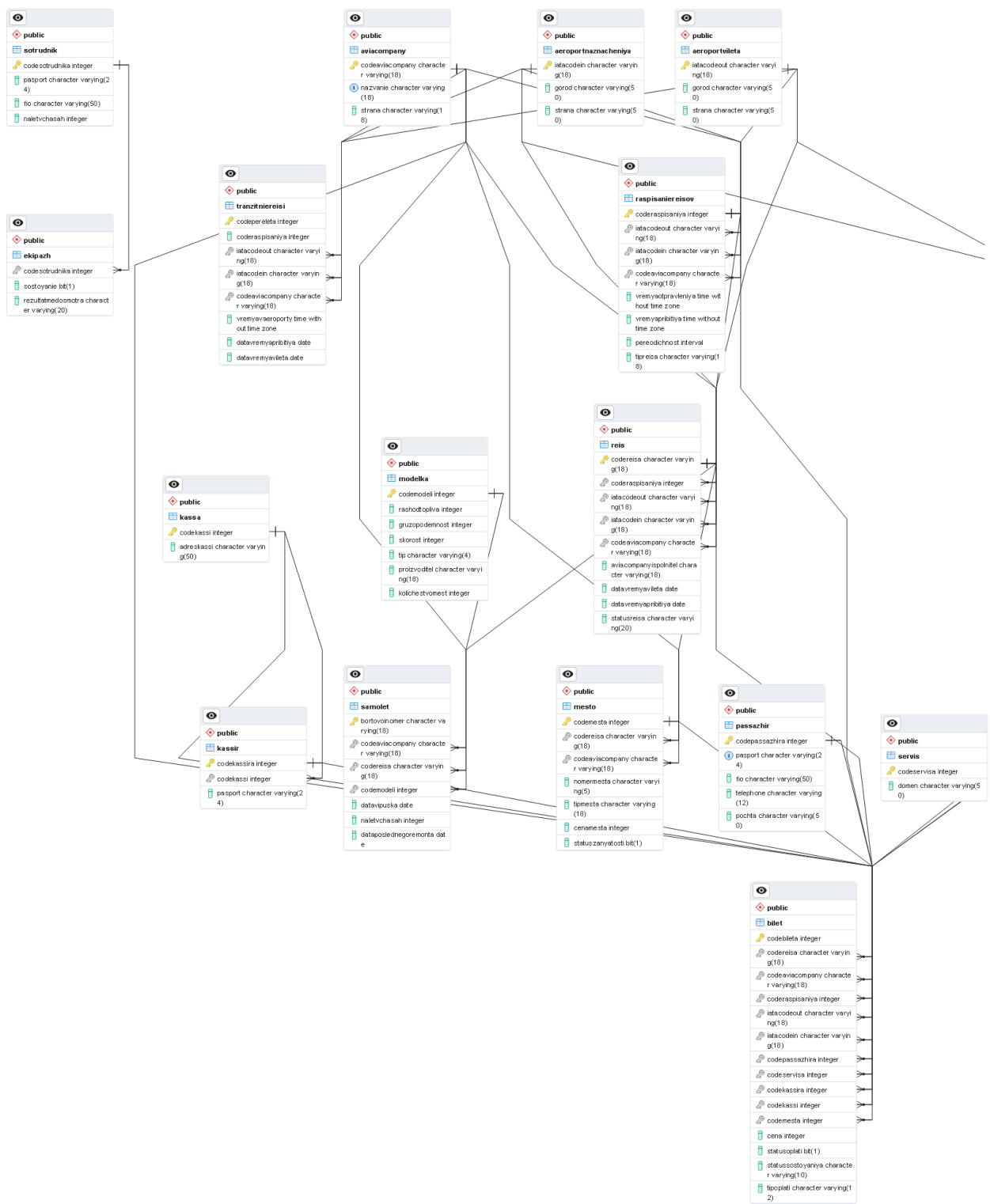


Схема логической модели БД.

Вывод: В рамках данной лабораторной работы были получены практические навыки работы с базой данных PostgreSQL 1X. В ходе выполнения работы были выполнены следующие задачи: Создание таблиц базы данных PostgreSQL 1X. Для этого были изучены основные типы данных, которые поддерживает PostgreSQL 1X, а также способы создания таблиц и определения их структуры. Были созданы несколько таблиц с различными полями, которые соответствуют различным типам данных. Заполнение таблиц рабочими данными. Для этого были изучены способы добавления данных в таблицы, а также правила заполнения таблиц. Были созданы тестовые данные для заполнения таблиц. Резервное копирование базы данных. Для этого были изучены способы создания резервных копий базы данных, а также хранения их на диске. Были созданы резервные копии базы данных для дальнейшего восстановления. Восстановление базы данных из резервной копии. Для этого были изучены способы восстановления базы данных из резервной копии, а также правила восстановления данных. Была проведена процедура восстановления базы данных из резервной копии. В результате выполнения лабораторной работы были получены необходимые навыки для создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД. Эти навыки могут быть полезны для решения задач в области хранения и обработки данных в современных информационных системах.

#### **Список использованных источников:**

1. Документация PostgreSQL [Электронный ресурс] // Официальный сайт PostgreSQL. 1996-2023. URL: <https://www.postgresql.org/docs/13/index.html> (дата обращения: 11.02.2023).
2. Документация pgAdmin 4 PostgreSQL [Электронный ресурс] // Официальный сайт pgAdmin. URL: <https://www.pgadmin.org/docs/pgadmin4/latest/> (дата обращения: 11.02.2023)



