

Министерство науки и высшего образования Российской  
Федерации Федеральное государственное автономное образовательное  
учреждение высшего

образования  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИТМО»

Факультет инфокоммуникационных технологий

ОТЧЕТ  
**О ЛАБОРАТОРНОЙ РАБОТЕ № 3**

**по теме: СОЗДАНИЕ БАЗЫ ДАННЫХ POSTGRESQL. ЗАПОЛНЕНИЕ ТАБЛИЦ БД  
РАБОЧИМИ ДАННЫМИ.**

по дисциплине: Проектирование и реализация баз данных

Специальность:  
09.03.03 Мобильные и сетевые технологии

Проверил:  
Говорова М.М. \_\_\_\_\_  
Дата: «\_\_» \_\_\_\_\_ 20\_\_ г.  
Оценка \_\_\_\_\_

Выполнила:  
студент группы К3241  
Каратецкая Мария

Санкт-Петербург 2020/2021

## ЦЕЛЬ РАБОТЫ

Создание таблиц базы данных PostgreSQL 1X, заполнение их рабочими данными, осуществление 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.

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.

# ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Вариант 8, БД «Аэропорт»

Выполнение

## 1. Название БД

Courses

## 2. Схема инфологической модели данных БД

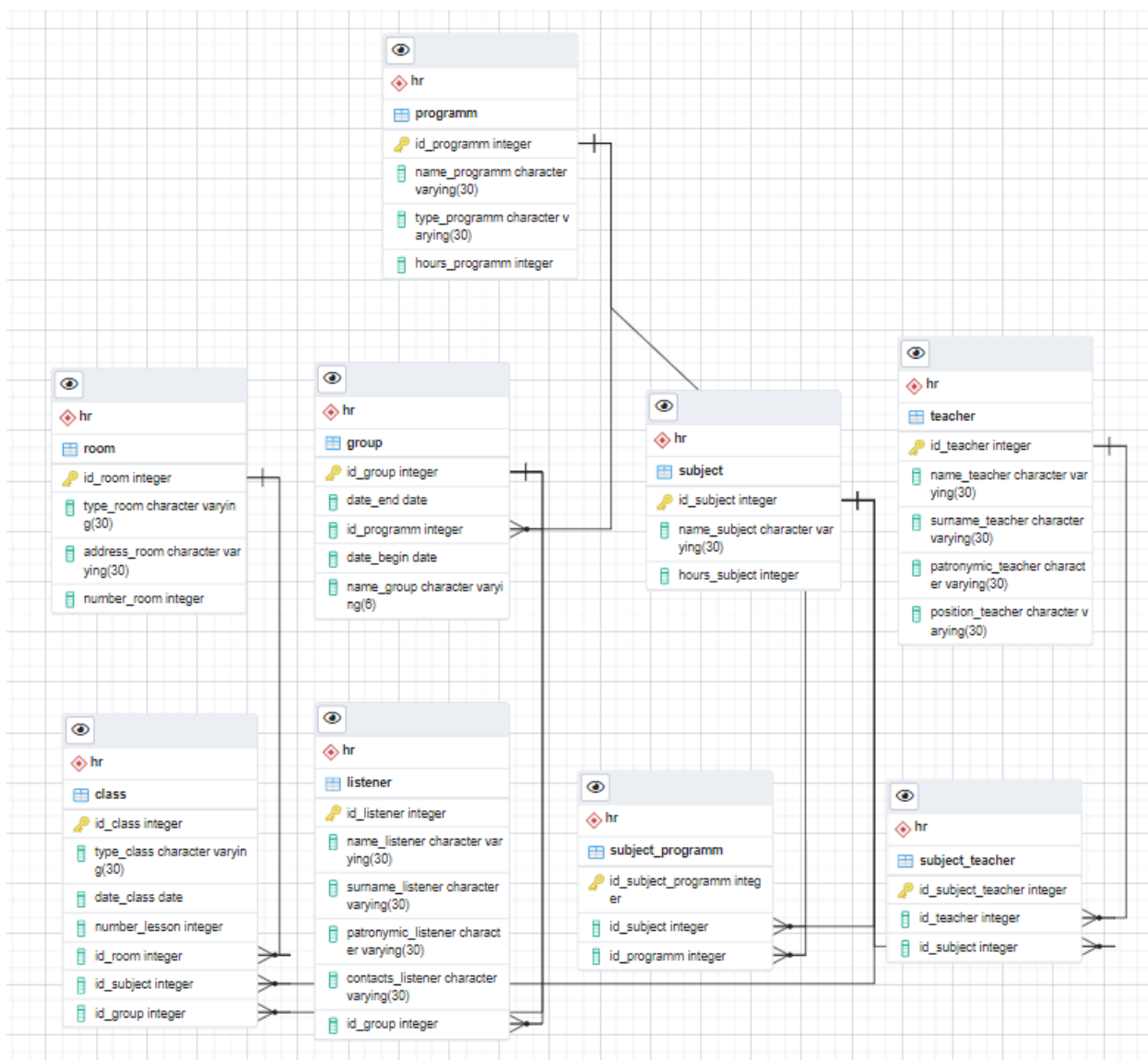


Рисунок 1 – Схема инфологической модели БД, сгенерированная в Generate ERD

## 3. Plain dump

1) Создание базы данных (схемы в бд):

```
CREATE SCHEMA hr;
```

```
ALTER SCHEMA hr OWNER TO postgres;
```

#### Создание таблицы проведение занятий

```
CREATE TABLE hr.class
```

```
(
```

```
    id_class integer NOT NULL,
```

```
    type_class character varying(30) NOT NULL,
```

```
    date_class date NOT NULL,
```

```
    number_lesson integer NOT NULL,
```

```
    id_room integer NOT NULL,
```

```
    id_subject integer NOT NULL,
```

```
    id_group integer NOT NULL,
```

```
    CONSTRAINT class_pkey PRIMARY KEY (id_class),
```

```
    CONSTRAINT id_group FOREIGN KEY (id_group) REFERENCES hr.group(group_id) NOT VALID,
```

```
    CONSTRAINT id_room FOREIGN KEY (id_room) REFERENCES hr.room(room_id) NOT VALID,
```

```
    CONSTRAINT id_subject FOREIGN KEY (id_subject) REFERENCES hr.subject(subject_id) NOT VALID,
```

```
    CONSTRAINT type_class CHECK (po) NOT VALID
```

```
)
```

```
ALTER TABLE hr.class
```

```
    OWNER to postgres;
```

```
/* таблицы проведение занятий*/
```

#### Создание таблицы Дисциплина/учитель

```
CREATE TABLE hr.subject_teacher
```

```
(
```

```
    id_subject_teacher integer NOT NULL,
```

```
    id_teacher integer NOT NULL,
```

```
    id_subject integer NOT NULL,
```

```
    CONSTRAINT class_teacher_pkey PRIMARY KEY (id_subject_teacher),
```

```
    CONSTRAINT id_subject FOREIGN KEY (id_subject) REFERENCES hr.subject (id_subject) NOT VALID,
```

```
    CONSTRAINT id_teacher FOREIGN KEY (id_teacher) REFERENCES hr.teacher (id_teacher) NOT VALID
```

```
ALTER TABLE hr.subject_teacher
```

```
    OWNER to postgres;
```

```
/*Создание таблицы Дисциплина/учитель*/
```

#### Создание таблицы группа

```
CREATE TABLE hr."group"  
(  
    id_group integer NOT NULL,  
    date_end date NOT NULL,  
    id_programm integer NOT NULL,  
    date_begin date NOT NULL,  
    name_group character varying(6) NOT NULL,  
    CONSTRAINT group_pkey PRIMARY KEY (id_group),  
    CONSTRAINT id_programm FOREIGN KEY (id_programm) REFERENCES hr.programm (id_programm)  
    NOT VALID  
    CONSTRAINT date_begin CHECK (date_begin > '1900-01-01'::date) NOT VALID,  
    CONSTRAINT date_end CHECK (date_end > date_begin) NOT VALID  
)  
ALTER TABLE hr."group"  
    OWNER to postgres;
```

/\*Создание таблицы группа\*/

#### создание таблицы слушатели

```
CREATE TABLE hr.listener  
(  
    id_listener integer NOT NULL,  
    name_listener character varying(30) NOT NULL,  
    surname_listener character varying(30) NOT NULL,  
    patronymic_listener character varying(30) NOT NULL,  
    contacts_listener character varying(30) NOT NULL,  
    id_group integer NOT NULL,  
    CONSTRAINT listener_pkey PRIMARY KEY (id_listener),  
    CONSTRAINT id_group FOREIGN KEY (id_group) REFERENCES hr."group" (id_group) NOT VALID  
)  
ALTER TABLE hr.listener  
    OWNER to postgres;
```

/\*создание таблицы слушатели\*/

#### Создание таблицы программа

```
CREATE TABLE hr.programm
(
    id_programm integer NOT NULL,
    name_programm character varying(30) NOT NULL,
    type_programm character varying(30) NOT NULL,
    hours_programm integer NOT NULL,
    CONSTRAINT programm_pkey PRIMARY KEY (id_programm)
)
ALTER TABLE hr.programm
    OWNER to postgres;
/* создание таблицы программа*/
```

#### Создание таблицы аудитория

```
CREATE TABLE hr.room
(
    id_room integer NOT NULL,
    type_room character varying(30) NOT NULL,
    address_room character varying(30) NOT NULL,
    number_room integer NOT NULL,
    CONSTRAINT room_pkey PRIMARY KEY (id_room)
)
ALTER TABLE hr.room
    OWNER to postgres;
/*Создание таблицы аудитория*/
```

#### создание таблицы дисциплина

```
CREATE TABLE hr.subject
(
    id_subject integer NOT NULL,
    name_subject character varying(30) NOT NULL,
    hours_subject integer NOT NULL,
```

```

        CONSTRAINT subject_pkey PRIMARY KEY (id_subject)
ONSTRAINT hours_subject CHECK (hours_subject > 0) NOT VALID
)
ALTER TABLE hr.subject
    OWNER to postgres;
/* таблицы дисциплина*/

```

#### Создание таблицы дисциплина/программа

```

CREATE TABLE hr.subject_programm
(
    id_subject_programm integer NOT NULL,
    id_subject integer NOT NULL,
    id_programm integer NOT NULL,
    CONSTRAINT subject_programm_pkey PRIMARY KEY (id_subject_programm),
    CONSTRAINT id_programm FOREIGN KEY (id_programm) REFERENCES hr.programm (id_programm)
    NOT VALID,
    CONSTRAINT id_subject FOREIGN KEY (id_subject) REFERENCES hr.subject (id_subject) NOT VALID
    CONSTRAINT hours_programm CHECK (hours_programm > 0) NOT VALID
)
ALTER TABLE hr.subject_programm
    OWNER to postgres;
/*Создание таблицы дисциплина/программа*/

```

#### Создание таблицы учитель

```

CREATE TABLE hr.teacher
(
    id_teacher integer NOT NULL,
    name_teacher character varying(30) NOT NULL,
    surname_teacher character varying(30) NOT NULL,
    patronymic_teacher character varying(30) NOT NULL,
    position_teacher character varying(30) NOT NULL,
    CONSTRAINT teacher_pkey PRIMARY KEY (id_teacher),
    CONSTRAINT position_teacher CHECK (position_teacher::text = ANY (ARRAY['professor'::character
    varying::text, 'lecturer'::character varying::text, 'assistant professor'::character varying::text, 'senior
    lecturer'::character varying::text])) NOT VALID

```



)

```
ALTER TABLE hr.teacher
```

```
OWNER to postgres;
```

```
/*Создание таблицы учитель*/
```

#### Заполнение таблицы Учитель

```
INSERT INTO hr."teacher"(name_teacher, surname_teacher, patronymic_teacher, position_teacher)
```

```
VALUES
```

```
('Marina', 'Tyakova', 'Ivanovna', 'professor'),
```

```
('Anna', 'Liskina', 'Petrovna', 'assistant professor'),
```

```
('Alex', 'Zanin', 'Sergeivich', 'senior lecturer'),
```

```
('Vadim', 'Bobov', 'Ivanovich', 'lecturer');
```

#### Заполнение таблицы Программа

```
INSERT INTO hr."programm"(name_programm, type_programm, hours_programm)
```

```
VALUES
```

```
('frontend', 'programming', 180),
```

```
('backend', 'programming', 220),
```

```
('web design', 'design', 160),
```

```
('room design', 'design', 160),
```

```
('civil law', 'law', 240),
```

```
('criminal law', 'law', 240);
```

#### Заполнение таблицы Группа

```
INSERT INTO hr."group"(date_begin, date_end, name_group, id_programm)
```

```
VALUES
```

```
('01/10/2018', '01/10/2022', 'K3243', 1),
```

```
('10/04/2019', '10/04/2021', 'K3247', 3),
```

```
('01/04/2020', '01/10/2024', 'K3248', 2),
```

```
('01/08/2018', '01/08/2022', 'K3240', 4),
```

```
('15/10/2020', '15/10/2024', 'K3249', 5);
```

#### Заполнение таблицы Комната

```
INSERT INTO hr."room"(type_room, address_room, number_room)
```

VALUES

```
('computer', 'Lomonosov 3', 1459),  
('lecture', 'Birzhevaya 14', 222),  
('lecture', 'Lomonosov 3', 1259),  
('computer', 'Belorusskaya 6', 323),  
('assembly hall', 'Birzhevaya 14', 979),  
('lecture', 'Lomonosov 3', 1213);
```

#### Заполнение таблицы Дисциплина

```
INSERT INTO hr."subject"(name_subject, hours_subject)
```

VALUES

```
('web design', 40),  
('android developer', 60),  
('ios developer', 60),  
('painter', 40),  
('constitution Russia', 60),  
('law', 40);
```

#### Заполнение таблицы Дисциплина/программа

```
INSERT INTO hr."subject_programm"(id_subject, id_programm)
```

VALUES

```
(1, 3),  
(2,3),  
(3,1),  
(4,3),  
(4,4),  
(5,4);
```

#### Заполнение таблицы Слушатель

```
INSERT INTO hr."listener"(name_listener, surname_listener, patronymic_listener, contacts_listener,  
id_group)
```

VALUES

```
('Anna', 'Koleva', 'Sergeyvna', '+7920-200-15-45', 1),  
('Alex', 'Boneva', 'Alexandrovna', '+7999-872-23-41', 3),
```

```
('Sergey', 'Unin', 'Sergeivich', '+7888-251-29-33', 5),  
('Vladimir', 'Hogov', 'Ivanovich', '+7937-214-14-39', 4);
```

Заполнение таблицы Проведение занятий

```
INSERT INTO hr."class"(type_class, date_class, number_lesson, id_room, id_subject, id_group)  
VALUES  
('practice', '01/10/2021', 1, 4, 2, 1),  
('lecture', '02/10/2021', 2, 5, 5, 5),  
('lecture', '02/10/2021', 1, 6, 3, 2),  
('practice', '03/10/2021', 1, 5, 4, 2);
```

Заполнение таблицы Дисциплина/Учитель

```
INSERT INTO hr."class_teacher"(id_teacher, id_teacher)  
VALUES  
(2, 3),  
(1, 4),  
(3, 1),  
(1, 1),  
(4, 3);
```

## **Вывод**

В ходе выполнения лабораторной работы были созданы таблицы базы данных PostgreSQL 1X. Были установлены ограничения на данные (первичный и внешний ключи, проверки на наличие и корректность значения). Затем таблицы были заполнены рабочими данными. Были созданы две резервные копии (в текстовом и кастомном вариантах), первая использовалась для листинга в отчете, а с помощью второй было произведено восстановление базы данных.