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 таблиц базы данных Postgresql

**по дисциплине:** Проектирование и реализация баз данных Специальность: 09.03.03 Мобильные и сетевые технологии

Проверил: Говорова М.М.

Дата: «27» апреля 2021г. Оценка

Выполнил: студент группы К3240

Балакин Денис

Санкт-Петербург 2021 г

# Цель работы

Овладение практическими навыками реализации базы данных с использованием PostgreSQL

# Практическое задание

1. На основе созданной инфологической модели данных создать базу данных с помощью pgAdmin 4;
2. Создать таблицы базы данных PostgreSQL с соответственными ограничениями полей PRIMARY KEY, FOREIGN KEY, CHECK, UNIQUE;
3. Заполнить таблицы данными.

# Ход работы

Вариант 13. БД «Отели»

Описание предметной области: Отели находятся в разных городах. Цены на номера одного типа во всех отелях одинаковы и зависят от типа номера и количества мест. Номер может быть забронирован, занят или свободен. При заезде в отель постояльцы проходят регистрацию. Информация о регистрации постояльцев отеля (выехавших из отеля) хранится в течение года и 1 января удаляется в архив.

БД должна содержать следующий минимальный набор сведений: Адрес отеля. Название отеля. Номер комнаты. Тип комнаты. Количество мест. Цена комнаты за сутки проживания. Имя постояльца. Фамилия постояльца. Отчество постояльца. Адрес постоянного проживания. Дата заезда. Дата отъезда.

Дополнить исходные данные информацией: по бронированию комнаты; по сотруднику, который регистрирует постояльца в отеле в день заезда; по оплате проживания; по составу удобств в комнате; по акциям, доступным при бронировании (скидки).

1. **Название создаваемой БД:** Hotels

# Cхема инфологической модели

Схема инфологической модели данных, которая сгенерирована в pgAdmin, представлена в Приложении 1.

# Скрипты работы с базой данных Hotels

* 1. Создание таблиц данных

Для реализации базы данных требуется создать 7 таблиц. При создании каждой таблицы были введены соответствующие ограничения на строковые типы данных, код создания находится в приложении 1

Коды вставки (INSERT) данных и их выборки (SELECT) и запросы приведены в Приложении 3.

# Приложение 1. Инфологическая модель данных



Рисунок 1 – Инфологическая модель данных

# Приложение 2. Создание баз данных

**Фрагмент 1 – Создание таблицы Room**

CREATE TABLE hotel\_sch."Room"

(

status character varying(20) COLLATE pg\_catalog."default" NOT NULL,

"number" integer NOT NULL,

type character varying(20) COLLATE pg\_catalog."default" NOT NULL,

facilities character varying(50) COLLATE pg\_catalog."default" NOT NULL,

id\_room integer NOT NULL,

id\_prom integer NOT NULL,

id\_hotel integer NOT NULL,

CONSTRAINT "Room\_pkey" PRIMARY KEY (id\_room),

CONSTRAINT id\_hotel\_fk FOREIGN KEY (id\_hotel)

REFERENCES hotel\_sch."Hotel" (id\_hotel) MATCH SIMPLE

ON UPDATE RESTRICT

ON DELETE RESTRICT,

CONSTRAINT id\_prom\_fk FOREIGN KEY (id\_prom)

REFERENCES hotel\_sch."Promotions" (id\_prom) MATCH SIMPLE

ON UPDATE RESTRICT

ON DELETE RESTRICT,

CONSTRAINT room\_check CHECK (number > 225)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE hotel\_sch."Room"

OWNER to postgres;

-- Index: fki\_id\_hotel\_fk

-- DROP INDEX hotel\_sch.fki\_id\_hotel\_fk;

CREATE INDEX fki\_id\_hotel\_fk

ON hotel\_sch."Room" USING btree

(id\_hotel ASC NULLS LAST)

TABLESPACE pg\_default;

-- Index: fki\_id\_prom\_fk

-- DROP INDEX hotel\_sch.fki\_id\_prom\_fk;

CREATE INDEX fki\_id\_prom\_fk

ON hotel\_sch."Room" USING btree

(id\_prom ASC NULLS LAST)

TABLESPACE pg\_default;

# Фрагмент 2 – Создание таблицы Hotel

CREATE TABLE hotel\_sch."Hotel"

(

id\_hotel integer NOT NULL,

type character varying(25) COLLATE pg\_catalog."default" NOT NULL,

adress character varying(225) COLLATE pg\_catalog."default" NOT NULL,

col\_star character varying(5) COLLATE pg\_catalog."default" NOT NULL,

id\_emp integer,

CONSTRAINT "Hotel\_pkey" PRIMARY KEY (id\_hotel),

CONSTRAINT id\_empt\_fk FOREIGN KEY (id\_emp)

REFERENCES hotel\_sch."Employee" (id\_emp) MATCH SIMPLE

ON UPDATE RESTRICT

ON DELETE RESTRICT

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE hotel\_sch."Hotel"

OWNER to postgres;

# Фрагмент 3 – Создание таблицы Lodger

CREATE TABLE hotel\_sch."Lodger"

(

id\_log integer NOT NULL,

cons\_adr character varying(255) COLLATE pg\_catalog."default" NOT NULL,

first\_name character varying(255) COLLATE pg\_catalog."default" NOT NULL,

second\_name character varying(25) COLLATE pg\_catalog."default" NOT NULL,

last\_name character varying(25) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Lodger\_pkey" PRIMARY KEY (id\_log)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE hotel\_sch."Lodger"

# OWNER to postgres;

# Фрагмент 4 – Создание таблицы Promotions

CREATE TABLE hotel\_sch."Promotions"

(

id\_prom integer NOT NULL,

price integer,

date\_end date NOT NULL,

date\_start date NOT NULL,

id\_type integer NOT NULL,

CONSTRAINT promotions\_pkey PRIMARY KEY (id\_prom),

CONSTRAINT id\_type\_fk FOREIGN KEY (id\_type)

REFERENCES hotel\_sch."Type" (id\_type) MATCH SIMPLE

ON UPDATE RESTRICT

ON DELETE RESTRICT

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE hotel\_sch."Promotions"

OWNER to postgres;

-- Index: fki\_id\_type\_fk

-- DROP INDEX hotel\_sch.fki\_id\_type\_fk;

CREATE INDEX fki\_id\_type\_fk

ON hotel\_sch."Promotions" USING btree

(id\_type ASC NULLS LAST)

TABLESPACE pg\_default;

# Фрагмент 5 – Создание таблицы Type

CREATE TABLE hotel\_sch."Type"

(

id\_type integer NOT NULL,

col\_room integer NOT NULL,

class character varying(25) COLLATE pg\_catalog."default" NOT NULL,

price\_night real NOT NULL,

facilities character varying(25) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Type\_pkey" PRIMARY KEY (id\_type),

CONSTRAINT type\_check\_id CHECK (id\_type > 0)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE hotel\_sch."Type"

OWNER to postgres;

# Фрагмент 6 – Создание таблицы Check\_in

CREATE TABLE hotel\_sch."Check\_in"

(

id\_check integer NOT NULL,

data\_begin date NOT NULL,

date\_in date NOT NULL,

date\_finish date NOT NULL,

date\_out date NOT NULL,

status character varying(20) COLLATE pg\_catalog."default" NOT NULL,

id\_emp integer NOT NULL,

id\_log integer NOT NULL,

id\_room integer,

CONSTRAINT "Check\_in\_pkey" PRIMARY KEY (id\_check),

CONSTRAINT id\_emp FOREIGN KEY (id\_check)

REFERENCES hotel\_sch."Check\_in" (id\_check) MATCH SIMPLE

ON UPDATE RESTRICT

ON DELETE RESTRICT,

CONSTRAINT id\_log\_fk FOREIGN KEY (id\_log)

REFERENCES hotel\_sch."Lodger" (id\_log) MATCH SIMPLE

ON UPDATE RESTRICT

ON DELETE RESTRICT,

CONSTRAINT id\_room\_fk FOREIGN KEY (id\_room)

REFERENCES hotel\_sch."Room" (id\_room) MATCH SIMPLE

ON UPDATE RESTRICT

ON DELETE RESTRICT,

CONSTRAINT check\_in\_data CHECK (date\_in > date\_out)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE hotel\_sch."Check\_in"

OWNER to postgres;

# Фрагмент 7 – Создание таблицы Eployee

CREATE TABLE hotel\_sch."Employee"

(

id\_emp integer NOT NULL,

firs\_name character varying(25) COLLATE pg\_catalog."default" NOT NULL,

second\_name character varying(25) COLLATE pg\_catalog."default" NOT NULL,

last\_name character varying(25) COLLATE pg\_catalog."default" NOT NULL,

place character varying(225) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Employee\_pkey" PRIMARY KEY (id\_emp)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE hotel\_sch."Employee"

OWNER to postgres;

# Приложение 3. Работа с данными Фрагмент 1 – Вставка данных о Promotions

INSERT INTO hotel\_sch."Promotions"(

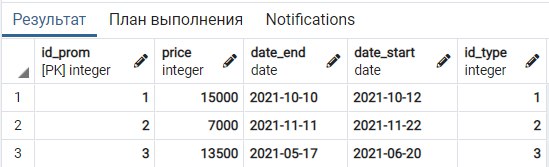
id\_prom, price, date\_end, date\_start, id\_type)

VALUES (1, 15000, '2021-10-10', '2021-10-12,1),

(2, 7000, '2021-11-11', '2021-11-22,2),

(3, 13500, '2021-05-17', '2021-06-20,3);

# Рисунок 1 – Результат запроса SELECT \* FROM hotels.”Promotions”

****

**Фрагмент 2 – Вставка данных о Room**

INSERT INTO hotel\_sch."Room"(

status, "number", type, facilities, id\_room, id\_prom, id\_hotel)

VALUES (1, 344, ‘one-room’, 234, 2, 3252, 53525);

**Фрагмент 3 – Вставка данных о Type**

INSERT INTO hotel\_sch."Type"(

id\_type, col\_room, class, price\_night, facilities)

VALUES (23, 3, ‘one-room’, ‘rare’, 0);

**Фрагмент 4 – Вставка данных о Lodger**

INSERT INTO hotel\_sch."Lodger"(

id\_log, cons\_adr, first\_name, second\_name, last\_name)

VALUES (53,’1я Утиная’, ‘Алексей’, ‘Новиков’, ‘Игоревич’);

**Фрагмент 5 – Вставка данных о Hotel**

INSERT INTO hotel\_sch."Hotel"(

id\_hotel, type, adress, col\_star, id\_emp)

VALUES (13, ‘casual’, ’1я Утиная’ , 4, 24114);

**Фрагмент 6 – Вставка данных о Employee**

INSERT INTO hotel\_sch."Employee"(

id\_emp, firs\_name, second\_name, last\_name, place)

VALUES (112, ‘Оля’, ‘Неманар’, ‘Олеговна’, ’1я Утиная’);

**Фрагмент 7 – Вставка данных о Check\_in**

INSERT INTO hotel\_sch."Check\_in"(

id\_check, data\_begin, date\_in, date\_finish, date\_out, status, id\_emp, id\_log, id\_room)

VALUES (664, 2021-12-05, 2021-12-10, 2021-12-05, 2021-12-10, ‘free’, 234, 5343, 34);

Выводы:

В ходе данной лабораторной работы была создана база данных в PostgreSQL, внесены столбцы, ограничения на столбцы, рабочие данные, и логическая модель бд и Dump.