

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**

### **2. Схема инфологической модели**

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

### **3. Скрипты работы с базой данных Hotels**

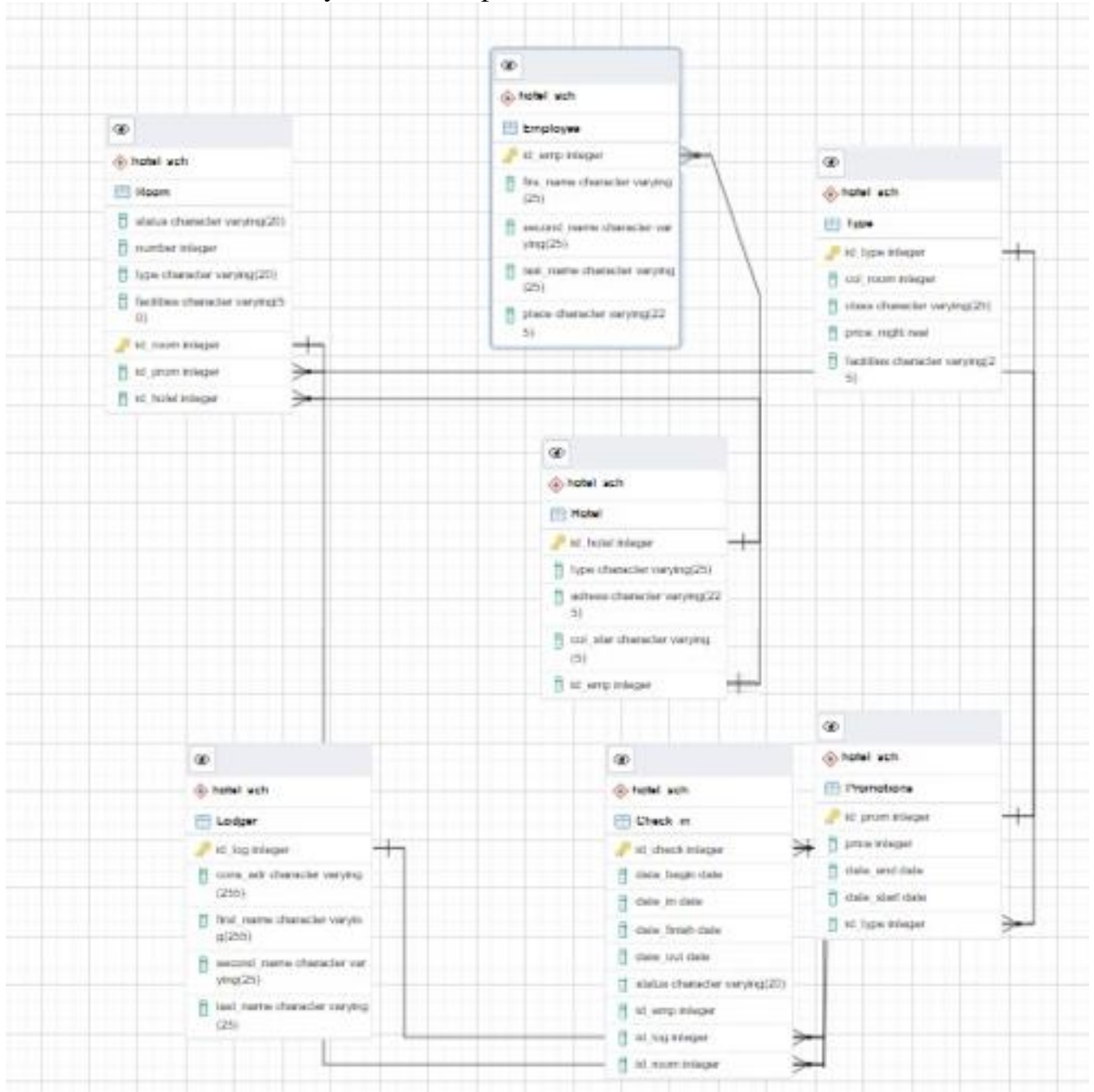
#### **3.1. Создание таблиц данных**

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,
    price 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 > 0)
)
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_emp)
    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 Employee

```
CREATE TABLE hotel_sch."Employee"  
(  
    id_emp integer NOT NULL,  
    first_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,  
    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"

Результат	План выполнения		Notifications			
	id_prom [PK] integer	price integer	date_end date	date_start date	id_type integer	
1	1	15000	2021-10-10	2021-10-12	1	
2	2	7000	2021-11-11	2021-11-22	2	
3	3	13500	2021-05-17	2021-06-20	3	

#### Фрагмент 2 – Вставка данных о Room

```
INSERT INTO hotel_sch."Room"(  
status, "number", type, facilities, id_room, id_prom, id_hotel)  
VALUES (1, 344, 'однокомнатная, 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, 'отель' , ' 2я 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.