

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЕТ

по лабораторной работе №3

на тему

РЕАЛИЗАЦИЯ SQL-ЗАПРОСОВ ДЛЯ СОЗДАНИЯ СХЕМЫ БАЗЫ  
ДАННЫХ

Студент:

Кутняк А. В.

Руководитель:

Игнатович А. О.

Минск 2024

## **1 ЦЕЛЬ**

Цель лабораторной работы – реализация схемы базы данных по ранее построенной реляционной схеме данных (см. лабораторную работу №2). Требуется сформировать SQL-запросы для создания таблиц базы данных и выполнить их в СУБД. Требуется заполнить таблицы данными с помощью оператора INSERT.

## **2 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ**

1. Создать в СУБД новую схему данных для хранения пользовательских объектов;

2. В этой новой схеме данных с помощью скрипта с запросами на языке DDL SQL реализовать таблицы, соответствующие реляционным отношениям схемы данных полученной в лабораторной работе №2, с помощью одного (желательно) оператора CREATE TABLE для каждой таблицы в следующем порядке:

(a) реализовать простую структуру таблиц, включающую только набор столбцов с добавлением описаний первичного ключа;

(b) дополнить описание таблицы реализацией ограничений для описания внешних ключей; для внешних ключей установить свойства контроля целостности данных, если это возможно;

(c) дополнить описание таблицы реализацией ограничений для описания бизнес-правил;

(d) дополнить описание таблицы реализацией комментариев для значимых элементов таблицы.

3. Заполнить с помощью SQL-скрипта строками данных для проверки правильного выбора первичных ключей и работоспособности ссылок между таблицами:

(a) строками данных сначала заполнять мастер-таблицы;

(b) в каждую таблицу добавить 5 – 10 строк осмысленных данных;

(c) если не удастся добавить данные в таблицу по причине нарушения уникальности первичного ключа, то следует перепроверить описание этого первичного ключа и его смысл для реального мира;

(d) если не удастся добавить данные в таблицу по причине нарушения ссылочной целостности, то следует убедиться, что целевые данные существуют, иначе перепроверить описание внешнего ключа.

4. Рассмотреть простые действия по изменению структуры таблицы (переименование столбца таблицы, добавление и удаление ограничений на столбец таблицы или всю таблицу) и реализовать их с помощью оператора ALTER TABLE.

5. Создать временную таблицу с помощью оператора CREATE TABLE и удалить ее с помощью оператора DROP TABLE.

### 3 ХОД ВЫПОЛНЕНИЯ РАБОТЫ

Во время построения таблиц были произведены некоторые изменения в архитектуре схемы данных, полученной в результате выполнения предыдущей лабораторной работы.

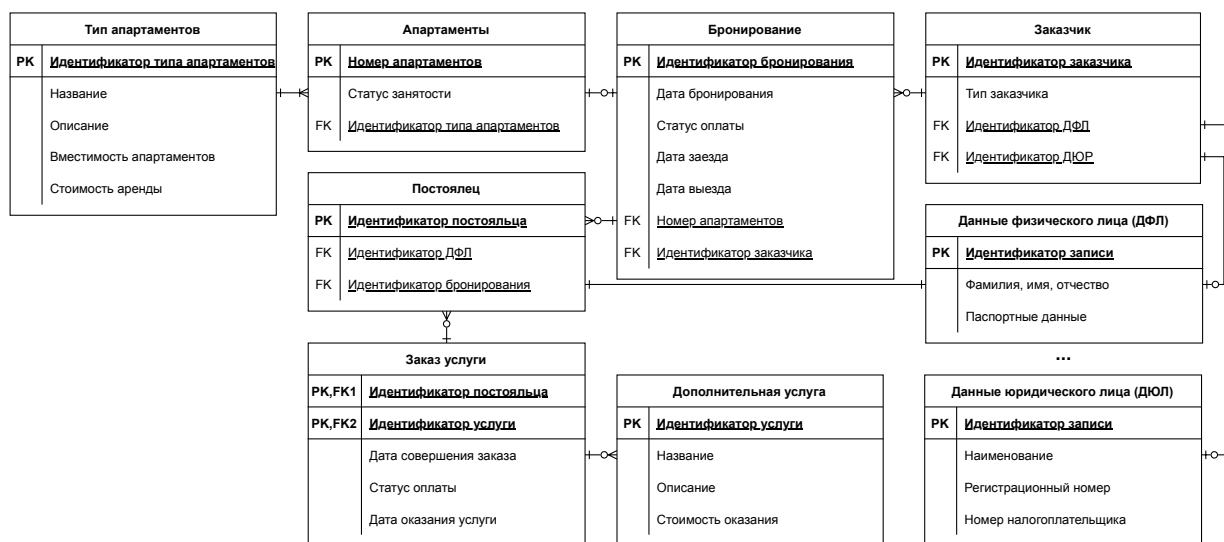


Рисунок 3.1 – Обновленная UML-схема модели данных

Обновленная схема, изображенная на рисунке 3.1, имеет следующие основные изменения:

1. исключены данные, связанные со сведениями об оплате - эта информация относится к платежной системе, а не к гостинице;
2. аутентификация заказчиков и постояльцев происходит через таблицы данных физических и юридических лиц;
3. место проживания гостя определяется опосредованно, через бронирование, что позволяет сформировать легко контролируемый каскад данных «гости - бронирование - заказчик».

Полноразмерную версию обновленной UML-схемы модели данных можно найти в приложении А.

### 3.1 Формирование схемы данных

Ниже представлен листинг скрипта, выполняющего определение базы данных гостиницы.

```
create database hotel;
\c hotel;

create table apartment_type(
    id          serial primary key,
    name        character varying(64) not null,
    description text,
    capacity    integer                not null check (capacity > 0),
    cost        integer                not null check (cost > 0),

    constraint apartment_type_name_and_capacity_unique
        unique (name, capacity)
);

create table service(
    id          serial primary key,
    name        character varying(64) not null unique,
    description text,
    cost        integer                not null check (cost > 0)
);

create table apartment(
    no          integer primary key check (no > 0),
    free        boolean                not null,
    type_id     integer references apartment_type (id) not null
);

create table individual_identity(
    id          serial primary key,
    name        character varying(64) not null,
    passport_id character varying(14) not null unique
);

create table company_identity(
    id          serial primary key,
    name        character varying(64) not null,
    taxpayer_id character varying(9)  not null unique
);
```

```

create table customer(
    id          serial primary key,
    individual_id integer references individual_identity (id) unique,
    company_id   integer references company_identity (id) unique,

    constraint customer_id_type check (
        individual_id is null and company_id is not null
        or
        individual_id is not null and company_id is null)
);

create table apartment_reservation(
    id          serial primary key,
    date        date                        not null,
    paid        boolean                     not null,
    check_in    date                        not null,
    check_out   date                        not null,
    apartment_no integer references apartment (no)      not null,
    customer_id integer references customer (id) on delete cascade not null,

    constraint apartment_reservation_consequence_dates
        check (date <= check_in and check_in < check_out)
);

create table guest(
    id          serial primary key,
    individual_id integer references individual_identity (id)
    reservation_id integer references apartment_reservation (id)
        on delete cascade not null
);

create table service_order(
    guest_id    integer references guest (id) on delete cascade not null,
    service_id  integer references service (id)      not null,

    order_date  date                        not null,
    service_date date                        not null,
    paid        boolean                     not null,

    constraint guest_service_pk primary key (guest_id, service_id),
    constraint service_order_consequence_dates
        check (order_date <= service_date)
);

```

### 3.2 Формирование строк данных

Скрипт заполнения базы данных тестовыми значениями содержит объемные строковые данные и в отчете не приводится; ниже представлены таблицы, полученные в результате выполнения скрипта инициализации данных.

	id	name	description	capacity	cost
1	1	Одноместный од...	Номер повышенной комфортн...	1	70
2	2	Одноместный од...	Номер, предназначенный дл...	1	50
3	3	Двухместный од...	Номер для двух человек. В...	2	95
4	4	Двухместный дв...	Идеальный вариант как для...	2	120
5	5	Двухместный тр...	Идеальный вариант для отд...	2	280
6	6	Трёхместный дв...	Номер, предназначенный дл...	3	115

Рисунок 3.2 – Таблица apartment-type

	id	name	description	cost
1	5	Вызов такси к гостинице	Вызов такси к назначенном...	14
2	4	Экскурсия в ХХХ	Групповая поездка с гидом...	40
3	1	Дополнительный завтрак	Завтрак в договорное время	12
4	3	Услуги прачечной	Стирка, сушка и глажка од...	10
5	2	Дополнительная уборка ном...	Уборка жилого номера в до...	15

Рисунок 3.3 – Таблица service

	id	name	passport_id
1	1	ПУШКИН АЛЕКСАНДР СЕРГЕЕВИЧ	1xxxxxxxxxxxxx
2	2	ТОЛСТОЙ ЛЕВ НИКОЛАЕВИЧ	2xxxxxxxxxxxxx
3	3	ГОГОЛЬ НИКОЛАЙ ВАСИЛЬЕВИЧ	3xxxxxxxxxxxxx
4	4	ЕСЕНИН СЕРГЕЙ АЛЕКСАНДРОВИЧ	4xxxxxxxxxxxxx
5	5	МАЯКОВСКИЙ ВЛАДИМИР ВЛАДИМИРОВИЧ	5xxxxxxxxxxxxx
6	6	ЦВЕТАЕВА МАРИНА ИВАНОВНА	6xxxxxxxxxxxxx
7	7	ФЕТ АФАНАСИЙ АФАНАСЬЕВИЧ	7xxxxxxxxxxxxx

Рисунок 3.4 – Таблица individual-identity

	id	name	taxpayer_id
1	1	000 "СИЛЬВЕР СОЛЮШНЗ"	0xxxxxxxxx

Рисунок 3.5 – Таблица company-identity

	id	individual_id	company_id
1	1	1	<null>
2	2	2	<null>
3	3	4	<null>
4	4	<null>	1

Рисунок 3.6 – Таблица customer

	id	date	paid	check_in	check_out	apartment_no	customer_id
1	1	2024-02-01	• true	2024-02-02	2024-02-09	101	1
2	5	2024-03-01	false	2024-03-08	2024-03-14	101	1
3	2	2024-02-02	• true	2024-02-03	2024-02-06	402	2
4	4	2024-02-12	• true	2024-02-14	2024-02-29	201	3
5	3	2024-02-04	• true	2024-02-09	2024-02-16	603	4

Рисунок 3.7 – Таблица apartment-reservation

	id	individual_id	reservation_id
1	1	1	1
2	2	2	2
3	3	3	2
4	4	4	3
5	5	5	3
6	6	6	3
7	7	7	4

Рисунок 3.8 – Таблица guest

	guest_id	service_id	order_date	service_date	paid
1	1	1	2024-02-08	2024-02-09	• true
2	2	2	2024-02-04	2024-02-04	• true
3	3	3	2024-02-05	2024-02-06	• true
4	4	4	2024-02-23	2024-02-28	false
5	6	5	2024-02-12	2024-02-12	• true

Рисунок 3.9 – Таблица service-order

### 3.3 Диаграмма базы данных

На рисунке 3.10 представлена диаграмма базы данных гостиницы, полученной в результате выполнения скрипта инициализации, представленного в подразделе 3.1.

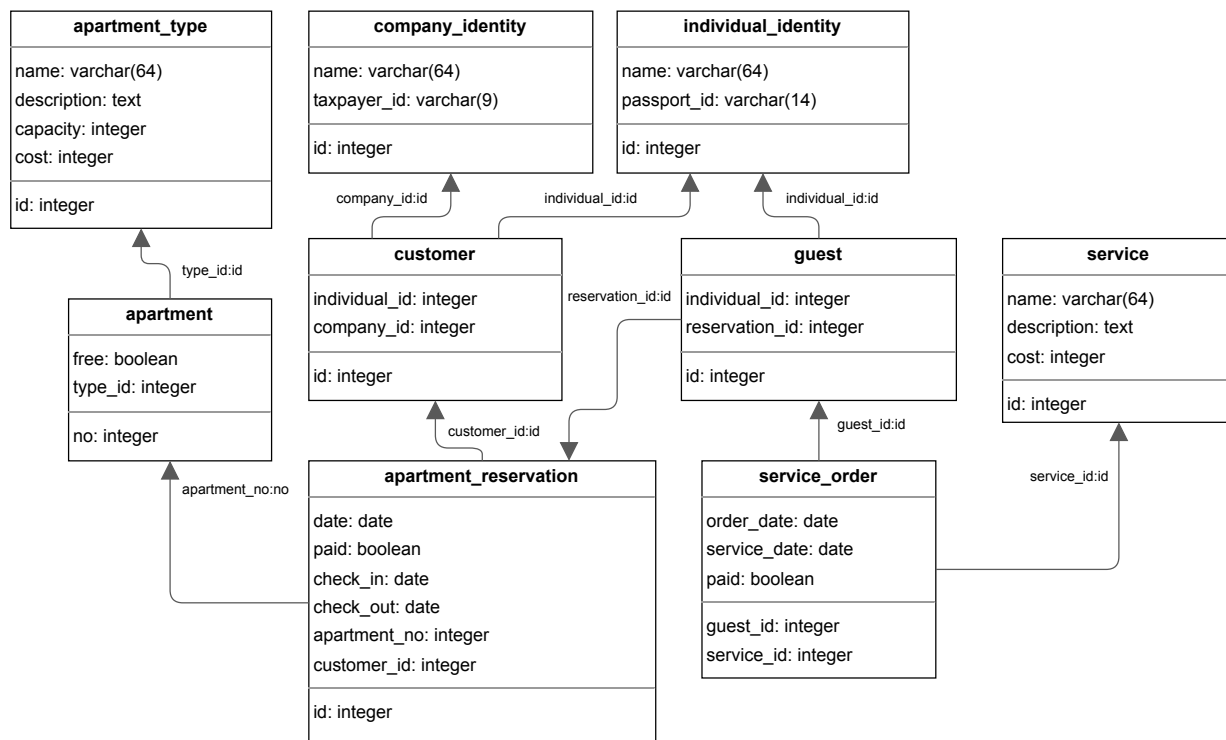


Рисунок 3.10 – Диаграмма базы данных

Полноразмерную версию диаграммы базы данных гостиницы можно найти в приложении А.



### 3.4 Использование команды ALTER

На рисунке 3.11 представлен процесс переименования столбца таблицы при помощи команды ALTER TABLE.

```
hotel=# select * from company_identity;
 id |          name          | taxpayer_id
-----+-----+-----
  1 | 000 "СИЛЬВЕР СОЛЮШНЗ" | 0xxxxxxxxx
(1 row)

hotel=# alter table company_identity rename column name to title;
ALTER TABLE
hotel=# select * from company_identity;
 id |          title          | taxpayer_id
-----+-----+-----
  1 | 000 "СИЛЬВЕР СОЛЮШНЗ" | 0xxxxxxxxx
(1 row)
```

Рисунок 3.11 – Переименование столбца таблицы

На рисунке 3.12 представлен процесс установки и снятия ограничения столбца таблицы при помощи команды ALTER TABLE.

```
hotel=# alter table company_identity add constraint restriction check(name != 'Рога и копыта');
ALTER TABLE
hotel=# insert into company_identity(name, taxpayer_id) values ('Рога и копыта', 'X');
ERROR:  new row for relation "company_identity" violates check constraint "restriction"
DETAIL:  Failing row contains (3, Рога и копыта, X).
hotel=# alter table company_identity drop constraint restriction;
ALTER TABLE
hotel=# insert into company_identity(name, taxpayer_id) values ('Рога и копыта', 'X');
INSERT 0 1
hotel=# select * from company_identity;
 id |          name          | taxpayer_id
-----+-----+-----
  1 | 000 "СИЛЬВЕР СОЛЮШНЗ" | 0xxxxxxxxx
  4 | Рога и копыта          | X
(2 rows)
```

Рисунок 3.12 – Установка и снятие ограничения столбца таблицы

**ПРИЛОЖЕНИЕ А**  
**(обязательное)**  
**Полноразмерные диаграммы**

