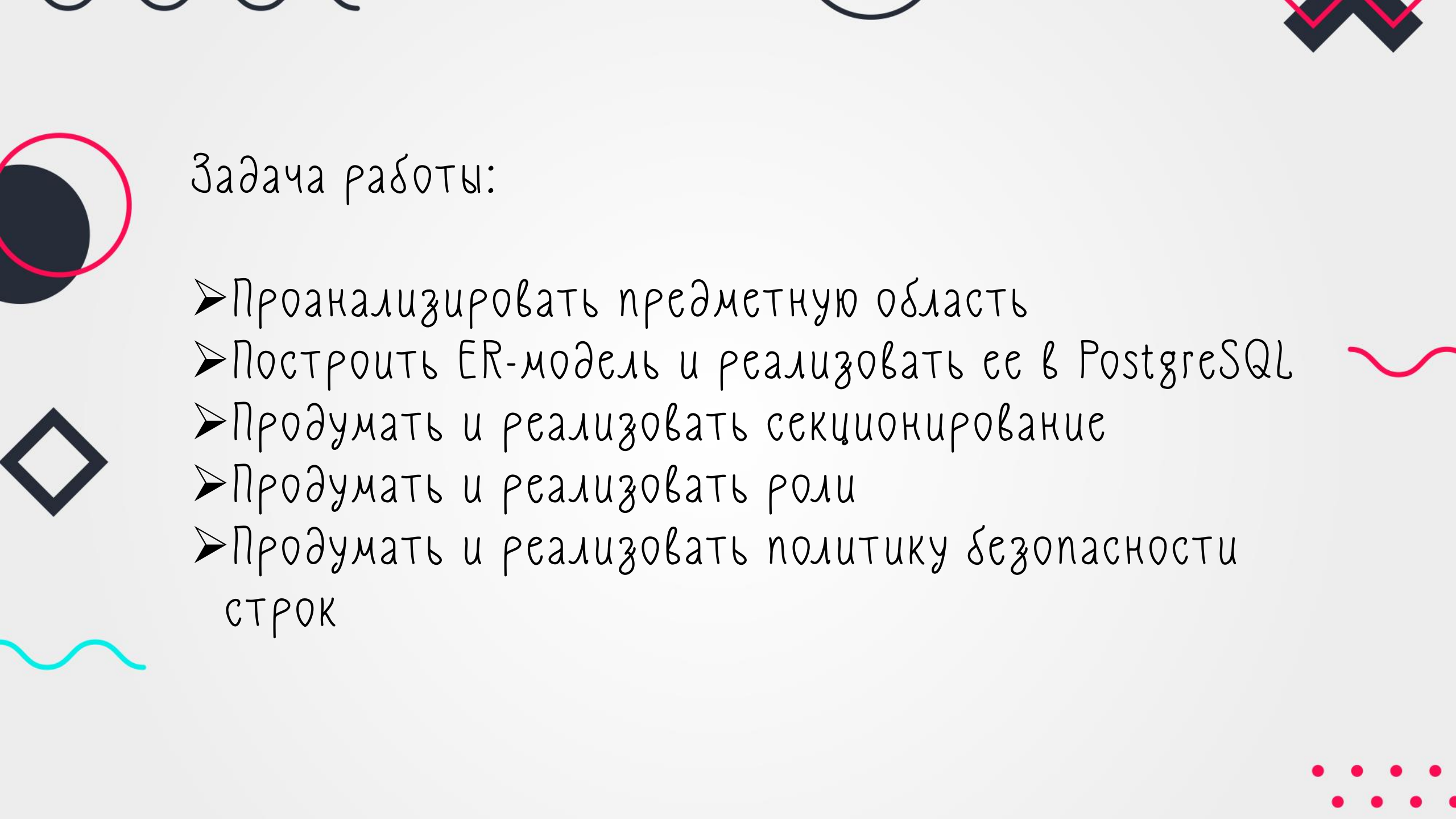


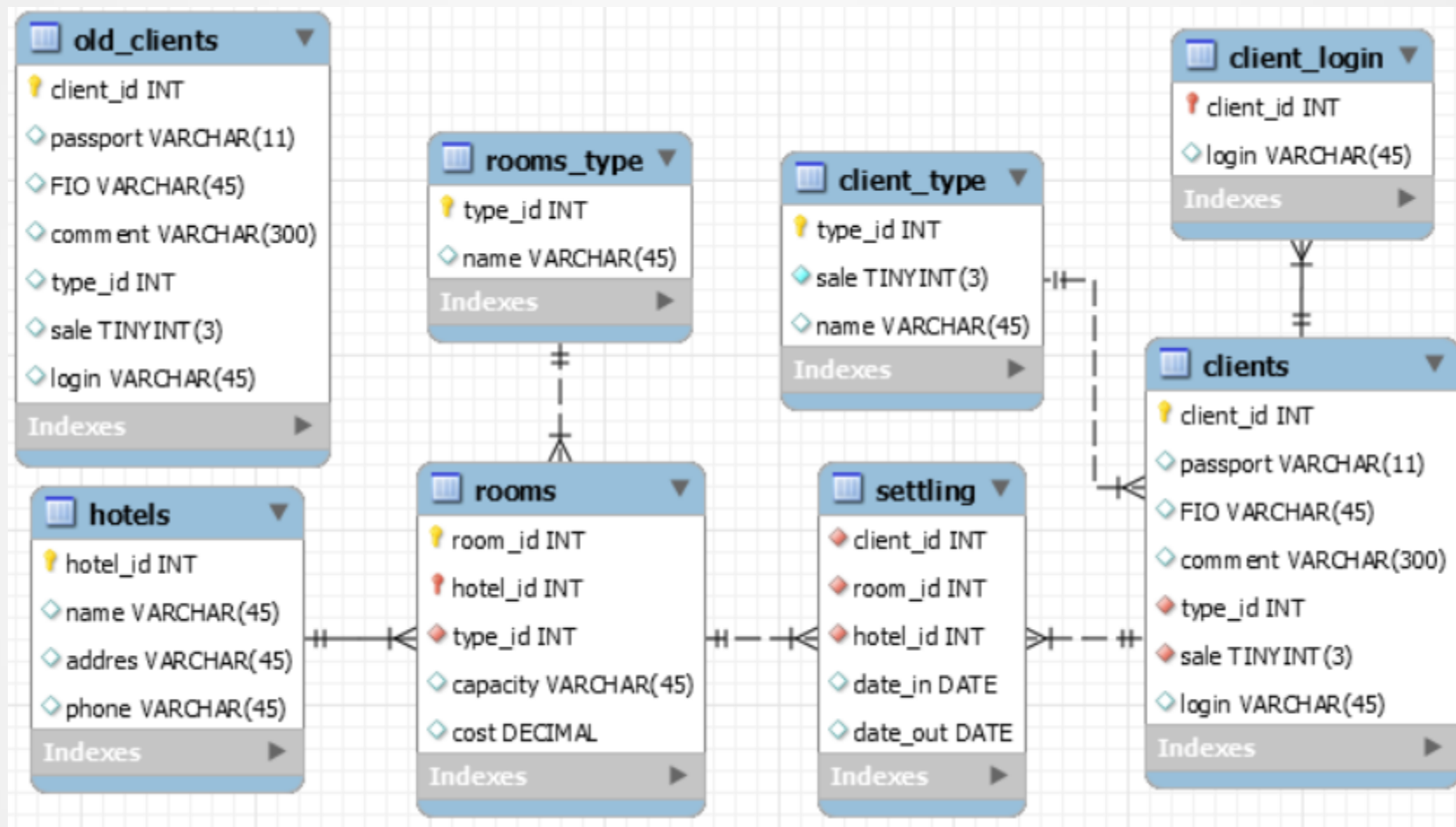
С  
Пол



Задача работы:

- Проанализировать предметную область
- Построить ER-модель и реализовать ее в PostgreSQL
- Продумать и реализовать секционирование
- Продумать и реализовать роли
- Продумать и реализовать политику безопасности строк

# Структура базы данных



# Создание и заполнение таблиц

```
CREATE TABLE IF NOT EXISTS
old_clients (
  client_id INT NOT NULL,
  passport VARCHAR(11) NULL,
  FIO VARCHAR(45) NULL,
  comment VARCHAR(300) NULL,
  type_id INT NULL,
  sale SMALLINT NULL,
  PRIMARY KEY (client_id));
```

```
CREATE TABLE IF NOT EXISTS rooms (
  room_id INT NOT NULL,
  hotel_id INT NOT NULL,
  type_id INT NOT NULL,
  capacity VARCHAR(45) NULL,
  cost DECIMAL NULL,
  PRIMARY KEY (room_id, hotel_id),
  FOREIGN KEY (hotel_id)
  REFERENCES hotels (hotel_id)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  FOREIGN KEY (type_id)
  REFERENCES rooms_type (type_id)
  ON DELETE CASCADE
  ON UPDATE CASCADE);
```

```
CREATE TABLE IF NOT EXISTS
client_type (
  type_id INT NOT NULL,
  sale smallint NOT NULL unique,
  name VARCHAR(45) NULL,
  PRIMARY KEY (type_id));
```

```
CREATE TABLE IF NOT EXISTS hotels (
  hotel_id INT NOT NULL,
  name VARCHAR(45) NULL,
  address VARCHAR(45) NULL,
  phone VARCHAR(45) NULL,
  PRIMARY KEY (hotel_id));
```

```
CREATE TABLE IF NOT EXISTS clients (
  client_id INT NOT NULL,
  passport VARCHAR(11) NULL,
  FIO VARCHAR(45) NULL,
  comment VARCHAR(300) NULL,
  type_id INT NOT NULL,
  sale SMALLINT NOT NULL,
  PRIMARY KEY (client_id),
  FOREIGN KEY (type_id )
  REFERENCES client_type (type_id )
  ON DELETE CASCADE
  ON UPDATE CASCADE);
```

```
CREATE TABLE IF NOT EXISTS
rooms_type (
  type_id INT NOT NULL,
  name VARCHAR(45) NULL,
  PRIMARY KEY (type_id));
```

```
CREATE TABLE IF NOT EXISTS settling (
  settling_id INT NOT NULL,
  client_id INT NOT NULL,
  room_id INT NOT NULL,
  hotel_id INT NOT NULL,
  date_in DATE NULL,
  date_out DATE NULL,
  PRIMARY KEY (settling_id, client_id,
  room_id, hotel_id),
  FOREIGN KEY (client_id)
  REFERENCES clients (client_id)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  FOREIGN KEY (room_id , hotel_id)
  REFERENCES rooms (room_id ,
  hotel_id)
  ON DELETE CASCADE
  ON UPDATE CASCADE)
PARTITION BY RANGE(hotel_id);
```

# Секционирование по отелям

```
CREATE TABLE IF NOT EXISTS settling (  
  settling_id INT NOT NULL,  
  client_id INT NOT NULL,  
  room_id INT NOT NULL,  
  hotel_id INT NOT NULL,  
  date_in DATE NULL,  
  date_out DATE NULL,  
  PRIMARY KEY (settling_id, client_id, room_id, hotel_id),  
  FOREIGN KEY (client_id)  
  REFERENCES clients (client_id)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
  FOREIGN KEY (room_id , hotel_id)  
  REFERENCES rooms (room_id , hotel_id)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE)  
  PARTITION BY RANGE(hotel_id);
```

```
1 CREATE INDEX ON settling_hot1 (hotel_id);  
2 CREATE INDEX ON settling_hot2 (hotel_id);  
3 CREATE INDEX ON settling_hot3 (hotel_id);
```

```
1 CREATE TABLE settling_hot1 PARTITION OF settling FOR VALUES FROM (1) TO (2);  
2 CREATE TABLE settling_hot2 PARTITION OF settling FOR VALUES FROM (2) TO (3);  
3 CREATE TABLE settling_hot3 PARTITION OF settling FOR VALUES FROM (3) TO (4);
```

# Проверка секционирования

```
1 insert into settling values (  
2     1, 1, 1, 1, '16-05-2021', '17-05-2021'),  
3     (2, 2, 2, 1, '4-05-2021', '9-05-2021'),  
4     (3, 3, 1, 2, '11-05-2021', '13-05-2021'),  
5     (4, 4, 2, 2, '1-05-2021', '07-05-2021'),  
6     (5, 5, 1, 3, '13-05-2021', '17-05-2021');
```

	settling_id [PK] integer	client_id [PK] integer	room_id [PK] integer	hotel_id [PK] integer	date_in date	date_out date
1	1	1	1	1	2021-05-16	2021-05-17
2	2	2	2	1	2021-05-04	2021-05-09
3	3	3	1	2	2021-05-11	2021-05-13
4	4	4	2	2	2021-05-01	2021-05-07
5	5	5	1	3	2021-05-13	2021-05-17



# Проверка секционирования










	settling_id [PK] integer	client_id [PK] integer	room_id [PK] integer	hotel_id [PK] integer	date_in date	date_out date
1	1	1	1	1	2021-05-16	2021-05-17
2	2	2	2	1	2021-05-04	2021-05-09

	settling_id [PK] integer	client_id [PK] integer	room_id [PK] integer	hotel_id [PK] integer	date_in date	date_out date
1	3	3	1	2	2021-05-11	2021-05-13
2	4	4	2	2	2021-05-01	2021-05-07

	settling_id [PK] integer	client_id [PK] integer	room_id [PK] integer	hotel_id [PK] integer	date_in date	date_out date
1	5	5	1	3	2021-05-13	2021-05-17

Идеи в будущем

Реализация секционирования по отелям для остальных таблиц также реализуема. Как всегда, не хватило времени на это.

- ▼  settling
  - >  RLS Policies
  - >  Индексы
  - >  Ограничения
  - >  Правила
  - ▼  Секции (3)
    - >  settling\_hot1
    - >  settling\_hot2
    - >  settling\_hot3



# Создание ролей

```
kr_sec_sec=# create role director;  
CREATE ROLE  
kr_sec_sec=# create role admin;  
CREATE ROLE  
kr_sec_sec=# create role porter;  
CREATE ROLE  
kr_sec_sec=# create role receptionist;  
CREATE ROLE  
kr_sec_sec=# create role cashier;  
CREATE ROLE  
kr_sec_sec=# create role concierge;  
CREATE ROLE  
kr_sec_sec=# create role doorman;  
CREATE ROLE
```

Директор  
Администратор  
Портье  
Регистратор  
Кассир  
Консьерж  
Швейцар

```
kr_sec_sec=# create role wardrobe;  
CREATE ROLE  
kr_sec_sec=# create role maid;  
CREATE ROLE  
kr_sec_sec=# create role security;  
CREATE ROLE  
kr_sec_sec=# create role Chef;  
CREATE ROLE  
kr_sec_sec=# create role bartender;  
CREATE ROLE  
kr_sec_sec=# create role sommelier;  
CREATE ROLE  
kr_sec_sec=# create role waiter;  
CREATE ROLE  
kr_sec_sec=# create role guest;  
CREATE ROLE
```

Гардеробщик  
Горничная  
Охрана  
Шеф-повар  
Бармен  
Сомелье  
Официант  
Гость

```
kr_sec_sec=# create role guest1 with login in role guest;  
CREATE ROLE  
kr_sec_sec=# create role guest2 with login in role guest;  
CREATE ROLE  
kr_sec_sec=# create role guest3 with login in role guest;  
CREATE ROLE  
kr_sec_sec=# create role guest4 with login in role guest;  
CREATE ROLE  
kr_sec_sec=# create role guest5 with login in role guest;  
CREATE ROLE
```

Логин гостя включает в себя части  
'guest' + client\_id

# Выдача прав

```
kr_sec_sec=# grant insert, update, delete on client_type, clients,  
hotels, old_clients, rooms, rooms_type, settling to director;  
GRANT
```

```
kr_sec_sec=# grant insert, update, delete on client_type, clients,  
hotels, old_clients, rooms, rooms_type, settling to admin;  
GRANT
```

```
kr_sec_sec=# grant insert, update, delete on clients, old_clients,  
settling to receptionist;  
GRANT
```

```
kr_sec_sec=# grant insert, update, delete on settling, rooms to cashier;  
GRANT
```

```
kr_sec_sec=# grant select on clients, rooms, rooms_type, hotels to guest;  
GRANT
```

## Политика для guest

create policy inf\_client on clients to guest  
using (client\_id in (select client\_id from  
client\_login where login = current\_user));

```
kr_sec_sec=# create policy inf_client on clients to guest using (client_id  
in (select client_id from client_login where login = current_user));  
CREATE POLICY
```

```
kr_sec_sec=# alter table clients enable row level security;  
ALTER TABLE
```

# Проверка политики

```
kr_sec_sec=> \c - postgres
Пароль пользователя postgres:
Вы подключены к базе данных "kr_sec_sec" как пользователь "postgres".
kr_sec_sec=# select * from clients;
```

client_id	passport	fio	comment	type_id	sale
1	7676545467	Жмышенко Н.П.	Мафиозник	2	50
2	9809545467	Хесус	Грустит по гаечке	1	10
3	1239088332	Скукин Д.А.	Мразь, курит, пьет, буянит. Не общителен.	3	2
4	1092380918	Братишкин	Медиагигант. Плохой мальчик	3	2
5	1092198198	Пакет	Майнкрафт это его жизнь	2	50

(5 строк)

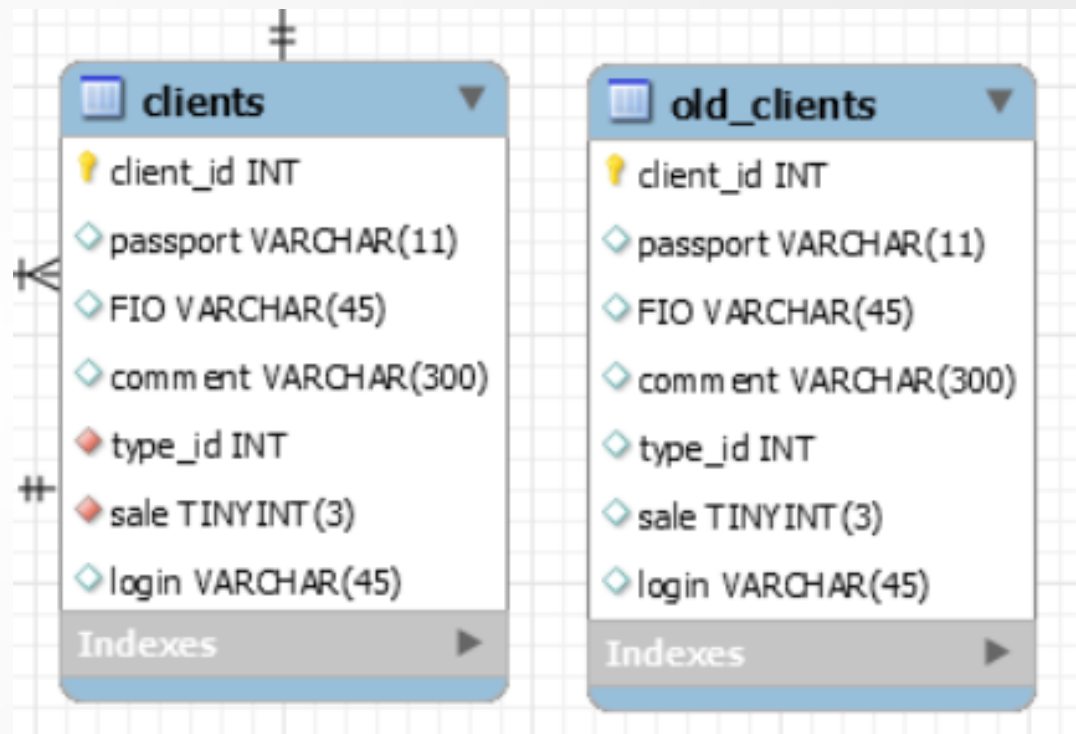
```
kr_sec_sec=# \c - guest1
Пароль пользователя guest1:
Вы подключены к базе данных "kr_sec_sec" как пользователь "guest1".
kr_sec_sec=> select * from clients;
```

client_id	passport	fio	comment	type_id	sale
1	7676545467	Жмышенко Н.П.	Мафиозник	2	50

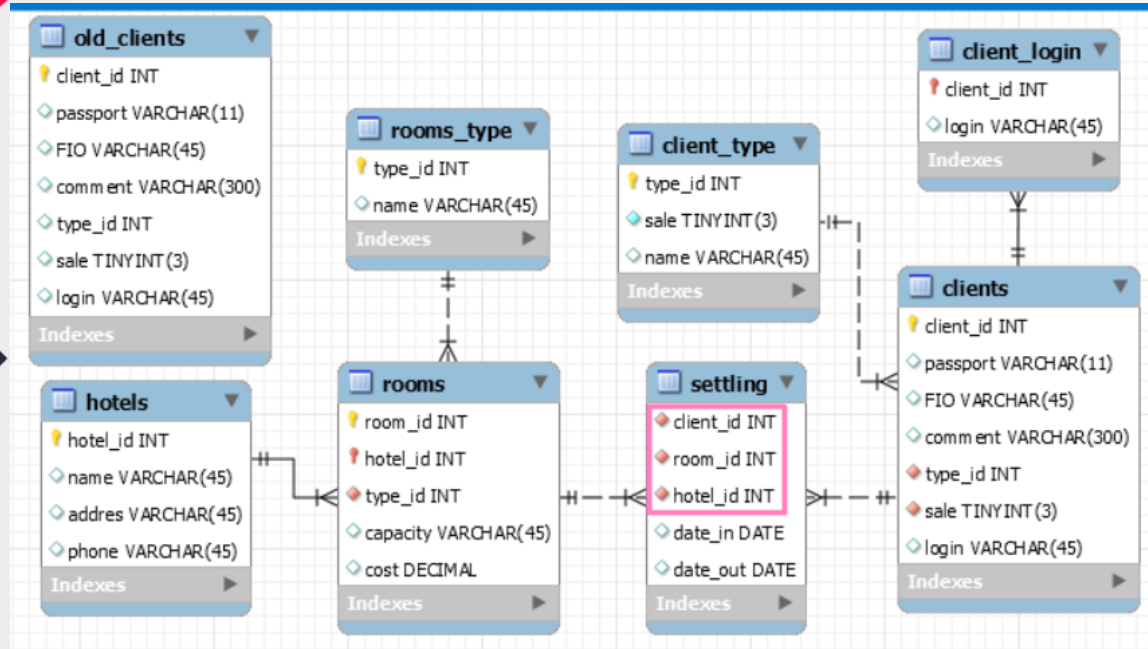
(1 строка)

Идеи в будущем

С помощью представления реализовать сохранение всех пользователей в архив. При этом удалять данные тех, кто съехал



# Идеи в будущем



Гость сможет  
смотреть данные  
своего отеля,  
своей комнаты и  
свой чек  
заселения



# Идеи в будущем

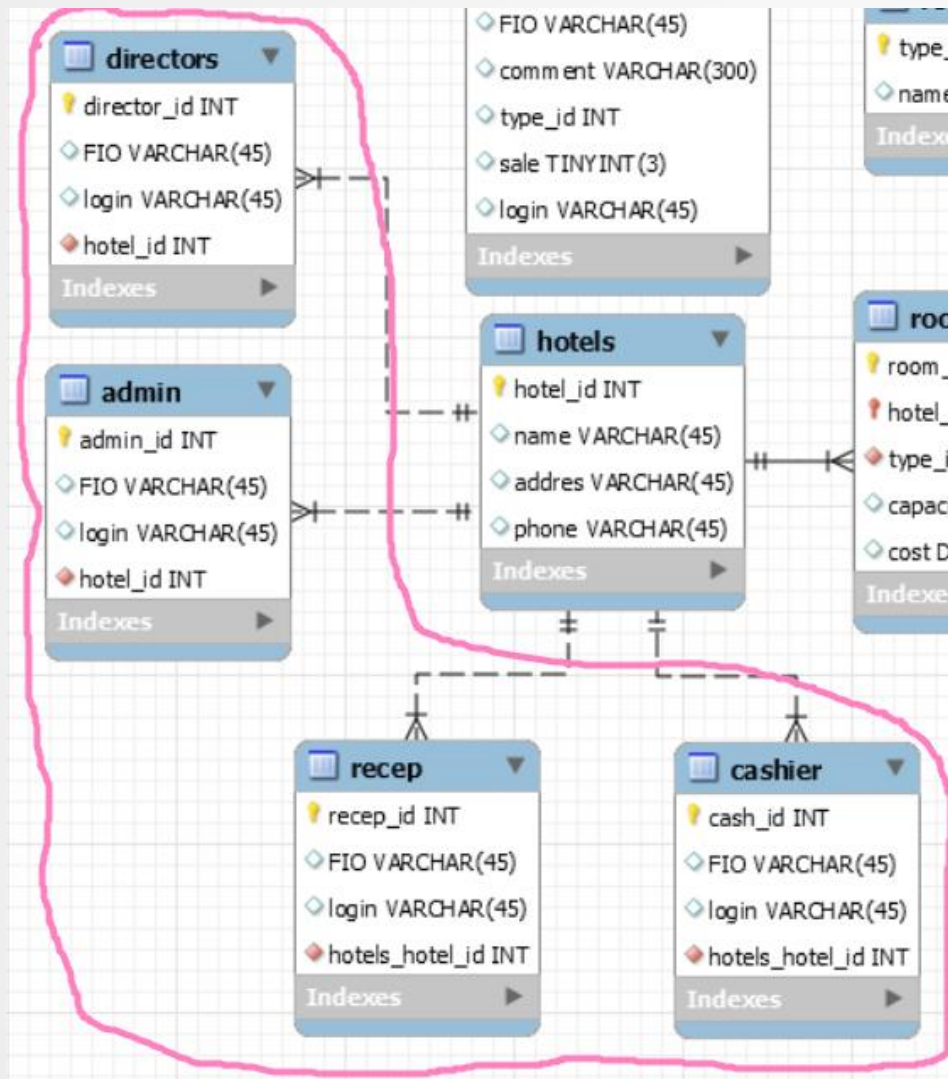


Таблица для каждой роли. Каждый пользователь (персонал) имеет доступ только к данным своего отеля.



# Вывод

Мной была проделана объемная работа. Многое не удалось реализовать. Есть понимание методов реализации. Нужно вложить много времени и сил, чего ограниченное количество. Секционирование по отелям, роли всего персонала и политика защиты ждет нас в будущем!