Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОННИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа №1

Создание UML-диаграммы. Описание технических требований к приложению

Студент: М.А. Ходосевич

Преподаватель: А.И. Крюков

МИНСК 2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc180786678)

[1 СОЗДАНИЕ UML-ДИАГРАММЫ 4](#_Toc180786679)

[1.1 Предметная область 4](#_Toc180786680)

[1.2 Типы объектов 4](#_Toc180786681)

[1.3 Атрибуты объектов 4](#_Toc180786682)

[1.4 Типы связей 5](#_Toc180786683)

[2 ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ 6](#_Toc180786684)

[2.1 Серверное приложение 6](#_Toc180786685)

[2.2 Клиентское приложение. Интерфейс 6](#_Toc180786686)

[3 ВЫПОЛНЕНИЕ ОСНОВНЫХ ОПЕРАЦИЙ В POSTGRESQL 7](#_Toc180786687)

[ЗАКЛЮЧЕНИЕ 9](#_Toc180786688)

# ВВЕДЕНИЕ

В лабораторной работе выполняется концептуального проектирование БД с использованием UML-модели представления данных (модели «сущность-связь»). Требуется разработать UML-модель данных с учетом семантических ограничений заданной предметной области и представить модель в виде UML-диаграммы.

Темой данной лабораторной работы является разработка UML-диаграммы сущностей и связей в организации «Прокат видеодисков».

«Прокат видеодисков» - это стандартная модель, работающая по принципу клиент, услуга, продавец. В качестве продавца выступает сотрудник магазина и взаимодействует напрямую с клиентом. В качестве клиента выступает клиент, который хочет арендовать диск на определенный срок, с определенным фильмом.

# СОЗДАНИЕ UML-ДИАГРАММЫ

Исходное задание: Создать концептуальную модель организации «Прокат видеодисков»и представить сущности и связи в виде UML-диаграммы.

Концептуальная UML-диаграмма представлена на рисунке 1.

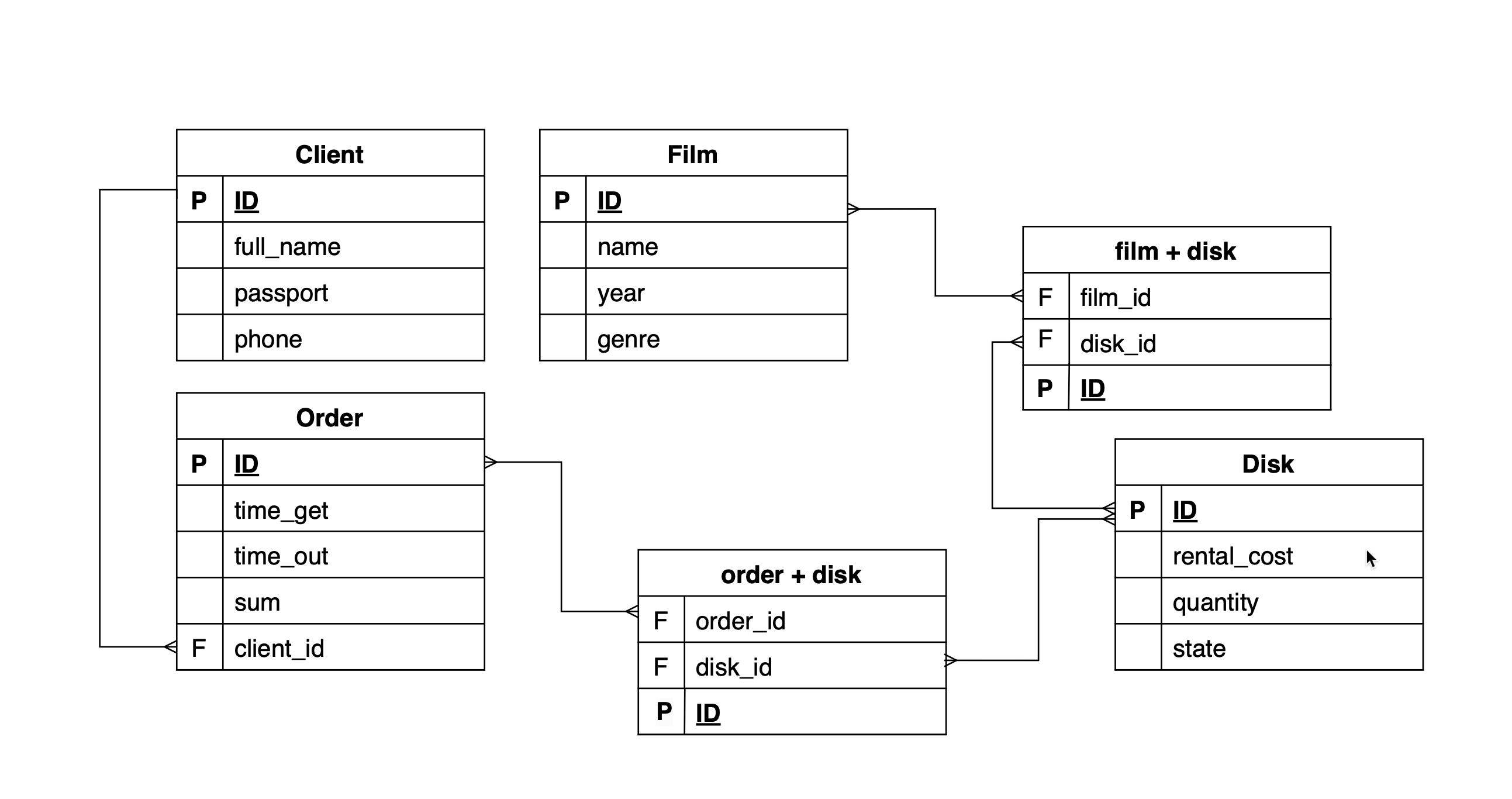


Рисунок 1.1 – UML-диаграмма

## Предметная область

Предметная область – «Прокат видеодисков». Модели по типу «клиент-продавец». Предоставляемая услуга – прокат видеодисков.

## Типы объектов

Для модели «Прокат видеодисков» было выделено 4 типов объектов

1) «Заказ» – заказ, который совершает клиент.

2) «Диск» – cодержит информацию о конкретных дисках (копиях фильмов), находящихся в прокате.

3) «Фильм» – описывает информацию о фильмах, доступных для проката.

4) «Клиент» – человек, арендующий диск.

## Атрибуты объектов

Сущность «Film» содержит атрибуты: name, year, genre;  
Cущность «Disk» содержит атрибуты: rental\_cost, quantity, state

Cущность «Director» содержит атрибуты: full\_name, email

Cущность «Client» содержит атрибуты: full\_name, passport, phone

## Типы связей

Для модели «Прокат видеодисков» можно выделить следующие связи:

1 – «film-disk», описывает фильм, который записан на диск. Мощность связи «многие-ко-многим».

2 – «disk-order», описывает, какие диски взяты в аренду. Мощность связи «многие-ко-многим».

3 – «client-order», описывает заказ, который сделал клиент. Мощность связи «один-ко-многим».

# ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Техтребования содержат принципы построения взаимодействия клиент-серверного приложения в рамках работы с базой данных, но оторвано от конкретной реализации будь то Postgres или BearkleyDB.

Техтребования подразделяются на требования для серверного приложения и требования для интерфейса клиентского приложения.

## Серверное приложение

1) Серверное приложение для реализации соединения с базой данный Postgres будет написано на языке NodeJS.

2) Должны быть предусмотрены CRUD операции для всех таблиц из UML-диаграммы представленной на рисунке 1.1

3) Серверное приложением должно представлять из себя REST API сервер.

4) Серверные операции должны быть описаны обще, для дальнейшего масштабирования и наследования.

5) В серверном приложении должны быть описаны все используемые сущности базы данных.

6) Приложение должно быть оптимизированным.

## Клиентское приложение. Интерфейс

1) Клиентское приложение должно быть написано в SPA, для обеспечения быстродействия и реактивности. Использовать один из популярных фреймворков.

2) Интерфейс приложения должен отвечать принципам UI/UX. Дизайн должен быть удобен, понятен и однозначен.

3) Взаимодействие с серверным приложением должно происходить через REST API.

4) Приложение должно иметь минималистичный дизайн.

5) Приложение должно быть оптимизированным.

# 3 ВЫПОЛНЕНИЕ ОСНОВНЫХ ОПЕРАЦИЙ В POSTGRESQL

Команды для создания и заполнения таблиц и их основные операции будут приведены ниже. Все команды будут представлены на примере создания и заполнения таблицы «Client».

Создание таблицы:

insert into client (full\_name, passport, phone) VALUES

('Петров Петр Петрович', '234567890C214CD3', '+375291234568'),

('Сидоров Сидор Сидорович', '345678901D345DE3', '+375291234569'),

('Александров Александр Александрович', '456789012E456EF5', '+375291234570'),

('Дмитриев Дмитрий Дмитриевич', '567890123F567FG5', '+375291234571'),

('Андреев Андрей Андреевич', '678901234G678GH7', '+375291234572'),

('Николаев Николай Николаевич', '789012345H789HI8', '+375291234573'),

('Алексеев Алексей Алексеевич', '890123456I890IJ9', '+375291234574'),

('Сергеев Сергей Сергеевич', '901234567J901JK9', '+375291234575'),

('Игорев Игорь Игоревич', '012345678K012KL0', '+375291234576'),

('Антонов Антон Антонович', '098765432M098MA2', '+375291234577'),

('Владимиров Владимир Владимирович', '987654321N987NB2', '+375291234578'),

('Михайлов Михаил Михайлович', '876543210O876OC3', '+375291234579'),

('Станиславов Станислав Станиславович', '765432109P765PD5', '+375291234580'),

('Васильев Василий Васильевич', '654321098Q654QE6', '+375291234581'),

('Егоров Егор Егорович', '543210987R543RF6', '+375291234582'),

('Максимов Максим Максимович', '432109876G432SG8', '+375291234583'),

('Федоров Федор Федорович', '321098765T321TH8', '+375291234584'),

('Павлов Павел Павлович', '210987654U210UI9', '+375291234585'),

('Геннадиев Геннадий Геннадиевич', '109876543V109VJ0', '+375291234586'),

('Данилов Даниил Данилович', '098765432W098WK2', '+375291234587'),

('Евгеньев Евгений Евгеньевич', '987654321X987XL2', '+375291234588'),

('Артемов Артем Артемович', '876543210Y876YM3', '+375291234589'),

('Ярославов Ярослав Ярославович', '765432109N765ZN5', '+375291234590'),

('Семенов Семен Семенович', '543210987B543BP6', '+375291234592'),

('Олегов Олег Олегович', '432109876C432CQ8', '+375291234593'),

('Аркадиев Аркадий Аркадиевич', '321098765R321DR8', '+375291234594'),

('Витальев Виталий Витальевич', '210987654S210ES0', '+375291234595'),

('Matvey Khodosevich', '777777777C203RD3', '+375447778888'),

('Rambo', '999999999G787GT7', '+375330990999'),

('Terminator', '111111111H989HI8', '+375259876543');

На рисунке 3.1 представлен результат заполнения таблицы Client.

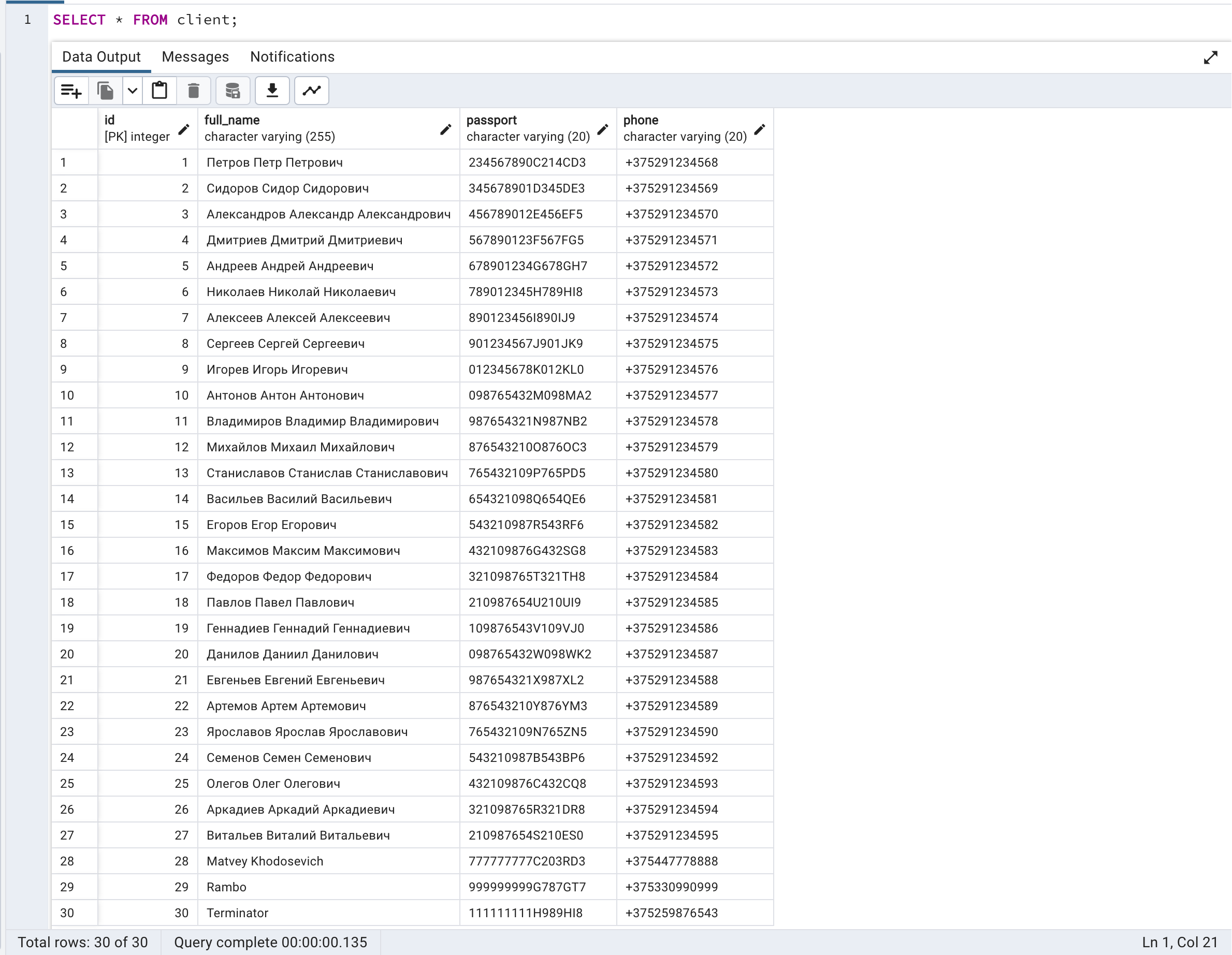


Рисунок 3.1 – Результат заполнения таблицы client

# ЗАКЛЮЧЕНИЕ

В результате работы над лабораторной работой была построена UML-диаграмма организации «Прокат видеодисков». Были выделены основные объекты и представлены связи между ними.

Были описаны технические требования для серверного и клиентского приложения с учетом специфики разработки на языках высокого уровня.

Программа для работы с базами данных PostgreSQL была успешно установлена на ПК.