Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №3

«Реализация SQL-запросов для создания схемы базы данных

и запросов на модификацию данных»

Вариант № 24: прокат видеодисков

Выполнил Проверила:

студент группы 150503: Игнатович А. О.

Ходосевич М. А.

МИНСК 2024

1. **Цель работы**

Реализация реляционной схемы данных с помощью SQL-операторов для создания схемы, таблиц и индексов. Формирование SQL-операторов для добавления, изменения и удаления данных. Реализация транзакций. Импорт и экспорт данных.

1. **Выполнение работы**

Создание базы данных выполняется через:

create database "prokat\_disks";

Далее необходимо создать таблицы соответственно UML-диаграмме из 2 лабораторной. Описание скрипта создания таблицы client приведен ниже. Введены ограничения на ввод номера телефона, так как предполагается, что номер телефона может иметь только вид +375\*\*\*\*\*\*\*\*\*. Также введено ограничение на номер паспорта пользователя, паспорт должен иметь вид \*\*\*\*\*\*\*\*\*X\*\*\*XX\*, где \* – цифра, а Х – буква.

create table client(

id INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY not null,

full\_name varchar(255) not null,

passport varchar(20) not null constraint is\_passport check (passport ~ '^\d{9}[A-Za-z]\d{3}[A-Za-z]{2}\d$'),

phone varchar(20) not null constraint is\_phone check (phone ~ '^(\+375)\d{9}$'));

Скрипт заполнения таблицы client имеет следующий вид:

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 2.1 представлен результат заполнения таблицы client.

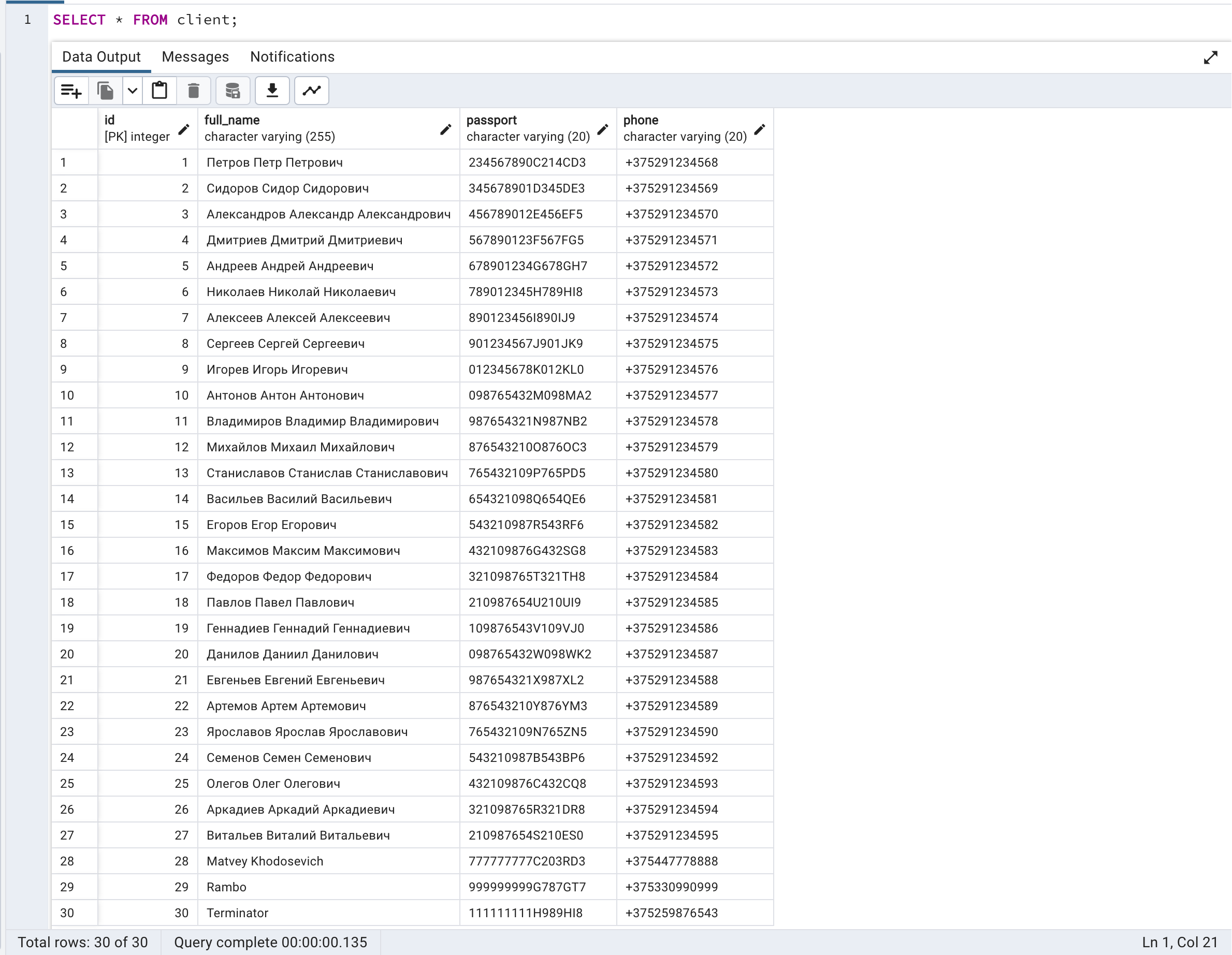


Рисунок 2.1 – Результат заполнения таблицы client

Описание скрипта создания таблицы director приведен ниже.

create table director (

id INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY not null,

full\_name VARCHAR(255) not null,

phone VARCHAR(20) not null,

email VARCHAR(255) not null

);

Скрипт заполнения таблицы director имеет следующий вид:

insert into director (full\_name, phone, email) VALUES

('Кристофер Нолан', '+1234567890', 'nolan@example.com'),

('Стивен Спилберг', '+1987654321', 'spielberg@example.com'),

('Мартин Скорсезе', '+1122334455', 'scorsese@example.com'),

('Квентин Тарантино', '+1555666777', 'tarantino@example.com'),

('Дэвид Финчер', '+1444333222', 'fincher@example.com'),

('Фрэнсис Форд Коппола', '+1999888777', 'coppola@example.com'),

('Питер Джексон', '+1888777666', 'pjackson@example.com'),

('Ридли Скотт', '+1777666555', 'rscott@example.com'),

('Джеймс Кэмерон', '+1666555444', 'cameron@example.com'),

('Стэнли Кубрик', '+1555444333', 'kubrick@example.com'),

('Клинт Иствуд', '+1444333222', 'eastwood@example.com'),

('Гая Ричи', '+1333222111', 'ritchie@example.com'),

('Анджей Вайда', '+1222111000', 'wajda@example.com'),

('Кристоф Вальц', '+1111000999', 'waltz@example.com'),

('Стивен Содерберг', '+1888777666', 'soderbergh@example.com'),

('Пол Томас Андерсон', '+1777666555', 'panderson@example.com'),

('Ким Ки-Дук', '+1666555444', 'kimkiduk@example.com'),

('Ларс фон Триер', '+1555444333', 'trier@example.com'),

('Кен Лоуч', '+1444333222', 'loach@example.com'),

('Майкл Мур', '+1333222111', 'mmoore@example.com'),

('Альфонсо Куарон', '+1222111000', 'cuaron@example.com'),

('Дэррен Аронофски', '+1111000999', 'aronofsky@example.com'),

('Тим Бёртон', '+1888777666', 'burton@example.com'),

('Майк Николс', '+1777666555', 'mikenichols@example.com'),

('Мартин Макдона', '+1666555444', 'macdonagh@example.com'),

('Терри Гиллиам', '+1555444333', 'gilliam@example.com'),

('Алан Паркер', '+1444333222', 'aparker@example.com'),

('Ричард Линклейтер', '+1333222111', 'linklater@example.com'),

('Гор Вербински', '+1222111000', 'gverbinski@example.com'),

('Данила Козловский', '+7722111000','kozlovski@example.com' );

На рисунке 2.2 представлен результат заполнения таблицы director.

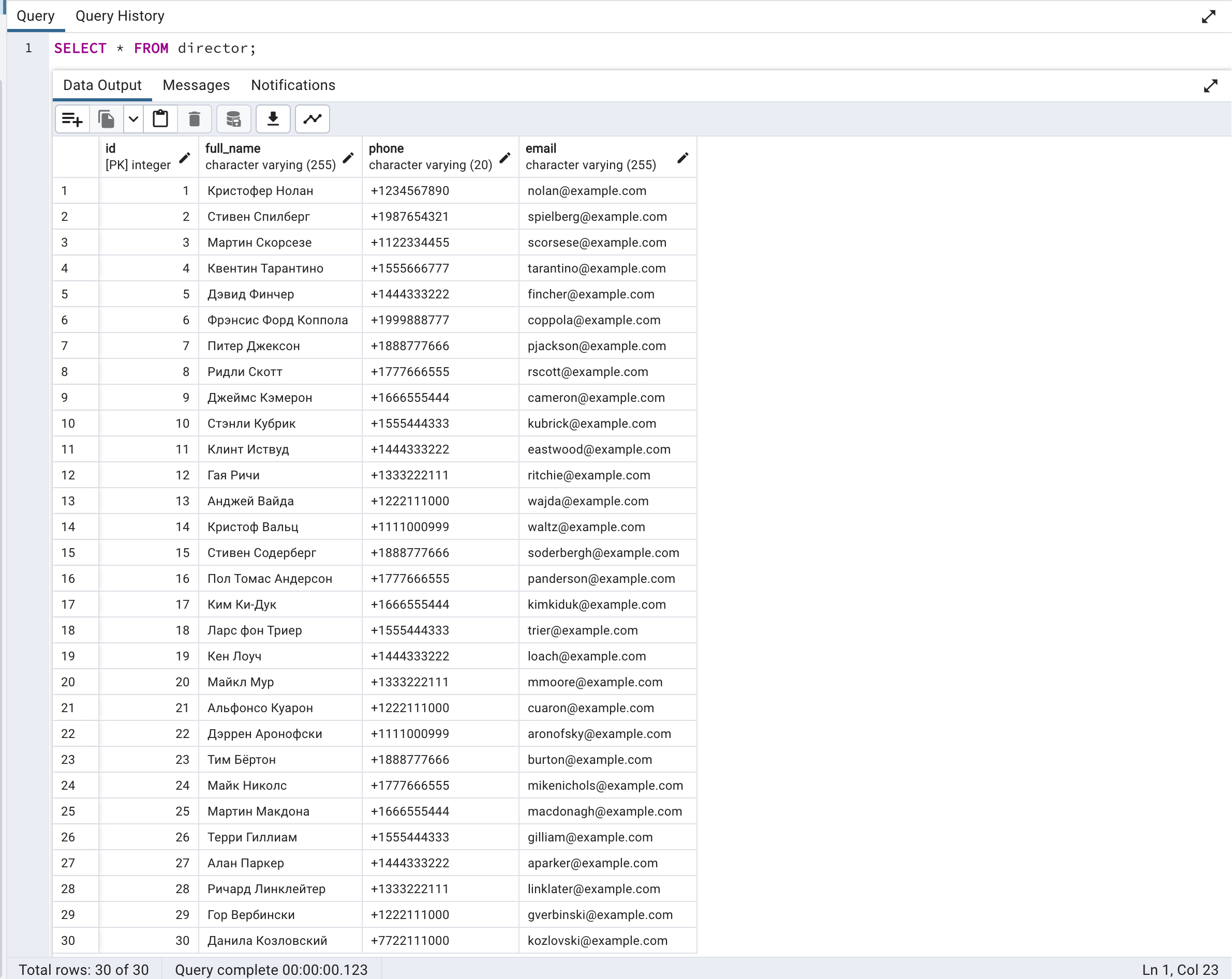


Рисунок 2.2 – Результат заполнения таблицы director

Описание скрипта создания таблицы disk приведен ниже. Введено ограничение is\_cost – цена диска не может быть отрицательной.

create table disk (

id INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY not null,

rental\_cost DECIMAL(10, 2) constraint is\_cost check (cost > 0)

not null,

quantity INT constraint is\_ quantity check (quantity > 0)

not null,

"state" VARCHAR(20) not null);

Скрипт заполнения таблицы disk имеет следующий вид:

INSERT INTO disk (rental\_cost, quantity, "state") VALUES

(5.99, 10, 'доступен'),

(4.99, 8, 'доступен'),

(6.99, 12, 'доступен'),

(3.99, 15, 'доступен'),

(7.99, 7, 'доступен'),

(5.99, 9, 'доступен'),

(6.99, 11, 'доступен'),

(4.99, 10, 'доступен'),

(6.99, 8, 'доступен'),

(5.99, 14, 'доступен'),

(7.99, 10, 'доступен'),

(4.99, 12, 'доступен'),

(5.99, 9, 'доступен'),

(6.99, 11, 'доступен'),

(4.99, 13, 'доступен'),

(7.99, 8, 'доступен'),

(5.99, 15, 'доступен'),

(6.99, 9, 'доступен'),

(4.99, 11, 'доступен'),

(5.99, 10, 'доступен'),

(6.99, 12, 'доступен'),

(3.99, 14, 'доступен'),

(7.99, 8, 'доступен'),

(5.99, 10, 'доступен'),

(6.99, 11, 'доступен'),

(4.99, 9, 'доступен'),

(5.99, 13, 'доступен'),

(7.99, 8, 'доступен'),

(4.99, 12, 'доступен'),

(2.99, 3, 'доступен');

На рисунке 2.3 представлен результат заполнения таблицы disk.

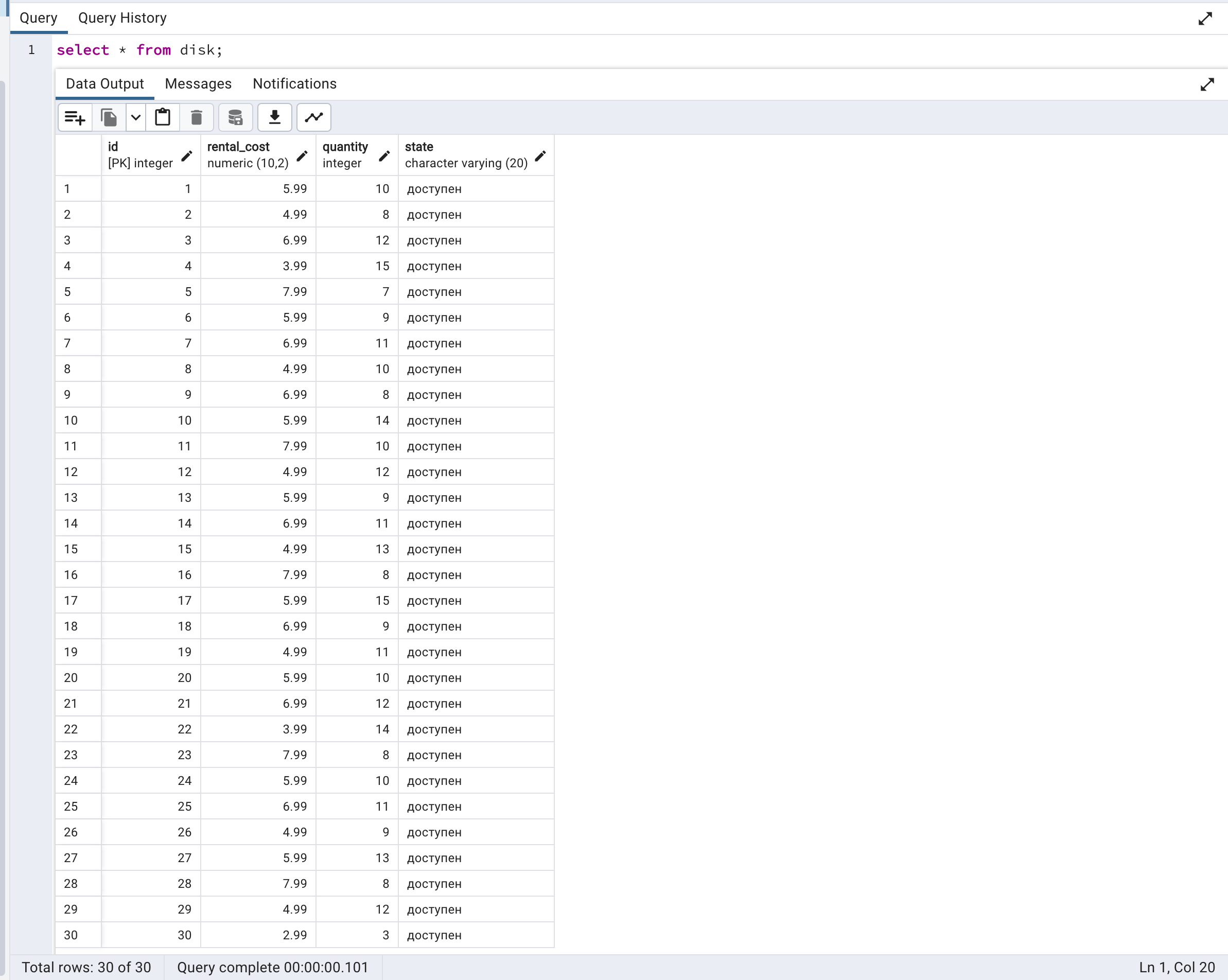


Рисунок 2.3 – Результат заполнения таблицы disk

Ниже приведен скрипт создания таблицы film.

create table film (

id INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY not null,

name VARCHAR(255) not null,

year DATE not null,

genre VARCHAR(100) not null,

director\_id INT not null,

constraint fk\_director\_id FOREIGN KEY (director\_id) REFERENCES director(id));

Скрипт заполнения таблицы film имеет следующий вид:

INSERT INTO film (name, year, genre, director\_id) VALUES

('Начало', '2010-07-16', 'фантастика', 1),

('Темный рыцарь', '2008-07-18', 'боевик', 1),

('Властелин колец: Возвращение короля', '2003-12-17', 'фэнтези', 7),

('Интерстеллар', '2014-11-07', 'фантастика', 1),

('Джентльмены', '2019-12-03', 'криминал', 12),

('Бегущий по лезвию', '1982-06-25', 'фантастика', 4),

('Побег из Шоушенка', '1994-10-14', 'драма', 2),

('Крестный отец', '1972-03-24', 'криминал', 6),

('Список Шиндлера', '1993-12-15', 'биография', 6),

('Парк Юрского периода', '1993-06-11', 'приключения', 7),

('Пятый элемент', '1997-05-07', 'фантастика', 10),

('Красота по-американски', '1999-09-15', 'драма', 11),

('Таксист', '1976-02-08', 'триллер', 9),

('Мементо', '2000-10-11', 'триллер', 1),

('Достучаться до небес', '1997-10-30', 'комедия', 14),

('Семь', '1995-09-22', 'триллер', 5),

('Искусственный разум', '2001-06-26', 'фантастика', 3),

('Плотина', '1975-11-04', 'драма', 15),

('Изгой', '2000-12-22', 'драма', 16),

('Зеленая миля', '1999-12-06', 'фэнтези', 2),

('Шоу Трумана', '1998-06-05', 'комедия', 8),

('Шестое чувство', '1999-08-06', 'триллер', 17),

('Гарри Поттер и философский камень', '2001-11-16', 'фэнтези', 18),

('Безумный Макс: Дорога ярости', '2015-05-14', 'боевик', 19),

('Сонная лощина', '1999-04-21', 'драма', 20),

('По социальным стандартам', '1980-09-19', 'драма', 21),

('Вечное сияние чистого разума', '2004-03-19', 'романтика', 22),

('Спасти рядового Райана', '1998-07-24', 'военный', 23),

('Волк с Уолл-стрит', '2013-12-25', 'криминал', 24),

('Красный воробей', '2018-02-15', 'триллер', 25);

На рисунке 2.4 представлен результат заполнения таблицы film.

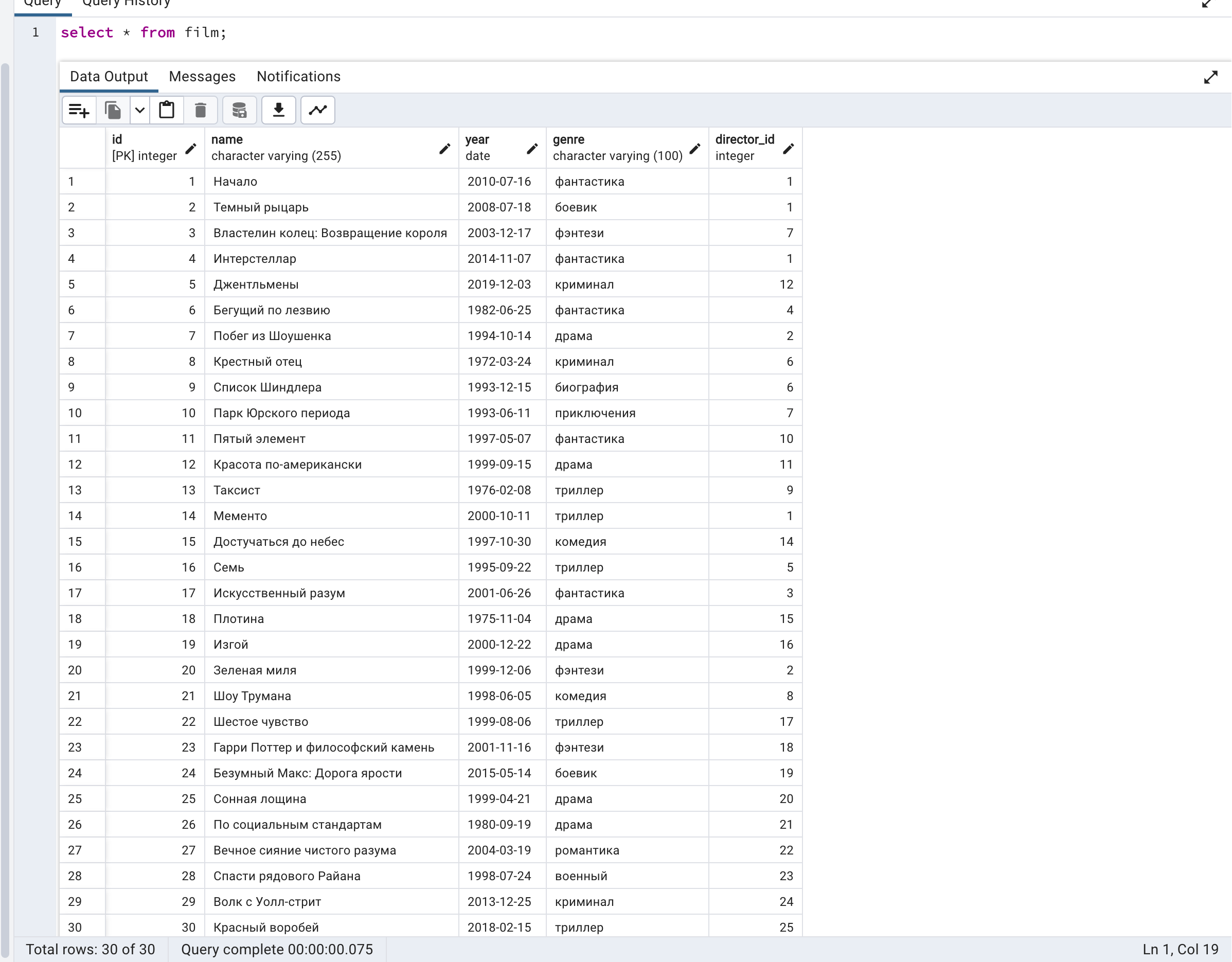


Рисунок 2.4 – Результат заполнения таблицы film

Описание скрипта создания таблицы order приведен ниже.

create table "order" (

id INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY not null,

time\_get TIMESTAMP not null,

time\_out TIMESTAMP not null,

"sum" DECIMAL(10, 2) not null,

client\_id INT not null,

constraint fk\_client\_director FOREIGN KEY (client\_id) REFERENCES client(id));

Скрипт заполнения таблицы order имеет следующий вид:

INSERT INTO "order" (time\_get, time\_out, "sum", client\_id) VALUES

('2024-02-15 08:00:00', '2024-02-15 12:00:00', 25.50, 1),

('2024-02-15 09:30:00', '2024-02-15 14:00:00', 45.75, 2),

('2024-02-15 10:45:00', '2024-02-15 16:30:00', 32.20, 3),

('2024-02-15 12:15:00', '2024-02-15 17:45:00', 18.90, 4),

('2024-02-15 14:00:00', '2024-02-15 19:30:00', 27.60, 5),

('2024-02-15 15:30:00', '2024-02-15 20:45:00', 40.00, 6),

('2024-02-15 17:00:00', '2024-02-15 22:00:00', 35.80, 7),

('2024-02-15 08:30:00', '2024-02-15 12:45:00', 22.50, 8),

('2024-02-15 10:00:00', '2024-02-15 14:15:00', 28.75, 9),

('2024-02-15 11:45:00', '2024-02-15 16:30:00', 30.90, 10),

('2024-02-15 13:30:00', '2024-02-15 18:45:00', 17.25, 11),

('2024-02-15 15:15:00', '2024-02-15 20:30:00', 36.40, 12),

('2024-02-15 17:00:00', '2024-02-15 22:15:00', 28.80, 13),

('2024-02-15 09:00:00', '2024-02-15 13:15:00', 24.75, 14),

('2024-02-15 10:45:00', '2024-02-15 15:30:00', 33.20, 15),

('2024-02-15 12:30:00', '2024-02-15 17:45:00', 19.90, 16),

('2024-02-15 14:15:00', '2024-02-15 19:30:00', 26.60, 17),

('2024-02-15 16:00:00', '2024-02-15 21:15:00', 41.00, 18),

('2024-02-15 17:45:00', '2024-02-15 22:45:00', 34.80, 19),

('2024-02-15 08:15:00', '2024-02-15 12:30:00', 23.50, 20),

('2024-02-15 09:45:00', '2024-02-15 14:00:00', 29.75, 21),

('2024-02-15 11:30:00', '2024-02-15 16:45:00', 31.90, 22),

('2024-02-15 13:15:00', '2024-02-15 18:30:00', 18.25, 23),

('2024-02-15 15:00:00', '2024-02-15 20:15:00', 37.40, 24),

('2024-02-15 16:45:00', '2024-02-15 21:45:00', 29.80, 25),

('2024-02-15 08:30:00', '2024-02-15 12:45:00', 24.50, 26),

('2024-02-15 10:15:00', '2024-02-15 14:30:00', 30.75, 27),

('2024-02-15 10:15:00', '2024-02-15 16:30:00', 50.75, 28),

('2024-02-15 10:15:00', '2024-02-15 17:30:00', 60.75, 29);

('2024-02-15 10:15:00', '2024-02-15 19:30:00', 70.75, 30);

На рисунке 2.5 представлен результат заполнения таблицы order.

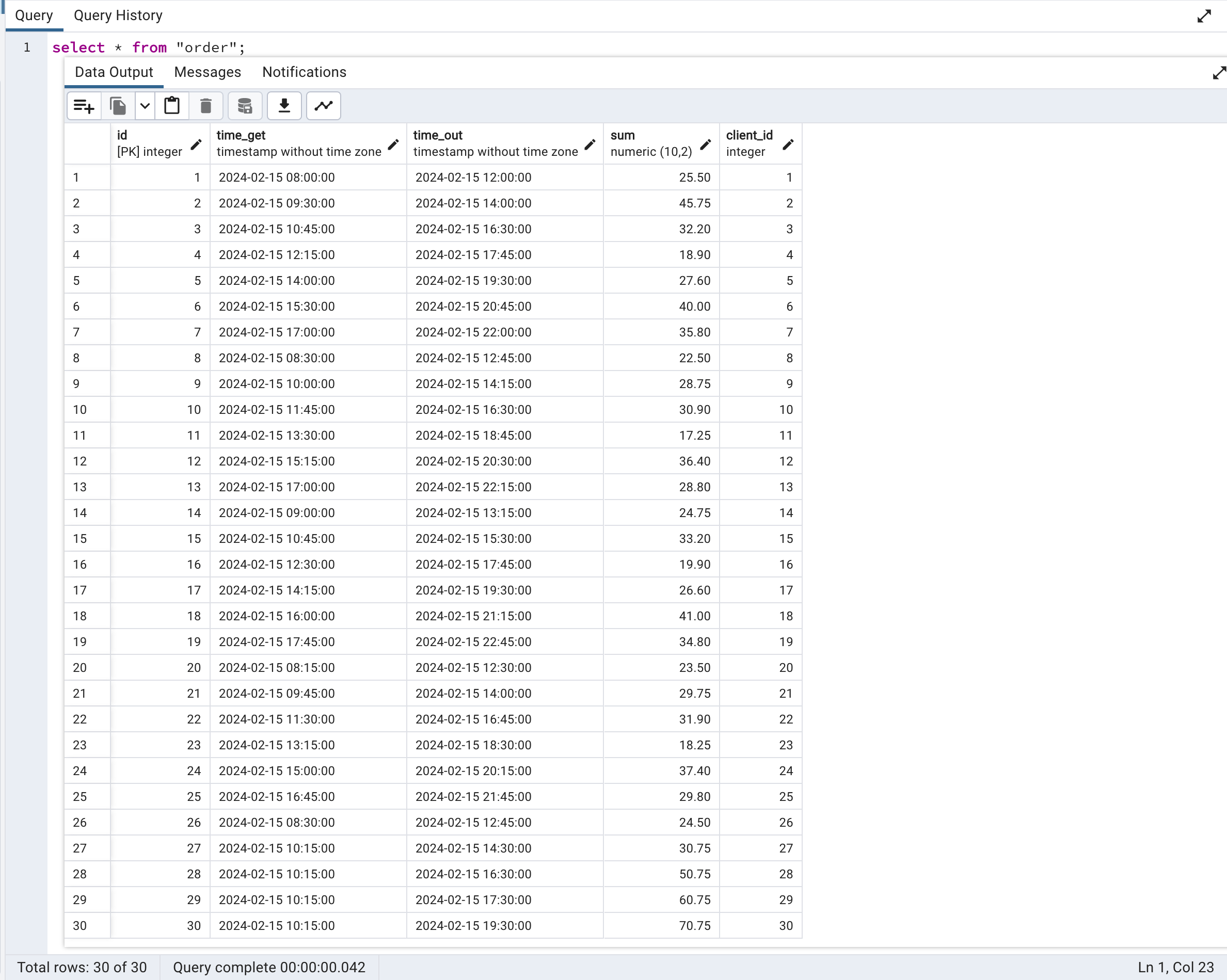


Рисунок 2.5 – Результат заполнения таблицы order

Ниже приведен скрипт создания таблицы review.

create table review (

id INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY not null,

mark INT not null,

"comment" TEXT not null,

film\_id INT not null,

client\_id INT not null,

constraint fk\_film\_id FOREIGN KEY (film\_id) REFERENCES film(id),

constraint fk\_client\_id FOREIGN KEY (client\_id) REFERENCES client(id));

Скрипт заполнения таблицы review имеет следующий вид:

INSERT INTO review (mark, "comment", film\_id, client\_id) VALUES

(5, 'Отличный фильм, сюжет завораживает!', 1, 1),

(4, 'Неплохой боевик, но слишком много сцен на грани фантастики.', 2, 2),

(5, 'Эпическое завершение трилогии, прекрасное кино!', 3, 3),

(4, 'Интересный фильм, но иногда сложно понять смысл.', 4, 4),

(3, 'Средний криминальный фильм, ничего особенного.', 5, 5),

(5, 'Шедевр на века!', 6, 6),

(5, 'Лучший фильм всех времен!', 7, 7),

(4, 'Иконический криминальный фильм, но не для слабонервных.', 8, 8),

(5, 'Великолепное кино, красивая история.', 9, 9),

(4, 'Захватывающее путешествие по острову динозавров.', 10, 10),

(5, 'Фантастический мир будущего, яркие образы и захватывающий сюжет!', 11, 11),

(3, 'Не для каждого, но все же интересно.', 12, 12),

(5, 'Отличный триллер, волнующий до самого конца.', 13, 13),

(4, 'Позитивное кино, поднимает настроение!', 14, 14),

(5, 'Увлекательный и стильный триллер.', 15, 15),

(4, 'Философский фильм с красивой картинкой.', 16, 16),

(5, 'Одна из лучших кинолент в истории.', 17, 17),

(3, 'Сложно понять суть, но фильм заставляет задуматься.', 18, 18),

(5, 'Завораживающая фантастика.', 19, 19),

(4, 'Отличное драматическое кино, оставляет многое на размышление.', 20, 20),

(5, 'Захватывающая история, полная неожиданных поворотов.', 21, 21),

(4, 'Шикарная комедия с интересным сюжетом.', 22, 22),

(5, 'Необычный триллер, который заставляет пересматривать фильмы.', 23, 23),

(4, 'Мир магии и приключений, знакомство с героями не оставляет равнодушным.', 24, 24),

(5, 'Отличная драма, сильная игра актеров.', 25, 25),

(3, 'Тяжелый фильм о жизни, который заставляет задуматься.', 26, 26),

(5, 'Фантастическая картина, которую стоит посмотреть каждому.', 27, 27),

(4, 'Захватывающий военный фильм, который тронет душу.', 28, 28),

(5, 'Живое и реалистичное кино о финансовом мире.', 29, 29),

(4, 'Тревожный и захватывающий триллер.', 30, 30),

(5, 'Неплохой фильм!', 4, 1);

На рисунке 2.6 представлен результат заполнения таблицы review.

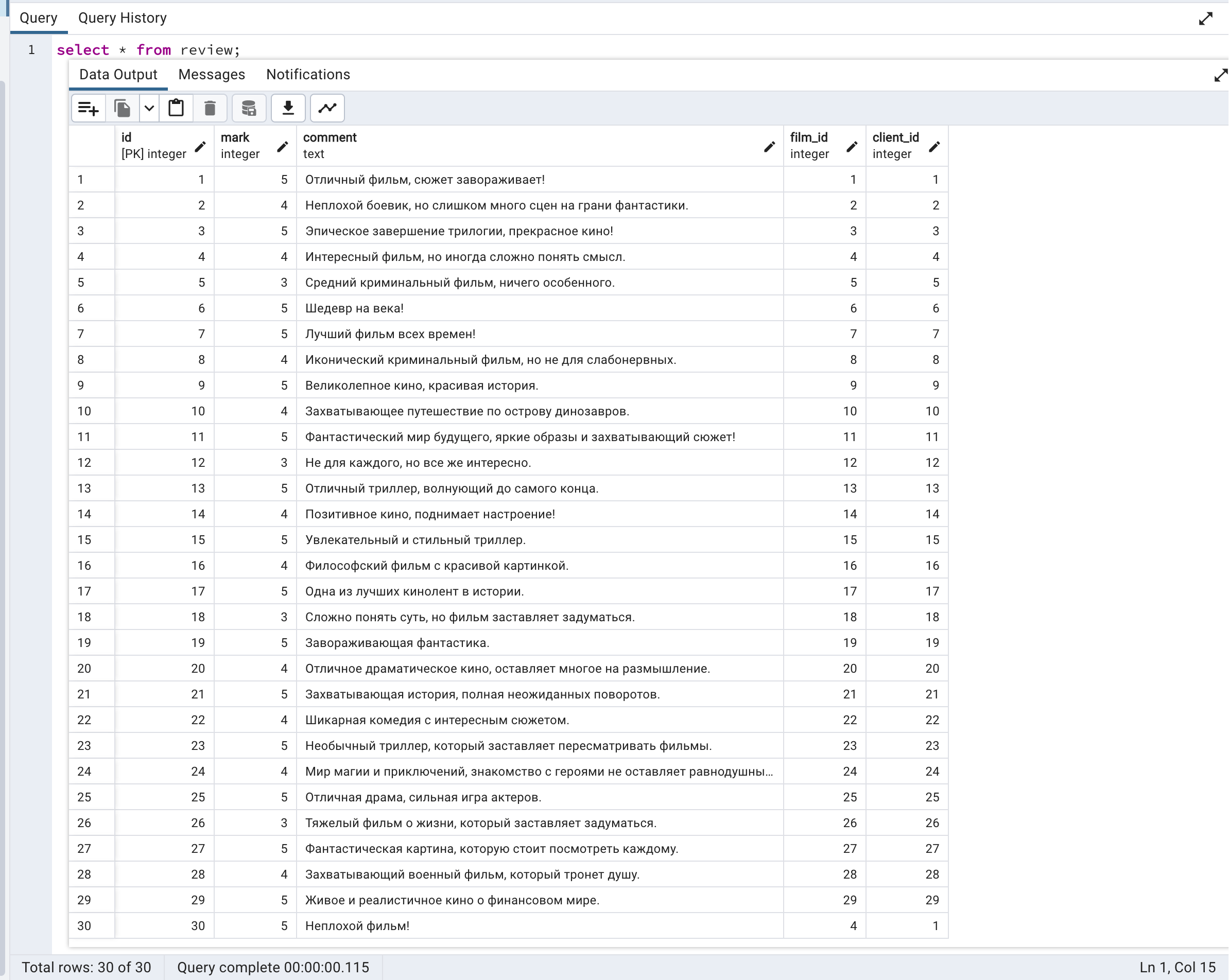


Рисунок 2.6 – Результат заполнения таблицы review

Описание скрипта создания таблицы film+disk приведен ниже.

create table "film+disk" (

id INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY not null,

film\_id INT not null,

disk\_id INT not null,

constraint fk\_film\_id FOREIGN KEY (film\_id) REFERENCES film (id),

constraint fk\_disk\_id FOREIGN KEY (disk\_id) REFERENCES disk(id));

Скрипт заполнения таблицы film+disk имеет следующий вид:

INSERT INTO "film+disk" (film\_id, disk\_id) VALUES

(1, 1), (2, 2), (3, 3), (4, 4), (5, 5),

(6, 6), (7, 7), (8, 8), (9, 9), (10, 10),

(11, 11), (12, 12), (13, 13), (14, 14), (15, 15),

(16, 16), (17, 17), (18, 18), (19, 19), (20, 20),

(21, 21), (22, 22), (23, 23), (24, 24), (25, 25),

(26, 26), (27, 27), (28, 28), (29, 29), (30,30);

На рисунке 2.7 представлен результат заполнения таблицы film+disk.

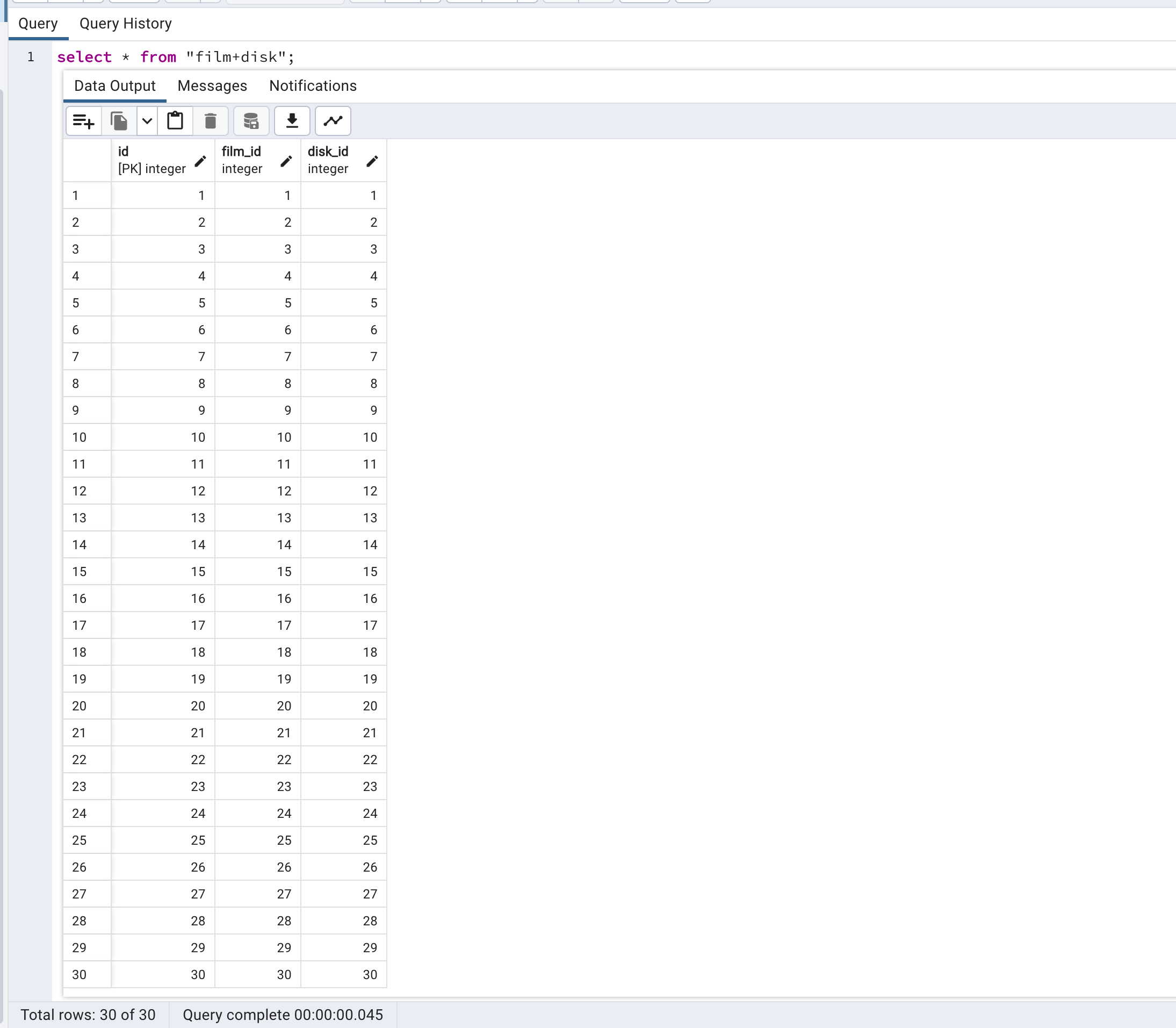


Рисунок 2.7 – Результат заполнения таблицы film+disk

Описание скрипта создания таблицы order+disk приведен ниже.

create table "order+disk" (

id INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY not null,

order\_id INT not null,

disk\_id INT not null,

constraint fk\_order\_id FOREIGN KEY (order\_id) REFERENCES "order"(id),

constraint fk\_disk\_id FOREIGN KEY (disk\_id) REFERENCES disk(id));

Скрипт заполнения таблицы order+disk имеет следующий вид:

INSERT INTO "order+disk" (order\_id, disk\_id) VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5),

(6, 6),

(7, 7),

(8, 8),

(9, 9),

(10, 10),

(11, 11),

(12, 12),

(13, 13),

(14, 14),

(15, 15),

(16, 16),

(17, 17),

(18, 18),

(19, 19),

(20, 20),

(21, 21),

(22, 22),

(23, 23),

(24, 24),

(25, 25),

(26, 26),

(27, 27),

(28, 28),

(29, 29),

(30, 30);

На рисунке 2.8 представлен результат заполнения таблицы order+disk.

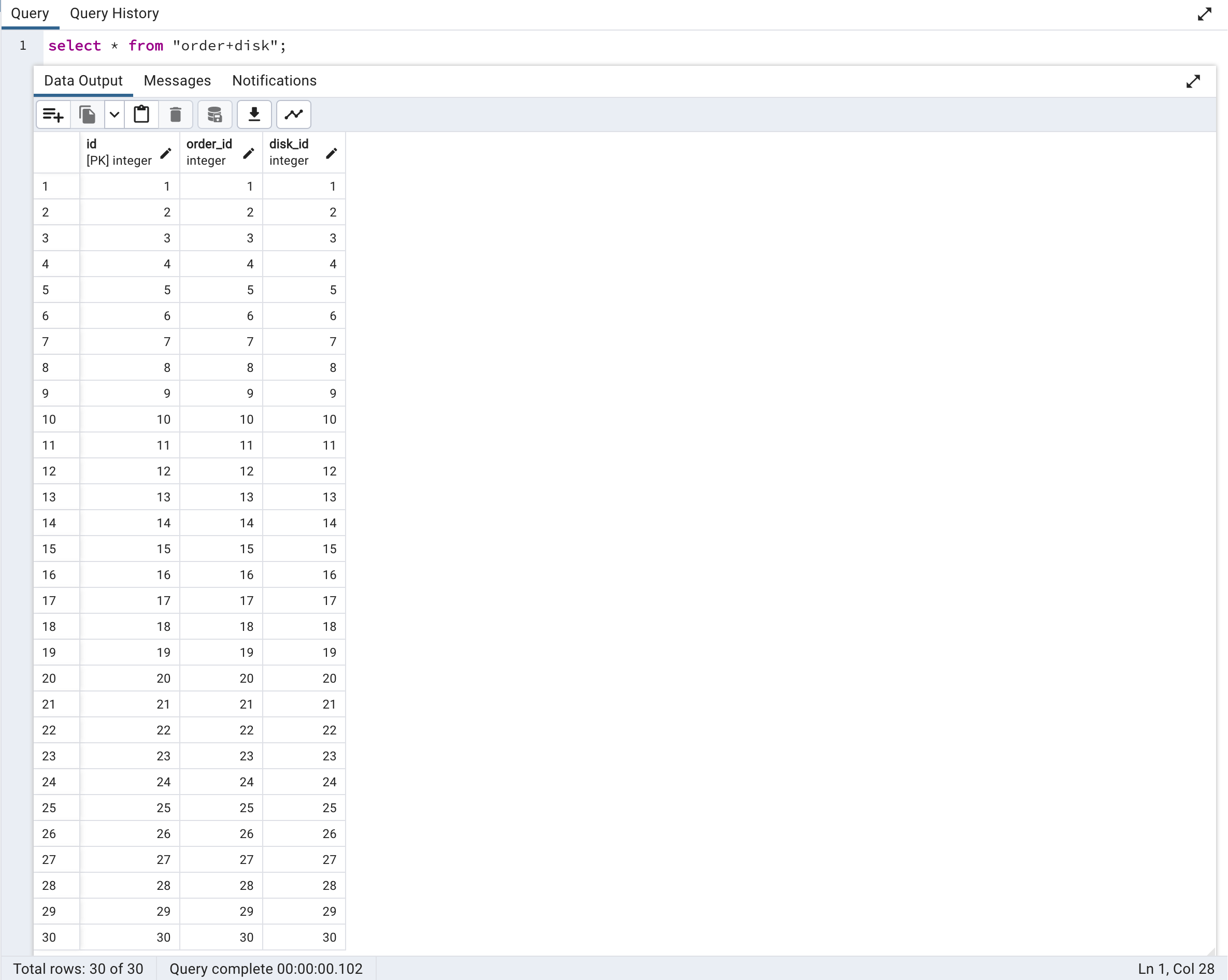


Рисунок 2.8 – Результат заполнения таблицы order+disk

Далее на рисунках 2.10, 2.11 и 2.12 будут приведены ER-диаграмма, UML-диаграмма из отчёта по 2 лабораторной работе и сгенерированная ERD в pgAdmin соответственно.

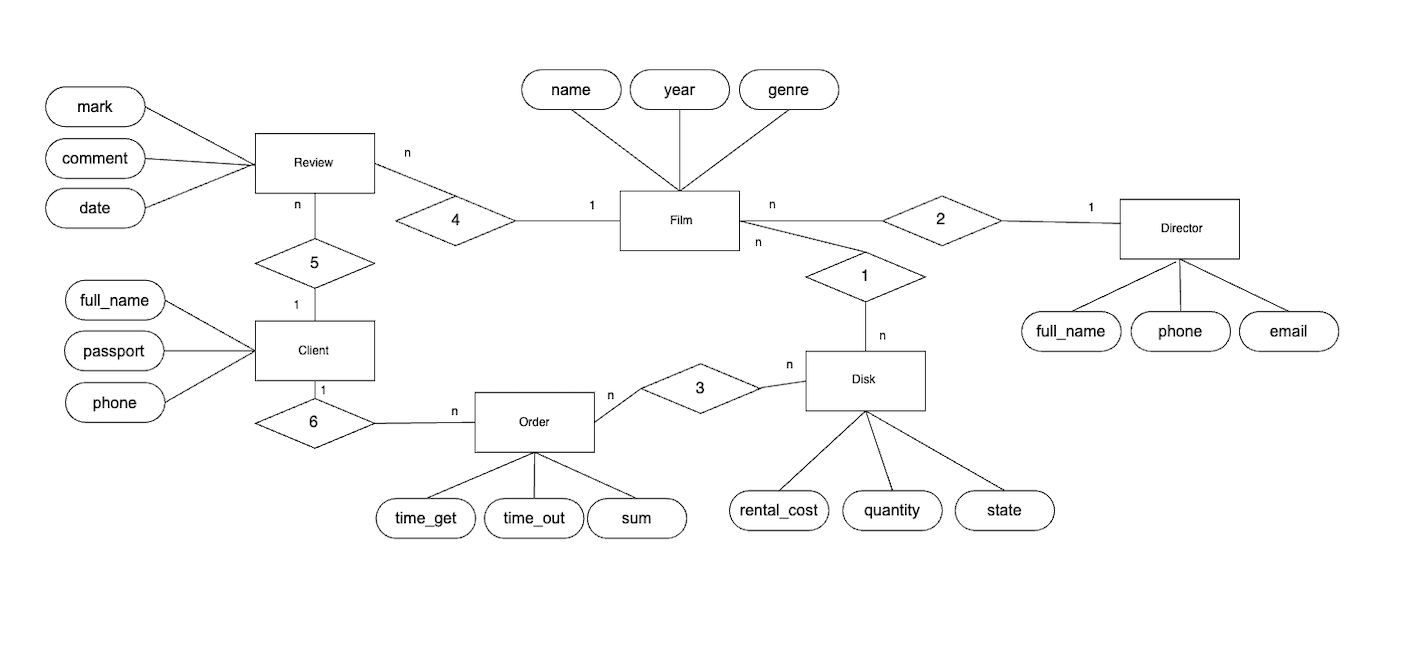


Рисунок 2.10 – ER-диаграмма модели «Прокат видеодисков»

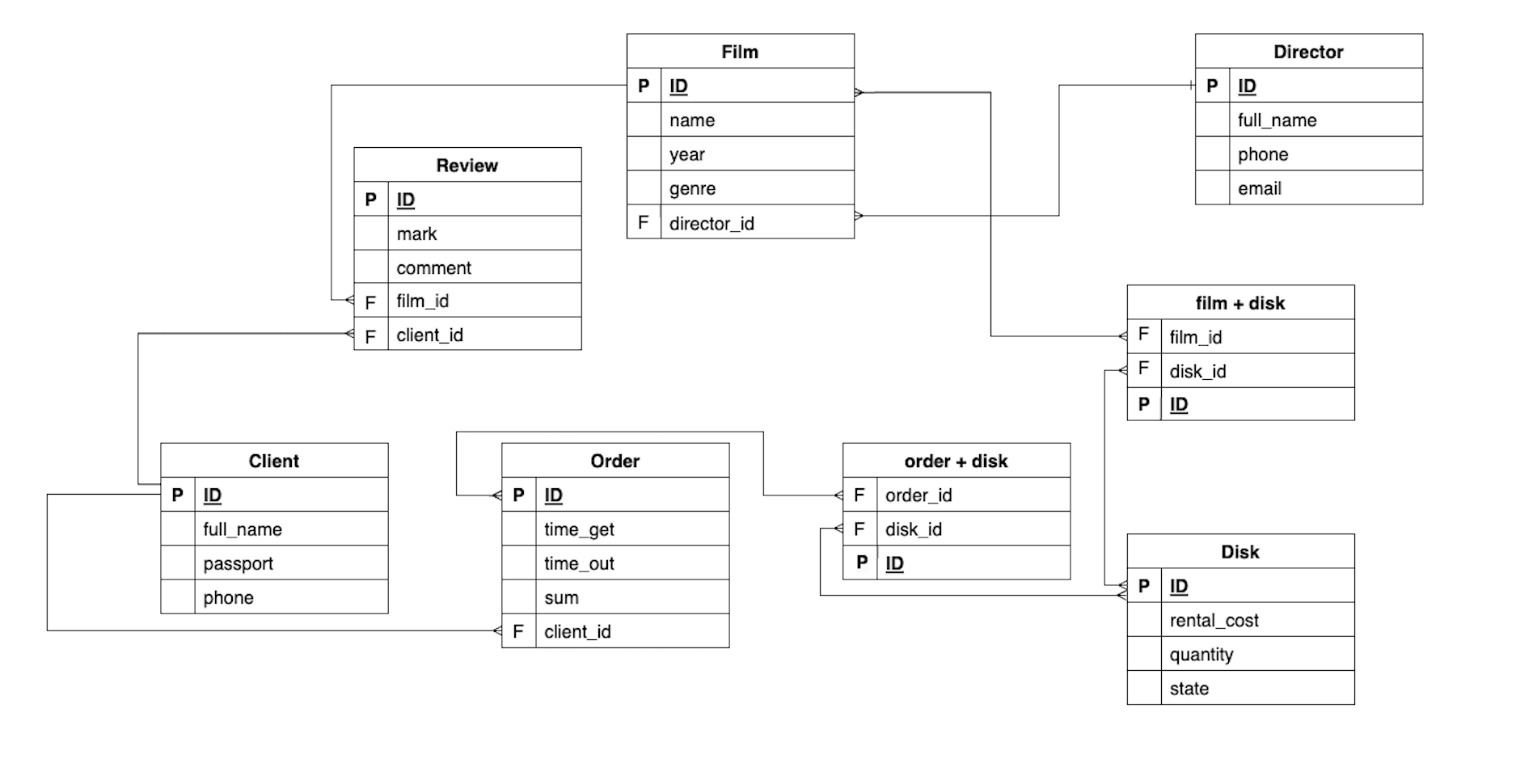


Рисунок 2.11 – UML-диаграмма модели «Прокат видеодисков»

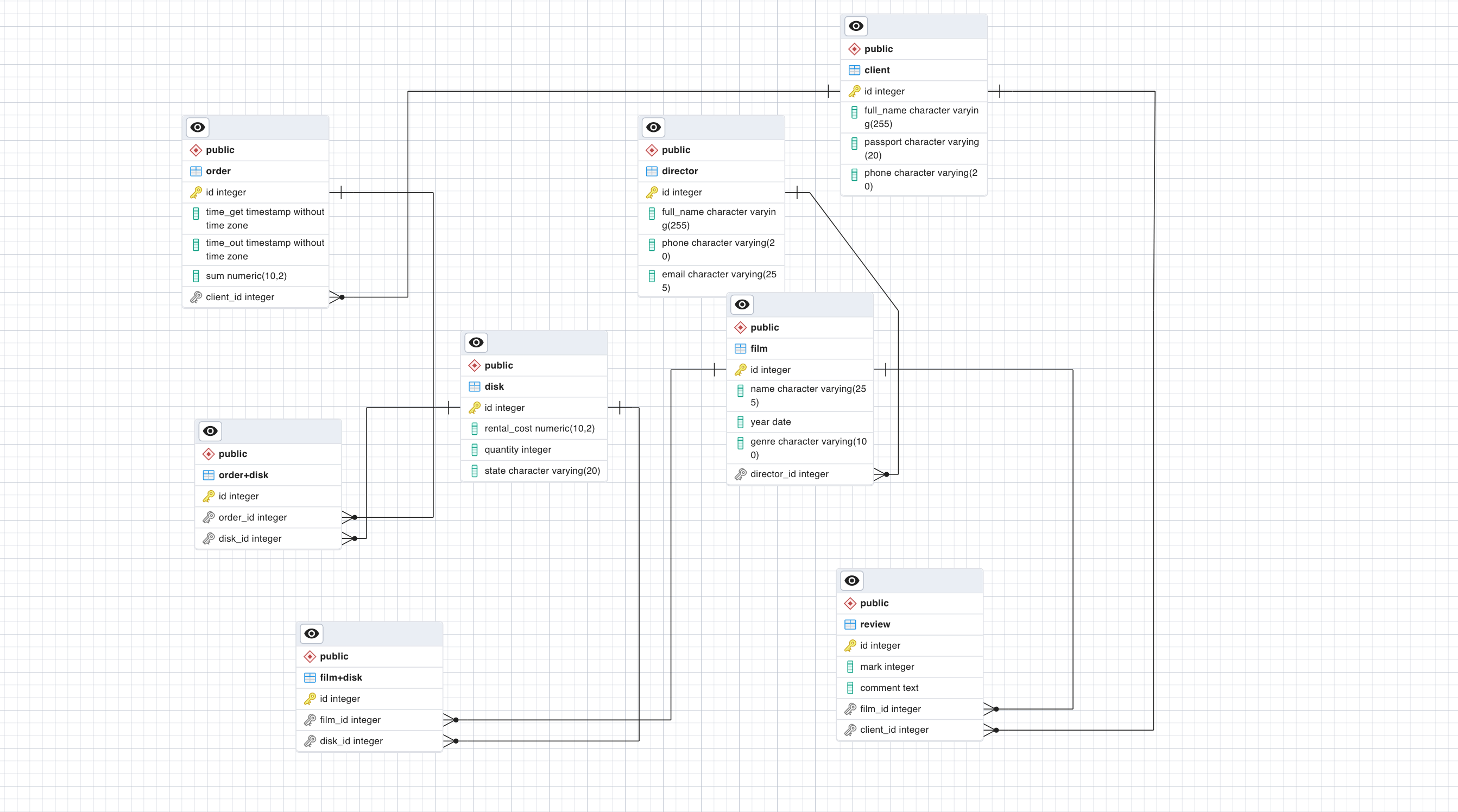


Рисунок 2.12 – ERD-диаграмма модели «Прокат видеодисков» из pgAdmin

Далее будут приведены скрипты для простых действий по изменению структуры таблицы. Создание временной таблицы с ограничением:

create table matvey(

id INT PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY not null,

full\_name VARCHAR(20) NOT NULL,

age INT CONSTRAINT is\_age CHECK (age > 0),

phone VARCHAR(20)

);

Скрипт переименования колонки “ phone ” в “email”:

ALTER TABLE matvey

RENAME COLUMN phone TO email;

Скрипт для удаления ограничения для age:

ALTER TABLE matvey

DROP CONSTRAINT is\_age;

Добавление этого ограничения заново (новое создание):

ALTER TABLE matvey

ADD CONSTRAINT is\_age CHECK (age > 0);

Удаление временной таблицы:

DROP TABLE matvey;