

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
студент группы 150503:
Ходосевич М. А.

Проверила:
Игнатович А. О.

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.

2 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 `X` – 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.

1 SELECT * FROM client;

Data Output Messages Notifications

	id [PK] integer	full_name character varying (255)	passport character varying (20)	phone character varying (20)
1	1	Петров Петр Петрович	234567890C214CD3	+375291234568
2	2	Сидоров Сидор Сидорович	345678901D345DE3	+375291234569
3	3	Александров Александр Александрович	456789012E456EF5	+375291234570
4	4	Дмитриев Дмитрий Дмитриевич	567890123F567FG5	+375291234571
5	5	Андреев Андрей Андреевич	678901234G678GH7	+375291234572
6	6	Николаев Николай Николаевич	789012345H789HI8	+375291234573
7	7	Алексеев Алексей Алексеевич	890123456I890IJ9	+375291234574
8	8	Сергеев Сергей Сергеевич	901234567J901JK9	+375291234575
9	9	Игорев Игорь Игоревич	012345678K012KL0	+375291234576
10	10	Антонов Антон Антонович	098765432M098MA2	+375291234577
11	11	Владимиров Владимир Владимирович	987654321N987NB2	+375291234578
12	12	Михайлов Михаил Михайлович	876543210O876OC3	+375291234579
13	13	Станиславов Станислав Станиславович	765432109P765PD5	+375291234580
14	14	Васильев Василий Васильевич	654321098Q654QE6	+375291234581
15	15	Егоров Егор Егорович	543210987R543RF6	+375291234582
16	16	Максимов Максим Максимович	432109876G432SG8	+375291234583
17	17	Федоров Федор Федорович	321098765T321TH8	+375291234584
18	18	Павлов Павел Павлович	210987654U210UI9	+375291234585
19	19	Геннадиев Геннадий Геннадиевич	109876543V109VJ0	+375291234586
20	20	Данилов Даниил Данилович	098765432W098WK2	+375291234587
21	21	Евгеньев Евгений Евгеньевич	987654321X987XL2	+375291234588
22	22	Артемьев Артем Артемович	876543210Y876YM3	+375291234589
23	23	Ярославов Ярослав Ярославович	765432109N765ZN5	+375291234590
24	24	Семенов Семен Семенович	543210987B543BP6	+375291234592
25	25	Олегов Олег Олегович	432109876C432CQ8	+375291234593
26	26	Аркадиев Аркадий Аркадиевич	321098765R321DR8	+375291234594
27	27	Витальев Виталий Витальевич	210987654S210ES0	+375291234595
28	28	Matvey Khodosevich	777777777C203RD3	+375447778888
29	29	Rambo	999999999G787GT7	+375330990999
30	30	Terminator	111111111H989HI8	+375259876543

Total rows: 30 of 30 Query complete 00:00:00.135 Ln 1, Col 21

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 Coppola', '+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'),
```

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'), (7.99, 8, 'доступен'),  
(4.99, 8, 'доступен'), (5.99, 15, 'доступен'),  
(6.99, 12, 'доступен'), (6.99, 9, 'доступен'),  
(3.99, 15, 'доступен'), (4.99, 11, 'доступен'),  
(7.99, 7, 'доступен'), (5.99, 10, 'доступен'),  
(5.99, 9, 'доступен'), (6.99, 12, 'доступен'),  
(6.99, 11, 'доступен'), (3.99, 14, 'доступен'),  
(4.99, 10, 'доступен'), (7.99, 8, 'доступен'),  
(6.99, 8, 'доступен'), (5.99, 10, 'доступен'),  
(5.99, 14, 'доступен'), (6.99, 11, 'доступен'),  
(7.99, 10, 'доступен'), (4.99, 9, 'доступен'),  
(4.99, 12, 'доступен'), (5.99, 13, 'доступен'),  
(5.99, 9, 'доступен'), (7.99, 8, 'доступен'),  
(6.99, 11, 'доступен'), (4.99, 12, 'доступен'),  
(4.99, 13, 'доступен'), (2.99, 3, 'доступен');
```

На рисунке 2.3 представлен результат заполнения таблицы disk.

Query Query History

1 select * from disk;

Data Output Messages Notifications

	id [PK] integer	rental_cost numeric (10,2)	quantity integer	state character varying (20)
1	1	5.99	10	доступен
2	2	4.99	8	доступен
3	3	6.99	12	доступен
4	4	3.99	15	доступен
5	5	7.99	7	доступен
6	6	5.99	9	доступен
7	7	6.99	11	доступен
8	8	4.99	10	доступен
9	9	6.99	8	доступен
10	10	5.99	14	доступен
11	11	7.99	10	доступен
12	12	4.99	12	доступен
13	13	5.99	9	доступен
14	14	6.99	11	доступен
15	15	4.99	13	доступен
16	16	7.99	8	доступен
17	17	5.99	15	доступен
18	18	6.99	9	доступен
19	19	4.99	11	доступен
20	20	5.99	10	доступен
21	21	6.99	12	доступен
22	22	3.99	14	доступен
23	23	7.99	8	доступен
24	24	5.99	10	доступен
25	25	6.99	11	доступен
26	26	4.99	9	доступен
27	27	5.99	13	доступен
28	28	7.99	8	доступен
29	29	4.99	12	доступен
30	30	2.99	3	доступен

Total rows: 30 of 30 Query complete 00:00:00.101 Ln 1, Col 20

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),
 ('Мemento', '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),
 ('Спаси 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.

Query Query history

1 select * from film;

Data Output Messages Notifications

	id [PK] integer	name character varying (255)	year date	genre character varying (100)	director_id integer
1	1	Начало	2010-07-16	фантастика	1
2	2	Темный рыцарь	2008-07-18	боевик	1
3	3	Властелин колец: Возвращение короля	2003-12-17	фэнтези	7
4	4	Интерстеллар	2014-11-07	фантастика	1
5	5	Джентльмены	2019-12-03	криминал	12
6	6	Бегущий по лезвию	1982-06-25	фантастика	4
7	7	Побег из Шоушенка	1994-10-14	драма	2
8	8	Крестный отец	1972-03-24	криминал	6
9	9	Список Шиндлера	1993-12-15	биография	6
10	10	Парк Юрского периода	1993-06-11	приключения	7
11	11	Пятый элемент	1997-05-07	фантастика	10
12	12	Красота по-американски	1999-09-15	драма	11
13	13	Таксист	1976-02-08	триллер	9
14	14	Мemento	2000-10-11	триллер	1
15	15	Достучаться до небес	1997-10-30	комедия	14
16	16	Семь	1995-09-22	триллер	5
17	17	Искусственный разум	2001-06-26	фантастика	3
18	18	Плотина	1975-11-04	драма	15
19	19	Изгой	2000-12-22	драма	16
20	20	Зеленая миля	1999-12-06	фэнтези	2
21	21	Шоу Трумана	1998-06-05	комедия	8
22	22	Шестое чувство	1999-08-06	триллер	17
23	23	Гарри Поттер и философский камень	2001-11-16	фэнтези	18
24	24	Безумный Макс: Дорога ярости	2015-05-14	боевик	19
25	25	Сонная лощина	1999-04-21	драма	20
26	26	По социальным стандартам	1980-09-19	драма	21
27	27	Вечное сияние чистого разума	2004-03-19	романтика	22
28	28	Сласти рядового Райана	1998-07-24	военный	23
29	29	Волк с Уолл-стрит	2013-12-25	криминал	24
30	30	Красный воробей	2018-02-15	триллер	25

Total rows: 30 of 30 Query complete 00:00:00.075 Ln 1, Col 19

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),
```


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`.

Query Query History

1 `select * from review;`

Data Output Messages Notifications

	id [PK] integer	mark integer	comment text	film_id integer	client_id integer
1	1	5	Отличный фильм, сюжет завораживает!	1	1
2	2	4	Неплохой боевик, но слишком много сцен на грани фантастики.	2	2
3	3	5	Эпическое завершение трилогии, прекрасное кино!	3	3
4	4	4	Интересный фильм, но иногда сложно понять смысл.	4	4
5	5	3	Средний криминальный фильм, ничего особенного.	5	5
6	6	5	Шедевр на века!	6	6
7	7	5	Лучший фильм всех времен!	7	7
8	8	4	Иконический криминальный фильм, но не для слабонервных.	8	8
9	9	5	Великолепное кино, красивая история.	9	9
10	10	4	Захватывающее путешествие по острову динозавров.	10	10
11	11	5	Фантастический мир будущего, яркие образы и захватывающий сюжет!	11	11
12	12	3	Не для каждого, но все же интересно.	12	12
13	13	5	Отличный триллер, волнующий до самого конца.	13	13
14	14	4	Позитивное кино, поднимает настроение!	14	14
15	15	5	Увлекательный и стильный триллер.	15	15
16	16	4	Философский фильм с красивой картинкой.	16	16
17	17	5	Одна из лучших кинолент в истории.	17	17
18	18	3	Сложно понять суть, но фильм заставляет задуматься.	18	18
19	19	5	Завораживающая фантастика.	19	19
20	20	4	Отличное драматическое кино, оставляет многое на размышление.	20	20
21	21	5	Захватывающая история, полная неожиданных поворотов.	21	21
22	22	4	Шикарная комедия с интересным сюжетом.	22	22
23	23	5	Необычный триллер, который заставляет пересматривать фильмы.	23	23
24	24	4	Мир магии и приключений, знакомство с героями не оставляет равнодушны...	24	24
25	25	5	Отличная драма, сильная игра актеров.	25	25
26	26	3	Тяжелый фильм о жизни, который заставляет задуматься.	26	26
27	27	5	Фантастическая картина, которую стоит посмотреть каждому.	27	27
28	28	4	Захватывающий военный фильм, который тронет душу.	28	28
29	29	5	Живое и реалистичное кино о финансовом мире.	29	29
30	30	5	Неплохой фильм!	4	1

Total rows: 30 of 30 Query complete 00:00:00.115 Ln 1, Col 15

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.

	id [PK] integer	film_id integer	disk_id integer
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	7	7
8	8	8	8
9	9	9	9
10	10	10	10
11	11	11	11
12	12	12	12
13	13	13	13
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	21	21	21
22	22	22	22
23	23	23	23
24	24	24	24
25	25	25	25
26	26	26	26
27	27	27	27
28	28	28	28
29	29	29	29
30	30	30	30

Total rows: 30 of 30 Query complete 00:00:00.045

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), (11, 11), (21, 21),  
(2, 2), (12, 12), (22, 22),  
(3, 3), (13, 13), (23, 23),  
(4, 4), (14, 14), (24, 24),  
(5, 5), (15, 15), (25, 25),  
(6, 6), (16, 16), (26, 26),  
(7, 7), (17, 17), (27, 27),  
(8, 8), (18, 18), (28, 28),  
(9, 9), (19, 19), (29, 29),  
(10, 10), (20, 20), (30, 30);
```

На рисунке 2.8 представлен результат заполнения таблицы order+disk.

Query Query History

1 `select * from "order+disk";`

Data Output Messages Notifications

	id [PK] integer	order_id integer	disk_id integer
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	7	7
8	8	8	8
9	9	9	9
10	10	10	10
11	11	11	11
12	12	12	12
13	13	13	13
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	21	21	21
22	22	22	22
23	23	23	23
24	24	24	24
25	25	25	25
26	26	26	26
27	27	27	27
28	28	28	28
29	29	29	29
30	30	30	30

Total rows: 30 of 30 Query complete 00:00:00.102 Ln 1, Col 28

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.

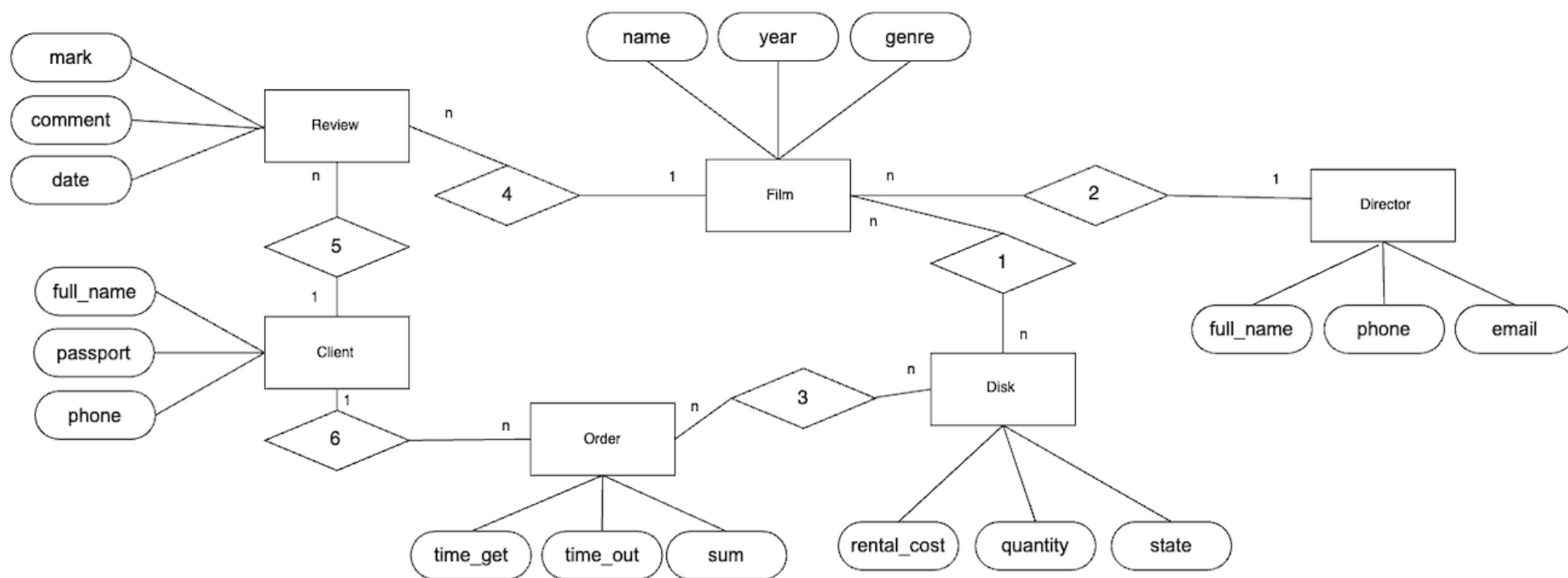


Рисунок 2.10 – ER-диаграмма модели «Прокат видеодисков»

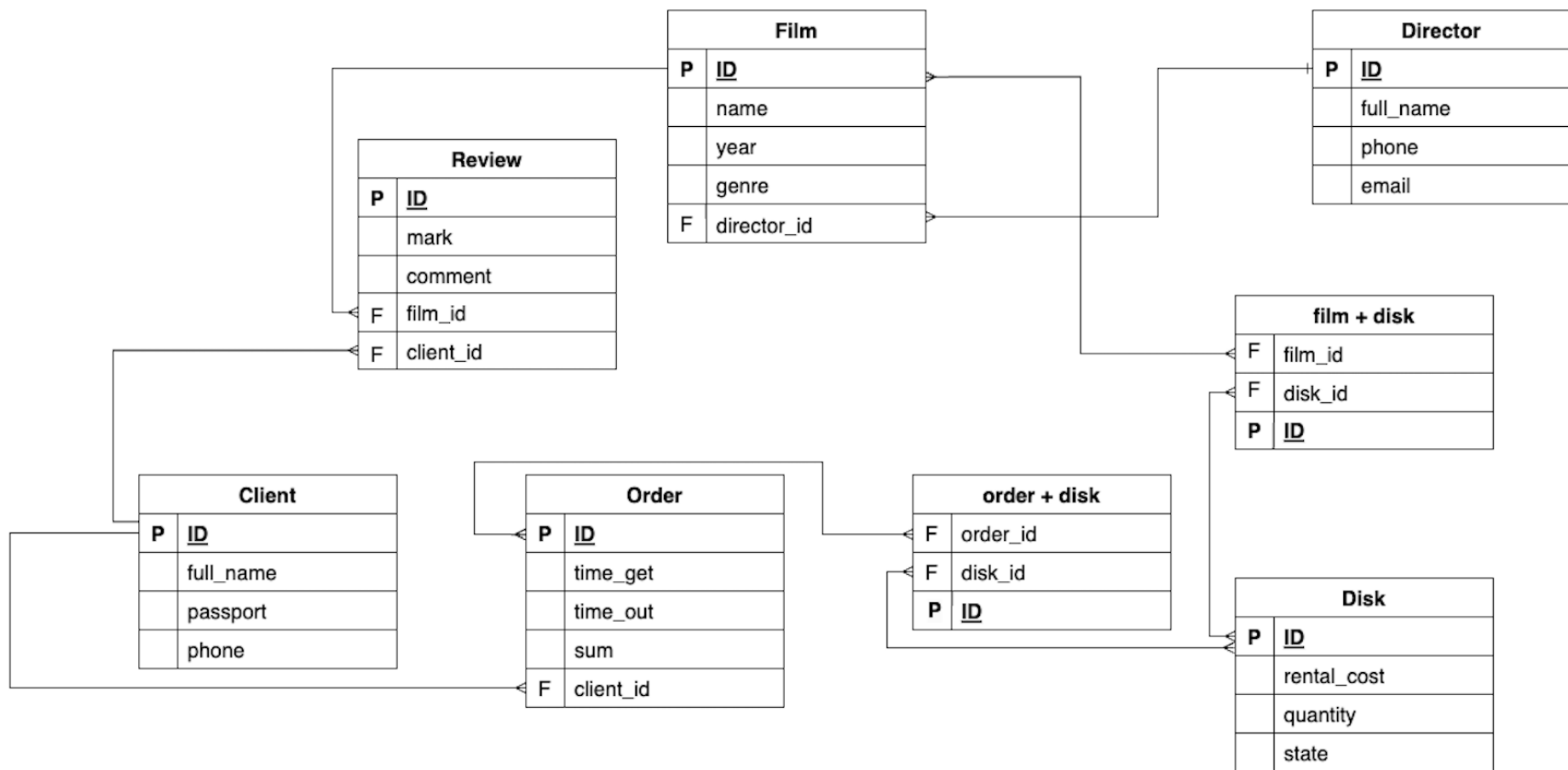


Рисунок 2.11 – UML-диаграмма модели «Прокат видеодисков»

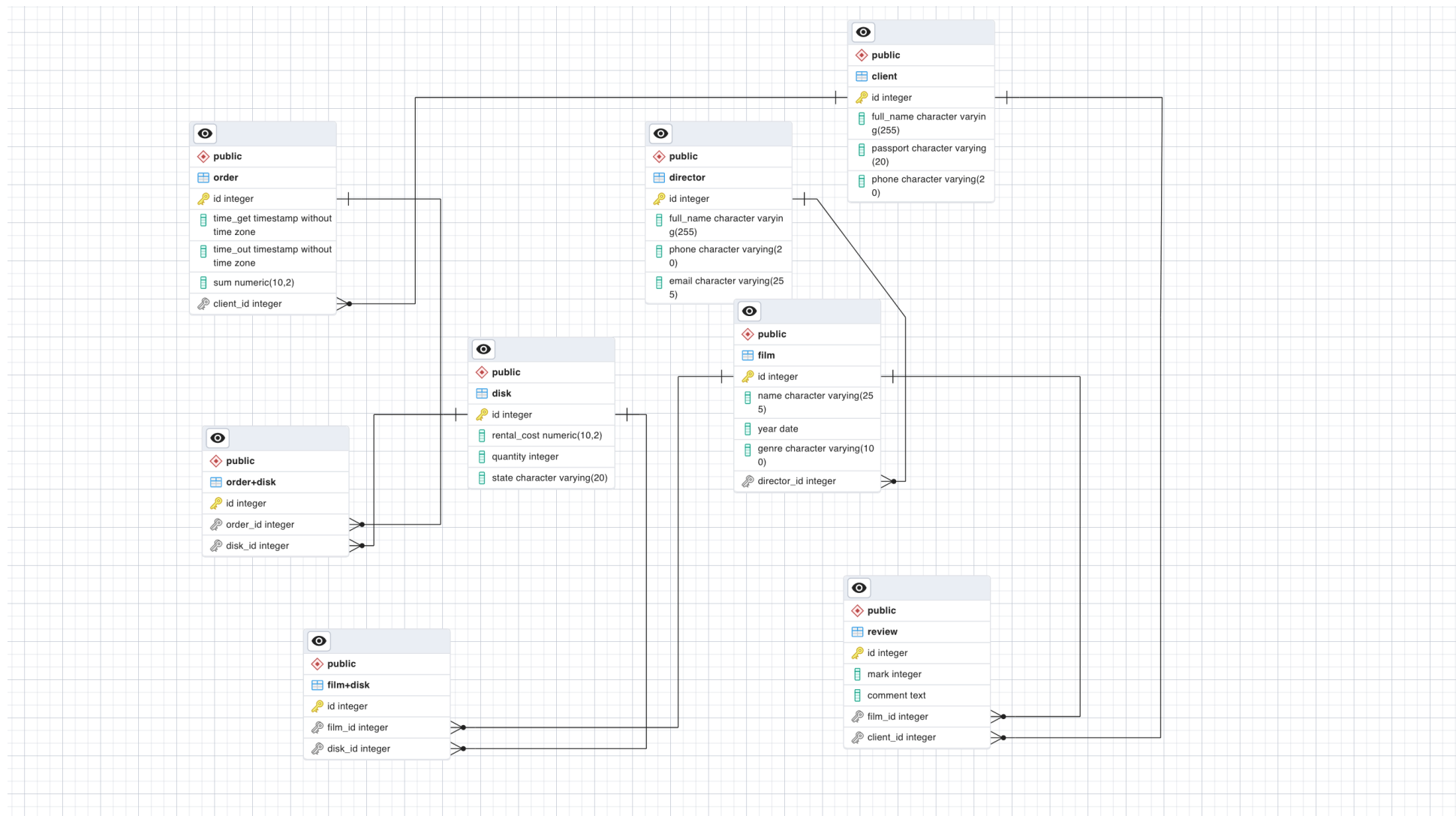


Рисунок 2.12 – ERD-диаграмма модели «Прокат видеодисков» из pgAdmin

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;
```