

Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный университет информатики и
радиоэлектроники»

Кафедра электронных вычислительных машин

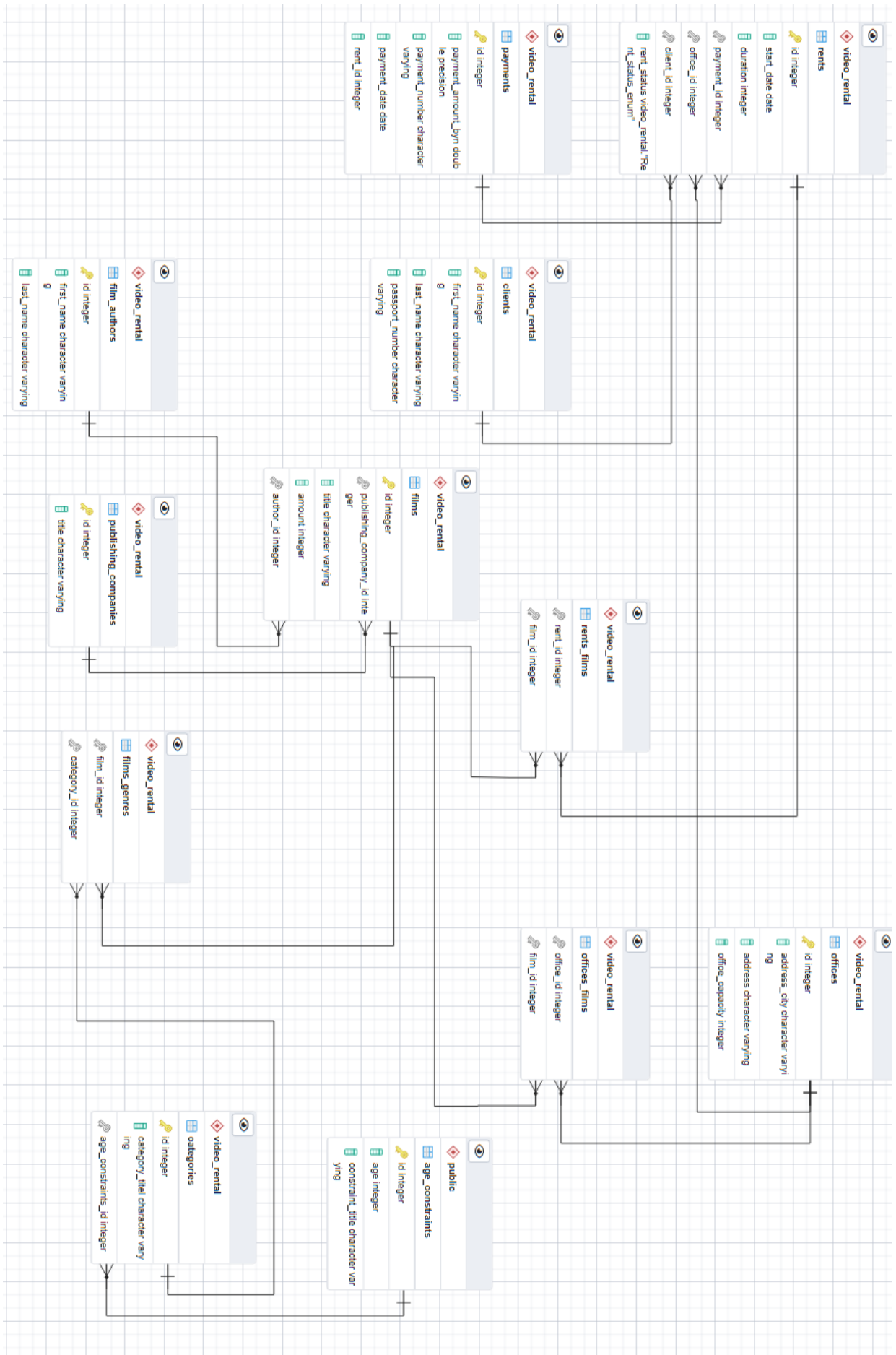
Лабораторная работа №4
«Реализация SQL-запросов на простую выборку данных»
(Видеопрокат)

Выполнили
студенты группы 950503:
Зарубо Д. Ю.
Яценко В. П.

Проверила:
Куприянова Д.В.

Минск 2022

ER-диаграмма «Видеопрокат»



1. Оператор SELECT

Оператор языка SQL SELECT - основная строительная конструкция для создания любого, простого или сложного запроса к базе данных. Без него, как и без фундамента для постройки, невозможно получить ни одну выборку данных из базы.

Оператор SELECT состоит из нескольких предложений (разделов), рассмотрим основной из них:

- SELECT определяет список возвращаемых столбцов (как существующих, так и вычисляемых), их имена, ограничения на уникальность строк в возвращаемом наборе, ограничения на количество строк в возвращаемом наборе, реализует операцию проекции, то есть указание подмножества столбцов из таблиц табличного выражения, а также операцию переименования столбцов и операцию добавления новых вычисляемых столбцов.

Можно выбрать все данные из таблицы:

```
Select * from video_rental.categories;
```

Рисунок 1. Выборка всех данных из таблицы

	id [PK] integer	title character varying	age_constraints_id integer
1	1	action	3
2	2	western	3
3	3	gangster movie	4
4	4	detective	3
5	5	drama	3
6	6	historical film	3
7	7	comedy	2
8	8	melodrama	3
9	9	musical film	2
10	10	noir	4
11	11	political film	3
12	12	adventure movie	2
13	13	fairy story	1
14	14	tragedy	3
15	15	tragicomedy	3
16	16	thriller	3
17	17	fantasy movie	2
18	18	horror	5

Рисунок 2. Результат выполнения команды

Можно выбрать только интересующие нас столбцы:

```
1 Select title, age_constraints_id from video_rental.categories;
```

Рисунок 3. Выборка конкретных данных из таблицы

	title character varying	age_constraints_id integer
1	action	3
2	western	3
3	gangster movie	4
4	detective	3
5	drama	3
6	historical film	3
7	comedy	2
8	melodrama	3
9	musical film	2
10	noir	4
11	political film	3
12	adventure movie	2
13	fairy story	1
14	tragedy	3
15	tragicomedy	3
16	thriller	3
17	fantasy movie	2
18	horror	5

Рисунок 4. Результат выполнения команды

Теперь сделаем выборку из таблицы films, содержащую все поля. Критерий выборки – цена аренды фильма price превышает 23. Записи отсортированы по полю price

```
Query Editor  Query History
1 Select * from video_rental.films
2 Where price > 23
3 Order By price
```

Рисунок 5. Выборка конкретных данных из таблицы

	id [PK] integer	publishing_company_id integer	title character varying	author_id integer	price numeric (5,3)
1	24	3	Начало	10	24.000
2	29	17	Побег из Шоушенка	12	24.000
3	18	2	Сонная Лощина	8	25.000
4	3	7	Бегущий по лезвию 2049	16	25.000
5	7	8	Шерлок Холмс	4	26.000
6	9	6	Карты, деньги, два ствола	4	27.000
7	23	3	Престиж	10	28.000
8	22	3	Темный рыцарь	10	30.000
9	21	3	Гран Торино	9	30.000
10	11	5	Криминальное чтиво	3	30.000
11	16	10	Терминатор 2: Судный день	6	33.000

Рисунок 6. Результат выполнения команды

В данном примере мы сделаем выборку из таблицы “films” записей, у которых номер компании-производителя четный.

```

1 Select * from video_rental.films
2 Where publishing_company_id % 2 = 0
3 Order By publishing_company_id desc

```

Рисунок 7. Выборка конкретных данных из таблицы

	id [PK] integer	publishing_company_id integer	title character varying	author_id integer	price numeric (5,3)
1	1	18	Психо	15	15.000
2	26	16	Семь	11	17.000
3	28	12	Мажестик	12	21.000
4	6	12	Список Шиндлера	1	19.000
5	16	10	Терминатор 2: Судный день	6	33.000
6	7	8	Шерлок Холмс	4	26.000
7	9	6	Карты, деньги, два ствола	4	27.000
8	10	4	Бешеные псы	3	19.000
9	12	2	Остров проклятых	2	17.000
10	18	2	Сонная Лощина	8	25.000
11	4	2	Форрест Гамп	13	17.000

Рисунок 8. Результат выполнения команды

Тут выполним выборку из таблицы payments записей, где сумма аренды больше 30 и меньше 50.

```

1 Select * from video_rental.payments
2 Where amount_byn > 30 and amount_byn < 50

```

Рисунок 9. Выборка конкретных данных из таблицы

	id [PK] integer	amount_byn double precision	number integer	date date	rent_id integer
1	2	40	2	2022-10-21	137
2	5	38	5	2022-10-25	140
3	7	36	7	2022-10-25	142
4	8	39	8	2022-10-28	143
5	10	46	10	2022-11-05	145
6	11	40	11	2022-11-07	146
7	12	35	12	2022-11-07	147

Рисунок 10. Результат выполнения команды

В данном примере мы сделаем выборку из таблицы “clients” записей, где имя клиента начинается с буквы М.

```

1 Select * from video_rental.clients
2 Where first_name like 'M%'

```

Рисунок 11. Выборка конкретных данных из таблицы

Data Output

Explain

Messages

Notifications

	id [PK] integer	first_name character varying	last_name character varying	passport_number character varying
1	12	Макаров	Сергей	12LDL86
2	13	Маликова	Кристина	87LDK34
3	14	Маркелова	Кристина	98KIJ66

Рисунок 12. Результат выполнения команды

Тут сделаем выборку из таблицы rents, где аренда длится больше 5 дней и фильм был выдан в отделении с четным номером.

```

Select * from video_rental.rents
Where duration > 5 and office_id %2 = 0

```

Рисунок 13. Выборка конкретных данных из таблицы

	id [PK] integer	start_date date	duration integer	payment_id integer	office_id integer	client_id integer	rental_day_price numeric (5,3)	status video_rental.rent_status_enum
1	136	2022-10-20	10	1	2	1	1.000	RETURNED
2	139	2022-10-25	10	4	6	8	1.000	ISSUED
3	142	2022-10-25	15	7	12	11	1.000	ISSUED
4	147	2022-11-07	15	12	4	1	1.000	ISSUED

Рисунок 14. Результат выполнения команды

Выберем из таблицы `age_constraints` названия возрастных ограничений, разрешенный возраст у которых ниже 16, которые начинаются на букву 'P' и выведем в обратном порядке.

```
1 Select constraint_title from video_rental.age_constraints
2 where age < 16 and constraint_title like 'P%'
```

Рисунок 15. Выборка конкретных данных из таблицы

	constraint_title character varying
1	PG-13
2	PG

Рисунок 16. Результат выполнения команды

Предложение WHERE

Оператор SQL WHERE служит для задания дополнительного условия выборки, операций вставки, редактирования и удаления записей. Предложение `[[WHERE (SQL)|WHERE]]` используется для определения, какие строки должны быть выбраны из табличного выражения в предложении FROM. Мы можем описывать несколько условий выборки:

```
Select * from video_rental.film_authors
Where first_name = 'Клинт' and last_name = 'Иствуд'
```

Рисунок 17. Выборка конкретных данных из таблицы

	id [PK] integer	first_name character varying	last_name character varying
1	9	Клинт	Иствуд

Рисунок 18. Результат выполнения команды

Мы выполнили выборку из таблицы режиссеров фильмов, причем искали режиссера с конкретным именем и фамилией.

Предложение ORDER BY

ORDER BY — необязательное предложение операторов SELECT и UNION, который означает что операторы SELECT, UNION возвращают набор строк, отсортированных по значениям одного или более столбцов. Его можно применять как к числовым столбцам, так и к строковым. В последнем случае, сортировка будет происходить по алфавиту. Использование предложения ORDER BY является единственным способом отсортировать результирующий набор строк. Без этого предложения СУБД может вернуть строки в любом порядке. Если упорядочение необходимо, ORDER BY должен присутствовать в SELECT, UNION. Сортировка может производиться как по возрастанию, так и по убыванию значений.

- Параметр ASC (по умолчанию) устанавливает порядок сортировки по возрастанию, от меньших значений к большим.
- Параметр DESC устанавливает порядок сортировки по убыванию, от больших значений к меньшим

Выберем клиентов из таблицы clients по возрастанию их имен, но по убыванию фамилий

```
1 Select * from video_rental.clients
2 Order By last_name asc, first_name desc
```

Рисунок 17. Выборка конкретных данных из таблицы

	id [PK] integer	first_name character varying	last_name character varying	passport_number character varying
1	1	Аксенова	Александра	AA0001LK
2	2	Андреев	Глеб	PO859ER
3	6	Голубев	Даниил	93LMK11
4	9	Зайцева	Дина	96MDL45
5	11	Котов	Евгений	11HHH40
6	15	Никитин	Егор	66OOO99
7	5	Виноградов	Егор	NJ123PO
8	22	Быкова	Елена	99AAB99
9	3	Быков	Иван	KM348NC
10	10	Иванова	Инна	94JDO66
11	14	Маркелова	Кристина	98KIJ66
12	13	Маликова	Кристина	87LDK34
13	7	Голубь	Мирон	55JDU56
14	16	Семёнов	Михаил	23KDM34
15	8	Дёмин	Мурат	95MDE44
16	4	Васильев	Николай	PL983J
17	12	Макаров	Сергей	12LDL86
18	20	Юсупов	Степан	00JDJ85
19	17	Сова	Степан	96KF045
20	21	Андреева	Юлия	76KFK77
21	18	Холод	Юрий	44JDN66
22	19	Царь	Ярослав	95KMK22

Рисунок 18. Результат выполнения команды

CROSS JOIN

Оператор SQL CROSS JOIN формирует таблицу перекрестным соединением (декартовым произведением) двух таблиц. При использовании оператора SQL CROSS JOIN каждая строка левой таблицы сцепляется с каждой строкой правой таблицы. В результате получается таблица со всеми возможными сочетаниями строк обеих таблиц.

Сделаем пересечение таблиц offices и films

```
Select * from video_rental.offices
Cross join video_rental.films
```

Рисунок 19. Выборка конкретных данных из таблицы

1	1	спуск Гоголя, 25	Брест	1	18	Психо
2	2	ул. Балканская, 62	Брест	1	18	Психо
3	3	[...] наб. Будапештская, 34	Гомель	1	18	Психо
4	4	пер. Домодедовская, 75	Минск	1	18	Психо
5	5	[...] проезд Чехова, 14	Гомель	1	18	Психо
6	6	ул. Славы, 72	Гомель	1	18	Психо
7	7	проспект Независимости, 33	Минск	1	18	Психо
8	8	проспект Космонавтов, 19	Минск	1	18	Психо
9	9	ул. Ломоносова	Минск	1	18	Психо
10	10	ул. Никитина	Гомель	1	18	Психо
11	11	ул. Кальварийская	Минск	1	18	Психо
12	12	пер. Шосейный	Гомель	1	18	Психо
13	13	ул. Нижняя	Брест	1	18	Психо
14	14	ул. Переводов	Минск	1	18	Психо
15	15	проспект Мира	Брест	1	18	Психо
16	1	спуск Гоголя, 25	Брест	2	3	В случае убийства набирайте «М»
17	2	ул. Балканская, 62	Брест	2	3	В случае убийства набирайте «М»
18	3	[...] наб. Будапештская, 34	Гомель	2	3	В случае убийства набирайте «М»
19	4	пер. Домодедовская, 75	Минск	2	3	В случае убийства набирайте «М»
20	5	[...] проезд Чехова, 14	Гомель	2	3	В случае убийства набирайте «М»
21	6	ул. Славы, 72	Гомель	2	3	В случае убийства набирайте «М»
22	7	проспект Независимости, 33	Минск	2	3	В случае убийства набирайте «М»

Рисунок 20. Часть результирующей таблицы

INNER JOIN

Объединяет записи из двух таблиц, если в связующих полях этих таблиц содержатся одинаковые значения. Операцию INNER JOIN можно использовать в любом предложении FROM. Это самый распространенный тип объединения. С его помощью происходит объединение записей из двух таблиц по связующему полю, если оно содержит одинаковые значения в обеих таблицах.

```
Select * from video_rental.payments p
left join video_rental.rents r
on r.id = p.rent_id
```

Рисунок 21. Выборка конкретных данных из таблицы

	id integer	amount_byn double precision	number integer	date date	rent_id integer	id integer	start_date date	duration integer	payment_id integer	office_id integer	client_id integer	rental_day_price numeric (5,3)	status video_rental.rent_status_с
1	1	15	1	2022-10-20	136	136	2022-10-20	10	1	2	1	1.000	RETURNED
2	2	40	2	2022-10-21	137	137	2022-10-21	5	2	3	5	1.000	RETURNED
3	3	58	3	2022-10-21	138	138	2022-10-21	30	3	7	3	1.000	ISSUED
4	4	58	4	2022-10-25	139	139	2022-10-25	10	4	6	8	1.000	ISSUED
5	5	38	5	2022-10-25	140	140	2022-10-25	15	5	1	7	1.000	ISSUED
6	6	24	6	2022-10-22	141	141	2022-10-22	5	6	8	3	1.000	RETURNED
7	7	36	7	2022-10-25	142	142	2022-10-25	15	7	12	11	1.000	ISSUED
8	8	39	8	2022-10-28	143	143	2022-10-28	3	8	4	13	1.000	ISSUED
9	9	21	9	2022-11-03	144	144	2022-11-03	10	9	7	8	1.000	ISSUED
10	10	46	10	2022-11-05	145	145	2022-11-05	3	10	3	2	1.000	RETURNED
11	11	40	11	2022-11-07	146	146	2022-11-07	5	11	2	14	1.000	RETURNED
12	12	35	12	2022-11-07	147	147	2022-11-07	15	12	4	1	1.000	ISSUED
13	13	30	13	2022-11-05	148	148	2022-11-05	3	13	8	21	1.000	RETURNED
14	14	25	14	2022-11-10	149	149	2022-11-10	5	14	1	9	1.000	RETURNED
15	15	23	15	2022-11-10	150	150	2022-11-10	5	15	6	1	1.000	ISSUED

Рисунок 22. Результат выполнения команды Inner Join

OUTER JOIN

Внешние соединения - это соединения, которые возвращают совпадающие

значения и несопоставимые значения из одной или обеих таблиц. Существует несколько типов внешних соединений:

- левое внешнее соединение (LEFT OUTER JOIN) - выборка будет содержать все строки из первой или левой таблицы;
- правое внешнее соединение (RIGHT OUTER JOIN) - выборка будет содержать все строки из второй или правой таблицы;
- полное внешнее соединение (FULL OUTER JOIN) - выборка будет содержать все строки из обеих таблиц.

Для примера воспользуемся FULL OUTER JOIN:

Сделаем left join над таблицами films и publishing_companies

```
Select * from video_rental.films f
left join video_rental.publishing_companies p
on f.publishing_company_id = p.id
```

Рисунок 23. Выборка конкретных данных из таблицы

	id integer	publishing_company_id integer	title character varying	author_id integer	price numeric (5,3)	id integer	title character varying
1	1	[null]	Психо	15	15.000	18	[null]
2	2	3	В случае убийства набирайте «М»	15	23.000	3	Warner Bros
3	3	7	Бегущий по лезвию 2049	16	25.000	7	Columbia Pictures
4	4	2	Форрест Гамп	13	17.000	2	Paramount
5	5	1	Поймай меня, если сможешь	1	14.000	1	DreamWorks
6	6	12	Список Шиндлера	1	19.000	12	Universal
7	7	8	Шерлок Холмс	4	26.000	8	Silver Pictures
8	8	7	Большой куш	4	22.000	7	Columbia Pictures
9	9	6	Карты, деньги, два ствола	4	27.000	6	Handmade Films
10	10	4	Бешеные псы	3	19.000	4	Dog Eat Dog Productions
11	11	5	Криминальное чтиво	3	30.000	5	Miramax
12	12	2	Остров проклятых	2	17.000	2	Paramount
13	13	3	Отступники	2	23.000	3	Warner Bros
14	14	1	Гладиатор	7	22.000	1	DreamWorks
15	15	11	Титаник	6	18.000	11	20th Century Fox
16	16	10	Терминатор 2: Судный день	6	33.000	10	Pacific Western
17	17	9	Манхэттен	5	20.000	9	Jack Rollins & Charles H. Joffe Productions
18	18	2	Сонная Лощина	8	25.000	2	Paramount
19	19	7	Крупная рыба	8	23.000	7	Columbia Pictures
20	20	13	Винсент	8	21.000	13	Walt Disney
21	21	3	Гран Торино	9	30.000	3	Warner Bros
22	22	3	Темный рыцарь	10	30.000	3	Warner Bros
23	23	3	Престиж	10	28.000	3	Warner Bros
24	24	3	Начало	10	24.000	3	Warner Bros
25	25	7	Социальная сеть	11	20.000	7	Columbia Pictures
26	26	16	Семь	11	17.000	16	New Line Cinema
27	27	15	Бойцовский клуб	11	20.000	15	Taurus Film
28	28	12	Мажестик	12	21.000	12	Universal
29	29	17	Побег из Шоушенка	12	24.000	17	Castle Rock Entertainment

Рисунок 24. Результат выполнения команды Left Join

СКАЛЯРНЫЕ ФУНКЦИИ

Скалярные функции Transact-SQL используются в создании скалярных выражений. (Скалярная функция выполняет вычисления над одним значением или списком значений, тогда как агрегатная функция выполняет вычисления над группой значений из нескольких строк.) Скалярные функции можно разбить на следующие категории:

- числовые функции;
- функции даты;
- строковые функции;
- системные функции;
- функции метаданных.

Выполним подсчет общей стоимости аренды всех фильмов

```
Select SUM(f.price) from video_rental.films f
```

Рисунок 25. Выборка конкретных данных из таблицы

	sum numeric
1	653.000

Рисунок 26. Результат подсчета суммы

Некоторые другие функции

Query	Query History
1	<code>SELECT bit_length ('Spark SQL');</code>
2	<code>SELECT CONCAT ('BSUIR', '.com');</code>
3	<code>SELECT ROUND (537.627, 3) AS RoundValue;</code>

Рисунок 27. Пример функций

Рисунок 27. Примеры использования скалярных функций

	bit_length integer
1	72

	concat text
1	BSUIR.com

	roundvalue numeric
1	537.627

Рисунок 27. Результат функций