

Министерство образования Республики Беларусь

Учреждение образования  
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И  
РАДИОЭЛЕКТРОНИКИ»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №2 по дисциплине «База данных»

Реализация SQL-запросов для создания схемы базы данных

Студента-заочника 3 курса  
группы №250541  
Власов Р. Е.

Проверила: магистр  
технических наук  
Куприянова Д. В.

МИНСК 2024

## Цель работы

Целью работы является создание запросов на выборку данных на языке SQL с использованием предложений SELECT, FROM (JOINS), WHERE, GROUP BY и ORDER BY оператора SELECT, а также подзапросов, скалярных и агрегатных функций.

## Результаты работы

1) Вывести список всех сотрудников и их должностей вместе с номерами телефонов, отсортированный по ФИО сотрудника.

```
SELECT Номер_сотрудника, ФИО, Должность, Телефон
FROM "Кинотеатр"."Сотрудник"
ORDER BY ФИО ASC;
```

1	▼	SELECT	Номер_сотрудника, ФИО, Должность, Телефон
2		FROM	"Кинотеатр"."Сотрудник"
3		ORDER BY	ФИО ASC;
4			

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑

🗑

📄

⬇

📈

SQL

	Номер_сотрудника [PK] integer	ФИО character varying (255)	Должность character varying (50)	Телефон character varying (50)
1	17	Васильев Андрей Николаевич	Администратор	+375-29-901-23-45
2	7	Васильев Сергей Михайлович	Администратор	+7-918-901-23-45
3	20	Григорьев Алексей Сергеевич	Бухгалтер	+375-29-234-56-78
4	10	Григорьева Светлана Ивановна	Бухгалтер	+7-921-234-56-78
5	1	Иванов Иван Иванович	Менеджер	+7-912-345-67-89
6	14	Иванова Наталья Сергеевна	Уборщик	+375-29-678-90-12
7	16	Козлова Марина Дмитриевна	Менеджер	+375-29-890-12-34
8	13	Кузнецов Николай Александрович	Техник	+375-29-567-89-01
9	4	Кузнецова Елена Владимировна	Уборщик	+7-915-678-90-12
10	9	Лебедев Павел Юрьевич	Уборщик	+7-920-123-45-67
11	18	Лебедева Елизавета Петровна	Техник	+375-29-012-34-56
12	5	Морозова Ольга Николаевна	Бухгалтер	+7-916-789-01-23
13	8	Николаева Анна Викторовна	Техник	+7-919-012-34-56
14	11	Новиков Сергей Иванович	Менеджер	+375-29-345-67-89
15	2	Петрова Мария Сергеевна	Администратор	+7-913-456-78-90
16	3	Сидоров Алексей Петрович	Техник	+7-914-567-89-01
17	15	Смирнов Павел Викторович	Бухгалтер	+375-29-789-01-23
18	12	Соколова Ольга Петровна	Администратор	+375-29-456-78-90
19	6	Федоров Дмитрий Александрович	Менеджер	+7-917-890-12-34
20	19	Федоров Максим Иванович	Уборщик	+375-29-123-45-67

2) Найти сеансы, у которых время начала после 18:00:00 и количество свободных мест меньше 100.

```
SELECT Номер_сеанса, Время_начала, Свободные_места
FROM "Кинотеатр"."Сеанс"
WHERE Время_начала > '18:00:00' AND Свободные_места < 100;
```

```
1  SELECT Номер_сеанса, Время_начала, Свободные_места
2  FROM "Кинотеатр"."Сеанс"
3  WHERE Время_начала > '18:00:00' AND Свободные_места < 100;
```

Data Output Messages Notifications

	Номер_сеанса [PK] integer	Время_начала time without time zone	Свободные_места integer
1	7	22:30:00	80
2	13	19:00:00	90
3	15	23:00:00	90

3) Подсчитать количество сеансов в каждом зале, сгруппировать по Номер\_зала и отсортировать по количеству сеансов по убыванию.

```
SELECT Номер_сеанса, Время_начала, Свободные_места
FROM "Кинотеатр"."Сеанс"
WHERE Время_начала > '18:00:00' AND Свободные_места < 100;
```

```
1  SELECT Номер_зала, COUNT(Номер_сеанса) AS Количество_сеансов
2  FROM "Кинотеатр"."Сеанс"
3  GROUP BY Номер_зала
4  ORDER BY Количество_сеансов DESC;
```

Data Output Messages Notifications

	Номер_зала integer	Количество_сеансов bigint
1	3	3
2	5	3
3	4	3
4	2	3
5	1	3

4) Вывести средний рейтинг фильмов, сгруппировать по Номер\_фильма и отсортировать по убыванию среднего рейтинга.

```
SELECT Номер_фильма, AVG(Рейтинг) AS Средний_рейтинг
FROM "Кинотеатр"."Отзыв"
GROUP BY Номер_фильма
ORDER BY Средний_рейтинг DESC;
```

1	SELECT	Номер_фильма	,	AVG	(Рейтинг)	AS	Средний_рейтинг
2	FROM	"Кинотеатр".	"Отзыв"				
3	GROUP BY	Номер_фильма					
4	ORDER BY	Средний_рейтинг	DESC	;			

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑

🗄

⬇

📈

SQL

	Номер_фильма integer	Средний_рейтинг double precision
1	6	5
2	2	5
3	7	4.9
4	10	4.8
5	3	4.75
6	8	4.6
7	5	4.5500000000000001
8	1	4.5
9	4	4.35
10	9	4.3

5) Подсчитать количество отзывов с рейтингом выше 4.5 для каждого фильма, отсортировать по убыванию количества отзывов.

```
SELECT Номер_фильма, COUNT(Номер_отзыва) AS
Количество_высоких_оценок
FROM "Кинотеатр"."Отзыв"
WHERE Рейтинг > 4.5
GROUP BY Номер_фильма
ORDER BY Количество_высоких_оценок DESC;
```

1	SELECT	Номер_фильма,	COUNT(Номер_отзыва)	AS	Количество_высоких_оценок
2	FROM	"Кинотеатр"."Отзыв"			
3	WHERE	Рейтинг	>	4.5	
4	GROUP BY	Номер_фильма			
5	ORDER BY	Количество_высоких_оценок	DESC;		

Data Output	Messages	Notifications
-------------	----------	---------------

≡+	📄	▼	📋	▼	🗑	🗄	⬇	📈	SQL
----	---	---	---	---	---	---	---	---	-----

	Номер_фильма integer	Количество_высоких_оценок bigint
1	3	2
2	7	2
3	6	2
4	10	2
5	8	2
6	2	2
7	5	1

6) Подсчитать общее количество мест и количество электронных мест в каждом зале, отсортировав по возрастанию номера зала.

```
SELECT Номер_зала,
       COUNT(Номер_сидения) AS Общее_количество_мест,
       SUM(CASE WHEN Является_электронным = true THEN 1 ELSE 0 END) AS
       Электронные_места
FROM "Кинотеатр"."Место"
GROUP BY Номер_зала
ORDER BY Номер_зала;
```

1	SELECT	Номер_зала,		
2		COUNT(Номер_сидения)	AS	Общее_количество_мест,
3		SUM(CASE WHEN Является_электронным = true THEN 1 ELSE 0 END)	AS	Электронные_места
4	FROM	"Кинотеатр"."Место"		
5	GROUP BY	Номер_зала		
6	ORDER BY	Номер_зала;		

Data Output	Messages	Notifications
-------------	----------	---------------

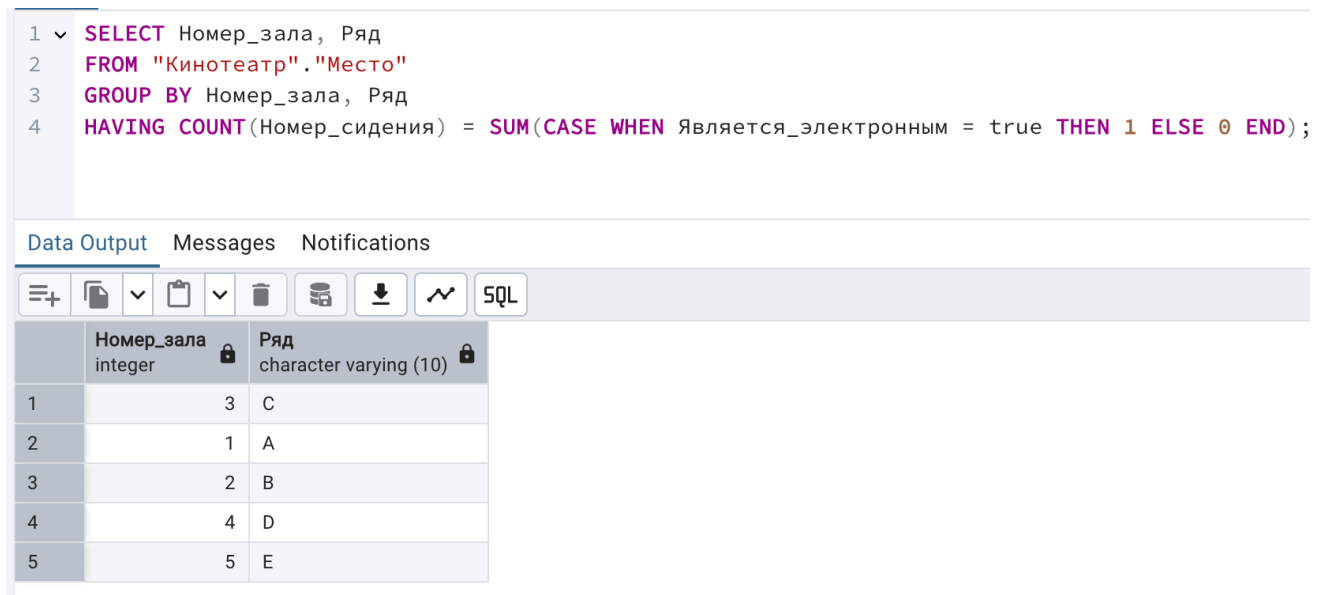
≡+	📄	▼	📋	▼	🗑	🗄	⬇	📈	SQL
----	---	---	---	---	---	---	---	---	-----

	Номер_зала integer	Общее_количество_мест bigint	Электронные_места bigint
1	1	10	10
2	2	10	10
3	3	10	10
4	4	10	10
5	5	10	10

7) Определить ряды, в которых все места являются электронными, и вывести номера залов и рядов.

```
SELECT Номер_зала, Ряд
FROM "Кинотеатр"."Место"
GROUP BY Номер_зала, Ряд
HAVING COUNT(Номер_сидения) = SUM(CASE WHEN
Является_электронным = true THEN 1 ELSE 0 END);
```



The screenshot shows a SQL IDE interface. The top pane contains a SQL query. The bottom pane shows the results of the query in a table format. The table has two columns: 'Номер\_зала' (integer) and 'Ряд' (character varying (10)). The results are as follows:

	Номер_зала integer	Ряд character varying (10)
1	3	C
2	1	A
3	2	B
4	4	D
5	5	E

8) Найти клиентов с максимальным количеством заказанных фильмов и вывести их ФИО, email и количество фильмов.

```
WITH КоличествоФильмов AS (
    SELECT ФИО, email,
           LENGTH(История_заказов) - LENGTH(REPLACE(История_заказов, ',', '')) +
           1 AS Количество_фильмов
    FROM "Кинотеатр"."Клиент"
)
SELECT ФИО, email, Количество_фильмов
FROM КоличествоФильмов
WHERE Количество_фильмов = (SELECT MAX(Количество_фильмов) FROM
КоличествоФильмов);
```

```

1 WITH КоличествоФильмов AS (
2     SELECT ФИО, email,
3         LENGTH(История_заказов) - LENGTH(REPLACE(История_заказов, ',', '')) + 1 AS Количество_фильмов
4     FROM "Кинотеатр"."Клиент"
5 )
6 SELECT ФИО, email, Количество_фильмов
7 FROM КоличествоФильмов
8 WHERE Количество_фильмов = (SELECT MAX(Количество_фильмов) FROM КоличествоФильмов);

```

Data Output Messages Notifications

SQL Showing

	ФИО character varying (255)	email character varying (255)	Количество_фильмов integer
1	Алексей Смирнов	alex@example.com	8
2	Мария Иванова	maria@example.com	8
3	Ольга Петрова	olga@example.com	8
4	Дмитрий Васильев	dmitry@example.com	8
5	Анна Николаева	anna@example.com	8
6	Павел Лебедев	pavel@example.com	8
7	Светлана Григорьева	svetlana@example.com	8
8	Елена Морозова	elena@example.com	8
9	Иван Федоров	ivan@example.com	8

9) Найти клиентов, у которых в истории заказов есть фильм “Матрица”, и вывести их ФИО, email и полную историю заказов.

```

SELECT ФИО, email, История_заказов
FROM "Кинотеатр"."Клиент"
WHERE История_заказов ILIKE '%Матрица%'
ORDER BY ФИО;

```

```

1 SELECT ФИО, email, История_заказов
2 FROM "Кинотеатр"."Клиент"
3 WHERE История_заказов ILIKE '%Матрица%'
4 ORDER BY ФИО;

```

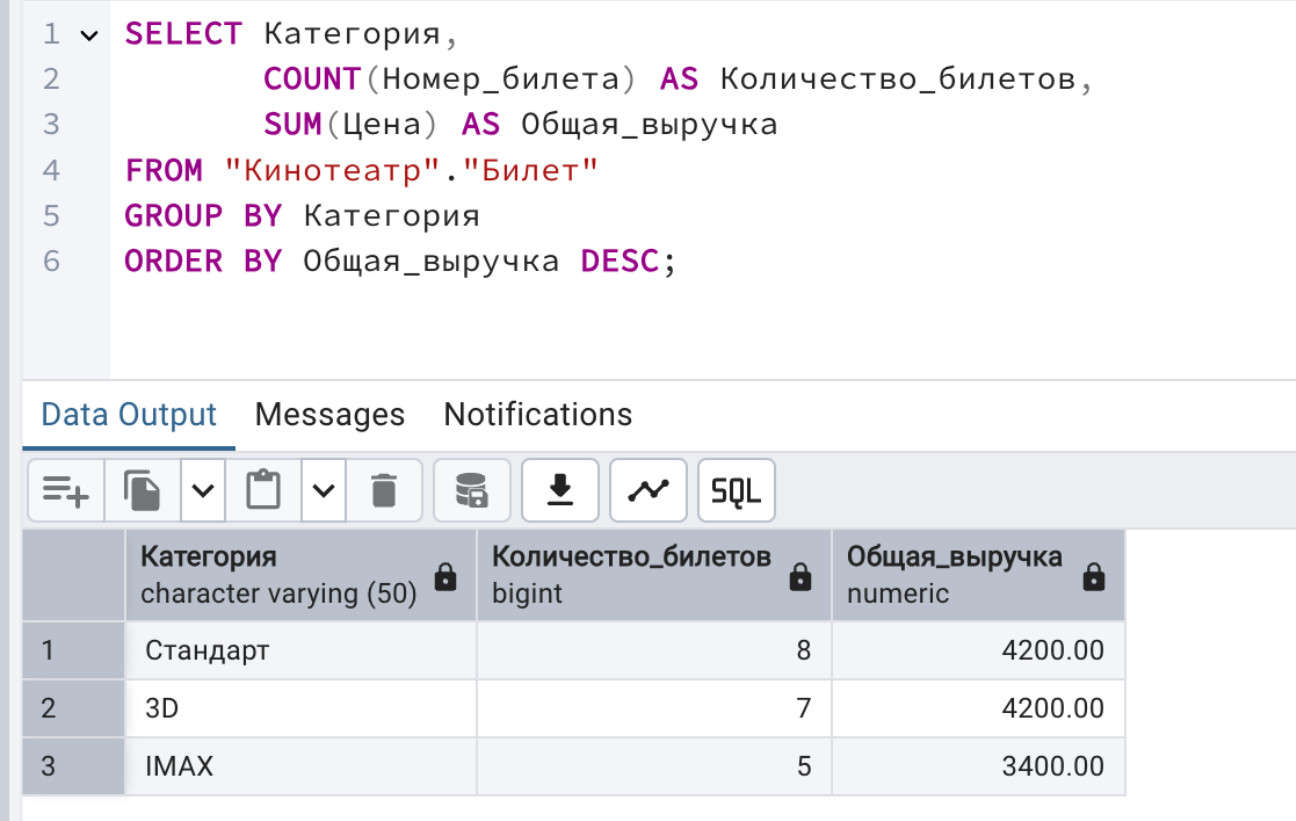
Data Output Messages Notifications

Showing rows: 1 to 7 Page No: 1

	ФИО character varying (255)	email character varying (255)	История_заказов text
1	Алексей Смирнов	alex@example.com	Властелин колец: Братство кольца, Интерстеллар, Начало, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского пер...
2	Дмитрий Васильев	dmitry@example.com	Титаник, Властелин колец: Братство кольца, Интерстеллар, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Криминальное чт...
3	Елена Морозова	elena@example.com	Титаник, Властелин колец: Братство кольца, Интерстеллар, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Криминальное чт...
4	Мария Иванова	maria@example.com	Интерстеллар, Начало, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского периода, Криминальное чтиво
5	Ольга Петрова	olga@example.com	Титаник, Властелин колец: Братство кольца, Начало, Матрица, Побег из Шоушенка, Форрест Гамп, Парк Юрского периода, Криминальное чт...
6	Павел Лебедев	pavel@example.com	Титаник, Интерстеллар, Начало, Матрица, Темный рыцарь, Форрест Гамп, Парк Юрского периода, Криминальное чтиво
7	Светлана Григорьева	svetlana@example.com	Титаник, Властелин колец: Братство кольца, Начало, Матрица, Побег из Шоушенка, Форрест Гамп, Парк Юрского периода, Криминальное чт...

10) Подсчитать общее количество билетов, проданных в каждой категории, и общую выручку по каждой категории.

```
SELECT Категория,  
       COUNT(Номер_билета) AS Количество_билетов,  
       SUM(Цена) AS Общая_выручка  
FROM "Кинотеатр"."Билет"  
GROUP BY Категория  
ORDER BY Категория, Общая_выручка DESC;
```



The screenshot shows a database management interface. At the top, an SQL query is entered in a text area. Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, displaying a table with the results of the query. The table has three columns: 'Категория' (Category), 'Количество\_билетов' (Number of tickets), and 'Общая\_выручка' (Total revenue). The results are sorted by category and then by total revenue in descending order.

	Категория character varying (50)	Количество_билетов bigint	Общая_выручка numeric
1	Стандарт	8	4200.00
2	3D	7	4200.00
3	IMAX	5	3400.00

### Вывод

При выполнении лабораторной работы были созданы запросы на выборку данных на языке SQL с использованием предложений SELECT, FROM, WHERE, GROUP BY и ORDER BY оператора SELECT, а также подзапросов, скалярных и агрегатных функций.