

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра Автоматизированных систем управления

Лабораторная работа № 4

по БАЗАМ ДАННЫХ

Проектирование запросов выборки данных

Студент

Улисков Н.В.

Группа АИ-19

Руководитель

Доцент

Алексеев В.А.

Липецк 2021г.

Цель работы

Изучить основы языка SQL, получить практические навыки разработки SELECT-запросов к базе данных с использованием конструкций селекции, сортировки, объединения, подзапросов, группировки данных и расчета статистических значений.

Задание кафедры

Разработать SELECT-запросы к БД, созданной в лабораторной работе №3, с использованием конструкций селекции, сортировки, объединения, подзапросов, группировки данных и расчета статистических значений. Проверить правильность работы запросов на тестовых данных.

1. Заполнение БД.

	cinema_id [PK] integer	cinema_address character varying (50)	working_hours character varying (30)	cinema_name character varying (30)
1	1	Советская,66	9:00-23:00	Европа
2	2	Ангарская,19	9:00-22:00	Рудничный
3	3	Московская,25	9:00-23:00	Салют
4	4	Водопьянова,22	8:00-23:00	Малина

Рисунок 1 — Таблица “cinema”

	cinema_id integer	hall_id [PK] integer	number_of_seats integer	hall_number integer
1	1	11	120	1
2	2	21	98	1
3	1	12	100	2
4	1	13	100	3
5	1	14	100	4
6	2	22	100	2
7	2	23	100	3
8	2	24	100	4
9	3	31	120	1
10	3	32	120	2
11	3	33	120	3
12	3	34	120	4

Рисунок 2 — Таблица “hall”

	id_film [PK] integer	film_name character varying (50)	film_genre character varying (40)	year_of_issue integer	start_date date	end_date date	duration integer	film_photo character varying (512)	film_cost integer
1	401	Человек Паук	экшен	2021	2021-12-15	2022-01-09	90	img/film1.jpg	300
2	402	Легенда	драма	2021	2021-12-12	2022-01-09	128	img/film2.jpg	300
3	403	Матрица	экшен	2021	2021-12-19	2022-01-14	110	img/film3.jpg	300
4	404	Драйв	боевик	2021	2021-10-01	2021-12-10	120	img/film4.jpg	300

Рисунок 3 — Таблица “movie”

	id_film integer	session_id [PK] integer	hall_id integer	session_date date
1	401	1001	11	2021-12-16
2	401	1002	11	2021-12-16
3	402	1003	12	2021-12-16
4	403	1004	12	2021-12-16
5	404	1005	13	2021-12-16
6	401	1006	21	2021-12-20
7	402	1007	22	2021-12-20
8	403	1008	23	2021-12-20
9	404	1009	24	2021-12-20
10	404	1010	31	2021-12-20

Рисунок 4 — Таблица “session”

	session_id integer	ticket_id [PK] integer	dispatcher_id integer	place_number integer	reservation smallint
1	1001	900000	701	9	1
2	1001	900001	702	18	1
3	1001	900002	701	86	0
4	1002	900003	701	32	1
5	1001	900004	701	41	1
6	1001	900005	703	15	0
7	1004	900006	701	2	0
8	1001	900007	704	11	0
9	1003	900008	702	7	1
10	1003	900009	702	6	1
11	1003	900010	702	5	0
12	1001	900011	702	22	1
13	1002	900012	701	33	1

Рисунок 5 — Таблица “ticket”

	dispatcher_id [PK] integer	login character varying (40)	password character varying (40)
1	701	user	p@ssword
2	702	root	qwerty123
3	705	IO1728	asd232ert
4	703	dis	zxc1289
5	704	print01	pr_020330

Рисунок 6 — Таблица “dispatcher”

2. Запрос выборки данных из одной таблицы

Выбрать все записи из таблицы “movie”, где продолжительность фильма меньше 120 минут и отсортировать по film_id.

```
SELECT * FROM movie WHERE duration<120 ORDER BY id_film;
```

Фильм		
Код фильма	INT4	<pk>
Название фильма	VARCHAR(40)	
Жанр	VARCHAR(40)	
Год выпуска	INT4	
Дата начала показа	DATE	
Дата завершения показа	DATE	
Продолжительность фильма	INT4	
Стоимость фильма	INT4	
Фото фильма	VARCHAR(512)	

Рисунок 7 — Фрагмент физической схемы данных

	id_film [PK] integer	film_name character varying (50)	film_genre character varying (40)	year_of_issue integer	start_date date	end_date date	duration integer	film_photo character varying (512)	film_cost integer
1	401	Человек Паук	экшен	2021	2021-12-15	2022-01-09	90	img/film1.jpg	300
2	403	Матрица	экшен	2021	2021-12-19	2022-01-14	110	img/film3.jpg	300

Рисунок 8 — Результат запроса 2

3. Запрос выборки данных из нескольких таблиц с использованием различных вариантов реализации соединения

3.1 Используя конструкцию WHERE

Выбрать все даты сеанса с фильмом “Человек паук” и отсортировать по session_id.

```
SELECT s.session_date
```

```
FROM movie m, session s
```

```
WHERE m.id_film=s.id_film AND m.film_name = 'Человек Паук'
```

```
ORDER BY session_id;
```

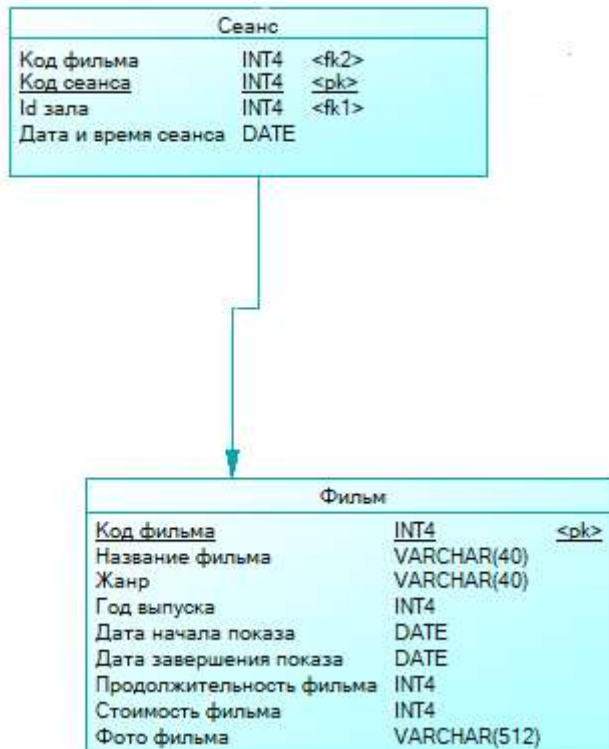


Рисунок 9 — Фрагмент физической схемы данных

	session_date	film_name
	date	character varying (50)
1	2021-12-16	Человек Паук
2	2021-12-16	Человек Паук
3	2021-12-20	Человек Паук

Рисунок 10 — Результат запроса 3.1

3.2 Используя конструкцию «внутренне соединение» - INNER JOIN

Вывести название фильма, код сеанса и жанр, где жанр – экшен.

Отсортировать по коду сеанса.

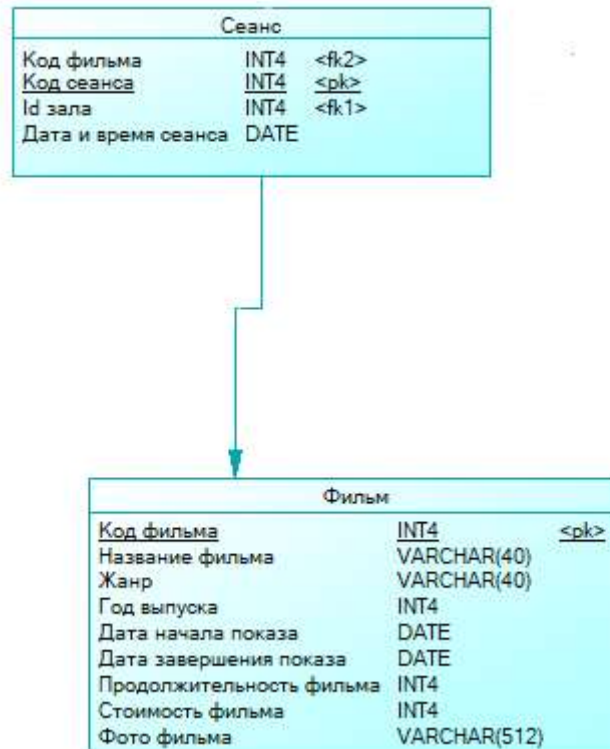


Рисунок 11 — Фрагмент физической схемы данных

	<u>session_id</u> integer	film_name character varying (50)	film_genre character varying (40)
1	1001	Человек Паук	экшен
2	1002	Человек Паук	экшен
3	1004	Матрица	экшен
4	1006	Человек Паук	экшен
5	1008	Матрица	экшен

Рисунок 12 — Результат запроса 3.2

3.3 Используя конструкцию «внешнее соединение» - OUTER JOIN

Вывести название фильма, жанр, код сеанса и цену фильма, где цена фильма меньше 300, отсортировать по коду сеанса.

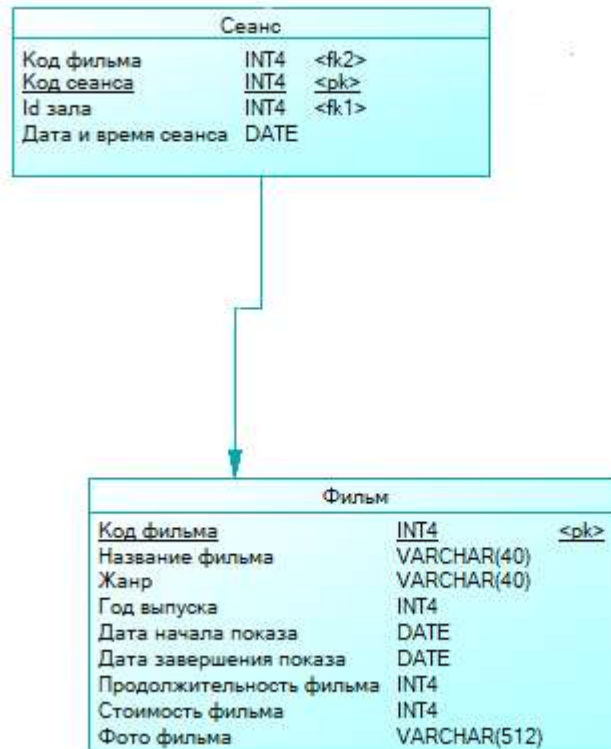


Рисунок 13 — Фрагмент физической схемы данных

	film_name character varying (50)	film_genre character varying (40)	film_cost integer	session_id integer
1	Драйв	боевик	250	1005
2	Драйв	боевик	250	1009
3	Драйв	боевик	250	1010

Рисунок 14 — Результат запроса 3.3

4. Запрос с подзапросом с использованием конструкции [NOT] IN

Вывести информацию о всех сеансах, у которых цена фильма больше 250

SELECT * FROM session

WHERE id_film NOT IN

(SELECT id_film FROM movie WHERE film_cost < 250);

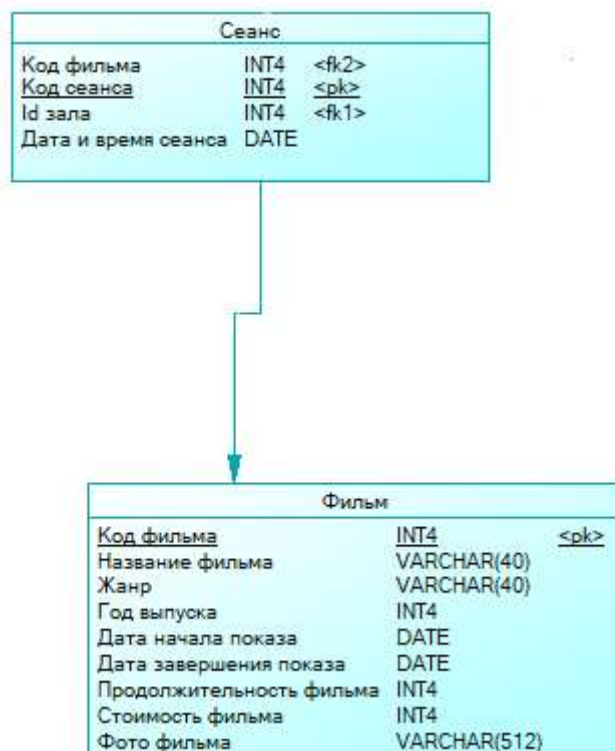


Рисунок 15 — Фрагмент физической схемы данных

	id_film integer	session_id [PK] integer	hall_id integer	session_date date
1	401	1001	11	2021-12-16
2	401	1002	11	2021-12-16
3	402	1003	12	2021-12-16
4	403	1004	12	2021-12-16
5	404	1005	13	2021-12-16
6	401	1006	21	2021-12-20
7	402	1007	22	2021-12-20
8	403	1008	23	2021-12-20
9	404	1009	24	2021-12-20
10	404	1010	31	2021-12-20

Рисунок 16 — Результат запроса 4

5. Запросы для получения статистических данных

5.1 Расчёт частных итогов для каждой комбинации значений атрибутов из списка

Сгруппировать фильмы по жанру.

```
SELECT COUNT(*) AS "Количество фильмов", film_genre  
FROM movie  
GROUP BY film_genre;
```

Фильм		
<u>Код фильма</u>	INT4	<pk>
Название фильма	VARCHAR(40)	
Жанр	VARCHAR(40)	
Год выпуска	INT4	
Дата начала показа	DATE	
Дата завершения показа	DATE	
Продолжительность фильма	INT4	
Стоимость фильма	INT4	
Фото фильма	VARCHAR(512)	

Рисунок 17 — Фрагмент физической схемы данных

	Количество фильмов bigint	film_genre character varying (40)
1	2	экшен
2	1	боевик
3	1	драма

Рисунок 18 — Результат запроса 5.1

5.2 Расчет частных итогов и итогов для каждой подкомбинации в порядке следования атрибутов из списка

Вывести название кинотеатра и количество мест в нем во всех залах, вместимость которых больше 98.

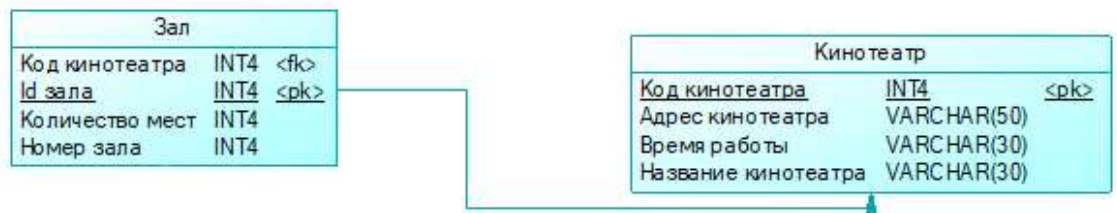


Рисунок 19 — Фрагмент физической схемы данных

	cinema_name character varying (30)	sum bigint
1	[null]	1320
2	Европа	420
3	Рудничный	300
4	Малина	120
5	Салют	480

Рисунок 20 — Результат запроса 5.2

5.3 Расчет частных итогов и итогов для каждой подкомбинации (все варианты) атрибутов из списка

Вывести цену фильма по его названию и дате сеанса.

```
SELECT SUM(m.film_cost),m.film_name, s.session_date
```

```
FROM session s
```

```
INNER JOIN movie m ON m.id_film = s.id_film
```

```
GROUP BY CUBE(m.film_name, s.session_date);
```

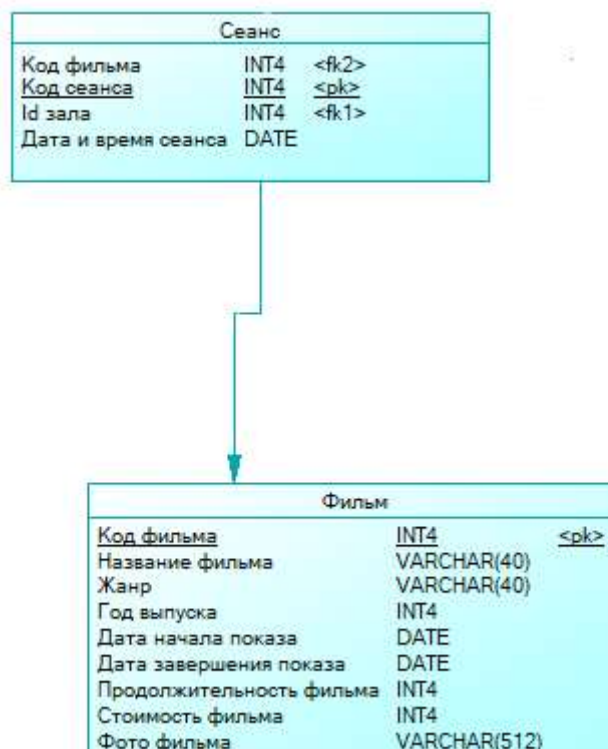


Рисунок 21 — Фрагмент физической схемы данных

	sum bigint	film_name character varying (50)	session_date date
1	2850	[null]	[null]
2	300	Человек Паук	2021-12-20
3	300	Матрица	2021-12-20
4	300	Легенда	2021-12-16
5	600	Человек Паук	2021-12-16
6	300	Матрица	2021-12-16
7	250	Драйв	2021-12-16
8	500	Драйв	2021-12-20
9	300	Легенда	2021-12-20
10	600	Легенда	[null]
11	600	Матрица	[null]
12	900	Человек Паук	[null]
13	750	Драйв	[null]

Рисунок 22 — Результат запроса 5.3

5.4 Расчет итогов для каждого уникального значения каждого атрибута из списка

Вывести цену всех сеансов фильма и дату сеанса

```
SELECT SUM(m.film_cost),m.film_name, s.session_date
```

```
FROM session s
```

```
INNER JOIN movie m ON m.id_film = s.id_film
```

```
GROUP BY GROUPING SETS(m.film_name, s.session_date);
```

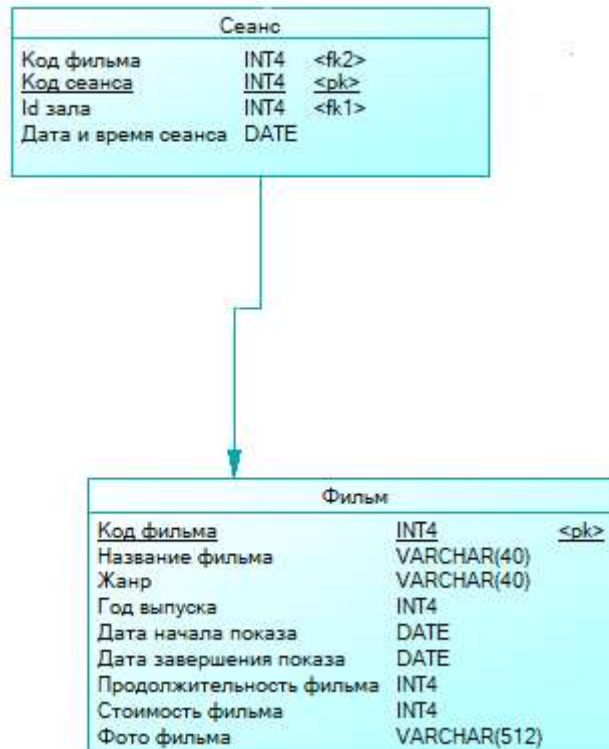


Рисунок 23 — Фрагмент физической схемы данных

	sum	film_name	session_date
	bigint	character varying (50)	date
1	600	Легенда	[null]
2	600	Матрица	[null]
3	900	Человек Паук	[null]
4	750	Драйв	[null]
5	1400	[null]	2021-12-20
6	1450	[null]	2021-12-16

Рисунок 24 — Результат запроса 5.4

Вывод

В ходе выполнения лабораторной работы я изучил основы языка SQL, получил практические навыки разработки SELECT-запросов к базе данных с использованием конструкций селекции, сортировки, объединения, подзапросов, группировки данных и расчета статистических значений.