

SCHEMAS

Filter objects

address

cinema

Tables

film

screening

tickets

Columns

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

employees

geodata

hw3

likes

sakila

sys

Administration

Schemas

Information

Table: **screening**

Columns:

id

film_id

time

final

int PK

int

datetime

datetime

Don't Limit

1 -- количество проданных билетов и выручка по фильмам

2 • select film.name as film_name, count(tickets.id) as tickets, sum(tickets.cost) as total_cost from film

3 join screening on film_id = film.id

4 join tickets on id_sc = screening.id

5 group by film.name

6 order by total_cost;

7

8 -- среднее количество зрителей на сеанс по фильмам

9 • select avg(tickets.id_sc) as avarange_quantity, film.name as film from tickets

10 join screening on id_sc = screening.id

11 join film on film_id = film.id

12 group by film.name;

13

14 -- продолжительность фильмов

15 • select distinct film.name as film_name, time(film.duration) as duration from film

16 join screening on film_id = film.id

17 order by film.name;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	film_name	tickets	total_cost
▶	Terminator	3	3000
	Pirates of Cribian See	4	4000
	Star Wars III	9	9000

SCHEMAS

Filter objects

- address
- cinema
 - Tables
 - film
 - screening
 - tickets
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - Views
 - Stored Procedures
 - Functions
- employees
- geodata
- hw3
- likes
- sakila
- sys

Administration Schemas

Information

Table: screening

Columns:

id	int PK
film_id	int
time	datetime
final	datetime

Don't Limit

```

1  -- количество проданных билетов и выручка по фильмам
2  • select film.name as film_name, count(tickets.id) as tickets, sum(tickets.cost) as total_cost from film
3      join screening on film_id = film.id
4      join tickets on id_sc = screening.id
5      group by film.name
6      order by total_cost;
7
8  -- среднее количество зрителей на сеанс по фильмам
9  • select avg(tickets.id_sc) as avarange_quantity, film.name as film from tickets
10     join screening on id_sc = screening.id
11     join film on film_id = film.id
12     group by film.name;
13
14  -- продолжительность фильмов
15  • select distinct film.name as film_name, time(film.duration) as duration from film
16     join screening on film_id = film.id
17     order by film.name;
  
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: ☐

	avarange_quantity	film
▶	1.8889	Star Wars III
	2.0000	Terminator
	4.0000	Pirates of Cribian See

SCHEMAS

Filter objects

- address
- cinema
 - Tables
 - film
 - screening
 - tickets
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - Views
 - Stored Procedures
 - Functions
 - employees
 - geodata
 - hw3
 - likes
 - sakila
 - sys

Administration Schemas

Information

Table: screening

Columns:

id	int PK
film_id	int
time	datetime
final	datetime

Don't Limit

```

9 • select avg(tickets.id_sc) as avarange_quantity, film.name as film from tickets
10 join screening on id_sc = screening.id
11 join film on film_id = film.id
12 group by film.name;
13
14 -- продолжительность фильмов
15 • select distinct film.name as film_name, time(film.duration) as duration from film
16 join screening on film_id = film.id
17 order by film_name;
18
19
20 -- накладки в сеансах
21 • select film.name as film_name, screening.final, screening.time, film.duration from screening
22 join film on film_id = film.id
23 group by film.name
24 having (time_to_sec(screening.final) - time_to_sec(screening.time) - time_to_sec(film.duration)) <= 0
25

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: I A

	film_name	final	time	duration
▶	Star Wars III	2023-01-01 11:00:00	2023-01-01 10:00:00	2023-01-01 01:00:00
	Terminator	2023-01-01 12:30:00	2023-01-01 11:00:00	2023-01-01 01:30:00