

```
1 /* 1 Вывести название, описание и длительность фильмов, выпущенных после
2 2000-ого года. Включить только фильмы длительностью в интервале от 60
3 до 120 минут (включ.). Показать первые 20 фильмов по длительности
4 (самые длинные).
5 */
6 SELECT title, description, length
7 FROM film
8 WHERE release_year > 2000 AND length BETWEEN 60 AND 120
9 ORDER BY length DESC
10 LIMIT 20;
11
12
13 /*2. Найти все платежи, совершенные в апреле 2007-го года, чья стоимость не
14 превышает 4 долларов. Показать идентификатор, дату (без времени), и
15 стоимость платежа. Платежи отобразить в порядке убывания стоимости.
16 При совпадении стоимости, отдать предпочтение более раннему платежу.
17 */
18 SELECT payment_id, DATE(payment_date), amount
19 FROM payment
20 WHERE DATE (payment_date) BETWEEN '2007-04-01' AND '2007-04-30' AND amount <= 4
21 ORDER BY amount DESC, payment_date;
22
23 /*3. Показать имена, фамилии и идентификаторы всех клиентов с именами
24 "Jack", "Bob", или "Sara", чья фамилия содержит букву "р". Переименовать
25 колонку с именем в "Имя", с идентификатором в "Идентификатор", с
26 фамилией в "Фамилия". Клиентов отобразить в порядке возрастания их
27 идентификатора.*/
28 SELECT first_name AS "Имя",
29        last_name AS "Фамилия",
30        customer_id AS "Идентификатор"
31 FROM customer
32 WHERE first_name IN ('Jack', 'Bob', 'Sara') AND last_name ILIKE '%p%'
33 ORDER BY customer_id;
34
35 /*4. Посчитать выручку в каждом месяце работы проката. Месяц должен
36 рассчитываться по rental_date, а не по payment_date. Округлить выручку до
37 одного знака после запятой. Отсортировать строки в хронологическом
38 порядке.
39 Подсказка: есть месяц проката, где выручки не было (нет данных о
40 платежах) - он должен присутствовать в отчете.
41 */
42 SELECT EXTRACT(YEAR FROM rental_date),
43        EXTRACT(MONTH FROM rental_date),
44        ROUND(SUM(p.amount),1)
45 FROM rental r
46 LEFT JOIN payment p ON r.rental_id=p.rental_id
47 GROUP BY 1,2
48 ORDER BY 1,2;
49
50 /*5. Найти средний платеж по каждому жанру фильма. Отобразить только те
51 жанры, к которым относится более 60 различных фильмов. Округлить
52 средний платеж до двух знаков после запятой. Дать названия столбцам.
53 Отобразить жанры в порядке убывания среднего платежа.*/
54 SELECT c.name AS "Жанр",
55        ROUND(AVG(amount),2) AS "Средний платеж",
56 FROM film_category fc
57 LEFT JOIN category c ON fc.category_id=c.category_id
58 LEFT JOIN film f ON fc.film_id=f.film_id
59 LEFT JOIN inventory i ON f.film_id=i.film_id
60 LEFT JOIN rental r ON i.inventory_id=r.inventory_id
61 LEFT JOIN payment p ON r.rental_id=p.rental_id
62 GROUP BY c.category_id
```

```
63 HAVING count(DISTINCT fc.film_id)>60
64 ORDER BY 2 DESC;
65
66
67 /*6 Какие фильмы чаще всего берут напрокат по субботам? Показать названия
68 первых 5 по популярности фильмов. Если у фильмов одинаковая
69 популярность, отдать предпочтение первому по алфавиту.
70 Подсказка: день недели можно извлечь с помощью EXTRACT*/
71 SELECT f.title,
72        count(rental_id)
73 FROM film f
74 LEFT JOIN inventory i ON f.film_id=i.film_id
75 LEFT JOIN rental r ON r.inventory_id=i.inventory_id
76 WHERE extract(dow from rental_date::timestamp)=6
77 GROUP BY 1
78 ORDER BY count(rental_id) DESC, f.title ASC
79 LIMIT 5;
80
81 /*7 Распределить фильмы в три категории по длительности:короткие менее 70
82 средние от 70 до 130 (не вкл.)
83 длинные 130 и выше
84 Рассчитать количество прокатов и количество фильмов в каждой такой
85 категории. Если прокатов у фильма не было, не включать его в расчеты
86 количества фильмов в категории (подумать над типом джоина ☺).
87 Подсказка: количество фильмов и количество прокатов не будут
88 одинаковыми числами, ведь фильмы берут напрокат много раз.
89 */
90 SELECT CASE
91        WHEN length < 70 THEN 'короткие'
92        WHEN length < 130 THEN 'средние'
93        WHEN length >=130 THEN 'длинные'
94        END AS "Категория",
95        count(DISTINCT f.film_id) AS "Кол-во фильмов",
96        count(DISTINCT r.rental_id)AS "Кол-во прокатов"
97 FROM film f
98 FULL JOIN inventory i ON f.film_id = i.film_id
99 INNER JOIN rental r ON i.inventory_id=r.inventory_id
100 GROUP BY 1
101 ORDER BY 1;
102
103
104 /* Для последующих запросов создадим таблицу weekly_revenue с выручкой по
105 неделям и будем работать с ней:*/
106 CREATE TABLE weekly_revenue AS
107     SELECT EXTRACT(YEAR FROM rental_date) AS r_year,
108            EXTRACT(week FROM rental_date) AS r_week,
109            sum(amount) AS revenue
110     FROM rental r
111     LEFT JOIN payment p ON p.rental_id=r.rental_id
112     GROUP BY 1,2
113     ORDER BY 1,2;
114
115 SELECT *
116 FROM weekly_revenue;
117
118 /*8. Рассчитать накопленную сумму недельной выручки бизнеса.
119 Вывести всю таблицу weekly_revenue с дополнительным столбцом с
120 накопленной суммой.
121 Округлить накопленную выручку до целого числа*/
122 SELECT *, ROUND( SUM(revenue) OVER(ORDER BY r_year, r_week)) AS cum_avarage
123 FROM weekly_revenue;
124
```

```
125 /* 9 Рассчитать скользящую среднюю недельной выручки бизнеса.
126 Использовать неделю до, текущую неделю, и неделю после для расчета
127 среднего значения.
128 Вывести всю таблицу weekly_revenue с дополнительными столбцами с
129 накопленной суммой и скользящей средней.
130 Округлить скользящую среднюю до целого числа.
131 */
132 SELECT *,
133     ROUND( SUM(revenue) OVER(ORDER BY r_year, r_week)) AS cum_avarage,
134     ROUND( AVG (revenue) OVER (ORDER BY r_year, r_week
135                               ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING)) AS mov_avarage
136 FROM weekly_revenue;
137
138 /*10.Посчитать прирост недельной выручки бизнеса в %.
139 Прирост в % =
140 Текущая выручка - Предшествующая выручка
141 Предшествующая выручка
142 * 100%
143 Вывести всю таблицу weekly_revenue с дополнительными столбцами с
144 накопленной суммой, скользящей средней и приростом.
145 Округлить прирост в процентах до 2 знаков после запятой.
146 */
147
148 SELECT *,
149     ROUND( SUM(revenue) OVER(ORDER BY r_year, r_week
150                               ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)) AS
    cum_avarage,
151     ROUND( AVG (revenue) OVER (ORDER BY r_year, r_week
152                               ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING)) AS mov_avarage,
153     ROUND((revenue-lag(revenue, 1) OVER my_window)*100.00/lag(revenue, 1) OVER
154 my_window, 2) AS pct_growth
155 FROM weekly_revenue
156 WINDOW my_window AS (
157 ORDER BY r_year, r_week ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW);
158
```