

Агрегатные функции

- Легко
- Нормально
- Сложно

№		Задание	Решение
1.		Найдите среднюю продолжительность фильма. Результат выведите в колонке avg_film_length.	<code>select avg(length) as "avg_film_length" from film;</code>
2.		Найдите минимальную и максимальную стоимость замены пленки. Отобразите результат в виде таблицы с двумя столбцами: minimal_replacement_cost и maximal_replacement_cost.	<code>select min(replacement_cost) as minimal_replacement_cost, max(replacement_cost) as maximal_replacement_cost from film;</code>
3.		Найдите среднее время проката фильма (в днях). Отобразите результат в колонке с именем average_rental_time. Время аренды рассчитывается как разница между return_date и rental_date в таблице аренда. Округлите результат до ближайшего целого числа.	<code>select round (avg(datediff(return_date,rental_date))) as average_rental_time from rental;</code>
4.		Найдите среднее время проката фильма (в днях). Отобразите результат в виде таблицы с одним столбцом average_rental_time. Время аренды рассчитывается как разница между return_date и rental_date в таблице rental	<code>select avg(datediff(return_date, rental_date)) as average_rental_time from rental;</code>
5.		Найти количество сотрудников в каждом подразделении. Выведите имя подразделения DEPARTMENT и число сотрудников в нём EMP_COUNT. Отсортируйте результат по убыванию числа сотрудников.	<code>select DEPARTMENT.DEPARTMENT , count(EMPLOYEE.EMP_NO) as EMP_COUNT from EMPLOYEE right join DEPARTMENT on EMPLOYEE.DEPT_NO=DEPARTMENT.DEPT_NO group by DEPARTMENT.DEPARTMENT order by EMP_COUNT desc;</code>
6.		Найти количество фильмов в каждой категории. Выведите таблицу состоящую из двух колонок category и film_count отсортируйте её по названию категорий в алфавитном порядке.	<code>select category.name as category, count(film_category.film_id) as film_count from category left join film_category on category.category_id=film_category.category_id group by category.name order by category.name asc;</code>
7.		Найдите среднюю стоимость проката фильма для каждой категории.	<code>select category.name as category, avg(film.rental_rate)as avg_rental_rate from category</code>

		<p>Результат выведите в две колонки category и avg_rental_rate отсортировав по убыванию цены.</p>	<pre>inner join film_category on category.category_id=film_category.category_id inner join film on film_category.film_id=film.film_id group by category order by avg_rental_rate desc;</pre>
8.		<p>Найдите минимальную и максимальную и среднюю продолжительность фильма для каждой категории. Отобразите результат в виде таблицы с колонками: category - название категории фильмов, min_length, max_length и avg_length отсортированной по категории в алфавитном порядке</p>	<pre>select category.name as category, min(film.length) as min_length, max(film.length) as max_length, avg(film.length) as avg_length from category inner join film_category on category.category_id=film_category.category_id inner join film on film_category.film_id=film.film_id group by category order by category asc;</pre>
9.		<p>Найдите категории со средней продолжительностью фильма более двух часов. Отобразите результат в виде таблицы с колонками: category - название категории фильмов и avg_length отсортированной по убыванию средней длины фильма</p>	<pre>select category.name as category, avg(film.length) as avg_length from category inner join film_category on category.category_id=film_category.category_id inner join film on film_category.film_id=film.film_id group by category.name having avg(film.length)>120 order by avg_length desc;</pre>
10.		<p>Среди фильмов имеющихся в наличии найдите те, которые реже всего берут напрокат, на основе количества записей в таблице rental. Выведите название фильма в столбце title и количество аренд в столбце rentals_count. Отсортируйте таблицу по названию фильма.</p>	<pre>select distinct film.title , count(rental.rental_id) as rentals_count from film inner join inventory on inventory.film_id=film.film_id inner join rental on rental.inventory_id=inventory.inventory_id group by film.title having rentals_count = (select count(rental.rental_id) as rentals_count from film inner join inventory on inventory.film_id=film.film_id inner join rental on rental.inventory_id=inventory.inventory_id group by film.title order by rentals_count asc limit 1) order by film.title asc;</pre>

11.	<p>Напишите SQL запрос, чтобы найти трех крупнейших клиентов с наибольшим общим объемом платежей в базе данных Sakila.</p> <p>Отобразите в таблице результатов имя, фамилию и общую сумму платежей клиентов в first_name, last_name и total_pay соответственно. Отсортируйте результаты по total_pay в порядке убывания.</p>	<pre>select customer.first_name, customer.last_name, SUM(payment.amount) as total_pay from customer inner join payment on customer.customer_id=payment.customer_id group by customer.customer_id order by total_pay desc LIMIT 3;</pre>
12.	<p>Найдите среднее время проката фильма для каждого из клиентов.</p> <p>Отобразите результат в виде таблицы с колонками customer_id, first_name, last_name и average_rental_time.</p> <p>Время аренды рассчитывается как разница между return_date и rental_date в таблице rental.</p> <p>Результат округлите до целого в большую сторону. Отсортируйте результат по возрастанию поля customer_id.</p>	<pre>select customer.customer_id, customer.first_name, customer.last_name, CEILING(avg(datediff(rental.return_date,rental.rental_date))) as average_rental_time from rental inner join customer on rental.customer_id=customer.customer_id group by customer.customer_id order by customer.customer_id asc;</pre>
13.	<p>Напишите запрос для расчета суммы платежей за каждый месяц и отображения результатов в порядке убывания месяца платежа.</p> <p>Запрос должен вернуть только два столбца: payment_month – месяц платежа (в формате «ГГГГ-ММ») и payment_amount – общая сумма платежа за каждый месяц.</p>	<pre>select date_format(payment_date,'%Y-%m') as payment_month, Sum(amount) as payment_amount from payment group by payment_month order by payment_month desc;</pre>
14.	<p>Найдите количество дисков с фильмами каждой категории в каждом из двух магазинов.</p> <p>Напишите запрос выводящий таблицу с колонками: category - название категории фильмов, store_1 и store_2 - количество дисков с фильмами данной категории в магазине.</p> <p>Отсортируйте результат по категории в алфавитном порядке.</p>	<pre>select category.name as category, count(IF (inventory.store_id=1, 1, null)) as store_1, count(IF (inventory.store_id=2, 1, null)) as store_2 from category inner join film_category on category.category_id=film_category.category_id inner join inventory on film_category.film_id=inventory.film_id group by category.name order by category.name asc;</pre>
15.	<p>Найдите сотрудников зарплата которых повышалась несколько раз в течении одного года.</p> <p>Результат должен содержать следующие столбцы EMP_NO, FIRST_NAME, LAST_NAME - данные сотрудника, CHANGES_YEAR - год повышения зарплаты, CHANGES_COUNT - количество повышений в году. Отсортируйте результат по номеру сотрудника</p>	<pre>select e.EMP_NO, e.FIRST_NAME, e.LAST_NAME, extract(year from s.CHANGE_DATE) as CHANGES_YEAR, count(*) as CHANGES_COUNT from EMPLOYEE as e join</pre>

		SALARY_HISTORY as s on e.EMP_NO = s.EMP_NO group by e.EMP_NO, e.FIRST_NAME, e.LAST_NAME, extract(year from s.CHANGE_DATE) having count(*) > 1 order by e.EMP_NO;
16.	Напишите запрос, вычисляющий отношение минимальной зарплаты в отделе к максимальной. Выведите таблицу из двух колонок DEPARTMENT - название отдела в котором он работает и SALARY_DIFF_RATIO отсортированную по убыванию коэффициента	select distinct DEPARTMENT.DEPARTMENT, min(distinct CAST(EMPLOYEE.SALARY AS REAL))/max(distinct CAST(EMPLOYEE.SALARY AS REAL)) as SALARY_DIFF_RATIO from EMPLOYEE inner join DEPARTMENT on EMPLOYEE.DEPT_NO=DEPARTMENT.DEPARTMENT_NO group by DEPARTMENT.DEPARTMENT order by SALARY_DIFF_RATIO desc;
17.	Напишите запрос для расчета выручки каждого магазина за каждый квартал 2005 года. Результат должен состоять из следующих полей: store_id и I, II, III, IV - суммы доходов за каждый квартал соответственно Чтобы определить значение store_id, используйте соединение с таблицей staff.	select s.store_id, sum(case when extract(quarter from sa.payment_date) = 1 then sa.amount else 0 end) as i, sum(case when extract(quarter from sa.payment_date) = 2 then sa.amount else 0 end) as ii, sum(case when extract(quarter from sa.payment_date) = 3 then sa.amount else 0 end) as iii, sum(case when extract(quarter from sa.payment_date) = 4 then sa.amount else 0 end) as iv from payment as sa join staff as st on sa.staff_id = st.staff_id join store as s on st.store_id = s.store_id where extract(year from sa.payment_date) = 2005 group by s.store_id;
18.	Найдите три страны с наибольшим количеством проживающих в них клиентов. Выведите результат в две колонки: country и customers_count отсортировав по количеству клиентов от большего к меньшему.	select country.country, count(customer.customer_id) as customers_count from country inner join city on city.country_id=country.country_id inner join address

		on city.city_id=address.city_id inner join customer on address.address_id=customer.address_id group by country.country_id order by customers_count desc limit 3;
19.	Найдите количество дисков находящихся в прокате на конец дня 2005-05-31 по каждому из пунктов проката. Выведите результат в две колонки store_id и rented_at_2005_05_31 Отсортируйте полученный список по возрастанию store_id.	select distinct inventory. store_id, count(rental.rental_date) rented_at_2005_05_31 from rental inner join inventory on rental.inventory_id=inventory.inventory_id where rental_date<'2005-06-1 00:00:00' and return_date>='2005-06-1 00:00:00' group by inventory.store_id order by inventory.store_id asc;
20.	Найдите количество дисков возвращённых в течении 2005-06-01 по каждому из пунктов проката. Выведите результат в две колонки store_id и returned_at_2005_06_01. Отсортируйте полученный список по возрастанию store_id.	select distinct inventory.store_id, count(rental.return_date) returned_at_2005_06_01 from rental inner join inventory on rental.inventory_id=inventory.inventory_id where return_date>='2005-06-01 00:00:00' and return_date<'2005-06-02 00:00:00' group by inventory.store_id order by inventory.store_id asc;
21.	Найдите фамилии актёров, которые встречаются в таблице более одного раза. Выведите результат в два столбца: last_name — фамилия актёра и first_names — список имен актёров носящих эту фамилию через запятую. Отсортируйте список по фамилиям в алфавитном порядке.	select last_name, group_concat(first_name) as first_names from actor group by last_name having count(last_name)>1 order by last_name asc;
22.	Получите название каждого фильма из таблицы фильмов вместе со списком актёров. Список актёров должен состоять из имени и фамилии актёра через пробел и разделен запятыми и отсортирован по алфавиту. (Как в этом примере: ELVIS MARX, SISSY SOBIESKI, VAL BOLGER). Результат запроса должен содержать два столбца: title – название фильма и actors, упорядоченные по количеству актёров в порядке убывания.	select distinct film.title, group_concat(distinct actor.first_name,' ', actor.last_name order by actor.first_name, actor.last_name separator ' , ') as actors from film left join film_actor on film.film_id=film_actor.film_id left join actor on film_actor.actor_id=actor.actor_id group by film.title having count(actor.first_name) > 0 order by count(actor.first_name) desc;
23.	Найдите всех актёров снимавшихся в фильме ARIZONA BANG . Выведите результирующую таблицу из двух колонок first_name, last_name отсортированных по убыванию	select actor.first_name, actor.last_name from actor

		популярности актера (актёр, снимавшийся в большем количестве фильмов, должен находиться в первой строке).	<pre> join film_actor on actor.actor_id = film_actor.actor_id join film on film_actor.film_id = film.film_id where film.title = 'arizona bang' order by (select count(*) from film_actor where film_actor.actor_id = actor.actor_id) desc; </pre>
24.		<p>Подсчитайте, сколько дисков в неделю брал в среднем каждый из клиентов в августе 2005 г.</p> <p>Результат представьте в двух столбцах: week_num - номер недели, weekly_average_rental_count - среднее количество дисков, взятых напрокат одним пользователем. Отсортируйте результаты по номеру недели.</p>	<pre> select week(rental_date) as week_num, count(DISTINCT rental_id)/count(DISTINCT customer_id) as weekly_average_rental_count from rental where DATE_FORMAT(rental_date,'%Y- %m')='2005-08' group by week_num order by week_num asc; </pre>
25.		<p>Найдите клиентов, которые взяли напрокат один и тот же фильм несколько раз.</p> <p>Представьте результат в виде таблицы со следующими столбцами: first_name, last_name — имя и фамилия клиента, title - название фильма, first_rental_date, last_rental_date - первая и последняя даты проката фильма rentals_count — сколько раз фильм был взят напрокат клиентом. Отсортируйте таблицу по названию фильма.</p>	<pre> select customer.first_name, customer.last_name, film.title , min(rental.rental_date) as first_rental_date, max(rental.rental_date) as last_rental_date, count(rental.rental_id) as rentals_count from customer inner join rental on customer.customer_id=rental.customer_id inner join inventory on rental.inventory_id=inventory.inventory_id inner join film on inventory.film_id=film.film_id group by film.film_id, customer.customer_id HAVING count(rental.rental_id)> 1 order by film.title asc; </pre>
26.		<p>Составьте список фильмов, копии которых есть только в одном из двух филиалов фирмы.</p> <p>Выведите таблицу со столбцами film_id, title, store_id — идентификатор пункта проката где присутствует копия, quantity — количество копий в филиале. Отсортируйте таблицу по названию фильма.</p>	<pre> select distinct q.film_id, q.title, inventory.store_id, q.quantity from (select film.film_id, film.title, count(distinct inventory.store_id) as cis, count(inventory.inventory_id) as quantity from film inner join inventory on film.film_id=inventory.film_id group by film.film_id having count(inventory.inventory_id)>0) as q left join (select film.film_id, film.title, count(distinct inventory.store_id), count(inventory.inventory_id) as quantity from film inner join inventory on film.film_id=inventory.film_id group by film.film_id having count(inventory.store_id)=2) as w on q.film_id=w.film_id inner join inventory on q.film_id=inventory.film_id where q.cis=1 order by q.title asc; </pre>

27.	<p>Составьте список фильмов, все копии которых находились в руках у клиентов на дату 2005-05-31 12:00:00</p> <p>Отобразите таблицу со столбцами film_id, title. Отсортируйте таблицу по названию фильма.</p>	<pre>select ss.film_id, ss.title from (select distinct inventory.film_id, film.title, count(inventory.inventory_id) as w from film inner join inventory on film.film_id=inventory.film_id group by inventory.film_id having count(inventory.inventory_id)>0) as ww inner join (select distinct inventory.film_id, film.title, count(rental.rental_id) as s from film inner join inventory on film.film_id=inventory.film_id inner join rental on inventory.inventory_id=rental.inventory_id where rental_date<='2005-05-31 12:00:00' and return_date>='2005-05-31 12:00:00' group by film.film_id) as ss on ww.film_id=ss.film_id where ww.w=ss.s order by ss.title asc;</pre>
28.	<p>Найдите количество фильмов в каждой категории.</p> <p>Верните результат в формате JSON вида [{"Action": 64}, {"Animation": 66}, ...] в столбце films_by_category</p>	<pre>SELECT JSON_ARRAYAGG(JSON_OBJECT (w.n, w.i)) as films_by_category FROM (select category.name as n, count(film_category.film_id) as i from category inner join film_category on category.category_id=film_category.category_ id group by category.category_id) as w;</pre>
29.	<p>Напишите SQL-запрос, чтобы найти фильм, который чаще всего брали в прокат в феврале 2006 года.</p> <p>Отобразите название фильма в столбце most_rented_movie.</p> <p>Результатом должна стать таблица с одним столбцом и одной строкой.</p>	<pre>select www.title as most_rented_movie from (select film.title, count(rental.rental_id) as s from film inner join inventory on film.film_id=inventory.film_id inner join rental on inventory.inventory_id=rental.inventory_id where YEAR(rental.rental_date)='2006' and MONTH(rental.rental_date)='02' group by film.film_id order by s desc limit 1) as www;</pre>
30.	<p>Найдите фильмы с наибольшим количеством аренд в 2005 году в каждой из категорий. Выведите таблицу со столбцами category, most_rented_films - название фильма (если несколько фильмов имеют одинаковое количество аренд выведите названия по алфавиту, через запятую) и rentals_count - количество аренд фильма в течении года.</p> <p>Отсортируйте таблицу по первому столбцу.</p>	<pre>select qqq.category , aaa.most_rented_films , qqq.rentals_count from (select www.name as category, max(www.ddd) as rentals_count from (select category.name, film.title, category.category_id, count(rental.rental_id) as ddd from category inner join film_category on category.category_id=film_category.category_ id inner join film on film_category.film_id=film.film_id inner join inventory on film.film_id=inventory.film_id inner join rental on inventory.inventory_id=rental.inventory_id where year(rental.rental_date)='2005' group by category.category_id, film.film_id) as www group by www.category_id) as qqq inner join (select www.name as category,</pre>

		<p>group_concat(www.title order by www.title) as most_rented_films, www.ddd as rentals_count from (select category.name, film.title, category.category_id, count(rental.rental_id) as ddd from category inner join film_category on category.category_id=film_category.category_id inner join film on film_category.film_id=film.film_id inner join inventory on film.film_id=inventory.film_id inner join rental on inventory.inventory_id=rental.inventory_id where year(rental.rental_date)='2005' group by category.category_id, film.film_id) as www group by www.category_id, www.ddd order by rentals_count desc) as aaa on qqq.category=aaa.category where qqq.rentals_count=aaa.rentals_count order by qqq.category asc;</p>
31.	<p>Найдите среднюю стоимость проката фильма и среднюю фактическую сумму аренды фильма для каждой категории. Результат выведите в три колонки category, avg_rental_rate и avg_payment_amount. Отсортируйте результат по убыванию разницы между фактической и номинальной стоимостью.</p>	<p>select aaa.category, aaa.avg_rental_rate, bbb.avg_payment_amount from (select category.name as category, avg(film.rental_rate) as avg_rental_rate from category inner join film_category on category.category_id=film_category.category_id inner join film on film_category.film_id=film.film_id GROUP BY category.category_id) as aaa inner join (select category.name as category, avg(payment.amount) as avg_payment_amount from category inner join film_category on category.category_id=film_category.category_id inner join film on film_category.film_id=film.film_id inner join inventory on film.film_id=inventory.film_id inner join rental on inventory.inventory_id=rental.inventory_id inner join payment on rental.rental_id=payment.rental_id GROUP BY category.category_id) as bbb on aaa.category=bbb.category order by (bbb.avg_payment_amount-aaa.avg_rental_rate) desc;</p>