

"Основы SQL"

- Легко
- Нормально
- Сложно

№		Задание	Решение
1.		Выберите все записи из таблицы actor.	SELECT*FROM actor;
2.		Напишите SQL-запрос для выбора столбцов sex - пол и body_mass_g - масса тела из таблицы little_penguins, отсортированных таким образом, чтобы сначала отображалась пингвины с наибольшей масса тела.	SELECT sex, body_mass_g FROM little_penguins ORDER By body_mass_g DESC;
3.		Получите все записи из таблицы address, для которых не указан почтовый индекс. Отсортируйте результат по address_id.	SELECT*FROM address WHERE postal_code is null ORDER BY address_id ASC ;
4.		Получите столбец name из таблицы language в алфавитном порядке.	SELECT name from language ORDER By name asc;
5.		Выберите все значения имён и фамилий актёров из таблицы actor.	SELECT first_name, last_name FROM actor;
6.		Получите список значений из колонки name таблицы language.	SELECT name FROM language;
7.		Выберите названия фильмов из таблицы film. Отсортируйте полученный список по алфавиту	SELECT title FROM film ORDER By title ASC;
8.		Из таблицы customer выберите все записи о фамилии - last_name, имени - first_name и адресе электронной почты email отсортировав их по фамилии в алфавитном порядке.	SELECT last_name, first_name, email FROM customer ORDER BY last_name ASC;
9.		Напишите SQL запрос, который выводит список уникальных значений rating из таблицы film в алфавитном порядке.	SELECT DISTINCT rating FROM film ORDER BY rating ASC;
10.		Получите названия пяти самых длинных фильмов, отсортированных по продолжительности в порядке убывания.	SELECT title FROM film ORDER BY length DESC LIMIT 5;
11.		Выберите название, описание и год выхода фильмов из таблицы film. Отсортируйте полученный список по названию в алфавитном порядке и выведите первые десять строк	SELECT title, description, release_year FROM film ORDER BY title ASC LIMIT 10;
12.		Для удобства показа мы разобьем список фильмов на страницы по десять записей на каждой. Для формирования третьей страницы списка выберите название, описание и год выхода фильмов из таблицы film. Отсортируйте полученный список по названию в алфавитном порядке и выведите десять строк начиная с двадцать первой.	SELECT title, description, release_year FROM film ORDER BY title ASC LIMIT 20,10;

13.	<p>Выберите название, стоимость проката и продолжительность фильмов из таблицы film.</p> <p>Отсортируйте полученный список по убыванию стоимости, фильмы с одинаковой стоимостью отсортируйте по возрастанию продолжительности фильма.</p>	<pre>SELECT title, rental_rate, length FROM film ORDER BY rental_rate DESC,length ASC;</pre>
14.	<p>Найдите самый длинный фильм в таблице film.</p> <p>Если несколько фильмов имеют одинаковую продолжительность, выберите фильм с наименьшей ценой замены replacement_cost</p> <p>Напишите запрос, без использования агрегатных функций, который возвращает два столбца: title и release_year.</p>	<pre>SELECT title, release_year FROM film ORDER BY length DESC, replacement_cost ASC LIMIT 1;</pre>
15.	<p>Найдите все фильмы продолжительностью более трёх часов.</p> <p>Напишите SQL запрос возвращающий результат состоящий из трёх столбцов: названия фильма, его описания и продолжительности в минутах отсортированный по длине фильма.</p>	<pre>SELECT title, description, length FROM film WHERE length>180 ORDER BY length ASC;</pre>
16.	<p>Найдите сотрудников, работающих в магазине номер 1, и получите все их данные.</p>	<pre>SELECT * FROM staff WHERE store_id =1;</pre>
17.	<p>Найдите всех активных в данный момент клиентов (active = 1) в таблице customer. Таблица результатов должна содержать следующие поля: customer_id, first_name и last_name.</p>	<pre>SELECT customer_id, first_name, last_name FROM customer WHERE active = 1;</pre>
18.	<p>Найдите актеров по имени Scarlett.</p>	<pre>SELECT*FROM actor WHERE first_name='Scarlett';</pre>
19.	<p>Найдите все фильмы, в описании которых есть слово Student. Выведите названия фильмов в алфавитном порядке.</p>	<pre>SELECT title FROM film WHERE description LIKE '%Student%' ORDER BY title ASC;</pre>
20.	<p>Найдите все фильмы продолжительностью более 3 часов и получите их название, год выпуска и продолжительность, отсортированные по продолжительности в порядке возрастания.</p>	<pre>SELECT title, release_year, length FROM film WHERE length>180 ORDER BY length ASC;</pre>
21.	<p>Найдите все комедии продолжительностью более трёх часов.</p> <p>Напишите SQL запрос возвращающий результат состоящий из трёх столбцов: названия фильма, года выхода на экран и продолжительности в минутах отсортированный по длине фильма.</p>	<pre>SELECT film.title, film.release_year, film.length FROM film inner join film_category ON film.film_id = film_category.film_id INNER JOIN category ON category.category_id = film_category.category_id WHERE category.name='comedy' and film.length>180 ORDER BY film.length ASC;</pre>

22.	<p>Выберите фамилии, имена и адреса электронной почты клиентов, чьи имя и фамилия не содержат ни одной буквы «А» (латинская буква).</p> <p>Отсортируйте результат по customer_id</p>	<pre>SELECT last_name, first_name, email FROM customer WHERE last_name NOT LIKE '%A%' AND first_name NOT LIKE '%A%' ORDER BY customer_id ASC;</pre>
23.	<p>Найдите все фильмы с рейтингом NC-17 (только для взрослых), в описании которых содержится подстрока Database Administrator.</p> <p>Выведите название, описание, год выпуска этих фильмов в алфавитном порядке по названию.</p>	<pre>SELECT title, description, release_year FROM film WHERE rating='NC-17' AND description LIKE '% Database Administrator%' ORDER BY title ASC;</pre>
24.	<p>Найдите все фильмы, в описании которых есть слова Dog или Cat, отмеченные рейтингом PG или PG-13 (для просмотра под контролем родителей).</p> <p>Выведите названия, описания, годы выпуска этих фильмов, отсортировав по названию в алфавитном порядке.</p>	<pre>SELECT title, description , release_year FROM film WHERE (description LIKE '%Dog%' OR description LIKE '%Cat%') AND rating LIKE 'PG%' ORDER BY title ASC;</pre>
25.	<p>Фильмы с рейтингом R (Ограниченный доступ) и NC-17 (Только для взрослых) не могут быть взяты напрокат молодежью.</p> <p>Получите список этих фильмов в две колонки title и rating, отсортированных по названию фильма.</p> <p>Для решения этой задачи используйте условие с ключевым словом OR.</p>	<pre>SELECT title, rating FROM film WHERE rating='R' OR rating='NC-17' ORDER BY title ASC;</pre>
26.	<p>Фильмы с рейтингом PG (рекомендуется родительский контроль) и PG-13 (родители должны быть осторожны) могут просматриваться детьми только под контролем родителей.</p> <p>Получите список этих фильмов в двух столбцах title, rating, отсортированных по названию.</p>	<pre>SELECT title, rating FROM film WHERE rating LIKE 'PG%' ORDER BY title ASC;</pre>
27.	<p>Найдите всех сотрудников, занятых на проекте "Video Database".</p> <p>Напишите запрос, который выводит номер сотрудника, имя, фамилию, дату приёма на работу и код должности.</p> <p>Отсортируйте результат по фамилиям в алфавитном порядке. Если фамилии совпадают, отсортируйте по коду должности.</p>	<pre>select EMPLOYEE.EMP_NO, EMPLOYEE.FIRST_NAME, EMPLOYEE.LAST_NAME, EMPLOYEE.HIRE_DATE, EMPLOYEE.JOB_CODE from EMPLOYEE INNER JOIN EMPLOYEE_PROJECT on EMPLOYEE.EMP_NO=EMPLOYEE_PROJECT. EMP_NO INNER JOIN PROJECT on PROJECT.PROJ_ID=EMPLOYEE_PROJECT.PR OJ_ID where PROJECT.PROJ_NAME='Video Database' order by EMPLOYEE.LAST_NAME asc, EMPLOYEE.JOB_CODE asc;</pre>

28.	<p>Напишите запрос, извлекающий список всех сотрудников, работающих за пределами США.</p> <p>Результат должен содержать все столбцы таблицы EMPLOYEE.</p>	<pre>select * from EMPLOYEE where JOB_COUNTRY not like 'USA';</pre>
29.	<p>Напишите запрос, извлекающий список всех сотрудников, принятых на работу в 1992 году.</p> <p>Результат должен содержать следующие столбцы FULL_NAME - полное имя сотрудника и HIRE_DATE - дата приёма на работу. Отсортируйте результат по возрастанию даты приёма</p>	<pre>select FULL_NAME, HIRE_DATE from EMPLOYEE where HIRE_DATE like '1992%' order by HIRE_DATE asc;</pre>
30.	<p>Напишите SQL запрос, чтобы получить список фильмов, отсутствующих в прокате (таблица inventory).</p> <p>Отобразите названия этих фильмов в столбце с названием film_title в алфавитном порядке.</p> <p>Используйте для решения задачи соединение таблиц.</p>	<pre>SELECT film.title AS film_title FROM film LEFT JOIN inventory ON inventory.film_id=film.film_id WHERE inventory.film_id IS NULL ORDER BY film.title ASC;</pre>
31.	<p>Напишите SQL запрос для получения списка языков из таблицы language, на которых нет доступных фильмов.</p> <p>Представьте результат в таблице с одним столбцом - language, отсортированным по алфавиту.</p> <p>Используйте для решения задачи соединение таблиц.</p>	<pre>SELECT distinct language.name AS language FROM language LEFT JOIN film ON language.language_id=film.language_id WHERE film.language_id IS NULL ORDER BY language.name ASC;</pre>
32.	<p>Напишите SQL запрос, который выводит названия всех фильмов и их категорий из базы данных Sakila.</p>	<pre>SELECT film.title, category.name FROM film INNER JOIN film_category ON film.film_id=film_category.film_id INNER JOIN category ON film_category.category_id=category.category_id;</pre>
33.	<p>Извлеките имя и домен из адресов электронной почты клиентов в базе данных Sakila.</p> <p>Напишите запрос, возвращающий три столбца: email, address – часть адреса электронной почты перед знаком «@» и domain — часть после «@».</p> <p>Отсортируйте результат по полю email.</p>	<pre>SELECT email, SUBSTRING_INDEX(email, '@', 1) as address, SUBSTRING_INDEX(email, '@', -1) as domain FROM customer ORDER BY email ASC;</pre>
34.	<p>Получить определения столбцов таблицы address</p>	<pre>DESCRIBE address;</pre>
35.	<p>Получить список индексов таблицы film и их определений</p>	<pre>SHOW index FROM film;</pre>
36.	<p>Найдите фильмы из базы данных Sakila, для которых нет записей об участвовавших в них актёрах используя соединение таблиц JOIN.</p>	<pre>SELECT film.title, film.release_year FROM film LEFT JOIN film_actor ON film_actor.film_id=film.film_id WHERE film_actor.film_id IS NULL ORDER BY film.title ASC;</pre>

		Выведите результирующую с полями title, release_year отсортированных по названию фильма.	
37.		Найдите клиентов чьё имя является фамилией другого клиента. Выведите таблицу с полями customer_id, first_name, last_name для первого клиента и такие же поля customer_id, first_name, last_name для второго. Отсортируйте по customer_id первого клиента.	<pre> SELECT customer1.customer_id, customer1.first_name, customer1.last_name, customer2.customer_id, customer2.first_name, customer2.last_name FROM customer as customer1 INNER JOIN customer as customer2 ON customer1.first_name= customer2.last_name ORDER BY customer1.customer_id ASC; </pre>
38.		Найдите клиентов которые встречали друг друга в одном из пунктов проката. Выведите таблицу с полями meet_time - согласно времени аренды, store_id, customers список встречавшихся клиентов в формате JOHN SHOW,DAENERYS TARGARYEN - в порядке их фамилий. Результирующую таблицу отсортируйте по времени встречи и номеру пункта проката (Клиенты встречались если брали в аренду фильмы в одном отделении в одно время.)	<pre> SELECT rental.rental_date AS meet_time, staff.store_id AS store_id, GROUP_CONCAT(DISTINCT customer.first_name,', ' customer.last_name order by customer.last_name,customer.first_name separator ',') AS customers FROM staff inner join rental ON staff.staff_id=rental.staff_id INNER JOIN customer ON rental.customer_id=customer.customer_id GROUP BY rental.rental_date, staff.store_id HAVING count(*) > 1 ORDER BY rental.rental_date ASC, staff.store_id ; </pre>
39.		Напишите SQL запрос для поиска фильмов в базе данных Sakila, которые есть в наличии (в таблице inventory), но никогда не выдавались в прокат. Выведите названия этих фильмов в алфавитном порядке. Для решения задачи используйте соединение таблиц.	<pre> SELECT film.title From film INNER JOIN inventory ON film.film_id=inventory.film_id LEFT JOIN rental ON inventory.inventory_id=rental.inventory_id WHERE rental.inventory_id IS NULL ORDER BY film.title ASC; </pre>
40.		Получите все фильмы в следующих категориях: Comedy, Music и Travel . Выведите таблицу со столбцами film_id, title и category, отсортированными по film_id. Напишите запрос без использования ключевого слова OR в условии.	<pre> SELECT film.film_id, film.title, category.name AS category FROM film INNER JOIN film_category ON film.film_id=film_category.film_id INNER JOIN category ON film_category.category_id=category.category_id WHERE category.name='Comedy' UNION SELECT film.film_id, film.title, category.name as category FROM film INNER JOIN film_category ON film.film_id=film_category.film_id INNER JOIN category ON film_category.category_id=category.category_id WHERE category.name='Music' UNION SELECT film.film_id, film.title, category.name as category </pre>

		<pre> FROM film inner join film_category ON film.film_id=film_category.film_id INNER JOIN category ON film_category.category_id=category.category_id WHERE category.name='Travel' ORDER BY film_id ASC; </pre>
41.	<p>Выберите имена и фамилии клиентов, чьи имя и фамилия начинаются на одну и ту же букву.</p> <p>Отсортируйте результат по имени и фамилии.</p>	<pre> SELECT DISTINCT first_name, last_name FROM customer WHERE left(first_name, 1) = left(last_name, 1) ORDER BY first_name ASC, last_name ASC ; </pre>
42.	<p>Найдите все фильмы взятые в прокат KATIE ELLIOTT. Выведите результат в два столбца title и rating.</p> <p>Отсортируйте список так что бы сначала шли фильмы "для взрослых" (с рейтингом R), а затем все остальные по алфавиту.</p>	<pre> SELECT film.title, film.rating FROM film JOIN inventory ON film.film_id = inventory.film_id JOIN rental ON inventory.inventory_id = rental.inventory_id JOIN customer ON rental.customer_id = customer.customer_id WHERE customer.first_name = 'katie' AND customer.last_name = 'elliott' ORDER BY case WHEN film.rating = 'r' THEN 1 ELSE 2 END, film. title ASC; </pre>