Основы SQL

**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

**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

**9. Напишите SQL запрос, который выводит список уникальных значений rating из таблицы film в алфавитном порядке.**

SELECT DISTINCT rating FROM film ORDER BY rating

**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 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 WHERE film\_id BETWEEN 21 and 30 ORDER BY title

SELECT title, description, release\_year FROM film WHERE film\_id BETWEEN 21 and 30 ORDER BY film\_id

**13. Выберите название, стоимость проката и продолжительность фильмов из таблицы film. Отсортируйте полученный список по убыванию стоимости, фильмы с одинаковой стоимостью отсортируйте по возрастанию продолжительности фильма.**

SELECT title, rental\_rate, length

FROM film

ORDER BY rental\_rate DESC, length ASC

**14. Найдите самый длинный фильм в таблице film.**

**Если несколько фильмов имеют одинаковую продолжительность, выберите фильм с наименьшей ценой замены replacement\_cost**

**Напишите запрос, без использования агрегатых функций, который возвращает два столбца: title и release\_year.**

SELECT title, release\_year

FROM film

ORDER BY length DESC, replacement\_cost ASC

LIMIT 1

15. Найдите все фильмы продолжительностью более трёх часов.

Напишите SQL запрос возвращающий результат состоящий из трёх столбцов: названия фильма, его описания и продолжительности в минутах отсортированный по длине фильма.

SELECT title, description, length

FROM film

WHERE length > 180

ORDER BY length

16. Найдите сотрудников, работающих в магазине номер 1, и получите все их данные.

SELECT \*

FROM staff

WHERE store\_id = 1

17. 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.

SELECT customer\_id, first\_name, last\_name

FROM customer

WHERE active =1

18. Найдите актеров по имени Scarlett

SELECT \*

FROM actor

WHERE first\_name = 'Scarlett'

19. Найдите все фильмы, в описании которых есть слово Student. Выведите названия фильмов в алфавитном порядке.

SELECT title

FROM film

WHERE description LIKE '%student%'

ORDER BY film\_id

20. Найдите все фильмы продолжительностью более 3 часов и получите их название, год выпуска и продолжительность, отсортированные по продолжительности в порядке возрастания.

SELECT title, release\_year, length

FROM film

WHERE length > 180

ORDER BY length

21. Найдите все комедии продолжительностью более трёх часов.

Напишите SQL запрос возвращающий результат состоящий из трёх столбцов: названия фильма, года выхода на экран и продолжительности в минутах отсортированный по длине фильма.

SELECT title, release\_year, length

FROM film

WHERE film\_id IN (select film\_id FROM film\_category WHERE category\_id=5)

AND length>180

group by film\_id

order by length;

22. Выберите фамилии, имена и адреса электронной почты клиентов, чьи имя и фамилия не содержат ни одной буквы «А» (латинская буква).

Отсортируйте результат по customer\_id

SELECT last\_name, first\_name, email

FROM customer

WHERE first\_name NOT LIKE '%a%' AND last\_name NOT LIKE '%a%'

ORDER BY customer\_id

23. Найдите все фильмы с рейтингом NC-17 (только для взрослых), в описании которых содержится подстрока Database Administrator.

Выведите название, описание, год выпуска этих фильмов в алфавитном порядке по названию.

SELECT title, description, release\_year

FROM film

WHERE rating = 'NC-17' and description LIKE '%Database Administrator%'

ORDER BY film\_id

24. Найдите все фильмы, в описании которых есть слова Dog или Cat, отмеченные рейтингом PG или PG-13 (для просмотра под контролем родителей).

Выведите названия, описания, годы выпуска этих фильмов, отсортировав по названию в алфавитном порядке.

SELECT title, description, release\_year

FROM film

WHERE (description LIKE '%dog%' OR description LIKE '%cat%') AND rating IN ('PG', 'PG-13')

25. Фильмы с рейтингом R (Ограниченный доступ) и NC-17 (Только для взрослых) не могут быть взяты напрокат молодежью.

Получите список этих фильмов в две колонки title и rating, отсортированных по названию фильма.

Для решения этой задачи используйте условие с ключевым словом OR.

SELECT title, rating

FROM film

WHERE rating in('R', 'NC-17')

ORDER BY film\_id

26. Фильмы с рейтингом PG (рекомендуется родительский контроль) и PG-13 (родители должны быть осторожны) могут просматриваться детьми только под контролем родителей.

Получите список этих фильмов в двух столбцах title, rating, отсортированных по названию.

SELECT title, rating

FROM film

WHERE rating in ('pg', 'pg-13')

ORDER BY film\_id

27. Найдите всех сотрудников, занятых на проекте "Video Database".

Напишите запрос, который выводит номер сотрудника, имя, фамилию, дату приёма на работу и код должности.

Отсортируйте результат по фамилиям в алфавитном порядке. Если фамилии совпадают, отсортируйте по коду должности.

SELECT e.EMP\_NO, e.FIRST\_NAME, e.LAST\_NAME, e.HIRE\_DATE, e.JOB\_CODE

FROM EMPLOYEE e

JOIN EMPLOYEE\_PROJECT ep

ON e.EMP\_NO = ep.EMP\_NO

WHERE ep.PROJ\_ID = 'VBASE'

ORDER BY e.LAST\_NAME, e.JOB\_CODE

28. Напишите запрос, извлекающий список всех сотрудников, работающих за пределами США.

Результат должен содержать все столбцы таблицы EMPLOYEE

SELECT \*

FROM EMPLOYEE

WHERE NOT JOB\_COUNTRY = 'USA'

29. Напишите запрос, извлекающий список всех сотрудников, принятых на работу в 1992 году.

Результат должен содержать следующие столбцы FULL\_NAME - полное имя сотрудника и HIRE\_DATE - дата приёма на работу. Отсортируйте результат по возрастанию даты приёма

SELECT FULL\_NAME, HIRE\_DATE

FROM EMPLOYEE

WHERE HIRE\_DATE LIKE '1992%'

ORDER BY HIRE\_DATE

30. Напишите SQL запрос, чтобы получить список фильмов, отсутствующих в прокате (таблица inventory).

Отобразите названия этих фильмов в столбце с названиемfilm\_title в алфавитном порядке.

Используйте для решения задачи соединение таблиц.

SELECT

f.title AS film\_title

FROM

film f

LEFT JOIN

inventory i ON f.film\_id = i.film\_id

WHERE

i.inventory\_id IS NULL -- Фильмы, не арендованные

ORDER BY

f.title ASC;

31. Напишите SQL запрос для получения списка языков из таблицы language, на которых нет доступных фильмов.

Представьте результат в таблице с одним столбцом - language, отсортированным по алфавиту.

Используйте для решения задачи соединение таблиц.

SELECT name AS language

FROM language l

LEFT JOIN

film f ON l.language\_id = f.language\_id

WHERE f.film\_id IS NULL

ORDER BY l.name

32. Напишите SQL запрос, который выводит названия всех фильмов и их категорий из базы данных Sakila.

SELECT f.title, c.name

FROM film f

JOIN

film\_category fc ON f.film\_id = fc.film\_id

JOIN

category c ON fc.category\_id = c.category\_id;

33. Извлеките имя и домен из адресов электронной почты клиентов в базе данных Sakila.

Напишите запрос, возвращающий три столбца: email, address – часть адреса электронной почты перед знаком «@» и domain — часть после «@».

Отсортируйте результат по полю email.

SELECT

email,

SUBSTRING\_INDEX(email, '@', 1) AS address,

SUBSTRING\_INDEX(email, '@', -1) AS domain

FROM

customer

ORDER BY

email ASC;

34. Получить определения столбцов таблицы address

DESCRIBE address;

35. Получить список индексов таблицы film и их определений

SHOW INDEX FROM film;

36. Найдите фильмы из базы данных Sakila, для которых нет записей об учавствоваших в них актёрах используя соединение таблиц JOIN.

Выведите результирующую с полями title, release\_year отсортированных по названию фильма.

SELECT f.title, f.release\_year

FROM film f

LEFT JOIN

film\_actor fa ON f.film\_id = fa.film\_id

WHERE fa.actor\_id IS NULL

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.

SELECT

c1.customer\_id

c1.first\_name,

c1.last\_name,

c2.customer\_id,

c2.first\_name,

c2.last\_name

FROM customer c1

JOIN

customer c2 ON c1.first\_name = c2.last\_name

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.)

SELECT r.rental\_date AS meet\_time,

s.store\_id,

GROUP\_CONCAT(DISTINCT CONCAT(c.first\_name, ' ', c.last\_name) ORDER BY c.last\_name SEPARATOR ',') AS customers

FROM rental r

JOIN customer c ON r.customer\_id = c.customer\_id

JOIN staff s ON r.staff\_id = s.staff\_id

GROUP BY r.rental\_date, s.store\_id

HAVING COUNT(DISTINCT c.customer\_id) > 1

ORDER BY meet\_time, s.store\_id;

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.

SELECT f.title

FROM film f

JOIN inventory i ON f.film\_id = i.film\_id

LEFT JOIN rental r ON i.inventory\_id = r.inventory\_id

WHERE r.rental\_id IS NULL

ORDER BY f.film\_id;

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.

SELECT f.film\_id, f.title, c.name AS category

FROM film f

LEFT JOIN

film\_category fc ON f.film\_id = fc.film\_id

LEFT JOIN

category c ON fc.category\_id = c.category\_id

WHERE

c.name IN ('comedy', 'music', 'travel')

ORDER BY f.film\_id

41. Выберите имена и фамилии клиентов, чьи имя и фамилия начинаются на одну и ту же букву.

Отсортируйте результат по имени и фамилии.

SELECT first\_name, last\_name FROM customer

WHERE LEFT (first\_name, 1) = LEFT (last\_name, 1)

GROUP BY customer\_id

ORDER BY first\_name, last\_name ASC;

42. Найдите все фильмы взятые в прокат KATIE ELLIOTT. Выведите результат в два столбца title и rating.

Отсортируйте список так что бы сначала шли фильмы "для взрослых" (с рейтингом R), а затем все остальные по алфавиту.

JOIN

inventory i ON r.inventory\_id = i.inventory\_id

JOIN

film f ON i.film\_id = f.film\_id

JOIN

customer c ON r.customer\_id = c.customer\_id

WHERE

c.first\_name = 'KATIE' AND c.last\_name = 'ELLIOTT'

ORDER BY

CASE

WHEN f.rating = 'R' THEN 0

ELSE 1

END,

f.title;

Вычисления

1. Напишите запрос для вычисления длины окружности диаметром 7. Результат выведите в колонке circle\_perimeter

SELECT

7 \* PI() AS circle\_perimeter;

2. Напишите запрос для вычисления площади круга радиусом 12. Результат округлите до шести десятичных знаков и выведите в колонке circle\_area

SELECT

ROUND(PI() \* POW(12, 2), 6) AS circle\_area;

3. Найдите длину гипотенузы прямоугольного треугольника с катетами, равными 2 и 3.

Выведите результат в столбце hypotenuse. Округлите результат до трёх десятичных знаков.

SELECT

ROUND(CAST(SQRT(POW(2, 2) + POW(3, 2)) AS NUMERIC), 3) AS hypotenuse;

4. Напишите запрос возвращающий таблицу значений факториала для целых чисел от 0 до 10.

Таблица должна содержать две колонки n - число от 0 до 10 и f значение факториала этого числа

WITH RECURSIVE r AS (

SELECT 0 AS n, 1 AS f

UNION

SELECT n+1 AS f, f \* (n+1) as f

FROM r

WHERE n < 10)

SELECT \* FROM r;

5. Сформируйте список фильмов в формате JSON вида {"id": 1, "title": "ACADEMY DINOSAUR", "category": "Documentary"} в таблице с одним столбцом film отсортированным по идентификатору фильма

SELECT

JSON\_OBJECT(

'id', f.film\_id,

'title', f.title,

'category', c.name

) AS film

FROM

film f

JOIN

film\_category fc ON f.film\_id = fc.film\_id

JOIN

category c ON fc.category\_id = c.category\_id

ORDER BY

f.film\_id;

6. Получите все записи из таблицы address, где почтовый индекс представляет собой четное число. Выведите таблицу с двумя столбцами address\_id и postal\_code, отсортированными по address\_id.

SELECT address\_id, postal\_code

FROM address

WHERE MOD(CAST(postal\_code AS UNSIGNED), 2) = 0

ORDER BY address\_id;

7. Составьте общий список адресов электронной почты клиентов и персонала. Выведите таблицу со следующими столбцами:

record\_type – customer или employee, last\_name, first\_name, email — персональные данные

Отсортируйте таблицу по фамилии и затем по имени.

SELECT

'customer' AS record\_type,

c.last\_name,

c.first\_name,

c.email

FROM

customer c

UNION ALL

SELECT

'employee' AS record\_type,

s.last\_name,

s.first\_name,

s.email

FROM

staff s

ORDER BY

last\_name, first\_name;

8. SELECT

f.title,

r.rental\_date,

r.return\_date,

p.payment\_date,

f.rental\_rate,

ABS(f.rental\_rate - p.amount) AS lateness\_penalty, -- Расчёт положительного штрафа

p.amount

FROM

rental r

JOIN

customer c ON r.customer\_id = c.customer\_id

JOIN

payment p ON r.rental\_id = p.rental\_id

JOIN

inventory i ON r.inventory\_id = i.inventory\_id

JOIN

film f ON i.film\_id = f.film\_id

WHERE

c.first\_name = 'DOROTHY' AND c.last\_name = 'TAYLOR'

AND r.rental\_date BETWEEN '2005-08-01' AND '2005-08-31'

UNION ALL

SELECT

'Total' AS title,

NULL AS rental\_date,

NULL AS return\_date,

NULL AS payment\_date,

SUM(f.rental\_rate) AS rental\_rate, -- Сумма всех ставок аренды

SUM(ABS(f.rental\_rate - p.amount)) AS lateness\_penalty, -- Сумма всех штрафов

SUM(p.amount) AS amount -- Общая оплаченная сумма

FROM

rental r

JOIN

customer c ON r.customer\_id = c.customer\_id

JOIN

payment p ON r.rental\_id = p.rental\_id

JOIN

inventory i ON r.inventory\_id = i.inventory\_id

JOIN

film f ON i.film\_id = f.film\_id

WHERE

c.first\_name = 'DOROTHY' AND c.last\_name = 'TAYLOR'

AND r.rental\_date BETWEEN '2005-08-01' AND '2005-08-31';

9. Составьте список фамилий встречающихся как среди пользователей так и среди актёров. Выведите таблицу с одной колонкой last\_name отсортированной по алфавиту.

Решите задачу при помощи табличного оператора.

SELECT last\_name

FROM customer

INTERSECT

SELECT last\_name

FROM actor

ORDER BY last\_name;

10. Найдите в таблице customer имена палиндромы.

Отсортируйте результат по first\_name

SELECT first\_name

FROM customer

WHERE first\_name = REVERSE(first\_name)

ORDER BY first\_name;

11.