Запросы SQL

**Выгрузи из таблицы client поля email, first\_name, last\_name.**

SELECT email,

first\_name,

last\_name

FROM client;

**Выгрузи первые четыре поля из таблицы invoice. Ограничь выгрузку первыми пятью записями.**

*Сначала узнаем какие поля есть*

SELECT \*

FROM invoice

LIMIT 5;

**Выведи первые 20 записей из таблицы track. Оставь в итоговой таблице только два поля: name и unit\_price.**

SELECT name,

unit\_price

FROM track

LIMIT 20;

**Выгрузи из таблицы track поля milliseconds и bytes. Оба поля должны быть строками. Используй тип данных для строк нефиксированной длины.**

SELECT CAST(milliseconds AS varchar), CAST(bytes AS varchar)

FROM track;

**Выгрузи из таблицы invoice поле total. Оставь только целую часть числа**

SELECT CAST(total AS integer)

FROM invoice;

**Выгрузи из таблицы staff дни рождения сотрудников. Необходимые значения в виде строк хранит поле birth\_date. Сформируй запрос так, чтобы на экране отобразилась только дата.**

SELECT CAST(birth\_date AS date)

FROM staff;

**Нужно выгрузить данные о заказах на сумму больше или равно 1.2. Информацию о счетах хранит таблица invoice, а поле total в таблице содержит данные о сумме заказа**

SELECT \*

/\* Нужно выгрузить всю информацию о заказах,

поэтому поля можно не указывать\*/

FROM invoice

WHERE total >= 1.2;

**Изучи таблицу invoice\_line, которая хранит информацию о купленных треках. Выгрузи все значения из таблицы, отфильтровав треки дороже 0.99. Нужные значения хранит столбец unit\_price.**

SELECT invoice\_line\_id,

invoice\_id,

track\_id,

unit\_price,

quantity

FROM invoice\_line

WHERE unit\_price > 0.99;

**Из таблицы client выгрузи несколько полей: first\_name с именем покупателя, last\_name с фамилией и city, в котором указан город проживания. Оставь записи только о тех покупателях, которые живут в Бразилии (англ. Brazil). Страна проживания хранится в поле country**

SELECT first\_name,

last\_name,

city

FROM client

WHERE country = 'Brazil';

**Нужно проверить, где и когда совершали самые крупные покупки. Из таблицы invoice выгрузи поле billing\_address, оно хранит адрес выставления счёта, и поле invoice\_date с датой покупки. Дата указана в формате 'YYYY-MM-DD HH:MM:SS'. Нужно оставить только дату 'YYYY-MM-DD'. Отфильтруй записи, в которых значение поля total больше или равно 8.**

SELECT billing\_address,

CAST(invoice\_date AS date)

FROM invoice

WHERE total >= 8;

**Теперь попробуй отфильтровать данные о счетах — их хранит таблица invoice. Выгрузи из таблицы поле с суммой заказа total и поле с идентификатором покупателя customer\_id. Счета должны быть оформлены в трёх городах: Дублине (англ. Dublin), Лондоне (англ. London) или Париже (англ. Paris). Информацию о городе хранит поле billing\_city.**

SELECT total,

customer\_id

FROM invoice

WHERE billing\_city = 'Dublin'

OR billing\_city = 'London'

OR billing\_city = 'Paris';

**Из таблицы invoice выгрузи поля total и customer\_id. Значение total должно быть больше или равно 5, а значение customer\_id должно равняться 40 или 46**

SELECT total,

customer\_id

FROM invoice

WHERE total >= 5

AND (customer\_id = 40 OR customer\_id = 46);

**Теперь объедини условия. Из таблицы invoice выгрузи поля total и customer\_id. Счёт должен быть оформлен в Дублине (англ. Dublin), Лондоне (англ. London) или Париже (англ. Paris). Значение total должно быть больше или равно 5, а значение customer\_id должно равняться 40 или 46.**

SELECT total,

customer\_id

FROM invoice

WHERE (billing\_city = 'Dublin'OR billing\_city = 'London'OR billing\_city = 'Paris')

AND total >= 5

AND (customer\_id = 40 OR customer\_id = 46);

**Выгрузи фамилии и номера телефонов пользователей из таблицы client. В итоговую таблицу должны войти данные о пользователях, которые живут в США (англ. USA) или Франции (англ. France). Убедись, что пользователь обращался в поддержку к сотруднику с идентификатором 3.**

SELECT last\_name,

phone

FROM client

WHERE (country = 'USA' OR country = 'France')

AND support\_rep\_id = 3;

**Потренируйся работать с диаграммой и описанием базы. Выгрузи адреса и города оформления заказов, сделанных в сентябре 2009 года. Отбери данные о заказах, оформленных во всех странах, кроме США и Бразилии. Стоимость заказа должна быть больше двух долларов.**

SELECT billing\_address,

billing\_city

FROM invoice

WHERE (billing\_country != 'USA' AND billing\_country != 'Brazil')

AND total > 2

AND (invoice\_date >= '2009-09-01 00:00:00' AND invoice\_date < '2009-10-01 00:00:00')

**Нужно отобрать фамилии сотрудников, родившихся в период с 1 января 1970 года по 31 декабря 1970 года.**

SELECT last\_name

FROM staff

WHERE birth\_date BETWEEN '1970-01-01' AND '1970-12-31';

**Выгрузи из таблицы playlist все названия плейлистов из поля name, в которых есть подстрока 'Classic'.**

SELECT name

FROM playlist

WHERE name LIKE '%Classic%';

**Выгрузи из таблицы invoice адреса из поля billing\_address и страну из поля billing\_country. Отфильтруй данные по стране: оставьте только США (англ. USA), Индию (англ. India), Канаду (англ. Canada), Аргентину (англ. Argentina) и Францию (англ. France).**

SELECT billing\_address,

billing\_country

FROM invoice

WHERE billing\_country IN ('USA',

'India',

'Canada',

'Argentina',

'France');

**Дополни предыдущий запрос. Исключи из городов выставления счёта billing\_city Редмонд (англ. Redmond), Лион (англ. Lyon) и Дели (англ. Delhi). Само поле billing\_city выводить не нужно.**

SELECT billing\_address,

billing\_country

FROM invoice

WHERE billing\_country IN ('USA',

'India',

'Canada',

'Argentina',

'France')

AND billing\_city NOT IN ('Redmond',

'Lyon',

'Delhi');

**Выгрузи все поля из таблицы invoice. Информацию о дате покупки хранит поле invoice\_date: выбери только те заказы, которые были оформлены в период с '2009-03-04' по '2012-02-09' включительно. Обрати внимание, что у поля invoice\_date тип данных varchar.**

**Сумма покупки total должна быть меньше 5. Кроме того, исключите из запроса Канаду (англ. Canada), Бразилию (англ. Brazil) и Финляндию (англ. Finland). Название страны содержит поле billing\_country.**

*Преобразовать invoice\_date пришлось в WHERE , а через CAST создался новый столбец, хоть его и не видно*

SELECT \*

FROM invoice

WHERE DATE (invoice\_date) BETWEEN '2009-03-04' AND '2012-02-09'

AND total <5

AND billing\_country NOT IN ('Canada',

'Brazil',

'Finland');

**Выгрузи из таблицы movie название фильмов, описание которых заканчивается подстрокой Mexico. Аренда фильмов из итоговой таблицы должна составлять меньше двух долларов либо их рейтинг не должен быть PG-13.**

SELECT title

FROM movie

WHERE description LIKE '%Mexico'

AND (rental\_rate < 2 OR rating !='PG-13');

**Из таблицы с треками отбери названия композиций, которые соответствуют условиям. В итоговую таблицу должны войти:**

* **треки длиннее 300000 миллисекунд, в создании которых принимал участие Bono, с идентификаторами жанра 7, 8, 9, 10;**
* **либо треки, размер которых превышает 1000000000 байт.**

SELECT name

FROM track

WHERE (milliseconds > 300000

AND composer LIKE '%Bono%'

AND genre\_id IN (7,8,9,10))

OR bytes > 1000000000;

**Выдели категории в таблице staff, которая хранит информацию о сотрудниках. Выведи на экран поля last\_name, first\_name и title. Категории нужно выделить на основе значений в поле title — оно содержит информацию о должности сотрудника:**

* **Если в title встречается слово 'IT', в новом поле будет отображена категория 'разработка'.**
* **Если в title встречается слово 'Manager' и нет слова 'IT', в новом поле отобразится категория 'отдел продаж'.**
* **Если в title встречается слово 'Support', в новом поле появится категория 'поддержка'.**

SELECT last\_name,

first\_name,

title,

CASE

WHEN title LIKE '%IT%' THEN 'разработка'

WHEN title LIKE '%Manager%' AND title NOT LIKE '%IT%' THEN 'отдел продаж'

WHEN title LIKE '%Support%' THEN 'поддержка'

END

FROM staff;

**Теперь попробуй разделить на категории фильмы из таблицы movie. Выведи на экран поля title и rental\_rate. Выделить категории нужно по цене аренды фильма:**

* **Если значение rental\_rate меньше 1, в новом поле появится категория 'категория 1'.**
* **Если значение rental\_rate больше либо равно 1, но меньше 3, в новом поле отобразится категория 'категория 2'.**
* **Если значение rental\_rate больше либо равно 3, в новом поле появится категория 'категория 3'.**

SELECT title,

rental\_rate,

CASE

WHEN rental\_rate < 1 THEN 'категория 1'

WHEN rental\_rate >= 1 AND rental\_rate < 3 THEN 'категория 2'

WHEN rental\_rate >= 3 THEN 'категория 3'

END

FROM movie;

**Из таблицы invoice выгрузи поле billing\_city с городами оформления счёта. Выгрузи только те записи, в которых на месте индекса стоит пропуск. Данные с индексами хранит поле billing\_postal\_code.**

SELECT billing\_city

FROM invoice

WHERE billing\_postal\_code IS NULL;

**Дополни предыдущий запрос. Исключи из выдачи записи с пропусками в поле billing\_state. Выбери записи, в которых сумма заказа в поле total не ниже 15 долларов.**

SELECT billing\_city

FROM invoice

WHERE (billing\_postal\_code IS NULL

AND billing\_state IS NOT NULL)

AND total >= 15;

**Найди треки длиннее 250000 миллисекунд, в названии которых есть слово Moon, но автор трека не указан.**

**Выгрузи номера альбомов, в которых находятся выбранные треки.**

SELECT album\_id

FROM track

WHERE milliseconds > 250000

AND name LIKE '%Moon%'

AND composer IS NULL;

**Отобрази имена, фамилии и страну проживания для покупателей, которые не указали информацию о месте работы, регионе проживания, а также телефон и факс.**

SELECT first\_name,

last\_name,

country

FROM client

WHERE company IS NULL

AND state IS NULL

AND phone IS NULL

AND fax IS NULL;

**Теперь можно посчитать суммарную выручку, а также найти самый маленький и самый крупный заказы.**

SELECT SUM(total),

MIN(total),

MAX(total)

FROM invoice;

**Перед тем как применить агрегирующие функции, нужно получить срез данных. Выгрузи все поля из таблицы invoice, выбери записи о заказах, оформленных в сентябре 2009 года. Информацию о дате заказа содержит поле invoice\_date. Тип данных поля — varchar.**

SELECT \*

FROM invoice

WHERE DATE (invoice\_date) BETWEEN '2009-09-01' AND '2009-09-30';

**Дополни запрос. Оставь в таблице поля с идентификатором покупателя (поле customer\_id), датой заказа без времени (поле invoice\_date) и суммой заказа (поле total). Выбери покупателей с идентификаторами 11, 13, 44, 36, 48, 52, 54, 56.**

SELECT customer\_id,

total,

CAST(invoice\_date AS date)

FROM invoice

WHERE DATE (invoice\_date) BETWEEN '2009-09-01' AND '2009-09-30'

AND customer\_id IN (11, 13, 44, 36, 48, 52, 54, 56);

**Дополни запрос. Найди минимальное и максимальное значения поля total. Условия для среза остаются прежними.**

SELECT MIN(total),

MAX(total)

FROM invoice

WHERE CAST(invoice\_date AS date) BETWEEN '2009-09-01' AND '2009-09-30' AND customer\_id in (11, 13, 44, 36, 48, 52, 54, 56);

**Добавь к минимуму и максимуму несколько новых полей:**

* **среднее значение поля total;**
* **количество уникальных идентификаторов покупателей из поля customer\_id;**
* **суммарная выручка.**

**Не меняй условия для среза данных.**

SELECT MIN (total),

MAX (total),

AVG (total),

COUNT (DISTINCT customer\_id),

SUM (total)

FROM invoice

WHERE CAST (invoice\_date AS date) BETWEEN '2009-09-01' AND '2009-09-30' AND customer\_id in (11, 13, 44, 36, 48, 52, 54, 56);

**Посчитай, сколько пропусков содержит поле fax из таблицы client.**

SELECT COUNT (fax IS NULL)

FROM client

WHERE fax IS NULL;

**Теперь можно выгрузить среднюю сумму заказов для каждого города с помощью группировки.**

SELECT billing\_city,

AVG(total)

FROM invoice

GROUP BY billing\_city

LIMIT 5;

**Группировка помогает сравнивать данные. Но сначала нужно получить срез. Напиши запрос, который выгрузит общую выручку (поле total) в США (англ. USA). Информацию о стране хранит поле billing\_country.**

SELECT SUM (total)

FROM invoice

WHERE billing\_country = 'USA';

**Теперь можно проверить, как отличаются данные по городам. Посчитай общую выручку, количество заказов, среднюю выручку для каждого города США. Нужное поле — billing\_city.**

SELECT SUM (total),

COUNT (invoice\_id),

AVG (total),

billing\_city

FROM invoice

WHERE billing\_country = 'USA'

GROUP BY billing\_city;

**Посчитай, сколько пользователей с почтовыми доменами yahoo и gmail обслужил каждый сотрудник. В итоговой таблице должны быть два поля:**

* **идентификатор сотрудника,**
* **количество пользователей.**

SELECT support\_rep\_id,

COUNT (customer\_id)

FROM client

WHERE email LIKE '%yahoo%' OR email LIKE '%gmail%'

GROUP BY support\_rep\_id

**Отбери пять самых крупных заказов из таблицы invoice.**

SELECT \*

FROM invoice

ORDER BY total DESC

LIMIT 5;

**Отбери пятерых самых активных клиентов в США с 25 мая 2011 по 25 сентября 2011. Дату хранит поле invoice\_date, тип данных поля — varchar.**

SELECT customer\_id,

COUNT(customer\_id)

FROM invoice

WHERE CAST (invoice\_date AS date) BETWEEN '2011-05-25' AND '2011-09-25'

AND billing\_country = 'USA'

GROUP BY customer\_id

ORDER BY COUNT(customer\_id) DESC

LIMIT 5;

**Сгруппировав данные по пользователям, можно применить оператор HAVING, чтобы отфильтровать нужные записи. Результат отсортирован по убыванию суммы выручки.**

SELECT customer\_id,

SUM(total)

FROM invoice

GROUP BY customer\_id

HAVING SUM(total) > 41

ORDER BY SUM(total) DESC;

**Сравни фильмы разных возрастных рейтингов. Найди среднее значение цены аренды фильма в поле rental\_rate для каждого рейтинга (поле rating). Оставь в таблице только те записи, в которых среднее значение rental\_rate больше 3.**

SELECT rating,

AVG (rental\_rate)

FROM movie

GROUP BY rating

HAVING AVG (rental\_rate) > 3;

**Изучи заказы, которые оформили в сентябре 2011 года. Сравни общую сумму выручки (поле total) за каждый день этого месяца: выведи день в формате '2011-09-01' и сумму. Информацию о дате заказа хранит поле invoice\_date. Не забудь изменить тип данных в этом поле, чтобы использовать операторы для работы с датой.**

**Оставь в таблице только те значения суммы, которые больше 1 и меньше 10.**

SELECT SUM (total),

CAST (invoice\_date AS date)

FROM invoice

WHERE CAST (invoice\_date AS date) BETWEEN '2011-09-01' AND '2011-09-30'

GROUP BY invoice\_date

HAVING SUM (total) > 1 AND SUM (total) <10;

**Посчитай пропуски в поле с почтовым индексом billing\_postal\_code для каждой страны (поле billing\_country). Получи срез: в таблицу должны войти только те записи, в которых поле billing\_address содержит слова Street, Way, Road или Drive.**

**Отобрази в таблице страну и число пропусков, если их больше 6.**

SELECT COUNT (billing\_postal\_code IS NULL),

billing\_country

FROM invoice

WHERE (billing\_address LIKE '%Street%' OR billing\_address LIKE '%Way%' OR billing\_address LIKE '%Road%' OR billing\_address LIKE '%Drive%')

AND billing\_postal\_code IS NULL

GROUP BY billing\_country

HAVING COUNT (billing\_postal\_code IS NULL) > 6;

**Нужно объединить данные двух таблиц: track и invoice\_line. Таблица track хранит информацию о музыкальных треках в магазине, названия треков указаны в поле name. Таблица invoice\_line содержит данные о купленных треках, их стоимость указана в поле unit\_price. В обеих таблицах есть поле track\_id — в нём содержатся идентификаторы музыкальных треков.**

**Выгрузи таблицу, в которой названию трека будет соответствовать его стоимость. Отбери все уникальные записи. Если какой-либо из треков не покупали или у купленного трека нет названия — такие записи не должны войти в таблицу. Оставь в итоговой таблице первые 20 записей.**

SELECT DISTINCT t.name,

i.unit\_price

FROM track AS t

INNER JOIN invoice\_line AS i ON t.track\_id = i.track\_id

LIMIT 20;

**Нужно дополнить запрос: добавь поле с идентификатором плейлиста playlist\_id. Такое поле можно получить из таблицы playlist\_track. В этой таблице собраны идентификаторы плейлистов и треков (поле track\_id). Условие остаётся прежним: если идентификаторы треков не совпадают во всех трёх таблицах, такие треки не должны войти в итоговую таблицу. Выведи первые 20 записей.**

SELECT t.name,

i.unit\_price,

p. playlist\_id

FROM track AS t

INNER JOIN invoice\_line AS i ON t.track\_id = i.track\_id

INNER JOIN playlist\_track AS p ON t.track\_id = p.track\_id

LIMIT 20;

**Идентификатор плейлиста теперь указан в итоговой таблице. Но что это за плейлисты — непонятно. Эту информацию можно взять в четвёртой таблице — playlist. Таблица содержит поле playlist\_id с идентификатором плейлиста и поле name — с его названием. Добавь в итоговую таблицу поле name. Условия те же: данные без совпадения не должны попасть в таблицу. Ограничь вывод первыми 20 записями.**

SELECT t.name,

i.unit\_price,

pt.playlist\_id,

pl.name

FROM track AS t

INNER JOIN invoice\_line AS i ON t.track\_id = i.track\_id

INNER JOIN playlist\_track AS pt ON t.track\_id = pt.track\_id

INNER JOIN playlist AS pl ON pt.playlist\_id = pl.playlist\_id

LIMIT 20;

**Выведи названия всех треков, добавив информацию о датах, в которые эти треки покупали. Ни один трек не должен потеряться, даже если его не покупали вообще. Чтобы получить нужный результат, надо соединить три таблицы сразу, ведь таблица invoice, которая хранит данные о дате заказа, не содержит информации о купленных треках.**

**Сначала соедини таблицы track и invoice\_line по ключу track\_id, а затем присоедини таблицу invoice по ключу invoice\_id. В итоговую таблицу помести два поля: name из таблицы track и invoice\_date из таблицы invoice. Приведи дату в нужный формат.**

SELECT t.name,

DATE (ind.invoice\_date)

FROM track AS t

LEFT OUTER JOIN invoice\_line AS inl ON t.track\_id = inl.track\_id

LEFT OUTER JOIN invoice AS ind ON inl.invoice\_id = ind.invoice\_id;

Или

SELECT t.name,

CAST (ind.invoice\_date AS DATE)

FROM track AS t

LEFT OUTER JOIN invoice\_line AS inl ON t.track\_id = inl.track\_id

LEFT OUTER JOIN invoice AS ind ON inl.invoice\_id = ind.invoice\_id;

**Выгрузи таблицу из двух полей: первое поле с фамилией сотрудника, второе — с количеством пользователей, чьи запросы этот сотрудник обработал. Назови поля employee\_last\_name и all\_customers соответственно. Сгруппируй записи по идентификатору сотрудника. Отсортируй количество пользователей по убыванию.**

SELECT s.last\_name AS employee\_last\_name,

COUNT (c.customer\_id) AS all\_customers

FROM staff AS s

LEFT OUTER JOIN client AS c ON s.employee\_id = c.support\_rep\_id

GROUP BY s.employee\_id

ORDER BY all\_customers DESC;

**Отобрази названия фильмов, в которых снимались актёры и актрисы, не указанные в базе.**

SELECT m.title

FROM movie AS m

LEFT OUTER JOIN film\_actor AS fa ON m.film\_id = fa.film\_id

**ПРОЕКТ**

1. Посчитай, сколько компаний закрылось.

SELECT COUNT (name)

FROM company

WHERE status = 'closed';

1. Отобрази количество привлечённых средств для новостных компаний США. Используй данные из таблицы company. Отсортируй таблицу по убыванию значений в поле funding\_total.

SELECT funding\_total

FROM company

WHERE category\_code = 'news'

AND country\_code = 'USA'

ORDER BY funding\_total DESC;

1. Отобрази имя, фамилию и названия аккаунтов людей в поле network\_username, которые начинаются на 'Silver'.

SELECT first\_name,

last\_name,

network\_username

FROM people

WHERE network\_username LIKE 'Silver%';

1. Выведи на экран всю информацию о людях, у которых названия аккаунтов в поле network\_username содержат подстроку 'money', а фамилия начинается на 'K'.

SELECT \*

FROM people

WHERE network\_username LIKE '%money%'

AND last\_name LIKE 'K%';

1. Для каждой страны отобрази общую сумму привлечённых инвестиций, которые получили компании, зарегистрированные в этой стране. Страну, в которой зарегистрирована компания, можно определить по коду страны. Отсортируй данные по убыванию суммы.

SELECT country\_code,

SUM (funding\_total)

FROM company

GROUP BY country\_code

ORDER BY SUM (funding\_total) DESC;

1. Отобрази имя и фамилию всех сотрудников стартапов. Добавь поле с названием учебного заведения, которое окончил сотрудник, если эта информация известна.

SELECT p.first\_name,

p.last\_name,

e.instituition

FROM people AS p

LEFT OUTER JOIN education AS e ON p.id = e.person\_id

1. Найди общую сумму сделок по покупке одних компаний другими в долларах. Отбери сделки, которые осуществлялись только за наличные с 2011 по 2013 год включительно.

SELECT SUM (price\_amount)

FROM acquisition

WHERE term\_code = 'cash'

AND acquired\_at BETWEEN '2011-01-01' AND '2013-12-31'

1. Выясни, в каких странах находятся фонды, которые чаще всего инвестируют в стартапы.

Для каждой страны посчитай минимальное, максимальное и среднее число компаний, в которые инвестировали фонды этой страны, основанные с 2010 по 2012 год включительно. Исключи страны с фондами, у которых минимальное число компаний, получивших инвестиции, равно нулю.

Выгрузи десять самых активных стран-инвесторов: отсортируй таблицу по среднему количеству компаний от большего к меньшему. Затем добавь сортировку по коду страны в лексикографическом порядке.

SELECT country\_code,

MIN (invested\_companies),

MAX (invested\_companies),

AVG (invested\_companies)

FROM fund

WHERE founded\_at BETWEEN '2010-01-01' AND '2011-12-31'

GROUP BY country\_code

HAVING MIN (invested\_companies) > 0

ORDER BY AVG (invested\_companies) DESC, country\_code ASC

LIMIT 10;