

Используя все знания полученные на последних двух лекциях составьте 5 запросов в БД.

Создание таблиц происходило в приложении MySQL Workbench. Сперва была создана база данных MyDatabase, а уже в ней были созданы три нужные таблицы. Проверим, что таблицы создались и посмотрим их содержимое. Таблицы существуют, с ними все хорошо.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows 'MyDatabase' expanded, with 'Tables' selected. Below it, a list of tables includes 'Client', 'Firm', and 'Orders'. The main area displays three SQL queries: 'SELECT * FROM Client;', 'SELECT * FROM Firm;', and 'SELECT * FROM Orders;'. Below the queries, the 'Result Grid' shows the data for the 'Client' table. The table has 8 rows of data. At the bottom, there are tabs for 'Client 7', 'Firm 8', and 'Orders 9'.

ID	Name	Surname	Firm_ID	Gender	Birthday	EMail
1	Алина	Астафьева	NULL	F	1993-04-05	A.Astafieva@mail.ru
2	Егор	Воронцов	1	M	1990-12-02	E.Vorontsov@mail.ru
3	Галина	Семенова	1	F	1993-05-13	G.Semenova@mail.ru
4	Вадим	Воронцов	1	M	1990-12-02	NULL
5	Даниил	Шершавый	2	M	1991-12-05	NULL
6	Елена	Васильева	NULL	F	1991-10-20	Vasilena@mail.ru
7	Ярик	Штейнмаер	2	M	1990-12-02	ya.steinmeier@mail.ru
8	Геворг	Воронцов	NULL	M	1995-11-02	NULL

Необходимо было выполнить следующие задания:

1. Мы отсылаем поздравительные сообщения по электронной почте всем клиентам Компании на 23 Февраля и 8 Марта. Одним запросом определите количество поздравлений, которые следует отправить отдельно 23 Февраля и 8 Марта.

Так как нам просто нужно количество поздравлений, а не данные клиентов, которым нужно отправить эти поздравления, достаточно будет посчитать количество клиентов и клиенток компании и вывести эти данные.

Это можно сделать с помощью CASE WHEN, где просто посчитать мужчин и женщин в отдельные столбцы. В результате получаем два столбца с количеством людей, которых поздравляем 23 февраля (мужчин), и людей,

которых поздравляем с 8 марта (женщин). Так как мы берем информацию из таблицы клиентов, они там все уникальные, и повторов не будет.

```
5 • SELECT
6     COUNT(CASE WHEN Gender = 'M' THEN 1 END) AS 23_февраля,
7     COUNT(CASE WHEN Gender = 'F' THEN 1 END) AS 8_марта
8 FROM Client;
```

100%

13:8

Result Grid

Filter Rows:

Search

Export:

23_февраля	8_марта	
5	3	

2. Вывести список заказов клиентов: [Фамилия + имя, название фирмы, дата заказа, сумма заказа], отсортированный по фамилии, имени, дате заказа.

С помощью CONCAT объединим столбцы фамилии и имени. Столбцы названия фирмы, даты заказа и суммы заказов возьмем из таблиц Orders и Firm соответственно с помощью JOIN. Для заказов возьмем просто INNER JOIN, так как ожидается, что каждому заказу соответствует свой клиент, и значений NULL нет. Для фирм подойдет LEFT JOIN, так как некоторым клиентам не соответствует фирма (как можно увидеть в выводе) и мы потеряем данные о клиентах в выводе.

ORDER BY указывает, в каком порядке сортировать данные. По умолчанию это ASC.

```
7 • SELECT
8     CONCAT(C.Surname, ' ', C.Name) AS фамилия_имя,
9     F.Name AS название_фирмы,
10    O.Ord_Time AS дата_заказа,
11    O.Amount AS сумма_заказа
12 FROM
13     Client C
14 INNER JOIN
15     Orders O ON O.Client_ID = C.ID
16 LEFT JOIN
17     Firm F ON C.Firm_ID = F.ID
18 ORDER BY
19     C.Surname, C.Name, O.Ord_Time;
```

0%	39:5	result Grid	Filter Rows:	Search	Export:
фамилия_имя	название_фирмы	дата_заказа	сумма_заказа		
Астафьева Алина	NULL	2023-12-01 11:35:45	114212		
Астафьева Алина	NULL	2023-12-02 12:25:56	314511		
Астафьева Алина	NULL	2023-12-03 14:05:18	669611		
Васильева Елена	NULL	2023-12-06 19:15:03	449020		
Васильева Елена	NULL	2024-01-10 19:15:03	37000		
Воронцов Вазген	ООО "Вареные Раки"	2023-12-04 16:45:30	148484		
Воронцов Егор	ООО "Вареные Раки"	2023-12-02 13:15:07	349142		
Воронцов Егор	ООО "Вареные Раки"	2023-12-06 18:25:52	650201		
Семенова Галина	ООО "Вареные Раки"	2023-12-04 15:55:29	71619		
Шершавый Даниил	ПАО "Раки Жаренные"	2023-12-05 17:35:41	250021		
Штейнмаер Ярик	ПАО "Раки Жаренные"	2024-01-11 10:15:03	22300		

Исправление:

Используем CASE WHEN, чтобы отфильтровать NULL-значения и написать вместо них “физ.лицо”.

```

7 • USE MyDatabase;
8
9 • SELECT
10     CONCAT(C.Surname, ' ', C.Name) AS фамилия_имя,
11     CASE
12         WHEN F.Name IS NULL THEN 'физ.лицо'
13         ELSE F.Name
14     END AS название_фирмы,
15     O.Ord_Time AS дата_заказа,
16     O.Amount AS сумма_заказа
17 FROM
18     Client C
19 INNER JOIN
20     Orders O ON O.Client_ID = C.ID
21 LEFT JOIN
22     Firm F ON C.Firm_ID = F.ID
23 ORDER BY
24     C.Surname, C.Name, O.Ord_Time

```

100% 1:6

Result Grid Filter Rows: Search Export:

	фамилия_имя	название_фирмы	дата_заказа	сумма_заказа	
	Астафьева Алина	физ.лицо	2023-12-01 11:35:45	114212	
	Астафьева Алина	физ.лицо	2023-12-02 12:25:56	314511	
	Астафьева Алина	физ.лицо	2023-12-03 14:05:18	669611	
	Васильева Елена	физ.лицо	2023-12-06 19:15:03	449020	
	Васильева Елена	физ.лицо	2024-01-10 19:15:03	37000	
	Воронцов Вазген	ООО "Вареные Раки"	2023-12-04 16:45:30	148484	
	Воронцов Егор	ООО "Вареные Раки"	2023-12-02 13:15:07	349142	
	Воронцов Егор	ООО "Вареные Раки"	2023-12-06 18:25:52	650201	
	Семенова Галина	ООО "Вареные Раки"	2023-12-04 15:55:29	71619	
	Шершавый Даниил	ПАО "Раки Жаренные"	2023-12-05 17:35:41	250021	
	Штейнмаер Ярик	ПАО "Раки Жаренные"	2024-01-11 10:15:03	22300	

- Если заказ выполнен в день рождения клиента, компания хочет ввести скидку в размере 20% от суммы заказа. Требуется определить общую сумму скидки по всем клиентам за 2023 год, как если бы скидка действовала.

Считаем сумму скидки, умножив на 0.2. INNER JOIN объединяет таблицы по столбцу id клиента. Выбираем также 2023 год и проверяем, что дата заказа совпадает с датой рождения клиента. Получилось, что никто из клиентов не делал заказ в свой день рождения.

```

1 • SELECT
2     SUM(O.Amount * 0.2) AS сумма_скидки
3 FROM
4     Orders O
5 INNER JOIN
6     Client C ON O.Client_ID = C.ID
7 WHERE
8     YEAR(O.Ord_Time) = 2023
9     AND DATE(O.Ord_Time) = DATE(C.Birthday);

```

100%	45:9	
Result Grid	Filter Rows: <input type="text" value="Search"/>	Export:
сумма_скидки		
NULL		

Исправление:

Теперь сравниваем день и месяц, а не всю дату.

```
1 USE MyDatabase;
2
3 SELECT
4     SUM(O.Amount * 0.2) AS сумма_скидки
5 FROM
6     Orders O
7 INNER JOIN
8     Client C ON O.Client_ID = C.ID
9 WHERE
10    YEAR(O.Ord_Time) = 2023
11    AND DAY(O.Ord_Time) = DAY(C.Birthday)
12    AND MONTH(O.Ord_Time) = MONTH(C.Birthday);
```

100% 28:10

Result Grid Filter Rows: Search Export:

сумма_скидки
119832.6

4. Перед Новым 2024-м Годом решено отправить подарки руководителям фирм, сотрудники которых за год в общей сложности сделали заказов более чем на 900 000 руб. Необходимо подготовить такой список:
[Название фирмы, адрес, сумма заказов].

Выбираем название фирмы и адрес из таблицы Firm и сумму оплат заказов из таблицы Orders. Используя ее как основную, объединяем с остальными таблицами в помощью INNER JOIN (так как нам нужны все данные и хоть NULL присутствует в столбце Firm_ID в таблице клиентов, нам нужны именно фирмы, а значит, клиентов, не относящихся к фирмам, можно не учитывать). Отсекаем заказы не за 2023 год и группируем по названию и адресу фирмы. В конце отсекаем фирмы, не подходящие условию в задании. Получим одну фирму. Если убрать HAVING, то можно увидеть, что у второй фирмы сумма заказов действительно меньше 900000.

```

5 • SELECT
6     F.Name AS название_фирмы,
7     F.Address AS адрес_фирмы,
8     SUM(O.Amount) AS сумма_заказов
9 FROM
10    Orders O
11 INNER JOIN
12    Client C ON O.Client_ID = C.ID
13 INNER JOIN
14    Firm F ON C.Firm_ID = F.ID
15 WHERE
16    YEAR(O.Ord_Time) = 2023
17 GROUP BY
18    F.Name, F.Address
19 HAVING
20    SUM(O.Amount) > 900000;

```

00% 64:3

Result Grid Filter Rows: Search Export:

название_фирмы	адрес_фирмы	сумма_заказов
ООО "Вареные Раки"	Уфа, ул. Караидельская, 22	1219446

5. В рамках одного SQL запроса вам необходимо подготовить список мужчин – теоретически возможных близнецов среди всех клиентов (с совпадением фамилии и даты рождения): [ID клиента, фамилия, имя, день рождения].




Два раза обращаемся к таблице Clients как к первому и второму клиенту. Объединяем по фамилии и дню рождения, отсекая объединение клиента с самим собой. Также добавляем условие, что это должны быть мужчины. В итоге получаем двух возможных близнецов.


```

1 • SELECT
2     c1.ID,
3     c1.Surname,
4     c1.Name,
5     c1.Birthday
6 FROM
7     Client c1
8 INNER JOIN
9     Client c2 ON c1.Surname = c2.Surname
10        AND c1.Birthday = c2.Birthday
11        AND c1.ID != c2.ID
12 WHERE
13     c1.Gender = 'M';

```

100% 21:13

Result Grid   Filter Rows: Export: 

	ID	Surname	Name	Birthday	
	4	Воронцов	Вазген	1990-12-02	
	2	Воронцов	Егор	1990-12-02	