## Домашнее задание

1) База данных музыкального сайта состоит из 4 таблиц: **genres**, **artists**, **albums** и **songs**. Получите все песни в жанрах Rock и Metal с 2008 по 2010 год включительно. Выведите id песни, её название, а также название альбома в поле **album** и название исполнителя в поле **artist**. Данные отсортируйте по названию композиции.

```
SELECT
    songs.id AS song_id,
    songs.name AS song name,
    albums.name AS album,
    artists.name AS artist
FROM
    songs
JOIN
    albums ON songs.album_id = albums.id
JOIN
    artists ON albums.artist_id = artists.id
JOIN
   genres ON artists.genre_id = genres.id
WHERE
   genres.name IN ('Rock', 'Metal')
   AND albums.year BETWEEN 2008 AND 2010
ORDER BY
    songs.name;
```

	☐ song_id ‡	□ song_name	<b>‡</b>	□ album	<b>‡</b>	□ artist ‡
1	13	Cyanide		Death Magnetic		Metallica
2	3	Know Your Enemy		21st Century Breakdown		Green Day
3	11	Peacemaker		21st Century Breakdown		Green Day
4	18	Tears Into Wine		Billy Talent III		Billy Talent
5	6	The Day That Never Comes		Death Magnetic		Metallica
6	16	Turn Your Back		Billy Talent III		Billy Talent
7	22	White Sparrows		Billy Talent III		Billy Talent

2) База данных музыкального сайта состоит из 4 таблиц: genres, artists, albums и songs. Получите количество композиций жанры каждом ИЗ них. Жанры выведите количество композиций В поле name. songs. отсортируйте Данные ПО имени жанра. Жанры, в которых нет композиций, выводить не надо.

SELECT gn.name AS genre, COUNT(s.id) AS song\_count FROM genres gn

LEFT JOIN artists a ON gn.id = a.genre\_id

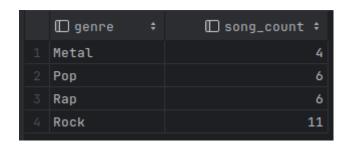
LEFT JOIN albums al ON a.id = al.artist\_id

LEFT JOIN songs s ON al.id = s.album\_id

GROUP BY gn.id

HAVING COUNT(s.id) > 0

ORDER BY gn.name;



- 3) В CRM системе компании есть 4 таблицы:
- 1. **managers** менеджеры (сотрудники) компании;
- 2. **companies** контрагенты компании;
- 3. **clients** сотрудники контрагентов;
- 4. **calls** звонки сотрудников контрагентов менеджерам компании.

В таблице также фиксируется дата (date) и продолжительность (duration\_sec) звонка в секундах.

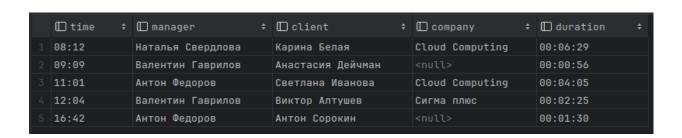
Получите данные о звонках за 5 апреля 2018 года. Выведите 5 колонок:

- 1. **time** время звонка в формате ЧЧ:ММ;
- 2. **manager** имя и фамилия менеджера, который принял звонок;
- 3. **client** имя и фамилия сотрудника контрагента, который позвонил;
- 4. **company** название компании контрагента, NULL если названия нет;
- 5. **duration** продолжительность звонка в формате Ч:ММ:СС.

Данные отсортируйте по дате и времени звонка.

```
SELECT
    DATE_FORMAT(c.date, '%H:%i') AS time,
    CONCAT(m.first_name, ' ', m.last_name) AS manager,
    CONCAT(IFNULL(cl.first_name, ''), ' ', IFNULL(cl.last_name, ''))
AS client,
    co.name AS company,
    SEC_TO_TIME(c.duration_sec) AS duration
FROM calls c

JOIN managers m ON c.manager_id = m.id
LEFT JOIN clients cl ON c.client_id = cl.id
LEFT JOIN companies co ON cl.company_id = co.id
WHERE DATE(c.date) = '2018-04-05'
ORDER BY c.date;
```



- 4) В CRM системе компании есть 4 таблицы:
- 1. **managers** менеджеры (сотрудники) компании;
- 2. **companies** контрагенты компании;
- 3. **clients** сотрудники контрагентов;
- 4. **calls** звонки сотрудников контрагентов менеджерам компании. В таблице также фиксируется дата (date) и продолжительность (duration\_sec) звонка в секундах.

Получите список **всех** компаний и общую продолжительность звонков по каждой из них. Выведите 2 колонки:

- 1. **company** название компании;
- 2. **duration** общая продолжительность звонков в формате Ч:ММ:СС.

Данные отсортируйте по продолжительности разговоров.

```
SELECT c.name AS company,

SEC_TO_TIME(SUM(calls.duration_sec)) AS duration

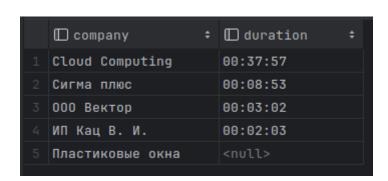
FROM companies c

LEFT JOIN clients cl ON c.id = cl.company_id

LEFT JOIN calls ON cl.id = calls.client_id

GROUP BY c.id

ORDER BY duration DESC;
```



- 5) В CRM системе компании есть 4 таблицы:
- 1. **managers** менеджеры (сотрудники) компании;
- 2. **companies** контрагенты компании;
- 3. **clients** сотрудники контрагентов;
- 4. **calls** звонки сотрудников контрагентов менеджерам компании.

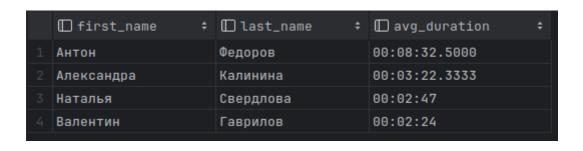
В таблице также фиксируется дата (date) и продолжительность (duration\_sec) звонка в секундах.

Получите среднее время общения менеджеров с сотрудниками Cloud Computing. Выведите 3 колонки:

- 1. **first\_name** имя менеджера;
- 2. **last\_name** фамилия менеджера;
- 3. **avg\_duration** средняя продолжительность звонка в формате ЧЧ:ММ:СС.

Данные отсортируйте по средней продолжительности разговоров в обратном порядке.

```
SELECT
    m.first_name,
    m.last name,
    SEC TO TIME(AVG(c.duration sec)) AS avg duration
FROM
    managers m
JOIN
    calls c ON m.id = c.manager id
JOIN
    clients cl ON c.client id = cl.id
JOIN
    companies co ON cl.company id = co.id
WHERE
    co.name = 'Cloud Computing'
GROUP BY
    m.id
ORDER BY
    avg_duration DESC;
```



6) В таблице **products** находятся товары, а в **categories** — категории, к которым они относятся. Поле **category\_id** является внешним ключом, который указывает на первичный ключ категории. При создании ключа в запросе указали **ON DELETE SET NULL**.

Перенесите товары из категории «Фрукты и овощи» в категории «Фрукты» и «Овощи» а затем удалите «Фрукты и овощи»

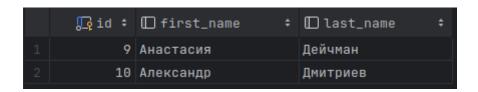
Также удалите категории «Молочные продукты» вместе с товарами.

```
UPDATE products
SET category id = (SELECT id FROM categories WHERE name = 'Φργκτω')
WHERE category id = (SELECT id FROM categories WHERE name = 'Фрукты и
овощи');
INSERT INTO products (name, count, price, category id)
SELECT name, count, price, (SELECT id FROM categories WHERE name =
'Овощи') AS category id
FROM products
WHERE category id = (SELECT id FROM categories WHERE name = 'Фрукты и
овощи');
DELETE FROM products
WHERE category id = (SELECT id FROM categories WHERE name = 'Фрукты и
овощи');
DELETE FROM categories WHERE name = 'Фрукты и овощи';
DELETE FROM products
WHERE category_id = (SELECT id FROM categories WHERE name = 'Молочные
продукты');
DELETE FROM categories WHERE name = 'Молочные продукты';
```

7) Таблицы **users** и **roles** связаны отношением многие ко многим через таблицу **users\_roles**.

Получите список всех сотрудников, которые не выполняют ни одной роли. Выведите id, имена и фамилии таких сотрудников. Данные отсортируйте по id.

```
SELECT u.id, u.first_name, u.last_name
FROM users u
LEFT JOIN users_roles ur ON u.id = ur.user_id
WHERE ur.user_id IS NULL
ORDER BY u.id;
```



8) Таблицы users и roles связаны отношением многие ко многим через таблицу users roles.

В компании произошли небольшие изменения и вам необходимо внести соответствующие правки в таблицы:

- 1. Уберите Светлану Иванову из менеджеров.
- 2. Сделайте менеджером Анастасию Дейчман.
- 3. Назначьте Александра Дмитриева программистом.

9) Таблицы **users**, **roles** и **projects** связаны отношением многие ко многим через таблицу **users\_rp**. Такая «тройная» связь позволяет задавать роли пользователей в конкретных проектах.

Получите состав команды с указанием ролей для проекта «Сайт оконный» Выведите id, имена и фамилии сотрудников, а также их роли в поле **role**. Данные отсортируйте по фамилии.

```
SELECT
    u.id,
    u.first_name,
    u.last_name,
    r.name AS role
FROM
    users u
JOIN
    users_rp urp ON u.id = urp.user_id
JOIN
    roles r ON urp.role_id = r.id
WHERE
    urp.project_id = (SELECT id FROM projects WHERE name = 'Сайт оконный')
ORDER BY
    u.last_name;

⇒ □ last_name

    □ role

     ☐ id ÷ ☐ first_name
           3 Елена
                                   Абрамова
                                                      Дизайнер
```

Иванова

Маф

Менеджер

Программист

2 Светлана

8 Игорь

- 10) Рассмотрим базу данных интернет-магазина:
- 1. Таблица **orders** содержит список заказов пользователей и связана с **users** отношением один ко многим через поле **user\_id**.
- 2. Также **orders** связана отношением многие ко многим с таблицей **products** через таблицу **orders\_details**, что дает возможность указывать какие именно товары находятся в заказе. Одна запись в **orders\_details** соответствует одному заказанному товару.
- 3. Поле **count** в таблице **products** отвечает за количество товаров на складе.

Получите выполненные заказы с указанием количества и стоимости товаров в них. Выведите три поля: id заказа, количество товаров в заказе (в поле **products**) и сумму заказа (в поле **amount**). Данные отсортируйте по сумме заказа.

```
SELECT
   o.id AS order_id,
   COUNT(od.product id) AS products count,
   SUM(p.price) AS amount
FROM
   orders o
JOIN
   orders details od ON o.id = od.order id
JOIN
   products p ON od.product id = p.id
WHERE
   o.status = 'success'
GROUP BY
   o.id
ORDER BY
    amount;
```

	□ order_id ÷	products_count ÷	□ amount ÷
1	8	1	2500
2	7	1	3000
3	1	3	19500
4	3	1	19900
5	10	1	31740
6	9	3	32800
7	6	2	35600
8	5	3	36540
9	11	3	39240