

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	album	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 жанры и количество композиций в каждом из них.
Жанры выведите в поле **name**, а количество композиций в **songs**.
Данные отсортируйте 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;
```

	genre	song_count
1	Metal	4
2	Pop	6
3	Rap	6
4	Rock	11

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;
```

	time	manager	client	company	duration
1	08:12	Наталья Свердлова	Карина Белая	Cloud Computing	00:06:29
2	09:09	Валентин Гаврилов	Анастасия Дейчман	<null>	00:00:56
3	11:01	Антон Федоров	Светлана Иванова	Cloud Computing	00:04:05
4	12:04	Валентин Гаврилов	Виктор Алтушев	Сигма плюс	00:02:25
5	16:42	Антон Федоров	Антон Сорокин	<null>	00:01:30

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;
```

	company	duration
1	Cloud Computing	00:37:57
2	Сигма плюс	00:08:53
3	000 Вектор	00:03:02
4	ИП Кац В. И.	00:02:03
5	Пластиковые окна	<null>

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;
```

	first_name	last_name	avg_duration
1	Антон	Федоров	00:08:32.5000
2	Александра	Калинина	00:03:22.3333
3	Наталья	Свердлова	00:02:47
4	Валентин	Гаврилов	00:02:24

- 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 = 'Фрукты')
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;
```

	id	first_name	last_name
1	9	Анастасия	Дейчман
2	10	Александр	Дмитриев

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.

```
DELETE FROM users_roles
WHERE user_id = (SELECT id FROM users WHERE first_name = 'Светлана' AND
last_name = 'Иванова')
AND role_id = (SELECT id FROM roles WHERE name = 'Менеджер');

INSERT INTO users_roles (user_id, role_id)
VALUES
((SELECT id FROM users WHERE first_name = 'Анастасия' AND last_name =
'Дейчман'),
(SELECT id FROM roles WHERE name = 'Менеджер'));

UPDATE users_roles
SET role_id = (SELECT id FROM roles WHERE name = 'Программист')
WHERE user_id = (SELECT id FROM users WHERE first_name = 'Александр' AND
last_name = 'Дмитриев')
AND role_id = (SELECT id FROM roles WHERE name = 'Менеджер');
```


- 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;
```

	id	first_name	last_name	role
1	3	Елена	Абрамова	Дизайнер
2	2	Светлана	Иванова	Менеджер
3	8	Игорь	Маф	Программист

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