

\*Все задания выполнены на MySQL\*

1. Получите список городов-миллионников в Европе.

Выведите название города, название страны и население города. Название страны выведите в поле **country**, города в поле **name**, а население в **population**. Данные отсортируйте по населению в обратном порядке (крупные города выше).

```
DELIMITER //
```

```
CREATE PROCEDURE GetEuropeanMillionCities()  
BEGIN  
    SELECT cities.name AS name, countries.name AS country,  
    cities.population AS population  
    FROM cities  
    JOIN countries ON cities.country = countries.id  
    WHERE countries.pw LIKE '%Europe%'  
    AND cities.population >= 1000000  
    ORDER BY cities.population DESC;  
END //
```

```
DELIMITER ;
```

```
CALL GetEuropeanMillionCities();
```

	name	country	population
1	Москва	Россия	12506468
2	Санкт-Петербург	Россия	5351935
3	Берлин	Германия	3670999
4	Киев	Украина	2925760
5	Минск	Беларусь	1974800
6	Гамбург	Германия	1803752
7	Прага	Чехия	1267449

2. На сайте интернет-магазина нужно выводить только те категории, в которых есть товары. Получите имена категорий, в которых есть товары, а также количество товаров в каждой категории. Данные выведите в двух колонках:
- 1). **category** с именем категории;
  - 2). **products** с количеством товаров.
- Отсортируйте итоговую таблицу по имени категории.

```
DELIMITER //
```

```
CREATE PROCEDURE GetCategoriesWithProducts(IN sortOrder VARCHAR(50))
BEGIN
    CREATE TEMPORARY TABLE temp_result AS
    SELECT
        c.name AS category,
        COUNT(p.id) AS products
    FROM
        categories c
    JOIN
        products p ON c.id = p.category
    GROUP BY
        c.id
    ORDER BY
        sortOrder;

    SELECT * FROM temp_result;

    DROP TEMPORARY TABLE IF EXISTS temp_result;
END //
```

```
DELIMITER ;
```

```
CALL GetCategoriesWithProducts('category'); # 1
CALL GetCategoriesWithProducts('products'); # 2
```

	category	products
1	Джинсы	7
2	Куртки	1
3	Платья	1
4	Жилетки	2
5	Юбки	2
6	Брюки	1
7	Водолазки	1
8	Косухи	1
9	Джеггинсы	1
10	Плащи	1
11	Леггинсы	1

3. Добавьте в таблицы новый жанр «Rap» и исполнителя «Eminem» в этом жанре.

```
DELIMITER //
```

```
CREATE PROCEDURE AddRapGenreAndEminem()
```

```
BEGIN
```

```
    DECLARE rapGenreId INT;
```

```
    INSERT INTO genres (name) VALUES ('Rap');
```

```
    SET rapGenreId = LAST_INSERT_ID();
```

```
    INSERT INTO artists (name, genre_id, is_group) VALUES ('Eminem',  
rapGenreId, FALSE);
```

```
END //
```

```
DELIMITER ;
```

```
CALL AddRapGenreAndEminem();
```

4. В таблице artists представлены исполнители, а в таблице genres — жанры в которых они играют. Получите названия всех жанров и количество исполнителей в каждом жанре. Жанры выведите в поле genres, а количество исполнителей в artists. Отсортируйте данные по названию жанра.

```
DELIMITER //
```

```
CREATE PROCEDURE city()
```

```
BEGIN
```

```
    select g.name, count(a.name) from genres as g, artists as a
```

```
    where a.genre_id = g.id
```

```
    group by g.name;
```

```
END; //
```

```
DELIMITER ;
```

  

```
CALL city();
```

	name	count(a.name)
1	Electronic	3
2	Rap	5
3	Metal	3
4	Rock	3
5	Pop	1

5. 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. Получите в одном запросе имя товара, цену товара и имя категории, к которой принадлежит товар. Отсортируйте данные в алфавитном порядке по имени товара. Имя категории выведите в колонке с именем category. Если у товара нет категории, то выведите NULL.

```
DELIMITER //
```

```
CREATE PROCEDURE prod_cost()
```

```
BEGIN
```

```
    SELECT products.name, categories.name AS category,  
products.price FROM products, categories  
    WHERE categories.id = products.category  
    ORDER BY products.name;
```

```
END; //
```

```
DELIMITER ;
```

```
CALL prod_cost();
```

	name	category	price
1	Брюки OLBE	Брюки	2898
2	Водолазка OPIMU	Водолазки	1527
3	Джеггинсы Sasha Rozhdestvenskaya	Джеггинсы	1691
4	Джинсы Ima jeans	Джинсы	1842
5	Джинсы Mango Jane2	Джинсы	200
6	Джинсы Mango Kim2	Джинсы	2499
7	Джинсы Mango Lonny 2	Джинсы	2990
8	Джинсы Mango Noa2	Джинсы	2499
9	Джинсы Mango Soho2	Джинсы	1999
10	Джинсы Uptown2 (Mango)	Джинсы	3499
11	Жилет FinFlare	Жилетки	6999
12	Жилет Zarina	Жилетки	2519
13	Косуха Mango	Косухи	9990
14	Куртка Magno Nico 2	Куртки	4999
15	Леггинсы Oodji	Леггинсы	349
16	Платье Love Republic	Платья	3999
17	Плащ Mango SHERLOCK	Плащи	6499
18	Юбка Katomi	Юбки	3199
19	Юбка Mango Park	Юбки	3999

6. В таблице roles хранится список ролей (должностей) в компании, а в employees — сотрудники с указанием их роли через поле role\_id. Получите список всех должностей и количество активных сотрудников для каждой должности. Выведите 2 столбца: название должности (name) и количество сотрудников (employees). Данные сперва отсортируйте по числу сотрудников в обратном порядке, а затем по имени роли.

```
DELIMITER //
```

```
CREATE PROCEDURE empl_count()
```

```
BEGIN
```

```
    SELECT roles.name, COUNT(employees.id) AS count
```

```
    FROM roles LEFT JOIN employees ON employees.role_id = roles.id
```

```
    WHERE roles.name IS NOT NULL
```

```
    GROUP BY roles.name
```

```
    ORDER BY count DESC, roles.name;
```

```
END; //
```

```
DELIMITER ;
```

  

```
CALL empl_count();
```

	name	count
1	Дизайнер	3
2	Программист	3
3	Менеджер	2
4	Маркетолог	0

7. Получите все страны и города из таблиц **countries** и **cities**.  
Свяжите города со странами, в которых они находятся.  
Выведите 2 столбца:
1. **country** — название страны;
  2. **city** — название города.

Данные отсортируйте сперва по названию страны, а затем по названию города.

CALL city();

	name	name
1	Беларусь	Минск
2	Германия	Берлин
3	Германия	Гамбург
4	Канада	Торонто
5	Китай	Шанхай
6	Россия	Москва
7	Россия	Санкт-Петербург
8	Россия	Саратов
9	США	Нью-Йорк
10	Украина	Киев
11	Украина	Львов
12	Франция	Марсель
13	Чехия	Прага