

Задание 1: Посчитайте, сколько компаний закрылось.

```
SELECT COUNT(id) AS closed_company  
FROM company  
WHERE status = 'closed'
```

Задание 2: Отобразите количество привлечённых средств для новостных компаний США. Используйте данные из таблицы company. Отсортируйте таблицу по убыванию значений в поле funding_total

```
SELECT funding_total  
FROM company  
WHERE country_code = 'USA'  
AND category_code = 'news'  
ORDER BY funding_total DESC
```

Задание 3: Найдите общую сумму сделок по покупке одних компаний другими в долларах. Отберите сделки, которые осуществлялись только за наличные с 2011 по 2013 год включительно.

```
SELECT SUM(price_amount)  
FROM acquisition  
WHERE term_code = 'cash'  
AND EXTRACT(YEAR FROM CAST(acquired_at AS timestamp)) IN (2011, 2012, 2013);
```

Задание 4: Отобразите имя, фамилию и названия аккаунтов людей в твиттере, у которых названия аккаунтов начинаются на 'Silver'.

```
SELECT first_name,  
       last_name,  
       twitter_username  
FROM people  
WHERE twitter_username LIKE 'Silver%';
```

Задание 5: Выведите на экран всю информацию о людях, у которых названия аккаунтов в твиттере содержат подстроку 'money', а фамилия начинается на 'K'.

```
SELECT *  
FROM people  
WHERE twitter_username LIKE '%money%'  
AND last_name LIKE 'K%';
```

Задание 6: Для каждой страны отобразите общую сумму привлечённых инвестиций, которые получили компании, зарегистрированные в этой стране. Страну, в которой зарегистрирована компания, можно определить по коду страны. Отсортируйте данные по убыванию суммы.

```
SELECT country_code AS country,  
       SUM(funding_total) AS total  
FROM company  
GROUP BY country_code  
ORDER BY total DESC;
```

Задание 7: Составьте таблицу, в которую войдёт дата проведения раунда, а также минимальное и максимальное значения суммы инвестиций, привлечённых в эту дату. Оставьте в итоговой таблице только те записи, в которых минимальное значение суммы инвестиций не равно нулю и не равно максимальному значению.

```
SELECT funded_at AS date_round,  
       MIN(raised_amount) AS min_sum,  
       MAX(raised_amount) AS max_sum  
FROM funding_round  
GROUP BY date_round  
HAVING MIN(raised_amount) <> 0  
       AND MIN(raised_amount) <> MAX(raised_amount);
```

Задание 8: Создайте поле с категориями:

- Для фондов, которые инвестируют в 100 и более компаний, назначьте категорию **high_activity**.
 - Для фондов, которые инвестируют в 20 и более компаний до 100, назначьте категорию **middle_activity**.
 - Если количество инвестируемых компаний фонда не достигает 20, назначьте категорию **low_activity**.
- Отобразите все поля таблицы fund и новое поле с категориями.**

```
SELECT *,  
CASE  
  WHEN invested_companies >= 100 THEN 'high_activity'  
  WHEN invested_companies >= 20 AND invested_companies < 100 THEN 'middle_activity'  
  WHEN invested_companies < 20 THEN 'low_activity'  
END AS category  
FROM fund;
```

Задание 9: Для каждой из категорий, назначенных в предыдущем задании, посчитайте округлённое до ближайшего целого числа среднее количество инвестиционных раундов, в которых фонд принимал участие. Выведите на экран категории и среднее число инвестиционных раундов. Отсортируйте таблицу по возрастанию среднего.

```
SELECT ROUND(AVG(investment_rounds)) AS average,  
CASE  
  WHEN invested_companies >= 100 THEN 'high_activity'  
  WHEN invested_companies >= 20 THEN 'middle_activity'  
  ELSE 'low_activity'  
END AS activity  
FROM fund  
GROUP BY activity  
ORDER BY average;
```

Задание 10: Проанализируйте, в каких странах находятся фонды, которые чаще всего инвестируют в стартапы. Для каждой страны посчитайте минимальное, максимальное и среднее число компаний, в которые инвестировали фонды этой страны, основанные с 2010 по 2012 год включительно. Исключите страны с фондами, у которых минимальное число компаний, получивших инвестиции, равно нулю. Выгрузите десять самых активных

стран-инвесторов: отсортируйте таблицу по среднему количеству компаний от большего к меньшему. Затем добавьте сортировку по коду страны в лексикографическом порядке.

```
SELECT country_code,  
       MIN(invested_companies) AS min_company,  
       MAX(invested_companies) AS max_company,  
       AVG(invested_companies) AS avg_company  
FROM fund  
WHERE EXTRACT(YEAR FROM founded_at) IN (2010, 2011, 2012)  
GROUP BY country_code  
HAVING MIN(invested_companies) <> 0  
ORDER BY avg_company DESC, country_code  
LIMIT 10;
```

Задание 11: Отобразите имя и фамилию всех сотрудников стартапов. Добавьте поле с названием учебного заведения, которое окончил сотрудник, если эта информация известна.

```
SELECT p.first_name,  
       p.last_name,  
       e.institution  
FROM people AS p  
LEFT OUTER JOIN education AS e ON p.id=e.person_id;
```

Задание 12: Для каждой компании найдите количество учебных заведений, которые окончили её сотрудники. Выведите название компании и число уникальных названий учебных заведений. Составьте топ-5 компаний по количеству университетов.

```
SELECT c.name AS company_name,  
       COUNT(DISTINCT e.institution) AS total_university  
FROM company AS c  
INNER JOIN people AS p ON c.id=p.company_id  
INNER JOIN education AS e ON p.id=e.person_id  
GROUP BY company_name  
ORDER BY total_university DESC  
LIMIT 5;
```

Задание 13: Составьте список с уникальными названиями закрытых компаний, для которых первый раунд финансирования оказался последним.

```
SELECT DISTINCT c.name  
FROM company AS c  
INNER JOIN funding_round AS fr ON c.id=fr.company_id  
WHERE is_first_round = 1  
      AND is_last_round = 1  
      AND status = 'closed';
```

Задание 14: Составьте список уникальных номеров сотрудников, которые работают в компаниях, отобранных в предыдущем задании.

```
SELECT id  
FROM people  
WHERE company_id IN (SELECT c.id  
                     FROM company AS c
```

```
INNER JOIN funding_round AS fr ON c.id=fr.company_id
WHERE is_first_round = 1 AND is_last_round = 1
AND status = 'closed');
```

Задание 15: Составьте таблицу, куда войдут уникальные пары с номерами сотрудников из предыдущей задачи и учебным заведением, которое окончил сотрудник.

```
SELECT DISTINCT p.id,
    e.institution
FROM people AS p
INNER JOIN education AS e ON p.id=e.person_id
WHERE company_id IN (SELECT c.id
    FROM company AS c
    INNER JOIN funding_round AS fr ON c.id=fr.company_id
    WHERE is_first_round = 1 AND is_last_round = 1
    AND status = 'closed')
```

Задание 16: Посчитайте количество учебных заведений для каждого сотрудника из предыдущего задания. При подсчёте учитывайте, что некоторые сотрудники могли окончить одно и то же заведение дважды.

```
SELECT DISTINCT p.id AS person,
    COUNT(e.institution) AS university
FROM people AS p
INNER JOIN education AS e ON p.id=e.person_id
WHERE p.company_id IN (SELECT c.id
    FROM company AS c
    INNER JOIN funding_round AS fr ON c.id=fr.company_id
    WHERE is_first_round=1 AND is_last_round=1
    AND status='closed')
GROUP BY person;
```

Задание 17: Дополните предыдущий запрос и выведите среднее число учебных заведений (всех, не только уникальных), которые окончили сотрудники разных компаний. Нужно вывести только одну запись, группировка здесь не понадобится.

```
WITH
un AS (SELECT DISTINCT p.id AS person,
    COUNT(e.institution) AS university
FROM people AS p
INNER JOIN education AS e ON p.id=e.person_id
WHERE p.company_id IN (SELECT c.id
    FROM company AS c
    INNER JOIN funding_round AS fr ON c.id=fr.company_id
    WHERE is_first_round=1 AND is_last_round=1
    AND status='closed')
GROUP BY person)

SELECT AVG(un.university) AS average_university
FROM un;
```

Задание 18:

Напишите похожий запрос: выведите среднее число учебных заведений (всех, не только уникальных), которые окончили сотрудники Facebook (сервис, запрещённый на территории РФ)

```
WITH
un AS (SELECT DISTINCT p.id AS person,
      COUNT(e.institution) AS university
      FROM people AS p
      INNER JOIN education AS e ON p.id=e.person_id
      WHERE p.company_id IN (SELECT id
                            FROM company
                            WHERE name='Facebook')
      GROUP BY person)

SELECT AVG(un.university)
FROM un;
```

Задание 19: Составьте таблицу из полей:

- name_of_fund — название фонда;
- name_of_company — название компании;
- amount — сумма инвестиций, которую привлекла компания в раунде.

В таблицу войдут данные о компаниях, в истории которых было больше шести важных этапов, а раунды финансирования проходили с 2012 по 2013 год включительно.

```
SELECT f.name AS name_of_fund,
       c.name AS name_of_company,
       fr.raised_amount AS amount
FROM fund AS f
INNER JOIN investment AS i ON f.id=i.fund_id
INNER JOIN company AS c ON i.company_id=c.id
INNER JOIN funding_round AS fr ON i.funding_round_id=fr.id
WHERE c.milestones > 6
      AND EXTRACT(YEAR FROM fr.funded_at) IN (2012,2013);
```

Задание 20: Выгрузите таблицу, в которой будут такие поля:

- название компании-покупателя;
 - сумма сделки;
 - название компании, которую купили;
 - сумма инвестиций, вложенных в купленную компанию;
 - доля, которая отображает, во сколько раз сумма покупки превысила сумму вложенных в компанию инвестиций, округлённая до ближайшего целого числа.
- Не учитывайте те сделки, в которых сумма покупки равна нулю. Если сумма инвестиций в компанию равна нулю, исключите такую компанию из таблицы.
- Отсортируйте таблицу по сумме сделки от большей к меньшей, а затем по названию купленной компании в лексикографическом порядке. Ограничьте таблицу первыми десятью записями.

```
SELECT a.name AS acquiring_name,
       acq.price_amount AS price_amount,
```

```

        b.name AS acquired_name,
        b.funding_total AS invest,
        ROUND(acq.price_amount / b.funding_total) AS per
FROM acquisition AS acq
INNER JOIN company AS a ON a.id=acq.acquiring_company_id
INNER JOIN company AS b ON b.id=acq.acquired_company_id
WHERE acq.price_amount <> 0
    AND b.funding_total <> 0
ORDER BY price_amount DESC, acquired_name
LIMIT 10;

```

Задание 21: Выгрузите таблицу, в которую войдут названия компаний из категории social, получившие финансирование с 2010 по 2013 год включительно. Проверьте, что сумма инвестиций не равна нулю. Выведите также номер месяца, в котором проходил раунд финансирования.

```

SELECT c.name,
       EXTRACT(MONTH FROM fr.funded_at) AS month
FROM company AS c
INNER JOIN funding_round AS fr ON c.id=fr.company_id
WHERE category_code = 'social'
    AND EXTRACT(YEAR FROM funded_at) IN (2010, 2011, 2012, 2013)
    AND raised_amount <> 0;

```

Задание 22: Отберите данные по месяцам с 2010 по 2013 год, когда проходили инвестиционные раунды. Сгруппируйте данные по номеру месяца и получите таблицу, в которой будут поля:

- номер месяца, в котором проходили раунды;
- количество уникальных названий фондов из США, которые инвестировали в этом месяце;
- количество компаний, купленных за этот месяц;
- общая сумма сделок по покупкам в этом месяце.

```

WITH
c AS (SELECT EXTRACT(MONTH FROM acquired_at) AS month,
            COUNT(acquired_company_id) AS company,
            SUM(price_amount) AS sum_price
FROM acquisition
WHERE EXTRACT(YEAR FROM acquired_at) IN (2010, 2011, 2012, 2013)
GROUP BY month),

m AS (SELECT EXTRACT(MONTH FROM fr.funded_at) AS month,
            COUNT(DISTINCT f.name) AS funds
FROM funding_round AS fr
INNER JOIN investment AS i ON fr.id=i.funding_round_id
INNER JOIN fund AS f ON i.fund_id=f.id
WHERE f.country_code='USA'
    AND EXTRACT(YEAR FROM fr.funded_at) IN (2010, 2011, 2012, 2013)
GROUP BY month)
SELECT m.month,
       m.funds,

```

```
c.company,  
c.sum_price  
FROM m LEFT OUTER JOIN c ON m.month=c.month;
```

Задание 23: Составьте сводную таблицу и выведите среднюю сумму инвестиций для стран, в которых есть стартапы, зарегистрированные в 2011, 2012 и 2013 годах. Данные за каждый год должны быть в отдельном поле. Отсортируйте таблицу по среднему значению инвестиций за 2011 год от большего к меньшему.

```
WITH  
y11 AS (SELECT country_code,  
             AVG(funding_total) AS average_2011  
        FROM company  
        WHERE EXTRACT(YEAR FROM founded_at) = 2011  
        GROUP BY country_code),  
  
y12 AS (SELECT country_code,  
             AVG(funding_total) AS average_2012  
        FROM company  
        WHERE EXTRACT(YEAR FROM founded_at) = 2012  
        GROUP BY country_code),  
y13 AS (SELECT country_code,  
             AVG(funding_total) AS average_2013  
        FROM company  
        WHERE EXTRACT(YEAR FROM founded_at) = 2013  
        GROUP BY country_code)  
SELECT y11.country_code,  
       y11.average_2011,  
       y12.average_2012,  
       y13.average_2013  
FROM y11 INNER JOIN y12 ON y11.country_code=y12.country_code  
INNER JOIN y13 ON y12.country_code=y13.country_code  
ORDER BY average_2011 DESC;
```