

1. Посчитайте, сколько компаний закрылось

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

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 category_code = 'news'
      AND country_code = 'USA'
ORDER BY funding_total DESC
```

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 EXTRACT(YEAR FROM CAST (acquired_at AS date)) IN (2011,2012,2013)
      AND term_code = 'cash'
```

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%'
```

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%'
```

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,
       SUM(funding_total)
FROM company
GROUP BY country_code
ORDER BY SUM(funding_total) DESC
```

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,  
       MIN(raised_amount),  
       MAX(raised_amount)  
FROM funding_round  
GROUP BY funded_at  
HAVING MIN(raised_amount) > 0  
       AND MIN(raised_amount) <> MAX(raised_amount)
```

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'  
    ELSE 'low_activity'  
  END,  
  *  
FROM fund
```

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

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_invested_companies,
       MAX(invested_companies) AS max_invested_companies,
       AVG(invested_companies) AS avg_invested_companies
FROM fund
WHERE EXTRACT(YEAR FROM CAST(founded_at AS date)) BETWEEN '2010' AND '2012'
GROUP BY country_code
HAVING MIN(invested_companies) <> 0
ORDER BY avg_invested_companies DESC,
         country_code
LIMIT 10
```

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 JOIN education AS e ON p.id = e.person_id;
```

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 count_institution
FROM company AS c
INNER JOIN people as p ON c.id=p.company_id
INNER JOIN education AS e ON e.person_id=p.id
GROUP BY company_name
ORDER BY count_institution desc
LIMIT 5
```

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

```
SELECT c.name
FROM company as c
LEFT JOIN funding_round as fr on fr.company_id = c.id
WHERE c.status = 'closed'
      and fr.is_first_round = 1
      and is_last_round = 1
GROUP BY name;
```

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

```
SELECT p.id
FROM people AS p
LEFT JOIN company AS c ON c.id=p.company_id
WHERE c.name IN (
SELECT c.name
FROM company as c
LEFT JOIN funding_round as fr on fr.company_id = c.id
WHERE c.status = 'closed'
      and fr.is_first_round = 1
      and is_last_round = 1
GROUP BY name)
```

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

```
SELECT p.id,
       e.institution
FROM education AS e
RIGHT JOIN people AS p ON p.id=e.person_id
WHERE p.id IN(
SELECT p.id
FROM people AS p
LEFT JOIN company AS c ON c.id=p.company_id
WHERE c.name IN (
SELECT c.name
FROM company as c
LEFT JOIN funding_round as fr on fr.company_id = c.id
WHERE c.status = 'closed'
      and fr.is_first_round = 1
      and is_last_round = 1
GROUP BY name))
group by p.id,e.institution
HAVING e.institution IS NOT NULL;
```

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,
                COUNT(ed.institution)
FROM company AS com
INNER JOIN people AS p ON com.id=p.company_id
LEFT JOIN education AS ed ON p.id=ed.person_id
WHERE status = 'closed'
      AND com.id IN (SELECT company_id
                     FROM funding_round
                     WHERE is_first_round = 1
                        AND is_last_round = 1)
AND ed.institution IS NOT NULL
GROUP BY p.id
```

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 inst_count AS(SELECT p.id,
                        COUNT(e.institution)
                    FROM people AS p
                    LEFT JOIN education AS e ON p.id = e.person_id
                    WHERE p.company_id IN (SELECT c.id
                                           FROM company AS c
                                           JOIN funding_round AS fr ON c.id =
fr.company_id
                                           WHERE STATUS = 'closed'
                                           AND is_first_round = 1
                                           AND is_last_round = 1
                                           GROUP BY c.id)
                    GROUP BY p.id
                    HAVING COUNT(DISTINCT e.institution) > 0)

SELECT AVG(COUNT)
FROM inst_count;
```

18. Напишите похожий запрос: выведите среднее число учебных заведений (всех, не только уникальных), которые окончили сотрудники Facebook*.

```
WITH inst_count AS(SELECT p.id,
                        COUNT(e.institution)
                    FROM people AS p
                    LEFT JOIN education AS e ON p.id = e.person_id
                    WHERE p.company_id IN (SELECT c.id
                                           FROM company AS c
                                           JOIN funding_round AS fr ON c.id =
fr.company_id
                                           WHERE c.name = 'Facebook'
                                           GROUP BY c.id)
                    GROUP BY p.id
                    HAVING COUNT(DISTINCT e.institution) > 0)

SELECT AVG(COUNT)
FROM inst_count;
```

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 fun.name AS name_of_fund,
       com.name AS name_of_company,
       fr.raised_amount AS amount
FROM investment AS inv
LEFT JOIN company AS com ON com.id = inv.company_id
LEFT JOIN fund AS fun ON fun.id = inv.fund_id
INNER JOIN funding_round as fr ON fr.id = inv.Funding_round_id
WHERE com.milestones > 6
      AND EXTRACT(YEAR FROM CAST(fr.funded_at AS date)) BETWEEN 2012 AND 2013
```

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.

```
WITH buyer AS (SELECT com.name AS buyer_name,
                      acq.price_amount AS price_amount,
                      acq.id AS buyer_id
                  FROM acquisition AS acq
                  LEFT JOIN company AS com ON acq.acquiring_company_id = com.id
                  WHERE acq.price_amount > 0),
```

```
purchase AS (SELECT com.name AS purchase_name,
                     com.funding_total AS funding_total,
                     acq.id AS purchase_id
               FROM acquisition AS acq
               LEFT JOIN company AS com ON acq.acquired_company_id = com.id
               WHERE com.funding_total > 0)
```

```
SELECT b.buyer_name,
       b.price_amount,
       p.purchase_name,
       p.funding_total,
       ROUND(b.price_amount / p.funding_total) AS uplift
```

```
FROM buyer AS b
JOIN purchase AS P ON b.buyer_id = p.purchase_id
ORDER BY price_amount DESC, purchase_name
LIMIT 10;
```

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 AS company_name,
       EXTRACT(MONTH FROM CAST(fr.funded_at AS date)) AS month

FROM company AS c
LEFT JOIN funding_round AS fr ON c.id = fr.company_id
WHERE EXTRACT(YEAR FROM CAST(fr.funded_at AS date)) BETWEEN 2010 AND 2013
      AND fr.raised_amount <> 0
      AND c.category_code = 'social'
```

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

```
buyer AS (SELECT EXTRACT(MONTH FROM CAST(funded_at AS date)) AS month_funding,  
            COUNT (DISTINCT fun.name) AS fund_count  
FROM funding_round AS fr  
LEFT JOIN investment as inv ON inv.funding_round_id = fr.id  
LEFT JOIN fund AS fun ON inv.fund_id = fun.id  
WHERE country_code = 'USA'  
            AND EXTRACT(YEAR FROM CAST(funded_at AS date)) BETWEEN 2010 AND 2013  
GROUP BY month_funding),
```

```
purchase AS (SELECT EXTRACT(MONTH FROM CAST(acquired_at AS DATE)) AS month_funding,  
                COUNT(acquired_company_id) AS purchases_count,  
                SUM(price_amount) AS purchases_sum  
FROM acquisition  
WHERE EXTRACT(YEAR FROM CAST(acquired_at AS DATE)) BETWEEN 2010 AND 2013  
GROUP BY month_funding)
```

```
SELECT b.month_funding,  
       b.fund_count,  
       p.purchases_count,  
       p.purchases_sum  
FROM buyer AS b  
LEFT JOIN purchase AS P ON b.month_funding=p.month_funding
```

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 AS country,  
                AVG(funding_total) AS avg_2011  
        FROM company  
        WHERE EXTRACT(YEAR FROM CAST(founded_at AS date)) = 2011  
        GROUP BY country),  
y12 AS (SELECT country_code AS country,  
                AVG(funding_total) AS avg_2012  
        FROM company  
        WHERE EXTRACT(YEAR FROM CAST(founded_at AS date)) = 2012  
        GROUP BY country),  
y13 AS (SELECT country_code AS country,  
                AVG(funding_total) AS avg_2013  
        FROM company  
        WHERE EXTRACT(YEAR FROM CAST(founded_at AS date)) = 2013  
        GROUP BY country)
```

```
SELECT y11.country,  
       y11.avg_2011,
```

```
        y12.avg_2012,  
        y13.avg_2013  
FROM y11  
INNER JOIN y12 ON y12.country = y11.country  
INNER JOIN y13 ON y13.country = y11.country  
  
ORDER BY y11.avg_2011 DESC
```