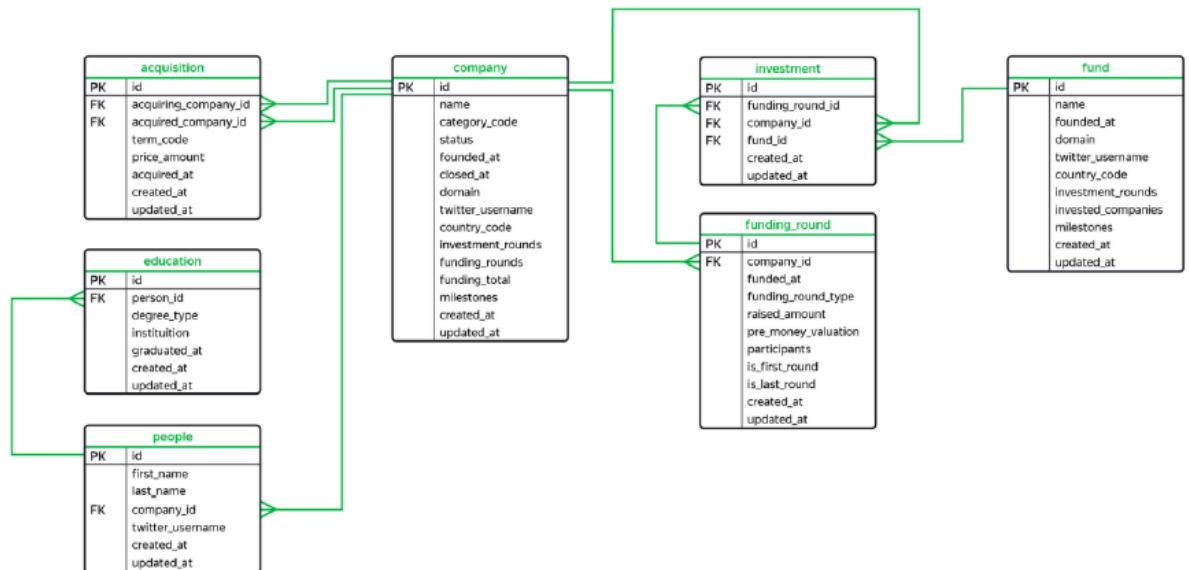


База данных хранит информацию о венчурных фондах и инвестициях в компании-стартапы. Основана на датасете Startup Investments, опубликованном на платформе Kaggle.

## Схема базы данных



## Задачи и запросы к БД

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 term_code = 'cash'
      AND acquired_at BETWEEN '2011-01-01' AND '2013-12-31';
```

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) AS fund_tot
FROM company
GROUP BY country_code
ORDER BY fund_tot 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::date,
       MIN(raised_amount),
       MAX(raised_amount)
FROM funding_round
GROUP BY funded_at::date
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 BETWEEN 20 AND 99 THEN 'middle_activity'
```

```

        WHEN invested_companies < 20 THEN '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.

```

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

```

10. 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 ;

```

11. 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.

```

WITH tab AS
    (SELECT p.id,
        p.company_id AS comp_id, --count(e.institution)--,
        e.institution
    FROM people AS p
    LEFT JOIN education AS e ON e.person_id=p.id)
SELECT c.name,
    count(DISTINCT institution)
FROM company AS c
LEFT JOIN tab ON c.id=tab.comp_id
GROUP BY c.id
ORDER BY count(DISTINCT institution) DESC
LIMIT 5 ;

```

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

```
SELECT name
FROM company
WHERE id in
  (SELECT fr.company_id AS c_id
   FROM funding_round AS fr
   WHERE fr.is_first_round=1
        AND fr.is_last_round=1 )
AND status = 'closed' ;
```

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

```
WITH comp_name AS
(SELECT id
 FROM company
 WHERE id in
  (SELECT fr.company_id AS c_id
   FROM funding_round AS fr
   WHERE fr.is_first_round=1
        AND fr.is_last_round=1 )
 AND status = 'closed'
 ORDER BY id)
SELECT p.id
FROM comp_name
JOIN people AS p ON p.company_id=comp_name.id ;
```

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

```
WITH comp_name AS
(SELECT id
 FROM company
 WHERE id in
  (SELECT fr.company_id AS c_id
   FROM funding_round AS fr
   WHERE fr.is_first_round=1
        AND fr.is_last_round=1 )
 AND status = 'closed'
 ORDER BY id)
SELECT DISTINCT p.id,
                e.institution
FROM people AS p
JOIN comp_name ON p.company_id=comp_name.id
JOIN education AS e ON p.id=e.person_id ;
```

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

```

WITH comp_name AS
(SELECT id
 FROM company
 WHERE id in
   (SELECT fr.company_id AS c_id
    FROM funding_round AS fr
    WHERE fr.is_first_round=1
      AND fr.is_last_round=1 )
  AND status = 'closed'
 ORDER BY id)
SELECT DISTINCT p.id,
               count(e.institution)
FROM people AS p
JOIN comp_name ON p.company_id=comp_name.id
JOIN education AS e ON p.id=e.person_id
GROUP BY p.id ;

```

16. 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 comp_name AS
(SELECT id
 FROM company
 WHERE id in
   (SELECT fr.company_id AS c_id
    FROM funding_round AS fr
    WHERE fr.is_first_round=1
      AND fr.is_last_round=1 )
  AND status = 'closed'
 ORDER BY id),
  comp_num_inst AS
(SELECT DISTINCT p.id,
               count(e.institution) AS inst_num
 FROM people AS p
 JOIN comp_name ON p.company_id=comp_name.id
 JOIN education AS e ON p.id=e.person_id
 GROUP BY p.id)
SELECT avg(inst_num)
FROM comp_num_inst ;

```

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

```

WITH comp_num_inst AS
(SELECT DISTINCT p.id,
               count(e.institution) AS inst_num
 FROM people AS p
 JOIN company AS c ON p.company_id=c.id
 JOIN education AS e ON p.id=e.person_id

```

```

WHERE c.name = 'Facebook'
GROUP BY p.id)
SELECT avg(inst_num)
FROM comp_num_inst ;

```

18. 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
from investment as i
join company as c on i.company_id=c.id
join fund as f on i.fund_id=f.id
join funding_round as fr on fr.id=i.funding_round_id
where fr.funded_at between '2012-01-01' and '2013-12-31'
      and c.milestones > 6
;

```

19. 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 c_acquiring.name AS acquiring,
       a.price_amount AS price,
       c_acquired.name AS acquired,
       c_acquired.funding_total AS sum_investments,
       round(a.price_amount / c_acquired.funding_total) AS rise
FROM acquisition AS a
JOIN company AS c_acquired ON a.acquired_company_id=c_acquired.id
JOIN company AS c_acquiring ON a.acquiring_company_id=c_acquiring.id
WHERE c_acquired.funding_total > 0
      AND a.price_amount > 0
ORDER BY a.price_amount DESC,
         c_acquired.name
LIMIT 10 ;

```

20. 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::date)  
FROM company AS c  
JOIN funding_round AS fr ON c.id=fr.company_id  
WHERE fr.funded_at BETWEEN '2010-01-01' AND '2013-12-31'  
      AND c.category_code = 'social'  
      AND fr.raised_amount > 0 ;
```

21. 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 acquis AS  
(SELECT extract(MONTH  
              FROM a.acquired_at) AS MONTH,  
       count(a.acquired_company_id) AS count_comp,  
       sum(a.price_amount) AS SUM  
FROM acquisition AS a  
WHERE a.acquired_at BETWEEN '2010-01-01' AND '2013-12-31'  
GROUP BY MONTH),  
fund AS  
(SELECT extract(MONTH  
              FROM fr.funded_at) AS MONTH,  
       count(DISTINCT f.name) AS count_fund  
FROM funding_round AS fr  
LEFT JOIN investment AS i ON fr.id=i.funding_round_id  
LEFT JOIN fund AS f ON i.fund_id=f.id  
WHERE fr.funded_at BETWEEN '2010-01-01' AND '2013-12-31'  
      AND f.country_code='USA'  
GROUP BY MONTH)  
SELECT fund.month,  
       fund.count_fund,  
       acquis.count_comp,  
       acquis.sum  
FROM fund  
LEFT JOIN acquis ON fund.month=acquis.month ;
```

22. 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 i1 AS
  (SELECT country_code,
    avg(funding_total) AS a2011
  FROM company
  WHERE founded_at BETWEEN '2011-01-01' AND '2011-12-31'
  GROUP BY country_code),
  i2 AS
  (SELECT country_code,
    avg(funding_total) AS a2012
  FROM company
  WHERE founded_at BETWEEN '2012-01-01' AND '2012-12-31'
  GROUP BY country_code),
  i3 AS
  (SELECT country_code,
    avg(funding_total) AS a2013
  FROM company
  WHERE founded_at BETWEEN '2013-01-01' AND '2013-12-31'
  GROUP BY country_code)
SELECT i1.country_code,
  i1.a2011,
  i2.a2012,
  i3.a2013
FROM i1
INNER JOIN i2 ON i1.country_code = i2.country_code
INNER JOIN i3 ON i2.country_code = i3.country_code
ORDER BY i1.a2011 DESC ;
```