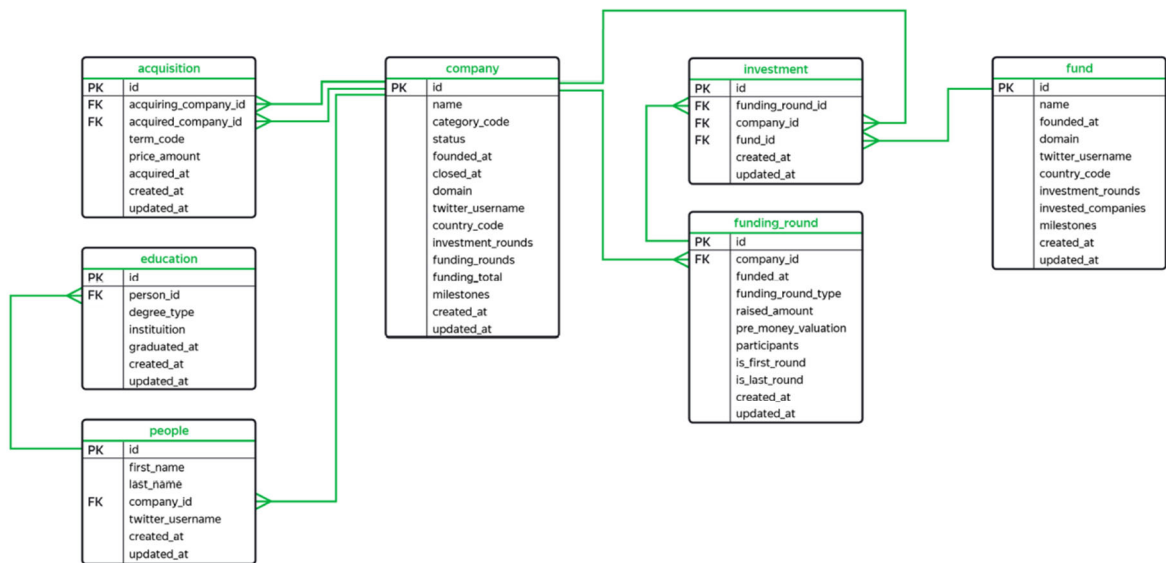


## Реляционная база данных о покупках одних компаний другими.



1. Сколько компаний закрылось.

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

2. Количество привлечённых средств для новостных компаний США (данные из таблицы company, сортировка по убыванию значений в поле funding\_total).

```
SELECT SUM(funding_total)
FROM company
WHERE country_code = 'USA'
AND category_code = 'news'
GROUP BY name
ORDER BY SUM (funding_total) DESC;
```

3. Общая сумма сделок по покупке одних компаний другими в долларах. Отобрать 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 date)) IN (2011, 2012, 2013);
```

4. Имя, фамилия и название 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. Информация 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 отобразить общую сумму привлечённых инвестиций, которые получили компании, зарегистрированные в этой стране.

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

7. Составить таблицу, в которую войдёт дата проведения раунда, а также минимальное и максимальное значения суммы инвестиций, привлечённых в эту дату.

Оставить 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. Создать поле с категориями:

Для фондов, которые инвестируют в 100 и более компаний, назначить категорию high\_activity.

Для фондов, которые инвестируют в 20 и более компаний до 100, назначить категорию middle\_activity.

Если количество инвестируемых компаний фонда не достигает 20, назначить категорию low\_activity.

Отобразить 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  
FROM fund;
```

9. Для каждой из вышеуказанных категорий посчитать округлённое до ближайшего целого числа среднее количество инвестиционных раундов, в которых фонд принимал участие. Вывести на экран категории и среднее число инвестиционных раундов. Отсортировать 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 rounds
FROM fund
GROUP BY activity
ORDER BY rounds;
```

10. В каких странах находятся фонды, которые чаще всего инвестируют в стартапы. Для каждой страны посчитать минимальное, максимальное и среднее число компаний, в которые инвестировали фонды этой страны, основанные с 2010 по 2012 год включительно. Исключить страны с фондами, у которых минимальное число компаний, получивших инвестиции, равно нулю.

Выгрузить десять самых активных стран-инвесторов: отсортировать таблицу по среднему количеству компаний от большего к меньшему. Добавить сортировку по коду страны в лексикографическом порядке.

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

11. Отобразить имя и фамилию всех сотрудников стартапов. Добавить 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;
```

12. Для каждой компании найти количество учебных заведений, которые окончили её сотрудники. Вывести название компании и число уникальных названий учебных заведений. Составить топ-5 компаний по количеству университетов.

```
SELECT c.name,
```

```

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

```

13. Составить 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 f ON c.id=f.company_id
WHERE c.status='closed'
AND f.is_first_round = 1
AND f.is_last_round = 1;

```

14. Составить список уникальных номеров сотрудников, которые работают в компаниях, отобранных выше.

```

SELECT id
FROM people
WHERE company_id IN (SELECT id
FROM company
WHERE status='closed'
AND id IN (SELECT company_id
FROM funding_round
WHERE is_first_round = 1
AND is_last_round = 1));

```

15. Составить таблицу, куда войдут уникальные пары с номерами сотрудников выше и учебным заведением, которое окончил сотрудник.

```

SELECT DISTINCT (person_id),
institution
FROM education
WHERE person_id IN (SELECT id
FROM people AS p
WHERE company_id IN (SELECT id
FROM company
WHERE status='closed'
AND id IN (SELECT company_id
FROM funding_round
WHERE is_first_round = 1
AND is_last_round = 1))));

```

16. Посчитать количество учебных заведений для каждого сотрудника выше. При подсчёте учитывать, что некоторые сотрудники могли окончить одно и то же заведение дважды.

```

SELECT person_id,
       COUNT(institution)
FROM education
WHERE person_id IN (SELECT id
                    FROM people AS p
                    WHERE company_id IN (SELECT id
                                        FROM company
                                        WHERE status='closed'
                                        AND id IN (SELECT company_id
                                                FROM funding_round
                                                WHERE is_first_round = 1
                                                AND is_last_round = 1)))
GROUP BY person_id;

```

17. Вывести среднее число учебных заведений (всех, не только уникальных), которые окончили сотрудники разных компаний выше.

```

SELECT AVG (p_i.count)
FROM
(SELECT person_id,
       COUNT(institution)
FROM education
WHERE person_id IN (SELECT id
                    FROM people AS p
                    WHERE company_id IN (SELECT id
                                        FROM company
                                        WHERE status='closed'
                                        AND id IN (SELECT company_id
                                                FROM funding_round
                                                WHERE is_first_round = 1
                                                AND is_last_round = 1)))
GROUP BY person_id) AS p_i;

```

18. Вывести среднее число учебных заведений (всех, не только уникальных), которые окончили сотрудники Facebook.

```

SELECT AVG (p_i.count)
FROM
(SELECT person_id,
       COUNT(institution)
FROM education
WHERE person_id IN (SELECT id
                    FROM people AS p
                    WHERE company_id IN (SELECT id
                                        FROM company
                                        WHERE name='Facebook')))
GROUP BY person_id) AS p_i;

```

19. Составить 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 company AS c  
INNER JOIN funding_round AS fr ON c.id=fr.company_id  
INNER JOIN investment AS i ON fr.id=i.funding_round_id  
INNER JOIN fund AS f ON i.fund_id=f.id  
WHERE c.milestones > 6  
AND EXTRACT(YEAR FROM CAST(fr.funded_at AS date)) IN (2012, 2013);
```

20. Выгрузить 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.

Если сумма инвестиций в компанию равна нулю, исключить такую компанию из таблицы.

Отсортировать таблицу по сумме сделки от большей к меньшей, а затем по названию купленной компании в лексикографическом порядке. Ограничить таблицу первыми десятью записями.

```
SELECT c.name AS buyer,  
       a.price_amount AS price,  
       co.name AS seller,  
       co.funding_total AS invest,  
       ROUND(a.price_amount/co.funding_total)  
FROM acquisition AS a  
LEFT OUTER JOIN company AS c ON a.acquiring_company_id=c.id  
JOIN company AS co ON a.acquired_company_id=co.id  
WHERE co.funding_total > 0  
AND a.price_amount > 0  
ORDER BY a.price_amount DESC, co.name  
LIMIT 10;
```

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

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

22. Отобрать данные по месяцам с 2010 по 2013 год, когда проходили инвестиционные раунды. Сгруппировать данные по номеру месяца и получить 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
fu AS (SELECT EXTRACT(MONTH FROM CAST(fr.funded_at AS date)) AS months1,
      COUNT(DISTINCT(i.fund_id)) AS funds
      FROM funding_round AS fr
      LEFT OUTER JOIN investment AS i ON fr.id=i.funding_round_id
      LEFT OUTER JOIN fund AS f On i.fund_id=f.id
      WHERE EXTRACT(YEAR FROM CAST(fr.funded_at AS date)) BETWEEN 2010 AND 2013
      AND f.country_code = 'USA'
      GROUP BY months1),
ac AS (SELECT EXTRACT(MONTH FROM CAST(acquired_at AS date)) AS months2,
      COUNT (acquired_company_id) AS acquired,
      SUM (price_amount) AS price
      FROM acquisition
      WHERE EXTRACT(YEAR FROM CAST(acquired_at AS date)) BETWEEN 2010 AND 2013
      GROUP BY months2)
SELECT fu.months1,
      fu.funds,
      ac.acquired,
      ac.price
FROM fu
LEFT OUTER JOIN ac ON fu.months1=ac.months2
ORDER BY fu.months1;
```

23. Составить сводную таблицу и вывести 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. Отсортировать таблицу по среднему значению инвестиций за 2011 год от большего к меньшему.

```
WITH
y11 AS (SELECT country_code,
      AVG(funding_total) AS total11
      FROM company
      WHERE EXTRACT(YEAR FROM CAST(founded_at AS date)) = 2011
      GROUP BY country_code),
y12 AS (SELECT country_code,
      AVG(funding_total) AS total12
      FROM company
      WHERE EXTRACT(YEAR FROM CAST(founded_at AS date)) = 2012
      GROUP BY country_code),
y13 AS (SELECT country_code,
      AVG(funding_total) AS total13
      FROM company
```

```
WHERE EXTRACT(YEAR FROM CAST(founded_at AS date)) = 2013
GROUP BY country_code)
SELECT y11.country_code,
       y11.total11,
       y12.total12,
       y13.total13
FROM y11
INNER JOIN y12 ON y11.country_code=y12.country_code
INNER JOIN y13 ON y12.country_code=y13.country_code
ORDER BY y11.total11 DESC;
```