

# Инвестиции в компании-стартапы

В нашем распоряжении база данных, которая хранит информацию о венчурных фондах и инвестициях в компании-стартапы. Эта база данных основана на датасете Startup Investments, опубликованном на популярной платформе для соревнований по исследованию данных Kaggle.

**Цель проекта:** проанализировать данные о фондах и инвестициях и с помощью различных SQL-запросов к базе данных. Продемонстрировать изученные операторы, функции и методы работы с базой данных.

## Описание проекта:

Познакомимся с важными понятиями, которые встретятся в работе с базой данных:

*Венчурные фонды* — это финансовые организации, которые могут позволить себе высокий риск и инвестировать в компании с инновационной бизнес-идеей или разработанной новой технологией, то есть в *стартапы*. Цель венчурных фондов — в будущем получить значительную прибыль, которая в разы превысит размер их трат на инвестиции в компанию. Если стартап подорожает, венчурный фонд может получить долю в компании или фиксированный процент от её выручки.

Чтобы процесс финансирования стал менее рискованным, его делят на стадии — *раунды*. Тот или иной раунд зависит от того, какого уровня развития достигла компания.

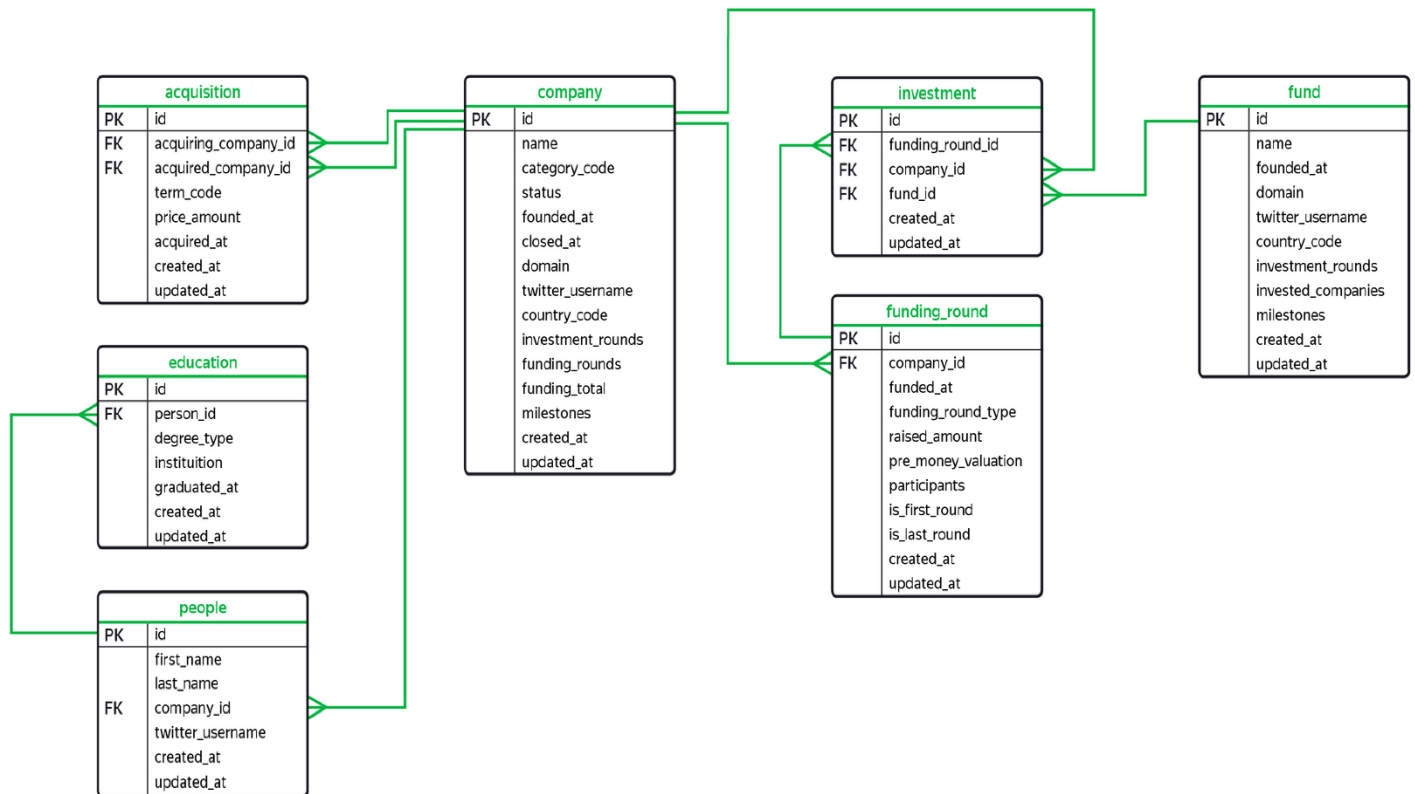
Первые этапы — *предпосевной* и *посевной* раунды. Предпосевной раунд предполагает, что компания как таковая ещё не создана и находится в стадии замысла. Следующий — посевной — раунд знаменует рост проекта: создатели компании разрабатывают бизнес-модель и привлекают инвесторов.

Если компании требуется ментор или наставник — она привлекает *бизнес-ангела*. Бизнес-ангелы — инвесторы, которые помимо финансовой поддержки предлагают экспертную помощь. Такой раунд называют *ангельским*.

Когда стартап становится компанией с проверенной бизнес-моделью и начинает зарабатывать самостоятельно, предложений инвесторов становится больше. Это раунд А, а за ним следуют и другие: В, С, D — на этих этапах компания активно развивается и готовится к IPO.

Иногда выделяют *венчурный раунд* — финансирование, которое могло поступить от венчурного фонда на любом этапе: начальном или более позднем.

## ER-диаграмма базы данных



Ознакомимся с данными, которые хранят таблицы:

### Acquisition

Содержит информацию о покупках одних компаний другими:

- первичный ключ `id` — идентификатор или уникальный номер покупки;
- внешний ключ `acquiring_company_id` — ссылается на таблицу `company` — идентификатор компании-покупателя, то есть той, что покупает другую компанию;
- внешний ключ `acquired_company_id` — ссылается на таблицу `company` — идентификатор компании, которую покупают;
- `term_code` — способ оплаты сделки:
  - `cash` — наличными;
  - `stock` — акциями компании;
  - `cash_and_stock` — смешанный тип оплаты: наличные и акции.
- `price_amount` — сумма покупки в долларах;
- `acquired_at` — дата совершения сделки;
- `created_at` — дата и время создания записи в таблице;
- `updated_at` — дата и время обновления записи в таблице.

## Company

Содержит информацию о компаниях-стартапах:

- первичный ключ `id` — идентификатор, или уникальный номер компании;
- `name` — название компании;
- `category_code` — категория деятельности компании, например:
  - `news` — специализируется на работе с новостями;
  - `social` — специализируется на социальной работе.
- `status` — статус компании:
  - `acquired` — приобретена;
  - `operating` — действует;
  - `ipo` — вышла на IPO;
  - `closed` — перестала существовать.
- `founded_at` — дата основания компании;
- `closed_at` — дата закрытия компании, которую указывают в том случае, если компании больше не существует;
- `domain` — домен сайта компании;
- `twitter_username` — название профиля компании в твиттере;
- `country_code` — код страны, например, `usa` для США, `gbw` для Великобритании;
- `investment_rounds` — число раундов, в которых компания участвовала как инвестор;
- `funding_rounds` — число раундов, в которых компания привлекала инвестиции;
- `funding_total` — сумма привлечённых инвестиций в долларах;
- `milestones` — количество важных этапов в истории компании;
- `created_at` — дата и время создания записи в таблице;
- `updated_at` — дата и время обновления записи в таблице.

## Education

Хранит информацию об уровне образования сотрудников компаний:

- первичный ключ `id` — уникальный номер записи с информацией об образовании;
- внешний ключ `person_id` — ссылается на таблицу `people` — идентификатор человека, информация о котором представлена в записи;
- `degree_type` — учебная степень, например:
  - `ba` — Bachelor of Arts — бакалавр гуманитарных наук;
  - `ms` — Master of Science — магистр естественных наук.
- `institution` — учебное заведение, название университета;
- `graduated_at` — дата завершения обучения, выпуска;
- `created_at` — дата и время создания записи в таблице;
- `updated_at` — дата и время обновления записи в таблице.

## Fund

Хранит информацию о венчурных фондах:

- первичный ключ `id` — уникальный номер венчурного фонда;
- `name` — название венчурного фонда;
- `founded_at` — дата основания фонда;
- `domain` — домен сайта фонда;
- `twitter_username` — профиль фонда в твиттере;
- `country_code` — код страны фонда;

- `investment_rounds` — число инвестиционных раундов, в которых фонд принимал участие;
- `invested_companies` — число компаний, в которые инвестировал фонд;
- `milestones` — количество важных этапов в истории фонда;
- `created_at` — дата и время создания записи в таблице;
- `updated_at` — дата и время обновления записи в таблице.

## Funding\_round

Содержит информацию о раундах инвестиций:

- первичный ключ `id` — уникальный номер инвестиционного раунда;
- внешний ключ `company_id` — ссылается на таблицу `company` — уникальный номер компании, участвовавшей в инвестиционном раунде;
- `funded_at` — дата проведения раунда;
- `funding_round_type` — тип инвестиционного раунда, например:
  - `venture` — венчурный раунд;
  - `angel` — ангельский раунд;
  - `series_a` — раунд A.
- `raised_amount` — сумма инвестиций, которую привлекла компания в этом раунде в долларах;
- `pre_money_valuation` — предварительная, проведённая до инвестиций оценка стоимости компании в долларах;
- `participants` — количество участников инвестиционного раунда;
- `is_first_round` — является ли этот раунд первым для компании;
- `is_last_round` — является ли этот раунд последним для компании;
- `created_at` — дата и время создания записи в таблице;
- `updated_at` — дата и время обновления записи в таблице.

## Investment

Содержит информацию об инвестициях венчурных фондов в компании-стартапы:

- первичный ключ `id` — уникальный номер инвестиции;
- внешний ключ `funding_round_id` — ссылается на таблицу `funding_round` — уникальный номер раунда инвестиции;
- внешний ключ `company_id` — ссылается на таблицу `company` — уникальный номер компании-стартапа, в которую инвестируют;
- внешний ключ `fund_id` — ссылается на таблицу `fund` — уникальный номер фонда, инвестирующего в компанию-стартап;
- `created_at` — дата и время создания записи в таблице;
- `updated_at` — дата и время обновления записи в таблице.

## People

Содержит информацию о сотрудниках компаний-стартапов:

- первичный ключ `id` — уникальный номер сотрудника;
- `first_name` — имя сотрудника;
- `last_name` — фамилия сотрудника;
- внешний ключ `company_id` — ссылается на таблицу `company` — уникальный номер компании-стартапа;
- `twitter_username` — профиль сотрудника в твиттере;

- created\_at — дата и время создания записи в таблице;
- updated\_at — дата и время обновления записи в таблице.

## Исследуем рынок венчурных фондов и инвестиций в компании-стартапы с помощью запросов к базе данных:

1. Посчитаем, сколько компаний закрылось:

```
SELECT COUNT(id)

FROM company

WHERE status = 'closed'
```

Результат

count
2584

2. Отобразим количество привлечённых средств для новостных компаний США. Отсортируем таблицу по убыванию суммы привлечённых инвестиций в долларах. Ограничим таблицу 5 строками:

```
SELECT SUM(funding_total)
FROM company
WHERE category_code = 'news' AND
country_code = 'USA'
GROUP BY id
ORDER BY SUM(funding_total) DESC
LIMIT 5
```

Результат

sum
6.22553e+08
2.5e+08
1.605e+08
1.28e+08
1.265e+08

3. Найдем общую сумму сделок по покупке одних компаний другими в долларах. Отберем 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 acquired_at) BETWEEN 2011 AND 2013
```

Результат

sum
1.37762e+11

4. Отобразим 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%'
```

Результат

first_name	last_name	twitter_username
Rebecca	Silver	SilverRebecca
Silver	Teede	SilverMatrixx
Mattias	Guilotte	Silverreven

5. Выведем 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%'
```

Результат

id	first_name	last_name	company_id	twitter_username	created_at	updated_at
63081	Gregory	Kim		gmoney75	2010-07-13 03:46:28	2011-12-12 22:01:34

6. Для каждой страны отобразим общую сумму привлечённых инвестиций, которые получили компании, зарегистрированные в этой стране. Ограничим таблицу 15 странами:

```
SELECT country_code,  
       SUM(funding_total)  
FROM company  
GROUP BY country_code  
ORDER BY SUM(funding_total) DESC  
LIMIT 15
```

Результат

country_code	sum
USA	3.10588e+11
GBR	1.77056e+10
CHN	1.06897e+10
CAN	9.86636e+09
IND	6.14141e+09
DEU	5.76577e+09
FRA	4.59514e+09
ISR	4.48009e+09
CHE	2.82925e+09
NLD	2.35397e+09

7. Составим таблицу, в которую войдёт дата проведения раунда, а также минимальное и максимальное значения суммы инвестиций, привлечённых в эту дату. Оставим в итоговой таблице только те записи, в которых минимальное значение суммы инвестиций не равно нулю и не равно максимальному значению. Ограничим таблицу 10 строками:

```
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)  
LIMIT 10
```

## Результат

<b>funded_at</b>	<b>min</b>	<b>max</b>
2012-08-22	40000	7.5e+07
2010-07-25	3.27825e+06	9e+06
2002-03-01	2.84418e+06	8.95915e+06
2010-10-11	28000	2e+08
2007-01-18	5.5e+06	2.3e+07
2007-02-27	1.29e+06	3.6e+07
2006-01-05	8.9e+06	2.65e+07
2011-10-31	35000	2.5e+07
2012-10-27	500000	9.3e+06
2007-08-16	2.51989e+06	9e+06

8. Создадим поле с категориями:

- Для фондов, инвестирующих в 100 и более компаний, назначим категорию high\_activity;
- Для фондов, инвестирующих в 20 и более компаний до 100 - middle\_activity;
- Если количество инвестируемых компаний фонда не достигает 20 - low\_activity.

Отообразим все поля таблицы fund и новое поле с категориями:

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



Результат								
activity	id	name	found ed_at	domain	country _code	investment _rounds	invested_ companies	mi les to ne s
high_ activity	1	Greylock Partners	1965- 01-01	greylock. com	USA	307	196	0
middle_ activity	10	Mission Ventures	1996- 01-01	missionve ntures. com	USA	58	33	0
low_ activity	100	Kapor Enterpris es		kei.com	USA	2	1	0
low_ activity	1000	Speed Ventures				0	0	1
low_ activity	10001	Salem Partners	1997- 01-01	salempart ners.com	USA	1	1	0
low_ activity	10002	3T Capital		3tcapital. com	FRA	3	3	0
low_ activity	10003	Merieux Develop pement	2009- 01-01	merieux- developpe ment.com	FRA	2	2	0
low_ activity	10004	Aquasou rca		aquasourc a.com	FRA	1	1	0

9. Посчитаем для каждой из категорий округлённое до ближайшего целого числа среднее количество инвестиционных раундов, в которых фонд принимал участие. Выведем на экран категории и среднее число инвестиционных раундов. Отсортируем 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 avg_investment_rounds
FROM fund
GROUP BY activity
ORDER BY avg_investment_rounds
```

Результат

activity	avg_investment_rounds
low_activity	2
middle_activity	51
high_activity	252

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

Выгрузим десять самых активных стран-инвесторов: отсортируем таблицу по среднему количеству компаний от большего к меньшему. Затем добавим сортировку по коду страны в лексикографическом порядке. Ограничим таблицу 10 странами:

```
SELECT country_code,
    MIN(invested_companies),
    MAX(invested_companies),
    AVG(invested_companies)
FROM fund
WHERE EXTRACT(YEAR FROM founded_at) BETWEEN 2010 AND 2012
GROUP BY country_code
HAVING MIN(invested_companies) <> 0
ORDER BY AVG(invested_companies) DESC, country_code
LIMIT 10
```

Результат

country_code	min	max	avg
BGR	25	35	30
CHL	29	29	29

country_code	min	max	avg
UKR	8	10	9
LTU	5	5	5
IRL	4	5	4.5
KEN	3	3	3
LBN	3	3	3
MUS	3	3	3
JPN	1	6	2.83333
HKG	2	3	2.66667

11. Отообразим имя и фамилию всех сотрудников стартапов. Добавим поле с названием учебного заведения, которое окончил сотрудник, если эта информация известна. Ограничим таблицу 5 строками:

```
SELECT p.first_name,  
       p.last_name,  
       e.institution  
FROM people p  
LEFT JOIN education e ON p.id = e.person_id  
LIMIT 5
```

Результат

first_name	last_name	institution
John	Green	Washington University, St. Louis
John	Green	Boston University
David	Peters	Rice University
Dan	Birdwhistell	University of Cambridge
Gal	Cohen	Tel Aviv University

12. Для каждой компании найдём количество учебных заведений, которые окончили её сотрудники. Выведем название компании и число уникальных названий учебных заведений. Составим топ-5 компаний по количеству университетов:

```
SELECT c.name,  
       COUNT(DISTINCT e.institution)  
FROM company c  
LEFT JOIN people p ON c.id = p.company_id  
INNER JOIN education e ON p.id = e.person_id  
GROUP BY c.name  
ORDER BY COUNT(DISTINCT e.institution) DESC  
LIMIT 10
```

Результат

name	count
Google	167
Yahoo!	115
Microsoft	111
Knight Foundation	74
Comcast	66
McKinsey & Company	52
Amazon	52
TechCrunch	43
MySpace	42
eBay	42

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

1 вариант:

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

2 вариант:

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

Результат

name
10BestThings
11i Solutions
169 ST.
1bib
1Cast
1DayMakeover
25eight
27 Perry
2Win-Solutions
3Touch

14. Составим список уникальных номеров сотрудников, которые работают в компаниях, отобранных в предыдущем задании. Ограничим таблицу 5 строками:

```
SELECT DISTINCT p.id
FROM people p
INNER JOIN company c ON c.id = p.company_id
INNER JOIN funding_round fr ON fr.company_id = c.id
WHERE fr.is_first_round = 1 AND
      fr.is_last_round = 1 AND
      status = 'closed'
LIMIT 5
```

Результат

id
62
97
98
225
226

15. Составим таблицу, куда войдут уникальные пары с номерами сотрудников из предыдущей задачи и учебным заведением, которое окончил сотрудник. Ограничим таблицу 5 строками:

```
SELECT DISTINCT p.id,
                e.institution
FROM people p
INNER JOIN education e ON p.id = e.person_id
INNER JOIN company c ON c.id = p.company_id
INNER JOIN funding_round fr ON fr.company_id = c.id
WHERE fr.is_first_round = 1 AND
      fr.is_last_round = 1 AND
      status = 'closed'
LIMIT 5
```

Результат

id	institution
349	AKI
349	ArtEZ Hogeschool voor de Kunsten
349	Rijks Akademie
699	Imperial College
779	Harvard University

16. Посчитаем количество учебных заведений для каждого сотрудника из предыдущего задания. Ограничим таблицу 5 строками:

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

SELECT a.id,
       COUNT(a.institution)
FROM a
GROUP BY a.id
LIMIT 5
```

Результат

id	count
8666	2
9397	2
6087	1
9643	2
73469	1

17. Дополним предыдущий запрос и выведем среднее число учебных заведений (всех, не только уникальных), которые окончили сотрудники разных компаний:

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

            GROUP BY p.id)

SELECT AVG(a.number_of_institutions) AS avg_number_of_institutions
FROM a
```

Результат

avg_number_of_institutions
1.41509

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

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

SELECT AVG(a.number_of_institutions) AS avg_facebook
FROM a
```

Результат

avg_facebook
1.51111

19. Составим 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. Ограничим таблицу 5 строками:

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

LIMIT 5
```

Результат

name_of_fund	name_of_company	amount
SAP Ventures	OpenX	2.50112e+07
Samsung Ventures	OpenX	2.50112e+07
Index Ventures	OpenX	2.50112e+07
Presidio Ventures	OpenX	2.50112e+07
DAG Ventures	Gigya	2.5e+07



20. Выгрузим 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.

При этом не будем учитывать те сделки, в которых сумма покупки равна нулю. Если сумма инвестиций в компанию равна нулю, также исключим такую компанию из таблицы.

Отсортируем таблицу по сумме сделки от большей к меньшей, а затем по названию купленной компании в лексикографическом порядке. Ограничим таблицу 10 строками:

```
SELECT c.name AS acquiring_company,  
       a.price_amount,  
       c_1.name AS acquired_company,  
       c_1.funding_total,  
       ROUND(a.price_amount / c_1.funding_total) AS share  
FROM acquisition AS a  
LEFT JOIN company c ON a.acquiring_company_id = c.id  
LEFT JOIN company c_1 ON a.acquired_company_id = c_1.id  
WHERE a.price_amount > 0 AND  
       c_1.funding_total > 0  
ORDER BY a.price_amount DESC, acquired_company  
LIMIT 10
```

Результат

acquiring_company	price_amount	acquired_company	funding_total	share
Microsoft	8.5e+09	Skype	7.6805e+07	111
Scout Labs	4.9e+09	Varian Semiconductor Equipment Associates	4.8e+06	1021
Broadcom	3.7e+09	Aeluros	7.97e+06	464
Broadcom	3.7e+09	NetLogic Microsystems	1.88527e+08	20
Level 3 Communications	3e+09	Global Crossing	4.1e+07	73
Yahoo!	2.87e+09	GeoCities	4e+07	72
eBay	2.6e+09	Skype	7.6805e+07	34
Salesforce	2.5e+09	ExactTarget	2.3821e+08	10
Johnson & Johnson	2.3e+09	Crucell	4.43e+08	5
IAC	1.85e+09	Ask.com	2.5e+07	74

21. Выгрузим таблицу, в которую войдут названия компаний из категории `social`, получившие финансирование с 2010 по 2013 год включительно. Сумма инвестиций не должна быть равна нулю. Выведем также номер месяца, в котором проходил раунд финансирования. Ограничим таблицу 5 строками:

```
SELECT c.name AS company_name,  
       EXTRACT(MONTH FROM funded_at) AS month  
FROM company c  
INNER JOIN funding_round fr ON fr.company_id = c.id  
WHERE category_code = 'social' AND  
       fr.raised_amount > 0 AND  
       EXTRACT(YEAR FROM funded_at) BETWEEN 2010 AND 2013  
LIMIT 5
```

Результат

company_name	month
Klout	1
WorkSimple	3
HengZhi	1
Twitter	1
SocialGO	1

22. Отберём данные по месяцам с 2010 по 2013 год, когда проходили инвестиционные раунды. В итоговой таблице должны быть следующие 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 a AS (SELECT EXTRACT(MONTH FROM fr.funded_at) AS month,  
                COUNT(DISTINCT f.name) AS fund_amount  
            FROM funding_round fr  
            LEFT JOIN investment i ON i.funding_round_id = fr.id  
            LEFT JOIN fund f ON i.fund_id = f.id  
            WHERE country_code = 'USA' AND  
                  EXTRACT(YEAR FROM fr.funded_at) BETWEEN 2010 AND 2013  
            GROUP BY month),  
  
      b AS (SELECT EXTRACT(MONTH FROM acquired_at) AS month,  
                COUNT(acquired_company_id) AS company_amount,  
                SUM(price_amount) AS total_price  
            FROM acquisition a  
            WHERE EXTRACT(YEAR FROM acquired_at) BETWEEN 2010 AND 2013  
            GROUP BY month)  
  
SELECT a.month,  
       a.fund_amount,  
       b.company_amount,  
       b.total_price  
FROM a  
LEFT JOIN b ON a.month = b.month
```

Результат

month	fund_amount	company_amount	total_price
1	815	600	2.71083e+10
2	637	418	4.13903e+10
3	695	458	5.95016e+10
4	718	411	3.03837e+10
5	695	532	8.60122e+10
6	785	525	5.20883e+10
7	803	488	4.98541e+10
8	726	454	7.77093e+10
9	793	491	6.97409e+10
10	764	473	4.85567e+10
11	661	414	4.79386e+10
12	590	433	3.74251e+10

23. Составим сводную таблицу и выведем среднюю сумму инвестиций для стран, в которых есть стартапы, зарегистрированные в 2011, 2012 и 2013 годах. Отсортируем таблицу по среднему значению инвестиций за 2011 год от большего к меньшему. Ограничим таблицу 15 странами:

```
WITH a AS (SELECT country_code,
  AVG(funding_total) AS avg_2011
  FROM company
  WHERE EXTRACT(YEAR FROM founded_at) = 2011
  GROUP BY country_code),

  b AS (SELECT country_code,
  AVG(funding_total) AS avg_2012
  FROM company
  WHERE EXTRACT(YEAR FROM founded_at) = 2012
  GROUP BY country_code),

  c AS (SELECT country_code,
  AVG(funding_total) AS avg_2013
  FROM company
  WHERE EXTRACT(YEAR FROM founded_at) = 2013
  GROUP BY country_code)
```

```

SELECT a.country_code,
       a.avg_2011,
       b.avg_2012,
       c.avg_2013
FROM a
INNER JOIN b ON a.country_code = b.country_code
INNER JOIN c ON a.country_code = c.country_code
ORDER BY a.avg_2011 DESC
LIMIT 15

```

Результат

country_code	avg_2011	avg_2012	avg_2013
PER	4e+06	41000	25000
USA	2.24396e+06	1.20671e+06	1.09336e+06
HKG	2.18078e+06	226227	0
PHL	1.75e+06	4218.75	2500
ARE	1.718e+06	197222	35333.3
JPN	1.66431e+06	674720	50000
AUT	1.5342e+06	147806	85773.3
BRA	1.38007e+06	240639	67944.4
DEU	1.1288e+06	1.32915e+06	66612.7
ISR	1.03076e+06	1.27121e+06	294022
PST	1e+06	0	0
FRA	977874	291227	642083
CHN	975918	611436	1e+06
AUS	963088	192949	26313.7
ZAF	962000	576000	0