

Венчурные фонды

Оглавление

- [Введение](#)
- [Описание БД](#)
- [Задача 1](#)
- [Задача 2](#)
- [Задача 3](#)
- [Задача 4](#)
- [Задача 5](#)
- [Задача 6](#)
- [Задача 7](#)
- [Задача 8](#)
- [Задача 9](#)
- [Задача 10](#)
- [Задача 11](#)
- [Задача 12](#)
- [Задача 13](#)
- [Задача 14](#)
- [Задача 15](#)
- [Задача 16](#)
- [Задача 17](#)
- [Задача 18](#)
- [Задача 19](#)
- [Задача 20](#)
- [Задача 21](#)
- [Задача 22](#)
- [Задача 23](#)

Введение ▲

В самостоятельном проекте этого курса вы будете работать с базой 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.

Анализировать рынок инвестиций без подготовки может быть непросто. Поэтому сначала познакомьтесь с важными понятиями, которые вам 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.

- 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.

Таблица `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, `GBR` 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.

Таблица `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.

Таблица `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.

Таблица `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 А.
- `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.

Таблица `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.

Таблица `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.

Задача 1 ▲

Посчитайте, сколько компаний закрылось.

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

Результат

count
2584

Задача 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
```

Результат	
	funding_total
	6.22553e+08
	2.5e+08
	1.605e+08
	1.28e+08

Задача 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 acquired_at) BETWEEN 2011 AND 2013
```

Результат	
	sum
	1.37762e+11

Задача 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%'
```

Результат		
first_name	last_name	twitter_username
Rebecca	Silver	SilverRebecca
Silver	Teede	SilverMatrixx
Mattias	Guillotte	Silverreven

Задача 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%'
```

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

Задача 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_total
FROM company
GROUP BY country_code
ORDER BY fund_total DESC
```

Результат

country_code	fund_total
USA	3.10588e+11
GBR	1.77056e+10
	1.08559e+10
CHN	1.06897e+10

Задача 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) AS min_fund,
    MAX(raised_amount) AS max_fund
FROM funding_round
```

```

GROUP BY funded_at
HAVING
    MIN(raised_amount) <> 0
    AND
    MIN(raised_amount) <> MAX(raised_amount)

```

Результат		
funded_at	min_fund	max_fund
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

Задача 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
FROM fund

```

Результат											
id	name	founded_at	domain	twitter_username	country_code	investment_rounds	invested_companies	milestones	created_at	updated_at	case
13131						0	0	0	2013-08-19 18:46:55	2013-08-19 19:55:07	low_activity
1	Greylock Partners	1965-01-01	greylock.com	greylockvc	USA	307	196	0	2007-05-25 20:18:23	2012-12-27 00:42:24	high_activity
10	Mission Ventures	1996-01-01	missionventures.com		USA	58	33	0	2007-06-05 05:24:58	2013-10-10 22:06:31	middle_activity
100	Kapor Enterprises, Inc.		kel.com		USA	2	1	0	2007-07-12 09:42:21	2008-11-21 05:41:53	low_activity

Задача 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
category_table 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
    activity,
    ROUND(AVG(investment_rounds)) AS avg_rounds
FROM category_table
GROUP BY activity
ORDER BY avg_rounds
```

Результат

activity	avg_rounds
low_activity	2
middle_activity	51
high_activity	252

Задача 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, а затем по коду страны в лексикографическом порядке.

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

Результат			
country_code	min_found	max_found	avg_found
BGR	25	35	30
CHL	29	29	29
UKR	8	10	9
LTU	5	5	5
IRL	4	5	4.5

Задача 11 ▲

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

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

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

Задача 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.

```
WITH
company_people AS (SELECT
    company.name AS company_name,
    people.id AS employee_id
FROM company
JOIN people ON company.id=people.company_id),

full_farm AS (SELECT
    company_name,
    institution
FROM company_people
JOIN education ON company_people.employee_id=education.person_id
WHERE education.institution IS NOT NULL
```

```
GROUP BY
```

```
company_name,  
institution)
```

```
SELECT
```

```
    company_name,  
    COUNT(institution) AS count_educations  
FROM full_farm  
GROUP BY company_name  
ORDER BY count_educations DESC  
LIMIT 5
```

Результат

company_name	count_educations
Google	167
Yahoo!	115
Microsoft	111
Knight Foundation	74
Comcast	66

Задача 13 ▲

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

```
WITH
```

```
f1 AS (SELECT DISTINCT company_id  
      FROM funding_round  
      WHERE  
          is_first_round = 1  
      AND  
          is_last_round = 1  
      ORDER BY company_id)
```

```
SELECT name  
FROM f1  
JOIN company AS c ON f1.company_id=c.id  
WHERE status = 'closed'
```

Результат

name
Omnidrive
Wis.dm
Babelgum
Video Passports

Задача 14 ▲

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

```
WITH
f1 AS (SELECT DISTINCT company_id
      FROM funding_round
      WHERE
        is_first_round = 1
        AND
        is_last_round = 1
      ORDER BY company_id),

company_filter AS (SELECT company_id AS filtered_comany_id
                  FROM f1
                  JOIN company AS c ON f1.company_id=c.id
                  WHERE status = 'closed')

SELECT people.id
FROM people
JOIN company_filter ON people.company_id=company_filter.filtered_comany_id
```

Результат

id
4989
359
358
701

Задача 15 ▲

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

```
WITH
f1 AS (SELECT DISTINCT company_id
      FROM funding_round
      WHERE
        is_first_round = 1
        AND
        is_last_round = 1
      ORDER BY company_id),

company_filter AS (SELECT company_id AS filtered_comany_id
                  FROM f1
                  JOIN company AS c ON f1.company_id=c.id
                  WHERE status = 'closed'),

people_filter AS (SELECT people.id
                  FROM people
                  JOIN company_filter ON
people.company_id=company_filter.filtered_comany_id)

SELECT DISTINCT
  people_filter.id,
```

```

        institution
FROM people_filter
JOIN education ON people_filter.id=education.person_id

```

Результат

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

Задача 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.

```

WITH
f1 AS (SELECT DISTINCT company_id
      FROM funding_round
      WHERE
        is_first_round = 1
        AND
        is_last_round = 1
      ORDER BY company_id),

company_filter AS (SELECT company_id AS filtered_comany_id
                  FROM f1
                  JOIN company AS c ON f1.company_id=c.id
                  WHERE status = 'closed'),

people_filter AS (SELECT people.id
                  FROM people
                  JOIN company_filter ON
people.company_id=company_filter.filtered_comany_id),

people_education AS (SELECT
                    people_filter.id,
                    institution
                  FROM people_filter
                  JOIN education ON people_filter.id=education.person_id)

SELECT
    id,
    COUNT(institution)
FROM people_education
GROUP BY id

```

Результат

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

Задача 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
f1 AS (SELECT DISTINCT company_id
      FROM funding_round
      WHERE
        is_first_round = 1
        AND
        is_last_round = 1
      ORDER BY company_id),

company_filter AS (SELECT company_id AS filtered_comany_id
                  FROM f1
                  JOIN company AS c ON f1.company_id=c.id
                  WHERE status = 'closed'),

people_filter AS (SELECT people.id
                  FROM people
                  JOIN company_filter ON
people.company_id=company_filter.filtered_comany_id),

people_education AS (SELECT
                    people_filter.id,
                    institution
                  FROM people_filter
                  JOIN education ON people_filter.id=education.person_id),

count_of_educations AS (SELECT
                        id,
                        COUNT(institution) AS count_e
                  FROM people_education
                  GROUP BY id)

SELECT SUM(count_e) / COUNT(*)
FROM count_of_educations
```

Результат

?column?

1.41509

Задача 18 ▲

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

```
WITH
facebook_people AS (SELECT people.id AS employes
                        FROM people
                        JOIN company ON people.company_id=company.id AND company.name
                        = 'Facebook'),

facebook_people_education AS (SELECT
                                employes,
                                COUNT(institution) AS count_i
                                FROM facebook_people
                                JOIN education ON
facebook_people.employes=education.person_id
                                GROUP BY employes)

SELECT SUM(count_i) / COUNT(*)
FROM facebook_people_education
```

Результат

?column?

1.51111

Задача 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 DISTINCT
    fund.name AS name_of_fund,
    company.name AS name_of_company,
    raised_amount
FROM funding_round AS f_r
JOIN company ON f_r.company_id=company.id AND milestones > 6
JOIN investment ON f_r.id=investment.funding_round_id
JOIN fund ON investment.fund_id=fund.id
WHERE EXTRACT(YEAR FROM funded_at) BETWEEN 2012 AND 2013
```

name_of_fund	name_of_company	raised_amount
Accel Partners	OpenX	2.50112e+07
Advance Publication	Gigya	1.53e+07
Benchmark	Gigya	1.53e+07
Benchmark	Gigya	2.5e+07
DAG Ventures	Gigya	1.53e+07

Задача 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

```

    c_ing.name AS acquiring_company_name,
    a.price_amount,
    c_red.name AS acquired_company_name,
    c_red.funding_total,
    ROUND(a.price_amount / c_red.funding_total) AS overprice_of_funding
FROM acquisition as a
LEFT JOIN company AS c_ing ON a.acquiring_company_id=c_ing.id
LEFT JOIN company AS c_red ON a.acquired_company_id=c_red.id
WHERE
    a.price_amount <> 0
    AND
    c_ing.name IS NOT NULL
    AND
    c_red.name IS NOT NULL
    AND c_red.funding_total <> 0
ORDER BY a.price_amount DESC, c_red.name
LIMIT 10

```


Результат

acquiring_company_name	price_amount	acquired_company_name	funding_total	overprice_of_funding
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

Задача 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.

WITH

```
social_companies AS (SELECT
                        id,
                        name AS company_name
                      FROM company
                      WHERE category_code = 'social')
```

SELECT

```
    company_name,
    EXTRACT(MONTH FROM funded_at) AS founded_month
FROM funding_round
JOIN social_companies AS s_c ON funding_round.company_id=s_c.id
WHERE
    EXTRACT(YEAR FROM funded_at) BETWEEN 2010 AND 2013
    AND
    raised_amount <> 0
```

Результат

company_name	founded_month
gorogo	4
CrushBlvd	1
Collective Bias	4
RuffWire	6
Get.com	9

Задача 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

```
funds_from_usa AS (SELECT
    i.funding_round_id,
    fund.id AS fund_company_id
FROM fund
JOIN investment AS i ON fund.id=i.fund_id
WHERE fund.country_code = 'USA'),

acquired_info AS (SELECT
    EXTRACT(MONTH FROM acquired_at) AS acquired_month,
    COUNT(acquired_company_id) AS count_of_acquired_companies,
    SUM(price_amount) AS sum_of_acquired
FROM acquisition
WHERE
    acquired_at IS NOT NULL
    AND
    EXTRACT(YEAR FROM acquired_at) BETWEEN 2010 AND 2013
GROUP BY acquired_month),

funding_round_month AS (SELECT
    EXTRACT(MONTH FROM funded_at) AS fund_month,
    funding_round.id AS funding_round_id
FROM funding_round
WHERE EXTRACT(YEAR FROM funded_at) BETWEEN 2010 AND 2013),

temp_table AS (SELECT
    fund_month,
    fund_company_id
FROM funds_from_usa
JOIN funding_round_month ON
funds_from_usa.funding_round_id=funding_round_month.funding_round_id
GROUP BY fund_month, fund_company_id),

month_and_company AS (SELECT
    fund_month,
    COUNT(fund_company_id) AS count_company_fund
FROM temp_table
GROUP BY fund_month)

SELECT
    fund_month,
    count_company_fund,
    count_of_acquired_companies,
    sum_of_acquired
FROM month_and_company
JOIN acquired_info ON month_and_company.fund_month=acquired_info.acquired_month
ORDER BY fund_month
```

Результат

fund_month	count_company_fund	count_of_acquired_companies	sum_of_acquired
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

Задача 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

```
year_2011 AS (SELECT
    country_code,
    AVG(funding_total) AS avg_2011
FROM company
WHERE
    EXTRACT(YEAR FROM founded_at) = 2011
    AND
    country_code IS NOT NULL
GROUP BY country_code),
```

```
year_2012 AS (SELECT
    country_code,
    AVG(funding_total) AS avg_2012
FROM company
WHERE
    EXTRACT(YEAR FROM founded_at) = 2012
    AND
    country_code IS NOT NULL
GROUP BY country_code),
```

```
year_2013 AS (SELECT
    country_code,
    AVG(funding_total) AS avg_2013
FROM company
WHERE
    EXTRACT(YEAR FROM founded_at) = 2013
    AND
    country_code IS NOT NULL
GROUP BY country_code)
```

SELECT

```
    year_2011.country_code,
    avg_2011,
    avg_2012,
    avg_2013
FROM year_2011
JOIN year_2012 ON year_2011.country_code=year_2012.country_code
```

JOIN year_2013 ON year_2011.country_code=year_2013.country_code
ORDER BY avg_2011 DESC

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