

Базовый SQL

Информация о таблицах:

acquisition

Содержит информацию о покупках одних компаний другими.

Таблица включает такие поля:

- 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, `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.

`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.

В самостоятельном проекте вам нужно проанализировать данные о фондах и инвестициях и написать запросы к базе.

1. Посчитайте, сколько компаний закрылось.

```
SELECT funding_total
FROM company
WHERE category_code = 'news'
AND country_code = 'USA'
ORDER BY funding_total DESC;
```

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)) BETWEEN 2011 AND 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 SUM(funding_total),
country_code
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,
MAX(raised_amount),
MIN(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 < 20 THEN 'low_activity'
WHEN invested_companies >= 20 AND invested_companies < 100 THEN 'middle_activity'
ELSE 'high_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 AVG
FROM fund
GROUP BY activity
ORDER BY 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, а затем 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)) BETWEEN 2010 AND 2012)  
GROUP BY country_code  
HAVING MIN(invested_companies) != 0  
ORDER BY AVG(invested_companies) DESC  
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 OUTER 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 COUNT(DISTINCT e.institution),  
c.name  
FROM education AS e  
LEFT JOIN people as p ON e.person_id=p.id  
JOIN company as c ON c.id = p.company_id  
GROUP BY c.name  
ORDER BY COUNT(DISTINCT e.institution) DESC  
LIMIT 5;
```

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


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

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

```
SELECT DISTINCT p.id
FROM people AS p
LEFT JOIN company as c ON p.company_id=c.id
LEFT JOIN funding_round AS fr ON c.id=fr.company_id
WHERE (status = 'closed')
AND is_first_round=1
AND is_last_round=1;
```

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

```
SELECT DISTINCT p.id,
e.institution
FROM education AS e
LEFT JOIN people as p ON p.id=e.person_id
LEFT JOIN company as c ON p.company_id=c.id
LEFT JOIN funding_round AS fr ON c.id=fr.company_id
WHERE (status = 'closed')
AND is_first_round=1
AND is_last_round=1;
```

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 p.id,
COUNT(e.institution) AS total_institution
FROM people AS p JOIN education AS e ON p.id = e.person_id
WHERE company_id IN (SELECT id
FROM company
WHERE id IN (SELECT company_id
FROM funding_round
```

```
WHERE is_first_round = 1 AND is_last_round = 1)
AND status = 'closed')
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.

```
SELECT AVG( best.total_institution)
FROM(
SELECT p.id,
COUNT(e.institution) AS total_institution
FROM people AS p JOIN education AS e ON p.id = e.person_id
WHERE company_id IN (SELECT id
FROM company
WHERE id IN (SELECT company_id
FROM funding_round
WHERE is_first_round = 1 AND is_last_round = 1)
AND status = 'closed')
GROUP BY p.id) AS best;
```

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

- (сервис, запрещённый на территории РФ)

```
SELECT AVG(best.total_institution)
FROM(
SELECT p.id,
COUNT(e.institution) AS total_institution
FROM people AS p JOIN education AS e ON p.id = e.person_id
WHERE company_id IN (SELECT id
FROM company
WHERE name = 'Facebook')
GROUP BY p.id) AS best;
```

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

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.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 percent
FROM acquisition AS a
LEFT JOIN company AS c ON a.acquiring_company_id = c.id
LEFT JOIN company AS 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;
```

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 name,  
EXTRACT(MONTH FROM funded_at) AS month  
FROM company AS c  
LEFT JOIN funding_round AS fr ON c.id = fr.company_id  
WHERE (category_code = 'social')  
AND (fr.raised_amount != 0)  
AND (EXTRACT(YEAR FROM funded_at) BETWEEN 2010 AND 2013);
```

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 month_fund AS  
(SELECT EXTRACT(MONTH  
FROM fr.funded_at) AS MONTH,  
COUNT(DISTINCT f.name) AS count_of_fund  
FROM funding_round AS fr  
LEFT JOIN investment AS i ON i.funding_round_id = fr.id  
LEFT JOIN fund AS f ON i.fund_id = f.id  
WHERE EXTRACT(YEAR  
FROM fr.funded_at) BETWEEN 2010 AND 2013  
AND f.country_code = 'USA'  
GROUP BY MONTH),  
month_acquired AS  
(SELECT EXTRACT(MONTH  
FROM acquired_at) AS MONTH,
```

```

COUNT(acquired_company_id) AS count_of_acquired,
SUM(price_amount) AS sum_of_acquired
FROM acquisition
WHERE EXTRACT(YEAR
FROM acquired_at) BETWEEN 2010 AND 2013
GROUP BY MONTH)
SELECT month_fund.month,
month_fund.count_of_fund,
month_acquired.count_of_acquired,
month_acquired.sum_of_acquired
FROM month_fund
JOIN month_acquired ON month_fund.month = month_acquired.month;

```

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 AS country,
AVG(funding_total) as year_2011
from company c
where
extract(year from founded_at::date) = 2011
group by
country_code
),
year_2012 AS (select
country_code AS country,
AVG(funding_total) as year_2012
from company c
where
extract(year from founded_at::date) = 2012
group by
country_code
),
year_2013 AS (
select
country_code AS country,
AVG(funding_total) as year_2013
from company c
where
extract(year from founded_at::date) = 2013
group by
country_code
)

```

```
select
year_2011.country,
year_2011.year_2011,
year_2012.year_2012,
year_2013.year_2013
from year_2011
INNER JOIN year_2012 ON year_2011.country = year_2012.country
INNER JOIN year_2013 ON year_2012.country = year_2013.country
ORDER BY year_2011.year_2011 DESC;
```