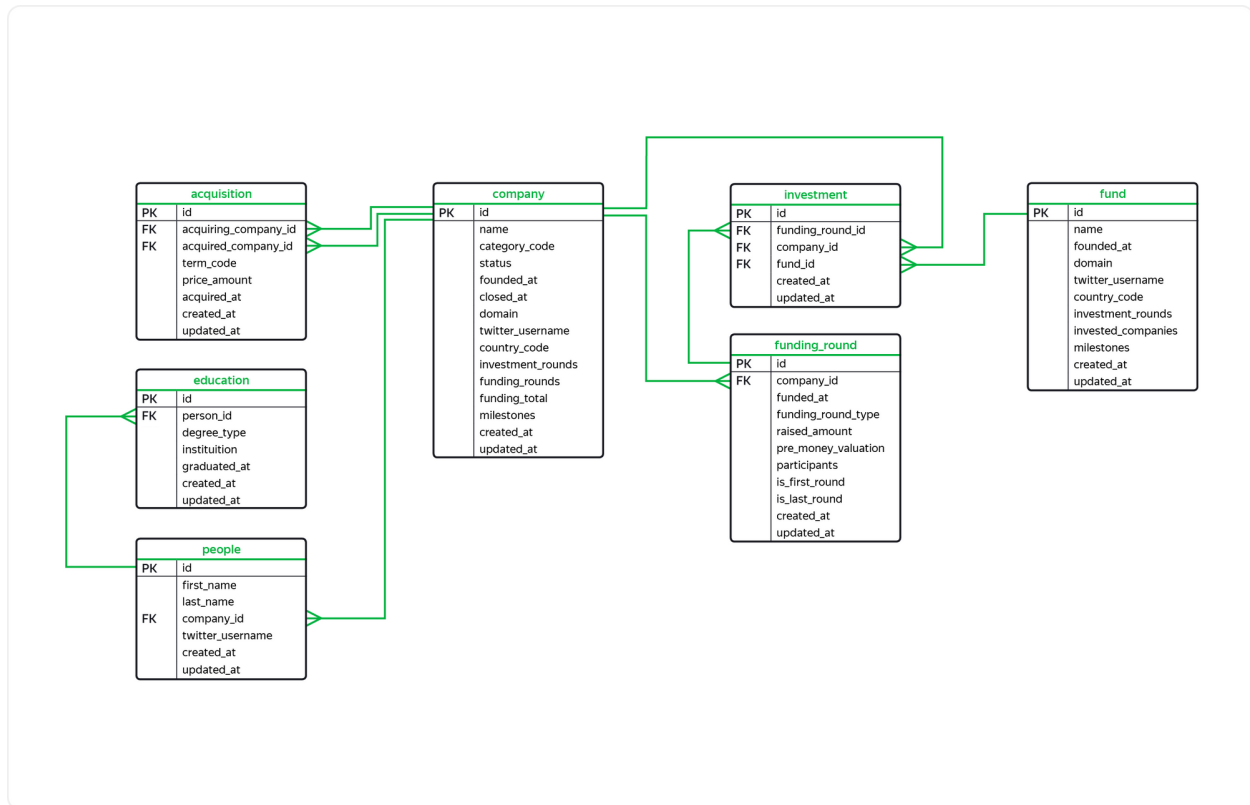


# Проект: Базовый SQL

**Задача проекта:** нужно проанализировать данные о фондах и инвестициях и написать запросы к базе.

**Схема базы данных:**



**Описание базы данных:**

## 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 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.

## 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.

## Задания:

1. Посчитайте, сколько компаний закрылось.

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

count

2584

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 country_code = 'USA' AND category_code = 'news'
ORDER BY funding_total DESC;
```

funding\_total

6.22553e+08

2.5e+08

1.605e+08

1.28e+08

...

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 CAST(acquired_at AS date)) BETWEEN 2011 AND 2013)
```

sum

1.37762e+11

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%'
```

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

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%'
```

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

country_code	sum
USA	3.10588e+11
GBR	1.77056e+10

CHN	1.06897e+10
CAN	9.86636e+09
IND	6.14141e+09
...	...

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),
MAX(raised_amount)
FROM funding_round
GROUP BY funded_at
HAVING MIN(raised_amount) != 0 AND MAX(raised_amount) != MIN(raised_amount)
```

funded_at	min	max
2012-08-22	40000	7.5e+07
2010-07-25	3.27825e+06	9e+06
2002-03-01	2.84418e+06	8.95915e+06
...	...	...

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
```

id	name	founded_at	domain	twitter_username	country_code	investment_rounds	investe
1	Greylock Partners	1965-01-01	greylock.com	greylockvc	USA	307	196
10	Mission Ventures	1996-01-01	missionventures.com		USA	58	33
...	...	...	...	...	...	...	...

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 ROUND(AVG(investment_rounds)),
CASE
    WHEN invested_companies>=100 THEN 'high_activity'
    WHEN invested_companies>=20 THEN 'middle_activity'
    ELSE 'low_activity'
END AS activity
FROM fund
GROUP BY activity
ORDER BY ROUND(AVG(investment_rounds))

```

round	activity
2	low_activity
51	middle_activity
252	high_activity

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. Затем добавьте 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)) BETWEEN 2010 AND 2012
GROUP BY country_code
HAVING MIN(invested_companies) != 0
ORDER BY AVG(invested_companies) DESC
LIMIT 10;

```

country_code	min	max	avg
BGR	25	35	30
CHL	29	29	29
UKR	8	10	9
...	...	...	...

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,
ed.institution
FROM people as p
LEFT JOIN education as ed ON p.id=ed.person_id

```

first_name	last_name	institution	
John	Green	Washington University, St. Louis	
John	Green	Boston University	

David	Peters	Rice University	
...	...	...	

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 DISTINCT c.name,
COUNT(DISTINCT (ed.institution))
FROM company as c
INNER JOIN people as p ON c.id=p.company_id
INNER JOIN education as ed ON p.id=ed.person_id
GROUP BY c.name
ORDER BY COUNT(DISTINCT (ed.institution)) DESC
LIMIT 5
```

name	count
Google	167
Yahoo!	115
Microsoft	111
Knight Foundation	74

13. Составьте 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)
```

name

10BestThings

11i Solutions

169 ST.

1bib

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

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

id

62

97

98



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

```
SELECT e.person_id,
e.institution
FROM education as e
WHERE e.person_id IN (SELECT DISTINCT p.id
                      FROM people as p
                      WHERE p.company_id IN (SELECT DISTINCT c.id
                                           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)))
GROUP BY e.institution, e.person_id
```

person_id	institution
349	AKI
48426	Arg
349	ArtEZ Hogeschool voor de Kunsten
...	...

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
a as (SELECT p.id
      FROM people as p
      WHERE p.company_id IN (SELECT DISTINCT c.id
                           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)))

SELECT a.id,
COUNT(e.institution)
FROM education as e
INNER JOIN a ON a.id=e.person_id
GROUP BY a.id
```

id	count
349	3
699	1
779	2
...	...

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
a as (SELECT person_id,
COUNT(e.institution) as count_institution
```

```

FROM education as e
WHERE person_id IN (SELECT p.id
                     FROM people as p
                     WHERE p.company_id IN (SELECT DISTINCT c.id
                                             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)))
GROUP BY person_id)

SELECT AVG(count_institution)
FROM a

avg

1.41509

```

18. Напишите похожий запрос: выведите среднее число учебных заведений (всех, не только уникальных), которые окончили сотрудники Facebook\*.  
 \*(сервис, запрещённый на территории РФ)

```

WITH
a as (SELECT COUNT(ed.institution) count_institution, person_id
      FROM education as ed
      WHERE person_id IN (SELECT p.id
                          FROM people as p
                          WHERE p.company_id IN (SELECT DISTINCT c.id
                                                  FROM company as c
                                                  WHERE c.name = 'Facebook')))
      GROUP BY person_id)

SELECT AVG(count_institution)
FROM a

avg

1.51111

```

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

```

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
...	...	...

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_acquiring.name,
       price_amount,
       c_acquired.name,
       c_acquired.funding_total as funding_total,
       ROUND(price_amount / funding_total)
FROM (SELECT acquiring_company_id,
              c.name,
              price_amount,
              acquired_company_id
       FROM acquisition as a
       LEFT JOIN company as c ON a.acquiring_company_id=c.id) as c_acquiring
LEFT JOIN company as c_acquired ON c_acquiring.acquired_company_id=c_acquired.id
WHERE price_amount != 0 AND c_acquired.funding_total != 0
ORDER BY price_amount DESC, c_acquired.name
LIMIT 10;
```

name	price_amount	funding_total	round
Skype	8.5e+09	7.6805e+07	111
Varian Semiconductor Equipment Associates	4.9e+09	4.8e+06	1021
Aeluros	3.7e+09	7.97e+06	464
...	...	...	...

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 c.name,
       EXTRACT(MONTH FROM funded_at)
FROM company as c
INNER JOIN funding_round as fr ON c.id=fr.company_id
WHERE category_code = 'social'
```

```
AND EXTRACT(YEAR FROM funded_at) BETWEEN 2010 AND 2013
AND raised_amount != 0
```

name	date_part
Klout	1
WorkSimple	3
HengZhi	1
...	...

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
a as (SELECT EXTRACT(MONTH FROM fr.funded_at) as month,
      COUNT(DISTINCT f.name) unique_name
      FROM funding_round as fr
      INNER JOIN investment as i ON i.funding_round_id=fr.id
      INNER 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 EXTRACT(MONTH FROM fr.funded_at)),
```

```
b as (SELECT EXTRACT(MONTH FROM acquired_at) as month,
      COUNT(acquired_company_id) count_acquired,
      SUM(price_amount) price
      FROM acquisition
      WHERE EXTRACT(YEAR FROM acquired_at) BETWEEN 2010 AND 2013
      GROUP BY EXTRACT(MONTH FROM acquired_at))
```

```
SELECT a.month,
a.unique_name,
b.count_acquired,
b.price
FROM a INNER JOIN b ON a.month=b.month
```

month	unique_name	count_acquired	price
1	815	600	2.71083e+10
2	637	418	4.13903e+10
3	695	458	5.95016e+10
...	...	...	...

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.

```
all_2012 as (SELECT country_code,
      AVG(funding_total) as total_2012
      FROM company
```

```
WHERE EXTRACT(YEAR FROM founded_at) = 2012
GROUP BY country_code),
```

```
all_2013 as (SELECT country_code,
                  AVG(funding_total) as total_2013
              FROM company
              WHERE EXTRACT(YEAR FROM founded_at) = 2013
              GROUP BY country_code)
```

```
SELECT all_2011.country_code,
all_2011.total_2011,
all_2012.total_2012,
all_2013.total_2013
FROM all_2011
INNER JOIN all_2012 ON all_2011.country_code=all_2012.country_code
INNER JOIN all_2013 ON all_2012.country_code=all_2013.country_code
ORDER BY total_2011 DESC;
```

country_code	total_2011	total_2012	total_2013
PER	4e+06	41000	25000
USA	2.24396e+06	1.20671e+06	1.09336e+06
HKG	2.18078e+06	226227	0
...	...	...	...