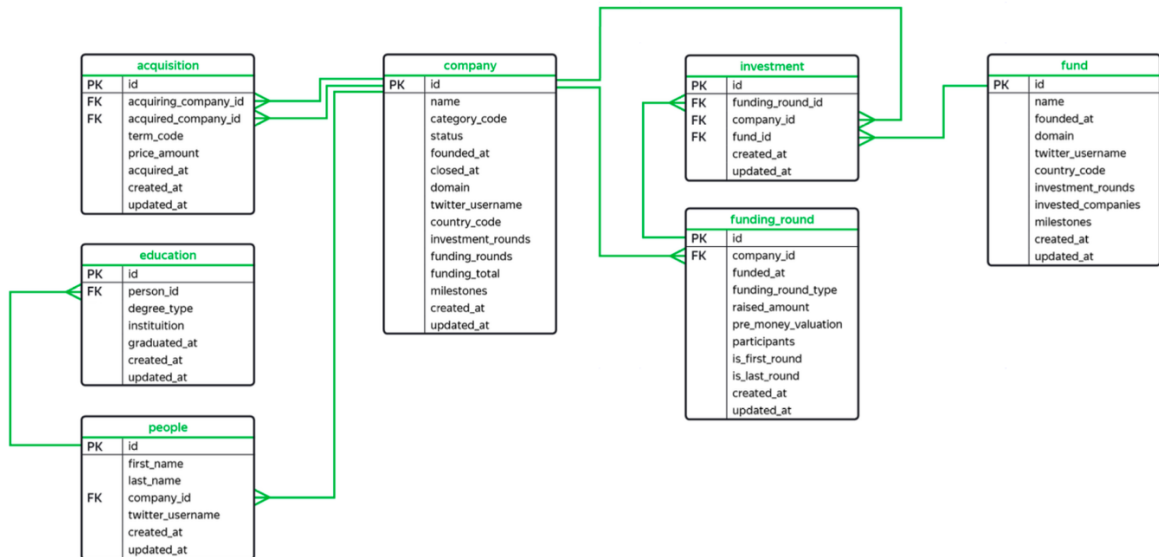


Проект состоит из отдельных базовых запросов SQL к базе данных с информацией об инвестициях, которые фонд направлял в компании.

Описание к структуре базы данных приведена в файле `basic_sql_description.pdf`.



1. Посчитайте, сколько компаний закрылось.

```
select count(id)
from company
where status='closed'
```

2. Отобразите количество привлечённых средств для новостных компаний США. Используйте данные из таблицы `company`. Отсортируйте таблицу по убыванию значений в поле `funding_total`.

```
select sum(price_amount)
from acquisition
where extract(year from acquired_at) between 2011 and 2013
and term_code='cash'
```

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 acquired_at) 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 country_code, sum(funding_total) as funding_total  
from company  
group by country_code  
order by 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, min(raised_amount) as min, max(raised_amount) as max  
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 *,
where
case
when invested_companies >=100 then 'high_activity'
when invested_companies between 20 and 99 then 'middle_activity'
when invested_companies <20 then 'low_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 activity, round(avg_round) from (select avg(investment_rounds) as
avg_round,
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 avg_round) as a
```

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* from(
select country_code, min(invested_companies) as min,
max(invested_companies) as max, avg(invested_companies) as avg
```

```
from fund
where extract(year from founded_at) between 2010 and 2012
group by country_code
having min(invested_companies)!=0
order by avg desc, country_code
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 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.

```
with a as (select id, company_id
from people),

b as (select person_id, institution
from education),

c as (select name, id
from company)

select c.name, d. institutions from c
inner join
(select a. company_id, count(distinct b.institution) as institutions
from a
inner join b on b.person_id=a.id
group by company_id) as d on c. id = d. company_id
order by d. institutions desc
limit 5
```

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

```
with a as (select name, id
```

```
from company
where status='closed'),

b as (select company_id
from funding_round
where is_first_round = is_last_round)
where is_first_round = 1
and is_last_round = 1)

select distinct(name) from a
inner join b on a.id=b.company_id
```

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

```
with a as (select name, id
from company
where status='closed'),

b as (select company_id
from funding_round
where is_first_round = 1
and is_last_round =1)

select id from people
where company_id in
(select id from a
inner join b on a.id=b.company_id)
```

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

```
with a as (select name, id
from company
where status='closed'),

b as (select company_id
from funding_round
where is_first_round = 1
and is_last_round =1)

select distinct c.id, d.institution
```

```
from
(select id from people
where company_id in
(select id from a
inner join b on a.id=b.company_id)) as c
inner join education as d on c.id=d.person_id
```

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 name, id
from company
where status='closed'),
```

```
b as (select company_id
from funding_round
where is_first_round = 1
and is_last_round =1)
```

```
select id, count(institution)
from
(select c.id, d.institution
from
(select id from people
where company_id in
(select id from a
inner join b on a.id=b.company_id)) as c
inner join education as d on c.id=d.person_id) as e
group by 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.

```
with a as (select name, id
from company
where status='closed'),
```

```
b as (select company_id
from funding_round
where is_first_round = 1
and is_last_round =1)
```

```

select avg(institutions)
from
(select id, count(institution) as institutions
from
(select c.id, d.institution
from
(select id from people
where company_id in
(select id from a
inner join b on a.id=b.company_id)) as c
inner join education as d on c.id=d.person_id) as e
group by id) as f

```

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

```

with a as (select id as facebook_id
from company
where name='facebook'),

```

```

b as (select person_id, institution
from education)

```

```

select avg(institutions)
from
(select person_id, count(institution) as institutions
from b
inner join
(select id from people as p
inner join a on p.company_id=a.facebook_id) as f on b.person_id= f.id
group by person_id) as g

```

18. 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 i.fund_id = f.id
inner join
(select*
from funding_round
where funded_at between '2012-01-01' and '2013-12-31')
as fr on fr.id = i.funding_round_id
WHERE c.milestones > 6

```

19. 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.

```

with acquiring as
(select c.name as buyer,
a.price_amount as price,
a.id as key
from acquisition as a
left join company as c on a.acquiring_company_id = c.id
where a.price_amount > 0),
acquired as
(select c.name as acquisition,
c.funding_total as investment,
a.id as key
from acquisition as a
left join company as c on a.acquired_company_id = c.id
where c.funding_total > 0)
select acqn.buyer,
acqn.price,
acqd.acquisition,
acqd.investment,
round(acqn.price / acqd.investment) as uplift
from acquiring as acqn
join acquired as acqd on acqn.key = acqd.key

```


order by price desc, acquisition
limit 10;

20. 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.

```
select c.name as social_co,  
extract (month from fr.funded_at) as funding_month  
from company as c  
left join funding_round as fr on c.id = fr.company_id  
where c.category_code = 'social'  
and fr.funded_at between '2010-01-01' and '2013-12-31'  
and fr.raised_amount <> 0;
```

21. 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 y_11 as  
(select country_code as country,  
avg(funding_total) as y_2011  
from company  
where extract(year from founded_at::date) in(2011, 2012, 2013)  
group by country, extract(year from founded_at)  
having extract(year from founded_at) = '2011'),  
y_12 as  
(select country_code as country,  
avg(funding_total) as y_2012  
from company  
where extract(year from founded_at::date) in(2011, 2012, 2013)  
group by country, extract(year from founded_at)  
having extract(year from founded_at) = '2012'),  
y_13 as  
(select country_code as country,  
avg(funding_total) as y_2013
```

```
from company
where extract(year from founded_at::date) in(2011, 2012, 2013)
group by country, extract(year from founded_at)
having extract(year from founded_at) = '2013')
select y_11.country, y_2011, y_2012, y_2013
from y_11
join y_12 on y_11.country = y_12.country
join y_13 on y_12.country = y_13.country
order by y_2011 desc;
```