1. Базовый SQL

Знакомство с базой данных

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.

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.

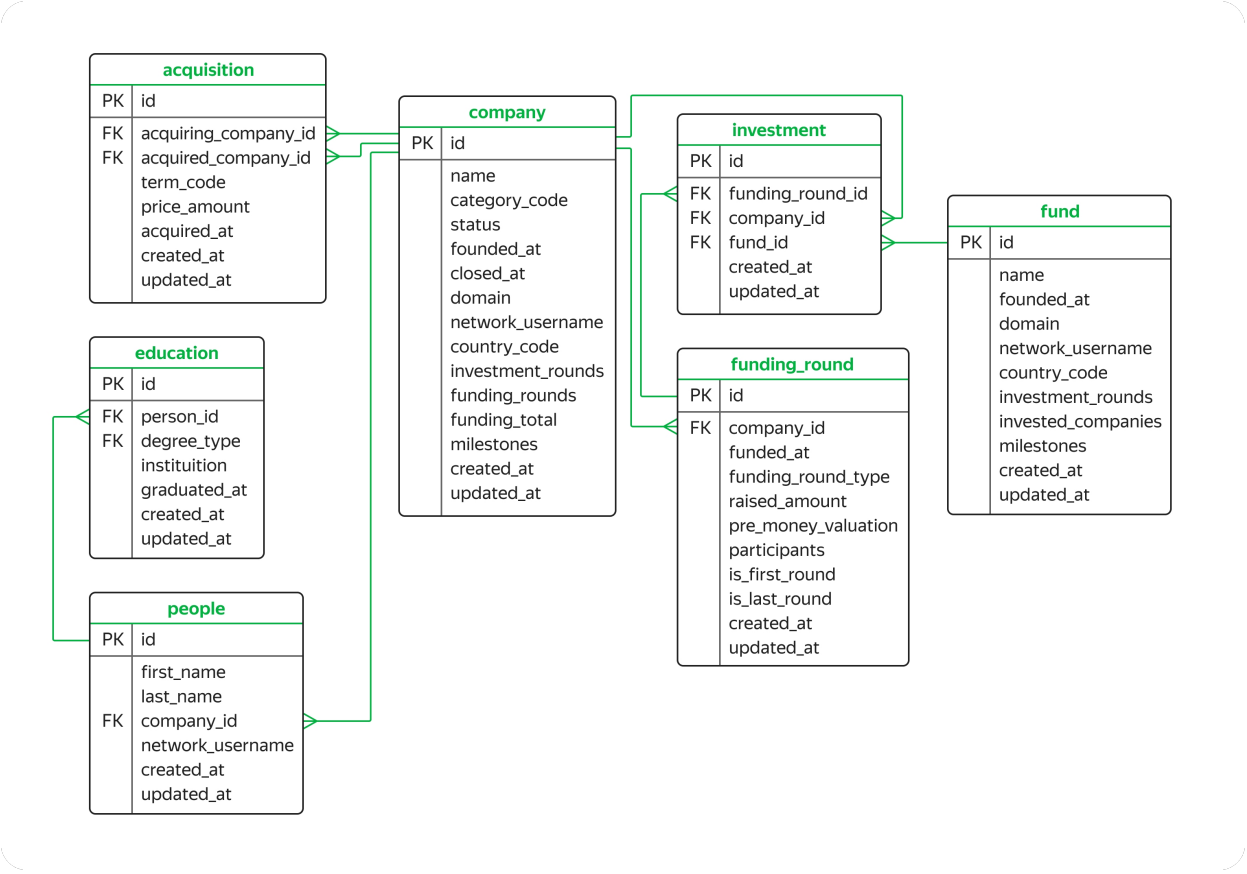
Если компании требуется ментор или наставник – она привлекает бизнес-ангела. Бизнес-ангелы – инвесторы, которые помимо финансовой поддержки предлагают экспертную помощь. Такой раунд называют *ангельским*.

Когда стартап становится компанией с проверенной бизнес-моделью и начинает зарабатывать самостоятельно, предложений инвесторов становится больше. Это раунд A, а за ним следуют и другие: B, C, 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.

В данных об инвестициях вам встретятся упоминания раундов, но самостоятельный проект не предполагает, что вы должны разбираться в их специфике лучше любого инвестора. Главное – понимать, как устроены данные.

Вы уже знаете, что такое ER-диаграмма. Работу с новой базой данных лучше начать с изучения схемы.



Теперь можно познакомиться с данными, которые хранят таблицы.

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:
  1. `cash` – наличными;
  2. `stock` – акциями компании;
  3. `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:
  1. `news` – специализируется на работе с новостями;
  2. `social` – специализируется на социальной работе.
- `status` – статус компании:
  1. `acquired` – приобретена;
  2. `operating` – действует;
  3. `ipo` – вышла на IPO;
  4. `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;
- `network_username` – профиль фонда в корпоративной сети биржи;
- `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:
  1. BA – Bachelor of Arts – бакалавр гуманитарных наук;
  2. MS – Master of Science – магистр естественных наук.
- instituition – 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;
- network_username – профиль фонда в корпоративной сети биржи;
- 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:
  1. venture – венчурный раунд;
  2. angel – ангельский раунд;
  3. 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;
- `network_username` – профиль фонда в корпоративной сети биржи;
- `created_at` – дата и время создания записи в таблице;
- `updated_at` – дата и время обновления записи в таблице.

Задания

1.Отобразите все записи из таблицы `company` по компаниям, которые закрылись.

```
SELECT *
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 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 term_code = 'cash' AND acquired_at BETWEEN '2011-01-01' AND '2013-12-31'
```

4.Отобразите имя, фамилию и названия аккаунтов людей в поле `network_username`, у которых названия аккаунтов начинаются на `'Silver'`.

```sql
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 поле `network_username` содержат подстроку `'money'`, а фамилия начинается на `'K'`.

```sql
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.

```sql
SELECT country_code, SUM(funding_total)
FROM company
GROUP BY 1
ORDER BY 2 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.

```sql
SELECT funded_at, MIN(raised_amount), MAX(raised_amount)
FROM funding_round
GROUP BY 1
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 fund.*,
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 AS cat
FROM fund
```

9.Для каждой из категорий, назначенных в предыдущем задании, посчитайте округлённое до ближайшего целого числа среднее количество инвестиционных раундов, в которых фонды принимали 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))
FROM fund
GROUP BY activity
ORDER BY 2;
```

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 founded_at) BETWEEN '2010' AND '2012'
GROUP BY 1
HAVING MIN(invested_companies) <> 0
ORDER BY AVG(invested_companies) DESC, 1
LIMIT 10
```

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.instituition
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.

```
SELECT c.name, COUNT(DISTINCT e.instituition)
FROM company AS c
INNER JOIN people AS p ON c.id = p.company_id
LEFT JOIN education AS e ON e.person_id = p.id
```

```
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5
```

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

```
SELECT DISTINCT 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 company AS c
LEFT JOIN people AS p ON c.id = p.company_id
WHERE company_id IN (SELECT DISTINCT c.id
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)
```

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

```
SELECT DISTINCT t1.id, e.instituition
FROM
(SELECT DISTINCT p.id
FROM company AS c
LEFT JOIN people AS p ON c.id = p.company_id
WHERE company_id IN (SELECT DISTINCT c.id
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)) AS t1
```

```
LEFT JOIN education AS e ON t1.id = e.person_id
WHERE e.instituition IS NOT NULL
```

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 t1.id, COUNT(e.instituition)
FROM
(SELECT DISTINCT p.id
FROM company AS c
LEFT JOIN people AS p ON c.id = p.company_id
WHERE company_id IN (SELECT DISTINCT c.id
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)) AS t1
LEFT JOIN education AS e ON t1.id = e.person_id
WHERE e.instituition IS NOT NULL
GROUP BY 1
```

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
t1 AS
(SELECT t1.id, COUNT(e.instituition) as cnt
FROM
(SELECT DISTINCT p.id
FROM company AS c
LEFT JOIN people AS p ON c.id = p.company_id
WHERE company_id IN (SELECT DISTINCT c.id
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)) AS t1
```

```
LEFT JOIN education AS e ON t1.id = e.person_id
WHERE e.instituition IS NOT NULL
GROUP BY 1)

SELECT AVG(cnt) FROM t1;
```

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

```
WITH
t1 AS
(SELECT t1.id, COUNT(e.instituition) as cnt
FROM
(SELECT DISTINCT p.id
FROM company AS c
LEFT JOIN people AS p ON c.id = p.company_id
WHERE company_id IN (SELECT DISTINCT c.id
FROM company AS c
WHERE name = 'Facebook')) AS t1
LEFT JOIN education AS e ON t1.id = e.person_id
WHERE e.instituition IS NOT NULL
GROUP BY 1)

SELECT AVG(cnt) FROM t1;
```

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 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;
```

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.

```
WITH
t1 AS
(SELECT c.name AS buyer,
        a.price_amount AS price,
        a.id AS id
FROM acquisition AS a
LEFT JOIN company AS c ON a.acquiring_company_id = c.id
WHERE a.price_amount > 0),


t2 AS
```

```sql
(SELECT c.name AS name,
        c.funding_total AS funding_total,
        a.id AS id
FROM acquisition AS a
LEFT JOIN company AS c ON a.acquired_company_id = c.id
WHERE c.funding_total > 0)


SELECT t1.buyer,
       t1.price,
       t2.name,
       t2.funding_total,
       ROUND(t1.price / t2.funding_total) AS part
FROM t1
JOIN t2 ON t1.id = t2.id
ORDER BY price DESC, name
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.

```sql
SELECT c.name, EXTRACT(MONTH FROM funded_at)
FROM company AS c
LEFT JOIN funding_round as fr ON fr.company_id = c.id
WHERE c.category_code = 'social'
AND funded_at BETWEEN '2010-01-01' AND '2013-12-31'
AND raised_amount <> 0
```

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
t1 AS
(SELECT EXTRACT(MONTH FROM CAST(fr.funded_at AS DATE)) AS month,
COUNT(DISTINCT f.id) AS cnt
FROM fund AS f
LEFT JOIN investment AS i ON f.id = i.fund_id
LEFT JOIN funding_round AS fr ON i.funding_round_id = fr.id
WHERE f.country_code = 'USA'
AND EXTRACT(YEAR FROM CAST(fr.funded_at AS DATE)) BETWEEN 2010 AND 2013
GROUP BY 1),

t2 AS
(SELECT EXTRACT(MONTH FROM CAST(acquired_at AS DATE)) AS month,
COUNT(acquired_company_id) AS cnt,
SUM(price_amount) AS total
FROM acquisition
WHERE EXTRACT(YEAR FROM CAST(acquired_at AS DATE)) BETWEEN 2010 AND 2013
GROUP BY 1)

SELECT t1.month, t1.cnt, t2.cnt, t2.total
FROM t1
LEFT JOIN t2 ON t1.month = t2.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
t1 AS
```

```sql
(SELECT country_code,
        AVG(funding_total) as year_2011
FROM company
WHERE EXTRACT(YEAR FROM CAST(founded_at AS date)) IN ('2011', '2012', '2013')
GROUP BY 1, EXTRACT(YEAR FROM CAST(founded_at AS date))
HAVING EXTRACT(YEAR FROM CAST(founded_at AS date)) = '2011'),

t2 AS
(SELECT country_code,
        AVG(funding_total) as year_2012
FROM company
WHERE EXTRACT(YEAR FROM CAST(founded_at AS date)) IN ('2011', '2012', '2013')
GROUP BY 1, EXTRACT(YEAR FROM CAST(founded_at AS date))
HAVING EXTRACT(YEAR FROM CAST(founded_at AS date)) = '2012'),

t3 AS
(SELECT country_code,
        AVG(funding_total) as year_2013
FROM company
WHERE EXTRACT(YEAR FROM CAST(founded_at AS date)) IN ('2011', '2012', '2013')
GROUP BY 1, EXTRACT(YEAR FROM CAST(founded_at AS date))
HAVING EXTRACT(YEAR FROM CAST(founded_at AS date)) = '2013')

SELECT t1.country_code, t1.year_2011, t2.year_2012, t3.year_2013
FROM t1
INNER JOIN t2 ON t1.country_code = t2.country_code
INNER JOIN t3 ON t1.country_code = t3.country_code
ORDER BY 2 DESC
```