**1  Cколько компаний закрылось.**

Подсказка У закрытых компаний отмечен статус 'closed'. Чтобы посчитать компании, используйте агрегирующую функцию.

SELECT COUNT(status)

FROM company

WHERE status = 'closed'

**2  Количество привлечённых средств для новостных компаний США. Используем данные из таблицы company. Отсортируем таблицу по убыванию значений в поле funding\_total .**

Подсказка Чем занимается компания, можно понять по значению в поле category\_code. Чтобы объединить условия, используйте логический оператор.

SELECT \* FROM fund

**3  Найдем общую сумму сделок по покупке одних компаний другими в долларах. Отберем сделки, которые осуществлялись только за наличные с 2011 по 2013 год включительно.**

Подсказка Всю информацию нужно взять из таблицы acquisition. Способ оплаты сделки указан в поле term\_code.

SELECT SUM(price\_amount)

FROM acquisition

WHERE term\_code = 'cash' AND EXTRACT(YEAR FROM acquired\_at) BETWEEN '2011' AND '2013'

**4  Отобразим имя, фамилию и названия аккаунтов людей в твиттере, у которых названия аккаунтов начинаются на 'Silver'.**

Подсказка Получите срез по шаблону.

SELECT first\_name,

last\_name,

twitter\_username

FROM people

WHERE twitter\_username LIKE 'Silver%'

**5  Выведем на экран всю информацию о людях, у которых названия аккаунтов в твиттере содержат подстроку 'money', а фамилия начинается на 'K'.**

Подсказка Получите срез по шаблону.

SELECT \*

FROM people

WHERE twitter\_username LIKE '%money%'

AND last\_name LIKE 'K%'

**6  Для каждой страны отобразим общую сумму привлечённых инвестиций, которые получили компании, зарегистрированные в этой стране. Страну, в которой зарегистрирована компания, можно определить по коду страны. Отсортируем данные по убыванию суммы.**

Подсказка Используйте агрегирующую функцию и группировку. При сортировке тоже можно обращаться к результату агрегирующей функции.

SELECT country\_code,

SUM(funding\_total)

FROM company

GROUP BY country\_code

ORDER BY SUM(funding\_total) DESC

**7  Составим 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. Подсказка Не забудьте, что срез до и после группировки получают разными операторами.

SELECT funded\_at,

MIN(raised\_amount),

MAX(raised\_amount)

FROM funding\_round

GROUP BY funded\_at

HAVING MIN(raised\_amount) != 0

AND MIN(raised\_amount) != MAX(raised\_amount)

**8  Создадим 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. Подсказка Создать новое поле с категориями можно оператором CASE.

SELECT \*,

CASE

WHEN invested\_companies >= 100 THEN 'high\_activity'

WHEN invested\_companies BETWEEN 20 AND 100 THEN 'middle\_activity'

WHEN invested\_companies < 20 THEN 'low\_activity'

END

FROM fund;

**9  Для каждой из категорий, назначим в предыдущем задании, посчитаем округлённое до ближайшего целого числа среднее количество инвестиционных раундов, в которых фонд принимал участие. Выведем на экран категории и среднее число инвестиционных раундов. Отсортируем таблицу по возрастанию среднего.**

Подсказка Сгруппируйте данные по категориям и вычислите среднее для каждой группы. Для округления используйте функцию ROUND.

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 mean

FROM fund

GROUP BY activity

ORDER BY mean

**10  Выгрузим таблицу с десятью самыми активными инвестирующими странами. Активность страны определим по среднему количеству компаний, в которые инвестируют фонды этой страны.**

Для каждой страны посчитайте минимальное, максимальное и среднее число компаний, в которые инвестировали фонды, основанные с 2010 по 2012 год включительно. Исключите из таблицы страны с фондами, у которых минимальное число компаний, получивших инвестиции, равно нулю. Отсортируйте таблицу по среднему количеству компаний от большего к меньшему. Для фильтрации диапазона по годам используйте оператор BETWEEN. Подсказка Используйте агрегирующие функции MIN, MAX и AVG.

SELECT country\_code,

MIN(invested\_companies) AS min\_count,

MAX(invested\_companies) AS max\_count,

AVG(invested\_companies) AS mean\_count

FROM fund

WHERE EXTRACT(YEAR FROM founded\_at) BETWEEN '2010' AND '2012'

GROUP BY country\_code

HAVING MIN(invested\_companies) != 0

ORDER BY mean\_count DESC

LIMIT (10)

**11  Отобразим имя и фамилию всех сотрудников стартапов. Добавим поле с названием учебного заведения, которое окончил сотрудник, если эта информация известна.**

Подсказка Необходимую информацию содержат несколько таблиц, нужно придумать способ присоединения.

SELECT first\_name,

last\_name,

instituition

FROM people p

LEFT JOIN education AS e ON p.id = e.person\_id

**12  Для каждой компании найдем количество учебных заведений, которые окончили её сотрудники. Выведем название компании и число уникальных названий учебных заведений. Составим топ-5 компаний по количеству университетов.**

Подсказка Необходимая информация содержится в трёх таблицах. Одного присоединения будет недостаточно.

SELECT c.name,

COUNT(DISTINCT(e.instituition))

FROM education e

LEFT JOIN people AS p ON e.person\_id = p.id

INNER JOIN company AS c ON p.company\_id = c.id

GROUP BY c.name

ORDER BY COUNT(DISTINCT(e.instituition)) DESC

LIMIT 5

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

Подсказка Можно использовать подзапрос или присоединение.

SELECT DISTINCT c.name

FROM company c

LEFT JOIN funding\_round fr ON c.id = fr.company\_id

WHERE c.status = 'closed'

AND fr.is\_first\_round = 1

AND fr.is\_last\_round = 1

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

Подсказка В этом задании нужно использовать несколько присоединений или подзапросов. Или комбинировать эти методы.

SELECT DISTINCT p.id

FROM people p

INNER JOIN company AS c

ON p.company\_id = c.id

WHERE c.name IN (SELECT DISTINCT c.name

FROM company c

LEFT JOIN funding\_round fr ON c.id = fr.company\_id

WHERE c.status = 'closed'

AND fr.is\_first\_round = 1

AND fr.is\_last\_round = 1)

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

Подсказка Дополните предыдущий запрос.

SELECT DISTINCT e.instituition,

p.id

FROM people p

INNER JOIN education e

ON p.id = e.person\_id

WHERE p.id IN (SELECT DISTINCT p.id

FROM people p

INNER JOIN company AS c

ON p.company\_id = c.id

WHERE c.name IN (SELECT DISTINCT c.name

FROM company c

LEFT JOIN funding\_round fr ON c.id = fr.company\_id

WHERE c.status = 'closed'

AND fr.is\_first\_round = 1

AND fr.is\_last\_round = 1))

GROUP BY p.id

**16  Посчитаем количество учебных заведений для каждого сотрудника из предыдущего задания.**

Подсказка К предыдущему запросу добавьте группировку и агрегирующую функцию.

SELECT DISTINCT COUNT(e.instituition),

p.id

FROM people p

INNER JOIN education e

ON p.id = e.person\_id

WHERE p.id IN (SELECT DISTINCT p.id

FROM people p

INNER JOIN company AS c

ON p.company\_id = c.id

WHERE c.name IN (SELECT DISTINCT c.name

FROM company c

LEFT JOIN funding\_round fr ON c.id = fr.company\_id

WHERE c.status = 'closed'

AND fr.is\_first\_round = 1

AND fr.is\_last\_round = 1))

**17  Дополним предыдущий запрос и выведите среднее число учебных заведений, которые окончили сотрудники разных компаний. Нужно вывести только одну запись, группировка здесь не понадобится.**

Подсказка Используйте подзапрос, чтоб найти среднее от данных, которые вы получили в результате предыдущего запроса.

SELECT AVG(c.count)

FROM (SELECT DISTINCT COUNT(e.instituition),

p.id

FROM people p

INNER JOIN education e

ON p.id = e.person\_id

WHERE p.id IN (SELECT DISTINCT p.id

FROM people p

INNER JOIN company AS c

ON p.company\_id = c.id

WHERE c.name IN (SELECT DISTINCT c.name

FROM company c

LEFT JOIN funding\_round fr ON c.id = fr.company\_id

WHERE c.status = 'closed'

AND fr.is\_first\_round = 1

AND fr.is\_last\_round = 1))

GROUP BY p.id) AS c

**18  Напишем похожий запрос: выведем среднее число учебных заведений, которые окончили сотрудники компании Facebook.**

Подсказка Этот запрос похож на предыдущий, нужно только изменить условия для среза.

SELECT AVG(c.count) Facebook

FROM (SELECT DISTINCT COUNT(e.instituition),

p.id

FROM people p

INNER JOIN education e

ON p.id = e.person\_id

WHERE p.id IN (SELECT DISTINCT p.id

FROM people p

INNER JOIN company AS c

ON p.company\_id = c.id

WHERE c.name IN (SELECT DISTINCT c.name

FROM company c

LEFT JOIN funding\_round fr ON c.id = fr.company\_id

WHERE c.name = 'Facebook'))

GROUP BY p.id) AS c

**19  Составим 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. Подсказка Начните с таблицы investment и присоедините к ней таблицы company и fund. Последней можете присоединить таблицу funding\_round, но это только рекомендация. Вы можете написать запрос по-своему.

WITH

i AS (SELECT \* FROM investment),

c AS (SELECT \* FROM company),

f AS (SELECT \* FROM fund),

fr AS (SELECT \* FROM funding\_round)

SELECT f.name AS name\_of\_fund, ---fund

c.name AS name\_of\_company, ---company

fr.raised\_amount AS amount ---funding\_round

FROM i

INNER JOIN company AS c ON i.company\_id = c.id

INNER JOIN fund AS f ON f.id = i.fund\_id

INNER JOIN funding\_round AS fr ON fr.id = i.funding\_round\_id

WHERE c.milestones > 6

AND EXTRACT(year FROM CAST(fr.funded\_at AS date)) BETWEEN '2012' AND '2013'

**ИЛИ 2Й ВАРИАНТ**

SELECT DISTINCT f.name AS name\_of\_fund, --fund

c.name AS name\_of\_company, --company

fr.raised\_amount AS amount --funding\_round

FROM fund AS f

JOIN investment AS i ON f.id = i.fund\_id

JOIN company AS c ON i.company\_id = c.id

JOIN funding\_round AS fr ON i.funding\_round\_id = fr.id

WHERE c.milestones > 6

AND EXTRACT(YEAR FROM fr.funded\_at) BETWEEN '2012' AND '2013'

**20  Выгрузим 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 алфавитном порядке. Ограничьте таблицу первыми десятью записями. Подсказка Самая важная таблица — acquisition. Эта таблица содержит информацию о покупках одних компаний другими, поэтому данные из неё пропасть не должны. Выбирайте способ присоединения внимательно.

SELECT buyer.name, --fund

a.price\_amount, --acquisition

purchase.name, --company

purchase.funding\_total, --company

ROUND(a.price\_amount/purchase.funding\_total) AS ratio

FROM acquisition a

LEFT JOIN company AS buyer ON buyer.id=a.acquiring\_company\_id

LEFT JOIN company AS purchase ON purchase.id=a.acquired\_company\_id

WHERE a.price\_amount > 0

AND purchase.funding\_total > 0

ORDER BY a.price\_amount DESC, purchase.name

LIMIT 10

**21  Выгрузим таблицу, в которую войдут названия компаний из категории social, получившие финансирование с 2010 по 2013 год. Выведем также номер месяца, в котором проходил раунд финансирования.**

Подсказка Вам понадобятся таблицы company и funding\_round.

WITH

fr AS (SELECT EXTRACT(MONTH FROM CAST(funded\_at AS timestamp)) AS month,

id

FROM funding\_round

WHERE EXTRACT(YEAR FROM CAST(funded\_at AS timestamp)) BETWEEN '2010' AND '2013'),

usf AS (SELECT name,

id

FROM fund

WHERE country\_code = 'USA'),

acq AS (SELECT EXTRACT(MONTH FROM CAST(acquired\_at AS timestamp)) AS acq\_month,

COUNT(acquired\_company\_id) AS acq\_comps,

SUM(price\_amount) acq\_amounts

FROM acquisition

WHERE EXTRACT(MONTH FROM CAST(acquired\_at AS timestamp)) IS NOT NULL

GROUP by acq\_month)

SELECT fr.month,

COUNT(DISTINCT(usf.name)) AS names,

COUNT(acq.acq\_comps) AS acquired\_companies,

SUM(acq.acq\_amounts) AS total\_amounts

FROM fr INNER JOIN investment AS inv ON inv.funding\_round\_id = fr.id INNER JOIN usf ON inv.fund\_id = usf.id FULL JOIN acq ON

acq.acq\_month=fr.month

GROUP BY fr.month;

**22  Отберем данные по месяцам с 2010 по 2013 год, когда проходили инвестиционные раунды. Сгруппируем 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. Подсказка Чтобы вычислить номер месяца проведения раундов, используйте значения в поле funded\_at в таблице funding\_round. Два последних поля — с количеством купленных компаний и с суммой сделок — можно получить, взяв данные из таблицы acquisition. Используйте временные таблицы — будет легче не запутаться в присоединении.

WITH

fr AS (SELECT EXTRACT(MONTH FROM funded\_at) AS fr\_month,

id

FROM funding\_round

WHERE EXTRACT(YEAR FROM funded\_at) BETWEEN 2010 AND 2013),

i AS (SELECT funding\_round\_id,

fund\_id

FROM investment),

f AS (SELECT id,

name

FROM fund

WHERE country\_code = 'USA'),

a AS (SELECT EXTRACT(MONTH FROM acquired\_at) AS a\_month,

COUNT(acquired\_company\_id) AS a\_company,

SUM(price\_amount) a\_amount

FROM acquisition

WHERE EXTRACT(YEAR FROM acquired\_at) BETWEEN 2010 AND 2013

GROUP BY a\_month)

SELECT fr\_month,

COUNT(DISTINCT(f.name)),

MAX( a.a\_company),

MAX(a.a\_amount)

FROM fr

INNER JOIN i ON fr.id = i.funding\_round\_id

INNER JOIN f ON i.fund\_id = f.id

INNER JOIN a ON fr.fr\_month = a.a\_month

GROUP BY fr\_month

**23  Составим сводную таблицу и выведем 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. Отсортируем таблицу по среднему значению инвестиций за 2011 год от большего к меньшему.**

Подсказка Вы уже решали похожую задачу на курсе. Для каждой страны посчитайте среднюю сумму инвестиций за 2011 год. Проделайте то же самое для 2012 и 2013 года. Объедините получившиеся таблицы по стране.

WITH

year\_2011 AS (SELECT country\_code, AVG(funding\_total) AS avg\_2011

FROM company

WHERE EXTRACT(YEAR FROM founded\_at) = 2011

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

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

GROUP BY country\_code)

SELECT year\_2011.country\_code,

year\_2011.avg\_2011,

year\_2012.avg\_2012,

year\_2013.avg\_2013

FROM year\_2011

INNER JOIN year\_2012 ON year\_2011.country\_code = year\_2012.country\_code

INNER JOIN year\_2013 ON year\_2012.country\_code = year\_2013.country\_code

ORDER BY year\_2011.avg\_2011 DESC