**Исследование данных об инвестиции венчурных фондов в компании-стартапы**

В проекте работа идёт 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.

**Задача**  
Проанализировать данные о фондах и инвестициях, произвести выгрузки данных и ответить на поставленные вопросы с помощью SQL.

**Задания:**

**Задание 1**

Посчитайте, сколько компаний закрылось.

'''  
SELECT COUNT(name)  
FROM company  
WHERE status ='closed'  
;  
'''

**Задание 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;  
'''

**Задание 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 timestamp)) BETWEEN 2011 AND 2013;  
'''

**Задание 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%';  
'''

**Задание 5**

Вывести 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%';  
'''

**Задание 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;  
'''

**Задание 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 CAST(funded\_at AS date),  
MIN(raised\_amount),  
MAX(raised\_amount)  
FROM funding\_round  
GROUP BY CAST(funded\_at AS date)  
HAVING MIN(raised\_amount) != 0  
AND MIN(raised\_amount) != MAX(raised\_amount);  
'''

**Задание 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;  
'''

**Задание 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))  
FROM fund  
GROUP BY activity  
ORDER BY ROUND(AVG(investment\_rounds));  
'''

**Задание 10**

Выгрузите таблицу с десятью самыми активными инвестирующими странами. Активность страны определите 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.

'''  
SELECT country\_code,  
AVG(invested\_companies),  
MIN(invested\_companies),  
MAX(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;  
'''

**Задание 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.instituition  
FROM people AS p  
LEFT JOIN education AS e ON e.person\_id=p.id;  
'''

**Задание 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  
JOIN people AS p ON c.id=p.company\_id  
JOIN education AS e ON p.id=e.person\_id  
GROUP BY c.name  
ORDER BY COUNT(DISTINCT e.instituition) DESC  
LIMIT 5;  
'''

**Задание 13**

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

'''  
SELECT DISTINCT name  
FROM company  
WHERE id IN (SELECT company\_id  
FROM funding\_round  
WHERE is\_first\_round = 1 AND is\_last\_round = 1)  
AND status = 'closed';  
'''

**Задание 14**

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

'''  
SELECT DISTINCT p.id  
FROM people AS p  
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');  
'''

**Задание 15**

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

'''  
SELECT DISTINCT p.id,  
e.instituition  
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');  
'''

**Задание 16**

Посчитайте количество учебных заведений для каждого сотрудника из предыдущего задания.

'''  
SELECT DISTINCT p.id,  
COUNT(e.instituition)  
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;  
'''

**Задание 17**

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

'''  
WITH  
res AS (SELECT p.id,  
COUNT(e.instituition) AS total\_instituition  
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)

SELECT AVG(res.total\_instituition)  
FROM res;  
'''

**Задание 18**

Напишите похожий запрос: выведите среднее число учебных заведений, которые окончили сотрудники компании Facebook.

'''  
WITH  
res AS (SELECT p.id,  
COUNT(e.instituition) AS total\_instituition  
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)

SELECT AVG(res.total\_instituition)  
FROM res;  
'''

**Задание 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.

'''  
WITH funding\_round\_filer AS  
(SELECT \*  
FROM funding\_round  
WHERE EXTRACT(YEAR FROM CAST(funded\_at AS date)) IN (2012, 2013))

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

JOIN funding\_round\_filer AS fr ON fr.id = i.funding\_round\_id  
WHERE c.milestones > 6;  
'''

**Задание 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 алфавитном порядке. Ограничьте таблицу первыми десятью записями.

'''  
SELECT c.name,  
a.price\_amount,  
c\_1.name,  
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  
LIMIT 10;  
'''

**Задание 21**

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

'''  
SELECT c.name,  
EXTRACT(MONTH FROM CAST(funded\_at AS date))  
FROM company AS c  
LEFT OUTER JOIN funding\_round AS f ON c.id = f.company\_id  
WHERE EXTRACT(YEAR FROM CAST(funded\_at AS date)) IN (2010, 2011, 2012, 2013)  
AND c.category\_code = 'social';  
'''

**Задание 21**

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 CAST(funded\_at AS date)) 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 CAST(funded\_at AS date)) BETWEEN 2010 AND 2013  
AND f.country\_code='USA'  
GROUP BY MONTH),

month\_acquired AS  
(SELECT EXTRACT(MONTH FROM CAST(acquired\_at AS date)) AS MONTH,  
COUNT(acquiring\_company\_id) AS count\_month\_acquired,  
SUM(price\_amount) AS sum\_price\_amount  
FROM acquisition  
WHERE EXTRACT(YEAR FROM CAST(acquired\_at AS date)) BETWEEN 2010 AND 2013  
GROUP BY MONTH)

SELECT month\_fund.MONTH,  
month\_fund.count\_of\_fund,  
month\_acquired.count\_month\_acquired,  
month\_acquired.sum\_price\_amount  
FROM month\_fund  
JOIN month\_acquired ON month\_fund.MONTH=month\_acquired.MONTH;  
'''

**Задание 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  
inv\_2011 AS (SELECT country\_code,  
AVG(funding\_total) AS year\_2011  
FROM company  
WHERE EXTRACT(YEAR FROM founded\_at) = 2011  
GROUP BY country\_code),  
inv\_2012 AS (SELECT country\_code,  
AVG(funding\_total) AS year\_2012  
FROM company  
WHERE EXTRACT(YEAR FROM founded\_at) = 2012  
GROUP BY country\_code),  
inv\_2013 AS (SELECT country\_code,  
AVG(funding\_total) AS year\_2013  
FROM company  
WHERE EXTRACT(YEAR FROM founded\_at) = 2013  
GROUP BY country\_code)

SELECT inv\_2011.country\_code,  
inv\_2011.year\_2011,  
inv\_2012.year\_2012,  
inv\_2013.year\_2013  
FROM inv\_2011  
JOIN inv\_2012 ON inv\_2011.country\_code=inv\_2012.country\_code  
JOIN inv\_2013 ON inv\_2013.country\_code=inv\_2012.country\_code  
ORDER BY year\_2011 DESC;  
'''