## **Запросы**

1. **Выбрать менеджера (работника, у которого в подчинении есть хотя бы один человек), имеющего наименьший стаж. Если таких менеджеров несколько, вывести всех.**
   1. Поля: ID менеджера, Hire\_date
   2. Сортировка: ID менеджера

WITH Managers AS (

SELECT e."EMPLOYEE\_ID" AS "Manager\_ID", e."HIRE\_DATE"

FROM public."EMPLOYEES" e

WHERE

EXISTS (

SELECT \*

FROM public."EMPLOYEES" ee

WHERE ee."MANAGER\_ID" = e."EMPLOYEE\_ID"

)

),

MinTenure AS (

SELECT MAX("HIRE\_DATE") AS "Latest\_Hire\_Date"

FROM Managers

)

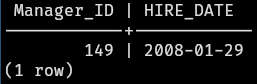
SELECT m."Manager\_ID", m."HIRE\_DATE"

FROM Managers m

JOIN MinTenure t

ON m."HIRE\_DATE" = t."Latest\_Hire\_Date"

ORDER BY m."Manager\_ID";



1. **Выбрать сотрудников, имеющих оклад выше среднего оклада подразделения, где они работают. Исключить из выборки сотрудников, которые не приписаны ни к какому подразделению.**
   1. Поля: идентификатор сотрудника, фамилия сотрудника, оклад, идентификатор подразделения, к которому приписан сотрудник
   2. Сортировка: идентификатор подразделения, оклад, фамилия сотрудника

SELECT e."EMPLOYEE\_ID", e."LAST\_NAME", e."SALARY", e."DEPARTMENT\_ID"

FROM public."EMPLOYEES" e

JOIN (

SELECT "DEPARTMENT\_ID", AVG("SALARY") AS avg\_salary

FROM public."EMPLOYEES"

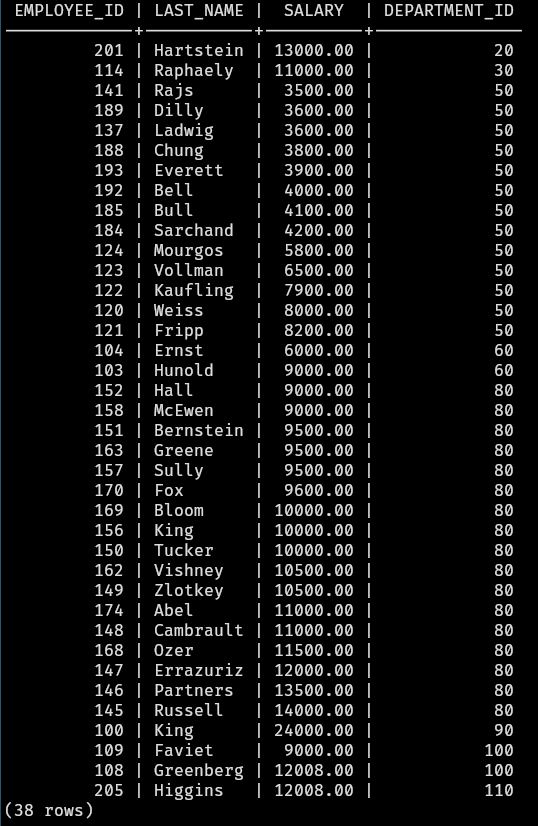
WHERE "DEPARTMENT\_ID" IS NOT NULL

GROUP BY "DEPARTMENT\_ID"

) department\_avg ON e."DEPARTMENT\_ID" = department\_avg."DEPARTMENT\_ID"

WHERE e."SALARY" > department\_avg.avg\_salary

ORDER BY e."DEPARTMENT\_ID", e."SALARY", e."LAST\_NAME";



1. **Выбрать сотрудников, имеющих оклад равный минимальному окладу подразделения, где они работают. Исключить из выборки сотрудников, которые не приписаны ни к какому подразделению.**
   1. Поля: идентификатор сотрудника, фамилию сотрудника, оклад, установленный сотруднику, идентификатор подразделения, к которому приписан сотрудник
   2. Сортировка: идентификатор подразделения, оклад, фамилия сотрудника

SELECT e."EMPLOYEE\_ID", e."LAST\_NAME", e."SALARY", e."DEPARTMENT\_ID"

FROM public."EMPLOYEES" e

JOIN (

SELECT "DEPARTMENT\_ID", MIN("SALARY") AS min\_salary

FROM public."EMPLOYEES"

WHERE "DEPARTMENT\_ID" IS NOT NULL

GROUP BY "DEPARTMENT\_ID"

) department\_min ON e."DEPARTMENT\_ID" = department\_min."DEPARTMENT\_ID" AND e."SALARY" = department\_min.min\_salary

ORDER BY e."DEPARTMENT\_ID", e."SALARY", e."LAST\_NAME";



1. **Выбрать среди работников Америки (region\_name = "Americas") тех, чья зарплата превосходит среднюю зарплату менеджеров из Европы (region\_name = "Europe") с наибольшим количеством подчиненных.**
   1. Поля: Фамилия сотрудника, Имя сотрудника, зарплата
   2. Сортировка: Фамилия сотрудника, Имя сотрудника

SELECT e."LAST\_NAME", e."FIRST\_NAME", e."SALARY"

FROM public."EMPLOYEES" e

WHERE e."DEPARTMENT\_ID" IN (

SELECT "DEPARTMENT\_ID"

FROM public."DEPARTMENTS" d

JOIN public."LOCATIONS" l ON d."LOCATION\_ID" = l."LOCATION\_ID"

JOIN public."COUNTRIES" c ON l."COUNTRY\_ID" = c."COUNTRY\_ID"

JOIN public."REGIONS" r ON c."REGION\_ID" = r."REGION\_ID"

WHERE r."REGION\_NAME" = 'Americas'

)

AND e."SALARY" > (

SELECT AVG(m."SALARY")

FROM public."EMPLOYEES" m

WHERE m."EMPLOYEE\_ID" IN (

SELECT d."MANAGER\_ID"

FROM public."DEPARTMENTS" d

JOIN public."LOCATIONS" l ON d."LOCATION\_ID" = l."LOCATION\_ID"

JOIN public."COUNTRIES" c ON l."COUNTRY\_ID" = c."COUNTRY\_ID"

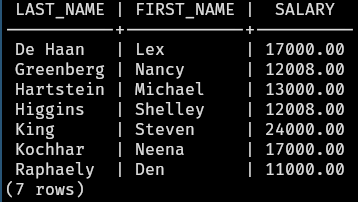
JOIN public."REGIONS" r ON c."REGION\_ID" = r."REGION\_ID"

WHERE r."REGION\_NAME" = 'Europe'

)

)

ORDER BY e."LAST\_NAME", e."FIRST\_NAME";



1. **Сформировать список подразделений с информацией о сумме окладов всех сотрудников и проценте, который эта сумма составляет от суммы окладов всех сотрудников компании. Если в подразделении нет сотрудников, то считать, что сумма их окладов равна нулю. В отчете отдельной строкой учесть сведения о сотрудниках, которые не приписаны ни к какому подразделению компании (null в поле подразделения). При вычислении процентов должно быть использовано округление результата до 2-х знаков после запятой.**
   1. Поля: идентификатор подразделения компании, сумма окладов сотрудников подразделения компании, процент
   2. Сортировка: процент по убыванию

SELECT d."DEPARTMENT\_ID", COALESCE(SUM(e."SALARY"), 0) AS total\_salary,

ROUND(COALESCE(SUM(e."SALARY"), 0) \* 100.0 / (SELECT SUM("SALARY") FROM public."EMPLOYEES"), 2) AS percentage

FROM public."DEPARTMENTS" d

LEFT JOIN public."EMPLOYEES" e ON d."DEPARTMENT\_ID" = e."DEPARTMENT\_ID"

GROUP BY d."DEPARTMENT\_ID"

UNION ALL

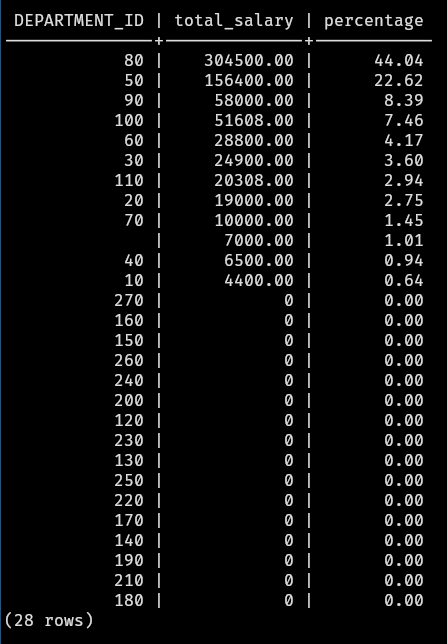
SELECT NULL, COALESCE(SUM("SALARY"), 0),

ROUND(COALESCE(SUM("SALARY"), 0) \* 100.0 / (SELECT SUM("SALARY") FROM public."EMPLOYEES"), 2)

FROM public."EMPLOYEES"

WHERE "DEPARTMENT\_ID" IS NULL

ORDER BY percentage DESC;



1. **Сформировать список подразделений с информацией о сумме среднего оклада сотрудников по подразделениям компании и проценте, который эта сумма составляет от средней величины окладов всех сотрудников компании. Если в подразделении нет сотрудников, то считать, что сумма их окладов равна нулю. В отчете отдельной строкой учесть сведения о сотрудниках, которые не приписаны ни к какому подразделению компании**
   1. Поля: идентификатор подразделения компании, средняя зарплата по подразделению, процент
   2. Сортировка: идентификатор подразделения компании, средняя зарплата по подразделению, процент по убыванию

WITH

"department\_salaries" AS (

-- Вычисляем среднюю зарплату по каждому подразделению

SELECT

d."DEPARTMENT\_ID",

COALESCE(AVG(e."SALARY"), 0) AS "avg\_salary"

FROM

public."DEPARTMENTS" d

LEFT JOIN public."EMPLOYEES" e ON d."DEPARTMENT\_ID" = e."DEPARTMENT\_ID"

GROUP BY

d."DEPARTMENT\_ID"

),

"overall\_average" AS (

-- Вычисляем среднюю зарплату по всем сотрудникам

SELECT

AVG("SALARY") AS "overall\_avg\_salary"

FROM

public."EMPLOYEES"

)

SELECT

ds."DEPARTMENT\_ID",

ds."avg\_salary",

(

ds."avg\_salary" / NULLIF(overall\_avg."overall\_avg\_salary", 0)

) \* 100 AS "percentage"

FROM

"department\_salaries" ds,

"overall\_average" overall\_avg

UNION ALL

-- Учитываем сотрудников, которые не приписаны ни к какому подразделению

SELECT

NULL AS "DEPARTMENT\_ID",

0 AS "avg\_salary",

-- Сумма зарплат для сотрудников без подразделений равна 0

0 AS "percentage" -- Процент тоже равен 0

FROM

"overall\_average" overall\_avg

WHERE

(

SELECT

COUNT(\*)

FROM

public."EMPLOYEES" e

WHERE

e."DEPARTMENT\_ID" IS NULL

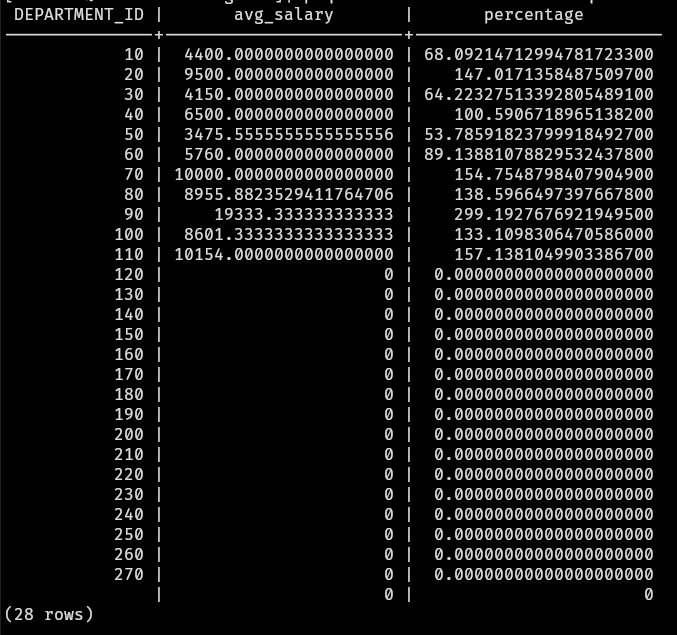
) > 0

ORDER BY

"DEPARTMENT\_ID" ASC,

"avg\_salary" DESC,

"percentage" DESC;



1. **Вывести имена сотрудников и количество работников с одним именем**
   1. Поля: Имя, Количество работников
   2. Сортировка: Количество работников, Имя

SELECT

"FIRST\_NAME" AS Имя,

COUNT(\*) AS Количество\_работников

FROM

public."EMPLOYEES"

GROUP BY

"FIRST\_NAME"

ORDER BY

Количество\_работников DESC,

Имя ASC;





1. **Вывести для каждого отдела из Америки (region\_name = "Americas") количество работников в иерархии до третьего уровня. Первый уровень - работники без руководителя, второй уровень - это их подчиненные, а третий уровень - подчиненные работников второго уровня.**
   1. Поля: Название отдела, количество работников I уровня, количество работников II уровня, количество работников III уровня
   2. Сортировка: Название отдела

WITH RECURSIVE

EmployeeHierarchy AS (

SELECT

e."EMPLOYEE\_ID",

e."DEPARTMENT\_ID",

1 AS level

FROM

public."EMPLOYEES" e

WHERE

e."MANAGER\_ID" IS NULL

UNION ALL

SELECT

e."EMPLOYEE\_ID",

e."DEPARTMENT\_ID",

eh.level + 1 AS level

FROM

public."EMPLOYEES" e

JOIN EmployeeHierarchy eh ON e."MANAGER\_ID" = eh."EMPLOYEE\_ID"

)

SELECT

d."DEPARTMENT\_NAME" AS Название\_отдела,

COUNT(

CASE

WHEN eh.level = 1 THEN 1

END

) AS количество\_работников\_I\_уровня,

COUNT(

CASE

WHEN eh.level = 2 THEN 1

END

) AS количество\_работников\_II\_уровня,

COUNT(

CASE

WHEN eh.level = 3 THEN 1

END

) AS количество\_работников\_III\_уровня

FROM

public."DEPARTMENTS" d

LEFT JOIN EmployeeHierarchy eh ON d."DEPARTMENT\_ID" = eh."DEPARTMENT\_ID"

JOIN public."LOCATIONS" l ON d."LOCATION\_ID" = l."LOCATION\_ID"

JOIN public."COUNTRIES" c ON l."COUNTRY\_ID" = c."COUNTRY\_ID"

JOIN public."REGIONS" r ON c."REGION\_ID" = r."REGION\_ID"

WHERE

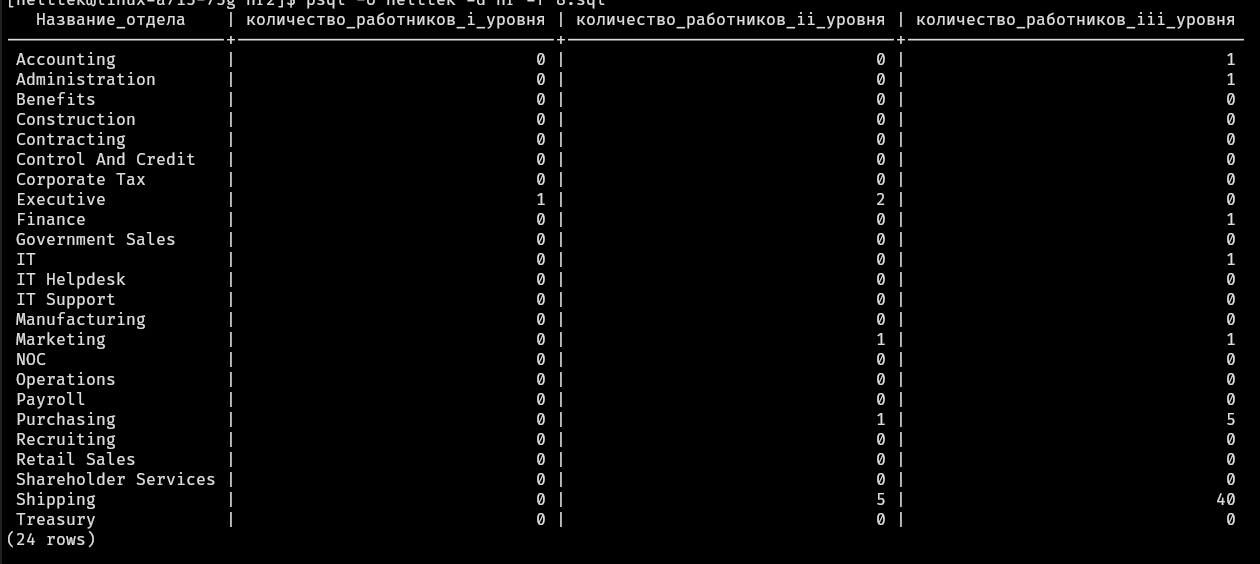
r."REGION\_NAME" = 'Americas'

GROUP BY

d."DEPARTMENT\_NAME"

ORDER BY

d."DEPARTMENT\_NAME";



1. **Вывести названия всех отделов, в которых наименьшая зарплата выше средней зарплаты в Америке (region\_name = "Americas")**
   1. Поля: Название отдела
   2. Сортировка: Название отдела

WITH

AverageSalary AS (

SELECT

AVG(e."SALARY") AS avg\_salary

FROM

public."EMPLOYEES" e

JOIN public."DEPARTMENTS" d ON e."DEPARTMENT\_ID" = d."DEPARTMENT\_ID"

JOIN public."LOCATIONS" l ON d."LOCATION\_ID" = l."LOCATION\_ID"

JOIN public."COUNTRIES" c ON l."COUNTRY\_ID" = c."COUNTRY\_ID"

JOIN public."REGIONS" r ON c."REGION\_ID" = r."REGION\_ID"

WHERE

r."REGION\_NAME" = 'Americas'

),

DepartmentMinSalaries AS (

SELECT

d."DEPARTMENT\_NAME",

MIN(e."SALARY") AS min\_salary

FROM

public."DEPARTMENTS" d

JOIN public."EMPLOYEES" e ON d."DEPARTMENT\_ID" = e."DEPARTMENT\_ID"

JOIN public."LOCATIONS" l ON d."LOCATION\_ID" = l."LOCATION\_ID"

JOIN public."COUNTRIES" c ON l."COUNTRY\_ID" = c."COUNTRY\_ID"

JOIN public."REGIONS" r ON c."REGION\_ID" = r."REGION\_ID"

GROUP BY

d."DEPARTMENT\_NAME"

)

SELECT

dms."DEPARTMENT\_NAME"

FROM

DepartmentMinSalaries dms

WHERE

dms.min\_salary > (

SELECT

avg\_salary

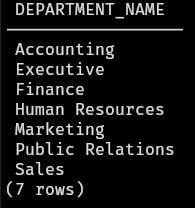
FROM

AverageSalary

)

ORDER BY

dms."DEPARTMENT\_NAME";



1. **Вывести общую зарплату работников, работающих в отделе с наименьшим количеством сотрудников**
   1. Поля: Номер отдела, общая сумма зарплаты
   2. Сортировка: Номер отдела

WITH

DepartmentEmployeeCount AS (

SELECT

"DEPARTMENT\_ID",

COUNT(\*) AS employee\_count

FROM

public."EMPLOYEES"

GROUP BY

"DEPARTMENT\_ID"

),

MinEmployeeDepartment AS (

SELECT

"DEPARTMENT\_ID"

FROM

DepartmentEmployeeCount

WHERE

employee\_count = (

SELECT

MIN(employee\_count)

FROM

DepartmentEmployeeCount

)

)

SELECT

e."DEPARTMENT\_ID" AS Номер\_отдела,

SUM(e."SALARY") AS Общая\_сумма\_зарплаты

FROM

public."EMPLOYEES" e

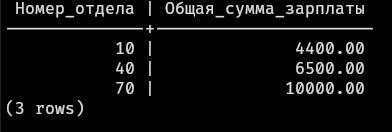
JOIN MinEmployeeDepartment d ON e."DEPARTMENT\_ID" = d."DEPARTMENT\_ID"

GROUP BY

e."DEPARTMENT\_ID"

ORDER BY

e."DEPARTMENT\_ID";



1. **Вывести однофамильцев. Если встречается более двух - вывести их количество.**
   1. Поля: Фамилия, Количество работников
   2. Сортировка: Фамилия

SELECT

e."LAST\_NAME",

COUNT(e."EMPLOYEE\_ID") AS employee\_count

FROM

public."EMPLOYEES" e

GROUP BY

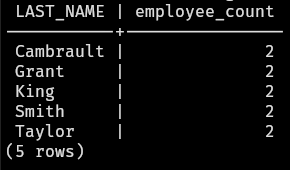
e."LAST\_NAME"

HAVING

COUNT(e."EMPLOYEE\_ID") > 1

ORDER BY

e."LAST\_NAME";



1. **В стране с наибольшим количеством подразделений (департаментов) выбрать работника с максимальной зарплатой. Если таких стран несколько, то найти такого работника для каждой из них (то же касается нескольких работников).**
   1. Поля: ID страны, ID работника
   2. Сортировка: ID страны, ID работника

WITH

CountryDepartmentCounts AS (

-- Считаем количество подразделений в каждой стране

SELECT

l."COUNTRY\_ID",

COUNT(d."DEPARTMENT\_ID") AS department\_count

FROM

public."LOCATIONS" l

LEFT JOIN public."DEPARTMENTS" d ON l."LOCATION\_ID" = d."LOCATION\_ID"

GROUP BY

l."COUNTRY\_ID"

),

MaxDepartments AS (

-- Находим страны с максимальным количеством подразделений

SELECT

"COUNTRY\_ID"

FROM

CountryDepartmentCounts

WHERE

department\_count = (

SELECT

MAX(department\_count)

FROM

CountryDepartmentCounts

)

),

EmployeesInTopCountries AS (

-- Находим всех сотрудников из стран с максимальным количеством подразделений

SELECT

e."EMPLOYEE\_ID",

e."SALARY",

l."COUNTRY\_ID"

FROM

public."EMPLOYEES" e

JOIN public."DEPARTMENTS" d ON e."DEPARTMENT\_ID" = d."DEPARTMENT\_ID"

JOIN public."LOCATIONS" l ON d."LOCATION\_ID" = l."LOCATION\_ID"

WHERE

l."COUNTRY\_ID" IN (

SELECT

"COUNTRY\_ID"

FROM

MaxDepartments

)

),

MaxSalaryEmployees AS (

-- Находим максимальную зарплату в каждой из стран

SELECT

"COUNTRY\_ID",

MAX("SALARY") AS max\_salary

FROM

EmployeesInTopCountries

GROUP BY

"COUNTRY\_ID"

)

-- Итоговый запрос: выбираем работников с максимальной зарплатой в странах с наибольшим количеством подразделений

SELECT

e."COUNTRY\_ID",

e."EMPLOYEE\_ID"

FROM

EmployeesInTopCountries e

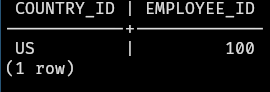
JOIN MaxSalaryEmployees m ON e."COUNTRY\_ID" = m."COUNTRY\_ID"

AND e."SALARY" = m.max\_salary

ORDER BY

e."COUNTRY\_ID",

e."EMPLOYEE\_ID";



1. **В стране с наибольшим количеством подразделений (департаментов) выбрать работника, у которого в подчинении больше всего человек. Если таких стран несколько, то найти такого работника для каждой из них.**
   1. Поля: ID работника; количество подчиненных сотрудников
   2. Сортировка: ID работника

WITH

CountryDepartmentCounts AS (

SELECT

l."COUNTRY\_ID",

COUNT(d."DEPARTMENT\_ID") AS department\_count

FROM

public."LOCATIONS" l

LEFT JOIN public."DEPARTMENTS" d ON l."LOCATION\_ID" = d."LOCATION\_ID"

GROUP BY

l."COUNTRY\_ID"

),

MaxDepartments AS (

SELECT

"COUNTRY\_ID"

FROM

CountryDepartmentCounts

WHERE

department\_count = (

SELECT

MAX(department\_count)

FROM

CountryDepartmentCounts

)

),

EmployeeSubordinateCounts AS (

SELECT

m."EMPLOYEE\_ID" AS manager\_id,

COUNT(e."EMPLOYEE\_ID") AS subordinate\_count

FROM

public."EMPLOYEES" e

JOIN public."EMPLOYEES" m ON e."MANAGER\_ID" = m."EMPLOYEE\_ID"

GROUP BY

m."EMPLOYEE\_ID"

),

EmployeesInTopCountries AS (

SELECT

e."EMPLOYEE\_ID",

l."COUNTRY\_ID",

COALESCE(sc.subordinate\_count, 0) AS subordinate\_count

FROM

public."EMPLOYEES" e

JOIN public."DEPARTMENTS" d ON e."DEPARTMENT\_ID" = d."DEPARTMENT\_ID"

JOIN public."LOCATIONS" l ON d."LOCATION\_ID" = l."LOCATION\_ID"

LEFT JOIN EmployeeSubordinateCounts sc ON e."EMPLOYEE\_ID" = sc.manager\_id

WHERE

l."COUNTRY\_ID" IN (

SELECT

"COUNTRY\_ID"

FROM

MaxDepartments

)

),

MaxSubordinateEmployees AS (

SELECT

"COUNTRY\_ID",

MAX(subordinate\_count) AS max\_subordinate\_count

FROM

EmployeesInTopCountries

GROUP BY

"COUNTRY\_ID"

)

SELECT

e."EMPLOYEE\_ID" AS ID\_работника,

e.subordinate\_count AS Количество\_подчиненных

FROM

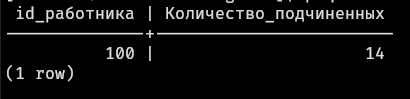
EmployeesInTopCountries e

JOIN MaxSubordinateEmployees m ON e."COUNTRY\_ID" = m."COUNTRY\_ID"

AND e.subordinate\_count = m.max\_subordinate\_count

ORDER BY

e."EMPLOYEE\_ID";



1. **В стране, в которой проживает сотрудник с наименьшим стажем (если таких сотрудников несколько, то рассмотреть страну для каждого из них), выбрать работника, в подчинении которого больше всего человек.**
   1. Поля: ID работника, количество подчиненных сотрудников
   2. Сортировка: ID работника

WITH

EmployeeSeniority AS (

SELECT

e."EMPLOYEE\_ID",

l."COUNTRY\_ID",

e."HIRE\_DATE"

FROM

public."EMPLOYEES" e

JOIN public."DEPARTMENTS" d ON e."DEPARTMENT\_ID" = d."DEPARTMENT\_ID"

JOIN public."LOCATIONS" l ON d."LOCATION\_ID" = l."LOCATION\_ID"

WHERE

e."HIRE\_DATE" = (

SELECT

MAX("HIRE\_DATE")

FROM

public."EMPLOYEES"

)

),

EmployeesInTargetCountries AS (

SELECT

e."EMPLOYEE\_ID",

e."MANAGER\_ID",

l."COUNTRY\_ID"

FROM

public."EMPLOYEES" e

JOIN public."DEPARTMENTS" d ON e."DEPARTMENT\_ID" = d."DEPARTMENT\_ID"

JOIN public."LOCATIONS" l ON d."LOCATION\_ID" = l."LOCATION\_ID"

WHERE

l."COUNTRY\_ID" IN (

SELECT DISTINCT

"COUNTRY\_ID"

FROM

EmployeeSeniority

)

),

SubordinateCounts AS (

SELECT

e."MANAGER\_ID" AS EMPLOYEE\_ID,

COUNT(e."EMPLOYEE\_ID") AS subordinate\_count

FROM

EmployeesInTargetCountries e

WHERE

e."MANAGER\_ID" IS NOT NULL

GROUP BY

e."MANAGER\_ID"

),

MaxSubordinateCounts AS (

SELECT

t."COUNTRY\_ID",

MAX(sc.subordinate\_count) AS max\_subordinate\_count

FROM

SubordinateCounts sc

JOIN EmployeesInTargetCountries t ON sc.EMPLOYEE\_ID = t."EMPLOYEE\_ID"

GROUP BY

t."COUNTRY\_ID"

)

SELECT

sc.EMPLOYEE\_ID AS ID\_работника,

sc.subordinate\_count AS Количество\_подчиненных

FROM

SubordinateCounts sc

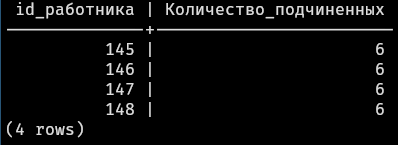
JOIN EmployeesInTargetCountries t ON sc.EMPLOYEE\_ID = t."EMPLOYEE\_ID"

JOIN MaxSubordinateCounts msc ON t."COUNTRY\_ID" = msc."COUNTRY\_ID"

AND sc.subordinate\_count = msc.max\_subordinate\_count

ORDER BY

sc.EMPLOYEE\_ID;



1. **В стране, в которой проживает сотрудник с наибольшим стажем (если таких несколько, то рассмотреть страну для каждого из них), выбрать работника, с наибольшей зарплатой его подчиненных.**
   1. Поля: ID работника; Сумма зарплат подчиненных
   2. Сортировка: ID работника

WITH

EmployeeSeniority AS (

SELECT

e."EMPLOYEE\_ID",

l."COUNTRY\_ID",

e."HIRE\_DATE"

FROM

public."EMPLOYEES" e

JOIN public."DEPARTMENTS" d ON e."DEPARTMENT\_ID" = d."DEPARTMENT\_ID"

JOIN public."LOCATIONS" l ON d."LOCATION\_ID" = l."LOCATION\_ID"

WHERE

e."HIRE\_DATE" = (

SELECT

MIN("HIRE\_DATE")

FROM

public."EMPLOYEES"

)

),

EmployeesInTargetCountries AS (

SELECT

e."EMPLOYEE\_ID",

e."MANAGER\_ID",

l."COUNTRY\_ID",

e."SALARY"

FROM

public."EMPLOYEES" e

JOIN public."DEPARTMENTS" d ON e."DEPARTMENT\_ID" = d."DEPARTMENT\_ID"

JOIN public."LOCATIONS" l ON d."LOCATION\_ID" = l."LOCATION\_ID"

WHERE

l."COUNTRY\_ID" IN (

SELECT DISTINCT

"COUNTRY\_ID"

FROM

EmployeeSeniority

)

),

ManagerSalarySums AS (

SELECT

e."MANAGER\_ID" AS EMPLOYEE\_ID,

SUM(e."SALARY") AS total\_subordinate\_salary

FROM

EmployeesInTargetCountries e

WHERE

e."MANAGER\_ID" IS NOT NULL

GROUP BY

e."MANAGER\_ID"

),

MaxSalaryByCountry AS (

SELECT

t."COUNTRY\_ID",

MAX(ms.total\_subordinate\_salary) AS max\_subordinate\_salary

FROM

ManagerSalarySums ms

JOIN EmployeesInTargetCountries t ON ms.EMPLOYEE\_ID = t."EMPLOYEE\_ID"

GROUP BY

t."COUNTRY\_ID"

)

SELECT

ms.EMPLOYEE\_ID AS ID\_работника,

ms.total\_subordinate\_salary AS Сумма\_зарплат\_подчиненных

FROM

ManagerSalarySums ms

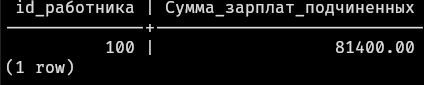
JOIN EmployeesInTargetCountries t ON ms.EMPLOYEE\_ID = t."EMPLOYEE\_ID"

JOIN MaxSalaryByCountry msc ON t."COUNTRY\_ID" = msc."COUNTRY\_ID"

AND ms.total\_subordinate\_salary = msc.max\_subordinate\_salary

ORDER BY

ms.EMPLOYEE\_ID;



1. **Определить год, в котором было трудоустроено больше всего человек.**
   1. Поля: Год; Кол-во трудоустроенных
   2. Сортировка: Год

SELECT EXTRACT(YEAR FROM "HIRE\_DATE") AS "Year", COUNT(\*) AS "Employee\_Count"

FROM public."EMPLOYEES"

GROUP BY EXTRACT(YEAR FROM "HIRE\_DATE")

ORDER BY "Employee\_Count" DESC LIMIT 1;

