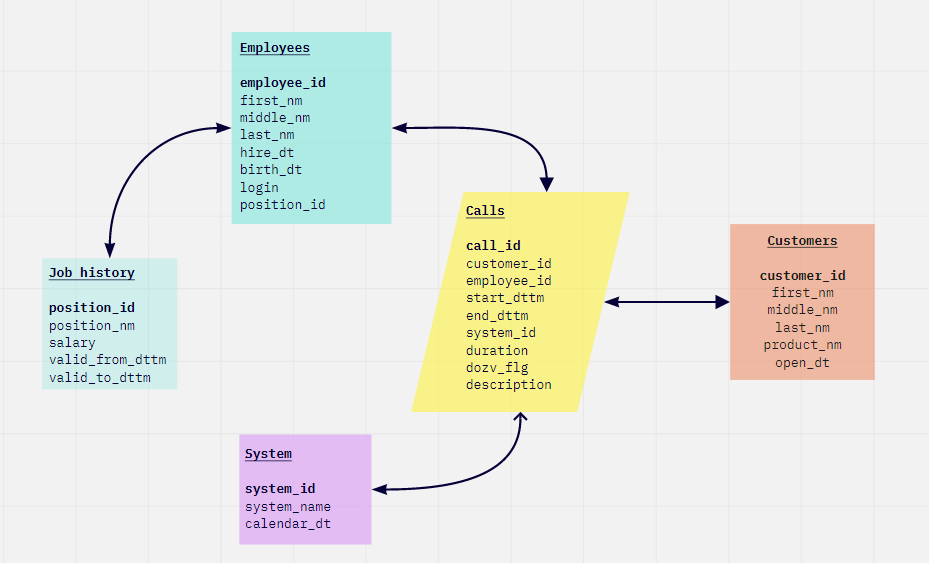
**Задания SQL - 1 lvl**

1. Дана следующая схема таблиц:



**Задание**:

* 1. Получить список всех сотрудников (*таблица Employees*), отсортированную по алфавиту

*Вывести: все поля из таблицы employees;*

SELECT \* FROM Employees ORDER BY last\_nm ASC

* 1. Получить список сотрудников с названиями их позиций (*position\_nm*), актуальных на 10.10.2020

*Вывести: last\_nm, first\_nm, middle\_nm, position\_nm*

SELECT Employees.last\_nm,Employees.first\_nm,Employees.middle\_nm,Job\_history.position\_nm FROM Employees

INNER JOIN Job\_history ON Employees.position\_id=Job\_history.position\_id

WHERE CAST(Job\_history.valid\_to\_dttm AS Date) > '2020-10-10'

* 1. Получить список сотрудников, у которых ЗП (salary) от 20 000 до 50 000, актуальных на текущий момент

*Вывести: last\_nm, first\_nm, middle\_nm, salary, hire\_dt*

SELECT last\_nm, first\_nm, middle\_nm, hire\_dt FROM Employees

INNER JOIN Job\_history ON Employees.position\_id=Job\_history.position\_id

WHERE (Job\_history.salary BETWEEN 20000 AND 50000) AND Job\_history.valid\_to\_dttm > GETDATE()

* 1. Получить список сотрудников в формате: «*Иванова - Наталья – Юрьевна»* (ФИО должно быть прописано в одном столбике, разделение ‘-’)

*Вывести: (новое* поле, назовем его fio), birth\_dt

SELECT CONCAT(last\_nm,'-',first\_nm,'-',middle\_nm) AS fio, birth\_dt FROM Employees

* 1. Получить количество звонков (*таблица Calls*) по дням, период с 01.10.2020 по текущий день

*Вывести: date (дата звонка), cnt\_calls (количество звонков)*

SELECT start\_dttm as call\_date, COUNT(\*) AS cnt\_calls FROM Calls

WHERE CAST(start\_dttm AS Date) BETWEEN '2020-10-01' AND CAST(GETDATE() AS Date)

GROUP BY start\_dttm

* 1. Вывести %% дозвона для каждого дня *(%%дозвона – это доля принятых звонков (dozv\_flg = 1) от всех поступивших звонков (dozv\_flg =1 ordozv\_flg = 0*)), период с 01.10.2020 по текущий день

*Вывести: date, sla (%% дозвона)*

SELECT CAST(start\_dttm AS Date) as date\_,(SELECT COUNT(\*) FROM Calls WHERE dozv\_flg = 1/(SELECT COUNT(\*) From Calls)) as sla FROM Calls

WHERE CAST(start\_dttm AS Date) BETWEEN '2020-10-01' AND CAST(GETDATE() AS Date)

GROUP BY CAST(start\_dttm AS Date)

* 1. Получить длительность самого длинного разговора, и самого короткого за 28.10.2020 (start\_dttm – начало звонка, end\_dttm - окончание звонка)

*Вывести: max\_call\_time, min\_call\_time*

SELECT MAX(duration) as max\_call\_time, MIN(duration) as min\_call\_time FROM Calls

WHERE start\_dttm = '2020-10-28'

* 1. Получить информация по всем звонкам, в какой системе они были обработаны (*таблица System*)

*Вывести: call\_id, start\_dttm, end\_dttm, system\_name*

SELECT Calls.call\_id,Calls.start\_dttm,Calls.end\_dttm,System\_.system\_name FROM Calls

INNER JOIN System\_ ON Calls.system\_id=System\_.system\_id

* 1. Получить информацию по клиентам(*таблица Customers*), которые звонили и не дозвонились 23.10.2020.

*Вывести: date (датазвонка), last\_nm, first\_nm, middle\_nm*

SELECT last\_nm,first\_nm,middle\_nm FROM Customers

INNER JOIN Calls ON Customers.customer\_id=Calls.customer\_id

WHERE Calls.dozv\_flg = 0 AND Calls.start\_dttm = '2020-10-23'

* 1. Получить информацию по продуктам, которые были открыты у клиентов во время звонка с 05.10.2020 по текущий день

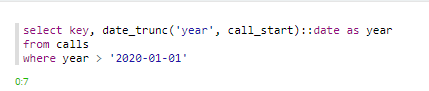
*Вывести*: *last\_nm, first\_nm, middle\_nm, date (датазвонка), open\_dt, product\_nm*

SELECT Customers.last\_nm,Customers.first\_nm,Customers.middle\_nm, Calls.start\_dttm, Customers.product\_nm, Customers.open\_dt FROM Customers

INNER JOIN Calls ON Customers.customer\_id=Calls.customer\_id

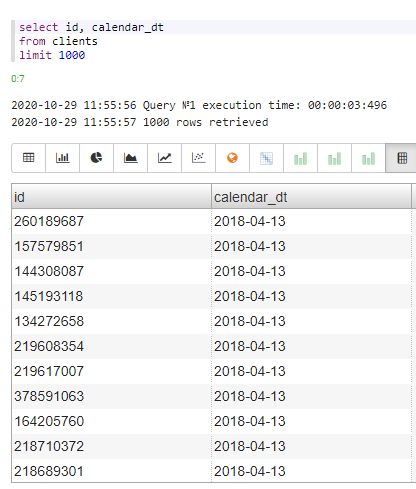
WHERE Calls.start\_dttm BETWEEN '2020-10-05' AND CAST(GETDATE() AS Date)

1. Что не так с этим запросом?



В первой строке от даты звонка (столбец call\_start) отделяется год, который записывается в новый столбец year. В третьей строке происходит попытка задать условие по столбцу, который только будет создан в результате этого запроса. Возможно, правильнее бы было определить условие для call\_start, а не year.

1. Есть таблица вида: idклиента, дата входа в МП.



* 1. Написать SQL запрос для расчетаDAU/WAU/MAU.

SELECT calendar\_dt, COUNT(DISTINCT id) as dau,

(SELECT COUNT(DISTINCT id)

FROM Clients WHERE calendar\_dt BETWEEN DATEADD(day,-7,calendar\_dt) AND calendar\_dt) AS wau,

(SELECT COUNT(DISTINCT id)

FROM Clients WHERE calendar\_dt BETWEEN DATEADD(day,-29,calendar\_dt) AND calendar\_dt) AS mau

FROM Clients

GROUP BY calendar\_dt

* 1. Написать SQL запрос для расчета CustomerRetentionRate (ежемесячного)

WITH t(id,month\_,actions,repeat\_action) AS (SELECT id,

DATEPART(month, calendar\_dt) AS month\_,

COUNT(\*) AS actions,

lag(DATEPART(month, calendar\_dt),1,0) OVER (PARTITION BY id ORDER BY DATEPART(month, calendar\_dt)) AS repeat\_action

FROM Customers

GROUP BY id, calendar\_dt)

SELECT month\_,

SUM(actions) AS num\_act,

COUNT (\*) AS num\_customers,

COUNT (repeat\_action) AS repeat\_customers,

ROUND(CASE WHEN SUM(actions) > 0 THEN COUNT(repeat\_action) / SUM(actions) \* 100 ELSE 0 END, 2) AS customer\_retention

FROM t

GROUP BY month\_

ORDER BY month\_;

1. Дана таблица формата:

|  |  |  |
| --- | --- | --- |
| date | customer\_id | account\_status |

*Account\_status* может принимать следующие значения:

новый

активирован

утилизирован

заблокирован

закрыт

**Задание**: Необходимо написать sql-запрос, в котором будет посчитано кол-во клиентов, сгруппированных по месяцам, с рабочим счетом (*новый, активированн, утилизирован*), и с нерабочим счетом (*заблокирован, закрыт*).

P.S. будет плюсом, если решение будет представлено несколькими способами (один из них – через оконные функции)

SELECT DATEPART(month,date) as month\_,

COUNT(CASE WHEN account\_status IN ('новый', 'активирован', 'утилизирован') THEN 1 ELSE null END) as valid,

COUNT(CASE WHEN account\_status IN ('заблокирован', 'закрыт') THEN 1 ELSE null END) as not\_valid

FROM test\_table

GROUP BY DATEPART(month,date)

или

SELECT DATEPART(month,date) as month\_,

COUNT(CASE WHEN account\_status IN ('новый', 'активирован', 'утилизирован') THEN 1 ELSE null END) OVER(PARTITION BY DATEPART(month,date)) as valid,

COUNT(CASE WHEN account\_status IN ('заблокирован', 'закрыт') THEN 1 ELSE null END) OVER(PARTITION BY DATEPART(month,date)) as not\_valid

FROM test\_table