Результаты выполнения задания №1

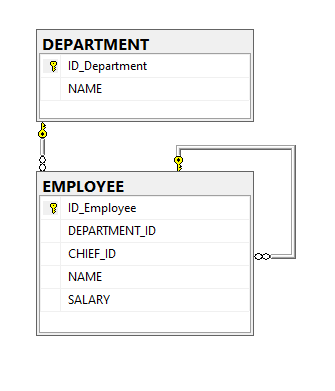


Рисунок 1 - Схема созданных таблиц

Обе таблицы были заполнены следующими данными.

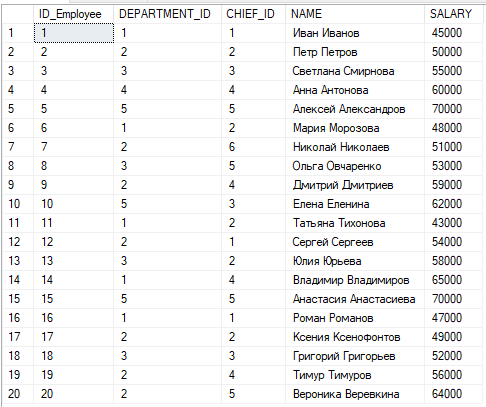


Рисунок 2 – Данные таблицы EMPLOYEE

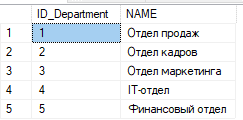


Рисунок 3 – Данные таблицы DEPARTMENT

Запросы:

1. Вывести список сотрудников, которые получают заработную плату ниже, чем у непосредственного руководителя.

Select e.ID\_Employee, e.DEPARTMENT\_ID, e.CHIEF\_ID, e.NAME, e.SALARY, c.SALARY as CHIEF\_SALARY

FROM EMPLOYEE e

LEFT JOIN EMPLOYEE c on e.CHIEF\_ID=c.ID\_Employee

WHERE e.SALARY<c.SALARY;

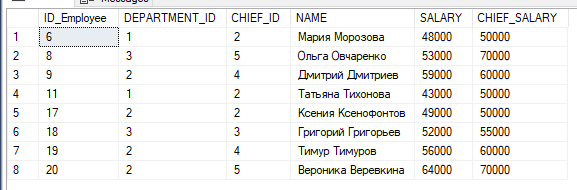


Рисунок 4 – Результат выполнение запроса 1

1. Вывести список сотрудников, которые получают в отделе минимальную заработную плату в своем отделе.

SELECT e.ID\_Employee, e.DEPARTMENT\_ID, e.CHIEF\_ID, e.NAME, e.SALARY

FROM EMPLOYEE e

WHERE e.SALARY = (Select min(SALARY) from EMPLOYEE Where DEPARTMENT\_ID = e.DEPARTMENT\_ID)

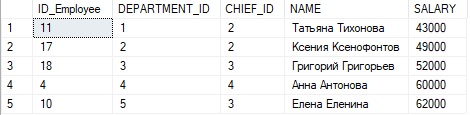


Рисунок 5 – Результат выполнение запроса 2

1. Вывести список ID отделов, количество сотрудников в которых не превышает трех человек.

SELECT DEPARTMENT\_ID

FROM Employee

GROUP BY DEPARTMENT\_ID

HAVING COUNT(ID\_Employee) <= 3;

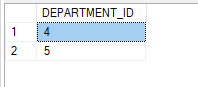


Рисунок 5 – Результат выполнение запроса 3

1. Вывести список сотрудников, не имеющих назначенного руководителя, который работал бы в том же отделе.

SELECT e.ID\_Employee, e.DEPARTMENT\_ID, e.CHIEF\_ID, e.NAME, e.SALARY

FROM EMPLOYEE e

LEFT JOIN EMPLOYEE c ON e.CHIEF\_ID = c.ID\_Employee

WHERE e.DEPARTMENT\_ID != c.DEPARTMENT\_ID;



Рисунок 5 – Результат выполнение запроса 4

1. Найти список ID отделов с максимальной суммарной заработной платой сотрудников.

SELECT DEPARTMENT\_ID

FROM Employee

GROUP BY DEPARTMENT\_ID

HAVING SUM(SALARY) = (

SELECT MAX(TotalSalary)

FROM (

SELECT SUM(SALARY) AS TotalSalary

FROM Employee

GROUP BY DEPARTMENT\_ID

) AS MaxTotalSalary

)

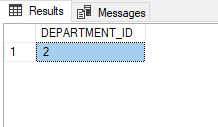


Рисунок 6 – Результат выполнение запроса 5

1. Составить SQL-запрос, вычисляющий сумму всех значений всех ЗП в конкретном столбце таблицы.

SELECT SUM(SALARY) AS TOTALSUM

FROM EMPLOYEE



Рисунок 7 – Результат выполнение запроса 6

Реализации хранимой процедуры UPDATESALARYFORDEPARTMENT.

Входные данные: ID отдела, PERCENT процент повышения ЗП

Логика: данная процедура должна у всех сотрудников в рамках отдела с заданным ID (кроме начальника отдела) повышать ЗП на заданный процент (PERCENT). В случае, если после повышения ЗП у кого-либо из сотрудников ЗП оказалась выше, чем у начальника отдела, то повысить ЗП для начальника до аналогичной ЗП. На выходе вернуть перечень сотрудников (все данные из таблицы employee) с обновленной и старой ЗП.

Примем, что начальником отдела является сотрудник, имеющий CHIEF\_ID = ID\_Employee.

Код реализации:

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[UPDATESALARYFORDEPARTMENT]

@ID int = 0,

@PERCENT float = 0

AS

BEGIN

SET NOCOUNT ON;

--Создание временной таблицы

CREATE TABLE TEMPEMPLOYEE

(

ID\_Employee int,

DEPARTMENT\_ID int,

CHIEF\_ID int,

NAME varchar(100),

NEW\_SALARY int,

OLD\_SALARY int,

);

-- Вставка данных в временную таблицу

INSERT INTO TEMPEMPLOYEE (ID\_Employee, DEPARTMENT\_ID, CHIEF\_ID, NAME , NEW\_SALARY , OLD\_SALARY)

SELECT

ID\_Employee,

DEPARTMENT\_ID,

CHIEF\_ID,

NAME,

Salary AS NewSalary,

Salary as OLD\_SALARY

FROM

EMPLOYEE

WHERE

DEPARTMENT\_ID = @ID;

--Обновление зарплат сотрудников

Update TEMPEMPLOYEE set NEW\_SALARY = NEW\_SALARY \* (1 + @PERCENT / 100) from TEMPEMPLOYEE where ID\_Employee != CHIEF\_ID;

--Зарплата начальника

DECLARE @CHIEF\_Salary int

Select @CHIEF\_Salary=OLD\_SALARY From TEMPEMPLOYEE where ID\_Employee = CHIEF\_ID;

-- Наибольшая зарплата сотрудников

DECLARE @MAX\_New\_Salary int

Select @MAX\_New\_Salary=MAX(NEW\_SALARY) From TEMPEMPLOYEE where ID\_Employee != CHIEF\_ID;

--Обновление зарплаты начальника

IF @CHIEF\_Salary< @MAX\_New\_Salary

Update TEMPEMPLOYEE set NEW\_SALARY = @MAX\_New\_Salary from TEMPEMPLOYEE where ID\_Employee = CHIEF\_ID;

--Обновление зарплат сотрудников

UPDATE EMPLOYEE

SET SALARY = TEMPEMPLOYEE.NEW\_SALARY

FROM EMPLOYEE

INNER JOIN TEMPEMPLOYEE ON EMPLOYEE.ID\_Employee = TEMPEMPLOYEE.ID\_Employee;

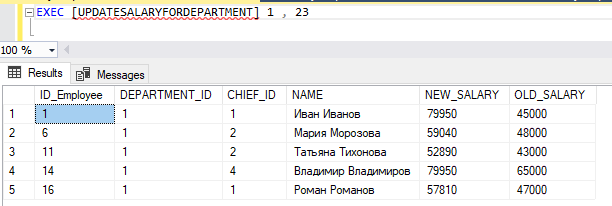
Select \* From TEMPEMPLOYEE;

--Удаление временной таблицы

drop table TEMPEMPLOYEE;

END

GO

  
Рисунок 8 – Пример выполнения процедуры

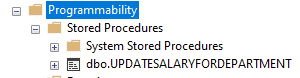


Рисунок 9 –Созданная процедура

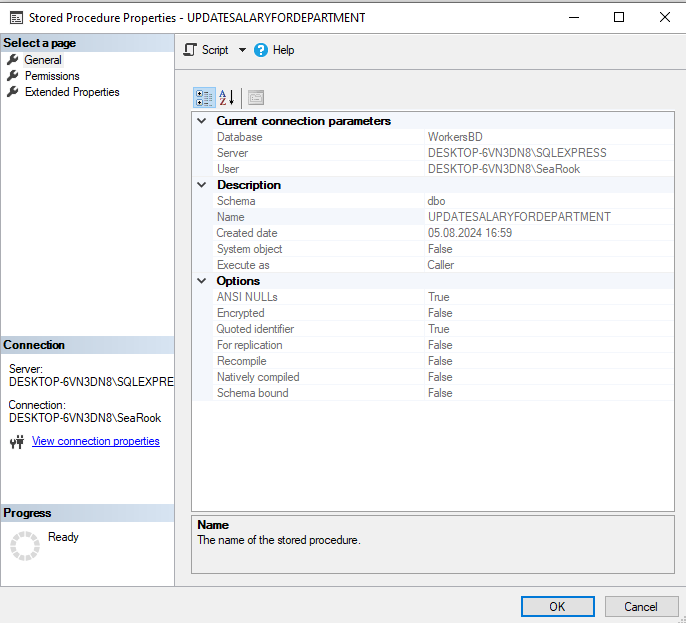


Рисунок 10 – Параметры процедуры