

Лабораторная работа №2

Вариант 2

Задание №1

1. Вывести на экран историю сотрудника, который работает на позиции 'Purchasing Manager'. В каких отделах компании он работал, с указанием периодов работы в каждом отделе:

```
SELECT emp.BusinessEntityID, JobTitle, dep.Name
      AS DepartmentName, StartDate, EndDate
FROM HumanResources.Employee AS emp
     INNER JOIN HumanResources.EmployeeDepartmentHistory AS eph
ON emp.BusinessEntityID = eph.BusinessEntityID
     INNER JOIN HumanResources.Department AS dep
ON eph.DepartmentID = dep.DepartmentID
WHERE JobTitle = 'Purchasing Manager';
GO
```

	BusinessEntityID	JobTitle	DepartmentName	StartDate	EndDate
1	250	Purchasing Manager	Marketing	2005-03-28	2005-08-30
2	250	Purchasing Manager	Quality Assurance	2005-08-31	2006-08-15
3	250	Purchasing Manager	Purchasing	2006-08-16	NULL

2. Вывести на экран список сотрудников, у которых почасовая ставка изменялась хотя бы один раз:

```
SELECT emp.BusinessEntityID, JobTitle, COUNT(*)
      AS RateCount
FROM HumanResources.EmployeePayHistory AS eph
     INNER JOIN HumanResources.Employee AS emp
ON emp.BusinessEntityID = eph.BusinessEntityID
GROUP BY emp.BusinessEntityID, JobTitle
HAVING COUNT(*) > 1;
GO
```

	BusinessEntityID	JobTitle	RateCount
1	4	Senior Tool Designer	3
2	16	Marketing Manager	3
3	167	Production Technician - WC30	3
4	170	Production Technician - WC30	3
5	172	Production Technician - WC30	3
6	174	Production Technician - WC30	3
7	175	Production Technician - WC30	3
8	176	Production Technician - WC30	3
9	177	Production Technician - WC30	3
10	178	Production Technician - WC30	3
11	224	Scheduling Assistant	3
12	234	Chief Financial Officer	3
13	250	Purchasing Manager	3

3. Вывести на экран максимальную почасовую ставку в каждом отделе. Вывести только актуальную информацию. Если сотрудник больше не работает в отделе — не учитывать такие данные:

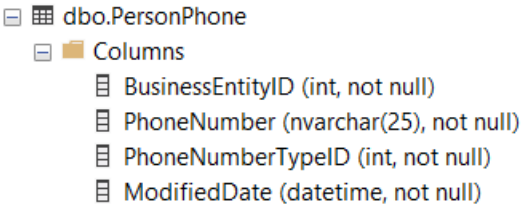
```
SELECT dep.DepartmentID, dep.Name, MAX(eph.Rate) AS MaxRate
FROM HumanResources.EmployeePayHistory AS eph
     INNER JOIN HumanResources.EmployeeDepartmentHistory AS edh
ON edh.BusinessEntityID = eph.BusinessEntityID
     INNER JOIN HumanResources.Department AS dep
ON dep.DepartmentID = edh.DepartmentID
WHERE EndDate IS NULL
GROUP BY dep.DepartmentID, dep.Name
ORDER BY dep.DepartmentID;
GO
```

	DepartmentID	Name	MaxRate
1	1	Engineering	63,4615
2	2	Tool Design	29,8462
3	3	Sales	72,1154
4	4	Marketing	37,50
5	5	Purchasing	30,00
6	6	Research and Development	50,4808
7	7	Production	84,1346
8	8	Production Control	24,5192
9	9	Human Resources	27,1394
10	10	Finance	43,2692
11	11	Information Services	50,4808
12	12	Document Control	17,7885
13	13	Quality Assurance	28,8462
14	14	Facilities and Maintenance	24,0385
15	15	Shipping and Receiving	19,2308
16	16	Executive	125,50

Задание №2

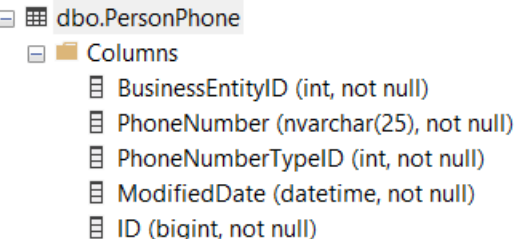
а) создайте таблицу `dbo.PersonPhone` с такой же структурой как `Person.PersonPhone`, не включая индексы, ограничения и триггеры;

```
CREATE TABLE dbo.PersonPhone (  
    BusinessEntityID INT NOT NULL,  
    PhoneNumber NVARCHAR(25) NOT NULL,  
    PhoneNumberTypeID INT NOT NULL,  
    ModifiedDate DATETIME NOT NULL  
);  
GO
```



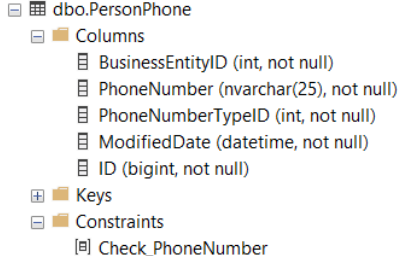
б) используя инструкцию `ALTER TABLE`, добавьте в таблицу `dbo.PersonPhone` новое поле `ID`, которое является уникальным ограничением `UNIQUE` типа `bigint` и имеет свойство `identity`. Начальное значение для поля `identity` задайте 2 и приращение задайте 2;

```
ALTER TABLE dbo.PersonPhone  
    ADD ID BIGINT IDENTITY(2,2) UNIQUE;  
GO
```



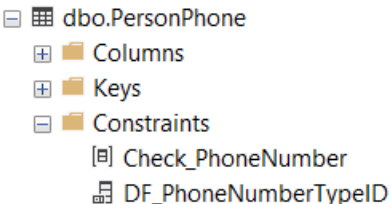
в) используя инструкцию `ALTER TABLE`, создайте для таблицы `dbo.PersonPhone` ограничение для поля `PhoneNumber`, запрещающее заполнение этого поля буквами;

```
ALTER TABLE dbo.PersonPhone  
    ADD CONSTRAINT Check_PhoneNumber  
    CHECK (PATINDEX('%[a-zA-Z]%', PhoneNumber) = 0);  
GO
```



г) используя инструкцию `ALTER TABLE`, создайте для таблицы `dbo.PersonPhone` ограничение `DEFAULT` для поля `PhoneNumberTypeID`, задайте значение по умолчанию 1;

```
ALTER TABLE dbo.PersonPhone  
    ADD CONSTRAINT DF_PhoneNumberTypeID  
    DEFAULT 1 FOR PhoneNumberTypeID;  
GO
```



е) заполните новую таблицу данными из `Person.PersonPhone`, где поле `PhoneNumber` не содержит символов '(' и ')' и только для тех сотрудников, которые существуют в таблице `HumanResources.Employee`, а их дата принятия на работу совпадает с датой начала работы в отделе;

```

INSERT INTO dbo.PersonPhone
SELECT
    pp.BusinessEntityID,
    pp.PhoneNumber,
    pp.PhoneNumberTypeID,
    pp.ModifiedDate
FROM Person.PersonPhone AS pp
    INNER JOIN HumanResources.Employee AS emp
ON pp.BusinessEntityID = emp.BusinessEntityID
    INNER JOIN HumanResources.EmployeeDepartmentHistory AS edh
ON emp.BusinessEntityID = edh.BusinessEntityID
WHERE PhoneNumber NOT LIKE '%[()]%' AND emp.HireDate = edh.StartDate;
GO

```

(286 rows affected)

Results		Messages				
	BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate	ID	
1	246	117-555-0185	3	2003-03-08 00:00:00.000	1410	
2	247	393-555-0186	3	2003-03-26 00:00:00.000	1412	
3	248	927-555-0150	3	2003-04-02 00:00:00.000	1414	
4	249	248-555-0134	3	2003-01-19 00:00:00.000	1416	
5	250	210-555-0193	3	2005-03-21 00:00:00.000	1418	
6	251	309-555-0170	3	2003-03-07 00:00:00.000	1420	
7	252	848-555-0163	1	2003-03-25 00:00:00.000	1422	

```

SELECT * FROM dbo.PersonPhone;
GO

```

f) измените поле PhoneNumber, разрешив добавление null значений.

```

ALTER TABLE dbo.PersonPhone
ALTER COLUMN PhoneNumber NVARCHAR(25) NULL;
GO

```

dbo.PersonPhone	
Columns	
BusinessEntityID	(int, not null)
PhoneNumber	(nvarchar(25), null)
PhoneNumberTypeID	(int, not null)
ModifiedDate	(datetime, not null)
ID	(bigint, not null)