

Лабораторная работа №2

Задание 1

1. Вывести на экран последнюю дату изменения почасовой ставки для каждого сотрудника.

```
-- Show last date of changing hourly rate for each worker
SELECT Employee.BusinessEntityID,
JobTitle, MAX(EmployeePayHistory.RateChangeDate) AS LastRateDate
FROM HumanResources.Employee
INNER JOIN HumanResources.EmployeePayHistory
ON HumanResources.Employee.BusinessEntityID = HumanResources.EmployeePayHistory.BusinessEntityID
GROUP BY Employee.BusinessEntityID, Employee.JobTitle;

-- Show number of years that each worker had worked in each department.If worker is working until now, number of years count until today
SELECT Employee.BusinessEntityID, JobTitle, Department.name as DepName, StartDate, EndDate,
(ISNULL(DATEPART(yy, EndDate), DATEPART(yy, GETDATE())) - DATEPART(yy, StartDate) ) as Years
FROM HumanResources.Employee
INNER JOIN HumanResources.EmployeeDepartmentHistory
ON HumanResources.Employee.BusinessEntityID = HumanResources.EmployeeDepartmentHistory.BusinessEntityID
INNER JOIN HumanResources.Department
ON HumanResources.EmployeeDepartmentHistory.DepartmentID = HumanResources.Department.DepartmentID;

-- Show information about all workers with info about department that they are working right now. Also show first word of department title
```

Results Messages

	BusinessEntit...	JobTitle	LastRateDate
1	1	Chief Executive Officer	2003-02-15 00:00:00.000
2	2	Vice President of Engineering	2002-03-03 00:00:00.000
3	3	Engineering Manager	2001-12-12 00:00:00.000
4	4	Senior Tool Designer	2006-01-15 00:00:00.000
5	5	Design Engineer	2002-02-06 00:00:00.000
6	6	Design Engineer	2002-02-24 00:00:00.000
7	7	Research and Development Manager	2003-03-12 00:00:00.000
8	8	Research and Development Engineer	2003-01-30 00:00:00.000
9	9	Research and Development Engineer	2003-02-17 00:00:00.000
10	10	Research and Development Manager	2003-06-04 00:00:00.000
11	11	Senior Tool Designer	2005-01-05 00:00:00.000
12	12	Tool Designer	2002-01-11 00:00:00.000
13	13	Tool Designer	2005-01-23 00:00:00.000
14	14	Senior Design Engineer	2005-01-30 00:00:00.000
15	15	Design Engineer	2005-02-18 00:00:00.000

Query executed successfully. MSI-CHUDUK\SQLEXPRESS (11.0... MSI-CHUDUK\Andrei_Chud... AdventureWorks2012 00:00:00 290 rows

2. Вывести на экран количество лет, которые каждый сотрудник проработал в каждом отделе. Если сотрудник работает в отделе по настоящее время, количество лет считайте до сегодняшнего дня.

```
-- Show number of years that each worker had worked in each department.If worker is working until now, number of years count until today
SELECT Employee.BusinessEntityID, JobTitle, Department.name as DepName, StartDate, EndDate,
(ISNULL(DATEPART(yy, EndDate), DATEPART(yy, GETDATE())) - DATEPART(yy, StartDate) ) as Years
FROM HumanResources.Employee
INNER JOIN HumanResources.EmployeeDepartmentHistory
ON HumanResources.Employee.BusinessEntityID = HumanResources.EmployeeDepartmentHistory.BusinessEntityID
INNER JOIN HumanResources.Department
ON HumanResources.EmployeeDepartmentHistory.DepartmentID = HumanResources.Department.DepartmentID;

-- Show information about all workers with info about department that they are working right now. Also show first word of department title
```

Results Messages

	BusinessEntit...	JobTitle	DepName	StartDate	EndDate	Years
1	1	Chief Executive Officer	Executive	2003-02-15	NULL	17
2	2	Vice President of Engineering	Engineering	2002-03-03	NULL	18
3	3	Engineering Manager	Engineering	2001-12-12	NULL	19
4	4	Senior Tool Designer	Engineering	2002-01-05	2004-06-30	2
5	4	Senior Tool Designer	Tool Design	2004-07-01	NULL	16
6	5	Design Engineer	Engineering	2002-02-06	NULL	18
7	6	Design Engineer	Engineering	2002-02-24	NULL	18
8	7	Research and Development Manager	Research and Development	2003-03-12	NULL	17
9	8	Research and Development Engineer	Research and Development	2003-01-30	NULL	17
10	9	Research and Development Engineer	Research and Development	2003-02-17	NULL	17
11	10	Research and Development Manager	Research and Development	2003-06-04	NULL	17
12	11	Senior Tool Designer	Tool Design	2005-01-05	NULL	15
13	12	Tool Designer	Tool Design	2002-01-11	NULL	18
14	13	Tool Designer	Tool Design	2005-01-23	NULL	15
15	14	Senior Design Engineer	Engineering	2005-01-30	NULL	15

Query executed successfully. MSI-CHUDUK\SQLEXPRESS (11.0... MSI-CHUDUK\Andrei_Chud... AdventureWorks2012 00:00:00 296 rows

3. Вывести на экран информацию обо всех сотрудниках, с указанием отдела, в котором они работают в настоящий момент. Вывести также первое слово из названия группы отделов.

```
-- Show information about all workers with info about department that they are working right now. Also show first word of department title
SELECT Employee.BusinessEntityID, JobTitle, Department.name as DepName, Department.GroupName, SUBSTRING(Department.GroupName, 0, CHARINDEX(' ', Department.GroupName)) as FirstWord
FROM HumanResources.Employee
INNER JOIN HumanResources.EmployeeDepartmentHistory
ON HumanResources.Employee.BusinessEntityID = HumanResources.EmployeeDepartmentHistory.BusinessEntityID
INNER JOIN HumanResources.Department
ON HumanResources.EmployeeDepartmentHistory.DepartmentID = HumanResources.Department.DepartmentID
AND EmployeeDepartmentHistory.EndDate IS NULL;
```

100 %

Results Messages

	BusinessEntit...	JobTitle	DepName	GroupName	(No column na...
1	1	Chief Executive Officer	Executive	Executive General and Administration	Executive
2	2	Vice President of Engineering	Engineering	Research and Development	Research
3	3	Engineering Manager	Engineering	Research and Development	Research
4	4	Senior Tool Designer	Tool Design	Research and Development	Research
5	5	Design Engineer	Engineering	Research and Development	Research
6	6	Design Engineer	Engineering	Research and Development	Research
7	7	Research and Development Manager	Research and Development	Research and Development	Research
8	8	Research and Development Engineer	Research and Development	Research and Development	Research
9	9	Research and Development Engineer	Research and Development	Research and Development	Research
10	10	Research and Development Manager	Research and Development	Research and Development	Research
11	11	Senior Tool Designer	Tool Design	Research and Development	Research
12	12	Tool Designer	Tool Design	Research and Development	Research
13	13	Tool Designer	Tool Design	Research and Development	Research
14	14	Senior Design Engineer	Engineering	Research and Development	Research
15	15	Design Engineer	Engineering	Research and Development	Research

Query executed successfully. MSI-CHUDUK\SQLEXPRESS (11.0... MSI-CHUDUK\Andrei_Chud... AdventureWorks2012 00:00:00 290 rows

Задание 2

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

The screenshot shows the SQL Server Enterprise Manager on the left, displaying the database structure for AdventureWorks2012. The 'Tables' folder is expanded, showing the 'dbo.PersonPhone' table. The table structure is as follows:

Column Name	Data Type	Nullable
BusinessEntityID	int	not null
PhoneNumber	nvarchar(25)	not null
PhoneNumberTypeID	int	not null
ModifiedDate	datetime	not null

The SQL Server Enterprise Query Editor on the right shows the following T-SQL script:

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

-- b) Using instruction ALTER TABLE create for table dbo.PersonPhone complex primary key from fields BusinessEntityID and PhoneNumber
ALTER TABLE dbo.PersonPhone
ADD CONSTRAINT PK_PersonPhone PRIMARY KEY (BusinessEntityID, PhoneNumber);

-- c) Using instruction ALTER TABLE create for table dbo.PersonPhone new field PostalCode nvarchar(15) and constraint for that field banis
ALTER TABLE dbo.PersonPhone
ADD PostalCode nvarchar(15),
CONSTRAINT Check_PostalCode CHECK (PostalCode NOT LIKE '% A-Za-z %');

-- d) Using instruction ALTER TABLE create for table dbo.PersonPhone constraint DEFAULT for field PostalCode and make default value = '0'
ALTER TABLE dbo.PersonPhone
ADD CONSTRAINT Def_PostalCode DEFAULT '0' FOR PostalCode;

-- e) Fill new table with data from Person.PersonPhone but with contacts with type 'Cell' from table PhoneNumberType
INSERT INTO dbo.PersonPhone
SELECT p.BusinessEntityID, p.PhoneNumber, p.PhoneNumberTypeID, p.ModifiedDate
FROM Person.PersonPhone p
JOIN PhoneNumberType t ON p.PhoneNumberTypeID = t.PhoneNumberTypeID
WHERE t.Name = 'Cell';
```

The Messages pane at the bottom shows the command(s) completed successfully.

- б) используя инструкцию `ALTER TABLE`, создайте для таблицы `dbo.PersonPhone` составной первичный ключ из полей `BusinessEntityID` и `PhoneNumber`

The screenshot shows the SQL Server Enterprise Manager on the left, displaying the database structure for AdventureWorks2012. The 'Tables' folder is expanded, showing the 'dbo.PersonPhone' table. The table structure is as follows:

Column Name	Data Type	Nullable
BusinessEntityID	int	not null
PhoneNumber	nvarchar(25)	not null
PhoneNumberTypeID	int	not null
ModifiedDate	datetime	not null

The SQL Server Enterprise Query Editor on the right shows the following T-SQL script:

```
-- b) Using instruction ALTER TABLE create for table dbo.PersonPhone complex primary key from fields BusinessEntityID and PhoneNumber
ALTER TABLE dbo.PersonPhone
ADD CONSTRAINT PK_PersonPhone PRIMARY KEY (BusinessEntityID, PhoneNumber);

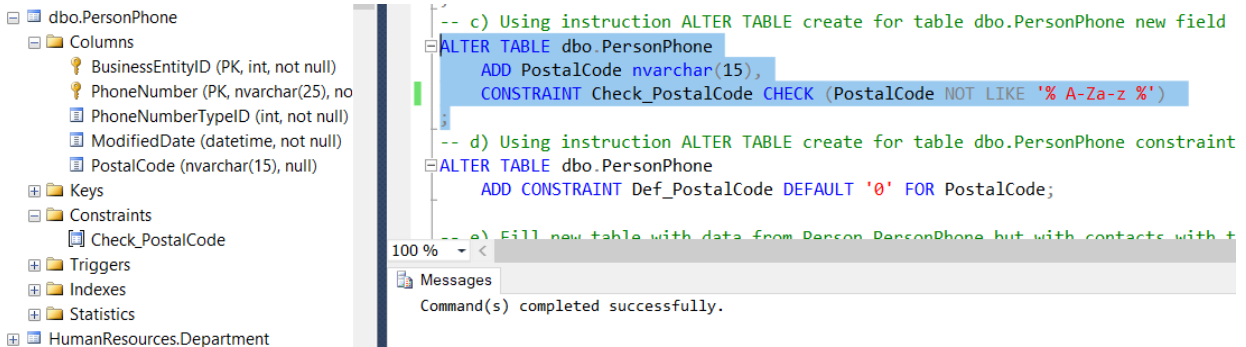
-- c) Using instruction ALTER TABLE create for table dbo.PersonPhone new field PostalCode nvarchar(15) and constraint for that field banis
ALTER TABLE dbo.PersonPhone
ADD PostalCode nvarchar(15),
CONSTRAINT Check_PostalCode CHECK (PostalCode NOT LIKE '% A-Za-z %');

-- d) Using instruction ALTER TABLE create for table dbo.PersonPhone constraint DEFAULT for field PostalCode and make default value = '0'
ALTER TABLE dbo.PersonPhone
ADD CONSTRAINT Def_PostalCode DEFAULT '0' FOR PostalCode;

-- e) Fill new table with data from Person.PersonPhone but with contacts with type 'Cell' from table PhoneNumberType
INSERT INTO dbo.PersonPhone
SELECT p.BusinessEntityID, p.PhoneNumber, p.PhoneNumberTypeID, p.ModifiedDate
FROM Person.PersonPhone p
JOIN PhoneNumberType t ON p.PhoneNumberTypeID = t.PhoneNumberTypeID
WHERE t.Name = 'Cell';
```

The Messages pane at the bottom shows the command(s) completed successfully.

- с) используя инструкцию ALTER TABLE, создайте для таблицы dbo.PersonPhone новое поле PostalCode nvarchar(15) и ограничение для этого поля, запрещающее заполнение этого поля буквами



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'dbo.PersonPhone' table is selected, and its columns are listed: BusinessEntityID (PK, int, not null), PhoneNumber (PK, nvarchar(25), not null), PhoneNumberTypeID (int, not null), ModifiedDate (datetime, not null), and PostalCode (nvarchar(15), null). The 'Constraints' folder is expanded, showing 'Check_PostalCode'. The main pane displays the following SQL commands:

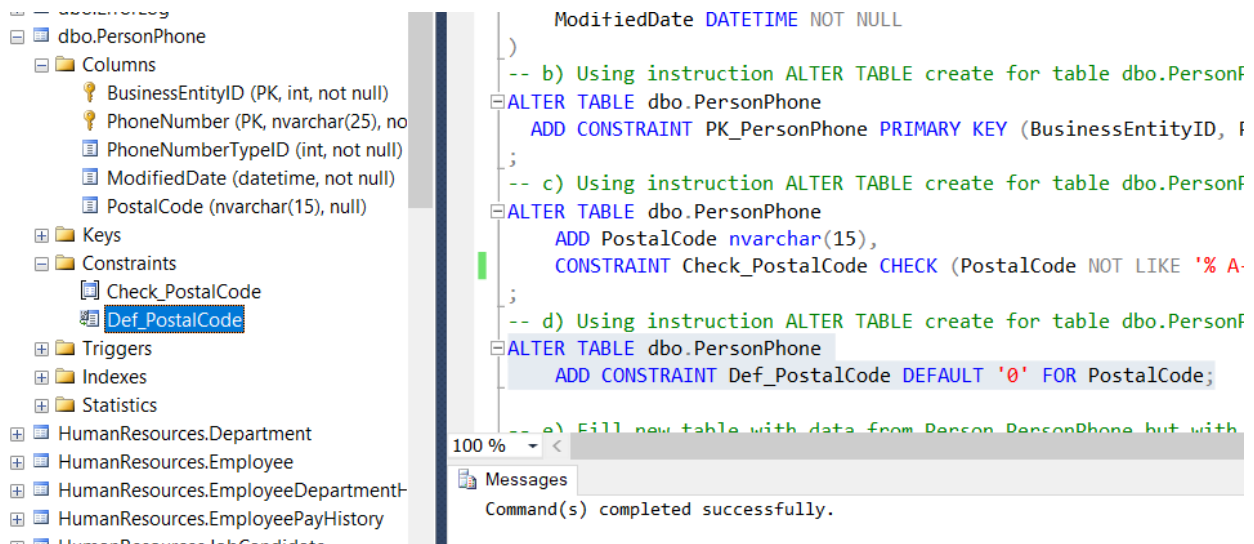
```
-- c) Using instruction ALTER TABLE create for table dbo.PersonPhone new field
ALTER TABLE dbo.PersonPhone
ADD PostalCode nvarchar(15),
CONSTRAINT Check_PostalCode CHECK (PostalCode NOT LIKE '% A-Za-z %')

-- d) Using instruction ALTER TABLE create for table dbo.PersonPhone constraint
ALTER TABLE dbo.PersonPhone
ADD CONSTRAINT Def_PostalCode DEFAULT '0' FOR PostalCode;

-- e) Fill new table with data from Person.PersonPhone but with contacts with t
```

The 'Messages' pane at the bottom shows 'Command(s) completed successfully.'

- d) используя инструкцию ALTER TABLE, создайте для таблицы dbo.PersonPhone ограничение DEFAULT для поля PostalCode, задайте значение по умолчанию '0'



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'dbo.PersonPhone' table is selected, and its columns are listed: BusinessEntityID (PK, int, not null), PhoneNumber (PK, nvarchar(25), not null), PhoneNumberTypeID (int, not null), ModifiedDate (datetime, not null), and PostalCode (nvarchar(15), null). The 'Constraints' folder is expanded, showing 'Check_PostalCode' and 'Def_PostalCode'. The main pane displays the following SQL commands:

```
ModifiedDate DATETIME NOT NULL

-- b) Using instruction ALTER TABLE create for table dbo.PersonPhone
ALTER TABLE dbo.PersonPhone
ADD CONSTRAINT PK_PersonPhone PRIMARY KEY (BusinessEntityID, PhoneNumber);

-- c) Using instruction ALTER TABLE create for table dbo.PersonPhone
ALTER TABLE dbo.PersonPhone
ADD PostalCode nvarchar(15),
CONSTRAINT Check_PostalCode CHECK (PostalCode NOT LIKE '% A-Za-z %');

-- d) Using instruction ALTER TABLE create for table dbo.PersonPhone
ALTER TABLE dbo.PersonPhone
ADD CONSTRAINT Def_PostalCode DEFAULT '0' FOR PostalCode;

-- e) Fill new table with data from Person.PersonPhone but with contacts with t
```

The 'Messages' pane at the bottom shows 'Command(s) completed successfully.'

- е) заполните новую таблицу данными из Person.PersonPhone, только контактами с типом 'Cell' из таблицы PhoneNumberType

```
-- e) Fill new table with data from Person.PersonPhone but with contacts with type 'Cell' from table PhoneNumberType
INSERT INTO dbo.PersonPhone(
    BusinessEntityID,
    PhoneNumber,
    PhoneNumberTypeID,
    ModifiedDate
)
SELECT PersonPhone.BusinessEntityID, PersonPhone.PhoneNumber, PersonPhone.PhoneNumberTypeID, PersonPhone.ModifiedDate
FROM Person.PersonPhone
INNER JOIN Person.PhoneNumberType as PhoneNumberType
ON PhoneNumberType.PhoneNumberTypeID = Person.PersonPhone.PhoneNumberTypeID
WHERE PhoneNumberType.Name = 'Cell';

(10096 row(s) affected)
```

```
/****** Script for SelectTopNRows command from SSMS *****/
SELECT TOP 1000 [BusinessEntityID]
, [PhoneNumber]
, [PhoneNumberTypeID]
, [ModifiedDate]
, [PostalCode]
FROM [AdventureWorks2012].[dbo].[PersonPhone]
```

	BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate	PostalCode
1	697-555-0142	1	2003-02-08 00:00:00.000	0	
3	212-555-0187	1	2001-12-05 00:00:00.000	0	
4	612-555-0100	1	2001-12-29 00:00:00.000	0	
5	849-555-0139	1	2002-01-30 00:00:00.000	0	
8	815-555-0138	1	2003-01-23 00:00:00.000	0	
9	185-555-0186	1	2003-02-10 00:00:00.000	0	
11	719-555-0181	1	2004-12-29 00:00:00.000	0	
14	465-555-0156	1	2005-01-23 00:00:00.000	0	
15	970-555-0138	1	2005-02-11 00:00:00.000	0	
17	150-555-0189	1	2001-02-19 00:00:00.000	0	
19	124-555-0114	1	2005-03-10 00:00:00.000	0	
21	138-555-0118	1	2003-03-27 00:00:00.000	0	
27	632-555-0129	1	2002-03-23 00:00:00.000	0	
30	955-555-0169	1	2003-02-23 00:00:00.000	0	
31	818-555-0128	1	2003-01-29 00:00:00.000	0	
32	314-555-0113	1	2003-01-23 00:00:00.000	0	
34	753-555-0129	1	2003-03-13 00:00:00.000	0	
35	429-555-0137	1	2003-03-05 00:00:00.000	0	
36	587-555-0115	1	2003-03-07 00:00:00.000	0	
37	315-555-0144	1	2003-04-01 00:00:00.000	0	
38	208-555-0114	1	2004-02-10 00:00:00.000	0	
41	712-555-0113	1	2003-02-15 00:00:00.000	0	
42	119-555-0117	1	2003-01-21 00:00:00.000	0	
43	970-555-0118	1	2003-01-27 00:00:00.000	0	
45	278-555-0118	1	2003-03-23 00:00:00.000	0	
46	173-555-0179	1	2003-03-05 00:00:00.000	0	
47	908-555-0159	1	2003-03-19 00:00:00.000	0	

Query executed successfully. MSI-CHUDUK\SQLEXPRESS (11.0.5512.1) | MSI-CHUDUK\Andrei_Chud... | master | 00:00:00 | 1000 rows

f) измените тип поля PhoneNumberTypeID на bigint и допускающим NULL значения

g)

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Columns' folder for the 'dbo.PersonPhone' table is expanded, showing the following columns: BusinessEntityID (PK, int, not null), PhoneNumber (PK, nvarchar(25), not null), PhoneNumberTypeID (bigint, null), ModifiedDate (datetime, not null), and PostalCode (nvarchar(15), null). The 'PhoneNumberTypeID' column is highlighted. The main pane shows a T-SQL script with the following content:

```
SELECT PersonPhone.BusinessEntityID, PersonPhone.PhoneNumber, PersonPhone.PhoneN
FROM Person.PersonPhone
INNER JOIN Person.PhoneNumberType as PhoneNumberType
ON PhoneNumberType.PhoneNumberTypeID = Person.PersonPhone.PhoneNumberTypeID
WHERE PhoneNumberType.Name = 'Cell';

-- f) change type of field PhoneNumberID to BIGINT and possible to be null
ALTER TABLE dbo.PersonPhone
ALTER COLUMN PhoneNumberTypeID BIGINT NULL;
```

At the bottom, the 'Messages' pane shows the command completed successfully.