

УО «Белорусский государственный университет информатики и  
радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе № 2

по предмету

«Базы данных»

Выполнил:  
Дубровинский А. В.  
группа 751001

Проверил:  
Салей О. А.

Минск 2020

## Вариант 2

### Задание №1

Операции объединений INNER JOIN с применением встроенных функций SQL Server без использования подзапросов. Задания направлены на выборку данных из таблиц, принадлежащих схеме HumanResources.

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

SQL запрос:

```
-- 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 [EmployeeDepartmentHistory].[BusinessEntityID],
       [Employee].[JobTitle],
       [Department].[Name] AS [DepartmentName],
       [EmployeeDepartmentHistory].[StartDate],
       [EmployeeDepartmentHistory].[EndDate]

FROM [HumanResources].[EmployeeDepartmentHistory]

INNER JOIN [HumanResources].[Employee]
    ON [EmployeeDepartmentHistory].[BusinessEntityID] = [Employee].[BusinessEntityID]

INNER JOIN [HumanResources].[Department]
    ON [EmployeeDepartmentHistory].[DepartmentID] = [Department].[DepartmentID]

WHERE [Employee].[JobTitle] = N'Purchasing Manager';

GO
```

Результат выполнения:

	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

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

SQL запрос:

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

```
SELECT [Employee].[BusinessEntityID],  
       [Employee].[JobTitle],  
       COUNT([EmployeePayHistory].[Rate]) AS [RateCount]  
  
FROM [HumanResources].[Employee]  
  
INNER JOIN [HumanResources].[EmployeePayHistory]  
    ON [Employee].[BusinessEntityID] = [EmployeePayHistory].[BusinessEntityID]  
  
GROUP BY [Employee].[BusinessEntityID], [Employee].[JobTitle]  
  
HAVING COUNT([EmployeePayHistory].[Rate]) >= 2;  
GO
```

Результат выполнения:

	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

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

SQL запрос:

```
-- 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 [Department].[DepartmentID],  
       [Department].[Name],  
       MAX([EmployeePayHistory].[Rate]) AS [MaxRate]  
  
FROM [HumanResources].[EmployeePayHistory]  
  
INNER JOIN [HumanResources].[Employee]  
    ON [Employee].[BusinessEntityID] = [EmployeePayHistory].[BusinessEntityID]  
  
INNER JOIN [HumanResources].[EmployeeDepartmentHistory]  
    ON [Employee].[BusinessEntityID] = [EmployeeDepartmentHistory].[BusinessEntityID]  
  
INNER JOIN [HumanResources].[Department]  
    ON [EmployeeDepartmentHistory].[DepartmentID] = [Department].[DepartmentID]  
  
WHERE [EmployeeDepartmentHistory].[EndDate] IS NULL  
  
GROUP BY [Department].[DepartmentID], [Department].[Name];  
GO
```

Результат выполнения:

	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

Создание и изменение таблиц, первичных ключей, ограничений. Оператор INSERT. Задания направлены на выборку данных из таблиц, принадлежащих схеме HumanResources и Person.

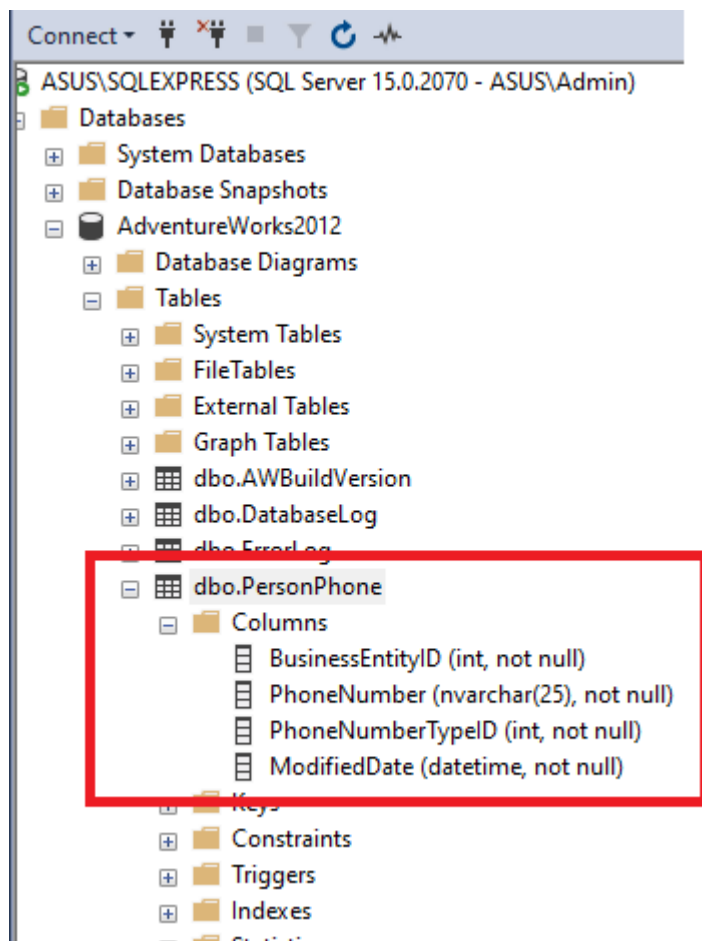
- а) создайте таблицу dbo.PersonPhone с такой же структурой как Person.PersonPhone, не включая индексы, ограничения и триггеры.

SQL запрос:

```
-- 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
```

Результат выполнения:



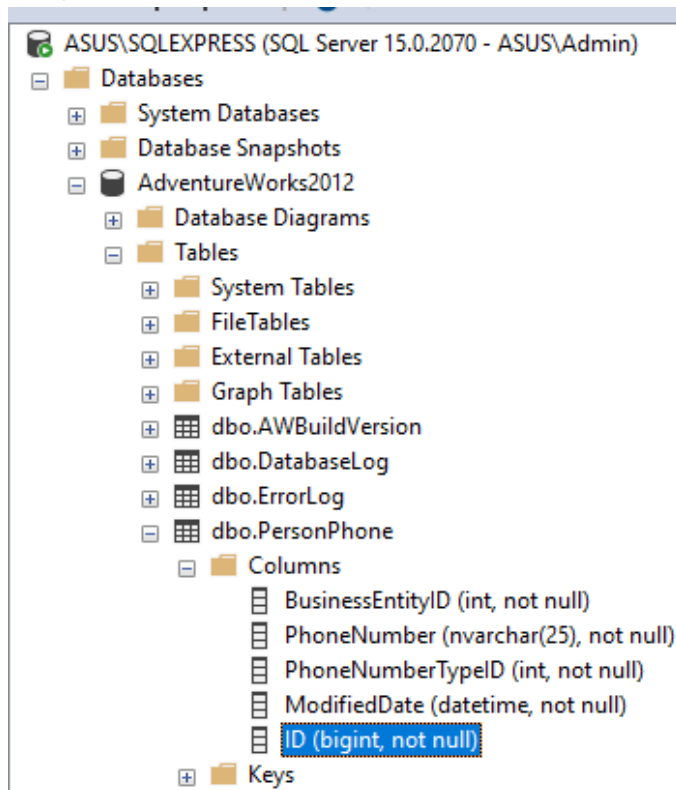
- b) используя инструкцию ALTER TABLE, добавьте в таблицу dbo.PersonPhone новое поле ID, которое является уникальным ограничением UNIQUE типа bigint и имеет свойство identity. Начальное значение для поля identity задайте 2 и приращение задайте 2;

SQL запрос:

```
-- b) 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 UNIQUE IDENTITY(2, 2);
GO
```

Результат выполнения:



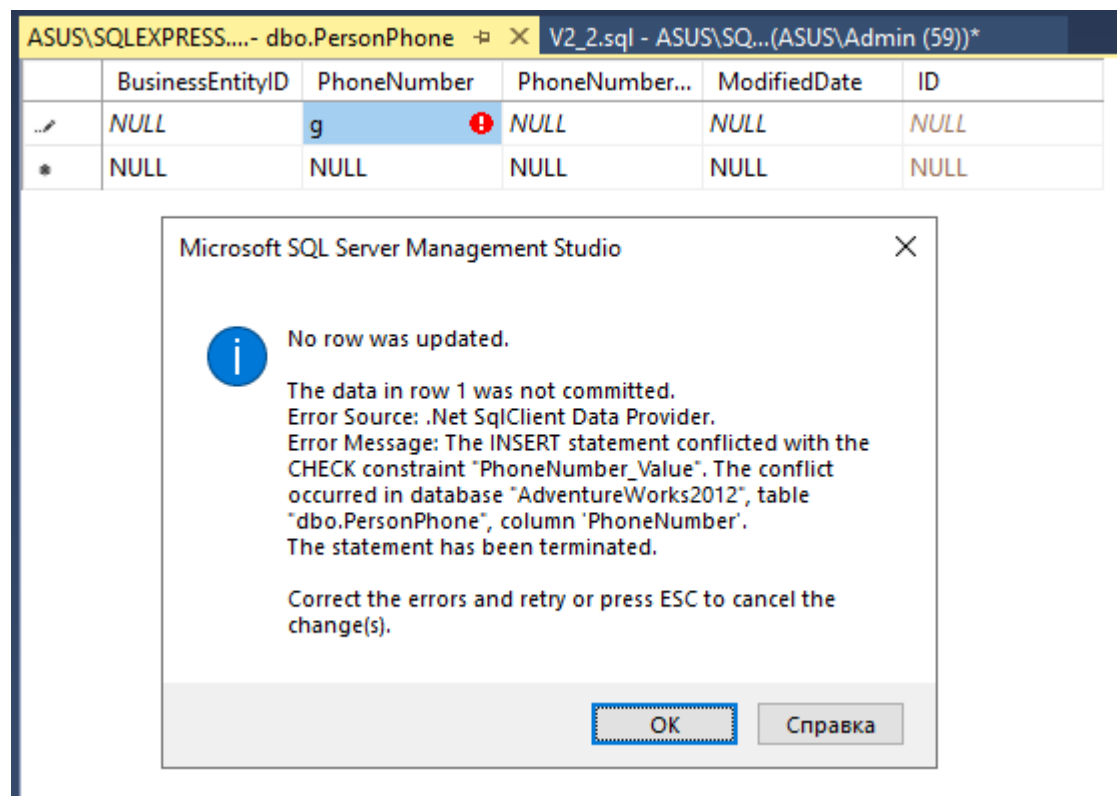
- c) используя инструкцию ALTER TABLE, создайте для таблицы dbo.PersonPhone ограничение для поля PhoneNumber, запрещающее заполнение этого поля буквами;

SQL запрос:

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

ALTER TABLE [dbo].[PersonPhone]
    ADD CONSTRAINT [PhoneNumber_Value] CHECK (PATINDEX('[a-zA-Za-яA-Я]', [PhoneNumber]) = 0);
GO
```

Результат выполнения:



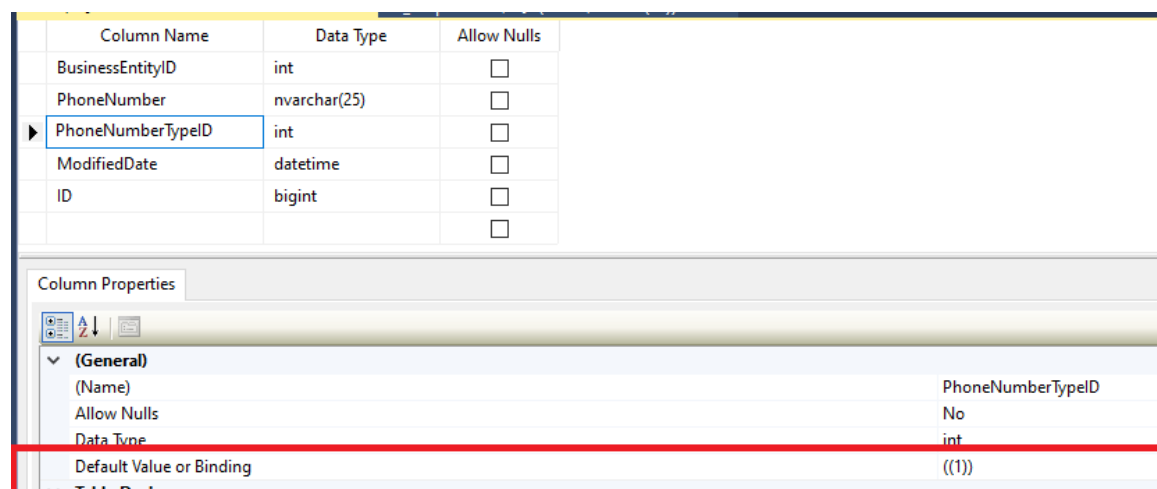
- d) используя инструкцию ALTER TABLE, создайте для таблицы dbo.PersonPhone ограничение DEFAULT для поля PhoneNumberTypeID, задайте значение по умолчанию 1;

SQL запрос:

```
-- d) 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 [PhoneNumberTypeID_Default] DEFAULT 1 FOR [PhoneNumberTypeID];
GO
```

Результат выполнения:



- 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;

SQL запрос:

```
-- 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] (
    [BusinessEntityID],
    [PhoneNumber],
    [PhoneNumberTypeID],
    [ModifiedDate]
)
SELECT
    [PersonPhone].[BusinessEntityID],
    [PersonPhone].[PhoneNumber],
    [PersonPhone].[PhoneNumberTypeID],
    [PersonPhone].[ModifiedDate]
FROM [Person].[PersonPhone]

INNER JOIN [HumanResources].[Employee]
    ON [Person].[PersonPhone].[BusinessEntityID] = [HumanResources].[Employee].[BusinessEntityID]

INNER JOIN [HumanResources].[EmployeeDepartmentHistory]
    ON [Employee].[BusinessEntityID] = [EmployeeDepartmentHistory].[BusinessEntityID]

WHERE [Employee].[HireDate] = [EmployeeDepartmentHistory].[StartDate]
    AND [PersonPhone].[PhoneNumber] NOT LIKE '%(%)%';

GO
```

SQL запрос для вывода результата:

```
-- Получение таблицы, 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;

SELECT [PersonPhone].[BusinessEntityID],
    [PersonPhone].[PhoneNumber],
    [PersonPhone].[PhoneNumberTypeID],
    [PersonPhone].[ID],
    [EmployeeDepartmentHistory].[StartDate],
    [Employee].[HireDate]

FROM [dbo].[PersonPhone]

INNER JOIN [HumanResources].[Employee]
    ON [dbo].[PersonPhone].[BusinessEntityID] = [HumanResources].[Employee].[BusinessEntityID]

INNER JOIN [HumanResources].[EmployeeDepartmentHistory]
    ON [Employee].[BusinessEntityID] = [EmployeeDepartmentHistory].[BusinessEntityID];

GO
```



Результат выполнения:

Results		Messages				
	BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ID	StartDate	HireDate
1	1	697-555-0142	1	2	2003-02-15	2003-02-15
2	2	819-555-0175	3	4	2002-03-03	2002-03-03
3	3	212-555-0187	1	6	2001-12-12	2001-12-12
4	4	612-555-0100	1	8	2002-01-05	2002-01-05
5	4	612-555-0100	1	8	2004-07-01	2002-01-05
6	5	849-555-0139	1	10	2002-02-06	2002-02-06
7	6	122-555-0189	3	12	2002-02-24	2002-02-24
8	7	181-555-0156	3	14	2003-03-12	2003-03-12
9	8	815-555-0138	1	16	2003-01-30	2003-01-30
10	9	185-555-0186	1	18	2003-02-17	2003-02-17
11	10	330-555-2568	3	20	2003-06-04	2003-06-04
12	11	719-555-0181	1	22	2005-01-05	2005-01-05
13	12	168-555-0183	3	24	2002-01-11	2002-01-11
14	13	473-555-0117	3	26	2005-01-23	2005-01-23
15	14	465-555-0156	1	28	2005-01-30	2005-01-30
16	15	970-555-0138	1	30	2005-02-18	2005-02-18
17	16	913-555-0172	3	32	2002-01-20	2002-01-20
18	16	913-555-0172	3	32	2003-08-16	2002-01-20
19	17	150-555-0189	1	34	2001-02-26	2001-02-26
20	18	486-555-0150	3	36	2005-03-10	2005-03-10
21	19	124-555-0114	1	38	2005-03-17	2005-03-17

f) измените поле PhoneNumber, разрешив добавление null значений.

SQL запрос:

-- f) измените поле PhoneNumber, разрешив добавление null значений.

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

Результат выполнения:

ASUS\SQLSERVER...- dbo.PersonPhone			
V2_2.sql - ASUS\SQL... (ASUS			
	Column Name	Data Type	Allow Nulls
▶	BusinessEntityID	int	<input type="checkbox"/>
	PhoneNumber	nvarchar(25)	<input checked="" type="checkbox"/>
	PhoneNumberTypeID	int	<input type="checkbox"/>
	ModifiedDate	datetime	<input type="checkbox"/>
	ID	bigint	<input type="checkbox"/>